Gliederung

1. Die Bedeutung des Datenmanagements
2. Datenbank-Architektur
3. Modellierung und Entwurf von DB-Systemen
4. Relationale Algebra und Normalisierung
5. **Definition und Abfrage von Datenbank-Systemen**
6. Dateiorganisation und Zugriffsstrukturen
7. Optimierung von Anfragen
8. Transaktionen
9. Wiederherstellung und Datensicherung

5. Definition und Abfrage von Datenbank-Systemen

1. Structured Query Language (SQL)
2. Data Control Language (DCL)
3. Data Manupulation Language (DML)
    1. *Abfrage anhand eines Selektionskriteriums*
    2. *Aggregatfunktionen*
    3. *Operationen auf Wertebereichen*
    4. *Verbundoperationen*
    5. *Mengen und Multimengen*

Weiterführende Webseiten:

- Offizielle MySQL Reference: http://dev.mysql.com/doc
- SQL Tutorial mit Übungen: http://www.schulserver.hessen.de/darmstadt/lichtenberg/SQLTutorial/home.html
- SQL Tutorial: http://sql.1keydata.com/de/

SQL (Structured Query Language)

- ist die **Norm-Datenbanksprache** für relationale Datenbanksysteme
- ist **Werkzeug zum Umsetzen des logischen Entwurfs** (Relationenmodell) in das Datenbankmodell eines konkreten Datenbanksystems
- Besteht aus den Teilsprachen DDL (**Data Definition Language**), DML (**Data Manipulation Language**) und DCL (**Data Control Language**)
- Ist sehr deskriptiv aufgebaut

3

5.2 Data Control Language (DCL)

DCL (Data Control Language)

- Teil der Datenbanksprache zur Vergabe und zum Entzug von Berechtigungen
- Sichert als „*Datenüberwachungssprache*“ die data security
- Ist stark mit der DDL verknüpft
- Beispiele: grant, revoke

Beispiel für die Anweisung `grant`

- Vergabe von Rechten (Privilegien):

```
GRANT Privileg [(Spalte, ...)]
[, Privileg [(Spalte, ...)] ...]
ON Tabelle TO Benutzer [IDENTIFIED BY 'Passwort']
[, Benutzer [IDENTIFIED BY 'Passwort'] ...]
[ WITH GRANT OPTION ]
```

- Beispiel:

```
GRANT CREATE ,INDEX ,ALTER
ON `testdb` . * TO 'testuser'@'localhost';
```

DML (Data Manipulation Language)

- Teil der Datenbanksprache zum Lesen, Schreiben, Ändern und Löschen von Daten
- Wird auch bei reinen Abfragen (ohne Veränderung der Daten) eingesetzt
- Beispiele: `select, delete, join, where`

Abfrage anhand eines Selektionskriteriums

```
select Vorname, Nachname, PLZ
from Person
where PLZ = `39106´
```

- **Relation** wird hinter **`from`** angegeben
- **Selektionsbedingung** wird hinter **`where`** angegeben erweiterbar mit **boolschen Operatoren** wie **`and`**, **`or`**, **`not`**
- **Projektion** wird hinter **`select`** angegeben

Vorname	Nachname	PLZ
Gunter	Saake	39106

Übung 1:

- Lösen Sie die Aufgaben der Übung 1
- Überprüfen Sie Ihre Anfrage in mySQL

Aggregatfunktionen

- **`sum()`** summiert die Werte einer Spalte
- **`count()`** zählt die Anzahl der von Null verschiedenen Einträge einer Spalte
- **`avg()`** gibt den Durchschnittswert der Werte einer Spalte aus
- **`min()` & `max()`** geben den kleinsten bzw. größten Wert der Spalte aus
- `Funktion` **`as`** `Spaltenname` benennt die Ergebnisspalte neu

Beispiel: Aggregatfunktionen

```
select count(PANr) as Anzahl_Mitarbeiter
from Person


select sum(Preis) as Ausgaben_nach_1990
from Buch_Version
where Jahr > ´1990´
```


Übung 2:

- Lösen Sie die Aufgaben der Übung 2
- Überprüfen Sie Ihre Anfrage in mySQL

Operationen auf Wertebereichen

- Ausgabe der Werte nach Umrechnung (*, /, +, -)
- Beispiel: Umrechnung des Europreises in Dollar

```
select ISBN, Preis / 0.89 as Dollar
from Buch_Versionen
```

ISBN	Dollar
3-89319-175-5	54,86
0-8053-1753-8	50,24
0-201-53771-0	60,73

Übung 3:

- Lösen Sie die Aufgaben der Übung 3
- Überprüfen Sie Ihre Anfrage in mySQL

Äußere Verbunde

LINKS

A	B
1	2
2	3

RECHTS

B	C
3	4
4	5

NATURAL JOIN

A	B	C
2	3	4

OUTER

A	B	C
1	2	⊥
2	3	4
⊥	4	5

LEFT

A	B	C
1	2	⊥
2	3	4

RIGHT

A	B	C
2	3	4
⊥	4	5

Der Verbund

- Verbundbedingung über **gleich lautende Attribute**
- Beispiel

```
select *
from Person, Pers_Telefon
where Person.PANr = Pers_Telefon.PANr
```

PANr	Vorname	Nach-name	PLZ	Ort	Strasse	HNr	Geb.Datum	PANr	Telefon
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.1958	4711	038203-12230
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.1958	4711	0381-498-3401
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.1958	4711	0381-498-3427
5588	Gunter	Saake	39106	MD	STS	55	05.10.1960	5588	0391-345677
5588	Gunter	Saake	39106	MD	STS	55	05.10.1960	5588	0391-5592-3800
9999	Christa	Loeser	69121	HD	TS	38	10.05.1969	9999	06221-400177

Der natürliche Verbund

- Natürlicher Verbund: Alternative 1

```
select *
from Buch_Exemplare, Ausleihe
where Buch_Exemplare.Inventarnr = Ausleihe.Inventarnr
```

- Natürlicher Verbund: Alternative 2

```
select *
from Buch_Exemplare join Ausleihe on
     Buch_Exemplare.Inventarnr = Ausleihe.Inventarnr
```


Übung 4:

- Lösen Sie die Aufgaben der Übung 4
- Überprüfen Sie Ihre Anfrage in mySQL

Multimenge

- Beispiel

```
select Fachbereich
from Mitarbeiter
```

Fachbereich
Informatik
Informatik
Mathematik
Informatik
Informatik
Linguistik

- Gründe
  - ist **einfacher zu implementieren**
  - wird manchmal **bei Aggregatfunktionen** benötigt, die die Werte dieser Spalte etwa summieren sollen (Beispiel: Gesamtsumme der Gehälter in der Mitarbeitertabelle)

Menge

- „Normale“ Mengensemantik:

```
select distinct (Fachbereich)
from Mitarbeiter
```

- … oder alternativ

```
select Fachbereich
from Mitarbeiter
group by Fachbereich
```

Gruppierungen mit `group by`

- Für Statistische Auswertungen

```
select Fachbereich, sum(Gehalt)
from Mitarbeiter
group by Fachbereich
```

Fachbereich	Sum(Gehalt)
Informatik	15350
Mathemati	2600
k	750
Linguisti	
k	

Gruppenwahl mit dem „`having`“-Befehl

```
select Fachbereich, sum(Gehalt)
from Mitarbeiter
group by Fachbereich
having sum(gehalt)>2500
```

Fachbereich	Sum(Gehalt)
Informatik	15350
Mathematik	2600

DHBW Duale Hochschule Baden-Württemberg Lörrach

Kombination von „where“ und „group by“

```
select Fachbereich, sum(Gehalt)
from Mitarbeiter
where Gehalt > 1000
group by Fachbereich
```

Fachbereich	Sum(Gehalt)
Informatik	14800
Mathematik	2600

DHBW Duale Hochschule Baden-Württemberg Lörrach

Übung 5:

- Lösen Sie die Aufgaben der Übung 5
- Überprüfen Sie Ihre Anfrage in mySQL

Geschachtelte Anfragen

- Beispiel: Differenzbildung

```
select *
from Person
where Person.PANr not in (
                select PANr from Pers_Telefon )
```

PANr	Vorname	Nachname	PLZ	Ort	Strasse	HNr	Geb.Datum
6834	Michael	Korn	39104	MD	BS	41	24.09.1974
7754	Andreas	Möller	18209	DBR	RS	31	25.02.1976
8832	Tamara	Jagellovsk	38106	BS	GS	12	11.11.1973
9912	Antje	Hellhof	18059	HRO	AES	21	04.04.1970

Verzahnt geschachtelte Anfragen

- es wird in der inneren Anfrage ein Relationen-/Tupelvariablen-Name aus dem **from**-Teil der äußeren Anfrage verwendet
- Beispiel

```
select Nachname from Person
where 1.0 in (
        select Note from Prueft
        where PANr = Person.PANr )
```

„Gebe die Nachnamen der Professoren aus, die schon einmal eine 1.0 in einer Prüfung gegeben haben“

15

Abarbeitung der verzahnt geschachtelten Anfrage

- in der äußeren Anfrage wird das erste Personen-Tupel untersucht (`PANr = 4711`)
- diese wird in der inneren Anfrage eingesetzt und ausgewertet

```
select Note from Prueft
where PANr = 4711
```

- Ergebnis ist die Werteliste `(2.0, 2.3)`
- das Ergebnis der inneren Anfrage wird in die äußere eingesetzt
- `1.0 in (2.0, 2.3)` nimmt den Wert `false` an → Prüfer mit `PANr = 4711` wird nicht berücksichtigt
- Anfrage wird mit allen Personen-Tupel wiederholt

Übung 6:

- Lösen Sie die Aufgaben der Übung 6
- Überprüfen Sie Ihre Anfrage in mySQL

Tupelvariablen und Relationennamen

- Beispiel: Falsches **select**

```
select ISBN, Titel, Stichwort
from Buecher, Buch_Stichwort
where Buecher.ISBN = Buch_Stichwort.ISBN
```


- Beispiel: Falsches **select**

```
select ISBN, Titel, Stichwort
from Buecher, Buch_Stichwort
where Buecher.ISBN = Buch_Stichwort.ISBN
```

(falsch!)

- Beispiel: Richtiges **select**

```
select Buecher.ISBN, Titel, Stichwort
from Buecher, Buch_Stichwort
where Buecher.ISBN = Buch_Stichwort.ISBN
```

DHBW Duale Hochschule Baden-Württemberg Lörrach

Bedingung hinter `where`

- **Verbundbedingung**

```
relation1.attribut = relation2.attribut
```

- **Bereichsselektion**

```
attribut between konstante1 and konstante2
```

ist Abkürzung für

```
attribut ≥ konstantel and attribut ≤ konstante2
```

- **Beispiel**

```
select Matrikelnummer from Prüft
where Note between 1.0 and 2.0
```


DHBW Duale Hochschule Baden-Württemberg Lörrach

- **Ungewissheitsselektion**

```
attribut like spezialkonstante
```

  - einfache Art der **Mustererkennung** in Strings (Teilzeichenketten)
  - Spezialkonstante kann die **Sondersymbole** `%´ und `- ´ beinhalten
    - `%´ steht für kein oder beliebig viele Zeichen
    - `-´ steht für genau ein Zeichen
  - Beispiel

```
select Vorname, Nachname from Person
where Nachname like 'H%'
```

18

Sortieren

- `order by asc` (aufsteigend) oder `desc` (absteigend)
- Beispiel: aufsteigend nach Noten, danach nach Matrikelnummer sortieren

```
select Matrikelnummer, Note
from Prüft
where V_Bezeichnung = `Datenbanken I´
order by Note, Matrikelnummer asc
```


Übung 7:

- Lösen Sie die Aufgaben der Übung 7
- Überprüfen Sie Ihre Anfrage in mySQL

Ändern von Daten

- Gegeben sei die rechts stehende Tabelle
- *„erhöhe das Gehalt aller Angestellten, die weniger als 5000 verdienen, um 1000“*

Name	Gehalt
Meyer	3000
Schulz	3500
Bond	7200
Schulz	4400

```
update Angestellte
set Gehalt = Gehalt + 1000
where Gehalt < 5000
```


Einfügen von Daten

- `insert into Buecher (ISBN, Titel)`

  `values (`3-89319-175-5´,`Wissensbanken´)`
- die Werte des fehlenden Attributes `Verlagsname` werden auf **`null`** gesetzt
- alle Werte werden besetzt

```
insert into Buecher
values (`3-89319-175-5´,`Wissensbanken´,`Addison-Wesley´)
```

Beispiel

- die folgende Anweisung fügt alle Lieferanten (aus der Relation `Lieferant`) als Kunden in die Relation `Kunde` mit dem vorläufigen Kontostand 0 ein

```
insert into Kunde
      ( select LName, LAdr, 0
        from Lieferant )
```


Beipiele: Löschen eines Tupels

- Löschen **eines Tupels** in der `Ausleihe`-Relation

```
delete from Ausleihe
where Inventarnr = 140
```

- Löschen **mehrerer Tupel** in der `Ausleihe`-Relation

```
delete from Ausleihe
where PANr = 7754
```

Übung 8:

- Lösen Sie die Aufgaben der Übung 8
- Überprüfen Sie Ihre Anfrage in mySQL

DDL (Data Definition Language)

- Teil der Datenbanksprache zum Beschreiben, Ändern oder Entfernen von Datenstrukturen
- Notwendig zur Erstellung/Änderung ganzer Datenbanken oder Tabellen
- Beispiele: `create, drop, alter`

Beispiele für die Anweisung `create table`

- Einfaches Beispiel

```
create table Bücher (
   ISBN char(10) not null,
   Titel varchar(200),
   Verlagsname varchar(30))
```

- Beispiel mit Primär- und Fremdschlüssel

```
create table Bücher (
   ISBN char(10),
   Titel varchar(200),
   Verlagsname varchar(30),
   primary key (ISBN),
   foreign key (Verlagsname)
        references Verlage (Verlagsname))
```

23

`primary key`

- definiert den **Primärschlüssel**
- es können **Attributmengen** angegeben werden

`foreign key`

- definiert einen oder mehrere **Fremdschlüssel**
- es können **Attributmengen** angegeben werden
- hinter **`references`** werden die Relation und dahinter die Attributmenge angegeben, auf die sich der Fremdschlüssel bezieht, in der er also als Schlüssel vorkommt

Verwaltung von Datenbanken und Tabellen

- Erstellen/Löschen einer leeren Datenbank:

```
create database db_name
drop database db_name
```

- Erstellen/Löschen/Ändern einer leeren Tabelle:

```
create/drop/alter table tbl_name (
Spalte1 Typ(Länge),
    Spalte2 Typ(Länge),
    Spalte3 Typ(Länge) )

    rename table tbl_name to tbl_name_new
```

Index

- Indizieren einer oder mehrer Spalten einer Tabelle:

```
create index indx_name on tbl_name (Spalte)
```

→ Dient der besseren Performance durch schnelleres Auffinden häufig angefragter Daten

Übung 9:

- Lösen Sie die Aufgaben der Übung 9
- Überprüfen Sie Ihre Anfrage in mySQL