



SoSe 2019

Einführung in Relationale Datenbanken

Prof. Dr.-Ing. Uwe Rüppel
Meiling Shi, M.Sc.

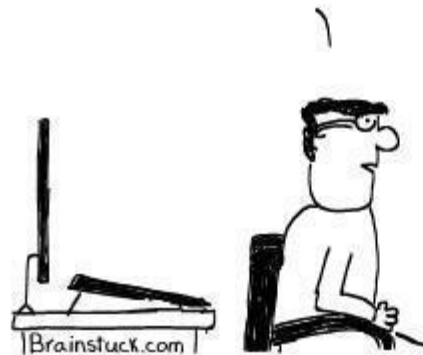
Einführung in MySQL



TECHNISCHE
UNIVERSITÄT
DARMSTADT



We won't be able
to deliver our product
in time because of
some issue with MySQL...



WHAT???

THEN USE
SOMEBODY ELSE'S
SQL, BUT I
WANT THE
PRODUCT
IN TIME.

- Relationales Datenbankmanagementsystem
- 1994 von MySQL AB entwickelt
 - 2008 Sun Microsystems
 - 2012 Oracle
- freie OpenSource-Version: „**MySQL Community Server**“
<http://dev.mysql.com/downloads/mysql/>
 - Installation über MySQL Installer MSI (empfehlenswert) **ODER** ZIP Archive
- Dokumentation:
<http://dev.mysql.com/doc/>

Benutzung von MySQL in Console(1)

- MySQL Community Server installieren (aktuell: V 5.7.21)
- Ggf. **Path**-Variable setzen → [MySQL Server Verzeichnis]\bin
- Starten von MySQL → MySQL Command Line Client oder Konsole

- Ggf. anmelden:

```
mysql -u <name> -p
```

- Datenbanken anzeigen:

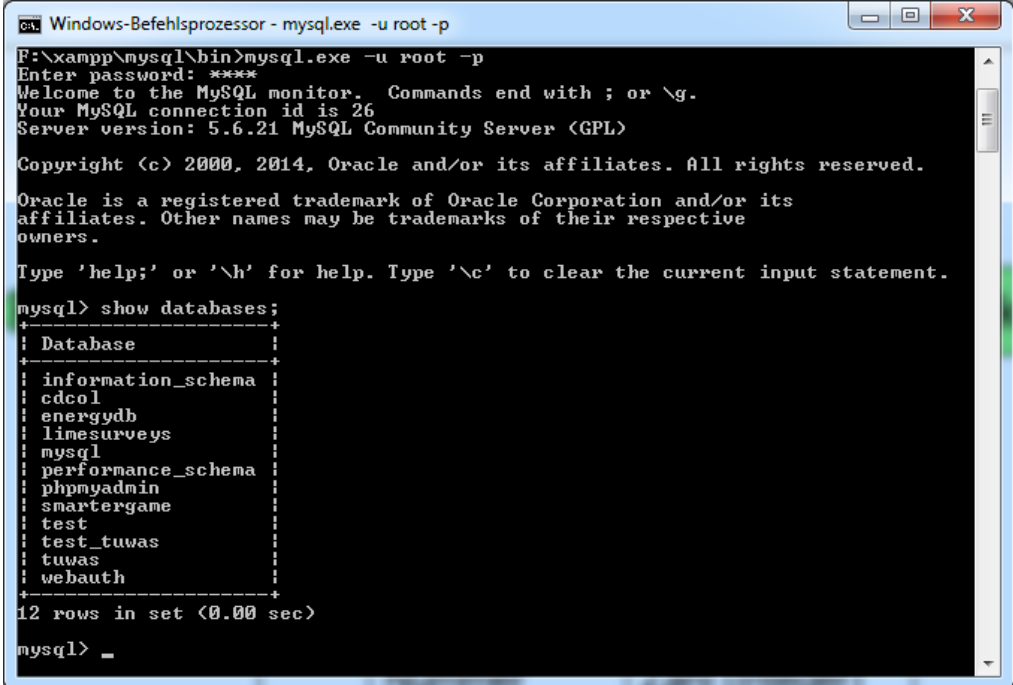
```
show databases;
```

- DB erstellen / verwenden:

```
create database <xyz>;
```

```
use <xyz>;
```

- Beenden: `exit;`



```
ca. Windows-Befehlsprozessor - mysql.exe -u root -p
F:\xampp\mysql\bin>mysql.exe -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 5.6.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol       |
| energydb    |
| limesurveys  |
| mysql       |
| performance_schema |
| phpmyadmin   |
| snartergame  |
| test        |
| test_tuwas   |
| tuwas       |
| webauth     |
+-----+
12 rows in set (0.00 sec)

mysql> _
```

Benutzung von MySQL in Console(2)

- Befehle zur Verwaltung von Tabellen
 - `show tables;`
 - `show columns from <table_name>;`
- Standard SQL
 - z.B. `SELECT`

```
mysql> use iib2_auftrag_kunde;
Database changed
mysql> select * from kunde;
+----+-----+-----+
| id | name  | vorname |
+----+-----+-----+
| 1  | peter | franz   |
+----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

```
Windows-Befehlsprozessor - mysql.exe -u root -p

mysql> show tables;
+-----+
| Tables_in_iib2_auftrag_kunde |
+-----+
| auftrag                       |
| kunde                         |
+-----+
2 rows in set (0.00 sec)

mysql> show columns from auftrag;
+----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| beschreibung | varchar(255)  | YES  |     | NULL    |                |
| erledigt   | tinyint(1)    | YES  |     | NULL    |                |
| kunde_id   | int(11)       | YES  | MUL | NULL    |                |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> show columns from kunde;
+----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| name       | varchar(100)  | YES  |     | NULL    |                |
| vorname    | varchar(100)  | YES  |     | NULL    |                |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> _
```

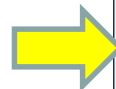
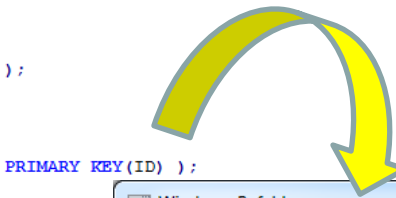
- Mehrere SQL-Befehle können in einer Textdatei (*.sql) an MySQL übergeben werden

```
create.sql
1 CREATE DATABASE IF NOT EXISTS Projektverwaltung;
2
3 USE Projektverwaltung;
4
5 CREATE TABLE Unternehmen(
6     ID INT NOT NULL AUTO_INCREMENT,
7     Name VARCHAR(100), Ort VARCHAR(100), PRIMARY KEY(ID) );
8
9 CREATE TABLE Projekt(
10     ID INT NOT NULL AUTO_INCREMENT,
11     Nummer INT, Name VARCHAR(100), Kosten DECIMAL(14,2), PRIMARY KEY(ID) );
12
13 CREATE TABLE beteiligt_an(
14     ID INT NOT NULL AUTO_INCREMENT,
15     U_ID INT, P_ID INT, Prozent DECIMAL(5,2),
16     PRIMARY KEY(ID),
17     FOREIGN KEY(U_ID) REFERENCES Unternehmen(ID),
18     FOREIGN KEY(P_ID) REFERENCES Projekt(ID) );
```

source datei.sql

oder

\. datei.sql



```
Windows-Befehlsprozessor - mysql -u root -p

F:\xampp\mysql\bin>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.6.21 MySQL Community Server <GPL>

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source E:\Lehre\IIB\IIB2\SoSe2015\Hoersaaluebungen\1\create.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.04 sec)

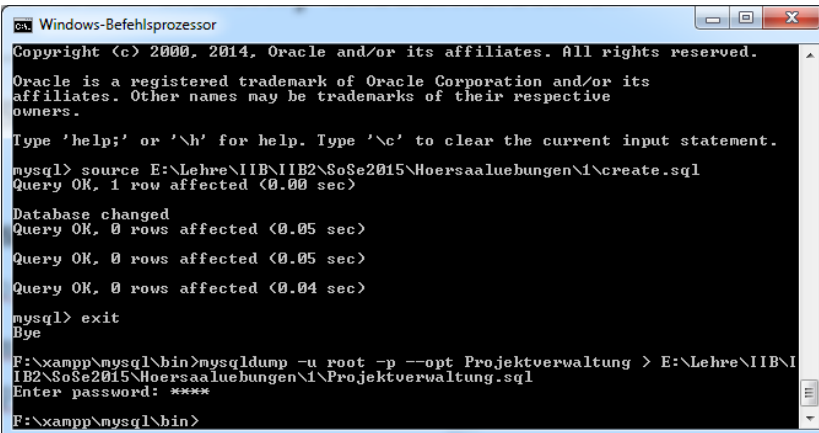
mysql>
```

- Sichern der Datenbank in einer Datei mit Hilfe von `mysqldump.exe`

Beispiel: Datenbank Projektverwaltung sichern in Datei

Projektverwaltung.sql

- `mysqldump -u <username> -p --opt Projektverwaltung > Projektverwaltung.sql`



```
Windows-Befehlsprozessor
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source E:\Lehre\IIB\IIB2\SoSe2015\Hoersaaluebungen\1\create.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.05 sec)

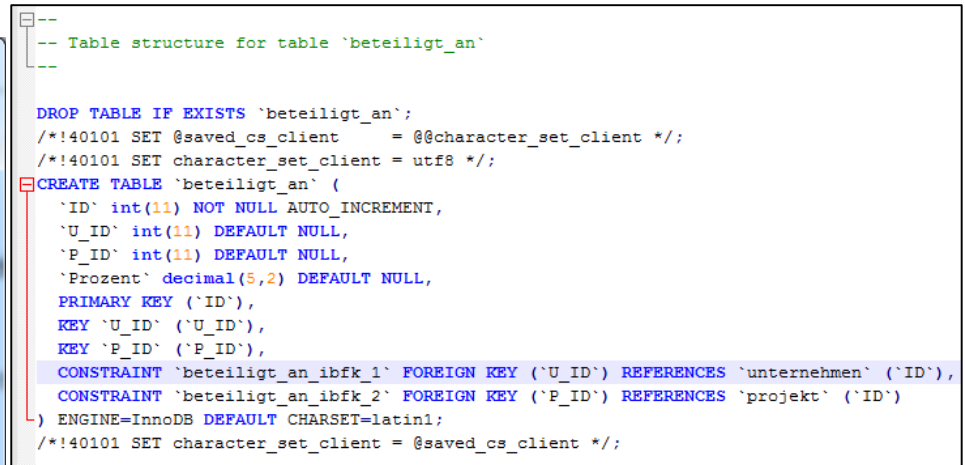
Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.04 sec)

mysql> exit
Bye

F:\xampp\mysql\bin> mysqldump -u root -p --opt Projektverwaltung > E:\Lehre\IIB\IIB2\SoSe2015\Hoersaaluebungen\1\Projektverwaltung.sql
Enter password: ****

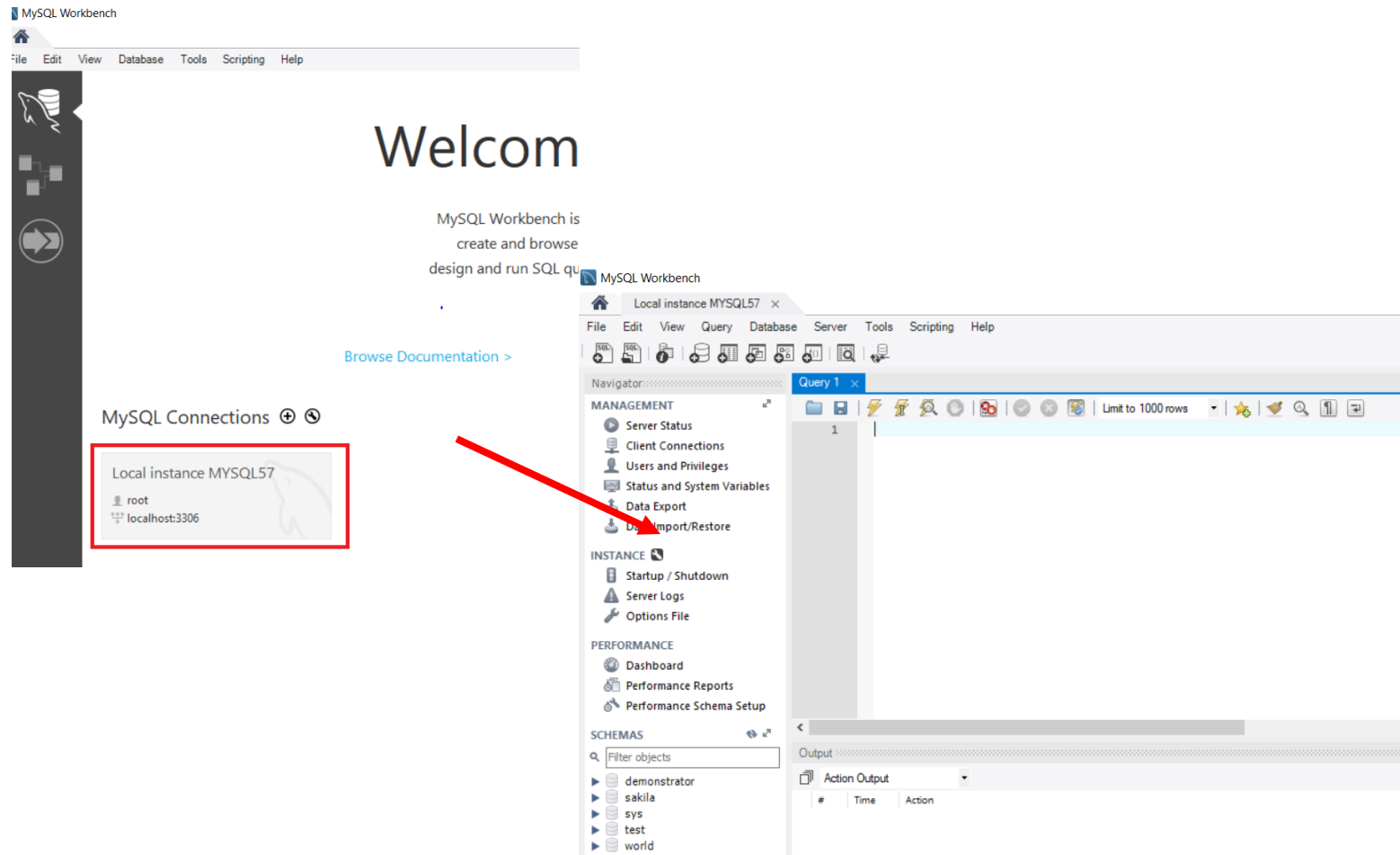
F:\xampp\mysql\bin>
```



```
--
-- Table structure for table `beteiligt_an`
--

DROP TABLE IF EXISTS `beteiligt_an`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `beteiligt_an` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `U_ID` int(11) DEFAULT NULL,
  `P_ID` int(11) DEFAULT NULL,
  `Prozent` decimal(5,2) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `U_ID` (`U_ID`),
  KEY `P_ID` (`P_ID`),
  CONSTRAINT `beteiligt_an_ibfk_1` FOREIGN KEY (`U_ID`) REFERENCES `unternehmen` (`ID`),
  CONSTRAINT `beteiligt_an_ibfk_2` FOREIGN KEY (`P_ID`) REFERENCES `projekt` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
```

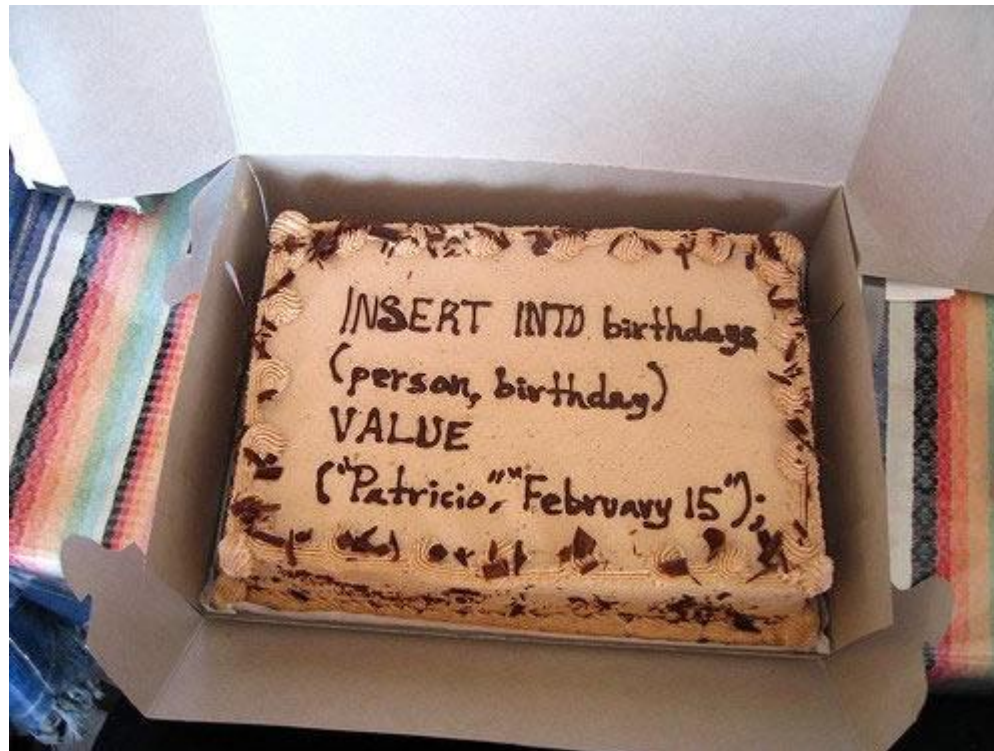
Benutzung von MySQL in *MySQL Workbench* (zu empfehlen)



Erzeugen der Datenbank mittels SQL



TECHNISCHE
UNIVERSITÄT
DARMSTADT



(My)SQL – Datentypen

Type {storage}	Name	Range	Default
Numeric {1 byte}	TINYINT[(M)]	-128 TO 127 [0 to 255 if UNSIGNED]	NULL [0 if NOT NULL]
Numeric {2 bytes}	SMALLINT[(M)]	-32,768 to 32,767 [0 to 65,535]	NULL [0 if NOT NULL]
Numeric {3 bytes}	MEDIUMINT[(M)]	-8,388,608 to 8,388,607 [0 to 16,777,215]	NULL [0 if NOT NULL]
Numeric {4 bytes}	INT[(M)]	-/+2.147E+9 [0 to 4.294E+9]	NULL [0 if NOT NULL]
Numeric {8 bytes}	BIGINT[(M)]	-/+9.223E+18 [0 to 18.45E+18]	NULL [0 if NOT NULL]
Numeric {4 or 8}	FLOAT(p)	p=0-24 --> "FLOAT" p=25-53 --> "DOUBLE"	NULL [0 if NOT NULL]
Numeric {4 bytes}	FLOAT[(M,D)]	Min=+/-1.175E-38 Max=+/-3.403E+38	NULL [0 if NOT NULL]
Numeric {8 bytes}	DOUBLE[(M,D)]	Min=+/-2.225E-308 Max=+/-1.798E+308	NULL [0 if NOT NULL]
Numeric {M+2}	DECIMAL[(M,[D])] Stored as string	Max Range = DOUBLE range Fixed point vs. DOUBLE float	NULL [0 if NOT NULL]
Bit {8 bytes}	BIT[(M)]	Binary. Display by [add zero or converting with BIN()]. M=1-64	NULL [0 if NOT NULL]

SQL – Datentypen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Type {storage}	Name	Range	Default
String {M char's}	CHAR[(M)]	M=0-255 Characters, FIXED. Right padded with spaces.	NULL ["" if NOT NULL]
String {M char's ¹ }	VARCHAR(M)	M=0-65,535 Characters M=0-255 <v5.0.3	NULL ["" if NOT NULL]
String {#char's ¹ }	TINYTEXT ²	0-255 Characters	NULL ["" if NOT NULL]
String {#char's ¹ }	TEXT ²	0-65,535 Char's	NULL ["" if NOT NULL]
String {#char's ¹ }	MEDIUMTEXT ²	0-16,777,215 Char's	NULL ["" if NOT NULL]
String {#char's ¹ }	LONGTEXT ²	0-4,294,967,295 Char's	NULL ["" if NOT NULL]
String {M bytes}	BINARY[(M)]	M=0-255 bytes, FIXED.	NULL ["" if NOT NULL]
String {M bytes}	VARBINARY(M)	0-65,535 bytes M=0-255 <v5.0.3	NULL ["" if NOT NULL]
String {#bytes ¹ }	TINYBLOB	0-255 bytes	NULL ["" if NOT NULL]
String {#bytes ¹ }	BLOB	0-65,535 bytes	NULL ["" if NOT NULL]
String {#bytes ¹ }	MEDIUMBLOB	0-16,777,215 bytes	NULL ["" if NOT NULL]
String {#bytes ¹ }	LOBLOB	0-4,294,967,295 bytes	NULL ["" if NOT NULL]
String {1-2 bytes}	ENUM ² ("A1","A2",...)	Column is exactly 1 of 1-65,535 values	NULL [1st value if NOT NULL]
String {1-8 bytes}	SET ² ("A1","A2",...)	Column is 0 or more values in list of 1- 64 members	NULL ["" if NOT NULL]

SQL – Datentypen

Type {storage}	Name	Range	Default
Date & Time {3 bytes}	DATE	"1000-01-01" - "9999-12-31"	NULL ["0000-00-00" if NOT NULL]
Date & Time {8 bytes}	DATETIME	"1000-01-01 00:00:00" - "9999-12-31 23:59:59"	NULL ["0000-00-00 00:00:00" if NOT NULL]
Date & Time {3 bytes}	TIME	"-838:59:59" - "838:59:59"	NULL ["00:00:00" if NOT NULL]
Date & Time {4 bytes}	TIMESTAMP	19700101000000 - 2037+	Current Date & Time
Date & Time {1 bytes}	YEAR	1900 - 2155	NULL ["0000" if NOT NULL]

¹ Storage will be # of characters or bytes, plus byte(s) to record length.

² These String data types are NOT case sensitive, unless given the "binary" attribute or have a case-sensitive CHARACTER SET collation

Vorgehensweise beim Erstellen einer DB

1. Datenbank anlegen (Container für Tabellen und andere Datenbankobjekte)



2. Erstellen der Tabellen (Spalten und Datentypen)

Gutachter	
PK	<u>gar_id : Int</u>
	gar_name : String
	gar_vorname : String
	gar_username : String
	gar_passwort : Hash

3. Füllen der Tabellen mit Daten

Gutachter

ID	Name	Vorname
1	Schmidt	Karl
2	Böhler	Maria
3	Haas	Frieda

4. Auswertung der Daten (Abfragen) und Löschen / Ändern der Daten



Befehle zum Erstellen & Bearbeiten von Databases

- Datenbank erstellen

```
CREATE DATABASE [IF NOT EXIST] datenbankname;
```

- Datenbank auswählen

```
USE datenbankname;
```

- Existierende Datenbanken anzeigen

```
SHOW DATABASES;
```

- Datenbank löschen

```
DROP DATABASE [IF EXISTS] datenbankname;
```

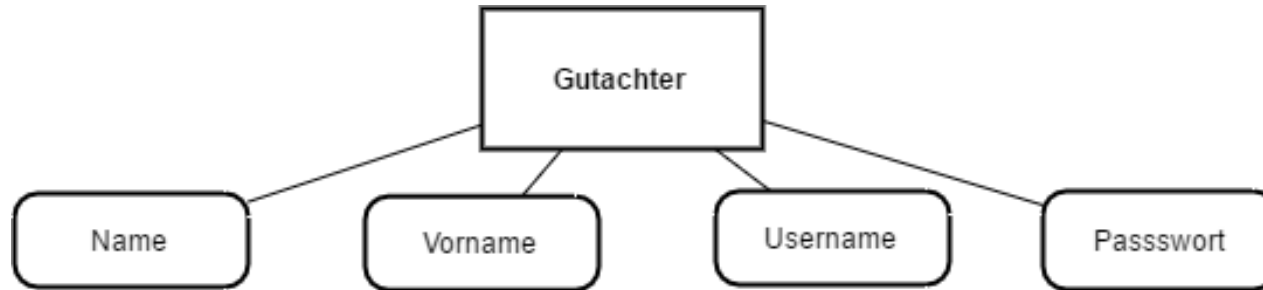
→ CREATE TABLE

Syntax

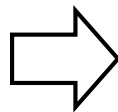
```
CREATE TABLE tabellenname1 (  
    PKName1 INT NOT NULL AUTO_INCREMENT,  
    datenfeld_1 datentyp_1, ...  
    datenfeld_X datentyp_X,  
    PKName2 INT,  
    PRIMARY KEY (PKName1),  
    FOREIGN KEY (FKName)  
    REFERENCES tabellenname2(PKName2)  
);
```

UNIQUE → Primärschlüssel
NOT NULL → Fremdschlüssel
→ Verweist auf den Primärschlüssel von tabellenname2

Erstellen von Tabellen – Beispiel ohne Relation



Gutachter	
PK	<u>gar_id : Int</u>
	gar_name : String
	gar_vorname : String
	gar_username : String
	gar_passwort : Hash



```
CREATE TABLE Gutachter (  
    gar_id INT NOT NULL AUTO_INCREMENT,  
    gar_name VARCHAR(50),  
    gar_vorname VARCHAR(50),  
    gar_username VARCHAR(50),  
    gar_passwort BLOB,  
    PRIMARY KEY (gar_id)  
);
```


Erstellen von Tabellen – Beispiel 1 : n



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Gebäude	
PK	<u>geb_id: Int</u>
	geb_ort : String
	geb_plz : Int
	geb_strasse : String
	geb_hausnummer : Int
	geb_guid : String
FK	geb_blt_id : Int

1

n

Stockwerk	
PK	<u>stw_id : Int</u>
	stw_bezeichnung : String
	stw_guid : String
FK	stw_geb_id : Int

```
CREATE TABLE Gebaeude (
    geb_id INT NOT NULL AUTO_INCREMENT,
    geb_ort VARCHAR(50), geb_plz INT(6),
    geb_strasse VARCHAR(50), geb_hausnummer VARCHAR(4),
    geb_guid VARCHAR(50),
    PRIMARY KEY(geb_id) );
```

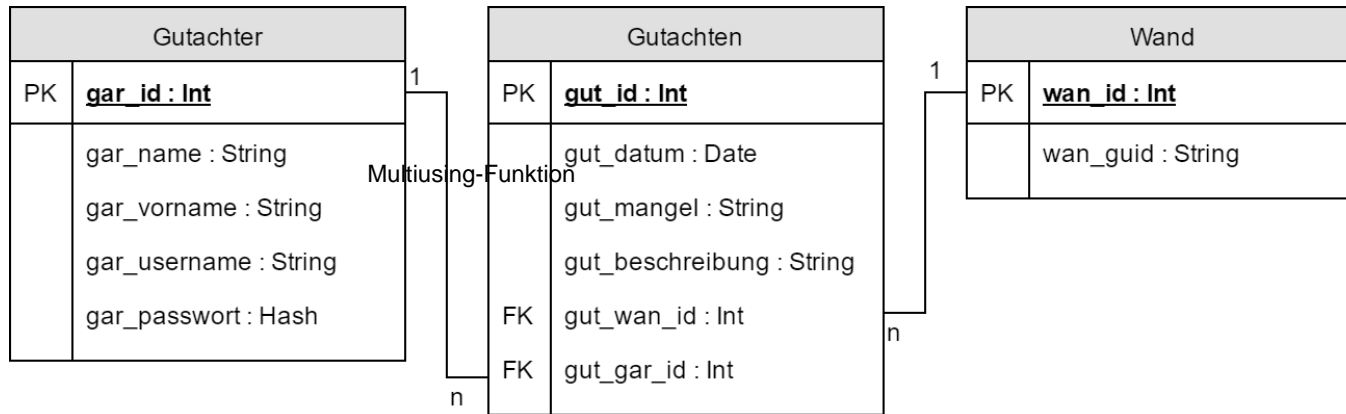
← Tabelle Gebäude erstellen

```
CREATE TABLE Stockwerk (
    stw_id INT NOT NULL AUTO_INCREMENT,
    stw_bezeichnung VARCHAR(50), stw_guid VARCHAR(50),
    stw_geb_id INT,
    PRIMARY KEY (stw_id),
    FOREIGN KEY (stw_geb_id) REFERENCES Gebaeude(geb_id) );
```

← Tabelle Stockwerk erstellen

Fremdschlüssel festlegen

Erstellen von Tabellen – Beispiel m : n



```
CREATE TABLE Gutachter (
    gar_id INT NOT NULL AUTO_INCREMENT,
    gar_name VARCHAR(50), gar_vorname
    VARCHAR(50), gar_username VARCHAR(50), gar_passwort BLOB, PRIMARY KEY (gar_id) );

CREATE TABLE Wand (
    wan_id INT NOT NULL AUTO_INCREMENT, wan_guid VARCHAR(50), PRIMARY KEY (wan_id) );

CREATE TABLE Gutachten (
    gut_id INT NOT NULL AUTO_INCREMENT, gut_datum DATE, gut_beschreibung VARCHAR(50),
    gut_mangel VARCHAR(50), gut_wan_id INT, gut_gar_id INT,
    PRIMARY KEY (gut_id),
    FOREIGN KEY (gut_wan_id) REFERENCES Wand(wan_id),
    FOREIGN KEY (gut_gar_id) REFERENCES Gutachter(gar_id) );
```

Fremdschlüsselparameter

```
CREATE TABLE tabellenname1 (  
    PKName 1 INT NOT NULL AUTO_INCREMENT, FKName INT,  
    datenfeld_1 datentyp_1, ... , FOREIGN KEY (FKName)  
    REFERENCES tabellenname2(PKName2)  
    [ON UPDATE referenzoption] [ON DELETE referenzoption]  
);
```

Referenzoption	Erklärung
CASCADE	Bei Löschen/Aktualisieren des Datensatzes werden auch die Werte in der referenzierenden Tabelle gelöscht/aktualisiert
SET DEFAULT	Alle referenzierten Datenfelder werden auf den Default-Wert gesetzt
SET NULL	Alle referenzierten Datenfelder werden auf NULL gesetzt

Fremdschlüsselparameter - Beispiel



```
CREATE TABLE Gutachten (  
  gut_id INT NOT NULL AUTO_INCREMENT, gut_datum DATE, gut_beschreibung VARCHAR(50),  
  gut_mangel VARCHAR(50), gut_wan_id INT, gut_gar_id INT, PRIMARY KEY (gut_id),  
  FOREIGN KEY (gut_wan_id) REFERENCES Wand(wan_id) ON DELETE CASCADE,  
  FOREIGN KEY (gut_gar_id) REFERENCES Gutachter(gar_id) ON DELETE SET NULL );
```

```
DELETE FROM Wand WHERE wan_id = 2
```

```
DELETE FROM Gutachter WHERE gar_id = 3
```

Wand	
wan_id	...
1	...
2	...
3	...

Gutachten			
gut_id	gut_wan_id	gut_gar_id	...
1	3	3	...
2	1	2	...
3	2	1	...
4	1	3	...
5	2	2	...

Gutachter	
gar_id	...
1	...
2	...
3	...

→ INSERT INTO

Syntax

```
INSERT INTO tabellenname  
  (datenfeld_1, datenfeld_2, ..., datenfeld_X)  
VALUES  
  (wert_1, wert_2, ..., wert_X),    Datensatz 1  
  ...,  
  (wert_1, wert_2, ..., wert_X)    Datensatz n  
;
```

**Es müssen nicht
alle Datenfelder
einer Tabelle mit
Werten belegt
werden (Ausnahme:
NOT NULL)**

Datensätze eingeben - Beispiel

Gutachter	
PK	<u>gar_id : Int</u>
	gar_name : String gar_vorname : String gar_username : String gar_passwort : Hash



```
INSERT INTO Gutachter (gar_name, gar_vorname,  
gar_username, gar_passwort)  
VALUES('Schmidt', 'Karl', 'karl_schmidt',  
'0dc3c11cd09782f5404591efc3631994065a774d'),  
( 'Böhler', 'Maria', 'm_boe',  
'98dcb2717ddae152d5b359c6ea97e4fe34a29d4c'),  
( 'Kaninchen', 'Frieda', 'fkan',  
'eb2719da85b5c8625c5a85709d0d0deb096828e5');
```



Gutachter				
gar_id	gar_name	gar_vorname	gar_username	gar_passwort
1	Schmidt	Karl	karl_schmidt	0dc3c11cd09782f5404591efc3631994065a774d
2	Böhler	Maria	m_boe	98dcb2717ddae152d5b359c6ea97e4fe34a29d4c
3	Kaninchen	Frieda	fkan	eb2719da85b5c8625c5a85709d0d0deb096828e5

→ UPDATE

Syntax

```
UPDATE tabellenname  
SET datenfeld_1 = wert_1 ,..., datenfeld_X = wert_X,  
[WHERE bedingung];
```

Beispiel

```
UPDATE Gutachter  
SET gar_username = 'kaninchen_f'  
WHERE gar_name = 'Kaninchen' AND gar_vorname = 'Frieda';
```

→ DELETE FROM

Syntax

```
DELETE FROM tabellenname  
[WHERE bedingung];
```

Ist keine Bedingung angegeben, so wird der gesamte Inhalt der Tabelle gelöscht!

Beispiel

```
DELETE FROM Gutachter  
WHERE gar_name = 'Schmidt' AND gar_vorname = 'Karl';
```


Beispiel für VIEW und GRANT

```
CREATE VIEW Gutachten_Boehler (boe_mangel, boe_wandId) AS
    SELECT gut_mangel, wan_guid
        FROM Gutachter, Gutachten, Wand
        WHERE gar_username = 'm_boe' AND gar_id = gut_gar_id
            AND wan_id = gut_wan_id
    WITH CHECK OPTION
```

-- Ständige Überwachung; Änderung: TABLE = VIEW

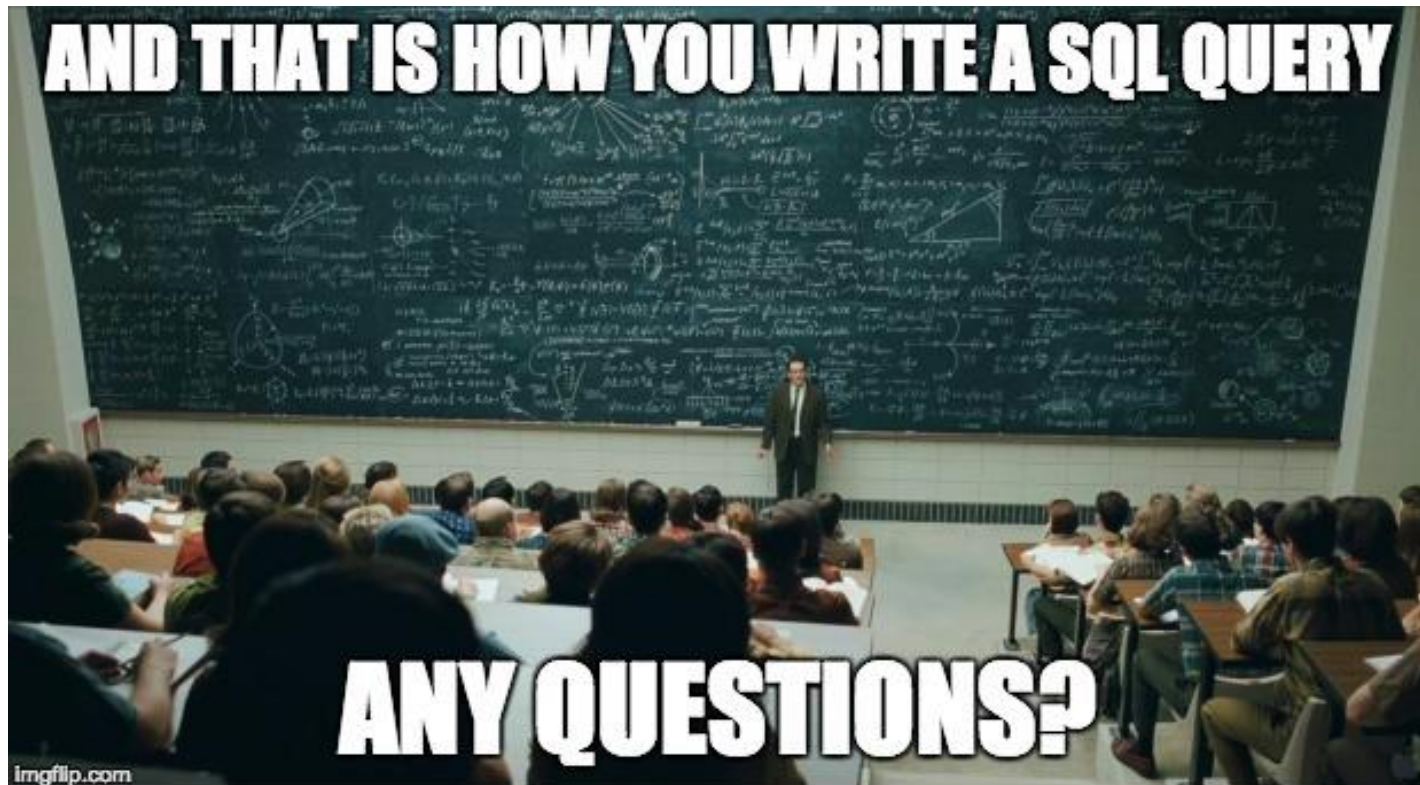
```
GRANT SELECT, UPDATE (boe_mangel) ON Gutachten_Boehler
    TO m_boe_MySQLUser@localhost
    WITH GRANT OPTION -- Weitergabe der Rechte
```

-- Nur Frau Böhler darf die von ihr begutachteten Mängel ändern.

Abfragen von Daten mittels SQL-Queries



TECHNISCHE
UNIVERSITÄT
DARMSTADT



→ SELECT

Syntax

```
SELECT [DISTINCT] feldX,...|* FROM tabellenname  
[WHERE bedingung] [ORDER BY feldX [ASC|DESC]];
```

Beispiel

```
SELECT * FROM Gutachten WHERE gut_gar_id = 3;
```

Beispiel

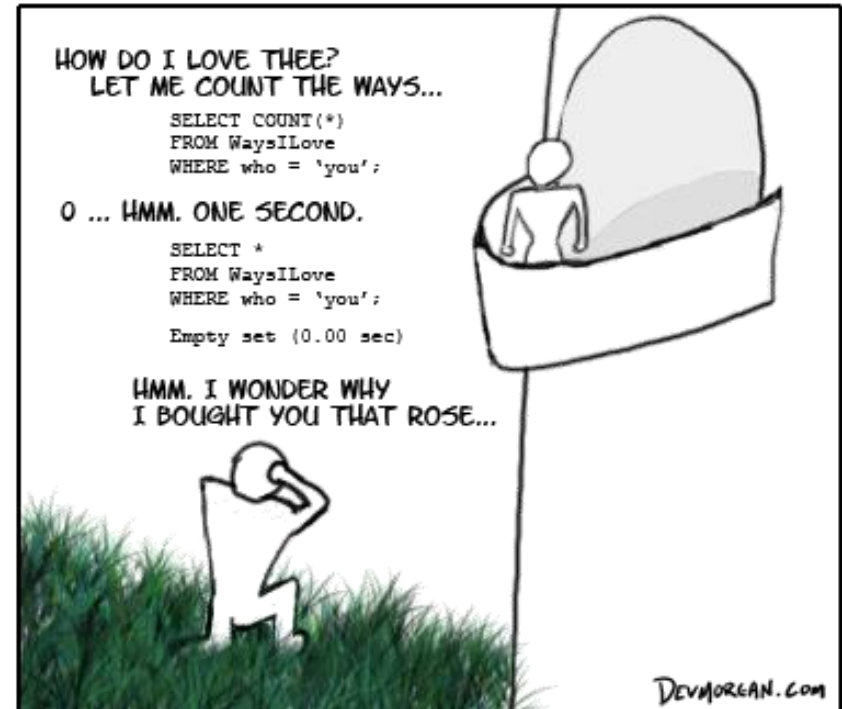
```
SELECT * FROM Gutachter ORDER BY gar_name ASC;
```

- Zusammenfassung von Ergebnissen
- Gruppierungen und einfache Rechenoperationen durchführen
- Beispiele:
 - `SUM()` : Summe
 - `AVG()` : Mittelwert
 - `COUNT()` : Anzahl der Tupel
 - `MIN()` / `MAX()` : Minimum, Maximum
 - `ROUND(zahl, stellen)` : Rundet die übergebene Zahl auf die angegebene Anzahl an Stellen

Aggregatfunktionen - Beispiel

Anzahl aller von allen Gutachtern begutachteten Wände

```
SELECT gar_name, gar_vorname,  
COUNT(wan_id) AS Anzahl_Bauteile  
FROM Gutachter AS GA,  
    Gutachten AS G, Wand AS W  
WHERE GA.gar_id = G.gut_gar_id  
AND W.wan_id = G.gut_wan_id  
GROUP BY GA.gar_username  
ORDER BY Anzahl_Bauteile DESC;
```



Alle Gutachter finden, deren Nachnamen mit „B“ beginnen.

```
SELECT * FROM Gutachter  
WHERE gar_name LIKE ('B%');
```

Alle Gutachter finden, deren Nachnamen ein „e“ beinhalten

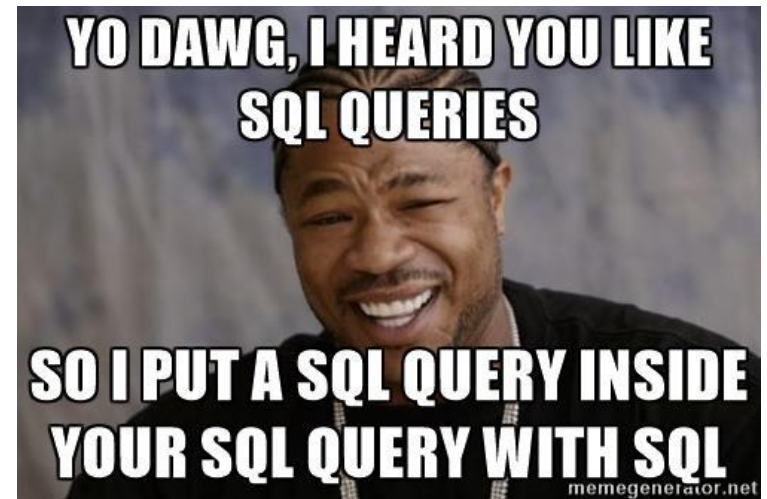
```
SELECT * FROM Gutachter WHERE gar_name LIKE ('%e%');
```

Alle Gutachter finden, deren Nachnamen mit „dt“ enden

```
SELECT * FROM Gutachter WHERE gar_name LIKE ('%dt');
```

Verschachtelte Abfragen

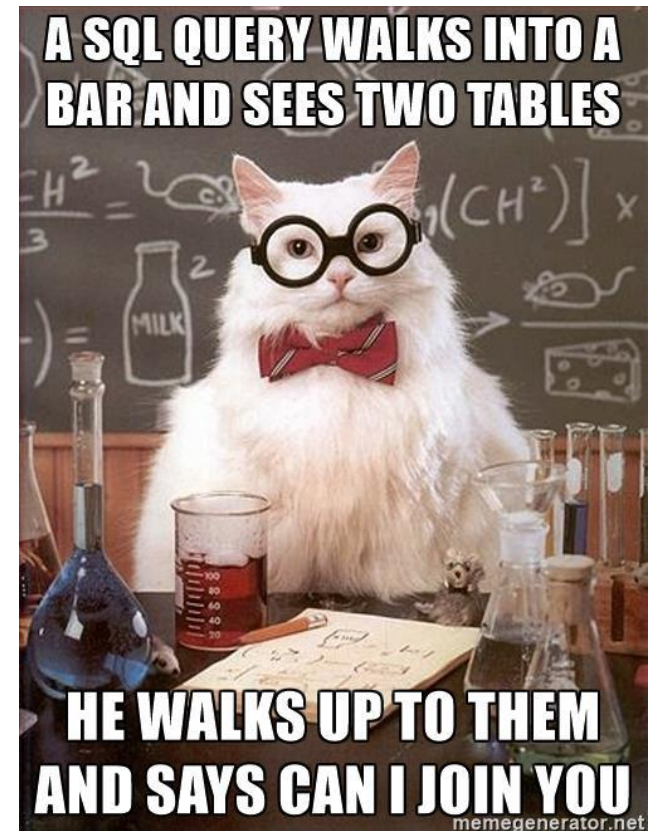
Beispiel: Alle Bauleiter finden, die an Hochhäusern (Häuser, die ein **Stockwerk mit der Bezeichnung ,OG_8‘** besitzen) in **Frankfurt** arbeiten und noch **keinen Usernamen (null)** besitzen.



```
SELECT * FROM Bauleiter
WHERE blt_id IN(
    SELECT DISTINCT geb_blt_id FROM Gebaeude, Stockwerk
    WHERE geb_ort = 'Frankfurt' AND geb_id = stw_geb_id AND
    stw_bezeichnung LIKE ('%8')
) AND blt_username IS NULL ;
```


Joins

- Ein `JOIN` führt Tabellen in Abhängigkeit bestimmter Kriterien zusammen.
- Weiterhin können Regeln für die Ergebnismenge festgelegt werden.
- Typische Joins:
 - `[INNER] JOIN`
 - `LEFT [OUTER] JOIN`
 - `FULL [OUTER] JOIN`

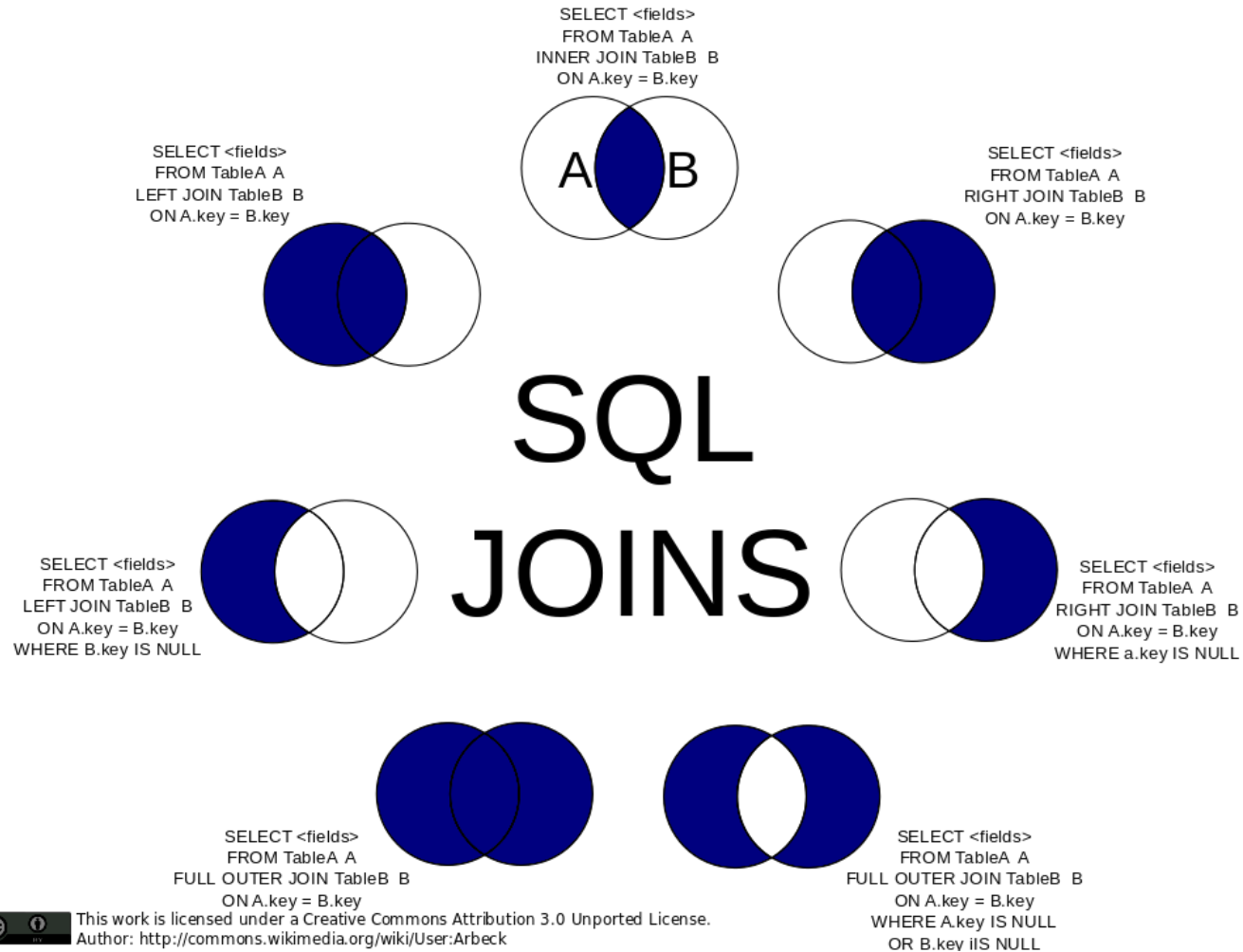


http://wiki.selfhtml.org/wiki/Datenbanken/Einf%C3%BChrung_in_Joins

Joins - Übersicht



TECHNISCHE
UNIVERSITÄT
DARMSTADT



This work is licensed under a Creative Commons Attribution 3.0 Unported License.
Author: <http://commons.wikimedia.org/wiki/User:Arbeck>

Joins - Beispiel

Beispiel: Alle Bauleiter finden, die an Hochhäusern (**Häuser mit mind. 10 Stockwerken**) in **Frankfurt** arbeiten und noch **keinen Usernamen (null)** besitzen.

```
SELECT blt_name, blt_vorname FROM Bauleiter
JOIN (
    SELECT DISTINCT geb_blt_id, COUNT(stw_id) AS CountStockwerke
    FROM Gebaeude, Stockwerk
    WHERE geb_ort = ,Frankfurt'
    AND geb_id = stw_geb_id
    GROUP BY geb_blt_id
) AS link on blt_id = link.geb_blt_id
WHERE link.CountStockwerke > 9 AND blt_username IS NULL;
```

Beispiel: Alle Bauleiter finden, die an Hochhäusern (Häuser mit mind. 10 Stockwerken) in Frankfurt arbeiten und noch keinen Usernamen (null) besitzen.

```
SELECT blt_name, blt_vorname, COUNT(stw_id) AS CountStockwerke
FROM Bauleiter, Gebaeude, Stockwerk
WHERE blt_username IS NULL AND blt_id = geb_blt_id
AND geb_ort = 'Frankfurt'
AND geb_id = stw_geb_id
GROUP BY geb_id
HAVING CountStockwerke > 9 ;
```

Indizierung von Spalten (1)

- Indizes entsprechen „Zeigern“ auf bestimmte Spalten in Tabellen
- Indizes werden in Form von B-Trees (sortierte Baumstruktur) in gesonderten Dateien verwaltet (MySQL → *.myi)
- Dies erlaubt das schnellere Auffinden von Werten in den indizierten Spalten (denn es wird nicht mehr die gesamte Tabelle durchsucht)
- Geschwindigkeitszuwächse vor allem bei Spalten die häufig in `WHERE` oder `ORDER BY` – Klauseln vorkommen
- Beschleunigt `SELECT`, aber verlangsamt `UPDATE`, `DELETE` und `INSERT` Anweisungen (da selbige auch immer in der Index-Datei vorgenommen werden müssen)
- Eher geeignet für Tabellen mit großer Anzahl an Zeilen

Indizierung von Spalten - Beispiel

Tabelle mit 150 Mio. Datensätzen

```
SELECT AVG(Value) FROM metering WHERE  
Timestamp = 1262304000 GROUP BY  
Timestamp;
```

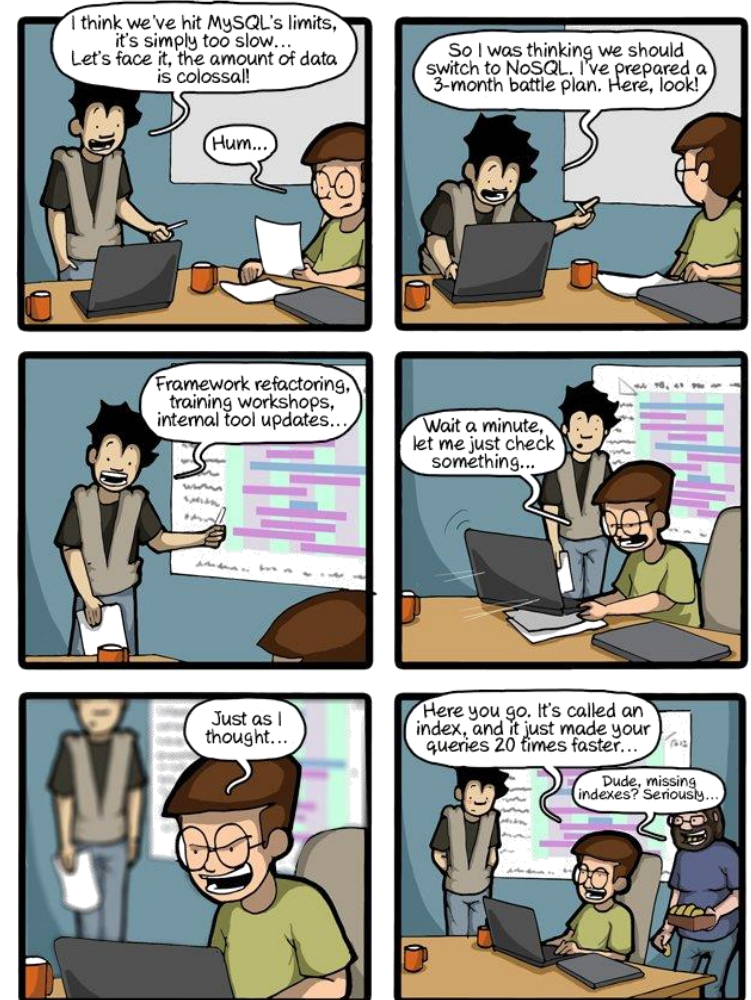
→ Dauer: **1056.111 sec**

Index für Spalte ‚Timestamp‘ erstellen:

```
CREATE INDEX timestamp_index ON  
metering(Timestamp);
```

```
SELECT AVG(Value) FROM metering WHERE  
Timestamp = 1262304000 GROUP BY  
Timestamp;
```

→ Dauer: **0.031 sec**



CommitStrip.com