MySQL – Informationsgewinnung
Teil 1

Stefan Maihack Dipl. Ing. (FH)
Datum: 19.4.2019

Die SELECT-Anweisung

Einführung

- Daten in einer Datenbank abfragen mit der SELECT-Anweisung

SELECT
[DISTINCT]
Auswahlausdruck,…
[INTO {OUTFILE | DUMPFILE} ,/Pfad/zu/Dateiname‘ Exportoptionen]
[FROM Liste der Tabellen]
[WHERE Suchbedingungen]
[GROUP BY {Spaltenname | Spaltenalias} [ASC | DESC], …]
[HAVING 2. Auswahlbedingung]
[ORDER BY {Spaltenname | Spaltenalias} [ASC | DESC], …]
[LIMIT [OFFSET,] Zeilenanzahl]

Beispiel:

SELECT * FROM artikel;

SELECT id, artikel_name, preis FROM artikel;
(Die Namen der Spalten erscheinen in der Ergebnismenge als Spaltenüberschriften)

Die SELECT-Anweisung

Zusätze

- Die WHERE-Klausel

  Mit der WHERE-Klausel kann man steuern, welche Zeilen abgerufen werden.

  Beispiel:

  SELECT artikel_name, preis FROM artikel WHERE artikel_id = 102;

- Es wird auch ermöglicht mit der SELECT-Anweisung Ausdrücke auszuwerten, ohne dass man sich auf eine Tabelle beziehen muss:

  SELECT 12.50 * 7, 1+2, ´hello world´;

- Die SELECT-Anweisung wertet auch Funktionen aus, die MySQL in einer umfangreichen Bibliothek bereitstellt. Z.B. Datums und Uhrzeitfunktionen:

  SELECT NOW();

Die SELECT-Anweisung

Ausgabespalten wählen

- Gib alle Spalten der Tabelle „buch“ aus:

  SELECT * FROM buch;

- Gib die Spalte „titel“ der Tabelle „buch“ aus.

  SELECT titel FROM buch;

- Gib die Spalten „auflage“ und „ISBN“ aus.

  SELECT auflage, isbn FROM buch;

➔ Die Reihenfolge der Spalten ist beliebig.

➔ Mehrere Spalten müssen durch Komma getrennt werden.

Die SELECT-Anweisung

Spezielle Werte auswählen

- Das reservierte Wort „WHERE“ leitet die Suchbedingung ein, z. B: einen Vergleich:

  SELECT titel FROM buch WHERE ISBN = 0596223514
  → Gibt den Titel des Buches mit der ISBN-Nummer „0596223514“ aus.

- Vergleichsoperatoren:

<	kleiner
<=	kleiner oder gleich
=	gleich
!= oder <>	ungleich
>=	größer oder gleich
>	größer

Hinweis:
Vergleich mit NULL ergibt NULL

NULL = 0	→ NULL
NULL = NULL	→ NULL

Die SELECT-Anweisung

Vergleich mit dem NULL-Wert

- Syntax: IS NULL oder IS NOT NULL
  (oder: Vergleichsoperator <=>) ab MySQL 3.23 = „IS NULL“

- Beispiele:

  SELECT titel FROM buch WHERE autor2 IS NOT NULL;
  → Gibt die Titel der Bücher aus, die einen zweiten Autor haben.

  SELECT ort FROM verlag WHERE ort <> NULL;
  → Gibt die Orte von Verlag aus, die in der Tabelle eingetragen sind.

- Verwendung von logischen Operatoren:

  SELECT titel FROM buch
  WHERE (autor2 is NOT NULL) AND (autor3 IS NOT NULL);
  → Gibt den Titel der Bücher aus, die einen zweiten und dritten Autor haben.

Die SELECT-Anweisung

Ergebnisse sortieren

- Syntax: ORDER BY Spaltenname [Sortierrichtung]

  Sortierrichtung: ASC = aufsteigend (ascending), von A nach Z
  DESC = abstzeigend (descending)
  ohne Angabe: aufsteigend (ASC)

- Beispiele:

  SELECT jahr, titel FROM buch ORDER BY jahr;
  → Gibt das Erscheinungsjahr und den Titel aller Bücher aus, die ältesten zu erst.

  SELECT jahr, titel FROM buch ORDER BY jahr DESC;
  → Gibt das Erscheinungsjahr und den Titel aller Bücher aus, die neusten zu erst.

Die SELECT-Anweisung

Zahl der Zeilen begrenzen

- Syntax: LIMIT [Erste_Zeile ,] Anzahl_zeilen

  Erste_Zeile = Nummer des ersten auszugebenden Datensatzes, die Nummerierung beginnt mit 0!

- Beispiele:

```
SELECT titel FROM buch LIMIT 10;
```

→ Gib den Titel der ersten 10 Bücher aus.

```
SELECT titel FROM buch limit 10, 15;
```

→ Gib den Titel von 15 Bücher aus, beginnend bei dem 11. Buch.

```
SELECT titel FROM buch
        WHERE (autor2 IS NULL)
        ORDER BY jahr desc
        LIMIT 10;
```

→ Gibt die 10 neusten Bücher von einem Autor aus.

Die SELECT-Anweisung

Numerische Funktionen

Funktion	Beschreibung
ABS(X)	Gibt den absoluten Wert von x zurück
SIN(x), COS(x), TAN(x), ASIN(x), ACOS(x), ATAN(x)	Winkelfunktionen
CEILING(x), CEIL(x)	Gibt den kleinsten Integer-Wert, der nicht kleiner ist als x ist. SELECT CEILING (1.23); ➔ 2 SELECT CEIL (-1.23); ➔ -1
FLOOR(x)	Gibt den größten Integer-Wert zurück, der nicht größer ist als der Wert x. SELECT FLOOR(1.23); ➔ 1 SELECT FLOOR(-1.23; ➔ -2
FORMAT(X,D)	Formatiert die Zahl x in ein Format „#,###,###.##“, gerundet auf D Dezimalstellen und gibt das Ergebnis als String zurück
MOD(N,M), N % M, N MOD M	Moduloperation. Gib den Rest von N geteilt durch M zurück. SELECT MOD(29,9); ➔ 2
POW(x,y)	Gibt den Wert von x potenziert zu y an. SELECT POW(2,2); ➔ 4

Die SELECT-Anweisung

Ergebnisse zusammenfassen - Aggregatsfunktionen

- Aggregatfunktionen führen Berechnungen für eine Wertemenge durch und geben einen einzelnen Wert zurück.
- Alle Aggregatfunktionen, außer COUNT, ignorieren NULL-Werte.
- Alle Aggregatfunktionen sind deterministisch ➔ Bei jedem Aufruf mit bestimmten Eingabewerte, liefern sie immer das gleiche Ergebnis zurück. Es treten nur reproduzierbare Zustände auf.
- Syntax: GROUP BY spaltenliste
  Sinnvoll im Zusammenhang mit Funktionen wie COUNT() (zählen), AVG() (Mittelwert bilden) etc.
- Beispiel:

  SELECT jahr, count(*) FROM buch GROUP BY jahr;

  → zählt, wie viele Bücher in einem Jahr erschienen sind.

Ergebnis:

Jahr	Count(*)
1999	1
2000	5
2001	20
2002	10

weitere Aggregatsfunktionen:

→AVG(): Mittelwert der Daten in der Gruppe
→MAX(): liefert den Maximalwert
→MIN(): liefert den Minimalwert
→STD(): gibt die Standardabweichung zurück
→ COUNT(): Anzahl der Werte in einer Spalte
→ SUM(): Summe aller Werte in einer Spalte

Die SELECT-Anweisung

Zweite Einschränkung der Werte

- Das SQL HAVING-Statement ist das SQL WHERE in einem SQL GROUP BY-Statement. Es ermöglicht eine gruppierte Ergebnismenge einzuschränken.
- Mit SQL HAVING kann die Ergebnismenge auf Basis der Aggregatfunktionen (AVG, COUNT, MAX, MIN, SUM) eingeschränkt und ausgegeben werden.
- Kennwort: HAVING leitet die eingeschränkte Bedingung ein. Wirkt auf die ausgewählten Daten Sinnvoll z.B. nach einer GROUP BY … Anweisung

- Beispiel:

```
SELECT jahr, count(*) FROM buch
        GROUP BY jahr HAVING jahr > 1999
```

→ Zählt wie viele Bücher in einem Jahr erschienen sind, und zwar nach dem Jahr 1999.

Ergebnis:

Jahr	Count(*)
2000	5
2001	20
2002	10

Die Aufstellung beginnt nun mit dem Jahr 2000.

Die SELECT-Anweisung

Volltextsuche

- Abfragen in ähnlicher Weise, wie man es von einer Suchmaschine im Web kennt.
- Zeilen werden nach bestimmten Zeichenfolgen durchsucht.
- Für eine Volltextsuche ist ein FULLINDEX für die durchsuchende(n) Spalte(n) erforderlich.
  ALTER TABLE cities ADD FULLTEXT(city_name);

- Beispiel: Die Tabelle „cities“ nach Städten mit dem Namen „Palm Springs“ durchsuchen:

  SELECT city_name FROM cities WHERE MATCH (city_name) AGAINST (‚Palm Springs‘);

- **Eine Suche mit MATCH erfordert weniger Zeit als mit dem LIKE-Operator.**

Die SELECT-Anweisung

Anspruchsvollere SELECT-Anweisungen
Unterabfragen

- Hierbei handelt es sich um geschachtelte SELECT-Anweisungen, d.h. eine SELECT-Anweisung in einer anderen aufführen.

- Bei einer Unterabfrage kann man eine SELECT-Anweisung auf eine Ergebnismenge ausführen, die selbst erst von einer anderen SELECT-Abfrage erzeugt worden ist.

  Beispiel:

  SELECT * FROM kinderartikel WHERE **id** IN
  (SELECT **id** FROM sichere_artikel);

Die SELECT-Anweisung

Anspruchsvollere SELECT-Anweisungen
Unterabfragen

- 2. Beispiel:

  Erstellen und befüllen von zwei Tabellen:
  CREATE TABLE Kunden (name CHAR(5), waggon_id INT);
  CREATE TABLE Waggons (waggon_id INT PRIMARY KEY, klasse CHAR(5));
  INSERT INTO Kunden VALUES (‚Klaus‘, 15);
  INSERT INTO Kunden VALUES (‚Rudi‘, 23);
  INSERT INTO Waggons VALUES (15, ‚erste‘);

  Alle Unterabfragen haben zwei Merkmale: sie stehen in Klammern und sind Teil eines übergeordneten SELECTs:

  SELECT name FROM Kunden WHERE **wagon_id** = /* Haupt */
  (SELECT **waggon_id** FROM Waggons WHERE klasse = ‚erste‘) /* Unter */

Die SELECT-Anweisung

Anspruchsvollere SELECT-Anweisungen
Ergebnismengen zusammenfassen - UNION

- Ergebnismengen mit UNION zusammenfassen
  Durch das Schlüsselwort UNION lassen sich zwei oder mehrere SELECT-Abfragen zusammenfassen, d.h. die Ergebnismengen.

```
SELECT ´teens´, MIN(preis) FROM teens_artikel
UNION
SELECT ´kids´, MIN(preis) FROM kinderartikel;
```

Teens	Min(preis)
Teens	19,00
Kids	7,50

  Die Datenformate der beiden SELECTs müssen identisch sein, andernfalls erhält man eine Fehlermeldung.

  Normalerweise arbeite jede SELECT-Abfrage so, als hätte man das Schlüsselwort DISTINCT angegeben. In der Ergebnismenge erscheinen deshalb nur eindeutige Zeilen, Wenn man das Schlüsselwort ALL noch anhängt, so werden alle Zeilen ausgegeben.

```
SELECT id FROM teens_artikel
UNION ALL
SELECT id FROM kinderartikel;
```

- ACHTUNG: Die Datenformate der einzelnen Ergebnismengen müssen identisch sein.
- Die Spaltennamen beziehen sich aus der ersten SELECT-Anweisung.

Die SELECT-Anweisung

Übungsaufgabe

- Folgende Tabellen der Datenbank „kunde_rechnung.sql“ liegen vor:

Relation: Rechnung

Rechnr	KdNr	Rechdat	RechBetrag
0081	2133	1.7.07	10600
0082	7533	23.7.07	28400
0083	2133	4.8.07	1000
0084	8511	5.8.07	20900

Relation: Kunden

KdNr	KdName	Ort	VNr
2133	Meier	Bayreuth	224
3557	Müller	Bonn	115
7533	Schmitz	Köln	115
8511	Schneider	Bonn	115

Relation: Vertreter

Vnr	Vname	Bezirk	UmsSum
211	Berger	Nord	74800
115	Adam	Mitte	127900
224	Zeiss	Süd	91750
315	Beck	Ost	0
452	Weih	West	10000

Relation: Lieferant

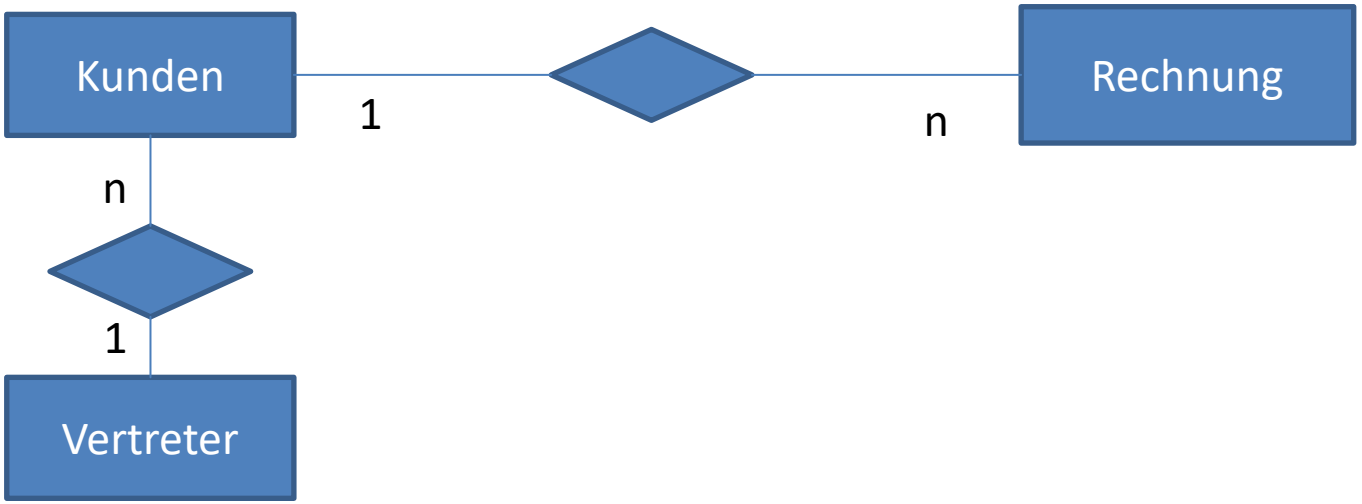
Liefnr	Liefname
1	Alu-GmbH
2	Bike-GmbH
3	Cross-GmbH
4	Wichtek&Sachs
5	Rad AG

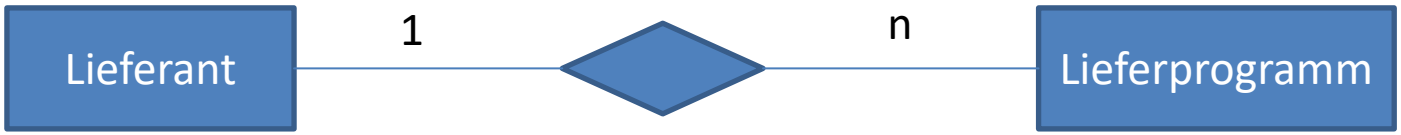
Relation: Lieferprogramm

Liefnr	Teilenr
1	0009
1	0010
2	0009
2	0010
2	0011
3	0003
3	0004
4	0012

Die SELECT-Anweisung

Übungsaufgabe. ER-Darstellung





Die SELECT-Anweisung

Übungsaufgaben.

- Wie sehen die SELECT-Anweisungen, der folgenden Anfragen aus?

- 1. Geben Sie folgende Spalten aus: [kdNr, Rechbetrag] (Tabelle: Rechnung)
-

```
+------+-----------+
| kdnr | rechbetrag |
+------+-----------+
| 2133 |   10600.00 |
| 7533 |   28400.00 |
| 2133 |    1000.00 |
| 8511 |   20900.00 |
+------+-----------+
4 rows in set (0.47 sec)

mysql>
```

- 2. Geben sie alle Rechnungsberträge > 20 000 aus (Tabelle: Rechnung)
-

```
+--------+------+------------+------------+
| rechnr | kdnr | rechdat    | rechbetrag |
+--------+------+------------+------------+
|     82 | 7533 | 2023-07-20 |   28400.00 |
|     84 | 8511 | 2005-08-20 |   20900.00 |
+--------+------+------------+------------+
2 rows in set (0.45 sec)

mysql>
```

- 3. Geben sie die Kunden aus, die eine Rechnung haben (Tabelle „Rechnung „mit Tabelle „Kunden“).
-

```
+------+----------+----------+------+
| kdnr | kdname   | ort      | vnr  |
+------+----------+----------+------+
| 2133 | Meier    | Bayreuth |  224 |
| 7533 | Schmitz  | K├âln     |  115 |
| 8511 | Schneider| Bonn     |  115 |
+------+----------+----------+------+
3 rows in set (0.48 sec)

mysql>
```

Die SELECT-Anweisung

Übungsaufgaben.

- 4. Geben Sie den maximalen Rechnungsbetrag aller kunden aus.

```
+-----------------+
| max(rechbetrag) |
+-----------------+
|        10600.00 |
|        28400.00 |
|        20900.00 |
+-----------------+
3 rows in set (0.00 sec)

mysql>
```

- 5. Welcher Vertreter hat noch keinen Umsatz gemacht?

```
+-----+-------+--------+--------+
| vnr | vname | bezirk | umssum |
+-----+-------+--------+--------+
| 315 | Beck  | Ost    |   0.00 |
+-----+-------+--------+--------+
1 row in set (0.45 sec)

mysql>
```

- 6. Welche Kunden fangen mit „M“ an?

```
+------+---------+----------+------+
| kdnr | kdname  | ort      | vnr  |
+------+---------+----------+------+
| 2133 | Meier   | Bayreuth |  224 |
| 3557 | M?ller  | Bonn     |  115 |
+------+---------+----------+------+
2 rows in set (0.00 sec)

mysql>
```

Die SELECT-Anweisung

Übungsaufgaben.

- 7. Alle Orte nur einmal ausgeben.
-

```
+----------+
| ort      |
+----------+
| Bayreuth |
| Bonn     |
| K├âln     |
+----------+
3 rows in set (0.00 sec)

mysql>
```

- 8. Anzahl der Kunden ausgeben.
-

```
+---------------+
| Anzahl Kunden |
+---------------+
|             4 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

- 9. Anzahl der Kunden pro Vertreter ausgeben.
-

```
+------+------------+
| vnr  | count(vnr) |
+------+------------+
|  115 |          3 |
|  224 |          1 |
+------+------------+
2 rows in set (0.00 sec)

mysql>
```

Die SELECT-Anweisung

Übungsaufgaben.

- 10. Summe über alle Rechnungen
-

```
+-----------------+
| sum(rechbetrag) |
+-----------------+
|        60900.00 |
+-----------------+
1 row in set (0.00 sec)

mysql>
```

- 11. Ausgabe der Rechnung, die ein aktuelleres Datum, wie die Rechnungsnummer 0084 hat.

```
+--------+------+------------+------------+
| rechnr | kdnr | rechdat    | rechbetrag |
+--------+------+------------+------------+
|     82 | 7533 | 2023-07-20 |   28400.00 |
+--------+------+------------+------------+
1 row in set (0.00 sec)

mysql>
```

Die SELECT-Anweisung

Übungsaufgaben.

- Wie sehen die SELECT-Anweisungen, der folgenden Anfragen aus?

- 1. Geben Sie folgende Spalten aus: [kdNr, Rechbetrag] (Tabelle: Rechnung)
- Select kdnr, rechbetrag FROM rechnung;

```
+------+-----------+
| kdnr | rechbetrag |
+------+-----------+
| 2133 |   10600.00 |
| 7533 |   28400.00 |
| 2133 |    1000.00 |
| 8511 |   20900.00 |
+------+-----------+
4 rows in set (0.47 sec)

mysql>
```

- 2. Geben sie alle Rechnungsberträge > 20 000 aus (Tabelle: Rechnung)
- SELECT * FROM rechnung WHERE rechbetrag > 20000;

```
+--------+------+------------+------------+
| rechnr | kdnr | rechdat    | rechbetrag |
+--------+------+------------+------------+
|     82 | 7533 | 2023-07-20 |   28400.00 |
|     84 | 8511 | 2005-08-20 |   20900.00 |
+--------+------+------------+------------+
2 rows in set (0.45 sec)

mysql>
```

- 3. Geben sie die Kunden aus, die eine Rechnung haben (Tabelle „Rechnung „mit Tabelle „Kunden“).
- SELECT * FROM kunden WHERE kdnr IN
  (SELECT kdnr FROM rechnung);

```
+------+----------+----------+------+
| kdnr | kdname   | ort      | vnr  |
+------+----------+----------+------+
| 2133 | Meier    | Bayreuth |  224 |
| 7533 | Schmitz  | K├ôln     |  115 |
| 8511 | Schneider| Bonn     |  115 |
+------+----------+----------+------+
3 rows in set (0.48 sec)

mysql>
```

Die SELECT-Anweisung

Übungsaufgaben.

- 4. Geben Sie den maximalen Rechnungsbetrag aller kunden aus.
- SELECT MAX(rechbetrag) FROM rechnung group by kdnr;

```
+-----------------+
| max(rechbetrag) |
+-----------------+
|        10600.00 |
|        28400.00 |
|        20900.00 |
+-----------------+
3 rows in set (0.00 sec)

mysql>
```

- 5. Welcher Vertreter hat noch keinen Umsatz gemacht?
- SELECT * FROM vertreter WHERE umssum=0;

```
+-----+-------+--------+--------+
| vnr | vname | bezirk | umssum |
+-----+-------+--------+--------+
| 315 | Beck  | Ost    |   0.00 |
+-----+-------+--------+--------+
1 row in set (0.45 sec)

mysql>
```

- 6. Welche Kunden fangen mit „M“ an?
- SELECT * FROM kunden WHERE KdName LIKE ´M%´;

```
+------+---------+----------+------+
| kdnr | kdname  | ort      | vnr  |
+------+---------+----------+------+
| 2133 | Meier   | Bayreuth |  224 |
| 3557 | M?ller  | Bonn     |  115 |
+------+---------+----------+------+
2 rows in set (0.00 sec)

mysql>
```

Die SELECT-Anweisung

Übungsaufgaben.

- 7. Alle Orte nur einmal ausgeben.
- SELECT DISTINCT ort FROM kunden;

```
+----------+
| ort      |
+----------+
| Bayreuth |
| Bonn     |
| K├âln     |
+----------+
3 rows in set (0.00 sec)

mysql>
```

- 8. Anzahl der Kunden ausgeben.
- SELECT count(*) „Anzahl Kunden“ FROM kunden;

```
+---------------+
| Anzahl Kunden |
+---------------+
|             4 |
+---------------+
1 row in set (0.00 sec)

mysql>
```

- 9. Anzahl der Kunden pro Vertreter ausgeben.
- SELECT Vnr, count(Vnr) FROM kunden GROUP BY Vnr;

```
+------+------------+
| vnr  | count(vnr) |
+------+------------+
|  115 |          3 |
|  224 |          1 |
+------+------------+
2 rows in set (0.00 sec)

mysql>
```

Die SELECT-Anweisung

Übungsaufgaben.

- 10. Summe über alle Rechnungen
- SELECT sum(rechbetrag) FROM rechnung;

```
+----------------+
| sum(rechbetrag) |
+----------------+
|         60900.00 |
+----------------+
1 row in set (0.00 sec)

mysql>
```

- 11. Ausgabe der Rechnung, die ein aktuelleres Datum, wie die Rechnungsnummer 0084 hat.
- SELECT * FROM rechnung WHERE rechdat >
  (SELECT rechdat FROM rechnung WHERE rechnr = 0084);

```
+--------+------+------------+------------+
| rechnr | kdnr | rechdat    | rechbetrag |
+--------+------+------------+------------+
|     82 | 7533 | 2023-07-20 |   28400.00 |
+--------+------+------------+------------+
1 row in set (0.00 sec)

mysql>
```