

Dokumentation ÜK 106

ZLI, 06.04.2022

ROBIN RÜEGG

Inhaltsverzeichnis

Block 1: Repetition und Einführung in MariaDB	3
Allgemeines über Datenbanken	4
Allgemeine Infos.....	4
Welche Aufgaben hat eine Datenbank?	4
Welche Datenbankwerkzeuge kennen Sie?	4
Welche Aufgaben hat eine Datenbanksprache?.....	5
Welche Vorteile haben relationale gegenüber hierarchischen Datenbanken?.....	5
Repetitionsfragen	5
Was ist eine Rationale Datenbank?	5
Nennen Sie drei verschiedene relationale Datenbank Management Systeme DBMS?	5
Welche Aufgaben erfüllt ein DBMS für Sie als Entwickler?	5
Handelt es sich bei SQL um eine Programmiersprache?	5
Ist die Sprache SQL auf allen Datenbanken gleich?.....	5
Was versteht man unter NoSQL Datenbank?	5
Was versteht man unter Datenbank Schema?	5
Wozu dient ein Entity Relationship Model (ERM)?	5
Wie sieht ein Enhanced Entity Relationship (EER) Model aus?	5
Was bedeuten die Begriffe: Entität, Attribut und Tupel?	5
Was versteht man unter Datenkonsistenz?	5
Was bedeutet Redundanz im Zusammenhang mit einer Datenbank?	5
Was versteht man unter Normalform?.....	6
Was versteht man unter Primär- und Fremdschlüssel?.....	6
Wie unterscheiden sich die Datentypen Char und Varchar?.....	6
Was ist ein Character Set und wie unterscheiden sich ASCII, LATIN1 und UTF-8?	6
Wie unterscheiden sich die Datentypen Integer und Float?	6
Wie unterscheiden sich die Datentypen Timestamp und Datetime?.....	6
Erste Schritte mit MariaDB.....	7
Befehle	7
Was bedeuten die Programmparameter "-u root" und "-p"?	7
Welche Rechte und Aufgaben hat der Benutzer "root"	7
Warum heisst das Konsolen-Programm von MariaDB mysql.exe?	7
Database SQL Operation	7

Cheat Sheet SQL:.....	8
Befehle:	8
Block 2: Repetition Schema entwerfen, Tabellen anlegen	12
Grundkurs Relationale Datenbanken	14
Tabellen erstellen	14
Datentypen	16
Tabellen ändern	14
Attribute ändern.....	15
Tabellenattribute indizieren	15
Index löschen	15
Beziehungen erstellen Foreign Key, Primary Key	15
Beziehungen entfernen	15
Primärschlüssel entfernen	15
Tabellen löschen.....	15
Neue Datensätze erzeugen.....	15
Theorie zu Entity Relationship Models ERM	16
Anomalien.....	16
Datentypen	16
Begriffe.....	18
Normalformen	18
ERM erstellt.....	19
SQL Create Befehl ERM	20

Block 1: Repetition und Einführung in MariaDB

- Ich kenne die verschiedenen Arten von Datenbanken und deren hauptsächlichsten Unterschiede
- Ich kenne die Aufgaben eines RDBMS sowie den Aufbau eines Datenbanksystems
- Ich kenne die verschiedenen Elemente der strukturierten Abfragesprache SQL
- Ich kann MariaDB installieren
- Ich kann die Datenbank-Konsole öffnen und SQL Statements eingeben
- Ich kann alle Datenbanken im System anzeigen
- Ich kann mich mit einer Datenbank verbinden
- Ich kann die Datenbank Struktur anzeigen

Allgemeines über Datenbanken

Allgemeine Infos

Eine Datenbank soll beliebige Daten verwalten, Informationen aus diesen Daten liefern und unberechtigten Personen den Zugriff auf die Daten verweigern können. Unter dem Verwalten von Daten versteht man das Eingeben von neuen Daten, das Löschen veralteter Daten sowie das Nachführen bestehender Daten.

Welche Aufgaben hat eine Datenbank?

- Sie soll dem Benutzer den Zugriff auf die gespeicherten Daten ermöglichen, ohne dass dieser wissen muss, wie die Daten im System organisiert sind.
- Sie muss verhindern, dass ein Benutzer Daten sichten oder manipulieren kann, für die er keine Zugriffsberechtigung hat. Außerdem darf es nicht passieren, dass wegen Fehlmanipulationen des Benutzers Daten zerstört werden können oder gar der ganze Datenbestand unbrauchbar wird.
- Es muss möglich sein, die interne Datenorganisation ändern zu können, ohne dass der Benutzer seine Anwenderprogramme (Applikationen) anpassen muss. Im Idealfall merkt der Benutzer von der Strukturänderung nichts.

Welche Datenbankwerkzeuge kennen Sie?

Liste verschiedener Datenbank-Verwaltungswerkzeuge [\[Bearbeiten | Quelltext bearbeiten \]](#)

Verwaltungswerkzeug	Unterstützte Datenbanksysteme
Aqua Data Studio	Apache Derby, DB2, MS SQL Server, MySQL, Oracle, PostgreSQL, Sybase, Teradata
EMS SQL Management Studio	DB2, Firebird, Interbase, MS SQL Server, MySQL, Oracle, PostgreSQL
IBExpert	InterBase, Firebird
Microsoft SQL Server Management Studio	MS SQL Server
MySQL Workbench	MySQL
Oracle SQL Developer	Access, DB2, MS SQL Server, MySQL, Oracle, Sybase, Teradata
Orbada	Apache Derby, Firebird, Interbase, H2, HSQLDB, MS SQL Server, MySQL, Oracle, PostgreSQL, SQLite
pgAdmin	PostgreSQL, Postgres Plus, Greenplum, PostGIS
TOAD	DB2, MySQL, Oracle, MS SQL Server
Database Studio	MaxDB
IBM Data Studio	DB2

Welche Aufgaben hat eine Datenbanksprache?

Dabei haben Datenbanksprachen die Aufgabe, die Kommunikation zwischen dem Benutzer oder einem Anwendungsprogramm und dem Datenbanksystem zu ermöglichen. Zudem umfasst der Umfang der Datenbanksprache üblicherweise auch die Möglichkeit, Abfragen zu formulieren.

Welche Vorteile haben relationale gegenüber hierarchischen Datenbanken?

Repetitionsfragen

Was ist eine Rationale Datenbank?

Eine relationale Datenbank ist eine digitale Datenbank, die zur elektronischen Datenverwaltung in Computersystemen dient und auf einem tabellenbasierten relationalen Datenbankmodell beruht.

Man kann Relationen abbilden.

Nennen Sie drei verschiedene relationale Datenbank Management Systeme DBMS?

Maria DB, My SQL Workbench

Welche Aufgaben erfüllt ein DBMS für Sie als Entwickler?

Handelt es sich bei SQL um eine Programmiersprache?

Nein, es ist offiziell eine Sprache um Daten abzufragen.

Ist die Sprache SQL auf allen Datenbanken gleich?

Die Grundsätzlichen Funktionen bleiben gleich.

Was versteht man unter NoSQL Datenbank?

NoSQL steht für „Not only SQL“ und bezeichnet Datenbanksysteme, die einen nicht-relationalen Ansatz verfolgen. Diese Datenbanken, denen verschiedene Datenbankmodelle zugrunde liegen können, sind horizontal skalierbar und lassen sich für Big-Data-Anwendungen einsetzen.

Man braucht dafür kein SQL

Was versteht man unter Datenbank Schema?

Es ist eine Visuelle Darstellung der Struktur der Datenbank.

Wozu dient ein Entity Relationship Model (ERM)?

Es ist ein Standard.

Wie sieht ein Enhanced Entity Relationship (EER) Model aus?

Erweiterung vom ERM. Wie stehen die Tabellen zu einander = Hauptfokus.

Was bedeuten die Begriffe: Entität, Attribut und Tupel?

Entität = Tabellename / Attribut = Themen in Tabelle / Tubel = Dateneinträge

Was versteht man unter Datenkonsistenz?

Die Korrektheit der Daten.

Was bedeutet Redundanz im Zusammenhang mit einer Datenbank?

Redundanzen sind doppelte Informationen in einer Datenbank bzw. Datenbank-Tabelle.

Was versteht man unter Normalform?

Darunter versteht man

Es gibt drei Stufen.

Keine Redundanzen mehr.

Was versteht man unter Primär- und Fremdschlüssel?

Einen verweis auf den Primärschlüssel.

Wie unterscheiden sich die Datentypen Char und Varchar?

Char benötigt genau den Speicher den man definiert hat und varchar passt sich selbstständig an.

Was ist ein Character Set und wie unterscheiden sich ASCII, LATIN1 und UTF-8?

Wie drei verschiedene Alphabete. Immer UTF-8 benutzen (Hat noch Emojis)

Eine Zeichenkodierung erlaubt die eindeutige Zuordnung von Schriftzeichen und Symbolen innerhalb eines Zeichensatzes.

Wie unterscheiden sich die Datentypen Integer und Float?

Integer = Ganze Zahlen

Float = Nachkommastellen könne angezeigt werden.

Decimal = Man kann die Nachkommazahlen definieren.

Wie unterscheiden sich die Datentypen Timestamp und Datetime?

Timestamp= Die Unixzeit zählt die seit dem 1. Januar 1970 abgelaufene Zeit in Sekunden.

Datetime= Datum und zeit

Erste Schritte mit MariaDB

Befehle

Hilfe anzeigen	mysql> \?
Liste aller Datenbanken anzeigen	mysql> SHOW DATABASES;
Mit einer bestimmten Datenbank verbinden	mysql> USE mysql;
Alle Tabellen anzeigen	mysql> SHOW TABLES;
Beschreibung (Struktur) einer Tabelle anzeigen	mysql> DESCRIBE user; mysql> DESC user;

Was bedeuten die Programmparameter "-u root" und "-p"?

«-p» ist für das Passwort.

Access denied= Eingeben « exe. -u root»

Welche Rechte und Aufgaben hat der Benutzer "root"

Alle Rechte

Warum heisst das Konsolen-Programm von MariaDB mysql.exe?

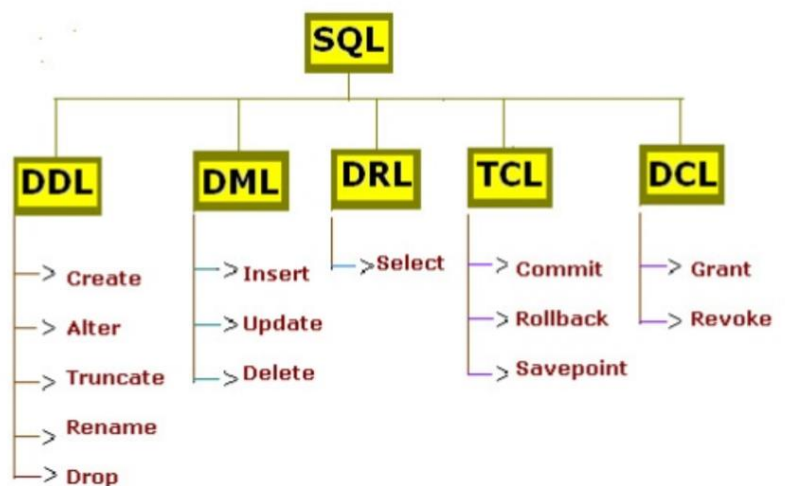
MYSQL gehört Oracle

Der Grund ist weil man etwas einprogrammiert hat.

Database SQL Operation (DDL,SQL,DML,DRL,TCL,DCL)

DATABASE SQL OPERATION

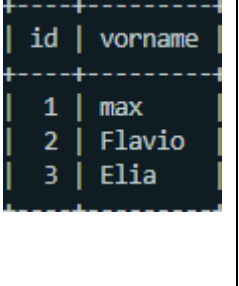
- **DDL:**
Data Definition Language
- **DML:**
Data Manipulation Language
- **DRL:**
Data Retrieval Language
- **TCL:**
Transaction Control Language
- **DCL:**
Data Control Language



Cheat Sheet SQL:

Befehle

Befehl in der Praxis	Beschreibung
CREATE TABLE Tabellen_Namen(ID integer UNSIGNED AUTO_INCREMENT NOT NULL , Name_Attribut Datentyp NOT NULL);	Mit Create Table kann man eine neue Tabelle erstellen . Erstes ist das Beispiel für eine ID. Als zweites gibt man als erstes einen Attribut ein zum Beispiel Name, danach den Datentyp(Varchar usw.)
ALTER TABLE Tabellen_Name ADD(Name_Attribut Datentyp);	Mit Alter Table...ADD kann man ein Attribut hinzufügen . Nach dem ADD gibt man den Namen des Attributes ein, danach die Datentypen
ALTER TABLE Tabellen_Name MODIFY(Name_Attribut Datentyp);	Mit Alter Table...Modify kann man die Datentypen eines Attributes ändern . Zuerst gibt man den Name des Attributes ein, danach die neuen Datentypen
CREATE (Unique) INDEX Index_Name ON Tabellen_Name(Attribut 1, Attribut 2,...);	Mit Create Index...ON... kann man einen Index erstellen . In den Klammern schreibt man die Attribute die in der Tabelle sind. Mit Unique erstellt man einen eindeutigen Index
ALTER TABLE dbo.doc_exb DROP COLUMN column_b;	
DROP INDEX Index_Name ON Tabellen_Name;	Mit DROP INDEX kann man einen INDEX löschen .
ALTER TABLE Tabellen_Name ADD CONSTRAINT Attribut_Name PRIMARY KEY(Attribut_Name);	Mit ALTER TABLE...ADD CONSTRAINT...Primary Key kann man einen Primärschlüssel erstellen .
ALTER TABLE Tabellen_Name ADD CONSTRAINT Beziehungs_Name FOREIGN KEY(Name_Fremdschlüssel) REFERENCES Mastertabellen_Name(Name_ID);	Mit Alter TABLE...ADD CONSTRAINT...FOREIGN KEY kann man eine Fremdschlüssel hinzufügen . Nach References gibt man den Namen der Tabelle ein, in dem der PK ist, dieser kommt in die Klammern danach
ALTER TABLE Tabellen_Name DROP CONTRAIN Beziehungs_Name;	Mit ALTER TABLE...DROP CONSTRAINT kann man eine Beziehung löschen. Mit Tabellen_Name ist die Tabelle mit dem FK gemeint.
ALTER TABLE Mastertabellen_Name DROP CONSTRAINT Name_ID;	Mit ALTER TABLE...DROP CONSTRAINT kann man einen PK entfernen . Geht nur wenn es keinen FK mehr von ihm gibt.
DROP TABLE Tabellen_Name;	Mit DROP TABLE... kann man eine Tabelle löschen
INSERT INTO Tabellen_Name [(Attribut 1, Attribut 2,...)] Values (Attributwert 1, Attributwert 2,...)	Mit INSERT INTO...Values kann man Daten in eine Tabelle einfügen . Zuerst gibt man den Namen der Tabelle ein, danach kann man die Reihenfolge bestimmen, optional. Zuletzt gibt man den Datensatz ein.
INSERT INTO Tabellen_Name[(Attribut 1, PNr., Name, Vorname,...)] SELECT 5, PNr., Name, Vorname From Tabellen_Namen WHERE PNr.=xxx	Mit INSERT INTO...SELECT kann man Angaben aus einer existierenden Tabelle holen , Attribute sind FK. Nach dem SELECT gibt kann man ein Attribut festlegen oder ein Datensatz einer anderen Tabelle holen. Man muss danach den Tabellen Namen eingeben und die Personen Nummer oder ID die man der Person gegeben hat.

<pre>UPDATE Tabellen_Name SET Attribut 1 = Wert 1, Attribut 2 = Wert 2 [WHERE Bedingung für Update] UPDATE kunde SET vorname = 'max' where id = 1;</pre>		<p>Mit UPDATE...SET kann man einen Datensatz ändern. Nach SET gibt man das Attribut ein und danach den Wert den er haben soll. Nach WHERE kommt eine Bedingung, zum Beispiel WHERE Vorname=Cyrill, dann ändert es den Eintrag bei allen die Cyrill zum Vorname heissen.</p>
<pre>ALTER TABLE TableName RENAME COLUMN OldColumnName TO NewColumnName;</pre>		<p>Wenn man den Column Name ändern will.</p>
<pre>INSERT INTO Tabellename [(Attribut1, Attribut2 ...)] VALUES (Attributwert1, Attributwert2, ...);</pre> <p>Beispiel : INSERT INTO Kursleiter (KLNr, Status, Vorname, Name, Kurserfahrung) VALUES (1, 'I', 'Hugo', 'Meier', 3);</p>		<p>Das Einfügen eines neuen Datensatzes in eine Tabelle erfolgt mit dem Befehl „INSERT“</p>
<pre>UPDATE Tabellename SET Attribut1 = Ausdruck1, Attribut2 = Ausdruck2, ... [WHERE Bedingung für Update];</pre> <p>Beispiel: UPDATE Personen SET Lohnstufe = Lohnstufe + 1 WHERE FNr = 4;</p>		<p>Das Nachführen (Update) bestehender Datensätze erfolgt mit der Anweisung „UPDATE“</p>
<pre>DELETE FROM Tabellename [WHERE Bedingung für Delete];</pre> <p>Beispiel: DELETE FROM Kursbesuche WHERE Name= Robin;</p>		<p>Mit diesem Befehl werden alle Datensätze aus der Tabelle „Kursbesuche“ gelöscht, welche den Namen Robin enthalten.</p>
<pre>BEGIN;</pre>		<p>Alle nun nachfolgenden SQL-Anweisungen (INSERT, UPDATE, DELETE) werden protokolliert</p>
<pre>COMMIT;</pre>		<p>Verbucht alle Änderungen am Datenbestand (seit Beginn der Transaktion) in der Datenbank und macht diese Änderungen für andere Benutzer sichtbar.</p>
<pre>ROLLBACK;</pre>		<p>Macht alle Änderungen am Datenbestand, seit Beginn der Transaktion, rückgängig</p>
<pre>ALTER TABLE TableName MODIFY ColumnName Int(10) unsigned null;</pre>		<p>Field 'fk_autos' doesn't have a default value</p>
<pre>CREATE USER backup@'localhost';</pre>		<p>Erzeugen Sie einen neuen lokalen Datenbankbenutzer namens backup</p>
<pre>CREATE USER backup@'localhost' IDENTIFIED BY 'password';</pre>		<p>Benutzer mit Passwort</p>
<pre>REVOKE SELECT ON Table_Name* FROM backup@'localhost';</pre>		<p>Rechte entfernen Benutzer</p>
<pre>DROP USER backup@'localhost';</pre>		<p>Löschen der Benutzer</p>

<p>SELECT * FROM Tabellename;</p>	<p>Damit werden alle Datensätze einer Tabelle mit den entsprechenden Attributnamen aufgelistet.</p>												
<p>SELECT COUNT(*) Anzahl, MIN(Datum) Erster, MAX(Datum) Letzter FROM Kursbesuche;</p> <table border="1" data-bbox="210 353 794 488"> <thead> <tr> <th>Anzahl</th> <th>Erster</th> <th>Letzter</th> </tr> </thead> <tbody> <tr> <td>14</td> <td>07-AUG-08</td> <td>25-AUG-08</td> </tr> </tbody> </table>	Anzahl	Erster	Letzter	14	07-AUG-08	25-AUG-08	<p>Die Funktion „COUNT“ zählt alle Datensätze. Die Funktionen MIN und MAX ermitteln den Minimum- bzw. Maximumwert aus einer Wertefolge.</p>						
Anzahl	Erster	Letzter											
14	07-AUG-08	25-AUG-08											
<p>SELECT PNr, Name, Vorname FROM Personen WHERE FNr = 1;</p> <table border="1" data-bbox="210 604 756 801"> <thead> <tr> <th>PNr</th> <th>Name</th> <th>Vorname</th> </tr> </thead> <tbody> <tr> <td>132442</td> <td>Osswald</td> <td>Kurt</td> </tr> <tr> <td>232452</td> <td>Müller</td> <td>Hugo</td> </tr> <tr> <td>345678</td> <td>Metzger</td> <td>Paul</td> </tr> </tbody> </table>	PNr	Name	Vorname	132442	Osswald	Kurt	232452	Müller	Hugo	345678	Metzger	Paul	<p>Abfragen mit Bedingungen</p> <p>Meistens möchte man nicht alle Datensätze einer Tabelle anzeigen lassen, sondern Auswahl treffen. Dies ist mit dem Schlüsselwort „WHERE“ möglich:</p>
PNr	Name	Vorname											
132442	Osswald	Kurt											
232452	Müller	Hugo											
345678	Metzger	Paul											
<p>SELECT PNr, Name, Vorname FROM Personen WHERE Lohnstufe >= 5 AND (FNr = 2 OR FNr = 3) AND NOT (Name = 'Steffen');</p> <table border="1" data-bbox="210 922 756 1048"> <thead> <tr> <th>PNr</th> <th>Name</th> <th>Vorname</th> </tr> </thead> <tbody> <tr> <td>334643</td> <td>Meier</td> <td>Hans</td> </tr> </tbody> </table>	PNr	Name	Vorname	334643	Meier	Hans	<p>Abfragen mit Bedingungen</p> <p>Statt „NOT(...)“ hätten wir auch Name<>'Steffen' schreiben können.</p>						
PNr	Name	Vorname											
334643	Meier	Hans											
<p>SELECT * FROM Funktionen ORDER BY Funktion;</p> <table border="1" data-bbox="210 1106 762 1370"> <thead> <tr> <th>FNr</th> <th>Funktion</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>Bereichsleiter</td> </tr> <tr> <td>3</td> <td>Chemiker</td> </tr> <tr> <td>5</td> <td>Informatiker</td> </tr> <tr> <td>2</td> <td>Meister</td> </tr> <tr> <td>1</td> <td>Vorarbeiter</td> </tr> </tbody> </table>	FNr	Funktion	4	Bereichsleiter	3	Chemiker	5	Informatiker	2	Meister	1	Vorarbeiter	<p>Datensätze sortieren</p>
FNr	Funktion												
4	Bereichsleiter												
3	Chemiker												
5	Informatiker												
2	Meister												
1	Vorarbeiter												
<p>SELECT * FROM Funktionen ORDER BY FNr DESC;</p> <table border="1" data-bbox="210 1420 756 1684"> <thead> <tr> <th>FNr</th> <th>Funktion</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>Informatiker</td> </tr> <tr> <td>4</td> <td>Bereichsleiter</td> </tr> <tr> <td>3</td> <td>Chemiker</td> </tr> <tr> <td>2</td> <td>Meister</td> </tr> <tr> <td>1</td> <td>Vorarbeiter</td> </tr> </tbody> </table>	FNr	Funktion	5	Informatiker	4	Bereichsleiter	3	Chemiker	2	Meister	1	Vorarbeiter	<p>Datensätze sortieren</p> <p>Das Schlüsselwort „DESC“ (descending) bewirkt, dass absteigend sortiert wird.</p>
FNr	Funktion												
5	Informatiker												
4	Bereichsleiter												
3	Chemiker												
2	Meister												
1	Vorarbeiter												
<p>SELECT PNr, KNr, Datum FROM Kursbesuche ORDER BY PNr ASC, KNr ASC, Datum DESC;</p>	<p>Datensätze sortieren</p> <p>ASC bewirkt aufsteigend</p>												
<p>show create table Table_name;</p>	<p>Fremdschlüssel foreignkey anzeigen</p>												
<p>SELECT column_name(s) FROM table_name WHERE condition GROUP BY column_name(s) HAVING condition ORDER BY column_name(s);</p>	<p>The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.</p>												
<p>SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country HAVING COUNT(CustomerID) > 5;</p>	<p>The following SQL statement lists the number of customers in each country. Only include countries with more than 5 customers:</p>												

Immer gleich aufgebaut wie alter create usw. / Constrain = Überprüft die DB

Berechtigungsliste	
SELECT	lesezugriff
INSERT	neue daten erfassen/erstellen
UPDATE	daten ändern
DELETE	Berechtigungen Datensätze zu löschen
ALTER	tabellen anpassen
DROP	tabellen etc. löschen
DELETE	datensatz löschen
GRANT	berechtigung vergeben

Funktionen

AVG	Mithilfe der AVG() Funktion wird der Durchschnittswert der ersten Spalte ermittelt und ausgegeben.
COUNT	Es werden alle Datensätze gezählt , deren Wert nicht NULL ist .
MAX	Die SQL MAX Funktion ermittelt den höchsten Wert einer Tabellenspalte und liefert ihn zurück.
MIN	Die SQL MIN Funktion ermittelt den niedrigsten Wert einer Tabellenspalte und liefert ihn zurück.
SUM	Die SQL SUM Funktion summiert die Werte einer Tabellenspalte und liefert sie zurück.
UCASE / Uppercase /Upper	Die SQL UCASE Funktion konvertiert den Wert der Tabellenspalte in Großbuchstaben.
LCASE	Die SQL LCASE Funktion konvertiert den Wert der Tabellenspalte in Kleinbuchstaben.
MID	Die SQL MID Funktion extrahiert eine vordefinierte Buchstabenanzahl aus einem Textfeld.
LEN	Die SQL LEN Funktion liefert die Zeichenlänge eines Wertes in einer Tabellenspalte zurück.
ROUND	Die SQL ROUND Funktion rundet einen numerischen Wert mit einer definierten Anzahl von Dezimalstellen auf.
NOW	Die SQL NOW Funktion liefert das aktuelle Datum und die Uhrzeit zurück.
FORMAT	Die SQL FORMAT Funktion formatiert die Tabellenspalte in ein gewünschtes Format.
month	Tag im Monat in der Tabelle ausgeben.
monthname	Monatsname in der Tabelle ausgeben.
year	Jahr in der Tabelle ausgeben.
length	Zählen Sie die Anzahl Buchstaben
substring_index	Datensatz trennen
concat	The CONCAT() function adds two or more strings together.
Datediff	Berechnet die Differenz zwischen zwei Datums-, Zeit- oder Zeitstempelausdrücken

Constrains

- [NOT NULL](#) - Ensures that a column cannot have a NULL value
- [UNIQUE](#) - Ensures that all values in a column are different
- [PRIMARY KEY](#) - A combination of a **NOT NULL** and **UNIQUE**. Uniquely identifies each row in a table
- [FOREIGN KEY](#) - Prevents actions that would destroy links between tables
- [CHECK](#) - Ensures that the values in a column satisfies a specific condition
- [DEFAULT](#) - Sets a default value for a column if no value is specified
- [CREATE INDEX](#) - Used to create and retrieve data from the database very quickly

Block 2: Repetition Schema entwerfen, Tabellen anlegen

- Ich kann ein Datenbank Schema in der 3. Normalform entwerfen
- Ich kenne dies SQL Befehle die zur Data Definition Language (DDL) gehören
- Ich kann Datenbanken und Tabellen mit SQL erzeugen
- Ich kann Datenbanken und Tabellen mit SQL löschen
- Ich kann die Tabellenstruktur mit SQL ändern

Grundkurs Relationale Datenbanken

Tabellen erstellen

Eine neue Tabelle wird mit dem Befehl „CREATE TABLE“ erstellt, welcher gemäß Bild 6.2 aufgebaut ist.

```
CREATE TABLE Tabellename (  
Attributname_1 Datentyp [NOT NULL],  
Attributname_2 Datentyp [NOT NULL],  
Attributname_n Datentyp [NOT NULL]);
```

```
CREATE TABLE Tabellename (  
Attributname_1 Datentyp [NOT NULL],  
Attributname_2 Datentyp [NOT NULL],  
Attributname_n Datentyp [NOT NULL]);
```

Beispiel: CREATE TABLE Personen (
PNr NUMERIC(6) NOT NULL,
Name VARCHAR(20) NOT NULL,
Vorname CHAR(15) NOT NULL,
GebDat DATE,
Grossesse NUMERIC(3,2))

Tabellen ändern

Für den Fall, dass eine bestehende Tabelle noch mit weiteren Attributen ergänzt werden soll oder bestehende Attribute geändert werden müssen, kann dies mit dem Befehl „ALTER TABLE“ erreicht werden, wie dies Bild 6.3 zeigt.

```
ALTER TABLE Tabellename ADD  
( Neuer_Attributname Datentyp [NOT NULL], ... );
```

```
ALTER TABLE Tabellename ADD  
( Neuer_Attributname Datentyp [NOT NULL], ... );
```

Beispiel: ALTER TABLE Personen ADD
(Lohnstufe NUMBER(1) NOT NULL);

Attribute ändern

Bestehende Attribute ändern:

```
ALTER TABLE Tabellename MODIFY  
(Bestehendes_Attribut neuer_Datentyp [NOT NULL], ... );
```

Beispiel: **ALTER TABLE** Personen **MODIFY**
(Lohnstufe **NUMERIC**(2) **NULL**);

Tabellenattribute indizieren

```
CREATE [UNIQUE] INDEX Indexname ON  
Tabellename (Attribut1, Attribut2, ... );
```

Index löschen

```
DROP INDEX Indexname ON Tabellename;
```

Beziehungen erstellen Foreign Key, Primary Key

- a) **ALTER TABLE** Mastertabelle **ADD CONSTRAINT**
Id-Schlüsselname **PRIMARY KEY**(Id-Schlüssel);
- b) **ALTER TABLE** Detailtabelle **ADD CONSTRAINT**
Beziehungsname **FOREIGN KEY**(Fremdschlüssel)
REFERENCES Mastertabelle(Id-Schlüssel);

Beziehungen entfernen

```
ALTER TABLE Detailtabelle  
DROP CONSTRAINT Beziehungsname;
```

Primärschlüssel entfernen

```
ALTER TABLE Mastertabelle  
DROP CONSTRAINT Id-Schlüsselname;
```

Tabellen löschen

```
DROP TABLE Tabellename;
```

Neue Datensätze erzeugen

```
INSERT INTO Tabellename [ ( Attribut1, Attribut2 ... ) ]  
VALUES ( Attributwert1, Attributwert2, ... );
```


Theorie zu Entity Relationship Models ERM

Anomalien

Es gibt drei Anomalien die bei einem unsauber realisierten Datenbank-Entwurf auftauchen können:

- Löschanomalie
- Einfügeanomalie
- Aktualisierungsanomalie /Änderungs-Anomalie

Datentypen

Datentypen

NUMERIC(m,n) : Numerische Felder, wobei m die max. Anzahl aller Ziffern und n die Anzahl der Ziffern nach dem Dezimalpunkt angibt.

CHAR(n): Alphanumerische Felder mit einer fixen Feldlänge von n Zeichen (vielfach begrenzt auf 255 Zeichen). Wenn nicht alle n Zeichen benötigt werden, dann werden die restlichen Stellen mit Leerschlägen gefüllt.

Int: immer bei primary key und wenn man rechnen muss (Ganze Zahlen)

VARCHAR(n): Alphanumerische Felder mit einer variablen Feldlänge bis zu n Zeichen (Standarddatentyp für Zeichenketten).

DATE: Felder für das Datum und die Zeit, wobei das Datumsformat je nach Datenbank anders aussieht / **Boolean**: Ja oder Nein antwort

double: Rechnen mit Kommazahlen

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2 ⁶³	0	2 ⁶³ -1	2 ⁶⁴ -1

```
CREATE TABLE person (
  id INT ✓ ,
  firstname VARCHAR(255) ✓ ,
  lastname VARCHAR(255) ✓ ,
  email VARCHAR(255) ✓ ,
  zip TINYINT(4) ✓ ,
  initials CHAR(2) ✓ ,
  isMarried BOOLEAN ✓ ,
  accountBalance DECIMAL(6,2) ✓ ,
  birthDate DATE ✓ ,
  description TEXT ✓ ,
  gender ENUM {'Male', 'Female', 'Other'}, ✓
  timeOfDeath DATETIME ✓
```

Datentyp	Beschreibung
CHARACTER(n)	Zeichenkette. Fester Länge n
VARCHAR(n) or CHARACTER VARYING(n)	Zeichenkette. Variable Länge. Maximale Länge n
BINARY(n)	Binär-String. Fester Länge n
BOOLEAN	Speichert WAHR oder FALSCH Werte
VARBINARY(n) or BINARY VARYING(n)	Binär-String. Variable Länge. Maximale Länge n
INTEGER(p)	Integer numerische (kein dezimal). Präzisions-p
SMALLINT	Integer numerische (kein dezimal). Precision 5
INTEGER	Integer numerische (kein dezimal). Precision 10
BIGINT	Integer numerische (kein dezimal). Precision 19
DECIMAL(p,s)	Exakte numerische, Präzision p, Skala s. Beispiel: dezimal (5,2) ist eine Zahl, die 3 Stellen vor dem Komma hat und 2 Stellen nach dem Komma
NUMERIC(p,s)	Exakte numerische, Präzision p, Skala s. (Das gleiche wie DECIMAL)
FLOAT(p)	Ungefähre numerische, Mantisse Präzision p. Eine schwimmende Zahl in der Basis 10 Exponentialnotation. Die Größe Argument für diese Art besteht aus einer einzigen Zahl, die die minimale Genauigkeit angibt
REAL	Ungefähre numerische, Mantisse Präzision 7
FLOAT	Ungefähre numerische, Mantisse Präzision 16
DOUBLE PRECISION	Ungefähre numerische, Mantisse Präzision 16
DATE	Speichert Jahr, Monat und Tag Werte
TIME	Speichert Stunde, Minute und Sekunde Werte
TIMESTAMP	Speichert Jahr, Monat, Tag, Stunde, Minute und Sekunde Werte
INTERVAL	Zusammengesetzt aus einer Anzahl von Integer-Felder, eine Zeitperiode darstellt, abhängig von der Art des Intervalls
ARRAY	Ein Set-Länge und geordnete Sammlung von Elementen
MULTISET	Ein variabler Länge und ungeordnete Sammlung von Elementen
XML	Speichert XML-Daten

Dateitypen

Int = Immer Schlüssel Fk und id

Varchar = Namen und Vornamen mit VARCHAR(30)

Decimal = Dezimalzahlen

Datetime = Datum

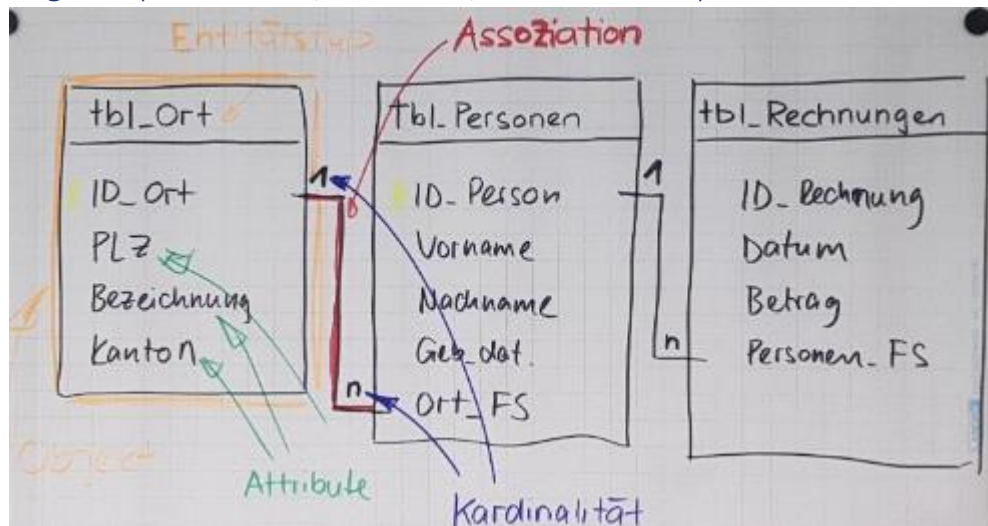
Blob = großes binäres Datenobjekt (Dateien wie Video-, Audio- oder Bilddateien)

NN= Not null (Hier muss etwas drin stehen)

UQ = Unique (Es gibt diesen Benutzer nur einmal) (E-Mail)

AI = Autoincrement fk pk

Begriffe (Assoziation, Attribute, Kardinalitäten)



integrität

Das ist wie eine Absicherung, das alles versichert ist und alles richtig drin ist.

Normalformen

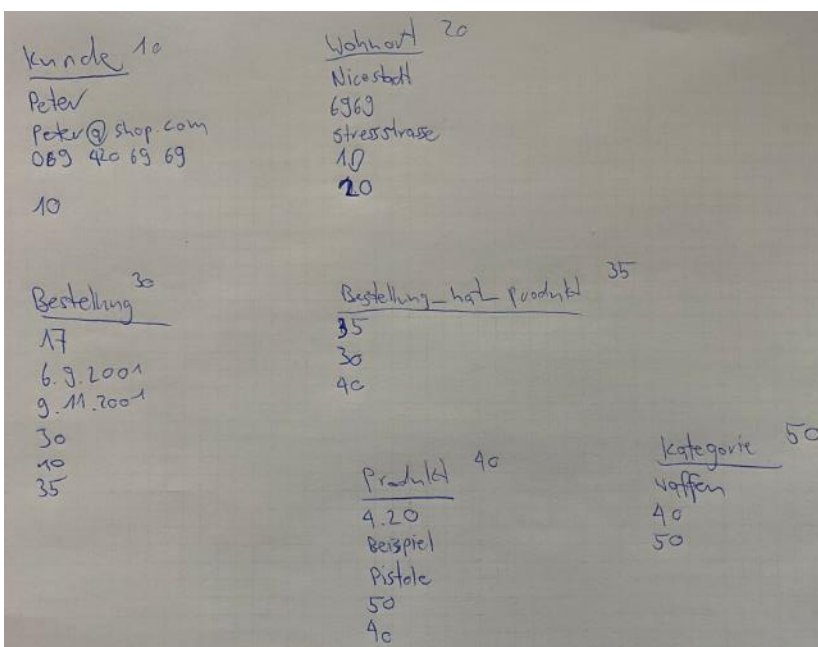
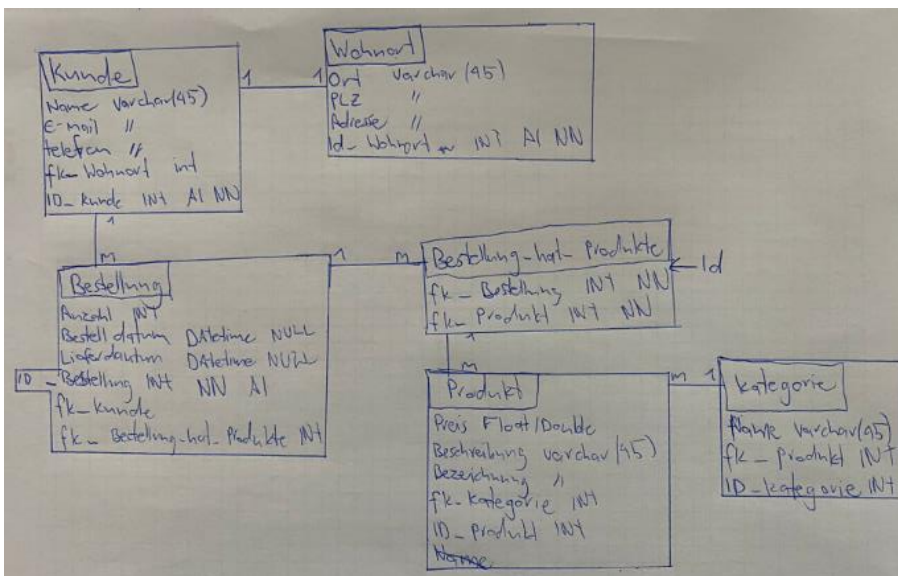
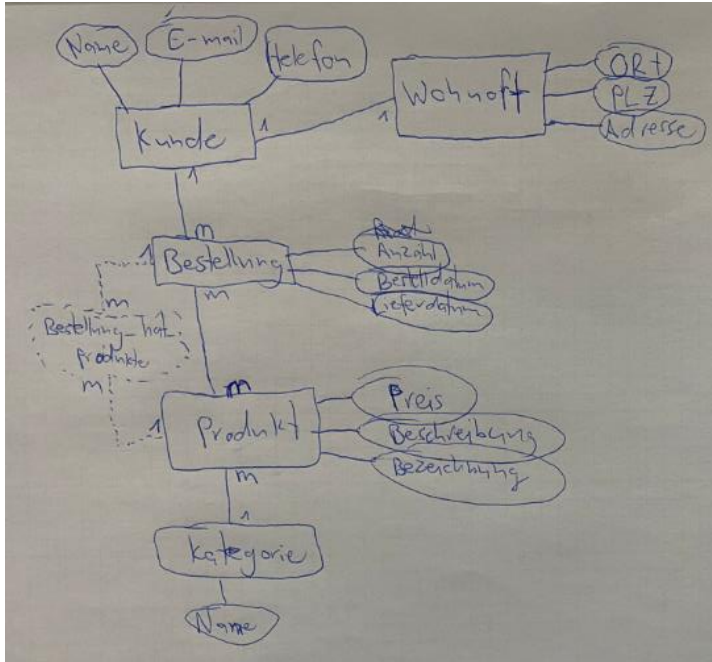
Normalisierung																																																																	
1 Normalform	<p>Alle Attribute weisen nur einfache Attributwerte auf (atomar). Daten sind auf einzelne Attribute aufgeteilt</p> <table border="1"> <thead> <tr> <th>SchülerNr</th> <th>Name</th> <th>Vorname</th> <th>Klasse</th> <th>Klassenlehrer</th> <th>LernangebotsNr</th> <th>Beschreibung</th> <th>Zeit in h</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Jürgens</td> <td>Ina</td> <td>11a</td> <td>Lempel</td> <td>2</td> <td>Tanz</td> <td>12</td> </tr> <tr> <td>2</td> <td>Schmidt</td> <td>Tom</td> <td>12a</td> <td>Breier</td> <td>3</td> <td>Chor</td> <td>22</td> </tr> <tr> <td>3</td> <td>Jäger</td> <td>Franz</td> <td>11a</td> <td>Lempel</td> <td>1, 2, 3</td> <td>Elektronik, Tanz, Chor</td> <td>15, 12, 2</td> </tr> <tr> <td>4</td> <td>Olsen</td> <td>Ina</td> <td>11b</td> <td>Sommer</td> <td>2</td> <td>Tanz</td> <td>5</td> </tr> <tr> <td>5</td> <td>Jürgens</td> <td>Paula</td> <td>12a</td> <td>Breier</td> <td>1</td> <td>Elektronik</td> <td>23</td> </tr> </tbody> </table>	SchülerNr	Name	Vorname	Klasse	Klassenlehrer	LernangebotsNr	Beschreibung	Zeit in h	1	Jürgens	Ina	11a	Lempel	2	Tanz	12	2	Schmidt	Tom	12a	Breier	3	Chor	22	3	Jäger	Franz	11a	Lempel	1, 2, 3	Elektronik, Tanz, Chor	15, 12, 2	4	Olsen	Ina	11b	Sommer	2	Tanz	5	5	Jürgens	Paula	12a	Breier	1	Elektronik	23																
SchülerNr	Name	Vorname	Klasse	Klassenlehrer	LernangebotsNr	Beschreibung	Zeit in h																																																										
1	Jürgens	Ina	11a	Lempel	2	Tanz	12																																																										
2	Schmidt	Tom	12a	Breier	3	Chor	22																																																										
3	Jäger	Franz	11a	Lempel	1, 2, 3	Elektronik, Tanz, Chor	15, 12, 2																																																										
4	Olsen	Ina	11b	Sommer	2	Tanz	5																																																										
5	Jürgens	Paula	12a	Breier	1	Elektronik	23																																																										
2 Normalform	<p>1 Normalform muss erfüllt sein. Attribute sind vom PK abhängig.</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>Relation Lernangebot</p> <table border="1"> <thead> <tr> <th>LernangebotsNr</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Tanz</td> </tr> <tr> <td>3</td> <td>Chor</td> </tr> <tr> <td>1</td> <td>Elektronik</td> </tr> </tbody> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Relation Schüler</p> <table border="1"> <thead> <tr> <th>SchülerNr</th> <th>Name</th> <th>Vorname</th> <th>Klasse</th> <th>Klassenlehrer</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Jürgens</td> <td>Ina</td> <td>11a</td> <td>Lempel</td> </tr> <tr> <td>2</td> <td>Schmidt</td> <td>Tom</td> <td>12a</td> <td>Breier</td> </tr> <tr> <td>3</td> <td>Jäger</td> <td>Franz</td> <td>11a</td> <td>Lempel</td> </tr> <tr> <td>4</td> <td>Olsen</td> <td>Ina</td> <td>11b</td> <td>Sommer</td> </tr> <tr> <td>5</td> <td>Jürgens</td> <td>Paula</td> <td>12a</td> <td>Breier</td> </tr> </tbody> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Relation Lernangebotsübersicht</p> <table border="1"> <thead> <tr> <th>SchülerNr</th> <th>LernangebotsNr</th> <th>Zeit in h</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>12</td> </tr> <tr> <td>2</td> <td>3</td> <td>22</td> </tr> <tr> <td>3</td> <td>1</td> <td>15</td> </tr> <tr> <td>3</td> <td>2</td> <td>12</td> </tr> <tr> <td>3</td> <td>3</td> <td>2</td> </tr> <tr> <td>4</td> <td>2</td> <td>5</td> </tr> <tr> <td>5</td> <td>1</td> <td>23</td> </tr> </tbody> </table> </div> </div>	LernangebotsNr	Beschreibung	2	Tanz	3	Chor	1	Elektronik	SchülerNr	Name	Vorname	Klasse	Klassenlehrer	1	Jürgens	Ina	11a	Lempel	2	Schmidt	Tom	12a	Breier	3	Jäger	Franz	11a	Lempel	4	Olsen	Ina	11b	Sommer	5	Jürgens	Paula	12a	Breier	SchülerNr	LernangebotsNr	Zeit in h	1	2	12	2	3	22	3	1	15	3	2	12	3	3	2	4	2	5	5	1	23		
LernangebotsNr	Beschreibung																																																																
2	Tanz																																																																
3	Chor																																																																
1	Elektronik																																																																
SchülerNr	Name	Vorname	Klasse	Klassenlehrer																																																													
1	Jürgens	Ina	11a	Lempel																																																													
2	Schmidt	Tom	12a	Breier																																																													
3	Jäger	Franz	11a	Lempel																																																													
4	Olsen	Ina	11b	Sommer																																																													
5	Jürgens	Paula	12a	Breier																																																													
SchülerNr	LernangebotsNr	Zeit in h																																																															
1	2	12																																																															
2	3	22																																																															
3	1	15																																																															
3	2	12																																																															
3	3	2																																																															
4	2	5																																																															
5	1	23																																																															
3 Normalform	<p>2 Normalform muss erfüllt sein. Attribute sind nur vom Primärschlüssel abhängig.</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>Relation Lernangebot</p> <table border="1"> <thead> <tr> <th>LernangebotsNr</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Tanz</td> </tr> <tr> <td>3</td> <td>Chor</td> </tr> <tr> <td>1</td> <td>Elektronik</td> </tr> </tbody> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Relation Schüler</p> <table border="1"> <thead> <tr> <th>SchülerNr</th> <th>Name</th> <th>Vorname</th> <th>Klasse</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Jürgens</td> <td>Ina</td> <td>11a</td> </tr> <tr> <td>2</td> <td>Schmidt</td> <td>Tom</td> <td>12a</td> </tr> <tr> <td>3</td> <td>Jäger</td> <td>Franz</td> <td>11a</td> </tr> <tr> <td>4</td> <td>Olsen</td> <td>Ina</td> <td>11b</td> </tr> <tr> <td>5</td> <td>Jürgens</td> <td>Paula</td> <td>12a</td> </tr> </tbody> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Relation Lernangebotsübersicht</p> <table border="1"> <thead> <tr> <th>SchülerNr</th> <th>LernangebotsNr</th> <th>Zeit in h</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>12</td> </tr> <tr> <td>2</td> <td>3</td> <td>22</td> </tr> <tr> <td>3</td> <td>1</td> <td>15</td> </tr> <tr> <td>3</td> <td>2</td> <td>12</td> </tr> <tr> <td>3</td> <td>3</td> <td>2</td> </tr> <tr> <td>4</td> <td>2</td> <td>5</td> </tr> <tr> <td>5</td> <td>1</td> <td>23</td> </tr> </tbody> </table> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Relation Klasse</p> <table border="1"> <thead> <tr> <th>Klasse</th> <th>Klassenlehrer</th> </tr> </thead> <tbody> <tr> <td>11a</td> <td>Lempel</td> </tr> <tr> <td>12a</td> <td>Breier</td> </tr> <tr> <td>11b</td> <td>Sommer</td> </tr> </tbody> </table> </div> </div>	LernangebotsNr	Beschreibung	2	Tanz	3	Chor	1	Elektronik	SchülerNr	Name	Vorname	Klasse	1	Jürgens	Ina	11a	2	Schmidt	Tom	12a	3	Jäger	Franz	11a	4	Olsen	Ina	11b	5	Jürgens	Paula	12a	SchülerNr	LernangebotsNr	Zeit in h	1	2	12	2	3	22	3	1	15	3	2	12	3	3	2	4	2	5	5	1	23	Klasse	Klassenlehrer	11a	Lempel	12a	Breier	11b	Sommer
LernangebotsNr	Beschreibung																																																																
2	Tanz																																																																
3	Chor																																																																
1	Elektronik																																																																
SchülerNr	Name	Vorname	Klasse																																																														
1	Jürgens	Ina	11a																																																														
2	Schmidt	Tom	12a																																																														
3	Jäger	Franz	11a																																																														
4	Olsen	Ina	11b																																																														
5	Jürgens	Paula	12a																																																														
SchülerNr	LernangebotsNr	Zeit in h																																																															
1	2	12																																																															
2	3	22																																																															
3	1	15																																																															
3	2	12																																																															
3	3	2																																																															
4	2	5																																																															
5	1	23																																																															
Klasse	Klassenlehrer																																																																
11a	Lempel																																																																
12a	Breier																																																																
11b	Sommer																																																																

Workbench (Not Null)

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idtable1	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
table1col	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

PK= Primarykey

ERM erstellt



SQL Create Befehl ERM

Beispiel

```
CREATE TABLE kunde (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
Vorname varchar(255) NOT NULL, -- es ist egal ob not null klein oder gross ist
Nachname varchar (255) NOT NULL,
Adresse varchar (255) Not null,
Email varchar (255),
Telefonnummer int not null,
fk_ort int unsigned not null,
foreign key(fk_ort_id) references ort(id)
);
```

Wenn Fehlermeldung: ERROR 1364 (HY000): Field 'fk_beispiel' doesn't have a default value

```
➔ ALTER TABLE produkt MODIFY fk_beispiel Int(10) unsigned null;
```

Joins

SQL JOIN Grundlagen

Die Welt der SQL JOINS

LEFT JOIN

SELECT <Auswahl>
FROM TabelleA A
LEFT JOIN TabelleB B
ON A.Schlüssel = B.Schlüssel

RIGHT JOIN

SELECT <Auswahl>
FROM TabelleA A
RIGHT JOIN TabelleB B
ON A.Schlüssel = B.Schlüssel

INNER JOIN

SELECT <Auswahl>
FROM TabelleA A
INNER JOIN TabelleB B
ON A.Schlüssel = B.Schlüssel

RIGHT JOIN (Filtered)

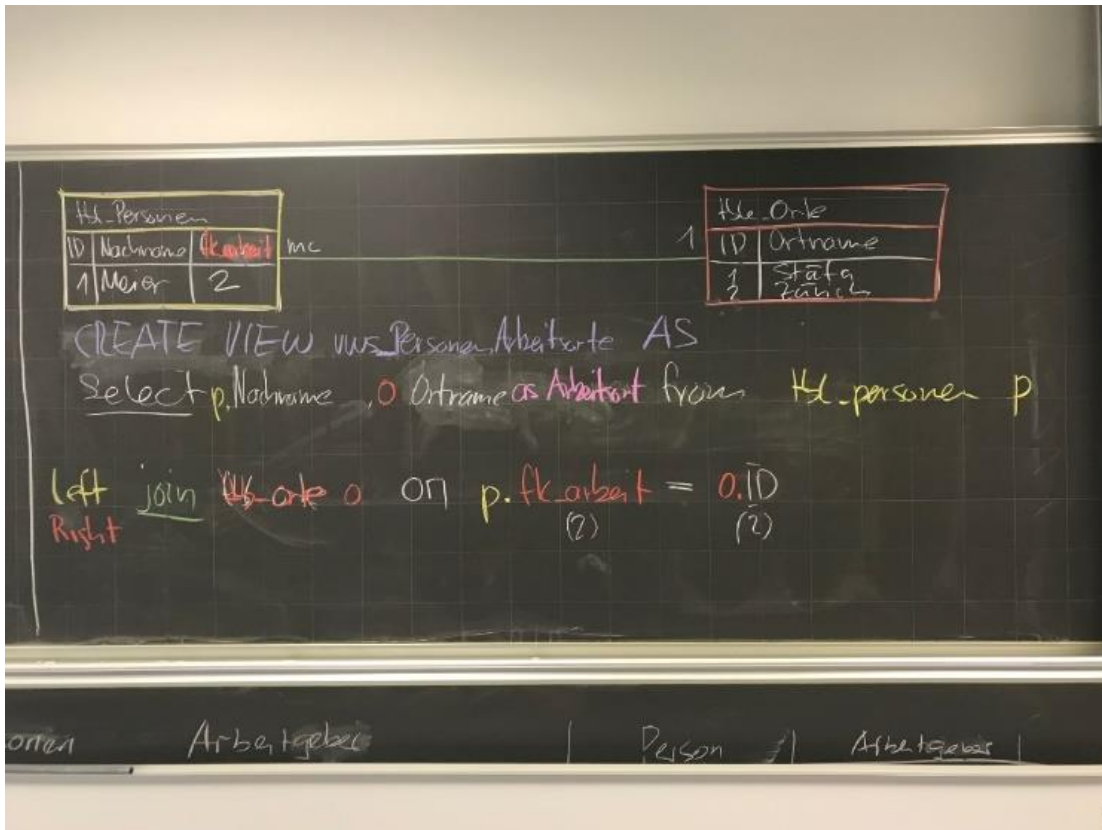
SELECT <Auswahl>
FROM TabelleA A
RIGHT JOIN TabelleB B
ON A.Schlüssel = B.Schlüssel
WHERE A.Schlüssel IS NULL

FULL OUTER JOIN

SELECT <Auswahl>
FROM TabelleA A
FULL OUTER JOIN TabelleB B
ON A.Schlüssel = B.Schlüssel

FULL OUTER JOIN (Filtered)

SELECT <Auswahl>
FROM TabelleA A
FULL OUTER JOIN Tabelle B
ON A.Schlüssel = B.Schlüssel
WHERE A.Schlüssel IS NULL
OR B.Schlüssel IS NULL



Fragen

UNSIGNED = , --unsigned heisst, dass die ID bei 0 startet

Default = DEFAULT-Einschränkung wird verwendet, um den Standardwert der Spalte einzufügen.

Null?

- Tabellenattribute indizieren (Index erstellen)
- Export database
- Befehl um etwas zu importieren.
- Unterscheid User Localhost und %
- W3 Schools alle durchspielen
- Interval

SQRT() = Quadratwurzel

Distinct = Nur ein Daten Satz ausgeben.

Serial= bigint unsigned not null auto_increment unique

Bigint = mehr zahlen als bei int

Interval = Anderen wert als Tag zuweisen

Foreignkey is incorrectly formed = muss gleicher typ sein wie der primarykey und unsigned

MOD(Spalte, 2)=1,0 = ungrad

=0 grad

SQL dump importieren. zusammendurchspielen

-- Vorname und Nachname aller Schauspieler des Films "HARRY IDAHO"

```
select first_name, last_name
```

```
from actor
```

```
Join film_actor on film_actor.actor_id=actor.actor_id
```

```
join film on film_actor.film_id=film.film_id
```

```
Where film.title = "HARRY IDAHO"
```

```
order by film.title;
```