

## Vorlesung



University of Applied Sciences

# DBSP

Unit  
Datenbanken  
SQL

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

1



University of Applied Sciences



### Prof. Dr. rer. nat. Nane Kratzke

*Praktische Informatik und  
betriebliche Informationssysteme*

- Raum: 17-0-10
- Tel.: 0451 300 5549
- Email: [kratzke@fh-luebeck.de](mailto:kratzke@fh-luebeck.de)



@NaneKratzke

Updates der Handouts auch über Twitter #dbsp

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

2

**Handout zur Vorlesung**  
**DBSP – Datenbanken und SQL – Unit 12**

## Übergreifende Ziele der Lehrveranstaltung



University of Applied Sciences

Client- und Serverseitige Entwicklung

PHP (Serverseitig)

JavaScript (Clientseitig)

„Hosten“ von Apps

Framework Erfahrungen

CMS (Drupal)

WebServices (Google-Maps)

jQuery

Datenbank-Integration

Berücksichtigung von Sicherheitsaspekten

HTML-Injections

SQL-Injections

Session Hijacking

Login-Systeme

Um sich weitere Web-Technologien autodidaktisch erarbeiten zu können.

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

3

## Units



University of Applied Sciences

Unit 1  
Cloud Computing  
IaaS

Unit 2  
CMS Drupal

Unit 3  
HTML und CSS

Unit 4 - 7  
PHP I - IV

Unit 8  
Sessions, Cookies,  
Formulare und  
Login-System

Unit 9  
JavaScript

Unit 10  
Drupal Module  
Development

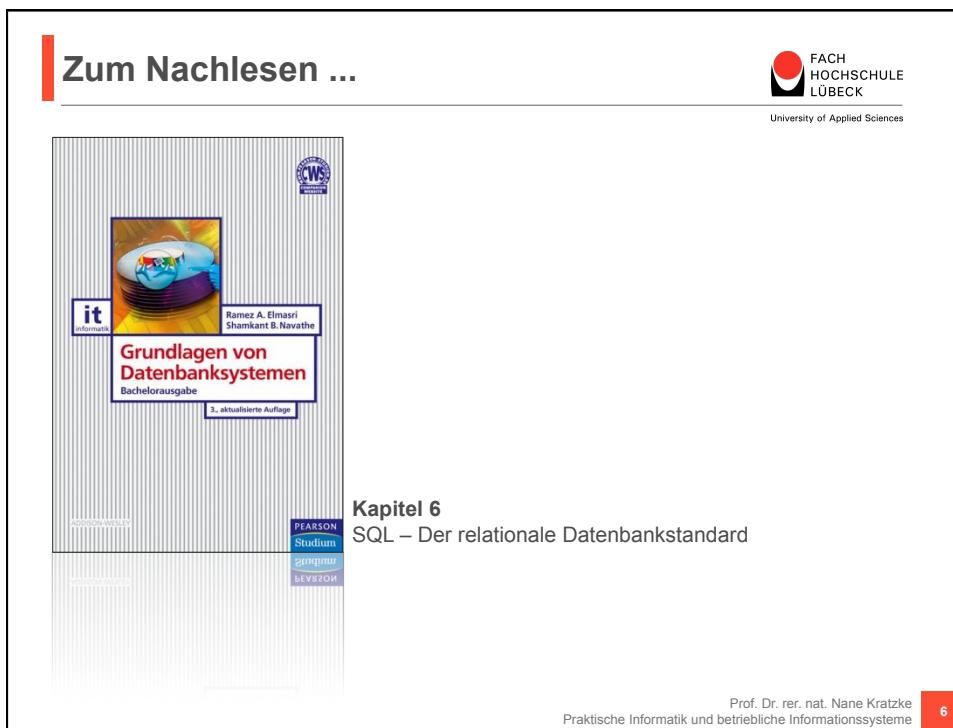
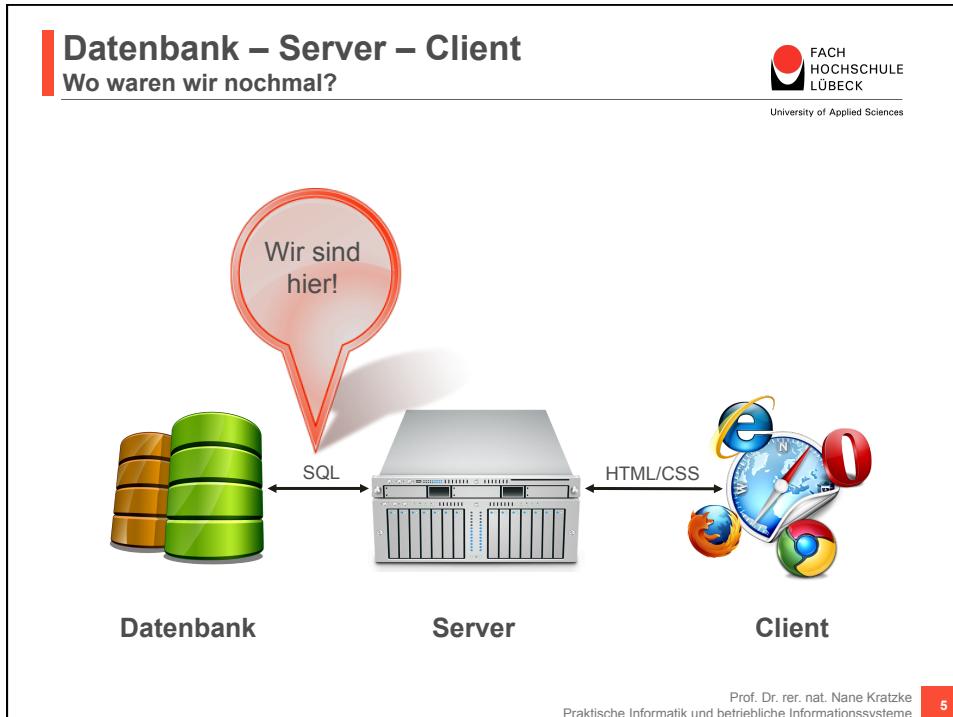
Unit 11  
Datenmodellierung

Unit 12 - 13  
Datenbanken und SQL  
Vom Datenmodell zur  
Datenbank

Unit 14  
Datenbank-gestützte  
Web-Anwendungen

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

4



## Zum Nachlesen ...



University of Applied Sciences



### Bereitgestellte Skripte:

#### SQL – Programmierung

- **Kapitel 2:** Sprachbestandteile der SQL
- **Kapitel 4:** Praktischer Einsatz von SQL Befehlen

<http://praktische-informatik.fh-luebeck.de/node/18>

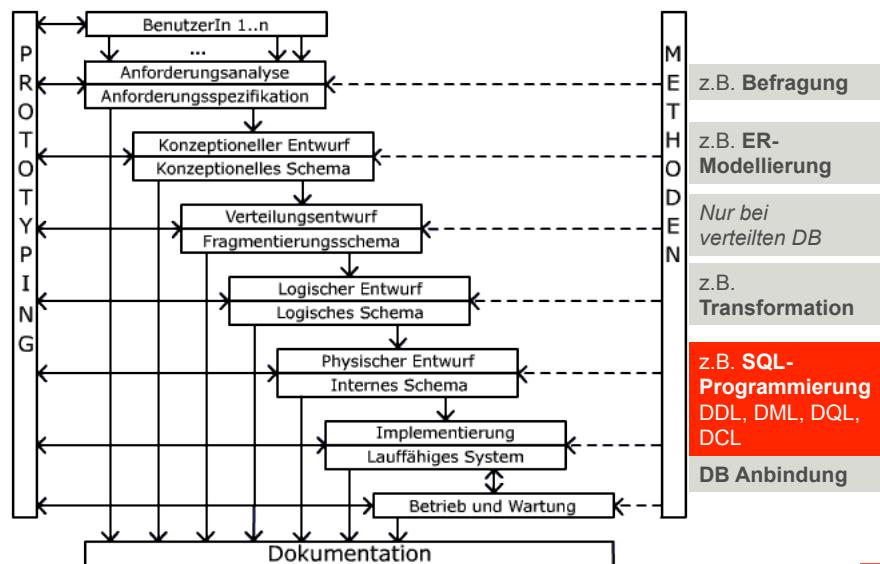
Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

7

## Phasenmodell des Entwurfsprozesses



University of Applied Sciences



Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

8

## Kategorien von SQL Befehlen



University of Applied Sciences

### DDL

Data Definition Language

Befehle zur  
Definition der  
Datenbankschema

### DQL

Data Query Language

Befehle zur  
Abfrage von  
Relationen  
•SELECT

### DML

Data Manipulation Language

Befehle zur  
Daten-  
manipulation  
•Ändern  
•Einfügen  
•Löschen

### DCL

Data Control Language

Befehle für die  
•Rechteverwaltung  
•Transaktionskontrolle

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

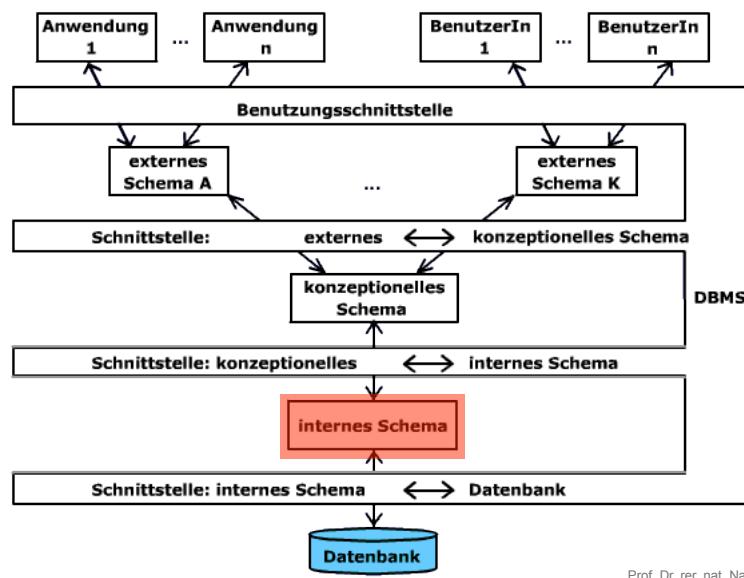
9

## Wo sind wir in der Drei-Ebenen- Schemaarchitektur?



University of Applied Sciences

Logische Sicht für Datenbanken



Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

10

## Data Definition Language **create**



FACH  
HOCHSCHULE  
LÜBECK  
University of Applied Sciences

<p><b>Anlegen einer Datenbank</b> hier Datenbank mit Namen <i>Bibliothek</i></p>	<pre><b>CREATE DATABASE</b> Bibliothek;</pre>
<p><b>Anlegen einer Tabelle</b> Hier Tabelle über alle Bücher mit Attributen <i>ISBN, Titel, Exemplare, Leihfrist</i>.</p>	<pre><b>CREATE TABLE</b> Buch (     ISBN VARCHAR(20) NOT NULL,     Titel VARCHAR(100),     Exemplare NUMBER(3,0) NOT NULL,     Leihfrist Frist);</pre>
<p><b>Anlegen eines Datentyps</b> hier zweistellige Frist mit Standardwert 30 und zulässigen Werten von 15, 30 und 60.</p>	<pre><b>CREATE DOMAIN</b> Frist AS NUMBER(2,0) <b>DEFAULT</b> 30 <b>CONSTRAINT</b> UngueltigeFrist <b>CHECK</b>     (VALUE IN (15, 30, 60));</pre>

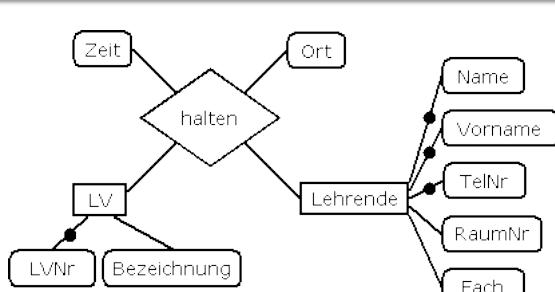
Prof. Dr. rer. nat. Nane Kratzke      11  
 Praktische Informatik und betriebliche Informationssysteme

## Data Definition Language **Erweiterte create table Anweisung**

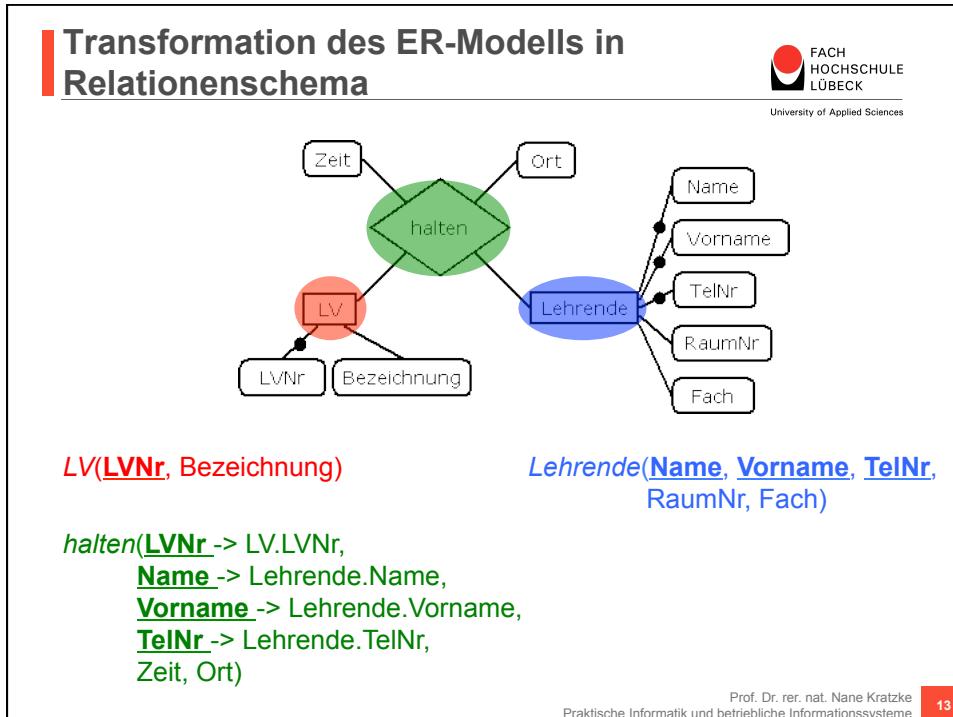


FACH  
HOCHSCHULE  
LÜBECK  
University of Applied Sciences

<p>Veranschaulichung an folgendem Beispiel</p>	<pre><b>CREATE TABLE</b> [Database.]Relname (     Attributdefinition, ...,     [<b>PRIMARY KEY</b> (Keys...),]     [<b>FOREIGN KEY</b> (Attr...) <b>REFERENCES</b> Relname(Keys...),]     [<b>CONSTRAINT</b> Regelname (<b>CHECK</b> <b>UNIQUE</b>)-Bedingung]);</pre>
--	--



Prof. Dr. rer. nat. Nane Kratzke      12  
 Praktische Informatik und betriebliche Informationssysteme



## Übersetzen der Relationenschema in create table Statements



University of Applied Sciences

*halten(LVNr -> LV.LVNr, Name -> Lehrende.Name,  
Vorname -> Lehrende.Vorname, TelNr -> Lehrende.TelNr,  
Zeit, Ort)*

```
CREATE TABLE halten(  
    LVNr NUMBER(4,0),  
    Name VARCHAR(32),  
    Vorname VARCHAR(32),  
    TelNr VARCHAR(32),  
    Zeit TIME,  
    Ort VARCHAR(32),  
    PRIMARY KEY (LVNr, Name, Vorname, TelNr),  
    FOREIGN KEY (LVNr REFERENCES LV(LVNr)),  
    FOREIGN KEY (Name REFERENCES Lehrende(Name)),  
    FOREIGN KEY (Vorname REFERENCES Lehrende(Vorname)),  
    FOREIGN KEY (TelNr REFERENCES Lehrende(TelNr))  
);
```

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

15

## Weitere DDL Befehle und Klauseln



University of Applied Sciences

- Erzeugen von Indizes
- Zur Beschleunigung von Queries

**CREATE INDEX**

- Änderungen von Relationen
- Löschen und Ergänzen von Attributen

**ALTER**

- Prüfung von Konsistenzbedingungen
- Sicherstellen das Fremdschlüssel nur auf ein Element zur Zeit verweisen

**CHECK/  
UNIQUE**

- Löschen von Tabellen

**DROP**

**Hinweis:** Eine Erläuterung aller SQL-DDL-Befehle und – Klauseln finden Sie in Ihrem Online Skript.



Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

16

## Kategorien von SQL Befehlen



University of Applied Sciences

### DDL

Data Definition Language

Befehle zur  
Definition der  
Datenbankschema

### DQL

Data Query Language

Befehle zur  
Abfrage von  
Relationen  
•SELECT

### DML

Data Manipulation Language

Befehle zur  
Daten-  
manipulation  
•Ändern  
•Einfügen  
•Löschen

### DCL

Data Control Language

Befehle für die  
•Rechteverwaltung  
•Transaktionskontrolle

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

17

## Data Query Language SQL SELECT Klausel



University of Applied Sciences

Abfragen  
innerhalb  
von  
Relationen

DQL

Geschach-  
telte  
Abfragen

Abfragen über  
mehrere  
Relationen

**INNER JOINS**  
**SELF JOINS**  
**OUTER JOINS**

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

18

## Abfragen innerhalb von Relationen



University of Applied Sciences

### Grundmuster einer Query

```
SELECT Attr, Attr, ...      // Selektion der Spalten
FROM Relation            // Angabe der Relation
WHERE Bedingung          // Selektion der Zeilen
[ORDER BY Attr [ASC|DESC]];// Sortierung
```

**Hinweis:** In Ihrem Skript „SQL Programmierung“ im Kapitel 2.3 und 4.3 finden Sie eine kurze und knappe Übersicht wie WHERE Klauseln zu formulieren sind.

Die WHERE Klausel ist vermutlich die wichtigste Klausel der DQL.

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

19

## Abfragen innerhalb von Relationen Beispiel



University of Applied Sciences

Buch	Autor	Titel	Seitenzahl
	A. Diavolo	Die Pest	21
	M. Engels	Der Himmel	19
	F. Marx	Das Kapital	1005
	M. Muster	Die Hölle	235



```
SELECT Titel, Autor
FROM Buch
WHERE Seitenzahl > 20 AND
    NOT Titel = ,Die Pest';
```

Autor	Titel
F. Marx	Das Kapital
M. Muster	Die Hölle

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

20

## Abfragen über mehrere Relationen INNER JOINS

Im Allgemeinen werden bei einer Datenbankanfrage Informationen aus mehreren Relationen benötigt und zusammengestellt.

Hier sehen sie das Grundmuster, wie dies in SQL mittels einer SELECT Klausel ausgedrückt werden kann.

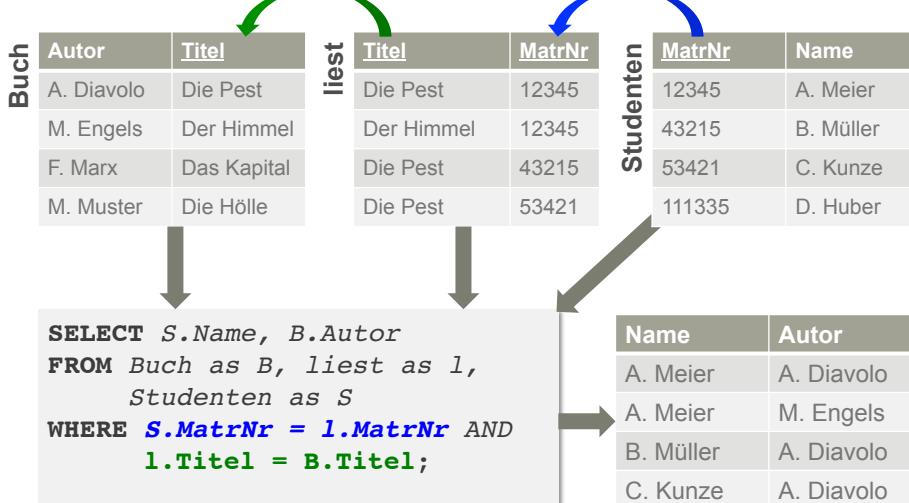
### Grundmuster eines INNER JOINS

```
SELECT r1.Attrs, ..., r2.Attrs  

FROM Rel1 AS r1, Rel2 AS r2, ...  

WHERE Bedingung;
```

## Abfragen über mehrere Relationen INNER JOINS - Beispiel



## Abfragen über ein und dieselbe Relation **SELF JOINS**



FACH  
HOCHSCHULE  
LÜBECK  
University of Applied Sciences

Studierende	Name	GebDat
	Meier	07.10.85
	Schulze	03.05.80
	König	07.10.85
	Baum	12.11.82
	Dreier	25.02.84
	Hesse	07.10.85

**Grundmuster eines SELF JOINS**

```
SELECT a.Attrs, ...
FROM Rel AS a, Rel AS b
WHERE Bedingung;
```

```
SELECT a.Name, a.GebDat
FROM Studierende AS a,
      Studierende AS b
WHERE a.GebDat = b.GebDat
AND a.Name != b.Name
ORDER BY GebDat;
```



Name	GebDat
Meier	07.10.85
König	07.10.85
Hesse	07.10.85

Prof. Dr. rer. nat. Nane Kratzke | 23
   
Praktische Informatik und betriebliche Informationssysteme

## Abfragen über mehrere Relationen **OUTER JOINS**



FACH  
HOCHSCHULE  
LÜBECK  
University of Applied Sciences

Neben den Inner-Joins gibt es auch noch die Outer-Joins. Ein Outer-Join übernimmt im Gegensatz zum Inner-Join auch die Datensätze aus den Ausgangstabellen, die nicht in beiden Tabellen Entsprechungen besitzen.

**Grundmuster eines OUTER JOINS**

```
SELECT LR.Attrs, ..., RR.Attrs
FROM Leftrelation AS LR
{LEFT|RIGHT|FULL} OUTER JOIN Rightrelation AS RR
ON Bedingung;
```

Prof. Dr. rer. nat. Nane Kratzke | 24
   
Praktische Informatik und betriebliche Informationssysteme

## OUTER JOINS

### Beispiel RIGHT OUTER JOIN


  
University of Applied Sciences

Titel	MatrNr
Die Pest	12345
Der Himmel	12345
Die Pest	43215
Die Pest	53421

MatrNr	Name
12345	A. Meier
43215	B. Müller
53421	C. Kunze
111335	D. Huber

```
SELECT S.Name, l.Titel
FROM liest AS l
RIGHT OUTER JOIN Studenten AS S
ON S.MatrNr = l.MatrNr;
```

Name	Titel
A. Meier	Die Pest
A. Meier	Der Himmel
B. Müller	Die Pest
C. Kunze	Die Pest
D. Huber	

Prof. Dr. rer. nat. Nane Kratzke | 25

## Geschachtelte Abfragen

### Bsp.: Gleicher Geburtstag mit einer Person


  
University of Applied Sciences

Name	GebDat
Meier	07.10.85
Schulze	03.05.80
König	07.10.85
Baum	12.11.82
Dreier	25.02.84
Hesse	07.10.85

```
SELECT Name, GebDat
FROM Studierende
WHERE GebDat IN
(SELECT GebDat
 FROM Studierende
 WHERE Name = 'Hesse')
ORDER BY GebDat;
```

GebDat
07.10.85

Prof. Dr. rer. nat. Nane Kratzke | 26

*Hinweis: Unterabfragen und Joins können oft dasselbe Ergebnis zurückliefern und sind in der Anwendung ähnlich komplex, so dass es häufig dem persönlichen Geschmack überlassen ist, welche Variante man nutzt.*

## Weitere SELECT-Klauseln



University of Applied Sciences

### DISTINCT

- Keine Doppelausgabe identischer Tupel

### ORDER BY

- Sortieren nach bestimmten Kriterien

### GROUP BY und HAVING

- Zusammenfassen und ggf. Filtern von Datensätzen mit gleichen Wertevorkommen in einem Attribut

### Aggregationsfunktionen

- COUNT, SUM, MAX, MIN, AVG

**Hinweis:** Eine Erläuterung aller SELECT-Klauseln finden Sie in Ihrem Online Skript.



Prof. Dr. rer. nat. Nane Kratzke

27

Praktische Informatik und betriebliche Informationssysteme

## Kategorien von SQL Befehlen



University of Applied Sciences

### DDL

Data Definition Language

Befehle zur Definition der Datenbankschema

### DQL

Data Query Language

Befehle zur Abfrage von Relationen  
•SELECT

### DML

Data Manipulation Language

Befehle zur Datenmanipulation  
•Ändern  
•Einfügen  
•Löschen

### DCL

Data Control Language

Befehle für die  
•Rechteverwaltung  
•Transaktionskontrolle

Prof. Dr. rer. nat. Nane Kratzke

28

Praktische Informatik und betriebliche Informationssysteme

## Data Manipulation Language

```

    graph TD
        DML((DML)) --- Insert((Insert))
        DML --- Delete((Delete))
        DML --- Update((Update))
    
```

 FACH  
HOCHSCHULE  
LÜBECK  
University of Applied Sciences

Der DML-Anteil von SQL dient dazu Daten einer Daten

- hinzuzufügen
- zu ändern
- und zu löschen

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

## INSERT INTO

**Grundmuster einer INSERT Klausel:**

```
INSERT INTO Relation(Attrs, ...)
VALUES (Vals, ...);
```

**INSERT INTO Kunde(KNr, Name)**  
**VALUES (3, 'C. Meier');**

**Zweite Variante:** Kopieren aus einer anderen Tabelle mittels **SELECT**

NeuKNr	Name
3	C. Meier
...	...

**INSERT INTO Kunde(KNr, Name)**  
**SELECT NeuKNr AS KNr, Name**  
**FROM Neukunden;**

 FACH  
HOCHSCHULE  
LÜBECK  
University of Applied Sciences

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

## UPDATE



University of Applied Sciences

### Grundmuster einer UPDATE Klausel:

```
UPDATE Relation
SET Attr = Wert
WHERE Bedingung;
```

Der **UPDATE** Befehl ist dazu gedacht, in der Datenbank befindliche Datensätze zu verändern. Die zu ändernden Datensätze werden durch eine **WHERE** Klausel ausgewählt.

KNr	Name
1	Müller
2	Kunze
3	Meier



```
UPDATE Kunde
SET Name = ,Mayer'
WHERE Name LIKE ,M%';
```

KNr	Name
1	<b>Mayer</b>
2	Kunze
3	<b>Mayer</b>

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

31

## DELETE



University of Applied Sciences

### Grundmuster einer DELETE Klausel:

```
DELETE FROM Relation
WHERE Bedingung;
```

Mit dem **DELETE** Befehl können vorhandene Datensätze aus einer Relation gelöscht werden. Es werden immer komplette Datensätze (Zeilen) gelöscht. Die zu löschen Datensätze werden durch eine **WHERE** Klausel ausgewählt.

KNr	Name
1	Müller
2	Kunze
3	Meier



```
DELETE FROM Kunde
WHERE Name LIKE ,M%';
```

KNr	Name
<b>4</b>	<b>Müller</b>
2	Kunze
<b>3</b>	<b>Meier</b>

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

32

## Kategorien von SQL Befehlen



University of Applied Sciences

### DDL

Data Definition Language

Befehle zur  
Definition der  
Datenbank-  
schema

### DQL

Data Query Language

Befehle zur  
Abfrage von  
Relationen  
•SELECT

### DML

Data Manipulation Language

Befehle zur  
Daten-  
manipulation  
•Ändern  
•Einfügen  
•Löschen

### DCL

Data Control Language

Befehle für die  
•Rechteverwaltung  
•Transaktionskontrolle

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

33

## Data Control Language (DCL) Transaction Control Language (TCL)



University of Applied Sciences

### DCL

- **GRANT** zur Erteilung von Zugriffsrechten
- **REVOKE** zur Entziehung von Zugriffsrechten

### TCL

- **COMMIT** zum abschließen einer Transaktion
- **ROLLBACK** zum wiederherstellen eines validen Zustands der Datenbank vor Transaktionsstart
- Die TCL dient der Integrität von Datenbanken in Mehrbenutzerszenarien mit konkurrierenden Zugriffen.

**Hinweis:** DCL und TCL werden in dieser Veranstaltung nicht weiter angesprochen und bleiben dem Selbststudium im Rahmen der Projektarbeit überlassen.

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

34

## Kategorien von SQL Befehlen



University of Applied Sciences

**DDL**

Data Definition Language

Befehle zur  
Definition  
Datenbank  
schema

**DQL**

Data Quer...  
SELECT

**DML**

Data  
Manipulation  
Language

Befehle zur  
Daten-  
manipulation  
•Ändern  
•Einfügen  
•Löschen

**DCL**

Data Control  
Language

Befehle für die  
•Rechteverwaltung  
•Transaktionskontrolle

Sichten

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

35

## Was ist der Unterschied zwischen einer Relation und einer Sicht?



University of Applied Sciences

### Relation

- Speicherstruktur in Datenbank
- Speichert physisch Daten

### Sicht

- Virtuelle Relation
- In einer Sicht sind keine Daten gespeichert
- eine Art „gespeicherter“ SELECT Befehl
- Zusammenstellung der Daten für spez. Zwecke
- Schutz von Daten (z.B. Ausblenden von Gehaltszahlen)

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

36

## Beispiel für Sichten mittels CREATE VIEW



University of Applied Sciences

liest

Titel	MatrNr
Die Pest	12345
Der Himmel	12345
Die Pest	43215
Die Pest	53421

Studenten

MatrNr	Name
12345	A. Meier
43215	B. Müller
53421	C. Kunze
111335	D. Huber

Name	Titel
A. Meier	Die Pest
A. Meier	Der Himmel
B. Müller	Die Pest
C. Kunze	Die Pest

```
CREATE VIEW Studenten_leSEN_Titel
  (Name, Titel)
SELECT S.Name, l.Titel
FROM   liest AS l, Studenten AS S
WHERE  S.MatrNr = l.MatrNr;
```

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

37

## Zusammenfassung



University of Applied Sciences

- Data Definition Language (DDL)
- Data Query Language (DQL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Views

Prof. Dr. rer. nat. Nane Kratzke  
Praktische Informatik und betriebliche Informationssysteme

38