



DATENBANKEN ERSTELLEN MIT MYSQL

Leandro Meister, Nathanael Bracher

1



WAS IST MYSQL?

- MySQL existiert seit 1994 Seit 2010 gehört es zu Oracle
- MySQL ist eines der weltweit verbreitetsten Datenbankverwaltungssysteme
- Das Haupteinsatzgebiet von MySQL ist die Datenspeicherung für Webservices



2

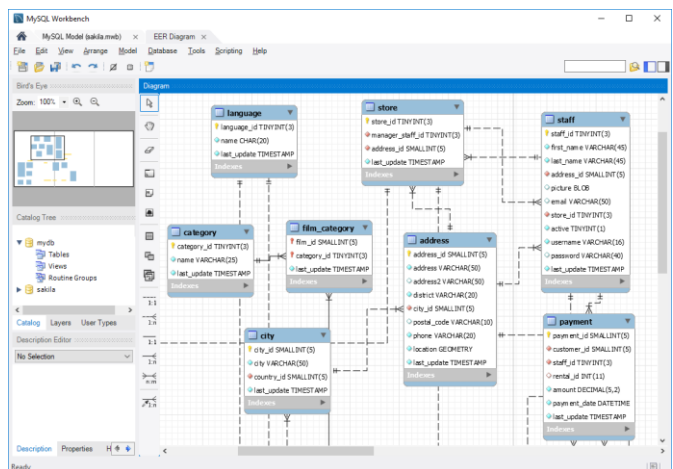
WAS IST DIE MYSQL WORKBENCH?

- Die MySQL Workbench ist eine grafische Benutzeroberfläche und eine Sammlung von Tools für die Arbeit mit MySQL Datenbanken.
- Mit der MySQL Workbench können auch bestehende Datenbanken übersichtlich dargestellt werden.



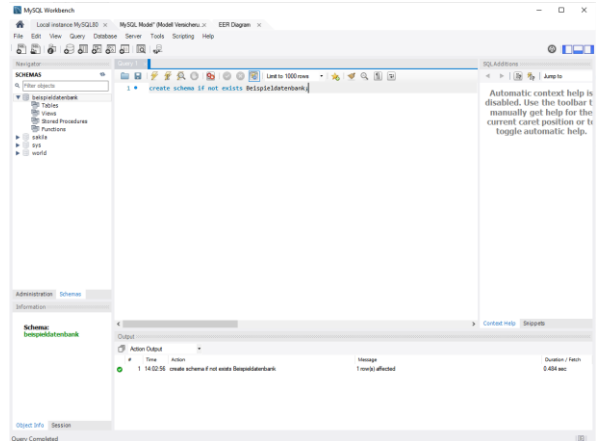
DATENBANKMODELL ERSTELLEN

- In der MySQL Workbench können Modelle von Datenbanken erstellt werden
- Daraus kann direkt eine Datenbank erzeugt werden
- Es kann auch von einer bestehenden Datenbank ein Modell erzeugt werden



DATENBANK MANUELL ERSTELLEN

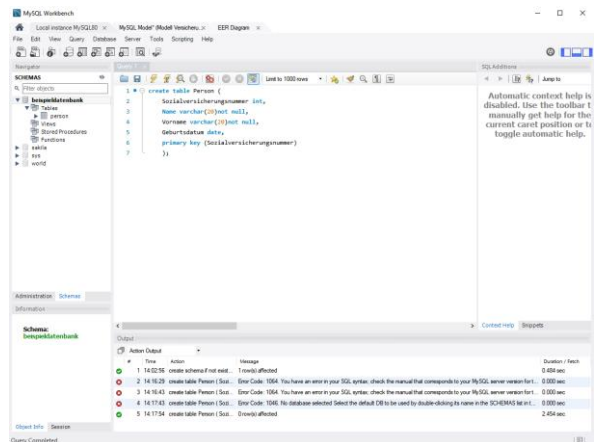
- In MySQL «new schema» anstatt «new database»
- If not exists verhindert das überschreiben einer Datenbank, falls diese bereits auf dem Server vorhanden ist.



5

TABELLE ERSTELLEN

- Tabelle erstellen mit dem Befehl «create table»
- Die Eigenschaften einer Spalte werden direkt nach dem Datentyp definiert
- Primary keys können entweder direkt in der Zeile oder zu unterst definiert werden.



6

BEZIEHUNGEN ZWISCHEN TABELLEN

- Beziehung über Fremdschlüssel (foreign key)
- Jedes Haus ist eindeutig einer Person (Besitzer des Hauses zugewiesen)
- Eine Person kann jedoch mehrere Häuser besitzen

```

1 create table if not exists Person (
2   Sozialversicherungsnummer int,
3   Name varchar(20) not null,
4   Vorname varchar(20) not null,
5   Geburtsdatum date,
6   primary key (Sozialversicherungsnummer)
7 );
8 create table if not exists Haus (
9   Hausnummer int,
10  Ort varchar(20),
11  Besitzer_ID int,
12  primary key (Hausnummer),
13  foreign key (Besitzer_ID) references Person (Sozialversicherungsnummer)
14 );

```

TABELLEN NACHTRÄGLICH BEARBEITEN ODER LÖSCHEN

- Tabellen bearbeiten mit dem Befehl «alter table»
- Spalten können gelöscht, hinzugefügt oder geändert werden
- Die gesamte Tabelle kann mit dem Befehl «drop table» gelöscht werden

```

1 alter table Person
2 drop column Telefonnummer,
3 add column email varchar(20),
4 modify column Geburtsdatum date not null;

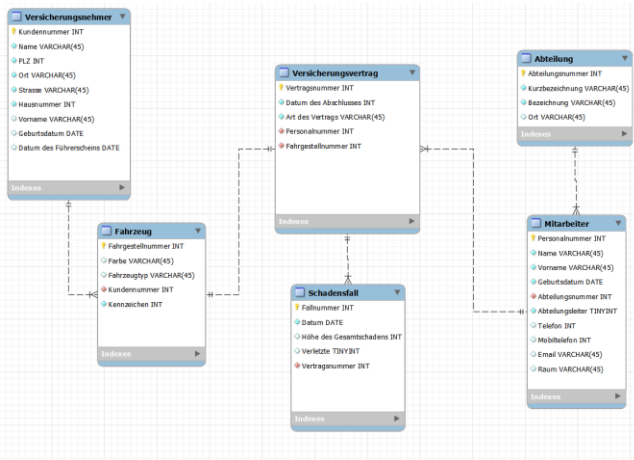
```

```

1 drop table Haus
2

```

BEISPIEL MOTORFAHRZEUGVERSICHERUNG



03.06.2022

Präsentation MySQL Leandro Meister, Nathanael Bracher

9

9

DATEN IN EINE TABELLE EINFÜGEN

- Daten werden mit dem Befehl «insert» in die Tabelle eingefügt werden.
- In der MySQL Workbench kann die gesamte Tabelle ausgewählt, und die Daten manuell eingegeben werden. Der code für den «insert» Befehl wird dann automatisch generiert.

```
1 • select*from abteilung
```

Abteilungsnummer	Kurzbezeichnung	Bezeichnung	Ort
1	sab	Sachbearbeitung	Bern
2	inf	Informatik	Zürich
3	buh	Buchhaltung	Bern
4	kud	Kundendienst	Bern

```
1 • insert into versicherungsnehmer
2 (Kundennummer, Name, Vorname, PLZ, Ort, Strasse, Hausnummer, Geburtsdatum, 'Datum des Führerscheins')
3 values (16, "Bauer", "Erika", 2839, "Zürich", "Lerchenweg", 28, "1940-03-03", "1960-09-01")
4 ;
```

03.06.2022

Präsentation MySQL Leandro Meister, Nathanael Bracher

10

10

DATEN IN EINER TABELLE BEARBEITEN ODER LÖSCHEN

- Einträge können mit dem Befehl «update table» bearbeitet werden
- Es können auch nur einzelne Einträge einer Zeile verändert werden.
- Einzelne Einträge können mit dem Befehl «delete from table» und einer where bedingung gelöscht werden
- Wird dieser Befehl ohne Bedingung ausgeführt, werden alle Einträge in der Tabelle gelöscht.

```
1 • update versicherungsnehmer
2   set Ort = "Bern", PLZ = 3105, Strasse = "Hauptstrasse", Hausnummer = 23
3   where Kundennummer = 12;
```

```
1 • delete from Versicherungsnehmer
2   where Kundennummer = 17
```

DATEN AUS EINER TABELLE ABFRAGEN

- Mit dem Befehl «select from table» können Einträge aus einer Tabelle abgerufen werden.
- Wenn anstelle des * eine konkrete Spalte angegeben wird, wird nur diese Spalte ausgegeben.

```
1 • select * from Versicherungsnehmer
```

Kundennummer	Name	PLZ	Ort	Strasse	Hausnummer	Kategorie	Geburtsdatum	Datum des Führerscheins
1	Muellermann	3563	Bern	Muellerstrasse	23	Max	2000-12-20	2022-01-01
2	Meiser	3300	Herrnhofen	Rindgasse	20	Leandro	1992-06-16	2015-12-13
3	Bracher	2503	Oltenen	Gürtenberg	20	Nathanael	2000-11-09	2021-01-15
4	Meier	6752	Zürich	Langweg	23	Sophie	1999-01-01	2022-05-20
5	Jackson	1007	New York	Bradway	85	Michael	1980-09-23	1980-10-12
6	Jahn	1007	London	Esplanadeplatz	17	Ethan	1990-04-20	1970-04-14
7	Höhnes	1643	London	Baker Street	221	Sherlock	1930-09-23	1948-10-12
8	Cip	3300	Sandering	Strangweg	11	Adolf	1940-12-25	1953-11-11
9	Berger	2333	Oberolten	Hauptstrasse	65	Sara	1922-11-24	1963-09-11
10	Palermis	1980	Riviera	village centrale	700	Nathanael	2002-11-09	2021-12-03
11	Andersson	2391	Aichendurf	Strandweg	1	Jula	1976-11-12	2000-10-09
12	Meier	3105	Bern	Hauptstrasse	23	Sandra	1998-09-12	2022-01-06
13	Hüller	7361	Siedlhorn	Aarweg	3	Hilke	1973-03-12	1990-01-13
14	Müller	2709	Oberglönges	Waldstrasse	67	Hane	1998-09-13	1970-04-10
15	Andersson	7051	Bern	Münsterstrasse	29	Marlen	1985-05-12	2021-04-10
16	Bauer	2839	Zürich	Lärchenweg	28	Erka	1940-03-03	1960-01-01
17	0000	0000	0000	0000	0000	0000	0000	0000

```
1 • select Name from Versicherungsnehmer
```

Name
Muellermann
Meiser
Bracher
Meier
Jackson
Jahn
Höhnes
Cip
Berger
Palermis
Andersson
Meier
Hüller
Müller
Andersson
Bauer

ABFRAGEN ÜBER MEHRERE TABELLEN

- Abfragen über mehrere Tabellen können mit dem Befehl «join» kombiniert mit einer «select from table» Anweisung realisiert werden.
- Wenn nur join geschrieben wird, wird ein «inner join» ausgeführt. Es kann auch ein «left join» «right join» oder «full join» ausgeführt werden.

```

1 • select
2   versicherungsnnehmer.Vorname,
3   versicherungsnnehmer.Name,
4   Marke, Typ, Farbe
5 from Fahrzeug
6 join versicherungsnnehmer
7   on Fahrzeug.Kundennummer = versicherungsnnehmer.Kundennummer
8 where versicherungsnnehmer.kundennummer = 3;
9

```

Vorname	Name	Marke	Typ	Farbe
Nathanael	Bracher	Ferrari	458	rot
Nathanael	Bracher	Audi	A4	braun

VIEW ERSTELLEN

- Ein view ist eine virtuelle Tabelle die auf einem Resultat einer SQL Abfrage basiert
- Da die Daten bereits vorhanden sind wäre es unsinnig eigens eine Tabelle dafür zu erstellen.
- Aus dem view können weitere Abfragen ausgeführt werden, wie mit einer richtigen tabelle

```

1 • create view Abteilungsleiter as
2   select name, vorname, Personalnummer
3   from Mitarbeiter
4   where abteilungsleiter=1;

```

```

1 • select*from Abteilungsleiter

```

name	vorname	Personalnummer
Siegenthaler	Martin	1
Müller	Hans	3
Hunziker	Sara	4
Hugentobler	Markus	5

STORED PROCEDURE ERSTELLEN

- Gespeichertes SQL Statement
- Eine Abfrage wird mit einem übergebenen parameter ausgeführt.
- Die Abfrage muss nicht mehrmals geschrieben werden. Es müssen lediglich die Parameter verändert werden.

```

1 DELIMITER //
2
3 • DROP PROCEDURE IF EXISTS UnfallstatistikJahr//
4
5 • CREATE PROCEDURE UnfallstatistikJahr(IN Jahr INT)
6 BEGIN
7     select
8     sum(schadensfall.Höhe des Gesamtschadens) as Gesamtschadenshöhe,
9     count(schadensfall.Fallnummer) as 'Anzahl der Unfälle',
10    sum(schadensfall.Verletzte) as 'verursachte Verletzungen',
11    max(schadensfall.datum) as 'Datum des letzten Unfalls'
12 from versicherungsnehmer
13 join Fahrzeug
14 on versicherungsnehmer.Kundennummer = Fahrzeug.Kundennummer
15 join versicherungsvertrag
16 on Fahrzeug.Fahrgestellnummer = versicherungsvertrag.Fahrgestellnummer
17 join schadensfall
18 on versicherungsvertrag.Vertragsnummer = schadensfall.Vertragsnummer
19 where YEAR(schadensfall.Datum) = Jahr
20 ;
21 END //
22
23 DELIMITER ;

```

```

1 • CALL UnfallstatistikJahr{2021}

```

Gesamtschadenshöhe	Anzahl der Unfälle	verursachte Verletzungen	Datum des letzten Unfalls
247234	2	3	2021-04-01