

# Zusammenfassung: Jahr 1

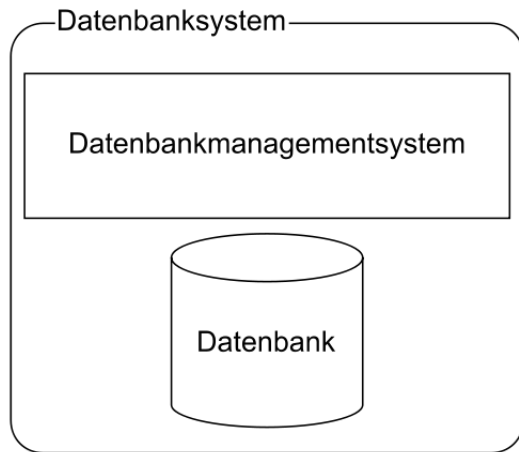
## Inhaltsverzeichnis

<b>1 Lernfeld 6 - Datenbanken</b>	<b>1</b>
1.1 tl;dr - Zusammenfassung der Zusammenfassung . . . . .	1
1.2 Datenbankenmodelle . . . . .	1
1.2.1 Hierarchisches Datenbankmodell . . . . .	1
1.2.2 Relationales Datenbankmodell . . . . .	1
1.2.3 Netzwerkdatenbankmodell . . . . .	1
1.2.4 Objektorientiertes Datenbankmodell . . . . .	1
1.2.5 Objektrationales Datenbankmodell . . . . .	1
1.3 MySQL . . . . .	1
1.3.1 DAS – Database Administration Statements . . . . .	1
1.3.2 DDL – Data Definition Language . . . . .	2
1.3.3 DQL – Data Query Language . . . . .	2
1.3.4 Wildcards . . . . .	3
1.3.5 DML – Data Manipulation Language . . . . .	4
1.3.6 DCL – Data Control Language . . . . .	5
1.3.7 Alternative zu MySQL: MariaDB . . . . .	5

---

# 1 Lernfeld 6 - Datenbanken

Im Lernfeld 6 werden neben Themen wie HTML, PHP und C# auch Datenbanken behandelt. Im Bereich der Datenbanken werden drei Begriffe unterschieden: (1) Datenbanken (DB), (2) Datenbanksystem (DBS) und (3) Datenbankmanagementsystem (DBMS). Die folgende Grafik veranschaulicht den Zusammenhang.



Als DBS wird die Verbindung aus DBMS und der dazugehörigen Datenbank bezeichnet. Das DBMS regelt den Zugriff auf die Datenbank, sodass die Daten – im besten Fall – immer konsistent sind.

## 1.1 tl;dr - Zusammenfassung der Zusammenfassung

## 1.2 Datenbankenmodelle

### 1.2.1 Hierarchisches Datenbankmodell

### 1.2.2 Relationales Datenbankmodell

### 1.2.3 Netzwerkdatenbankmodell

### 1.2.4 Objektorientiertes Datenbankmodell

### 1.2.5 Objektrationales Datenbankmodell

## 1.3 MySQL

Bei SQL handelt es sich um einen Standard zur Abfrage von Datenbanken. SQL wird klassisch in vier Gebiete unterteilt: (1) Data Definition Language, (2) Data Manipulation Language, (3) Data Query Language und (4) Data Control Language. In den folgenden Abschnitten werden die einzelnen Gebiete und deren Befehle anhand von Beispielen erklärt.

Einige Befehle, wie etwa `show` werden nach dem Handbuch nicht unter eines der vier Gebiete gefasst. Das Handbuch verwendet eine andere Klassifizierung der Befehle. Um nicht willkürlich eine eigene Klassifizierung vorzunehmen, wird neben den vier genannten Gebieten versucht, der offiziellen Klassifizierung Rechnung zu tragen.

Wenn man mal die Syntax vergisst oder nicht weiß, was ein bestimmter Befehl macht, kann man sich dies unter MySQL mit dem Kommando `help <command>` anzeigen lassen.

Weitere Informationen zu den verschiedenen Befehlen lassen sich auch im Handbuch unter <http://dev.mysql.com/doc/refman/5.6/en/index.html> nachlesen.

---

### 1.3.1 DAS – Database Administration Statements

Wie schon erwähnt, wird im Handbuch eine feinere Klassifizierung verwendet, als die klassische Unterteilung in vier Gebiete. Beispielsweise ist die Befehle der Data Control Language unter den Bereich Database Administration Statements (DAS) gefasst. Weil zu den Database Administration Statements auch der Befehl `show` fällt, beginnt diese Kapitel mit einem Abschnitt über DAS.<sup>1</sup>

Wenn man sich mit einem MySQL-Server verbunden hat, möchte man in der Regel wissen, welche Datenbanken auf diesem zur Verfügung stehen. Einen Überblick darüber lässt sich mit dem Befehl `show databases;` bekommen. Dasselbe gilt für Tabellen einer Datenbank. Mit dem Befehl `show tables;` zeigt einem MySQL an, welche Tabelle in einer Datenbank enthalten sind.

```
mysql> show databases;
```

Database
information_schema
geo
mysql
performance_schema

```
mysql> show tables;
```

Tables_in_geo
fluss
kontinent
land
ort
stadtfluss

Listing 1: DDL: show-Befehl

In diesem Fall benutzen wir die Datenbank, welche uns durch das Skript `geo.sql` zur Verfügung gestellt wird. Um die Datenbank in MySQL einzuspielen nutzen wir den Befehl `source geo.sql;` und wechseln mit `use geo;` in die Datenbank.

### 1.3.2 DDL – Data Definition Language

Beispiele für DDL-Befehle:

- `source /path/to/geo.sql;`
- `drop table;`
- `alter`
- `create`

### 1.3.3 DQL – Data Query Language

Enthält nur den Befehl „`select`“. Diesem sind so viele Optionen zugeordnet, dass für ihn die eigene Kategorie „DQL“ vorgesehen ist. Es können nicht nur einzelne Spalten oder alle Spalten ausgelesen werden, sondern auch Funktionen auf die Spalten angewendet werden.

---

<sup>1</sup>Beachte, dass laut Herr Abu Shebika sowohl `show` als auch `use` und `describe` unter DDL fallen.

- count()
- avg()
- sum()
- distinct()

# Alle Kontinente

```
mysql> SELECT * FROM kontinent;
```

# Die Anzahl der Fluesse, Ueberschrift der Anzeige : "Anz. Fluesse"

```
mysql> SELECT COUNT(name) AS "Anz. Fluesse" FROM fluss;
```

# Die durchschnittliche Flusslaenge fuer Fluesse zum Atlantischen Ozean

```
mysql> SELECT AVG(laenge) FROM fluss WHERE meer LIKE 'atlantischer ozean';
```

```
mysql> SELECT AVG(laenge) FROM fluss WHERE meer='atlantischer ozean';
```

# Eine Liste aller Meere (Liste heisst keine Mehrfachnennungen)

```
mysql> SELECT meer AS "Keine Mehrfachnennung" FROM fluss WHERE meer \
      IS NOT null GROUP BY meer;
```

```
mysql> SELECT DISTINCT(meer) AS "Keine Mehrfachnennung" FROM fluss \
      WHERE meer IS NOT null;
```

# Anzahl aller Einwohner in Afrika

```
mysql> SELECT SUM(einwohner) FROM land WHERE knr LIKE 'AF';
```

# Der 6te bis 15te Ort in Grossbritannien nach Ortsnamen sortiert

```
mysql> SELECT * FROM ort WHERE lnr LIKE 'GB' ORDER BY name LIMIT 5,10;
```

Listing 2: DQL: select-Befehle

Darüber hinaus lassen sich mit Hilfe von Subselects Abfragen miteinander verbinden. Die folgenden Befehle zeigen Beispiele für die Verwendung von Subselects:

# Zeige alle Orte die mehr als Durchschnitt der Einwohner haben

# und ueber dem 50.ten Breitengrad liegen

```
select name from ort where einwohner > (select avg(einwohner) \
      from ort) and laenge > 50;
```

# Alle Staedte die mit 'a' beginnen, deren Laengengrad negativ ist.

```
select name from ort where name like "a%" and laenge < 0;
```

# Alle Landesteile mit hoechster Einwohnerzahl, die mit 'n' beginnen.

```
select name from land where einwohner = (select max(einwohner) \
      from land) and name like "n%";
```

# Die 3 kleinsten laender der EU.

```
select * from land where knr like "eu" order by flaeche limit 0,3;
```

# Zeige alle Fluesse, die 'a' nicht am Anfang o. Ende haben und unter

# der Durchschnittslaenge liegen.

```
select name from fluss where not (name like "a%") and not \
      (name like "%a") and laenge < (select avg(laenge) from fluss);
```

Listing 3: DQL: subselect-Befehle

### 1.3.4 Wildcards

Abfragen unter MySQL können auch mit Hilfe von Wildcards formuliert werden. Wildcards sind Platzhalter für ein beliebiges Zeichen und beliebig viele beliebige Zeichen. Die folgende Auflistung gibt einen Überblick der Wildcards in MySQL. Die Beispielausgabe zeigt anschließend, wie sich Wildcards verwenden lassen, um Einträge mit einem bestimmten Muster abzufragen.

% : beliebige Zeichen

\_ : für genau ein Zeichen

a-c %: Zeichenkette, die mit a,b oder c beginnt

!a-c %: Zeichenkette, die nicht a,b oder c beginnt

#Beispielausgabe

```
mysql> SELECT * FROM fluss WHERE name LIKE "M\_k\%";
```

FNR	Name	Meer	Laenge
MEK	Mekong	Suedchinesisches Meer	4500
MSC	Mokscha	NULL	656

Listing 4: Wildcards

Wildcards beziehen sich nur auf Tabelleninhalte und nicht auf ihre Struktur. Unterschied bestehen auch zwischen den Operatoren `like` und `=`. Nur der Operator `like` beherrscht Wildcards, wohingegen `=` die Eingabe als String interpretiert, d.h. in der Tabelle bspw. nach dem Eintrag „M\_k%“ sucht und ihn – in unserem Fall – nicht findet.

### 1.3.5 DML – Data Manipulation Language

Wichtig ist an dieser Stelle, dass statt mit Namen mit den Primary Keys gearbeitet wird. Denn im Gegensatz zu Namen müssen die Primary Keys eindeutig sein. Mit dem Befehl `describe <tabelle>`; lässt sich herauszufinden, welche Spalte den Primary Key darstellt in der Tabelle `<tabelle>` darstellt.

```
mysql> describe land;
```

Field	Type	Null	Key	Default	Extra
LNR	varchar(4)	NO	PRI	NULL	
Name	varchar(50)	YES		NULL	
KNR	varchar(3)	YES		NULL	
Einwohner	double(20,2)	YES		NULL	
Flaeche	double(53,0)	YES		NULL	
HauptONR	varchar(6)	YES		NULL	

Listing 5: DDL: Primary Key ermitteln

Wir sehen anhand des Outputs, dass das Feld LNR der Primary Key der Tabelle land ist. Daher werden wir für die folgenden Befehle auf das Feld LNR zurückgreifen. Es gilt: was weg ist, ist im Zweifelsfall weg. Es sollte also im Betrieb vorsichtig mit den Datenbeständen umgegangen werden. Um den Wert von LNR für bspw. Ägypten herauszufinden, lässt sich der Befehl `SELECT lnr FROM land WHERE bane LIKE "%gypt%"` nutzen.

---

## #Aufgaben zum Thema DML:

```
# a)      Aktualisieren der Einwohnerzahl fuer Aegypten
#         heute : 80.471.869 Einwohner
update land set einwohner=80.47 where lnr="ET";

# b)      Korrektur der Daten von Zaire in den betroffenen Tabellen:
#         heute : Staatsname Kongo, Einwohnerzahl 71.712.867,
#         Flaeche 2.345.000 qkm
update land set name="Kongo", einwohner=71.71, flaeche=234500 \
      where lnr="ZRE";

# c)      Entfernen aller Daten von Vatikanstadt, da wegen zu kleiner
#         Werte zu ungenau
delete from land where name like "SCV";

# d)      Korrektur von Umlauten: in ae, oe und ue in den Tabellen
#         Kontinent, Land, Fluss
update land set name=REPLACE(name,'<umlaut>*', 'ae');
...

# e)      Aendern der Flusslaenge von NULL zu:
#         Baro 306km, Han 498km, Weisser Nil 3800km
update fluss set laenge=306 where FNR="BAR";
...

*die listing-Umgebung mag keine Umlaut, daher hier ein "ae" einsetzen
```

Listing 6: DML: Beispiele

### 1.3.6 DCL – Data Control Language

Weil Excel keinerlei Rechtemanagement implementiert, handelt es sich dabei auch nicht um eine Datenbank. Das Rechtemanagement, welches eine Datenbank ausmacht und mit dem einzelnen Usern feingranular Zugriffe auf bestimmte Teile gewährt werden kann, wird als Data Control Language bezeichnet.

Unter MySQL fallen vor allem die Kommandos `grant` und `revoke` darunter. Mit `grant` lassen sich Rechte zuweisen und mit `revoke` entziehen.

- `grant`
- `revoke`

### 1.3.7 Alternative zu MySQL: MariaDB

Bei SQL handelt es sich um einen Standard zur Abfrage von Datenbanken. Wie die Datenbank darunter aussieht, ist von SQL unabhängig. Das bedeutet, dass dieselben Befehle, die wir oben gelernt haben, auch auf dieselbe Weise unter MariaDB verwendet werden können.

Was ist MariaDB? Und worin bestehen die Unterschiede zu MySQL? MariaDB ist in erster Linie ein Fork von MySQL, der 2009 initiiert wurde nachdem MySQL von Oracle übernommen wurde. Dadurch kann sichergestellt werden, dass MariaDB auch in Zukunft frei unter der GPL(2) verwendet werden kann.

---

Bis zur Version 5.5 bestehen vom Funktionsumfang her keine Unterschiede zwischen MySQL und MariaDB. Nach dem Release von Version 5.5 wurde eine Version 10.0 von MariaDB angekündigt. Die Nummerierung soll verdeutlichen, dass die nächste Version von MariaDB nicht mehr alle Features von MySQL 5.6 abdecken wird.

Im [Wikipedia-Artikel](#) zu MariaDB werden einige prominente Nutzer aufgelistet. Darunter befinden sich zwischen bekannten Distributionen wie RHEL, Gentoo und ArchLinux auch Größen wie Google, Mozilla und die Wikipedia Foundation.