

ETH zürich



Grundlagen von SQL

Informatik 2, FS18

Dr. Hermann Lehner (Material von Dr. Markus Dahinden)
Departement Informatik, ETH Zürich

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 1

ETH zürich

Grundlagen von SQL

(Structured Query Language)

- Datenbanksprache
- Befehle
 - Datenbanken und Tabellen erstellen/verändern
 - Daten manipulieren (eingeben, ändern, löschen)
 - Datenbank durchsuchen (Queries erstellen)
- Die meisten Datenbanksysteme unterstützen SQL

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 2

ETH zürich

Google, Facebook und Co. setzen auf SQL



Google "Grundlagen der Informatik" ETH

Alle Bilder Shopping Videos News Mehr Einstellungen Tools

Ungefähr 10'300 Ergebnisse (0.44 Sekunden)

GDI - Aktuelles
<https://www.gdi.ethz.ch/>
Willkommen zur Vorlesung "Grundlagen der Informatik". Aktuelles ... erhalten Sie im Fallblatt n.ethz.ch - Kurzübersicht eine Antwort. Beispielprüfung: Computerräume - Lösungsschweise

Grundlagen der Informatik - Vorlesungsverzeichnis
www.vz.ethz.ch/lerneinheit/Pre.do?semkez=2015W&lerneinheit: Suche im Lehrangebot der ETH Zürich. ... 252-0862-00L Grundlagen Herbstsemester 2015. Dozierende: J. Hornikovic, H.-J. ...

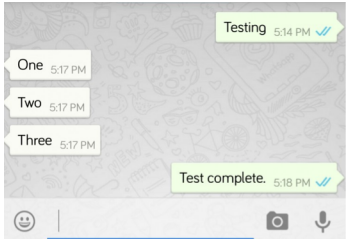
Aktuelles – Grundlagen der Informatik (Gdi), ETH Z
https://www.gdi.ethz.ch/14_hs14/

facebook

inf Informatik Computer Science goo.gl/R5sbo8 (Google) goo.gl/pduZqI (Facebook) Markus Dahinden | 13.05.18 | 3

ETH zürich

Whatsapp



wa.db msgstore.db

Tables (3) android_metadata locale sqlite_sequence name seq wa_contacts id jid is_whatsapp_user is_iphone status status number raw_contact_id media_uri display_name phone_type phone_label unseen_msg_count photo_ts

Tables (3) chat_list _id key_remote_jid message_table_id messages _id key_remote_jid key_from_me key_id status needs_nuch data timestamp media_uri media_mime_type media_wa_type media_size media_name latitude longitude thumb_image remote_resource received_timestamp send_timestamp receipt_server_timestamp receipt_device_timestamp sqlite_sequence name seq

Pro Tag werden 44 Mia Nachrichten verschickt. Für eine Nachricht sind 4 SQL Statements nötig. Das sind 2x176 Milliarden SQL Statements pro Tag!

inf Informatik Computer Science goo.gl/fj7d9u

ETH zürich

Liste >SQL> relationale Datenbank >SQL> Liste

Datenbanksystem

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 5

ETH zürich

Befehle zur Daten-Definition	Befehle zur Daten-Manipulation/Abfrage	Alias und Funktionen
CREATE SCHEMA	INSERT INTO .. VALUES ..	AS
CREATE TABLE	SELECT	COUNT()
NOT NULL	FROM	AVG()
UNIQUE	WHERE	MAX()
PRIMARY KEY	GROUP BY	MIN()
FOREIGN KEY	ORDER BY	ROUND()
REFERENCES	UPDATE	SUBSTRING()
CHECK	DELETE	
CREATE INDEX	COMMIT	
CREATE VIEW	ROLLBACK	
ALTER TABLE	= > < <= >= <> LIKE	
CREATE TABLE AS	AND OR NOT	
DROP TABLE	ASC DESC	
DROP VIEW	JOIN	

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 6

ETH zürich

Datenbasis für nachfolgende Übungen

Studierende

Vorname	Nachname	Legi	Wohnort
Heinz	Meier	12-333-333	Zürich
Anne	Meier	13-123-456	Luzern
Michaela	Zbinden	13-222-222	Bern
Karl-Heinz	Meierhans	14-444-444	Zürich

Faecher

ID	Bezeichnung	Departement
1	Botanik	D-BIOL
2	Statistik	D-MATH
3	Informatik	D-INFK
4	Lin. Algebra	D-MATH

Noten

ID	Legi	FachID	Note
1	12-333-333	1	6
2	13-222-222	3	5
4	13-222-222	1	5,5
5	14-444-444	2	3,5

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 7

ETH zürich

Studierende

Vorname	Nachname	Legi	Wohnort

Tabelle erstellen

```
CREATE TABLE Studierende
( Vorname TEXT NOT NULL,
  Name TEXT NOT NULL,
  Legi TEXT NOT NULL,
  Wohnort TEXT NOT NULL,
  PRIMARY KEY (Legi)
);
```

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 8

ETH zürich

Tabelle erstellen

Noten

ID	Legi	FachID	Note

```
CREATE TABLE Noten
( ID INTEGER NOT NULL CHECK(typeof(Id) = 'integer'),
  Legi TEXT NOT NULL,
  FachID INTEGER NOT NULL CHECK(typeof(Id) = 'integer'),
  Note REAL NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (Legi) REFERENCES Studierende (Legi)
);
```

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 9

ETH zürich

```
FOREIGN KEY (Legi) REFERENCES Studierende (Legi)
```

Studierende

Vorname	Nachname	Legi	Wohnort
H		12-333-333	Zürich
An		13-123-456	Luzern
Mic		13-222-222	Bern
Karl		14-444-444	Zürich

erlaubt die Kontrolle der referentiellen Integrität!

Noten

ID	Legi	FachID	Note
1	12-333-333	1	6
2	13-222-222	3	5
4	13-222-222	1	5,5
5	14-444-444	2	3,5

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 10

ETH zürich

Datensatz in eine Tabelle einfügen

Studierende

Vorname	Nachname	Legi	Wohnort
Heinz	Meier	12-333-333	Zürich
Anne	Meier	13-123-456	Luzern
Michaela	Zbinden	13-222-222	Bern
Karl-Heinz	Meierhans	14-444-444	Zürich

```
INSERT INTO Studierende (Vorname, Nachname, Legi, Wohnort)
VALUES ('Heinz', 'Meier', '12-333-333', 'Zürich');
```

- Bei **Zahlen** sind keine Gänsefüßchen erlaubt!

```
INSERT INTO Noten (ID) VALUES (1)
```

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 11

ETH zürich

Einfachste Version einer SQL-Abfrage

```
SELECT *
FROM Studierende;
```

→ liefert alle Einträge der Tabelle "Studierende"

- Wildcard-Zeichen (*) bedeutet: "alle Attribute dieser Tabelle"
- Semikolon (;) schliesst eine Abfrage ab

inf Informatik Computer Science Markus Dahinden | 13.05.18 | 12

ETH zürich

Wahl der anzuzeigenden Attribute

```
SELECT Vorname, Nachname
FROM Studierende;
```

→ liefert von allen Tuples der Tabelle "Studierende" die Vor- und Nachnamen

inf informatik
Computer Science Markus Dahinden | 13.05.18 | 13

ETH zürich

Einschränken der Tuples mit WHERE

```
SELECT Vorname, Nachname
FROM Studierende
WHERE Nachname = 'Meier';
```

→ liefert nur jene Einträge der Tabelle "Studierende", die den Nachnamen 'Meier' tragen.

- WHERE erlaubt logische Operatoren z.B. AND/OR

inf informatik
Computer Science Markus Dahinden | 13.05.18 | 14

ETH zürich

Festlegen der Sortierreihenfolge

```
SELECT Vorname, Nachname
FROM Studierende
WHERE Nachname = 'Meier'
ORDER BY Vorname ASC;
```

→ liefert das Ergebnis aufsteigend sortiert nach Vornamen

inf informatik
Computer Science Markus Dahinden | 13.05.18 | 15

ETH zürich

Übung: einfache Abfragen I

Welche Resultate liefern folgende SQL-Abfragen?

```
SELECT *
FROM Faecher;
```

```
SELECT ID, Bezeichnung
FROM Faecher;
```

```
SELECT Bezeichnung;
FROM Faecher
ORDER BY Departement DESC;
```

inf informatik
Computer Science Markus Dahinden | 13.05.18 | 16

ETH zürich

Übung: einfache Abfragen I (Lösungen)

Welche Resultate liefern folgende SQL-Abfragen?

```
SELECT *
FROM Faecher;
```

ID	Bezeichnung	Departement
1	Botanik	D-BIOL
2	Statistik	D-MATH
3	Informatik	D-INFK
4	Lin. Algebra	D-MATH

```
SELECT ID, Bezeichnung
FROM Faecher;
```

ID	Bezeichnung
1	Botanik
2	Statistik
3	Informatik
4	Lin. Algebra

```
SELECT Bezeichnung
FROM Faecher
ORDER BY Departement DESC;
```

Bezeichnung
Lin. Algebra
Statistik
Informatik
Botanik

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 17

ETH zürich

Übung: einfache Abfragen II

Welche Resultate liefern folgende SQL-Abfragen?

```
SELECT Legi, Vorname, Nachname
FROM Studierende
WHERE Nachname = 'Brunner';
```

```
SELECT count(Legi) AS Anzahl
FROM Studierende
WHERE Wohnort = 'Zürich';
```

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 18

ETH zürich

Übung: einfache Abfragen II (Lösungen)

Welche Resultate liefern folgende SQL-Abfragen?

```
SELECT Legi, Vorname, Nachname
FROM Studierende
WHERE Nachname = 'Brunner';
```

Legi	Vorname	Nachname

```
SELECT count(Legi) AS Anzahl
FROM Studierende
WHERE Wohnort = 'Zürich';
```

Anzahl
2

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 19

ETH zürich

SQL: einfache Abfragen III (Übung)

Schreiben Sie folgende SQL-Query:

Liste mit allen Studierenden (Legi, Vorname, Nachname), die in Bern beziehungsweise in Luzern wohnen, sortiert nach Nachnamen von A bis Z.

Legi	Vorname	Nachname
13-123-456	Anne	Meier
13-222-222	Michaela	Zbinden

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 20

ETH zürich

Abfrage über zwei Tabellen hinweg

```

SELECT *
FROM Noten, Faecher;
    
```

ID	Legi	FachID	Note	ID	Bezeichnung	Departement
1	12-333-333	1	6	1	Botanik	D-BIOL
2	13-222-222	3	5	3	Informatik	D-INFK
4	13-222-222	1	5,5	1	Botanik	D-BIOL
5	14-444-444	2	3,5	2	Statistik	D-MATH

inf | 13.05.18 | 21

ETH zürich

erwartetes Resultat

ID	Legi	FachID	Note	ID	Bezeichnung	Departement
1	12-333-333	1	6	1	Botanik	D-BIOL
2	13-222-222	3	5	3	Informatik	D-INFK
4	13-222-222	1	5,5	1	Botanik	D-BIOL
5	14-444-444	2	3,5	2	Statistik	D-MATH

erhaltenes Resultat

Kreuzprodukt

ID	Legi	FachID	Note	ID	Bezeichnung	Departement
1	12-333-333	1	6	1	Botanik	D-BIOL
1	12-333-333	1	6	2	Statistik	D-MATH
1	12-333-333	1	6	3	Informatik	D-INFK
1	12-333-333	1	6	4	Lin. Algebra	D-MATH
2	13-222-222	3	5	1	Botanik	D-BIOL
2	13-222-222	3	5	2	Statistik	D-MATH
2	13-222-222	3	5	3	Informatik	D-INFK
2	13-222-222	3	5	4	Lin. Algebra	D-MATH
4	13-222-222	1	5,5	1	Botanik	D-BIOL
4	13-222-222	1	5,5	2	Statistik	D-MATH
4	13-222-222	1	5,5	3	Informatik	D-INFK
4	13-222-222	1	5,5	4	Lin. Algebra	D-MATH
5	14-444-444	2	3,5	1	Botanik	D-BIOL
5	14-444-444	2	3,5	2	Statistik	D-MATH
5	14-444-444	2	3,5	3	Informatik	D-INFK
5	14-444-444	2	3,5	4	Lin. Algebra	D-MATH

inf | 13.05.18 | 22

ETH zürich

Aufteilen der Daten auf zwei Tabellen

ID	Legi	FachID	ID	Bezeichnung	Departement
1	12-333-333	1	1	Botanik	D-BIOL
2	13-222-222	3	3	Statistik	D-MATH
4	13-222-222	1	1	Informatik	D-INFK
5	14-444-444	2	2	Lin. Algebra	D-MATH

inf | 13.05.18 | 23

ETH zürich

Beziehungen zwischen Tabellen berücksichtigen

ID	Legi	FachID	Note	ID	Bezeichnung	Departement
1	12-333-333	1	6	1	Botanik	D-BIOL
1	12-333-333	1	6	2	Statistik	D-MATH
1	12-333-333	1	6	3	Informatik	D-INFK
1	12-333-333	1	6	4	Lin. Algebra	D-MATH
2	13-222-222	3	5	1	Botanik	D-BIOL
2	13-222-222	3	5	2	Statistik	D-MATH
2	13-222-222	3	5	3	Informatik	D-INFK
2	13-222-222	3	5	4	Lin. Algebra	D-MATH
4	13-222-222	1	5,5	1	Botanik	D-BIOL
4	13-222-222	1	5,5	2	Statistik	D-MATH
4	13-222-222	1	5,5	3	Informatik	D-INFK
4	13-222-222	1	5,5	4	Lin. Algebra	D-MATH
5	14-444-444	2	3,5	1	Botanik	D-BIOL
5	14-444-444	2	3,5	2	Statistik	D-MATH
5	14-444-444	2	3,5	3	Informatik	D-INFK
5	14-444-444	2	3,5	4	Lin. Algebra	D-MATH

FachID = ID

inf | 13.05.18 | 24

ETH zürich

Korrekte Verknüpfung von *Noten* und *Faecher*

```

SELECT *
FROM Noten JOIN Faecher
ON Noten.FachID = Faecher.ID;

```

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 25

ETH zürich

Übung: komplexere Abfragen I

Schreiben sie die entsprechende SQL Abfrage

Die Nachnamen aller Studierenden, welche eine ungenügende Note erzielt haben.

Die Nachnamen und die Noten aller Studierenden mit der Note 6 oder mit dem Nachnamen 'Zbinden'.

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 26

ETH zürich

Übung: komplexere Abfragen I (Lösungen)

Welche Resultate liefern folgende SQL-Abfragen?

Die Nachnamen aller Studierenden, welche eine ungenügende Note erzielt haben.

```

SELECT Studierende.Nachname
FROM Studierende JOIN Noten
ON Studierende.Legi = Noten.Legi
WHERE Noten.Note < 4;

```

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 27

ETH zürich

Übung: komplexere Abfragen I (Lösungen)

Welche Resultate liefern folgende SQL-Abfragen?

Die Nachnamen und die Noten aller Studierenden mit der Note 6 oder mit dem Nachnamen 'Zbinden'.

```

SELECT Studierende.Nachname, Noten.Note
FROM Studierende JOIN Noten
ON Studierende.Legi = Noten.Legi
WHERE Noten.Note = 6
OR Studierende.Nachname = 'Zbinden';

```

inf Informatik
Computer Science Markus Dahinden | 13.05.18 | 28

Kürzel für Tabellen verwenden*Welche Resultate liefern folgende SQL-Abfragen?*

Die Nachnamen und die Noten aller Studierenden mit der Note 6 oder mit dem Nachnamen 'Zbinden'.

```
SELECT s.Nachname, n.Note
FROM Studierende s JOIN Noten n
  ON s.Legi = n.Legi
WHERE n.Note = 6
  OR s.Nachname = 'Zbinden'
```

Verknüpfen mit JOIN...ON statt WHERE

```
SELECT *
FROM Noten JOIN Faecher
  ON Noten.FachID = Faecher.ID;
```

```
SELECT *
FROM Noten, Faecher
WHERE Noten.FachID = Faecher.ID;
```

Abfrage ohne JOIN (dafür mit WHERE)

```
SELECT Studierende.Nachname, Noten.Note
FROM Studierende JOIN Noten
  ON Studierende.Legi = Noten.Legi
WHERE Noten.Note = 6
  OR Studierende.Nachname = 'Zbinden';
```

```
SELECT Studierende.Nachname, Noten.Note
FROM Studierende, Noten
WHERE Studierende.Legi = Noten.Legi
  AND ( Noten.Note = 6
  OR Studierende.Nachname = 'Zbinden');
```

Welches Resultat erhalten Sie, wenn Sie die Klammer weglassen?

```
SELECT Studierende.Nachname, Noten.Note
FROM Studierende, Noten
WHERE Studierende.Legi = Noten.Legi
  AND ( Noten.Note = 6
  OR Studierende.Nachname = 'Zbinden');
```


Rangfolge unterschiedlicher Operatoren

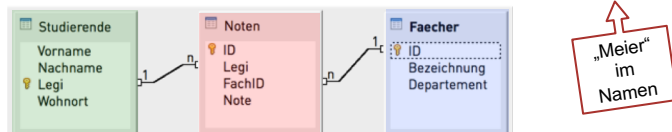
1. Arithmetische Operatoren (Punkt vor Strich)
2. Relationale Operatoren (Vergleichsoperatoren)
3. Logische Operatoren:
 1. Negation (NOT)
 2. Konjunktion (AND)
 3. Disjunktion (OR)

Das OR macht die ganze Arbeit kaputt!

ID	Legi	FachID	Note	Vorname	Nachname	Legi	Wohnort
1	12-333-333	1	6	Heinz	Meier	12-333-333	Zürich
1	12-333-333	1	6	Anne	Meier	13-123-456	Luzern
1	12-333-333	1	6	Michaela	Zbinden	13-222-222	Bern
1	12-333-333	1	6	Karl-Heinz	Meierhans	14-444-444	Zürich
2	13-222-222	3	5	Heinz	Meier	12-333-333	Zürich
2	13-222-222	3	5	Anne	Meier	13-123-456	Luzern
2	13-222-222	3	5	Michaela	Zbinden	13-222-222	Bern
2	13-222-222	3	5	Karl-Heinz	Meierhans	14-444-444	Zürich
4	13-222-222	1	5.5	Heinz	Meier	12-333-333	Zürich
4	13-222-222	1	5.5	Anne	Meier	13-123-456	Luzern
4	13-222-222	1	5.5	Michaela	Zbinden	13-222-222	Bern
4	13-222-222	1	5.5	Karl-Heinz	Meierhans	14-444-444	Zürich
5	14-444-444	2	3.5	Heinz	Meier	12-333-333	Zürich
5	14-444-444	2	3.5	Anne	Meier	13-123-456	Luzern
5	14-444-444	2	3.5	Michaela	Zbinden	13-222-222	Bern
5	14-444-444	2	3.5	Karl-Heinz	Meierhans	14-444-444	Zürich

Übung: Abfrage über drei Tabellen

Legi	Note	Bezeichnung
12-333-333	4,5	Botanik
14-444-444	3,5	Statistik

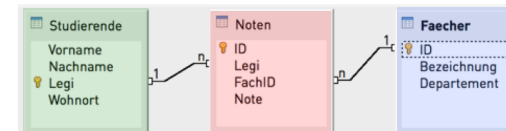


```

SELECT Studierende.Nachname, Noten.Note,
Faecher.Bezeichnung
FROM Studierende JOIN Noten
  ON Studierende.Legi = Noten.Legi
JOIN Faecher
  ON Noten.FachID = Faecher.ID
WHERE Studierende.Nachname LIKE 'Meier%';

```

Übung: Umformen in ein WHERE-Statement



```

SELECT Studierende.Legi, Noten.Note,
Faecher.Bezeichnung
FROM Studierende, Noten, Faecher
WHERE Studierende.Legi = Noten.Legi
  AND Noten.FachID = Faecher.ID
  AND Studierende.Nachname LIKE 'Meier%';


```

ETH zürich

Referentielle Integrität von Datenbanken

Die Integrität einer Datenbank ist dann sichergestellt, wenn:

- Alle Primärschlüssel eindeutig sind
- Alle Fremdschlüssel auf einen Primärschlüssel verweisen

 Wenn die Integrität verletzt ist, sind die Daten in der Datenbank fehlerhaft!

inf Informatik
Computer Science

Markus Dahinden | 13.05.18 | 37

ETH zürich

Welche Tuples können gelöscht werden, ohne die referentielle Integrität zu verletzen?

Studierende

Vorname	Nachname	Legi	Wohnort
Heinz	Meier	12-333-333	Zürich
Anne	Meier	13-123-456	Luzern
Michaela	Zbinden	13-222-222	Bern
Karl-Heinz	Meierhans	14-444-444	Zürich

Faecher

ID	Bezeichnung	Departement
1	Botanik	D-BIOL
2	Statistik	D-MATH
3	Informatik	D-INFK
4	Lin. Algebra	D-MATH

Noten

ID	Legi	FachID	Note
1	12-333-333	1	6
2	13-222-222	3	5
4	13-222-222	1	5,5
5	14-444-444	2	3,5

inf Informatik
Computer Science

Markus Dahinden | 13.05.18 | 38

ETH zürich

Google

ETH OR ETHZ "Grundlagen der Informatik" -EPFL

About 48.800 results (0,72 seconds)

GDI - Aktuelles
<https://www.gdi.ethz.ch/> - Translate this page
 Willkommen zur Vorlesung "Grundlagen der Informatik". Aktuelles... (E.Tutorials). 29.08.2017, n.ethz
 Infobroschüre Was ist n.ethz? Wie kann ich mein Passwort ändern? Wie komme ich mit meinem Laptop auf das ETH-Netz? Auf diese und weitere Fragen erhalten Sie im Falblatt n.ethz.ch - Kurzübersicht eine Antwort.

252-0852-00L Grundlagen der Informatik - ETH Zürich ...
www.vz.ethz.ch/lehre/nethz/Pre.do?semkez=2015W...&lang... - Translate this page
 Infobroschüre zum Pre-Startkolloquium, dem ersten Vorlesungstag und dem ersten Informatik-Prüfungstag.

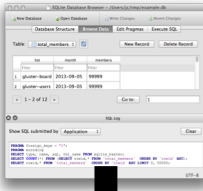
```
SELECT URL, Seitentitel, substring(Seitentext,0,150)
FROM AlleWebseitenWeltweit
WHERE Seitentext like '%Grundlagen der Informatik%';
AND Seitentext like '%ETH%'
AND NOT Seitentext = 'EPFL';
```

inf Informatik
Computer Science


Markus Dahinden | 13.05.18 | 39

ETH zürich

Probleme aus dem Übungsbetrieb (Teil I)



```
CREATE TABLE Noten
( ID INTEGER NOT NULL
CHECK(typeof(Id) = 'Integer'),
PRIMARY KEY (ID),
);
```


sqlite-test.db

inf Informatik
Computer Science

Markus Dahinden | 13.05.18 | 40

Probleme aus dem Übungsbetrieb (Teil II)

```

CREATE TABLE Studierende      CREATE TABLE Noten
( Name TEXT NOT NULL,         ( ref_Studi TEXT NOT NULL,
  Legi TEXT NOT NULL,         Note REAL NOT NULL,
  PRIMARY KEY (Legi)          FOREIGN KEY(ref_Studi)
);                             REFERENCES Studierende(Legi),
                              PRIMARY KEY(ref_Studi,Note)
                              );

INSERT INTO Noten(ref_Studi,Note) VALUES ('11-111-111',3.5)
>> Query successful
INSERT INTO Noten(ref_Studi,Note) VALUES ('33-333-333',5.5)
>> Foreign Key Constraint!
INSERT INTO Noten(ref_Studi,Note) VALUES ('11-111-111',3.5)
>> Unique Key Constraint!

```

Finde die Fehler

```

select kunden.id
from kunden
where kunden.name = "Meier%"      1 Fehler

select kunden.name
from kunden, einkauf
where artikel = "Milch"          2 Fehler

select count(kunden.name) as Anzahl
from kunden inner join einkauf on
      kunden.id = einkauf.kunden_id  1(+1) Fehler
where einkauf.artikel like "Milch" OR "Reis"

```

Zusammenfassung

- CREATE TABLE legt in der DB eine neue Tabelle an
- Mit INSERT INTO können Datensätze in eine Tabelle eingefügt werden
- Mit SELECT können Sie Daten aus einer oder mehreren Tabellen abrufen
- JOIN...ON resp. FROM...WHERE führt Datensätze aus mehreren Tabellen zusammen
- Fehlende Beziehungen zwischen Tabellen liefern deren Kreuzprodukt
- Wenn die Referentielle Integrität einer Datenbank verletzt ist, kann das Resultat einer Abfrage falsch sein.