

# Übungen zu Datenbanken I

## Datenbankentwurf

WS 2019 / 2020

# MySQL – Datenbank erzeugen und löschen

**CREATE DATABASE** [IF NOT EXISTS] database\_name;

**USE** database\_name;

**DROP DATABASE** database\_name;

Bemerkung: **USE** database\_name ist nur einmalig notwendig.

# MySQL – Tabellen erzeugen und löschen

```
CREATE TABLE [IF NOT EXISTS] table_name (  
    attribute1 data_typ [NOT NULL],  
    ...,  
    attributeN data_typ [NOT NULL],  
    PRIMARY KEY (attribute1, ..., attributen),  
    [FOREIGN KEY (attributei)  
        REFERENCES foreign_table (attributej)]  
);
```

```
DROP TABLE table_name;
```

# MySQL – Datentypen

Datentyp	Bedeutung	Beispiel
INTEGER( <i>n</i> )	Ganze Zahl mit maximal <i>n</i> Zeichen	4711
DOUBLE	Gleitkommazahl	47.11
BOOLEAN	Wahrheitswert	TRUE oder FALSE
DATE	Daum	„2019-03-25“
TIME	Zeit	„09:00:00“
VARCHAR( <i>n</i> )	Text mit maximaler Länge <i>n</i>	„Hallo “
ENUM(value <sub>1</sub> , ..., value <sub><i>n</i></sub> )	Aufzählung	„Herr“, „Frau“

# MySQL – Daten einfügen und ändern

```
INSERT INTO    table_name (column_name1, ..., column_namen)  
VALUES        (1, ..., n),  
                (N + 1, ..., 2n),  
                ...
```

```
UPDATE    table_name  
SET       column_name = 'xxx'  
WHERE     column_name = 'zzz'
```

Bemerkung: Zeichenketten gehören stets in Anführungszeichen.

# MySQL – Tabelle ändern

## **ALTER TABLE** table\_name

- **ADD COLUMN** column\_name column\_definition
- **ADD PRIMARY KEY** (column\_name, ...)
- **ADD FOREIGN KEY** (column\_name, ...)  
**REFERENCES** foreign\_table (column\_name, ...)
- **DROP COLUMN** column\_name
- **DROP PRIMARY KEY**
- **DROP FOREIGN KEY** fk\_symbol

Bemerkung: fk\_symbol: Im Navigator Tabelle auswählen und unter Foreign Keys nachlesen.

# MySQL – Fremdschlüssel

## CREATE TABLE

- Enthält eine Relation einen Fremdschlüssel, muss auch die referenzierte Relation existieren.
- Regeln:
  - 1 Erst Entity-Typen, dann Beziehungstypen
  - 2 Erst starke Entity-Typen, dann schwache Entity-Typen
  - 3 Erst vererbende Entity-Typen, dann erbende Entity-Typen

## DROP TABLE

- Eine referenzierte Relation kann erst gelöscht werden, wenn alle sie referenzierende Relationen gelöscht wurden.
- Regeln:
  - 1 Erst Beziehungstypen, dann Entity-Typen
  - 2 Erst schwache Entity-Typen, dann starke Entity-Typen
  - 3 Erst erbende Entity-Typen, dann vererbende Entity-Typen

# MySQL – Anfragen

```
SELECT [DISTINCT] (* | attribute1, ..., attributeN)  
FROM table_reference  
[WHERE where_condition]  
  
[GROUP BY attribute]  
[ORDER BY attribute [ASC | DESC]]
```



# Vom Relationenmodell zur Datenbank

Studierende (Vorname, Nachname, Geburtstag, Matrikelnummer, SemesterAnzahl)

$\mathcal{K}_{\text{Studierende}} = \{\{\text{Vorname}, \text{Nachname}\}, \{\text{Matrikelnummer}\}\}$

Studierende (Vorname, Nachname)  
→ Person (Vorname, Nachname)

```
CREATE TABLE Studierende (  
    Vorname VARCHAR(50) NOT NULL,  
    Nachname VARCHAR(50) NOT NULL,  
    Geburtstag DATE,  
    Matrikelnummer INTEGER(15) NOT NULL,  
    Semesteranzahl INTEGER(2),  
    PRIMARY KEY (Vorname, Nachname),  
    UNIQUE (Matrikelnummer),  
    FOREIGN KEY (Vorname, Nachname)  
        REFERENCES Person (Vorname, Nachname));
```

# Aufgabe

Gegeben seien die Relationen:

- HOTEL(Hotelname, Kategorie, Typ, Telefon, Beschreibung)
- ZIMMERTYP(Hotelname, ZTyp, Anzahl, Foto, Preis, Beschreibung)
- ZIMMER(Hotelname, ZTyp, Zimmernummer, WLAN)
- GAST(Vorname, Nachname, Rabattstufe).

Erstellen Sie die Relationen

- RECHNUNG(Rechnungsnummer, Datum, Summe, Zahlungsart, bezahlt)
- BELEGT(Rechnungsnummer, von, bis, Hotelname, ZTyp, Zimmernummer, Vorname, Nachname).

Füllen Sie diese anschließend mit ein paar Datensätzen.

⇒ Geben Sie die jeweils zugehörigen SQL-Anweisungen an!

# Lösung I

```
CREATE TABLE Rechnung (  
    Rechnungsnummer INT(8) NOT NULL,  
    Datum DATE,  
    Summe DOUBLE,  
    Zahlungsart VARCHAR(10),  
    bezahlt BOOLEAN,  
    PRIMARY KEY (Rechnungsnummer)  
);  
  
INSERT INTO Rechnung (Rechnungsnummer, Datum, Summe,  
    Zahlungsart, bezahlt)  
VALUES (12345678, 2018-12-01, 23.34, 'Kreditkarte', TRUE),  
    (23812847, 2018-12-23, 233.04, 'Rechnung', FALSE);
```

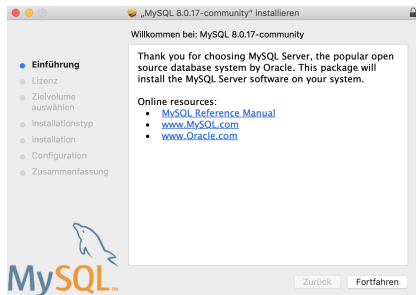
# Lösung II

```
CREATE TABLE Belegt (  
    Rechnungsnummer INT(8) NOT NULL,  
    von DATE,  
    bis DATE,  
    Hotelname VARCHAR(15),  
    ZTyp VARCHAR(10),  
    Zimmernummer INT(3),  
    Vorname VARCHAR(15),  
    Nachname VARCHAR(15),  
    PRIMARY KEY (Rechnungsnummer),  
    FOREIGN KEY (Rechnungsnummer)  
        REFERENCES Rechnung (Rechnungsnummer),  
    FOREIGN KEY (Hotelname, ZTyp, Zimmernummer)  
        REFERENCES Zimmer (Hotelname, ZTyp, Zimmernummer),  
    FOREIGN KEY (Vorname, Nachname)  
        REFERENCES Gast (Vorname, Nachname)  
);
```

# Voraussetzungen für den Rechnerübungsanteil

- Installation auf Windows  
(<https://dev.mysql.com/downloads/windows/installer/8.0.html>)
- Installation auf Ubuntu  
(<https://dev.mysql.com/downloads/repo/apt/>)
- Installation auf Mac  
(<https://dev.mysql.com/downloads/mysql/>)
- Installation von MySQL Workbench (sofern notwendig)  
(<https://www.mysql.com/de/products/workbench/>)
- stabile Internetverbindung

# MySQL Community Server installieren (Mac)



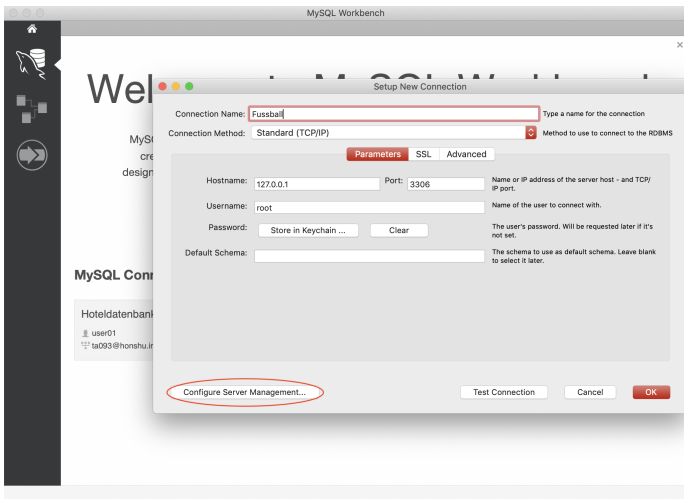
## Tutorial:

<https://dev.mysql.com/doc/refman/8.0/en/installing.html>

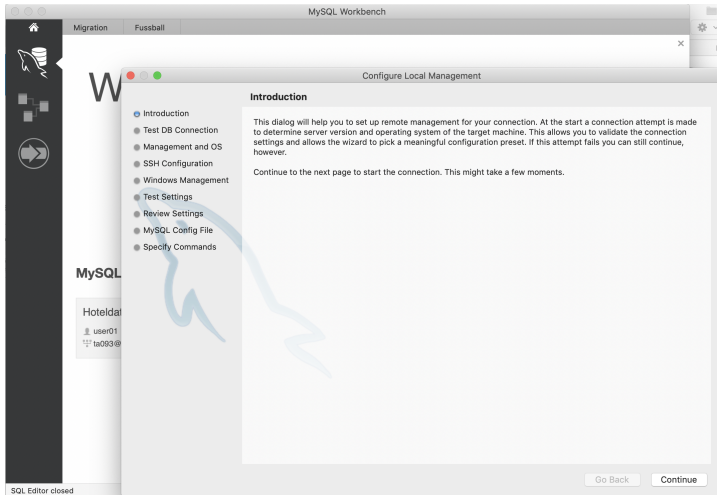
⇒ Root-Passwort setzen

evtl. in den Systemeinstellungen eine Datenbank initialisieren

# Datenbank in MySQL Workbench einbinden



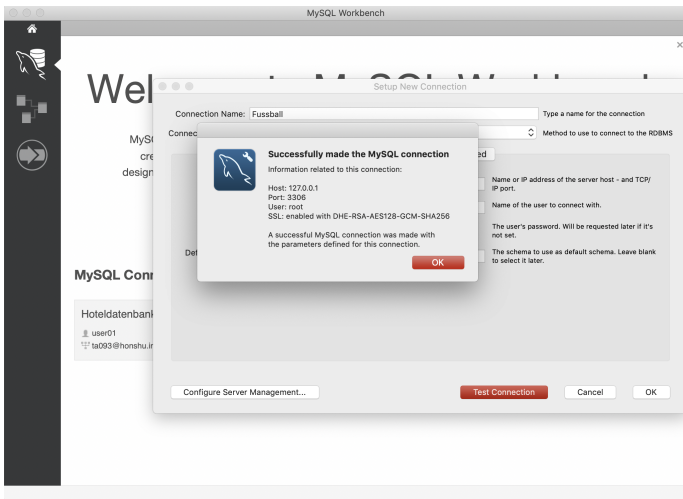
# Server Management konfigurieren (Mac)



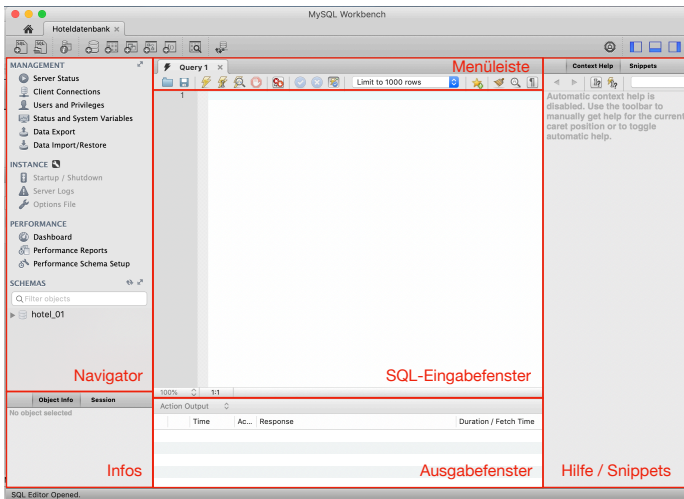
**Tutorial:** <https://dev.mysql.com/doc/workbench/en/wb-getting-started-tutorial-create-connection.html>



# Connection-Test



# Übersicht



# Menüleiste



Abbildung: Menüleiste

Symbole der Menüleiste:

- Ordner: SQL-Skript laden
- Diskette: SQL-Skript speichern
- Blitz: Anfrage ausführen
- Hand: Anfrage stoppen
- Stern: Anfrage in Schnellzugriff ablegen

# Navigator und Infos

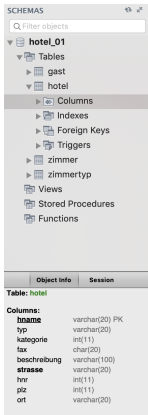


Abbildung: Datenbankschema

Schema besteht aus:

- Tabelle
  - Attribut
    - Typ
    - Schlüssel
  - Fremdschlüssel
- Sicht
- Funktion

Default-Schema festlegen!!



# Hilfe / Snippets und Log-Informationen

## Hilfe / Snippets:

- Hilfe zu einzelnen SQL-Konstrukten
- Snippets für Schnellzugriff bei häufiger Verwendung

## Log-Informationen:

- Letzte Anfrage
- Fehlermeldungen








Action Output 				
	3	Time 01	Action * FROM gast LIMIT 0, 1000	Response : 1046. No database selected Select
	4	15:44:37	SELECT * FROM hotel LIMIT 0, 1000	Error Code: 1046. No database selected Select
	5	15:45:09	SELECT * FROM hotel LIMIT 0, 1000	5 row(s) returned
	6	16:21:31	SELECT typ FROM hotelhname LIMIT...	Error Code: 1146. Table 'hotel_01.hotelhname' c
	7	16:21:56	SELECT typ FROM hotel LIMIT 0, 1000	5 row(s) returned
	8	16:22:11	SELECT typ,hname FROM hotel LIMIT...	5 row(s) returned

Abbildung: Ausgabe Log-Informationen