

**Modul 105**

Olivier Winkler

Bahnhöheweg 70 / 3018 Bern

INF2017

Stephanie Ramseyer

Abgabe: 13.04.2018

# Kompetenzen

|  |  |  |  |
| --- | --- | --- | --- |
| Berufliche Praxis | Lernzielkontrolle |  |  |
| C2.1: Wählen das geeignete Datenbankmodell (relationales, hierarchisches etc.) und bestimmen ein Produkt (DBMS). | Selbständig: | Geübt: | Erklärt: |
| C2.2: Erstellen ein physisches Modell (z.B. DDL, Referential Integrity, Constraints) und denormalisieren es (Performance). | Selbständig: | Geübt: | Erklärt: |
| Lösen Betriebliche Vorgaben mit technischen Vorgaben (Web, Mobile, Desktop, fat Client, Automaten) | Selbständig: | Geübt: | Erklärt: |
| C2.3: Führen Load- und Performancetests durch, optimieren entsprechend und stellen Wartbarkeit sicher. | Selbständig: | Geübt: | Erklärt: |
| C2.4: Stellen Datensicherheit (Backup, Verfügbarkeit usw.) und Datenschutz (u.A. Zugriffsberechtigung) sicher. | Selbständig: | Geübt: | Erklärt: |
| 1: Bestimmen Zugriffsschnittstelle und –Technologie bestimmen (z.B. statisch/dynamisches SQL, ADO, HQL, OR-Mapper, Stored Procedures, etc.). | Selbständig: | Geübt: | Erklärt: |

Inhaltsverzeichnis

[Kompetenzen 2](#_Toc511392877)

[1. Tagesjournale Mittwoch – Freitag 04.04.18 – 06.03.18 6](#_Toc511392878)

[1.1 Tagesjournal Mittwoch 04.04.18 6](#_Toc511392879)

[1.2 Tagesjournal Donnerstag 05.04.18 7](#_Toc511392880)

[1.3 Tagesjournal Freitag 06.04.18 8](#_Toc511392881)

[2. Tagesjournale Montag – Freitag 09.04.18 – 13.03.18 9](#_Toc511392882)

[2.1 Tagesjournal Montag 09.04.18 9](#_Toc511392883)

[2.2 Tagesjournal Dienstag 10.04.18 10](#_Toc511392884)

[2.3 Tagesjournal Mittwoch 10.04.18 11](#_Toc511392885)

[2.4 Tagesjournal Donnerstag 10.04.18 12](#_Toc511392886)

[2.5 Tagesjournal Freitag 10.04.18 13](#_Toc511392887)

[3. Grundlagen von SQL 14](#_Toc511392888)

[3.1 Was ist SQL (Structured Query Language) 15](#_Toc511392889)

[3.2 Was ist MySQL? 15](#_Toc511392890)

[3.3 Was sind Daten? 15](#_Toc511392891)

[3.4 Was ist eine Datenbank / Datenbanksystem DBS 16](#_Toc511392892)

[3.4.1 Hierarchisches Modell 16](#_Toc511392893)

[3.4.2 Netzwerk-Modell 17](#_Toc511392894)

[3.4.3 Relationales Modell 17](#_Toc511392895)

[3.4.4 Objektorientiertes Modell 18](#_Toc511392896)

[3.5 Wie ist eine Tabelle aufgebaut? 19](#_Toc511392897)

[3.6 Beziehungen (Assoziationen) 19](#_Toc511392898)

[3.6.1 Notationen 20](#_Toc511392899)

[3.7 Redundanzen 21](#_Toc511392900)

[3.8 Datenkonsistenz 21](#_Toc511392901)

[3.9 Inkonsistenz 21](#_Toc511392902)

[3.10 ERM und ERD 22](#_Toc511392903)

[3.11 Normalformen / Normalisierung 23](#_Toc511392904)

[3.12 Anomalien 24](#_Toc511392905)

[4. Installation SQL Server 2017 25](#_Toc511392906)

[4.1 Installation 26](#_Toc511392907)

[5. Datenbank exportieren / importieren 32](#_Toc511392908)

[5.1 Datenbank exportieren 33](#_Toc511392909)

[5.2 Datenbank importieren 33](#_Toc511392910)

[6. Datenbank – Projekt 34](#_Toc511392911)

[6.1 Projekt – Museumsdatenbank 35](#_Toc511392912)

[6.2 CREATE – Befehle 36](#_Toc511392913)

[6.3 INSERT - Befehle 37](#_Toc511392914)

[6.4 ALTER Tabel - Befehl 38](#_Toc511392915)

[6.5 Abfragen mit Datenbank 38](#_Toc511392916)

[6.6 Benutzer hinzufügen und Berechtigungen verteilen 42](#_Toc511392917)

[6.6.1 Berechtigungen abfragen 43](#_Toc511392918)

[7. SQL Befehle 44](#_Toc511392919)

[7.1 SELECT – SQL 45](#_Toc511392920)

[7.2 Das \* Zeichen – SQL 45](#_Toc511392921)

[7.3 SELECT DISTINCT – SQL 45](#_Toc511392922)

[7.4 SELECT INTO – SQL 45](#_Toc511392923)

[7.5 INSERT INTO SELECT – SQL 45](#_Toc511392924)

[7.6 WHERE – SQL 45](#_Toc511392925)

[7.7 AND, OR, NOT – SQL 46](#_Toc511392926)

[7.8 ORDER BY – SQL 46](#_Toc511392927)

[7.9 INSERT INTO – SQL 46](#_Toc511392928)

[7.10 NULL Values – SQL 47](#_Toc511392929)

[7.12 UPDATE – SQL 47](#_Toc511392930)

[7.13 DELETE – SQL 47](#_Toc511392931)

[7.14 SELECT TOP (Limit) – SQL 47](#_Toc511392932)

[7.15 MIN() und MAX() – SQL 48](#_Toc511392933)

[7.16 COUNT(), AVG() und SUM() – SQL 48](#_Toc511392934)

[7.17 LIKE – SQL 48](#_Toc511392935)

[7.18 IN – SQL 49](#_Toc511392936)

[7.19 BETWEEN – SQL 49](#_Toc511392937)

[7.20 ALIASES – SQL 49](#_Toc511392938)

[7.21 JOINS – SQL 49](#_Toc511392939)

[7.22 INNER JOIN – SQL 50](#_Toc511392940)

[7.23 LEFT JOIN– SQL 50](#_Toc511392941)

[7.24 RIGHT JOIN– SQL 51](#_Toc511392942)

[7.25 FULL JOIN – SQL 51](#_Toc511392943)

[7.26 SELF JOIN – SQL 52](#_Toc511392944)

[7.27 UNION – SQL 52](#_Toc511392945)

[7.28 GROUP BY – SQL 52](#_Toc511392946)

[7.29 HAVING – SQL 53](#_Toc511392947)

[7.30 EXISTS – SQL 53](#_Toc511392948)

[7.31 ANY und ALL – SQL 53](#_Toc511392949)

[7.32 Datentypen von SQL 54](#_Toc511392950)

[8. Reflexion 55](#_Toc511392951)

[9. Quellenverzeichnis 56](#_Toc511392952)

[10. Abbildungsverzeichnis 58](#_Toc511392953)

[11. Selbstständigkeitserklärung 60](#_Toc511392954)

# 1. Tagesjournale Mittwoch – Freitag 04.04.18 – 06.03.18

## 1.1 Tagesjournal Mittwoch 04.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Programmiert***

Am Morgen habe ich mein TicTacToe

fertig programmiert. Ich 3h

***Installation von MySQL-Server etc.***

Zusammen mit Stephanie haben wir die

benötigten Programme für Modul 105

installiert. Klasse 2h

***Aufträge im Modul 105***

Arbeitsblätter von MySQL und Lernvideos Ich 3h

Total: ~ 8h12min

***Tagesablauf***

Am Morgen habe ich noch das TicTacToe, welches wir gestern angefangen haben, fertiggemacht. Gestern waren wir alleine und mussten Aufträge von Stephanie lösen. In den Aufträgen ging es hauptsächlich ums programmieren von kleineren Programmen und zum Schluss konnte man noch das TicTacToe anfangen. Als dann Stephanie gekommen ist, haben wir zusammen mit der Klasse die benötigten Programme für das neue Modul 105 installiert. Dies hat einige Zeit in Anspruch genommen, da die Installation teilweise viel Zeit für sich benötigt hat. Während der Installation und für den restlichen Tag mussten wir auch wieder einige Arbeitsblätter lösen. Falls man danach nichts zu tun hatte, konnte man die Lernvideos aus dem Moodle anschauen.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Neues Modul

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Morgen werden wir noch unbekannte SQL-Befehle anschauen.

## 1.2 Tagesjournal Donnerstag 05.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Moduljournal***

Moduljournal geschrieben Ich 6.5h

***Einzelauftrag ERD*** 1.5h

Ein ERD zu einer Aufgabe erstellen. Ich

Total: ~ 8h12min

***Tagesablauf***

Heute war ein Tag, den ich für mein Journal gut nutzen konnte. Den ganzen Morgen konnte ich an meinem Journal schreiben. Heute habe ich viele SQL-Befehle aufgeschrieben in meinem Journal. Auch habe ich die Installation von SQL Server beschrieben. Am Nachmittag hat uns Stephanie dann einen Auftrag gegeben, den wir bis Morgen fertig haben müssen. Der Auftrag war ein ERD zu erstellen mit den verschiedenen Beziehungen. Am Ende von diesem Arbeitstag habe ich noch das Tagesjournal geschrieben.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Viel geschrieben im Journal.

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Aufträge bearbeiten.

## 1.3 Tagesjournal Freitag 06.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Moduljournal***

Moduljournal geschrieben Ich 4h

***Normen***

Normen von Tabellen angeschaut Klasse 2h

***Aufgaben SQL***

Aufgaben von SQL gelöst Ich 2h

Total: ~8h12min

***Tagesablauf***

Den ganzen Morgen habe ich an meinem Journal geschrieben. Auch heute bin ich weit gekommen mit meinem Journal. Ab dem Mittag habe ich die neuen SQL-Aufgaben gelöst. Kurz darauf kam dann Stephanie und hat uns die verschiedenen Normalformen von Tabellen-Modell gezeigt und erklärt. Zusammen haben wir auch ein Beispiel gemacht und danach mussten wir unser ERD, welches wir gestern erstellt haben, in diese Formen umwandeln. Nachdem ich dies fertiggemacht habe, habe ich weiter an den Aufträgen gearbeitet und zum Schluss habe ich noch dieses Tagesjournal geschrieben.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Vieles gelernt

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Nächste Woche werde ich die Aufträge fertigmachen und neue bearbeiten.

# 2. Tagesjournale Montag – Freitag 09.04.18 – 13.03.18

## 2.1 Tagesjournal Montag 09.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Moduljournal***

Moduljournal geschrieben. Ich 4h

***Aufträge im Modul 105***

Arbeitsblätter von MySQL und arbeiten am

Datenbank-Projekt Ich 4h

Total: ~ 8h12min

***Tagesablauf***

Am Morgen habe ich an meinem Journal gearbeitet. Auch heute konnte wieder über -viele Themen schreiben. Heute habe ich richtig mit unserem Datenbank-Projekt angefangen. Bis und mit Mittwoch muss dieses Projekt fertig sein. Im Projekt geht es darum, mit unseren Tabellen ein ERM zu erstellen und anschliessend die Daten in eine Datenbank einfügen und damit einige Aufgaben zu lösen. Heute habe ich bereits das ERM gemacht und die Tabellen in SQL erstellt.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Datenbank-Projekt

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Morgen werde ich an meinem Projekt weiterarbeiten.

## 2.2 Tagesjournal Dienstag 10.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Moduljournal***

Moduljournal geschrieben. Ich 30min

***Aufträge im Modul 105***

Arbeitsblätter von MySQL und arbeiten am

Datenbank-Projekt Ich 7.5h

Total: ~ 8h12min

***Tagesablauf***

Am Morgen habe ich mit an meinem Projekt weitergearbeitet. Den ganzen Tag habe ich in mein Projekt investiert. Am Morgen hatte ich noch einige Probleme mit dem Importieren der CSV-Dateien. Als ich es dann endlich geschafft habe, konnte ich die restlichen Aufgaben lösen. Heute habe ich alle Aufgaben bis auf die Letzte lösen können. Am Nachmittag haben wir auch noch mit Stephanie abgemacht, dass wir die erste Prüfung am Donnerstagnachmittag machen werden. Die Prüfung können wir schon vor den Ferien machen, da wir schon viel Vorwissen haben. Zum Schluss von diesem Arbeitstag habe ich noch dieses Tagesjournal geschrieben.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Datenbank-Projekt

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Morgen werde ich mein Projekt beenden.

## 2.3 Tagesjournal Mittwoch 11.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Moduljournal***

Moduljournal geschrieben. Ich 6.5h

***Aufträge im Modul 105***

Arbeitsblätter von MySQL und arbeiten am

Datenbank-Projekt Ich 1.5h

Total: ~ 8h12min

***Tagesablauf***

Heute musste ich mit dem Datenbank-Projekt fertig sein. Darum habe ich am Morgen als erstes noch die letzte Aufgabe von dem Projekt gelöst. Morgen werden wir die Prüfung schreiben und dürfen dazu das Journal benutzten. Darum habe ich den restlichen Tag genutzt um mein Journal soweit es geht fertig zu machen. Auch habe ich mein Projekt teilweise in mein Journal integriert. Am Ende von diesem Tag habe ich noch das Tagesjournal geschrieben.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Rechtzeitig fertig mit Projekt

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Morgen werden wir den ersten Teil der ÜK-Prüfung schreiben.

## 2.4 Tagesjournal Donnerstag 12.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Moduljournal***

Moduljournal geschrieben. Ich 6h

***ÜK-Prüfung Teil 1***

Prüfung geschrieben Ich 2h

Total: ~ 8h12min

***Tagesablauf***

Heute konnte ich mein Journal eigentlich fertig stellen. Mir fehlt noch ein bisschen von der Reflexion und das Tagesjournal von morgen. Den ganzen Morgen konnte ich an meinem Journal schreiben. Heute habe ich vor allem Verzeichnisse und noch kleine Ergänzungen gemacht. Nach dem Mittag etwa um 13:00 Uhr haben wir mit dem ersten Teil der ÜK-Prüfung begonnen. Nach der Prüfung habe ich dann noch das Tagesjournal und ein kleiner Teil der Reflexion geschrieben.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Moduljournal fast fertig

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Morgen werden wir den zweiten Teil der ÜK-Prüfung schreiben und ich werde mein Journal abgeben.

## 2.5 Tagesjournal Freitag 13.04.18

Tätigkeiten Beteiligte Personen Aufwand h

***Moduljournal***

Moduljournal geschrieben. Ich 3h

***Zusätzliche Aufgabe***

Zusätzliche Aufgaben gelöst. Ich 3h

***ÜK-Prüfung Teil 2***

Prüfung geschrieben Ich 2h

Total: ~ 8h12min

***Tagesablauf***

Heute Morgen habe ich an meinem Journal geschrieben und dies schliesslich am Ende des Tages rechtzeitig abgegeben. Heute haben wir auch den zweiten Teil der Prüfung geschrieben. Mit der Prüfung haben wir zwischen 12:45 Uhr und 13:00 Uhr begonnen. Auch diese Prüfung dauerte 90 Minuten wie der erste Teil. Vor dem Mittag und vor der Prüfung habe ich noch die Zusatzaufgaben vom Moodle gelöst. Bevor ich nach Hause gehen konnte, musste ich noch das Tagesjournal schreiben und mein Journal abgeben.

***Hilfestellungen***

Am heutigen Tag habe ich keine Hilfe benötigt.

***Reflexion***

**Positives:**

Ein weiteres ÜK-Modul beendet.

**Kritisches:**

Heute habe ich nichts Kritisches gehabt.

**Erkenntnisse:**

Heute habe ich keine Erkenntnisse gemacht.

**Nächste Schritte:**

Ferien geniessen.





















































# 3. Grundlagen von SQL



## 3.1 Was ist SQL (Structured Query Language)

SQL ist eine Datenbanksprache. Diese Sprache dient zur Definition von Datenstrukturen in relationalen Datenbanken. Auch dazu kommt das Bearbeiten und Abfragen von Datensätzen in einer Datenbank. Fast alle Datenbanksysteme unterstützen SQL. Die Befehle, die in SQL angewendet werden, sind in vier Kategorien unterteilt.

* DML (*Data Manipulation Language*)
* Befehle zur Datenmanipulation wie Ändern, Einfügen, Löschen und lesbaren Zugriff
* DDL (*Data Definition Language*)
* Befehle zur Definition des Datenbankschemas wie Elemente zu ändern oder zu entfernen)
* DCL (*Data Control Language*)
* Befehle für die Rechteverwaltung und Transaktionskontrolle wie Lese- und Schreibrechte
* DQL (*Data Query Language*)
* Befehle für Abfragen von Daten

## 3.2 Was ist MySQL?

MySQL ist eines von vielen Datenbankverwaltungssysteme. MySQL gehört dabei zu einem der weltweit verbreitesten Programmen dieser Art. Es kann grosse Datenmengen schnell verarbeiten. In der Datenbank werden die Daten so gespeichert, damit sie möglichst wenig Speicherplatz benötigen. MySQL ist für verschiedene Betriebssysteme verfügbar. MySQL wird vor allem für die Datenspeicherung für Webservices, als Verbindung mit dem Webserver Apache und PHP genutzt. Auch grosse Unternehmer wie Flickr, YouTube, Facebook und Twitter nutzen MySQL.

## 3.3 Was sind Daten?

Daten sind streng genommen einfache Informationen, die von einem Computer verarbeitet werden. Diese Verarbeitung kann dabei auf unterschiedliche Wegen passieren (Kopieren, Bearbeiten und Speichern). Diese Informationen können mit der Zeit zu komplexen Daten werden. Unter komplexen Daten versteht man eine grosse Datenmenge wie es bei Musik oder Videos vorkommt. Mehrere Daten, die logisch zusammengehören, werden als Datensätze bezeichnet. Diese Informationen können auf einer Festplatte gespeichert werden. Meistens werden sie lokal auf der Computerfestplatte gespeichert. Sie können auch auf einem Server oder in einer Datenbank gespeichert werden.

## 3.4 Was ist eine Datenbank / Datenbanksystem DBS

Eine Datenbank ist ein elektronisches Verwaltungssystem von Daten. Eine Datenbank arbeitet meistens mit grossen Datenmengen. Die Aufgabe einer Datenbank ist diese grosse Menge von Daten effizient, widerspruchfrei, dauerhaft zu speichern und bedarfsgerechte Darstellungsformen für Benutzer und Programme zur Verfügung stellen. Eine Datenbank besteht aus der Verwaltungssoftware (DBMS) und der Menge der zu verwaltenden Daten. Die Verwaltungssoftware verwaltet intern die strukturierte Speicherung der Daten und kontrolliert die Zugriffe auf die Datenbank. Damit diese Daten verwaltet werden können, bietet jedes Datenbanksystem eine Sprache wie SQL an. Diese Datenbanksysteme gibt es in verschiedenen Formen oder auch Modelle genannt. Diese Formen beschreiben die Art und Weise wie ein solches System Daten speichert und verwaltet.

* Hierarchisches Modell
* Netzwerk-Modell
* Relationales Modell
* Objektorientiertes Modell

### 3.4.1 Hierarchisches Modell

Das hierarchische Modell ist das älteste Datenbankmodell und wird als hierarchische Baumstruktur dargestellt. Durch diese Darstellung ist dieses Modell sehr schnell wenn es ums Thema der lesenden Zugriffe geht. Jeder Datensatz hat genau einen Vorgänger ausser der Wurzel der Baumstruktur. Der Nachteil bei dieser Darstellung liegt bei der Speicherung der Daten, da die Verweise untereinander zuerst ermittelt werden müssen. Viele Dateisysteme von Betriebssystemen benutzen dieses Modell für die Datenabbildung. Heutzutage ist dieses Datenbankmodell weitestgehend von anderen Modellen abgelöst worden.

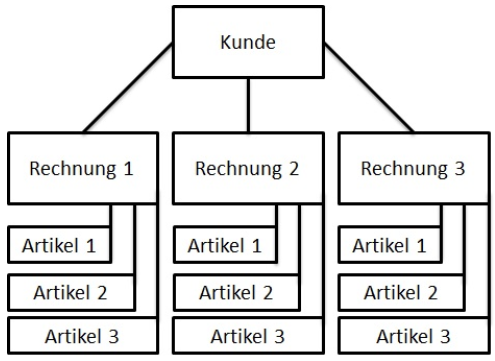


Abbildung 1 Hierarchisches Modell

### 3.4.2 Netzwerk-Modell

Das Netzwerk-Modell wurde etwa gleichzeitig mit dem relationalen Modell veröffentlicht worden. Bei dem Netzwerk-Modell wurden bei der Veröffentlichung drei Datenbanksprachen mitgeliefert worden. Diese drei Sprachen (DML, DDL, DCL) werden auch im relationalen Modell angewendet. Das Netzwerk-Modell ähnelt sehr stark mit dem hierarchischen Modell. In diesem Modell gibt es keine strenge Hierarchie. Die Datenfelder bestehen aus einem Namen und einem Wert.

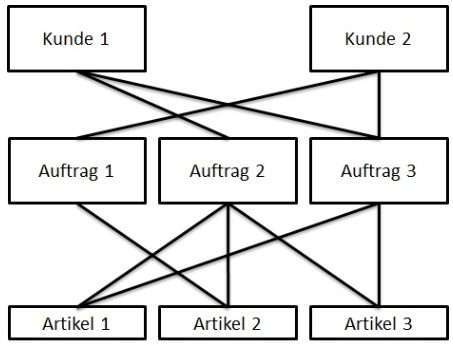


Abbildung 2 Netzwerk-Modell

### 3.4.3 Relationales Modell

Dieses Modell ist das weitverbreitetste Datenmodell, welches als Standard genutzt wird. Das Datenbankmodell ist eine Ansammlung von Tabellen, die miteinander verknüpft sind. Das Fundament dieses Modells besteht aus vier Elementen: Tabellen, Attributen, Beziehungen und die Grundlagen der relationalen Algebra. Das Modell stellt eine mathematische Beschreibung einer Tabelle und ihre Beziehung zu anderen möglichen Tabellen dar.

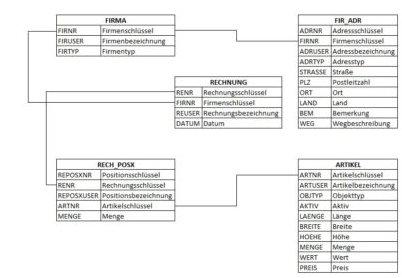


Abbildung 3 Relationales Modell

### 3.4.4 Objektorientiertes Modell

Dieses Modell ist dazu da Daten zusammen mit ihren Funktionen in ein Objekt zu speichern. Das Modell hat einzelne Gleichheiten mit den objektorientierten Programmiersprachen. Die Abfrage kann über die Funktionen der Daten oder über die Objektabfragesprache abgefragt werden. Dabei kommt eine ähnliche Sprache wie bei SQL zum Einsatz, nämlich „Object Query Language (OQL)“. Der Einsatz von diesem Modell erschwert aber die Integration in eine relationale Datenbank. Zudem können objektorientierte Datenbanken multimediale Inhalte speichern und komplexe Datentypen annehmen.

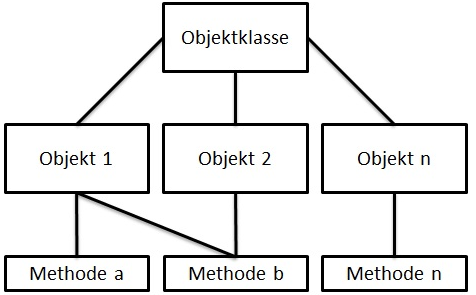


Abbildung 4 Objektorientiertes Modell

## 3.5 Wie ist eine Tabelle aufgebaut?

Attribute

(Spaltenname)

Tabelle / Entitätsmenge

Primärschlüssel, Identifikationsschlüssel

|  |  |  |
| --- | --- | --- |
| ID | Namen | Vornamen |
| 1 | Muster | Hans |
| 2 | Müller | Hans |

Daten, Values, Zelleninhalt

Zeile, Entität, Tupel, Datensatz

Eine Tabelle ist mit den oben genannten Objekten aufgebaut. Der Tabellennamen wird auch Entität genannt. Eine Tabelle hat einen Identifikationsschlüssel. Bei dieser Tabelle ist dieser ***„ID“***. Mit diesem Schlüssel kann jeder Datensatz deutlich identifiziert werden. Dieser Schlüssel kann auch Primarykey genannt werden, falls zwei Tabellen oder mehrere verbunden werden. Dazu wird dann auch der sogenannte Foreignkey benutzt, der auf eine weitere Tabelle verweist. Ein Attribut beschreibt den Spaltennamen z.B. „Namen“. Wenn man von einem Attribut spricht, meint man die Spalte einer Tabelle. Bei einer Tabelle spricht man von einer Zeile. Diese können auch Entität, Tupel oder Datensatz genannt werden. Die Daten in der Tabelle werden Daten, Values oder Zelleninhalt genannt.

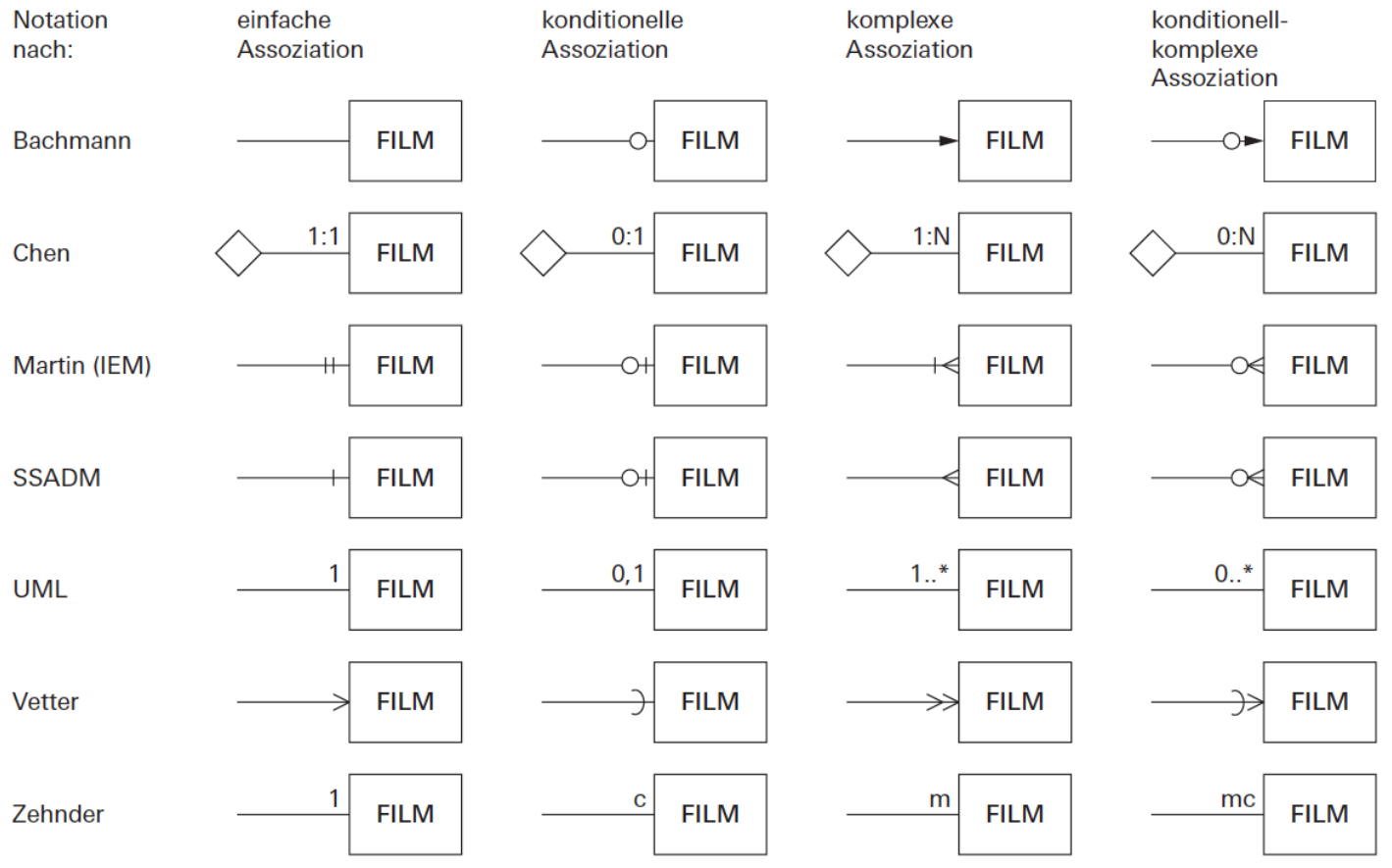
## 3.6 Beziehungen (Assoziationen)

Unter Beziehungen versteht man sozusagen eine Verknüpfung von zwei oder mehreren Tabellen. Meistens werden Tabellen verknüpft, die etwas miteinander gemeinsam haben oder zum gleichen Thema passen. Aus einer Tabelle kann auch eine weiter entstehen. Man kann einige Datensätze sozusagen in eine neue Tabelle verfrachten und diese dank diesen Beziehungen verknüpfen. Bei Beziehungen sind vor allem die Identifikationsschlüssel eine wichtige Rolle, denn durch diese Primary- und Foreignkeys können die Tabellen verknüpft werden. Eine Beziehung wird meistens mithilfe von einem ERD dargestellt. In diesem Diagramm sind die Entitäten mit ihren Attributen abgebildet. Nebenbei sind die Beziehungen zu sehen. Auch kommen in einem solches ERD Kardinalitäten zum Einsatz. Kardinalitäten sind Mengenangaben mit denen der Beziehungstyp festgelegt wird, wie viele Entitäten mit genauer einer Entität des anderen Beziehungstyps in Beziehung stehen.

### 3.6.1 Notationen

Die Notationen sind verschiedene Arten von der Kardinalität. Die Notation kann in verschiedenen Versionen geschrieben werden. Unten sind alle Notationen aufgelistet, die bei einem ERD und ERM verwendet werden. Die Notationen, welche ich schon verwendet habe sind Martin, UML und Zehnder.

Abbildung 5 Notationen



1

0, 1

1 oder mehrere

0, 1, mehrere

## 3.7 Redundanzen

Falls eine Datenbank viele redundante Daten enthält, ist dies ein Zeichen dafür, dass die Datenbank ein schlechtes Datenbankdesign besitzt. Redundanzen sind Daten, die doppelt oder sogar mehrmals in einer Datenbanktabelle vorkommen. Solche Redundanzen kann man vermeiden indem man auf die Normalisierung aufpasst. Die Normalisierung entfernt doppelte Informationen ohne Informationsverlust. Redundante Daten können auch die Performance einer Datenbank beeinflussen.

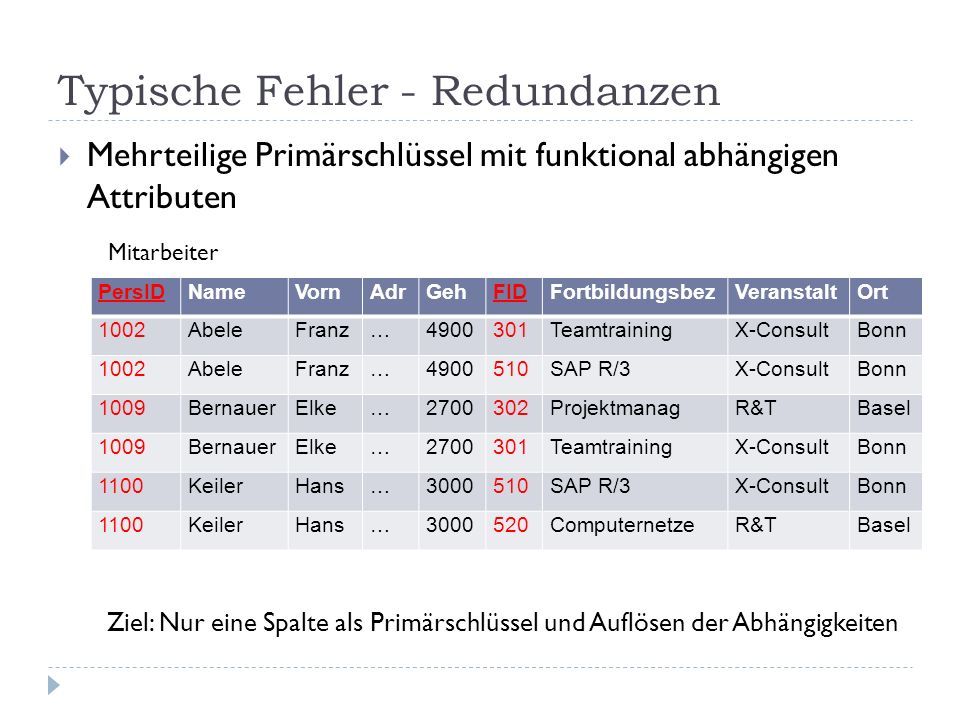


Abbildung 6 Redundanzen

## 3.8 Datenkonsistenz

Wenn man von der Datenkonsistenz spricht, meint man die Korrektheit der in einer Datenbank gespeicherten Daten. Das Gegenteil, die inkonsistenten Daten können zu schweren Fehler im System führen, falls die obere Anwendungsschicht nicht darauf vorbereitet ist.

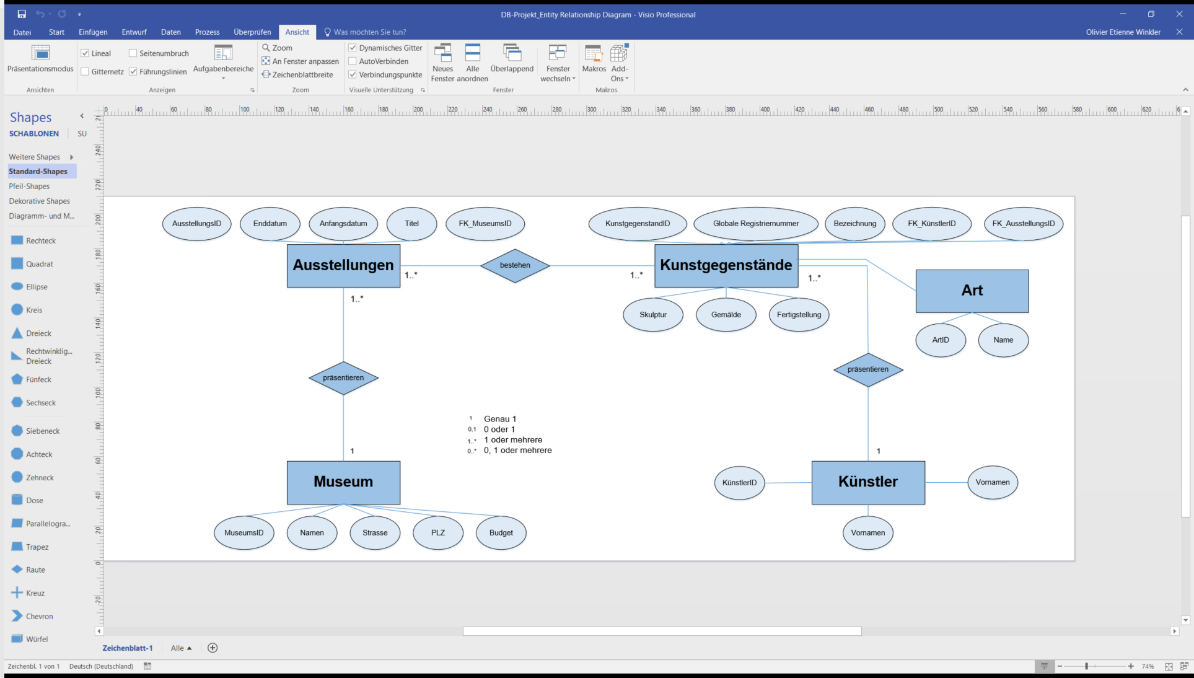
## 3.9 Inkonsistenz

Unter dem Begriff Inkonsistenz versteht man die Widersprüchlichkeit von Daten. In einer Tabelle dienen einige Daten als Verbindungschlüssel. Bei Inkosistenz dieser Daten können so diese nicht mehr als Verknüpfung dienen, da sie nicht mehr eindeutig sind. Meistens werden Inkonsistenzen von den Datenbank-Management-Systemen mit Hilfe von Fehlermeldungen angezeigt.

## 3.10 ERM und ERD

Ein ERM oder auch „Entity Relationship Model“ genannt dient zur grafischen Darstellung für die Erstellung einer Datenbank. Wenn man eine Datenbank plant, erstellt man zuerst das Diagramm. In dem Diagramm zeichnet man alle Tabellen und die dazugehörigen Attribute auf. Danach kommen noch die Beziehungen dazu. In dem Modell kommen dann die Datentypen von den Attributen dazu. Nachdem man das Modell gemacht hat, kann man die Datenbank erstellen. Bei der Erstellung von einer Tabelle muss man die Datentypen der verschiedenen Spalten angeben. Dazu ist das Modell sehr nützlich.

Abbildung 7 ERM und ERD



## 3.11 Normalformen / Normalisierung

Unter der Normalisierung von einem relationalen Datenbankmodell versteht die Aufteilung der Attribute in mehrere Tabellen. Dies geschieht mithilfe von den verschiedenen Normalformen und dadurch wird verhindert, dass Redundanzen entstehen. Das Ziel der Normalisierung ist die redundanzfreie Datenspeicherung von einer Datenbank.

* Nullte Normalform (0NF)
* Erste Normalform (1NF)
* Zweite Normalform (2NF)
* Dritte Normalform (3NF)

In der nullten Normalform ist die Tabelle in der sogenannten Rohform. Dies ist eine Tabelle, die noch nicht normalisiert wurde.



Abbildung 8 0NF

Bei der ersten Normalform ist das Ziel alle Informationen in einer Tabelle atomar zu machen und somit die Suche nach Daten zu vereinfachen. Das heisst zum Beispiel, dass die Postleitzahl und die Ortschaft nicht in einer Spalte, sondern in zwei Spalten aufgeteilt werden.



Abbildung 9 1NF

Die zweite Form ist ein wichtiger Schritt für eine voll normalisierte relationale Datenbank. Diese Form dient zur Überprüfung ob es einige Schlüsselkandidaten gibt, mit denen die restlichen Attribute in der Tabelle abgerufen werden können. Die nicht Schlüsselattribute sind nur von Schlüsselkandidaten voll funktional abhängig.

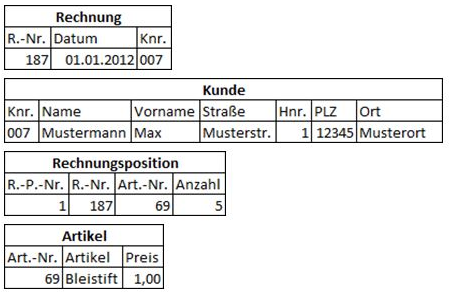


Abbildung 10 2NF

Das Ziel der dritten Form ist es, die Normalisierung erfolgreich abzuschliessen für ein relationales Datenbankmodell und nur noch einen Schlüssel haben (PrimaryKey). Diese Form verhindert Anomalien, Redundanzen und stellt genügend Performance für SQL-Abfragen zur Verfügung. Auch hängt ein nicht Schlüsselattribut transitiv von einem Schlüsselkandidaten ab.

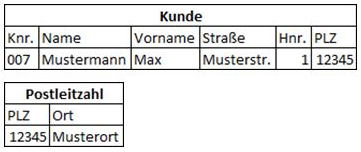


Abbildung 11 3NF

## 3.12 Anomalien

Unter Anomalien versteht man ein Fehlverhalten oder ein Fehler in der Datenbank. Anomalien treten in Datenbanken auf, bei denen keine oder eine fehlerhafte Normalisierung gemacht wurde. Es gibt drei Arten von Anomalien

* Einfüge(Insert)-Anomalien
* Änderungs(Update)-Anomalie
* Lösch(Delete)-Anomalie

Bei einem fehlerhaften oder inkorrekten Datenbankdesign kann es bei der Einfüge-Anomalie passieren, dass Daten nicht in die Datenbank eingefügt werden können. Dies kann geschehen, wenn zum Beispiel der Primärschlüssel keinen Wert erhalten hat, oder eine fehlerhafte Eingabe von Daten zu einer Inkonsistenz führen.

Bei einer Änderungs-Anomalie oder Update-Anomalie genannt, werden die gleichen Attribute eines Datensatzes bei einer Transaktion nicht automatisch geändert. Durch diesen Fehler können Inkonsistenzen entstehen.

Bei einer Lösch-Anomalie kann es passieren, dass ein Benutzer in der Datenbank einige Informationen löschen will und damit indirekt wegen der Anomalie andere zusammenhängende Informationen mitlöscht. So können Löcher in der Datenbank entstehen und eine grosse Menge an wichtigen Daten plötzlich fehlen, obwohl sie niemand absichtlich gelöscht hat.





















































# 4. Installation SQL Server 2017



## 4.1 Installation

Für unser Modul 105 mussten wir auch einen SQL Server installieren. Den Installer kann man von der Microsoftwebseite direkt downloaden und danach konfigurieren. Man muss dabei aber schauen welche Sprache man downloadet. Der Installer funktioniert nicht mit jeder Sprache. Falls man den Computer auf Englisch eingestellt hat, downloadet man die englische Version. Falls man aber Deutsch herunterladen möchte, muss man zuerst die Sprache auf Deutsch (Deutschland) stellen. Mit dem Schweizer-Layout funktioniert der Installer nicht.

Im Installer muss man für die Installation auf Installation klicken.

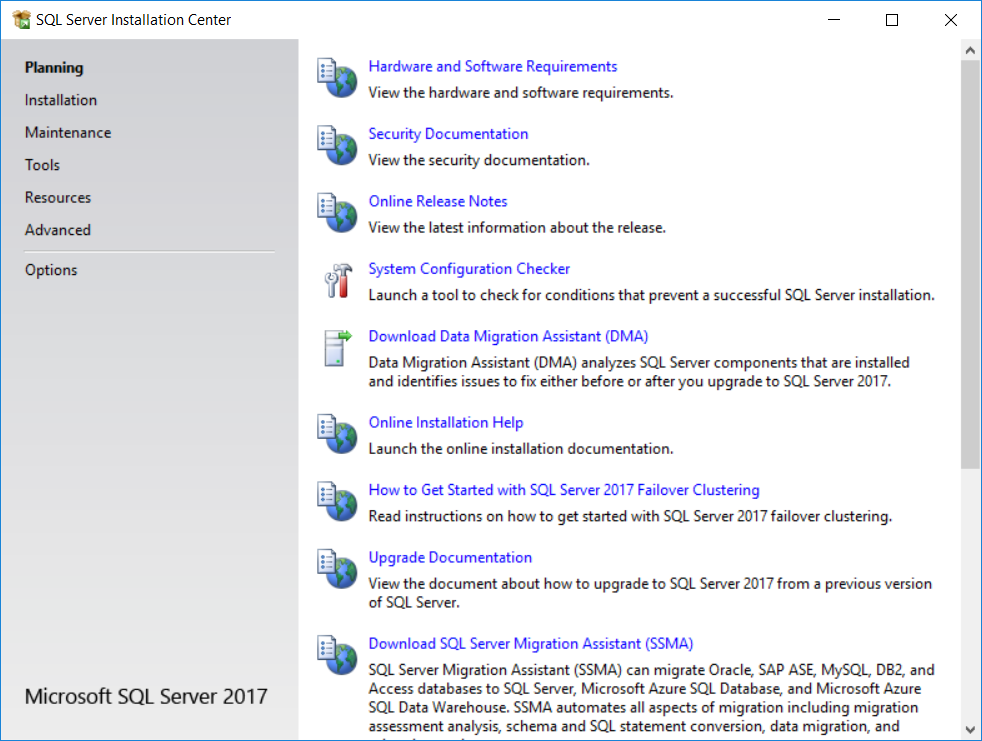


Abbildung 12 Installation 1

Beim nächsten Fenster muss man das oberste wählen. Wie bei einem Cluster könnte man mehrere Server miteinander verbinden. Das wollen wir aber nicht und darum wählt man die Stand-Alone Variante aus.

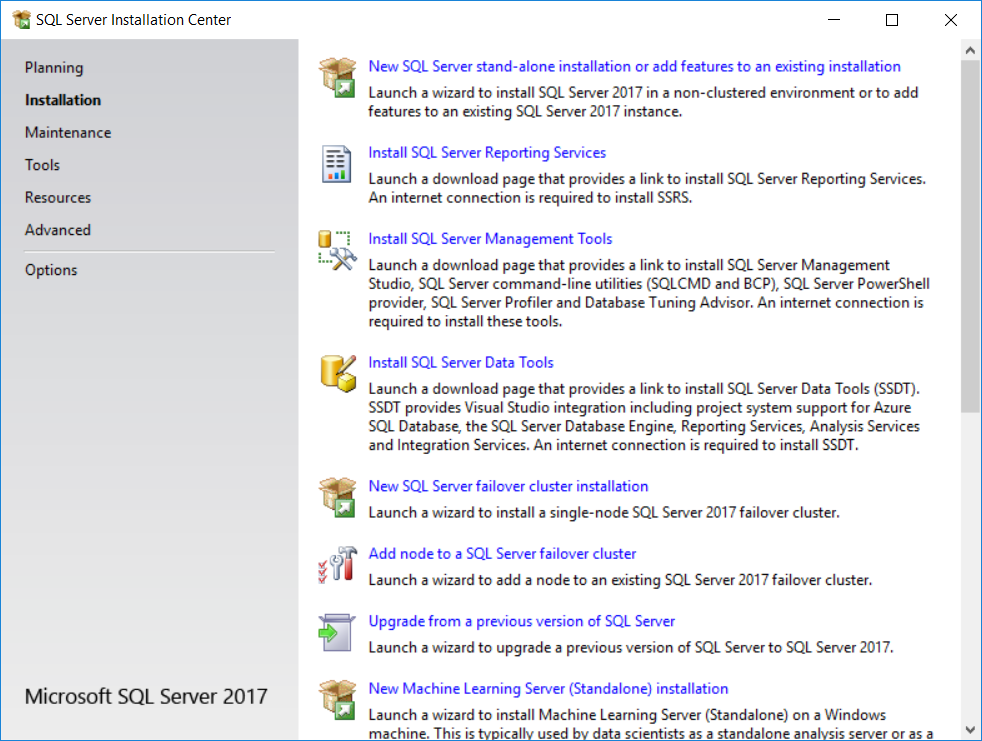


Abbildung 13 Installation 2

Mit dem nächsten Fenster werden verschiedene Funktionen getestet, ob diese auf dem Computer funktionieren.

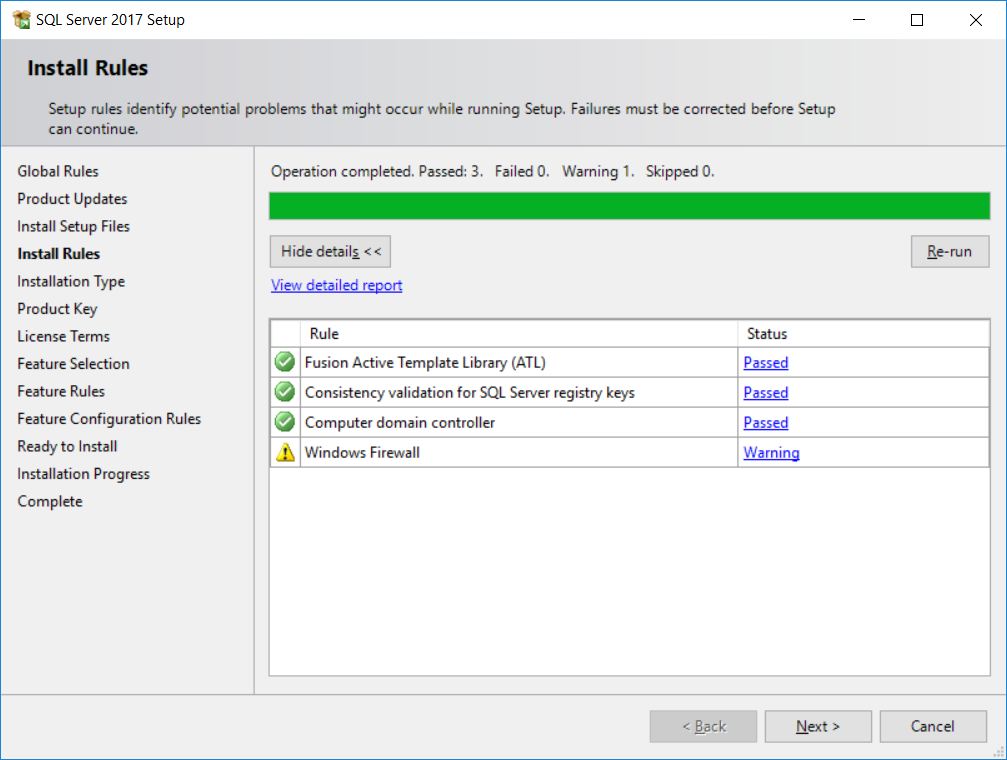


Abbildung 14 Installation 3

Grundsätzlich muss man in den meisten Fenster der Installation die Einstellungen auf Standard lassen. Es kann sein, dass sonst der Server nicht richtig funktioniert.

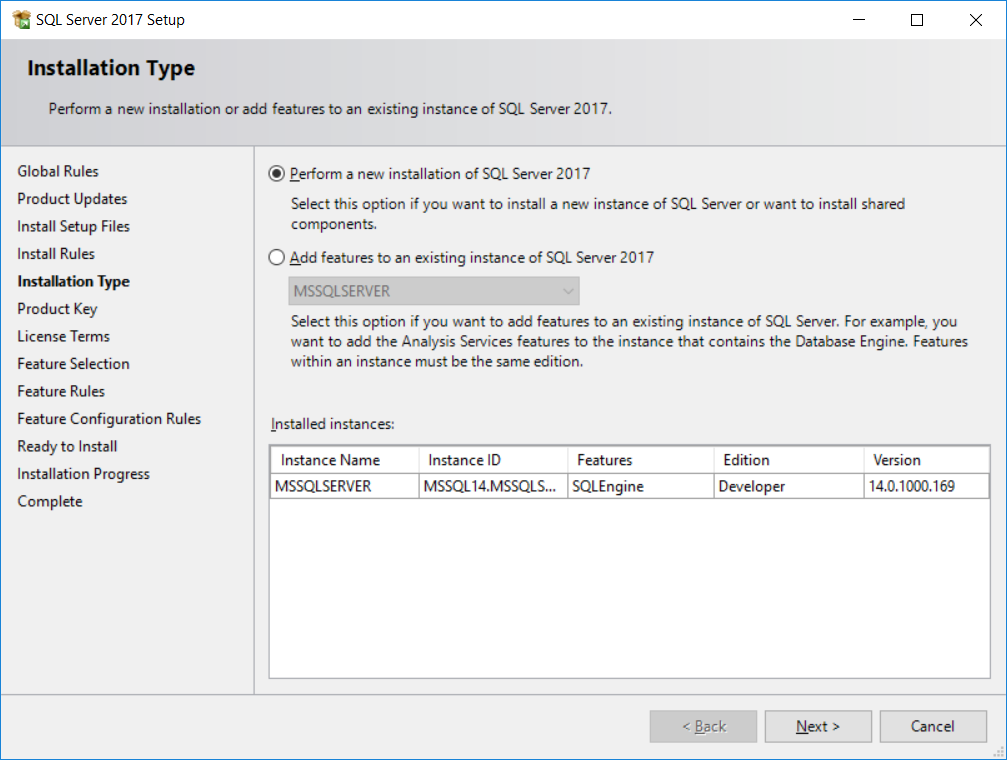


Abbildung 15 Installation 4

Im nächsten Fenster kann man verschiedene Versionen von der Gratisversion auswählen. Standardmässig ist diese Einstellung auf die Version des Installers eingestellt.

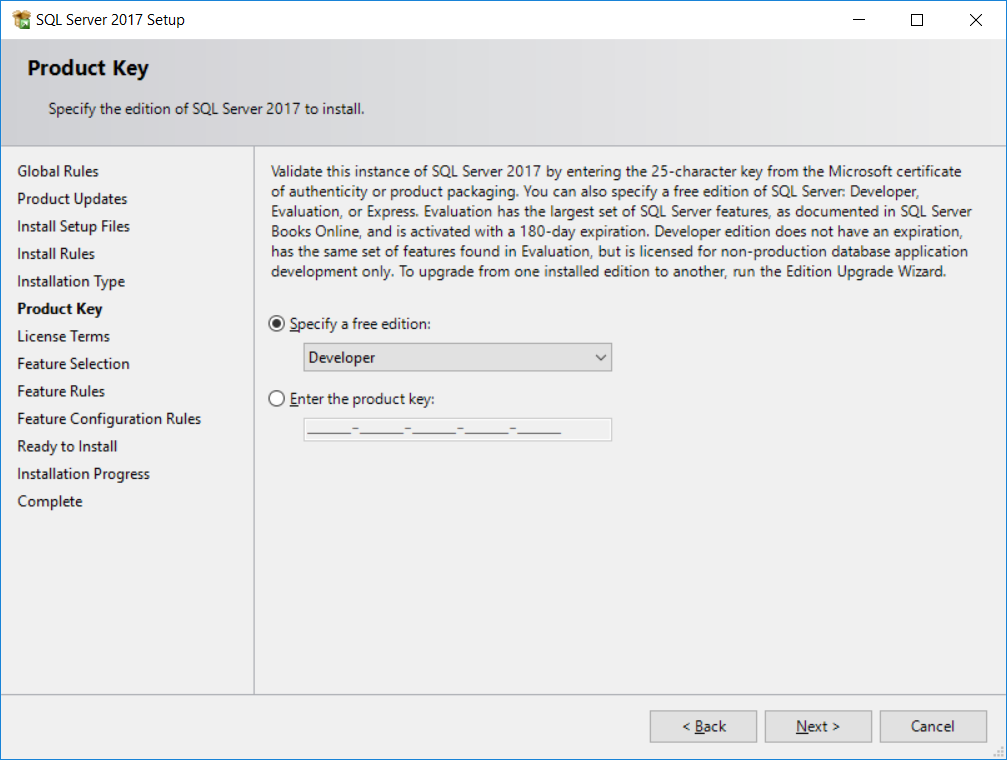


Abbildung 16 Installation 5

Mit dem nächsten Fenster werden alle Features installiert. Für unsere Zwecke brauchen wir nur ***„Database Engine Services“.***

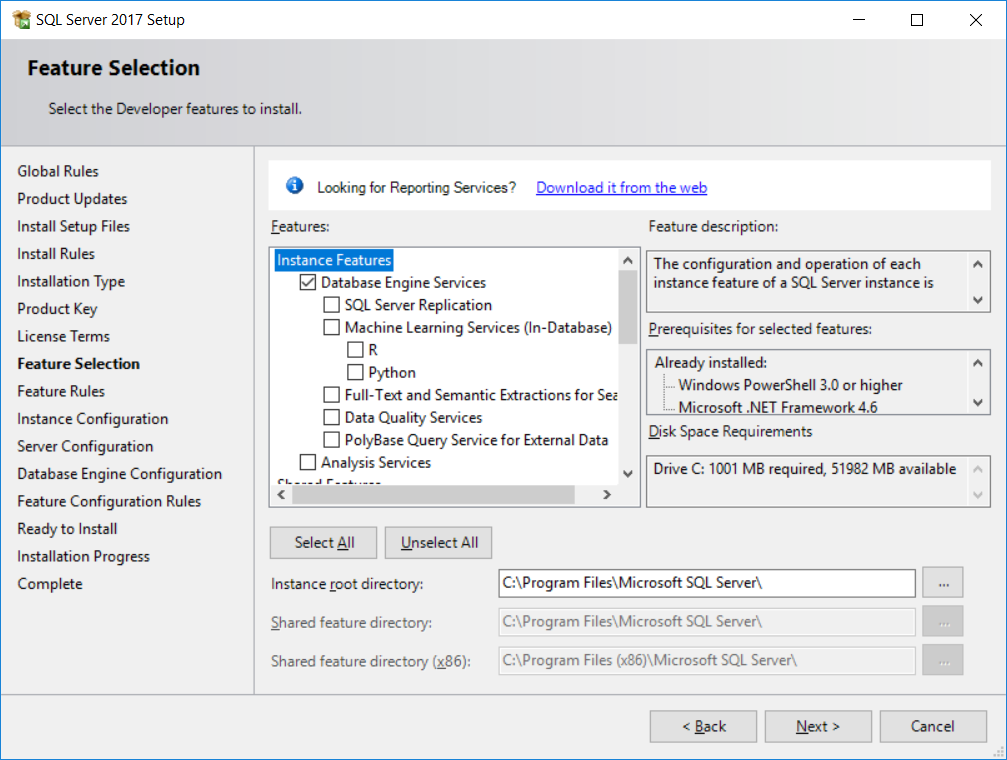


Abbildung 17 Installation 6

Mit dem nächsten Fenster kann man auswählen ob man eine neue Instanz erstellen möchte. Da es aber zu Problemen kommen könnte, lassen wir diese auf Standard.

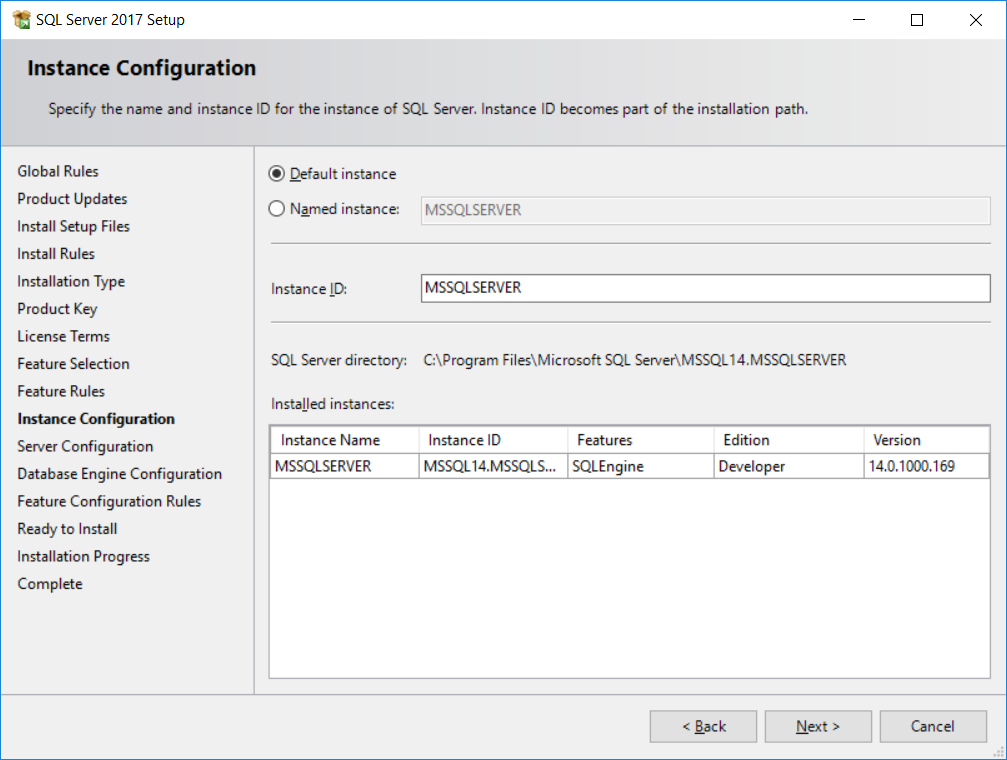


Abbildung 18 Installation 7

Auch dieses Fenster sollte man auf den Standardeinstellungen lassen.

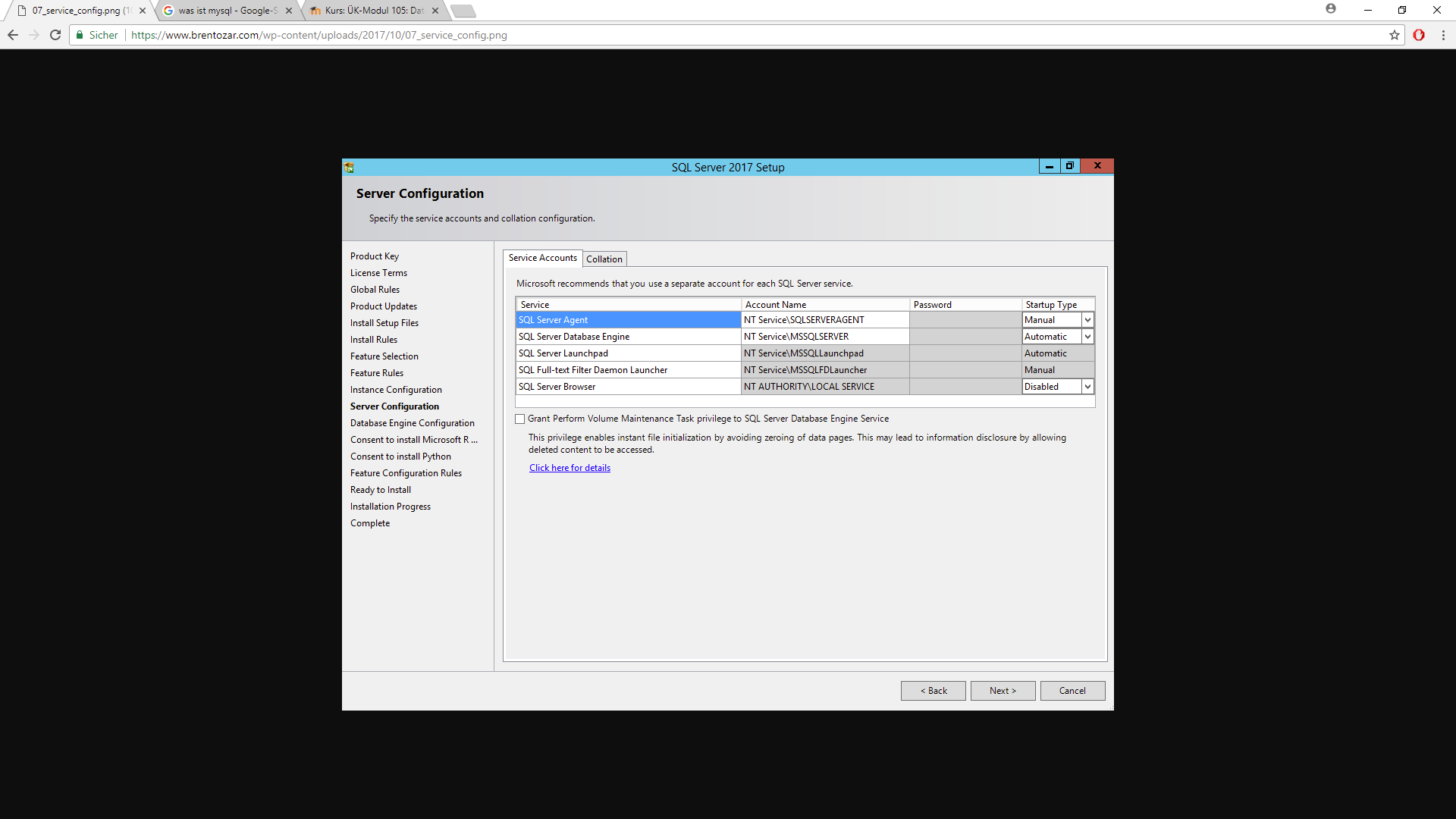


Abbildung 19 Installation 8

Im nächsten Fenster kann man ein Passwort für die Sicherung des SQL Servers eingeben. Auch muss man einen Benutzer aus einer Domäne hinzufügen.

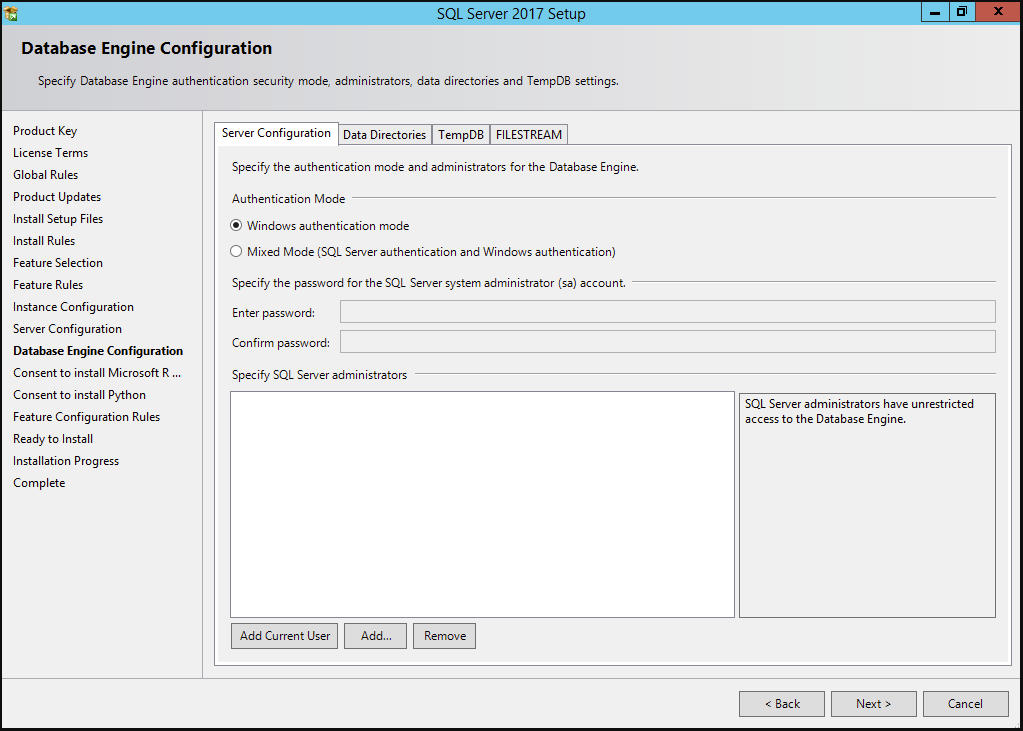


Abbildung 20 Installation 9

Nun kann man die Pfade auswählen wo Daten installiert werden sollen. Die Partition, die man auswählt, sollte genug gross sein, damit man keine Probleme mit Speicherplatz bekommt.

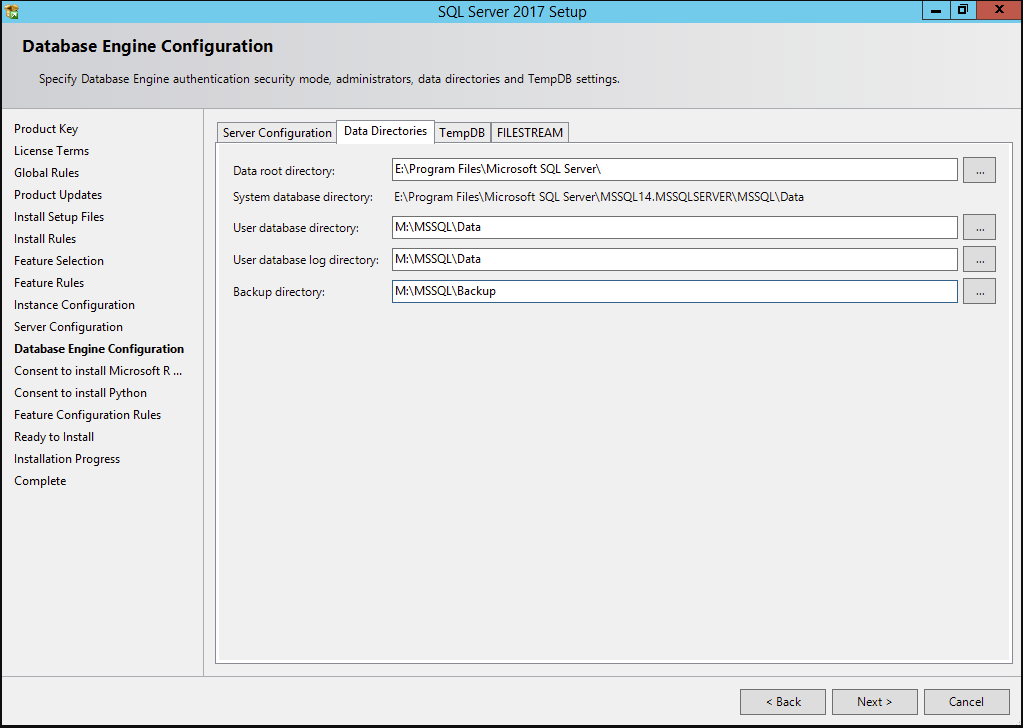


Abbildung 21 Installation 10

Zum Schluss kann man sehen welche Features gerade installiert werden und ob es einen Fehler bei der Installation gab. Zum Schluss muss der Computer noch neu gestartet werden.

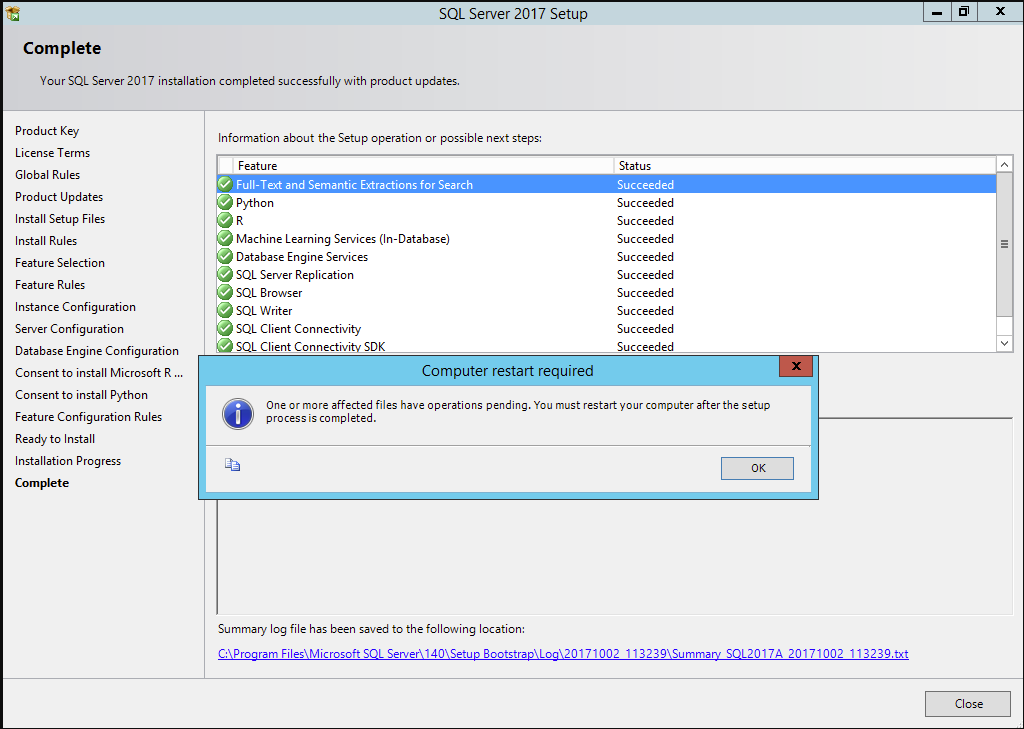


Abbildung 22 Installation 11





















































# 5. Datenbank exportieren / importieren



## 5.1 Datenbank exportieren

Eine Datenbank zu exportieren ist nicht sehr schwierig mit dem GUI. Zuerst klickt man auf „Data Export“ in der Navigationsleiste.

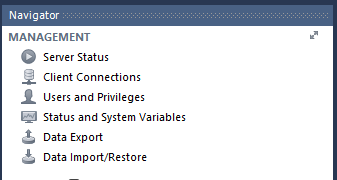


Abbildung 23 Export 1

Das nächste Fenster ist bereits das Letzte. In diesem Fenster kann man auswählen welche Datenbank exportiert werden soll und wohin. Eine Datenbank zu exportieren heisst aber nicht die ganze Datenbank wird exportiert. Nur die Tabellen mit den Informationen werden exportiert.

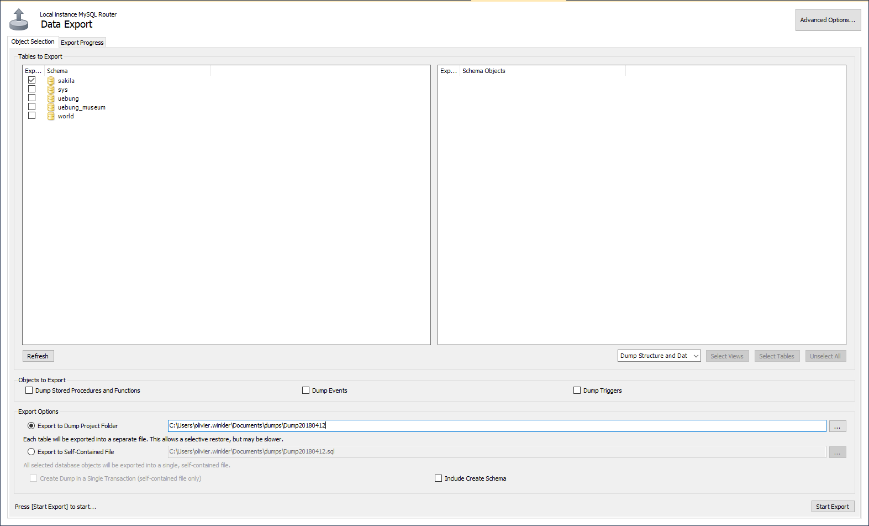


Abbildung 24 Export 2

## 5.2 Datenbank importieren

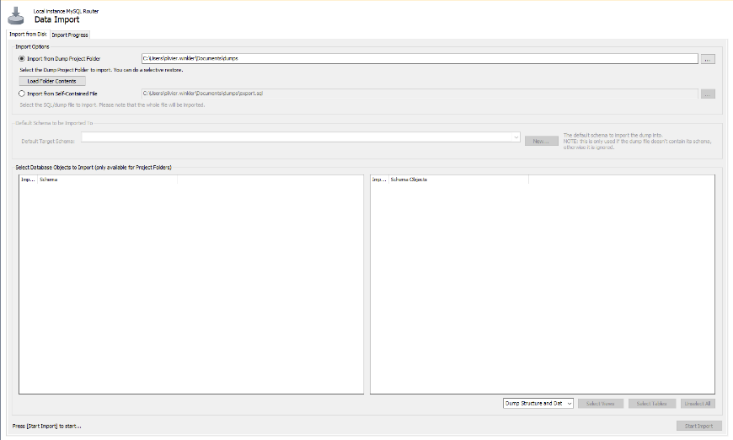
Der Import ist ebenfalls sehr einfach. Man muss im Navigator „Data Import/Restore“ auswählen. Danach muss man auswählen wo die Datenbank gespeichert ist und fertig ist man. Man muss aber zuerst wieder eine Datenbank erstellen, denn bei dem Export sind nur die Tabellen mit Informationen exportiert werden.

Abbildung 25 Import 1





















































# 6. Datenbank – Projekt



## 6.1 Projekt – Museumsdatenbank

Für eine Datenbank braucht man natürlich auch eine Datenbank. In einer Datenbank sind die Tabellen mit den Daten gespeichert. Eine solche Datenbank zu erstellen ist nicht sehr schwierig. In dem Programm MySQL Workbench werden solche Datenbanken als Schemas bezeichnet. Um nun ein Schema zu erstellen macht man in dem Feld „Schemas“ einen Rechtsklick und klickt dann auf „Create Schema“. Danach kann man dem Schema einen Namen geben.

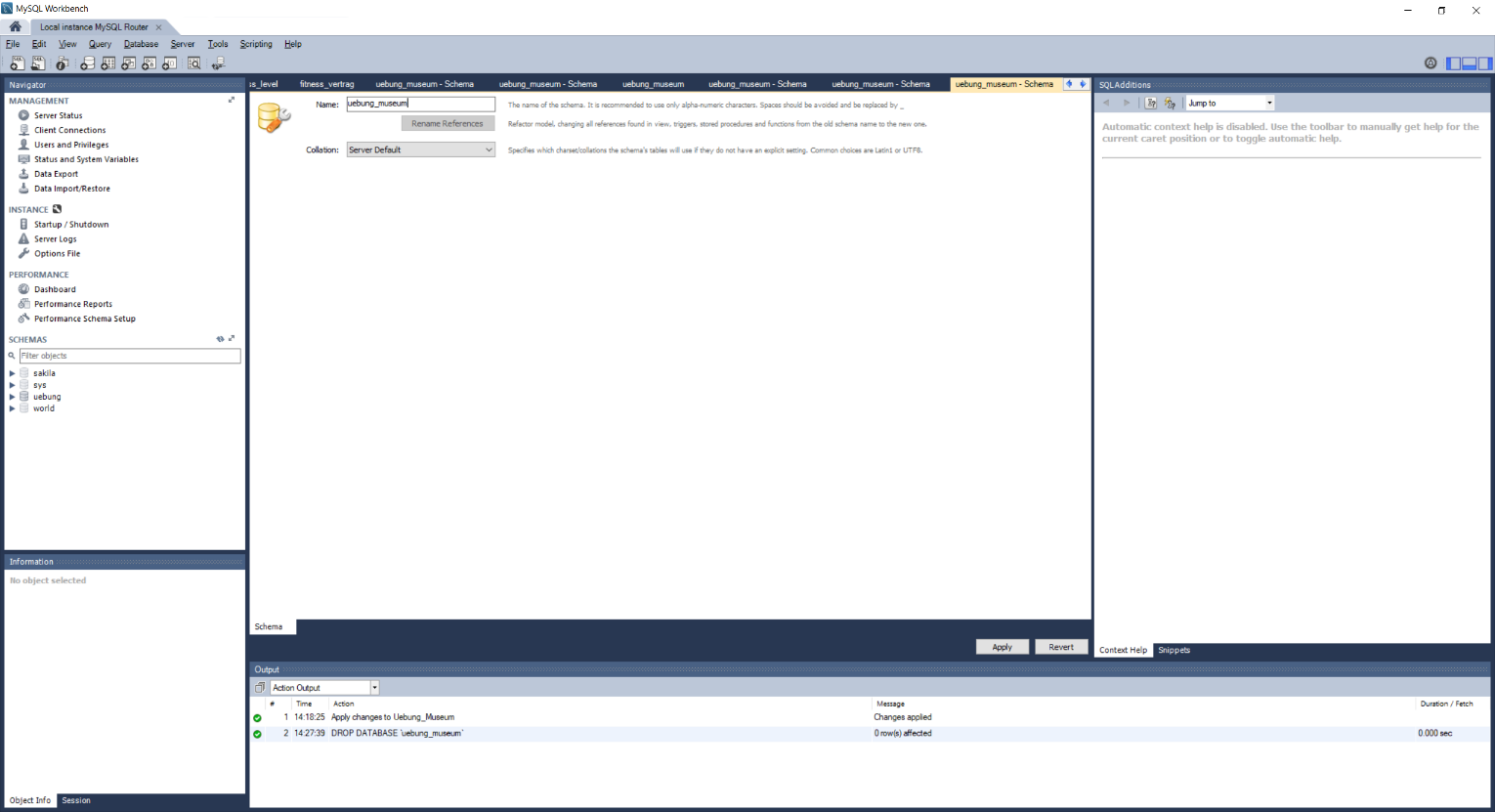
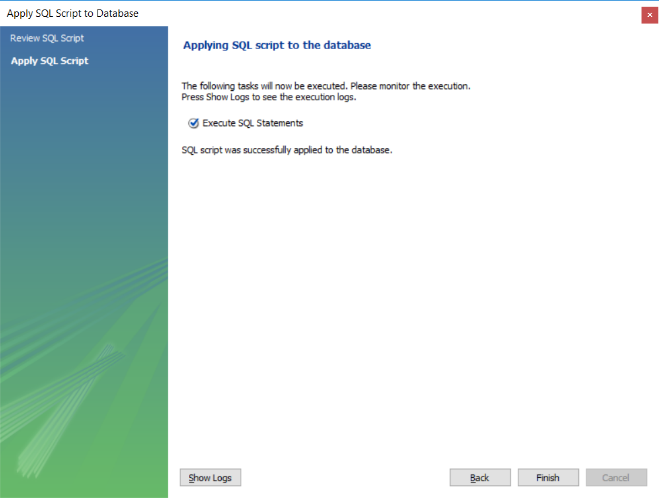
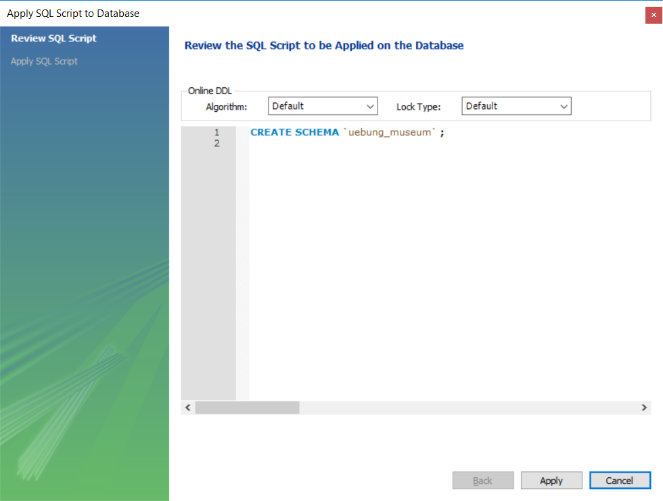


Abbildung 26 Datenbankerstellung 1

Danach kommt ein Fenster, indem man den Erstellungsbefehl noch bearbeiten oder erweitern kann. Im letzten Fenster wird dann gefragt ob der Befehl ausgeführt werden soll und die Datenbank erstellt werden. Nun ist eine Datenbank erstellt ohne Daten.

Abbildung 27 Datenbankerstellung 2



## 6.2 CREATE – Befehle

Nun müssen die Tabellen in unserer Datenbank erstellt werden. Dank dem ERM können wir nur die Datentypen abschreiben für die jeweiligen Spalten. In dem CREATE-Befehl werden alle Spaltennamen aufgelistet mit dem dazugehörigen Datentyp. Danach kann z.B. beim PrimaryKey noch gesagt werden, dass dieses Feld nie leer sein darf (auto\_increment NOT NULL).

Tabelle Museum

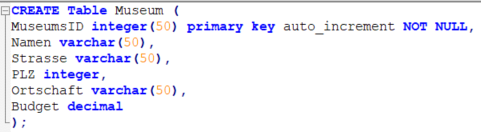


Abbildung 28 CREATE 1

Tabelle Ausstellungen

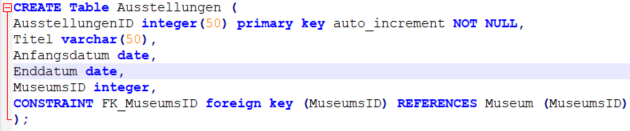
In dieser Tabelle kommt der erste ForeignKey. Zuerst wird als ForeignKey sozusagen registriert und dann mit dem dazugehörigen PrimaryKey verknüpft (REFERENCES …)

Abbildung 29 CREATE 2

Tabelle Kuenstler

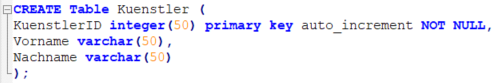


Abbildung 30 CREATE 3

Tabelle Art

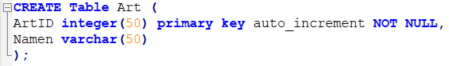


Abbildung 31 CREATE 4

Tabelle Kunstgegenstand

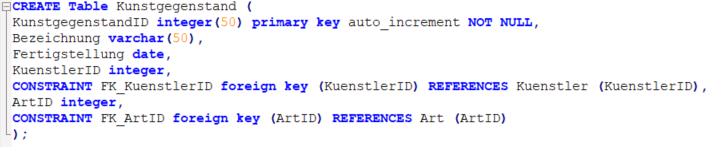
Diese Tabelle ist nun eine von zwei Tabellen, die zwei ForeignKeys besitzt. Das Vorgehen ist das gleiche wie bei nur einem ForeignKey, aber natürlich muss dieser Befehl zweimal eingefügt werden.

Abbildung 32 CREATE 5

Tabelle AusstellungenKunstgegenstand

Diese Tabelle ist unsere Zwischentabelle. In diese kommen nur ForeignKeys von der Tabelle Ausstellungen und Kunstgegenstand. Das Vorgehen von zwei ForeignKeys ist wie bei der Tabelle Kunstgenstand.

Abbildung 33 CREATE 6

## 6.3 INSERT - Befehle

Mit dem INSERT-Befehl werden die Daten oder Werte in die bereits vorgegebenen Tabellen eingefügt. Der INSERT-Befehl besteht aus der Angabe von wo die Datei genommen werden soll und wo sie hineingefügt werden soll und zusätzlich noch wie die Daten in der CSV-Datei voneinander getrennt sind. Für die anderen Tabellen sieht der Befehl eigentlich gleich aus bis auf den Dateinamen und der Zieltabelle.

INSERT in Museum

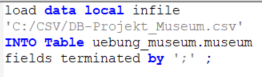


Abbildung 34 INSERT 1

## 6.4 ALTER Tabel - Befehl

In der nächsten Aufgabe von unserem Projekt mussten wir einen Spaltennamen in der Tabelle Kunstgegenstand umbenennen. Dies kann man mit dem ALTER Table – Befehl machen. Bei diesem Befehl wird zuerst die Tabelle ausgewählt und dann wird der alte Spaltennamen aufgelistet gefolgt von dem Neuen und dem Datentyp, der die Spalte haben soll.

Abbildung 35 ALTER 1

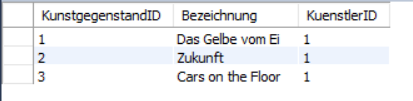
## 6.5 Abfragen mit Datenbank

Nun ist die Datenbank vollständig erstellt. Im Projekt sind dann 10 Aufgaben rund um die Datenbank gekommen. Diese Aufgaben war Abfragen von Daten aus der Datenbank. Hauptsächlich sind JOINS bei diesen Aufgaben zum Einsatz gekommen. Ich werde jeweils die Aufgabenstellung, den Befehl und das Resultat aufschreiben.

Aufgabe 1

Alle Kunstgegenstände eines Künstlers auf Grund des Künstlernamens.

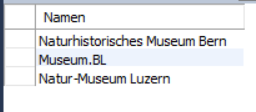
Abbildung 36 Befehl 1



Aufgabe 2

Alle Museen, deren Namen den Begriff «Museum» enthalten

Abbildung 37 Befehl 2



Aufgabe 3

Der Name und die Adresse des Museums mit dem höchsten Umsatz

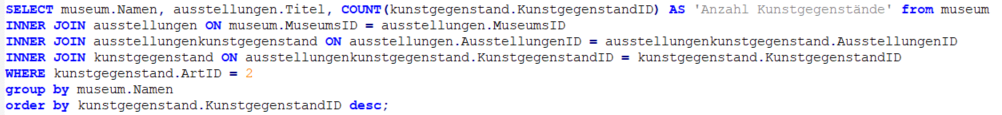
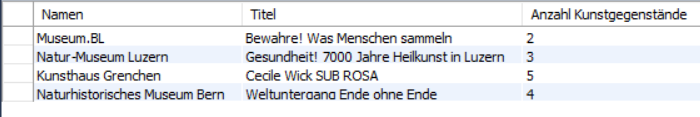
Abbildung 38 Befehl 3



Aufgabe 4

Alle Museumsnamen, Ausstellungsnamen und Anzahl der Kunstgegenstände mit der Art «Gemälde» pro Ausstellung, gruppiert nach Museum

Abbildung 39 Befehl 4



Aufgabe 5

Das Museum, das am meisten Kunstgegenstände zeigt

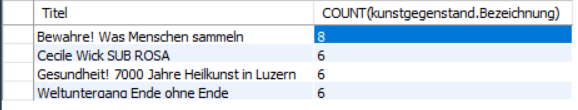
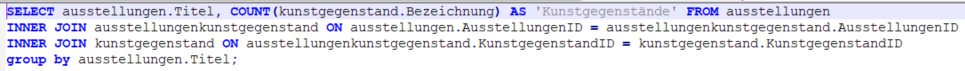
Abbildung 40 Befehl 5



Aufgabe 6

Die Kunstgegenstände, die in mehr als drei Ausstellungen gezeigt werden

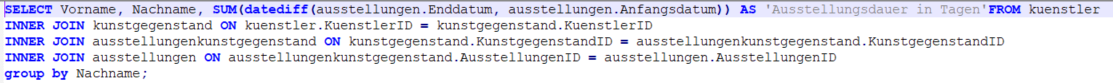
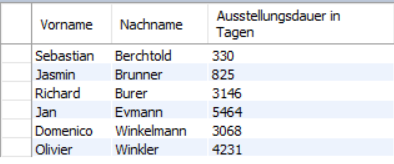
Abbildung 41 Befehl 6



Aufgabe 7

Den Künstlernamen und die totale Ausstellungsdauer in Tagen, dessen Kunstgegenstände am längsten insgesamt ausgestellt werden

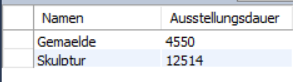
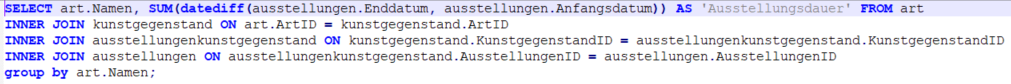
Abbildung 42 Befehl 7



Aufgabe 8

Jeweils die Ausstellungsdauer in Tagen der Kunstgegenstände mit der Art «Gemälde» und der Art «Skulptur»

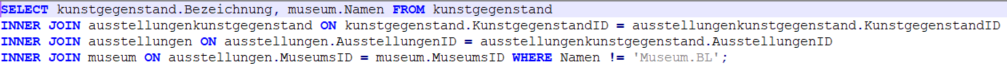
Abbildung 43 Befehl 8



Aufgabe 9

Alle Kunstgegenstände, die nicht Teil einer Ausstellung im «Museum.BL» sind

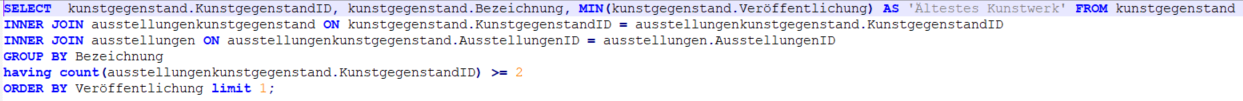
Abbildung 44 Befehl 9



Aufgabe 10

Der älteste Kunstgegenstand, der in mindestens zwei Ausstellungen gezeigt wird

Abbildung 45 Befehl 10



## 6.6 Benutzer hinzufügen und Berechtigungen verteilen

In der letzten Aufgabe von dem Projekt mussten wir drei verschiedene Benutzer mit verschiedenen Berechtigungen erstellen in unserer Datenbank. In der Aufgabe musste man einen „Administrator“, einen „Editor“ und einen „User“ erstellen. Nachdem man die Benutzer mit den dazugehörigen Berechtigungen erstellt hat, musste man sie noch mit einigen Befehlen testen ob die Berechtigungen wirklich greifen.

Unten ist nun der Befehl für die Benutzer- und Berechtigungserstellung dargestellt. Der Benutzer wird mit dem CREATE-Befehl erstellt. Danach kommt der Username und mit dem @-Zeichen wird gesagt von wo der Benutzer alles Zugriff hat. Beim Administrator steht hinter dem @-Zeichen nur localhost. Das heisst, dass der Administrator nur sich am lokalen Computer anmelden kann. Mit einem %-Zeichen nach dem @-Zeichen kann sich der Benutzer von jedem Computer aus anmelden. Danach wird noch das Passwort für den Benutzer festgelegt. In der grant-Zeile werden dann die Berechtigungen vergeben. Bei dem ersten Benutzer „User“ bekommt dieser nur die SELECT-Berechtigung. Mit dem \*.\*-Zeichen kann gesagt werden in welcher Datenbank oder Datenbank-Tabelle der Benutzer angewendet werden kann. Man kann also für jede Tabelle in einer Datenbank einen eigenen Benutzer erstellen. Mit dem \*.\*- Zeichen werden alle Datenbanken im Programm angesprochen. Also kann der Benutzer für jede Datenbank verwendet werden. Das erste Sternchen steht für die Datenbanken und das Zweite für die Tabellen.

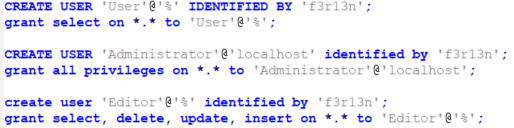
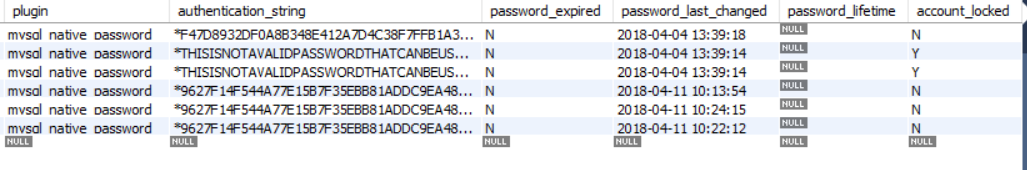
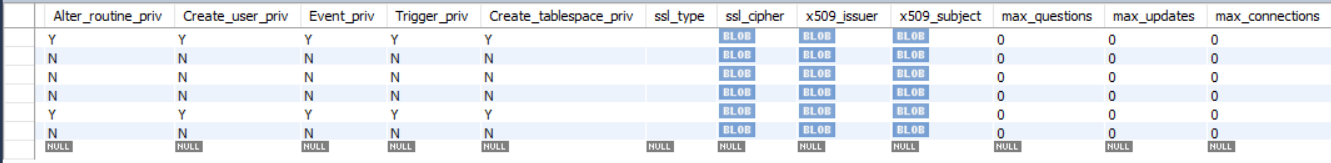
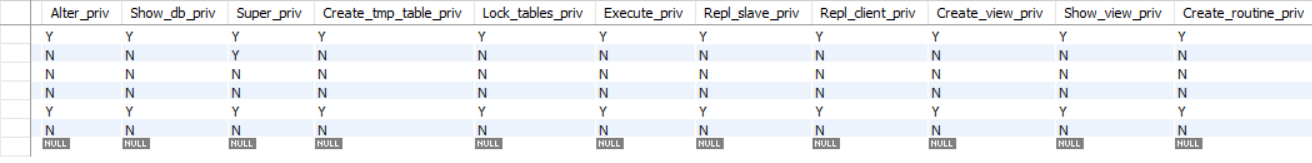
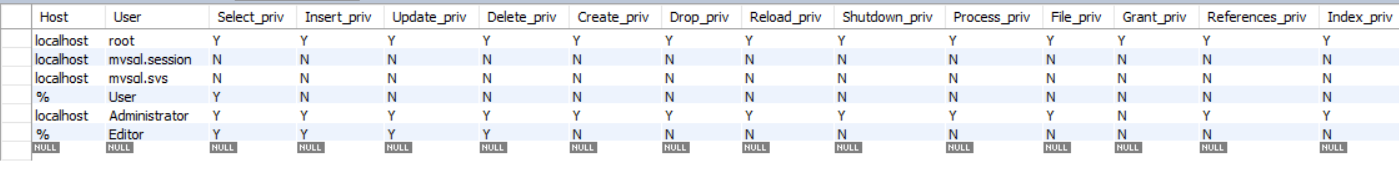


Abbildung 46 Benutzer hinzufügen

### 6.6.1 Berechtigungen abfragen

Danach kann man mit dem SELECT-Befehl von mysql.user sehen welche Benutzer welche Berechtigungen haben. Ausgegeben wird eine Tabelle mit allen Benutzer und Berechtigungen. Entweder steht bei dem Benutzer bei einer Berechtigung ein „Y“ für Yes oder ein „N“ für No.

Abbildung 47 Berechtigungen



Der letzte Schritt war noch die Überprüfung ob die Benutzer wirklich nur das können, was sie auch nach den Berechtigungen dürfen. Mit einigen Abfragen kann man dies testen.

Abbildung 48 Befehl für Berechtigungen















































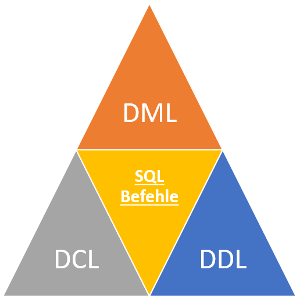








# 7. SQL Befehle



## 7.1 SELECT – SQL

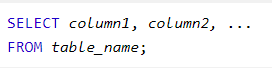
Der SELECT-Befehl kann man eigentlich mit der deutschen Word „auswählen“ vergleichen. Um Daten aus einer Datenbank auszulesen, benötigt man diesen Befehl.

Abbildung 49 SELECT

## 7.2 Das \* Zeichen – SQL

Mit diesem Zeichen werden mit einem SELECT-Befehl alle Daten ausgegeben. Das heisst jede Spalte mit dem gesamten Inhalt. Dieses Zeichen hat den gleichen Wert wie das Wort ALL. Entweder man schreibt ***„SELECT \* FROM tabel“*** oder ***SELECT ALL FROM tabel“.***

Abbildung 50 \*

## 7.3 SELECT DISTINCT – SQL

Dieser Befehl ist ähnlich wie der normale SELECT-Befehl. Mit DISTINCT werden nur Daten aus einer Entitätsmenge herausgelesen, die nur einmal vorkommen. Viele Tabelle haben redundante Daten. Das sind Daten die mehrfach vorkommen. Mit diesem Befehl werden aber nur einzigartige Daten ausgegeben.

Abbildung 51 SELECT DISTINCT

## 7.4 SELECT INTO – SQL

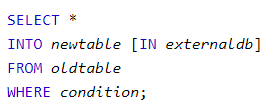
Mit diesem Befehl können auch wieder Datensätze ausgelesen werden. Der Befehl dient aber dazu Daten von einer Tabelle in eine neue Tabelle zu kopieren.

Abbildung 52 SELECT INTO

## 7.5 INSERT INTO SELECT – SQL

Dieser Befehl ist ähnlich wie der SELECT INTO-Befehl. Dieser Befehl hier aber kopiert Daten aus einer Tabelle und fügt sie in eine andere Tabelle ein.

Abbildung 53 INSERT INTO SELECT

## 7.6 WHERE – SQL

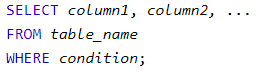
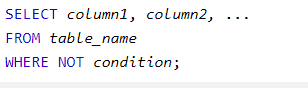
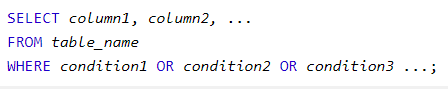
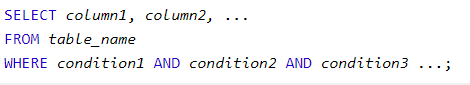
Dieser Befehl oder Zusatz von einem SELECT-Befehl wird für die Filterung eines SELECT-Befehls benutzt. Mit einem WHERE werden diese Daten ausgelesen, die auf eine bestimmte Zuweisung zustimmen.

Abbildung 54 WHERE

## 7.7 AND, OR, NOT – SQL

Diese Operatoren werden mit einem WHERE verwendet. AND- und OR-Operatoren werden benutzt falls es mehrere Zuweisungen gibt. Meistens werden diese beiden Operatoren auch verwendet falls etwas richtig sein muss.

Abbildung 55 AND, OR, NOT



## 7.8 ORDER BY – SQL

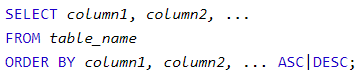
Dieser Befehl verwendet um die ausgegebenen Daten sortiert werden. Zusätzlich zu ORDER BY kommen noch die zwei Befehle ***„ASC“*** und ***„DESC“***. Diese zwei Befehle sortieren die Daten, die mit ORDER BY sortiert wurden auf- oder absteigend. Aufsteigend heisst vom dem kleinsten Wert bis zum grössten Wert. Absteigend heisst logischerweise von dem grössten Wert zum kleinsten Wert. In Tabellen werden meistens Buchstaben verwendet. Der kleinste Buchstabe ist A und der grösste meistens Z.

Abbildung 56 ORDER BY

## 7.9 INSERT INTO – SQL

Dieser Befehl wird verwendet um in eine Tabelle neue Daten einzufügen. Diesen Befehl kann man in zwei Versionen schreiben. In der ersten Version wird angegeben in welche Spalten die Daten kommen. Falls die Daten in alle Spalten kommen sollen, kann man die Spaltenangaben weglassen.

Abbildung 57 INSERT INTO

## 7.10 NULL Values – SQL

In SQL ist es so, dass ein Feld erst kein Volumen mehr hat, wenn in dem Feld ***„NULL“*** angezeigt wird. Falls ein Feld einfach nur leer ist, zählt dies bereits als ein Wert. Mit dem Befehl kann man die Felder auswählen, die keinen Wert haben.

Abbildung 58 NULL

## 7.12 UPDATE – SQL

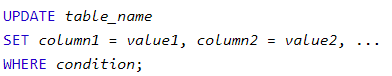
Der Update-Befehl wird wie der Name schon sagt für die Aktualisierung eines Datensatzes genutzt. Falls bei einer Tabelle z.B. einen Spaltennamen umbenennt werden soll, verwendet man diesen Befehl. Man muss aber aufpassen, dass man nicht das WHERE vergisst. Ansonsten wird die ganze Tabelle aktualisiert.

Abbildung 59 UPDATE

## 7.13 DELETE – SQL

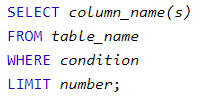
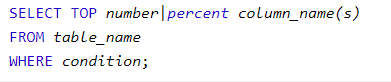
Auch dieser Befehl verrät seine Funktion durch seinen Namen. Der Befehl wird eingesetzt um Daten zu löschen. Auch hier muss man aufpassen, dass man nicht das WHERE vergisst. Ansonsten wird die ganze Tabelle gelöscht.

Abbildung 60 DELETE

## 7.14 SELECT TOP (Limit) – SQL

Mit diesem Befehl kann der Ausgabe ein Limit gesetzt werden. Falls man alle Angaben einer Datenbank sehen möchte, diese aber mehere Tausend Einträge hat, kann man diese limitieren. Dieser Befehl ist aber in verschiedenen Versionen bekannt. Bei SQL Server verwendet man den Befehl ***„SELECT TOP“*** und bei MySQL ***„LIMIT“.***

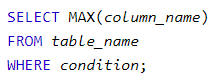
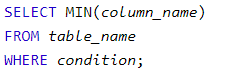
Abbildung 61 SELECT TOP LIMIT



## 7.15 MIN() und MAX() – SQL

Der Befehl MIN() gibt den kleinsten Wert der Spalte aus. MAX() gibt den grössten Wert der Spalte aus.

Abbildung 62 MIN und MAX



## 7.16 COUNT(), AVG() und SUM() – SQL

Der COUNT-Befehl gibt die Anzahl aus. Dies kann die Anzahl Zeilen von einer Tabelle sein. Der AVG-Befehl berechnet den Durchschnitt von einer Spalte. SUM zählt alle Zahlen in einer Spalte zusammen. Z.B. die Spalte Preise, die Preise enthält. Mit SUM() werden alle Preise zusammengezählt und ausgegeben.

Abbildung 63 COUNT, AVG und SUM



## 7.17 LIKE – SQL

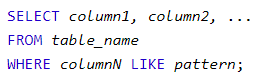
Dieser Befehl wird mit einem WHERE verwendet. Der Befehl wird eingesetzt um Wörter oder Zahlen genau auszugeben. Man kann z.B. sagen, dass alle Wörter, die mit einem A beginnen, ausgegeben werden sollen. Dies kann man mit verschiedenen Operatoren machen.

Abbildung 64 LIKE

|  |  |
| --- | --- |
| **LIKE Operator** | **Beschreibung** |
| LIKE ´a%´ | Zeigt alle Werte an, die mit einem „a“ anfangen. |
| LIKE ´%a´ | Zeigt alle Werte an, die mit einem „a“ enden. |
| LIKE ´%or%´ | Zeigt alle Werte an, die ein „or“ im Wort haben. |
| LIKE ´\_r%´ | Zeigt alle Werte an, die an zweiter Stelle ein „r“ haben. |
| LIKE ´a\_%\_%´ | Zeigt alle Werte an, die mit einem „a“ beginnen und mindestens 3 Buchstaben lang sind. |
| LIKE ´a%o´ | Zeigt alle Werte an, die mit einem „a“ beginnen und mit einem „o“ aufhören. |

## 7.18 IN – SQL

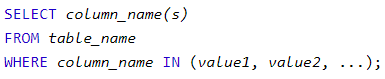
Ein „IN“ wird mit einem WHERE verwendet. Es ist sozusagen ein Zusatz für eine Abfrage ob dieses Objekt vorhanden ist.

Abbildung 65 IN

## 7.19 BETWEEN – SQL

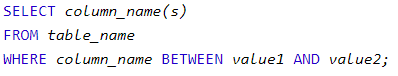
Der BETWEEN-Befehl gibt aus was zwischen zwei Werten sich befindet. Der Befehl kann für vieles eingesetzt werden wie Nummern, Text oder für ein Datum.

Abbildung 66 BETWEEN

## 7.20 ALIASES – SQL

Ein Alias wird im Code als AS dargestellt. Mit diesem Befehl kann man einen Spaltennamen oder der Tabelle einen temporären Namen geben.

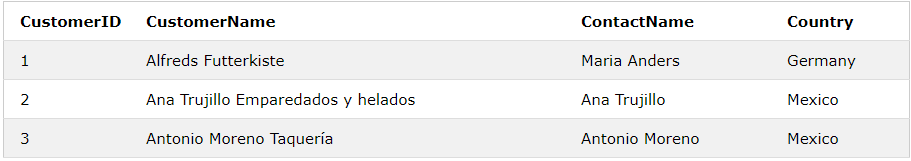
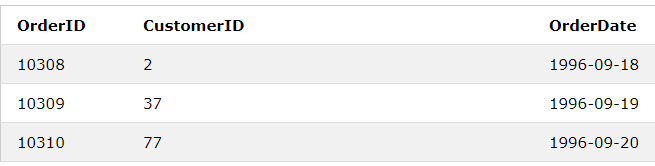
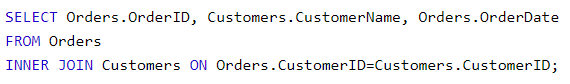
Abbildung 67 AS



## 7.21 JOINS – SQL

Ein JOIN wird verwendet um Zeilen und Spalten von mehreren Tabellen miteinander zu Verknüpfen. Man möchte zum Beispiel eine Tabelle mit Kundendaten und Bestellungen. In der Bestellungen-Tabelle ist nur die KundenID vorhanden. Diese Nummer bringt einem nicht, da man nicht weiss welche Person hinter welcher Nummer ist. Nun kann man mit einem JOIN diese KundenID mit dem richtigen Namen sozusagen importieren und sehen wer genau eine Bestellung gemacht hat oder welche Person diese Nummer hat. Es gibt verschiedene Varianten von Joins.

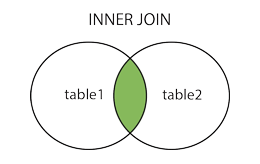
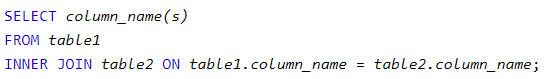
Abbildung 68 JOINS



## 7.22 INNER JOIN – SQL

Ein INNER JOIN zeigt die Daten, die in beiden Tabellen vorhanden sind.

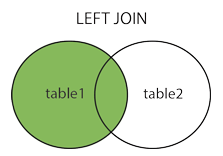
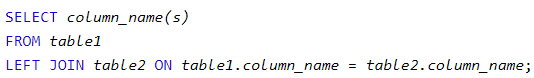
Abbildung 69 INNER JOIN



## 7.23 LEFT JOIN– SQL

Bei diesem JOIN werden alle Werte von der linken Tabelle ausgelesen. Zusätzlich kommen noch diese Werte hinzu, die in der linken als auch in der rechten Tabelle vorhanden sind. Das Resultat ergibt NULL falls keine gleichen Werte gefunden werden.

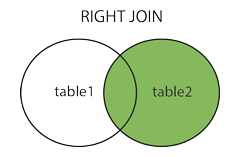
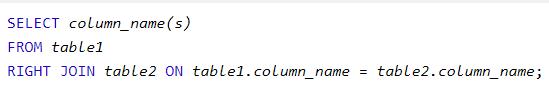
Abbildung 70 LEFT JOIN



## 7.24 RIGHT JOIN– SQL

Dieser JOIN ist eigentlich genau umgekehrt wie der LEFT JOIN. Alle Werte werden von der rechten Tabelle ausgelesen und dazu kommen noch die Werte, die bei beiden Tabellen gleich sind.

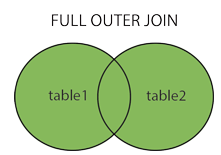
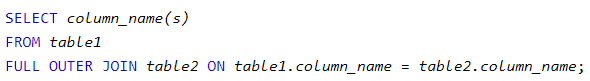
Abbildung 71 RIGHT JOIN



## 7.25 FULL JOIN – SQL

Bei einem FULL JOIN werden alle Werte ausgegeben, falls mindestens ein Wert bei beiden gleich ist.

Abbildung 72 FULL JOIN



## 7.26 SELF JOIN – SQL

Ein SELF JOIN ist ein normaler JOIN. Bei diesem Befehl wird aber die Tabelle mit sich selbst zusammengefügt.

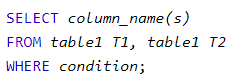


Abbildung 73 SELF JOIN

## 7.27 UNION – SQL

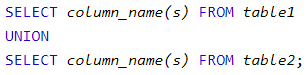
UNION ist ein Operator, welcher das Resultat von zwei oder mehreren SELECT-Befehlen zusammenfasst. Für den UNION-Befehl braucht es in beiden SELECTS gleich viele Zeilen. Auch müssen die Daten gleich sein in den beiden Tabellen.

Abbildung 74 UNION

## 7.28 GROUP BY – SQL

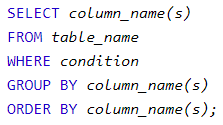
Dieser Befehl dient auch zur Sortierung der Ausgabe. GROUP BY kann man mit vielen anderen Operatoren (COUNT, AVG etc.) verwenden. GROUP BY sortiert die Ausgabe der Reihe nach.

Abbildung 75 GROUP BY

## 7.29 HAVING – SQL

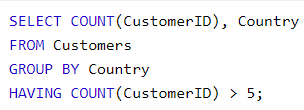
Der HAVING-Befehl wurde zu SQL hinzugefügt, weil man mit WHERE keine Operatoren wie COUNT etc. verwenden kann.

Abbildung 76 HAVING

## 7.30 EXISTS – SQL

Der EXISTS-Befehl wird eingesetzt um ein Subselect auszulesen. Mit dem Befehl kann der Subselect ausgelesen werden und in den normalen SELECT integriert werden.

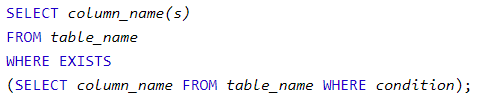
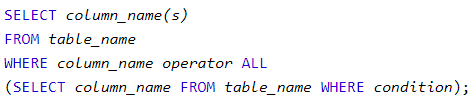
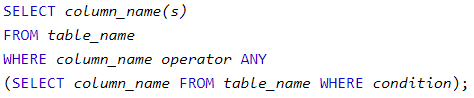


Abbildung 77 EXISTS

## 7.31 ANY und ALL – SQL

Diese beiden Befehle werden mit einem WHERE oder HAVING genutzt. Auch mit ALL und ANY können Subselects ausgelesen werden. Die beiden Befehle benutzen immer Operatoren (=, <>, !=, >, >=, <, <=)

Abbildung 78 ANY und ALL



## 7.32 Datentypen von SQL

In SQL gibt es auch verschiedene Datentypen für die Spalten. Jeder Datentyp ist für eine bestimmte Bestimmung gedacht. Zum Beispiel gibt es für eine Datumsspalte extra einen Datentyp dafür.

|  |  |
| --- | --- |
| Datentyp | Beschreibung |
| integer | ganze Zahlen |
| float | Gleitkommazahlen |
| char | Zeichenkette mit der festen Länge <= 255 |
| varchar | Zeichenkette mit der maximalen Lange <= 255 |
| date | Datum in ‚XXXX-XX-XX‘ |
| time | Zeit ‚YY:YY:YY‘ |
| datetime | Zeitstempel ‚XXXX-XX-XX YY:YY:YY‘ |
| text | Text mit max Länge von 65535 |
| blob | Binäre Zeichenfolge mit max Länge von 65535 |
| enum(‚vaule‘, ) | Zeichenkette, die NULL annehmen kann. |
| set(‚vaule‘, ) | Menge aus den angegebenen Werten. |

# 8. Reflexion

Das Modul 105 war kurz und knackig. In diesem Modul habe ich zwar nicht so viel gelernt, da wir schon sehr viel Stoff in der Berufsschule hatten. Trotzdem konnte ich viel Übung durch dieses Modul bekommen. Ich konnte im Unterricht gut meine Arbeit verrichten und die Aufgaben lösen. Mit meinen Leistungen bin ich zufrieden. Für die kurze Zeit habe ich ein gutes Journal geschrieben, welches informativ ist und mir falls ich später einmal etwas zu diesem Thema nachschlagen muss, helfen. Nun kann ich dank diesem Modul besser auf den neuen Stoff in der Schule aufbauen, da jetzt mein Wissen in diesem Modul vertieft wurde. Durch die Arbeitsblätter, die Stephanie und zur Verfügung gestellt hat, konnte ich viel Praxis erlernen. Ich fand die Arbeitsblätter sehr gut und besser als bisherige Arbeitsblätter. Ich denke auch, dass ich mein Bestes in diesem Modul gegeben habe und es nichts gibt, indem ich mich verbessern könnte.

# 9. Quellenverzeichnis

Titelseite Grundlagen

<https://azure.microsoft.com/svghandler/sql-database/?width=600&height=315>

Bild Hierarchisches Datenbankmodell

<http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/hierarchisches-datenbankmodell/>

Bild Netzwerk-Modell

<http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/netzwerkdatenbankmodell/>

Bild Relationales Datenbankmodell

<http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/relationales-datenbankmodell/>

Bild Objektorientiertes Datenbankmodell

<http://www.datenbanken-verstehen.de/datenbank-grundlagen/datenbankmodell/objektorientiertes-datenbankmodell/>

Bild Notation von GiBB-Arbeitsblatt

Bild Redundanzen

<http://slideplayer.org/slide/661394/1/images/31/Typische+Fehler+-+Redundanzen.jpg>

Bild ERM und ERD von meinem Datenbankprojekt

Bilder Normalformen

<http://www.datenbanken-verstehen.de/datenmodellierung/normalisierung/erste-normalform/>

Titelseite SQL Installation

<https://timetoact-group.de/wp-content/uploads/Microsoft_SQL_Server_Logo1.jpg>

Bilder von SQL Server Installation

<https://www.brentozar.com/archive/2017/10/sql-server-2017-installation-screenshot-tour-windows/>

Titelseite Import Export

<https://premium.wpmudev.org/blog/wp-content/uploads/2013/03/import-export-featured-image.jpg>

Bilder Datenbank Import / Export von meinem Projekt

Titelseite Projekt

<http://cdn.shopify.com/s/files/1/0164/8994/t/4/assets/logo.png?10115403884272341966>

Bilder von meinem Datenbankprojekt von Projekt

Titelseite Befehle

<http://www.sql-lernen.de/bilder/sql-befehle-pyramide.png>

Bilder von Befehlen

<https://www.w3schools.com/sql/default.asp>

# 10. Abbildungsverzeichnis

[Abbildung 1 Hierarchisches Modell 16](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392955)

[Abbildung 2 Netzwerk-Modell 17](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392956)

[Abbildung 3 Relationales Modell 17](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392957)

[Abbildung 4 Objektorientiertes Modell 18](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392958)

[Abbildung 5 Notationen 20](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392959)

[Abbildung 6 Redundanzen 21](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392960)

[Abbildung 7 ERM und ERD 22](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392961)

[Abbildung 8 0NF 23](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392962)

[Abbildung 9 1NF 23](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392963)

[Abbildung 10 2NF 23](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392964)

[Abbildung 11 3NF 24](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392965)

[Abbildung 12 Installation 1 26](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392966)

[Abbildung 13 Installation 2 27](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392967)

[Abbildung 14 Installation 3 27](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392968)

[Abbildung 15 Installation 4 28](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392969)

[Abbildung 16 Installation 5 28](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392970)

[Abbildung 17 Installation 6 29](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392971)

[Abbildung 18 Installation 7 29](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392972)

[Abbildung 19 Installation 8 30](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392973)

[Abbildung 20 Installation 9 30](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392974)

[Abbildung 21 Installation 10 31](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392975)

[Abbildung 22 Installation 11 31](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392976)

[Abbildung 23 Export 1 33](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392977)

[Abbildung 24 Export 2 33](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392978)

[Abbildung 25 Import 1 33](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392979)

[Abbildung 26 Datenbankerstellung 1 35](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392980)

[Abbildung 27 Datenbankerstellung 2 35](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392981)

[Abbildung 28 CREATE 1 36](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392982)

[Abbildung 29 CREATE 2 36](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392983)

[Abbildung 30 CREATE 3 36](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392984)

[Abbildung 31 CREATE 4 36](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392985)

[Abbildung 32 CREATE 5 37](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392986)

[Abbildung 33 CREATE 6 37](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392987)

[Abbildung 34 INSERT 1 37](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392988)

[Abbildung 35 ALTER 1 38](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392989)

[Abbildung 36 Befehl 1 38](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392990)

[Abbildung 37 Befehl 2 38](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392991)

[Abbildung 38 Befehl 3 38](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392992)

[Abbildung 39 Befehl 4 39](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392993)

[Abbildung 40 Befehl 5 39](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392994)

[Abbildung 41 Befehl 6 39](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392995)

[Abbildung 42 Befehl 7 40](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392996)

[Abbildung 43 Befehl 8 40](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392997)

[Abbildung 44 Befehl 9 41](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392998)

[Abbildung 45 Befehl 10 41](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511392999)

[Abbildung 46 Benutzer hinzufügen 42](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393000)

[Abbildung 47 Berechtigungen 43](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393001)

[Abbildung 48 Befehl für Berechtigungen 43](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393002)

[Abbildung 49 SELECT 45](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393003)

[Abbildung 50 \* 45](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393004)

[Abbildung 51 SELECT DISTINCT 45](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393005)

[Abbildung 52 SELECT INTO 45](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393006)

[Abbildung 53 INSERT INTO SELECT 45](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393007)

[Abbildung 54 WHERE 45](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393008)

[Abbildung 55 AND, OR, NOT 46](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393009)

[Abbildung 56 ORDER BY 46](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393010)

[Abbildung 57 INSERT INTO 46](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393011)

[Abbildung 58 NULL 47](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393012)

[Abbildung 59 UPDATE 47](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393013)

[Abbildung 60 DELETE 47](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393014)

[Abbildung 61 SELECT TOP LIMIT 47](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393015)

[Abbildung 62 MIN und MAX 48](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393016)

[Abbildung 63 COUNT, AVG und SUM 48](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393017)

[Abbildung 64 LIKE 48](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393018)

[Abbildung 65 IN 49](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393019)

[Abbildung 66 BETWEEN 49](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393020)

[Abbildung 67 AS 49](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393021)

[Abbildung 68 JOINS 49](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393022)

[Abbildung 69 INNER JOIN 50](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393023)

[Abbildung 70 LEFT JOIN 50](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393024)

[Abbildung 71 RIGHT JOIN 51](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393025)

[Abbildung 72 FULL JOIN 51](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393026)

[Abbildung 73 SELF JOIN 52](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393027)

[Abbildung 74 UNION 52](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393028)

[Abbildung 75 GROUP BY 52](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393029)

[Abbildung 76 HAVING 53](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393030)

[Abbildung 77 EXISTS 53](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393031)

[Abbildung 78 ANY und ALL 53](file:///H:\Modul%20105\M105_Journal_INF2017_Winkler_Olivier.docx#_Toc511393032)

# 11. Selbstständigkeitserklärung

**Für die Selbstständigkeitserklärung folgenden Wortlaut in die letzte Seite des Moduljournals kopieren:**

Ich habe diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen verwendet. Wörtlich zitierte Textstellen sind als solche gekennzeichnet und gemäss den Vorgaben im «Merkblatt zu Zitat- und Quellangaben» ausgewiesen und formatiert.

Ich bestätige, dass ich das in den Richtlinien zum Moduljournal erhaltene Merkblatt zu Zitat- und Quellenangeben gelesen und verstanden habe. Mir ist klar, dass mein Moduljournal elektronisch auf Plagiate überprüft wird. Die Sanktionen beim Erstellen von Plagiaten sind mir bekannt

Bahnhöhenweg 70 / 3018 Bern

13.04.18

Olivier Winkler