

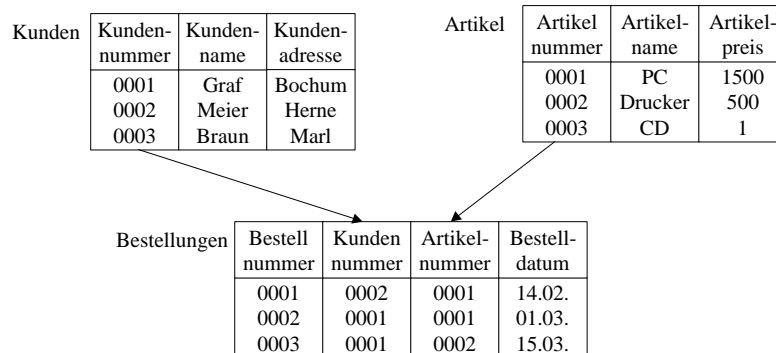
Typen von Datenbank-Management-Systemen

- Hierarchische DBMS:** - Datenstrukturen über Zeiger verknüpft
- Anordnung z.B. in „Baumstrukturen“
- Netzwerkmodell:** - wie oben: Zeigerstrukturen
- aber beliebige Verknüpfungen („Netz“)
- Relationale DBMS:** - Daten in Tabellen angeordnet weit verbreitet
- Datenmanipulationssprache („SQL“)
- Wissensbanksysteme:** - Daten in Tabellen, aber „logisch“ verknüpft
- Deklarative Datenmanipulationssprache
- Objektorientierte DBMS:** - Daten in komplexen Objektstrukturen abgelegt
- navigierende Datenmanipulationssprache
- Objektrelationale DBMS:** - Verbindung von relationalen DBMS und Objektorientierung („SQL-99“)

Beispiel: Relationale Datenbank

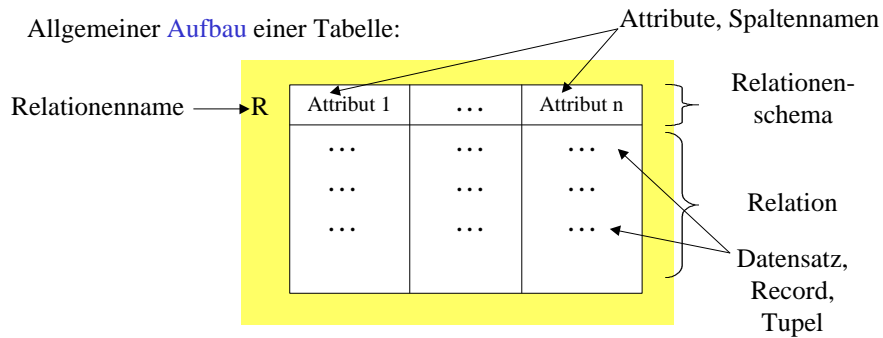
Daten einer **Kundendatenbank**:

Relationale Datenbank =
Ansammlung von **Tabellen**



Begriffe im Relationenmodell

Allgemeiner Aufbau einer Tabelle:



Relationenschema = gibt die prinzipielle Struktur der Tabelle vor

Datensatz (Record, Tupel) = eine Zeile der Tabelle

Relation = alle Tupel

➡ hinter der Tabelle steht die mathematische Idee der „**Relation**“

Quelle:
Heuer/
Saake

Beispiel für eine Abfrageoperation mit SQL

Frage: Was hat Herr Graf bisher bestellt und wann war das ?

Befehl: `select` Kundenname, Artikelnummer, Bestelldatum
`from` Kunden, Bestellungen
`where` Kundenname = ‚Graf‘ `and`
 Kunden.Kundennummer = Bestellungen.Kundennummer

Ergebnis:

Kundenname	Artikelnummer	Bestelldatum
Graf	0001	01.03
Graf	0002	15.03.

RDBMS-Produktnamen

Weit verbreitete relationale Datenbank-Management-Systeme:

MySQL	kostenlos für nichtkommerzielle Anwendungen	
DB2	IBM Datenbanksystem	
Oracle	am weitesten verbreitet, derzeit Marktführer	
INGRES	entstand aus University-INGRES an der Berkley-University	
dBASE	Quasi-Standard im PC-Bereich	} nicht alle Merkmale eines DBMS vorhanden, z.B.: keine Mehrbenutzer-Systeme, keine Sichten...
MS-Access	Microsoft-Office-Produkt	

MySQL-Entstehungsgeschichte

1979: Datenbank-Tool UNIREG der schwedischen Firma TcX
 Programmierer: Michael Widenius

1994: MySQL entsteht als SQL-basierter Server für Web-Anwendungen

1996: MySQL als Binär-Distribution für Linux und Solaris frei verfügbar

inzwischen: Quell-Distribution verfügbar, Portierung auf Windows erfolgte
 aber: MySQL ist **kein** Open-Source-Produkt

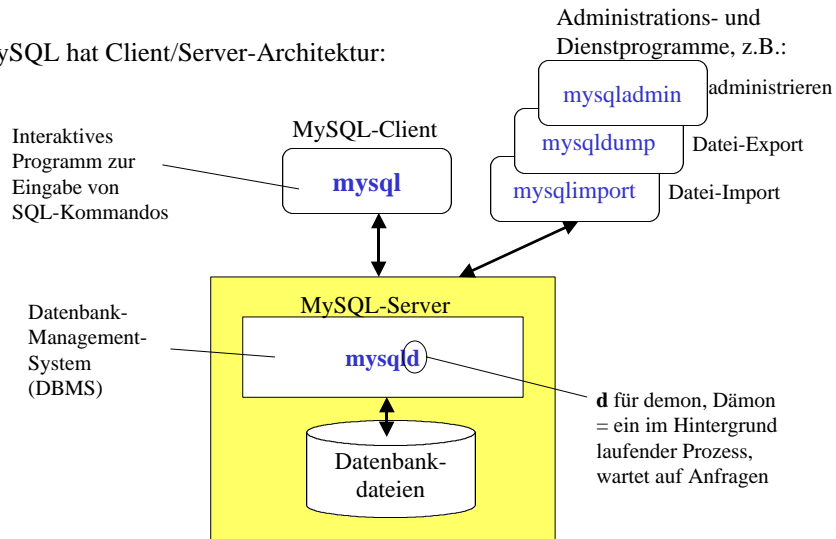
Lizenzbedingungen:

- ältere MySQL-Versionen kostenlos verfügbar und frei nutzbar
- aktueller **MySQL-Server** bei kommerziellem Einsatz lizenzpflichtig

Quelle:
 Dubois,
 „MySQL“

MySQL-Komponenten

MySQL hat Client/Server-Architektur:



Wie startet man MySQL ?

- 1) MySQL-Server starten, falls dieser nicht schon läuft

z.B.: `.. \mysqld <Eingabe>`

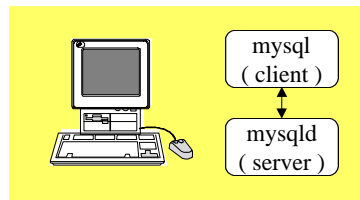
Hier steht der
aktuelle Ver-
zeichnisname, z.B.:
c:\mysql\bin

meist: Server wird beim Systemstart automatisch gestartet

abhängig von der Dienstekfiguration des Betriebssystems

- 2a) MySQL-Client auf dem selben Rechner starten

z.B.: `... \mysql <Eingabe>`



Beide Programme laufen auf dem selben Rechner

Wie startet man MySQL (2) ?

2b) falls MySQL-Server auf einem Server und Client auf eigenem Computer:

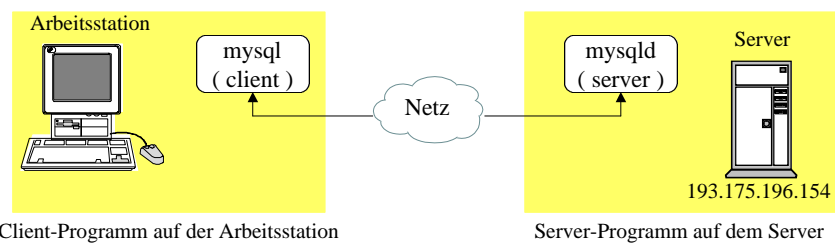
z.B.: `.\mysql -h host_name <Eingabe>`

`-h` = Option zur Eingabe des Host-Namens

`host_name` = symbolischer Name oder IP-Adresse

z.B.: `mysqlserver.fh-gelsenkirchen.de`

oder: `193.175.196.154`



Wie startet man MySQL (3) ?

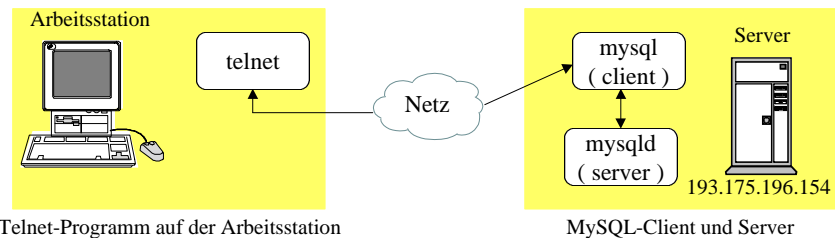
2c) falls MySQL-Server-Programm und -Client auf anderem Computer:

(a) Telnet-Verbindung zum MySQL-Server aufbauen

z.B.: `telnet host_name <Eingabe>`

danach: Nutzererkennung und Passwort für den anderen Computer

(b) Client starten (siehe oben)



Wie startet man MySQL (4) ?

Anmeldung im Mehrbenutzerbetrieb:

z.B.: `...\mysql -u user_name -p <Eingabe>`

`-u` = Option zur Eingabe des Nutzernamens

`user_name` = symbolischer Name, z.B.: `gast`

`-p` = Option zur Eingabe des Passwort, das Passwort wird dann von mysql abgefragt: `Enter Password: ****`

im Praktikum: Nutzer sind `pc01`, `pc02`, `pc03` ...

Also z.B.: `>mysql -u pc01 -p <Eingabe>`

→ jeder Nutzer startet einen eigenen MySQL-Client-Prozess

Bedienung des mysql-Clients

Der Client (Monitor) meldet sich mit:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.22.34
Type ,help` for help
Mysql>
```

Danach können Kommandos eingegeben werden, z.B.:

```
mysql> status;
```

Kommando wird von `mysql` zu `mysqld` geschickt und dort `bearbeitet`.

Ergebnis wird von `mysqld` zu `mysql` geschickt und dort `angezeigt`.

Wichtig: `Semikolon` nicht vergessen (schließt das Kommando ab) !

Hilfe anzeigen

Anzeigen der Online-Hilfe:

```
mysql> help;

help      (\h)    Display this text
?         (\h)    Synonym for `help`
clear     (\c)    Clear command
connect   (\r)    Reconnect to the server.
edit      (\e)    Edit command with $EDITOR
exit      (\q)    Exit mysql. Same as quit
go        (\g)    Send command to mysql server
print     (\p)    Print current command
quit      (\q)    Quit mysql
source    (\.)    Execute a SQL script file w. file name as argument
status    (\s)    Get status information from the server
use       (\u)    Use another database. Takes database name as argument
...

```

Daten einspeichern und auslesen

Überblick über die Kommandoabfolge:

- 1) Eine Datenbank anlegen `create database ...`
- 2) Die Datenbank auswählen `use ...`
- 3) Eine Tabelle anlegen `create table ...`
- 4) Datensätze einfügen `insert ...`
- 5) Datensätze auslesen `select ...`
- 6) Datensätze löschen `delete ...`

Nützliche Kommandos zur Information: `show ...`

z.B.: `show databases` zeigt alle vorhandenen Datenbanken an
`show tables` zeigt alle Tabellen der aktuellen Datenbank an
`describe tabellen_name` zeigt die Attribute der Tabelle

Datenbank anlegen und auswählen

Anlegen der Datenbank:

Syntax: `create database datenbank_name;`

Beispiel: `create database webshop;`

Ergebnismeldung: `Query OK, 1 rows affected (0.07 sec)`

(falls genügend Rechte vorhanden, sonst Fehlermeldung)

Auswählen der Datenbank:

Syntax: `use datenbank_name;`

Beispiel: `use webshop;`

Ergebnismeldung: `Database changed`

Alle Befehle ohne Datenbanknamen nutzen nun diese Datenbank

Tabelle anlegen

Eine Tabelle anlegen:

Vereinfachte Syntax:

↙ Es gibt noch
weitere Optionen,
die hier nicht
aufgeführt sind

```
create table tabellenname
(
  attribut_1 typ,
  attribut_2 typ, ...
);
```

Das `attribut` stellt den „Spaltennamen“ der Tabelle dar und hat einen `Typ`

Beispiel:

```
create table kunden
(
  kundennummer int,
  kundenname varchar(10),
  kundenadresse varchar(10)
);
```

Wichtige Typen: `int` = Integer, ganze Zahlen

`varchar(n)` = Zeichenkette variabler Länge, maximal `n` Zeichen

Daten eingeben

Einen Datensatz (eine Zeile) in eine Tabelle einfügen:

Vereinfachte Syntax: `insert into tabellename
(attribut1, attribut2, ...)
values (wert1, wert2, ...);`

In dieser Form: Liste von Attributen und Liste von zugehörigen Werten.

Beispiel: `insert into kunden
(kundennummer, kundenname, kundenadresse)
values (4, "Baumann", "Gladbeck");`

Da **alle** Spalten beschrieben werden, ist **Kurzschreibweise** möglich:

```
insert into kunden
values ( 4, "Baumann", "Gladbeck");
```

Daten ausgeben

Alle Daten der Tabelle ausgeben:

Syntax: `select * from tabellename;`

Beispiel: `select * from kunden;`

Ergebnis:

```
-----+-----+-----+
| kundennummer | kundenname | kundenadresse |
-----+-----+-----+
|           1 | Graf       | Bochum        |
|           2 | Meier      | Herne         |
|           3 | Braun      | Marl          |
|           4 | Baumann    | Gladbeck      |
-----+-----+-----+
4 rows in set (0.00 sec)
```

Datenbanken und Tabellen anzeigen

Befehl: `show databases;`

Ergebnis:

```
+-----+
| Database |
+-----+
| webshop |
+-----+
1 row in set (0.01 sec)
```

Befehl: `show tables;`

Ergebnis:

```
+-----+
| Tables in webshop |
+-----+
| kunden            |
+-----+
1 row in set (0.00 sec)
```

Datensätze löschen

Datensätze (Zeilen) aus einer Tabelle löschen:

Syntax: `delete from tabellenname where bedingung;`

Beispiel: `delete from kunden where
kundenname = "Graf";`

Ergebnis: Query OK, 1 rows affected (0.07 sec)

Test: `select * from kunden;`

Ergebnis:

```
+-----+-----+-----+
| kundennummer | kundenname | kundenadresse |
+-----+-----+-----+
|          2 | Meier      | Herne         |
|          3 | Braun     | Marl          |
|          4 | Baumann   | Gladbeck     |
+-----+-----+-----+
3 rows in set (0.00 sec)
```