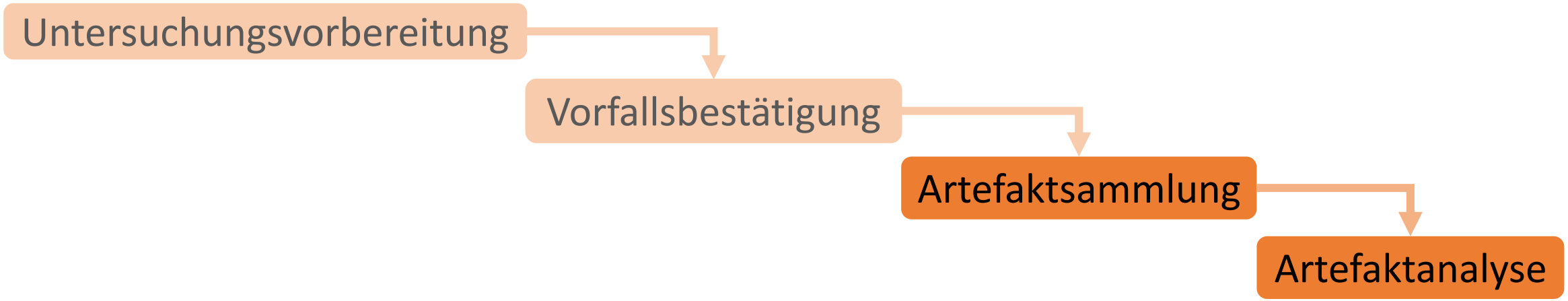
ForensikVorgaenge-Db & SQL-Injection

Vorgehensmodell: Vorgehen bei einer forensischen Db-Untersuchung nach Kevvie Fowler

Agenda



1. Vorgehen
2. MySQL, PostgreSQL *(Web-App: Flask [Python])*
3. Cloud: MS Azure mit SQL-Server *(Web-App: Razor [C#])*
4. Fazit

1. Vorgehen

Was haben wir gemacht und wie sind wir vorgegangen?

1) Vorbereitung

2) Datenbank anlegen und initialisieren

3) Bei Cloud: Webapplikation entwickeln

3) SQL-Injection Angriffe durchführen

4) Forensische Auswertung (Artefaktsammlung & -analyse)

Docker: Allgemeines

Flask [Python]

2. Konfiguration Docker

- Ursprüngliche Inbetriebnahme problemlos

- Anpassungen nötig:

```
    if current_db == 'mysql':
        result = session.execute(f"SELECT P_Vorg_Nr,
Aktenzeichen, Einlaufdatum, Auslaufdatum FROM
T_Forensik_Vorgaenge WHERE P_Vorg_Nr LIKE '%{search}%' ORDER BY
P_Vorg_Nr")

    elif current_db == 'postgres':
        result = session.execute(f"SELECT CAST (P_Vorg_Nr AS
VARCHAR), Aktenzeichen, Einlaufdatum, Auslaufdatum FROM
T_Forensik_Vorgaenge WHERE CAST (P_Vorg_Nr AS VARCHAR) LIKE '%-
{search}%' ORDER BY P_Vorg_Nr")
```

```
logging.basicConfig(filename='record.log', level=logging.DEBUG,
format=f'%(asctime)s %(levelname)s %(name)s %(threadName)s : %
(message)s')
```

Vorgänge Tools

Vorgänge

Vorgangsnummer Suchen

Vorgangsnummer	Aktenzeichen	Einlaufdatum	Auslaufdatum
2021100	XX736145-20	2020-04-29	2020-05-10
2021101	XX736145-20	2020-09-17	2020-09-22
2021102	XX610814-20	2020-10-19	2020-10-31
2021103	XX666115-20	2020-12-28	None

2. DB Erstellung

- .slq Skript für PostgreSQL aus Semester 4 ohne Änderungen anwendbar
- Für MySQL kleinere Änderungen nötigt (z. B. Case-Sensitivity)

2. DB Konfiguration

- PostgreSQL: logging_collector = on

- MySQL: general_log aktivieren
  Binary Logs bereits aktiviert aber Modus "ROW"
  -> Modus zu "MIXED" ändern

PostgreSQL & MySQL

Flask [Python]

2. PostgreSQL - Szenario

- John G. ist Beschuldigter in einem Verfahren
- Er will seinen Namen aus der Datenbank tilgen
- John G. ist ein Opportunist

2. PostgreSQL - Aufarbeitung

- PostgreSQL Log (/var/lib/postgresql/data/log/...)

```
2022-01-22 12:19:42.342 UTC [37] LOG:  statement: BEGIN
2022-01-22 12:19:42.343 UTC [37] LOG:  statement: SELECT CAST (P_Vorg_Nr AS VARCHAR), Aktenzeichen, Einlaufdatum, Ausla
ufdatum FROM T_Forensik_Vorgaenge WHERE CAST (P_Vorg_Nr AS VARCHAR) LIKE '%'; SELECT  version();-- %' ORDER BY P_Vorg_N
r
2022-01-22 12:21:42.777 UTC [51] LOG:  statement: BEGIN
2022-01-22 12:21:42.780 UTC [51] LOG:  statement: SELECT CAST (P_Vorg_Nr AS VARCHAR), Aktenzeichen, Einlaufdatum, Ausla
ufdatum FROM T_Forensik_Vorgaenge WHERE CAST (P_Vorg_Nr AS VARCHAR) LIKE '%%' ORDER BY P_Vorg_Nr
2022-01-22 12:21:47.963 UTC [52] LOG:  statement: BEGIN
2022-01-22 12:21:47.963 UTC [52] LOG:  statement: SELECT CAST (P_Vorg_Nr AS VARCHAR), Aktenzeichen, Einlaufdatum, Ausla
ufdatum FROM T_Forensik_Vorgaenge WHERE CAST (P_Vorg_Nr AS VARCHAR) LIKE '%2021101%' ORDER BY P_Vorg_Nr
```

- Docker Logs (..:#docker logs $Containername)

```
172.31.0.1 - - [22/Jan/2022 12:30:31] "GET /favicon.ico HTTP/1.1" 404 -
172.31.0.1 - - [22/Jan/2022 12:30:32] "GET /vorgaenge HTTP/1.1" 200 -
172.31.0.1 - - [22/Jan/2022 12:30:34] "GET /vorgaenge HTTP/1.1" 200 -
172.31.0.1 - - [22/Jan/2022 12:36:50] "GET /vorgaenge?search=%27%3B+SELECT++version%28%29%3B-- HTTP/1.1" 200 -
172.31.0.1 - - [22/Jan/2022 12:37:02] "GET /vorgaenge?search=%27%3B+SELECT+P_Pers_Nr%2C+Vname%2C+Nname%2C+Geburts
FROM+T_Personen%2C+T_Beschuldigte+WHERE+P_Pers_Nr+%3D+PF_Beschuldigter_Nr%3B-- HTTP/1.1" 200 -
```

2. PostgreSQL - Aufarbeitung

- Flask Log

```
2022-01-22 13:14:38,372 INFO werkzeug Thread-2 : 192.168.0.1 - - [22/Jan/2022 13:14:38] "GET /vorgaenge?search=%27%3B+SELECT++version%28%29%3B-- HTTP/1
.1" 200 -
2022-01-22 13:14:41,964 INFO werkzeug Thread-3 : 192.168.0.1 - - [22/Jan/2022 13:14:41] "POST /set_db HTTP/1.1" 302 -
2022-01-22 13:14:42,035 INFO werkzeug Thread-4 : 192.168.0.1 - - [22/Jan/2022 13:14:42] "GET /vorgaenge HTTP/1.1" 200 -
2022-01-22 13:14:47,392 INFO werkzeug Thread-5 : 192.168.0.1 - - [22/Jan/2022 13:14:47] "GET /vorgaenge?search=%27%3B+SELECT++version%28%29%3B-- HTTP/1
.1" 200 -
2022-01-22 13:14:56,219 INFO werkzeug Thread-6 : 192.168.0.1 - - [22/Jan/2022 13:14:56] "GET /vorgaenge?search=%27%3B+SELECT+P_Pers_Nr%2C+Vname%2C+Nnam
e%2C+Geburtsdatum+FROM+T_Personen%2C+T_Beschuldigte+WHERE+P_Pers_Nr+%3D+PF_Beschuldigter_Nr%3B-- HTTP/1.1" 200 -
```

- Fehler von John G. (Redundanz)

SELECT * FROM "t_beschuldigte" LIMIT 50 (0.001 s) Bearbei

Ändern	pf_beschuldigter_nr	geburtsdatum
bearbeiten	7	1902-08-24
bearbeiten	9	1902-07-04
bearbeiten	10	1949-12-01
bearbeiten	11	1910-10-01
bearbeiten	14	1909-03-24
bearbeiten	8	1902-08-24

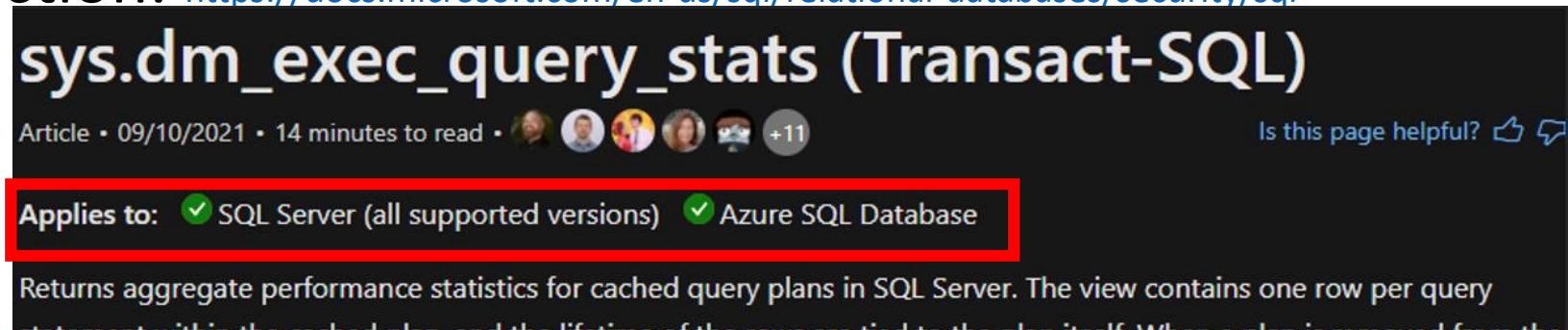
MS Azure, SQL-Server (Cloud)

Razor [C#]

3. MS Azure, SQL-Server (Cloud)

Vorbereitung – Microsoft-Dokumentation

- Razor Tutorial: https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-6.0&tabs=visual-studio
- MS Azure SQL-Überwachung: https://docs.microsoft.com/de-de/azure/azure-sql/database/auditing-overview
- Abfragepläne: https://docs.microsoft.com/en-us/sql/relational-databases/performance/execution-plans?view=sql-server-ver15, https://docs.microsoft.com/en-us/sql/relational-databases/query-processing-architecture-guide?view=sql-server-ver15#execution-plan-caching-and-reuse
- Transaktions-Log: https://docs.microsoft.com/en-us/sql/relational-databases/logs/the-transaction-log-sql-server?view=sql-server-ver15, https://docs.microsoft.com/de-de/sql/relational-databases/sql-server-transaction-log-architecture-and-management-guide?view=sql-server-ver15
- Best Practices SQL-Injection: https://docs.microsoft.com/en-us/sql/relational-databases/security/sql-injection?view=sql-server-ver15

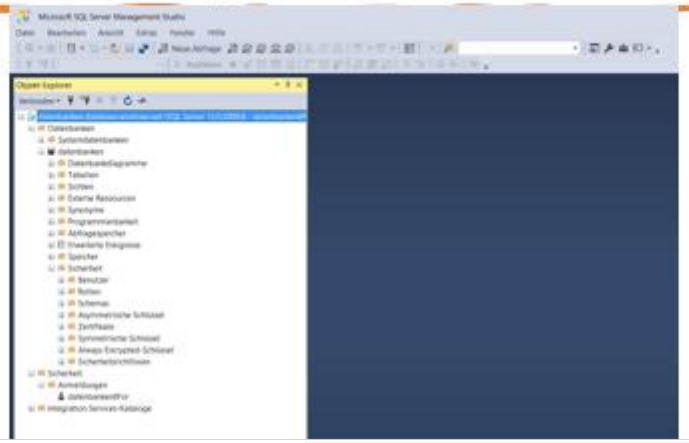


3. MS Azure, SQL-Server (Cloud)

Vorbereitung – Tools

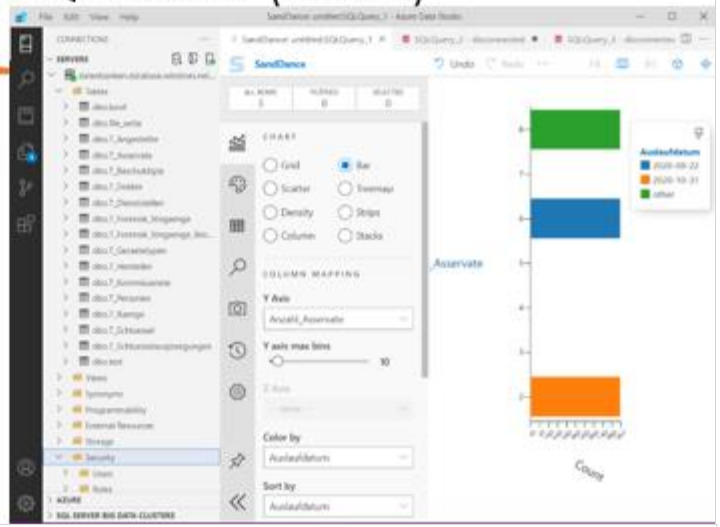
4. MS Azure, SQL-Server (Cloud)

Tools – SQL-Server Management-Studio



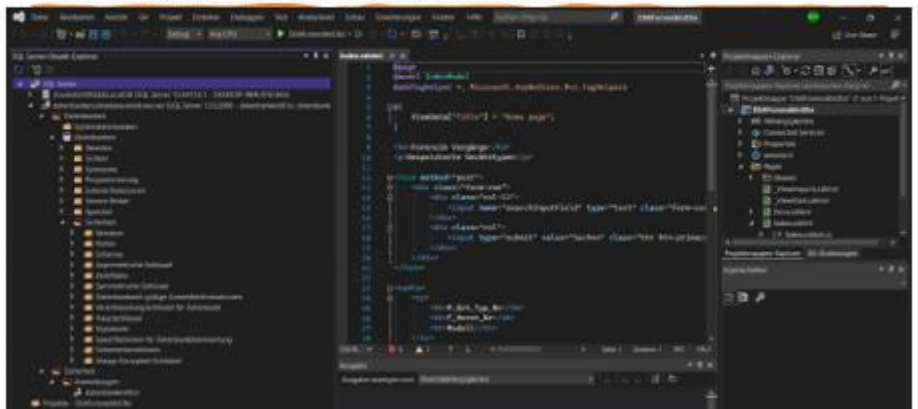
4. MS Azure, SQL-Server (Cloud)

Tools – MS Azure Data Studio



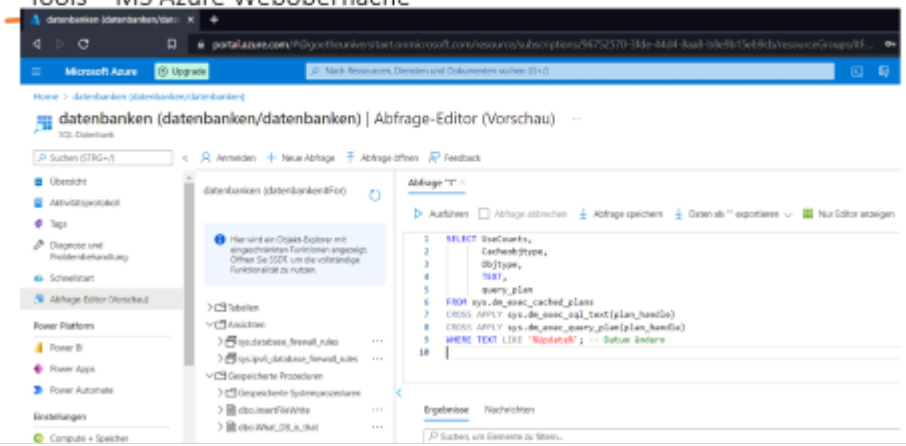
4. MS Azure, SQL-Server (Cloud)

Tools – Visual Studio SQL-Server Object-Explorer



3. MS Azure, SQL-Server (Cloud)

Tools – MS Azure Weboberfläche



3. MS Azure, SQL-Server (Cloud)

Datenbank anlegen und initialisieren

Microsoft Azure | Nach Ressourcen, Diensten und Dokumenten suchen (G+/)

Home >

Microsoft.SQLDatabase.newDatabaseNewServer_81eb11bd37c44181b0cae | Übersicht

Bereitstellung

Suchen (STRG+/)

- Übersicht
- Eingaben
- Ausgaben
- Vorlage

Löschen | Abbrechen | Erneut bereitstellen | Aktualisieren

Wir freuen uns über Ihr Feedback! →

Ihre Bereitstellung wurde abgeschlossen.

Bereitstellungsname: Microsoft.SQLDatabase.newDatabaseNewServ...
Abonnement: Azure subscription 1
Ressourcengruppe: itForensik

Startzeit: 24.1.2022, 22:06:11
Korrelations-ID: 28f1573e-b972-4306-a152-c582f3b8fb90

Bereitstellungsdetails (Herunterladen)

Nächste Schritte

Zu Ressource wechseln

Microsoft Azure | Nach Ressourcen, Diensten und Dokumenten suchen (G+/)

Home > datenbanken (datenbanken/datenbanken)

datenbanken (datenbanken/datenbanken) | Abfrage-E

SQL-Datenbank

Suchen (STRG+/)

- Übersicht
- Aktivitätsprotokoll
- Tags
- Diagnose und Problembehandlung
- Schnellstart
- Abfrage-Editor (Vorschau)

Power Platform

- Power BI
- Power Apps
- Power Automate

Einstellungen

- Compute + Speicher
- Verbindungszeichenfolgen
- Eigenschaften
- Sperren

Datenverwaltung

- Replikate

Anmelden | Neue Abfrage | Abfrage öffnen

datenbanken (datenbankenIIFor)

Hier wird ein Objekt-Explorer mit eingeschränkten Funktionen angezeigt. Öffnen Sie SSDT, um die vollständige Funktionalität zu nutzen.

- Tabellen
  - dbo.T_Angestellte
    - PF_Angest_Nr (PK, int, not null)
    - F_Rang_Nr (int, null)
  - dbo.T_Asservate
  - dbo.T_Beschuldigte
  - dbo.T_Delikte
  - dbo.T_Dienststellen
  - dbo.T_Forensik_Vorgaenge
  - dbo.T_Forensik_Vorgaenge_Besc
  - dbo.T_Geraetetypen
  - dbo.T_Hersteller
    - P_Herst_Nr (PK, int, not null)
    - Hersteller (nvarchar, not null)
  - dbo.T_Kommissariate

Abfrage

```
1
2
3
4
5
6
7
8  INSERT INTO T_Schluessel( Schl_Bez)
9  VALUES ('Dienststelle');
10
```

Ergebnisse | Nachrichten

Suchen, um Elemente zu filtern...

P_Pers_Nr	VName	NName
1	Susi	Schulz
2	Frank	Sommer
3	Friedrich	Dachs
4	Jane	Marple
5	Hercule	Poirot

Abfrage erfolgreich | 0s

3. MS Azure, SQL-Server (Cloud)

Razor [C#] – Webapplikation entwickeln

1

```
// Diese Methode wird bei jedem Post-Aufruf der Applikation aufgerufen.
// In diesem Fall immer, wenn der "Suchen"-Button geklickt wird.
// Schichtentrennung (Service-, Datenhaltungsschicht) und Aufrufreihenfolge gem. Demeter-Prinzip sollte eingeführt werden.
0 Verweise
public void OnPost(String searchInputField)
{
    // Error-Handling sollte eingefügt werden, auch um fehlerbasierte
    // SQL-Injection zu erschweren.

    Geratetypen = FindAndReturnGeraetetypen(searchInputField);
}
```

2

```
private DataTable FindAndReturnGeraetetypen(String whereArgument = "%")
{
    String sqlQuery = CreateAndLogSqlQuery(ref whereArgument);

    // --> private Boolean validateSqlQuery()

    return ExecuteSqlQuery(sqlQuery);
}
```

3

```
// Wenn der Nutzer eine leere Suche startet, werden alle Einträge angezeigt (hierfür erfolgt eine Prüfung
// Erstellung und gibt das anhand der Nutzereingabe erstellte SQL-Statement zurück.
1 Verweis
private string CreateAndLogSqlQuery(ref string whereArgument)
{
    string sqlQuery;
    if (whereArgument == "" || whereArgument == null)
    {
        whereArgument = "%";
    }

    sqlQuery = "Select * FROM [dbo].[T_Geraetetypen] WHERE Modell LIKE \'" + whereArgument + "\'";

    _logger.LogInformation("SQL Transaktion: {0}", sqlQuery);
    return sqlQuery;
}
```

4

```
1 Verweis
private static DataTable ExecuteSqlQuery(string sqlQuery)
{
    SqlCommand sqlCommand;
    SqlDataAdapter adapter;
    DataTable table = new DataTable();
    SqlConnection connection = new SqlConnection(connectionString);

    // Error-Handling sollte zur weiteren Härtung der Anwendung eingefügt werden.

    connection.Open();

    sqlCommand = new SqlCommand(sqlQuery, connection);

    using (adapter = new SqlDataAdapter(sqlCommand))
    {
        adapter.Fill(table);
    }

    // connection.Close(); // Für Demo deaktiviert, damit die beiden ausgeführten
                           // Statements in einer Verbindung ausgeführt werden.

    return table;
}
```

3. MS Azure, SQL-Server (Cloud)

Razor [C#] – SQL-Injection Angriffe durchführen

localhost:5164

DbIlForensikInDbs Home Privacy

Forensik Vorgänge

Gespeicherte Gerätetypen

1 ' AND 1=0 Union SELECT 0, 1, @@version; --

Suchen

P_Grt_Typ_Nr	F_Herst_Nr	Modell
0	1	Microsoft SQL Azure (RTM) - 12.0.2000.8 Sep 18 2021 19:01:34 Copyright (C) 2019 Microsoft Corporation

2 ' AND 1 = 0; UPDATE T_Personen SET VName = 'unbekannt', NName = 'unbekannt' WHERE VName = 'John' AND NName = 'Gotti'; --

Suchen

P_Grt_Typ_Nr	F_Herst_Nr	Modell

Fazit

4. Fazit

Funktionalität	MySQL (Docker)	PostgreSQL (Docker)	SQL-Server (Cloud)
Logging ohne Konfigurationsaufwand aktiviert?	✗	✗	✓
Zugriff auf Ausführungspläne/standard-mäßig aktiviert?	✗ / ✗	✗ / ✗	✓ / ✓
Zugriff auf Transaktions-Logs/standardmäßig aktiviert?	✓ / ✗	✓ / ✗	✓ / ✓
Ausführen von Kommandozeilen-Befehlen standardmäßig aktiviert?	✓	✓	✗
Syntax Case-Sensitive?	✓	✗	✗
Finales Ranking	2	2	1

A wise man once said…

The end is merely the start.