

# MS-SQL Trigger

Trigger werden ausgeführt, wenn in einer Tabelle ein Ereignis ausgelöst wird.

Konkret heißt das, dass bei einem INSERT, UPDATE oder DELETE wir eingreifen können und einen oder mehrere Befehle „danach“ oder „anstelle“ von ausführen können.

[crayon-6769bbbf2680f433832797/]

Wählen wir AFTER, dann wird der Datensatz erstellt/geupdated oder gelöscht und **erst dann** greift der Trigger ein.

Wählen wir stattdessen Instead Of, wird **anstelle von** dem Insert, Update oder Delete der Trigger ausgeführt.

Beim Trigger stehen uns 2 weitere Temporäre Tabellen zur Verfügung, nämlich inserted und deleted.

wenn wir z.B. einen neuen Datensatz id, Vorname, Nachname in die Datenbank geschrieben haben,

so können wir aus dem Trigger mit

[crayon-6769bbbf26815678432765/]

die Id, die gerade eben eingetragen wurde abfragen.

Machen wir ein Update, so wird der alte Datensatz erst einmal erst in die temporäre Tabelle deleted gesteckt, welchen

wir ebenso mit

[crayon-6769bbbf26816693253547/]

abfragen könnten.

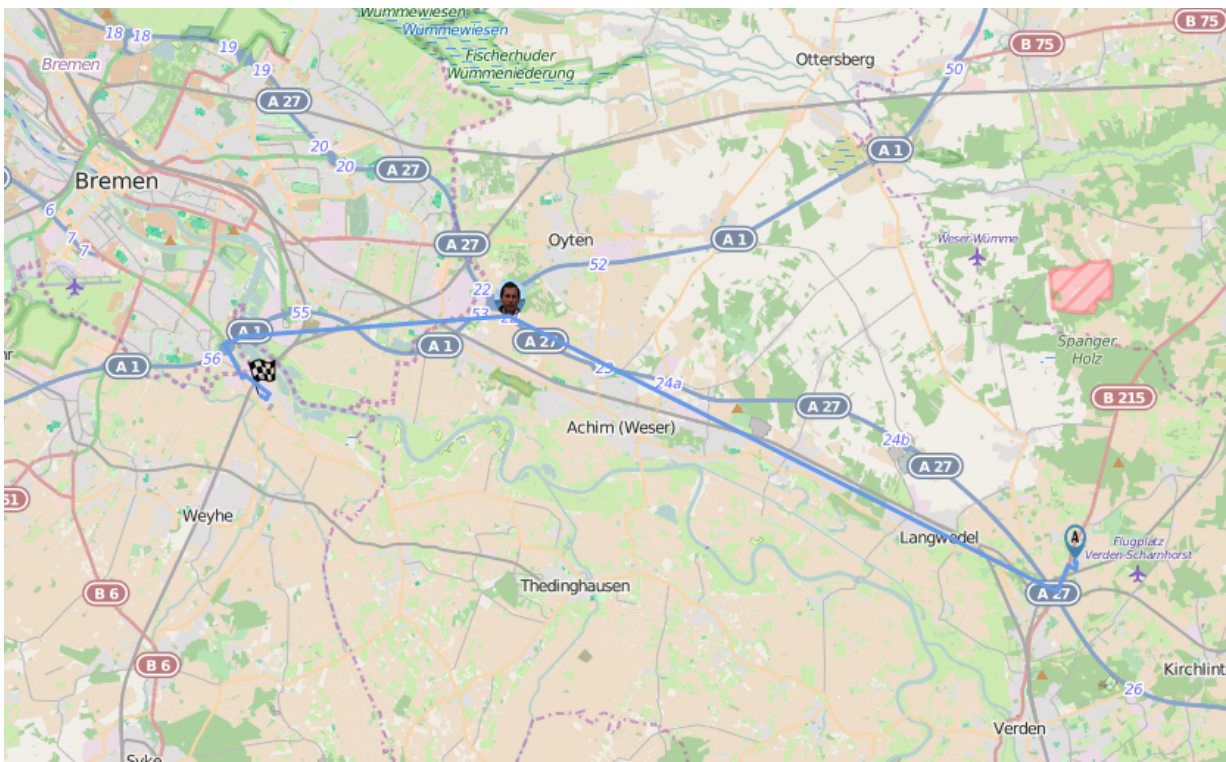
---

# OpenStreetMap, OpenLayers Map anzeigen, Marker setzen und Route hinzufügen

einfacher Open Layer Code, mit dem man eine Linie zwischen 2 Punkten erstellen kann.

Diese bekommen am Anfangspunkt einen Marker, am Ende einen Marker und dazwischen einen.

Dies könnte z.B. so Aussehen:



[crayon-6769bbbf26c43934987372/]  
Download Beispielprojekt: OpenLayers

---

# SQL Server – Spalteninformationen einer Tabelle selektieren

Manchmal möchte man Infos über die Spalten einer bestimmte Tabelle

oder einer ganze Datenbank abrufen. Ich habe 2 Queries geschrieben mit den man diverse Informationen bekommt.

**Schemainformationen für 1 Tabelle:**

[crayon-6769bbbf26e3a257307441/]

**Schemainformationen für ganze Datenbank:**

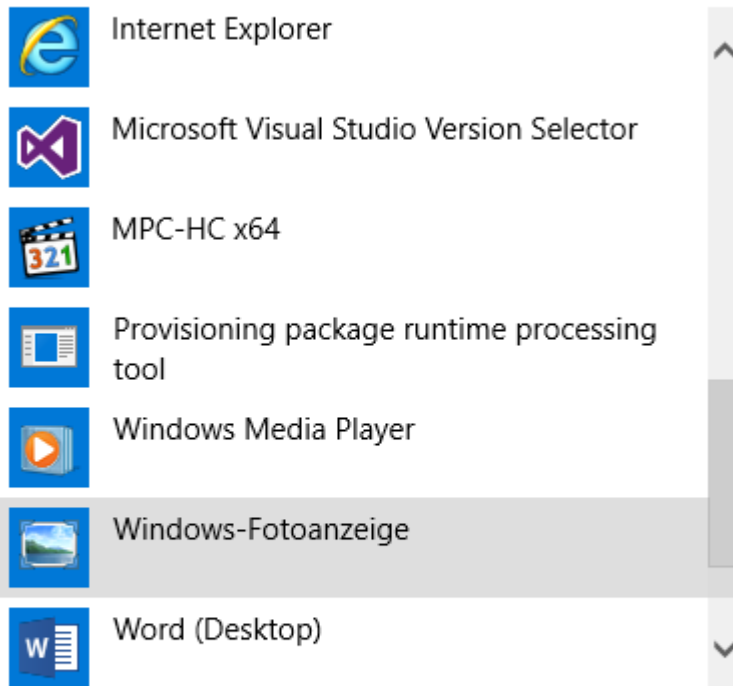
[crayon-6769bbbf26e5a621103648/]

---

# Windows 10 mit alter Fotoanzeige nutzen

Einfach folgenden Registry Eintrag ausführen und man kann ein Bild mit rechter Maustaste und Öffnen mit wieder mit der gewohnten Fotogalerie nutzen.

Wie soll diese Datei geöffnet werden?



Immer diese App zum Öffnen von .JPG-Dateien verwenden

OK

---

– Download Fotogalerie Registry –  
[crayon-6769bbbf27002819168113/]

---

## Windows, mit Microsoft Konto automatisch anmelden

Ich habe ja kein Problem auch meinen Microsoft Account dazu zu nutzen um mich an meinen lokalen Rechner anzumelden.

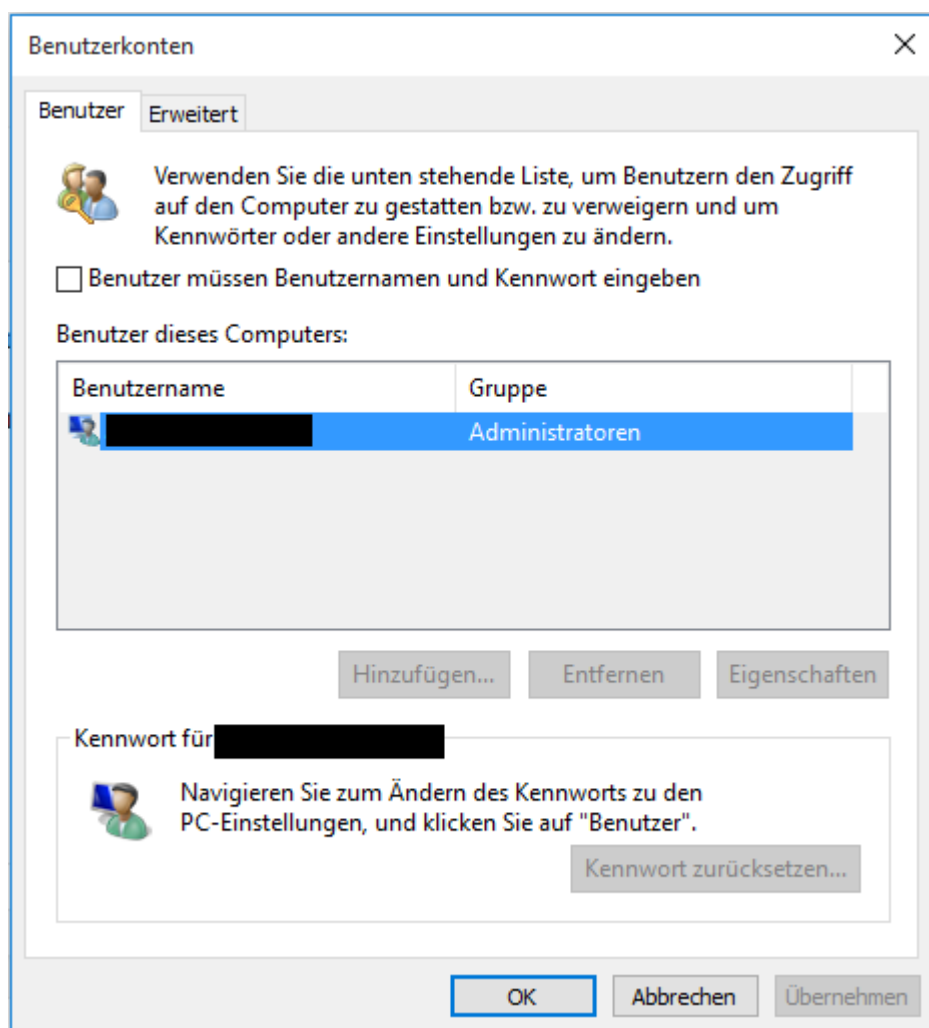
Allerdings nervt mich, dass ich nach jedem neustart das Passwort des Microsoft Accounts eingeben muss.

Dies geht auch einfacher:

1. Das lokale Konto evtl. unter Kontoeinstellungen (Systemsteuerung) in Microsoft-Konto wechseln und dann

In die Suche oder unter Ausführen netplwiz eingeben und den Haken bei „Benutzer müssen Benutzernamen und Kennwort eingeben“ wegnehmen.

dann das letzte Mal das Passwort eingeben und Fertig



# SQL Server Querys

Duplicate löschen, welche sich aus col1 und col2 zusammen setzen. Alternativ kann man natürlich auch nur mit col1 arbeiten

```
[crayon-6769bbbf2718c494347487/]
```

Update Spalte, wenn Datum Heute > dann 1 sonst 0:

```
[crayon-6769bbbf27191947371440/]
```

Zeilennummerierung erzeugen:

```
[crayon-6769bbbf27193857228998/]
```

---

## LINQ Querys

SQL:

```
IF EXISTS (SELECT 1 FROM tblUsers WHERE Username = „whatever“)  
// Return 1, wenn Username vorhanden ist, 0, wenn eben nicht
```

LINQ:

```
[crayon-6769bbbf2739c881551221/]
```

---

## My SQL Querys

eine Sammlung netter Querys für MySQL:

`ifnull(expression,“)` – Wenn `expression` `NULL` ist, dann zählt der rechte Wert.

`Concat(,a’,123,(SELECT col from tab WHERE a = b))` – erstellt einen zusammengesetzten String aus allen teilen der Funktion

---

## Eigenen Event erstellen

Ein Event oder deutsch Ereignis ist immer an einen Deleगतentypen gebunden, welcher wiederum eine bestimmte Signatur von (Rückgabewert + Parameter).

Ungefähr, stellt es dieses Muster da:

Zuerst brauchen wir einen delegaten. Jetzt stellt sich die Frage, die Methoden, auf die der Delegat später verweist, wie soll die aussehen? soll sie einen Rückgabewert haben? Soll irgend ein Wert an die Methode übergeben werden? Ja? Dann brauchen wir Parameter. Um das noch ein wenig verständlicher zu machen, wenn wir eine for-Schleife haben und wollen, dass das Event nach jedem Schleifendurchgang immer abgefeuert wird, und uns den Wert liefert. Dann würden wir einen Delegaten mit einem Parameter vom Typ `int` z.B. machen.

```
[crayon-6769bbbf2754f131915361/]
```

Dann braucht man das Event an sich. Dieser muss vom Typ des Delegaten sein.

```
[crayon-6769bbbf27554581035601/]
```

Jetzt muss gesagt werden, wann das Event abgefeuert werden muss. Und, wichtig, wir haben dem Delegaten einen Parameter (`int a`) angegeben. Dieser Parameter muss nun durch das Event

gefüllt werden. Das machen wir, indem wir

```
[crayon-6769bbbf27555869660339/]
```

Jetzt wird bei jedem Schleifendurchgang das Event abgefeuert.

Ok, das war unsere Quell klasse.

Jetzt erstellen wir ein Objekt dieser Klasse und weisen dem Event eine Methode zu, die ausgeführt werden soll, sobald das Event abgefeuert wird. Die Methode muss die selbe Signatur haben wie unser Delegat. Zum Schluss müssen wir die Methode in der Quell klasse ausführen und das Event wird abgefeuert und führt ebenso die Methode mit dem Parameter aus, die wir in der Ziel Methode eingegeben haben.

```
[crayon-6769bbbf27557698520336/]
```

```
[crayon-6769bbbf27559197270450/]
```

Download Beispielprojekt Eigener Event

---

## Delegaten in .NET

Das .Net Framework beinhaltet bereits Delegaten, welche bestimmte Signaturen haben.

### 1. Der EventHandler

stellt einen Delegaten dar, der keinen Rückgabewert hat (void), und 2 Parameter hat. 1. Ist der object sender , welcher das Senderobject implementiert und EventArgs[] welcher evtl. Argumente beinhalten kann. Diesen Delegaten findet man, wenn man zu einem Event von Steuerelementen eine Methode aufruft.



## 2. Action<>

Der Action Delegat besitzt keinen Rückgabewert, wohl aber bis zu 16 Parameterelemente. Dieser Delegat kann z.B. dazu verwendet werden um eine Methode mit der Task Klasse aufzurufen und diese im neuen Thread zu starten.

[crayon-6769bbbf2775f168998215/]

Das geht natürlich auch mit anonymen Methoden

[crayon-6769bbbf27778754642108/]

## 3. Func<>

Der Func Delegat ist dem Action<> Delegaten gleich, liefert im Gegensatz zu ihm aber einen Rückgabewert