



FACH
HOCHSCHULE
LÜBECK
University of Applied Sciences

Web-Technologien

SQL

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

1



FACH
HOCHSCHULE
LÜBECK
University of Applied Sciences

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

- **Raum: 17-0.10**
- **Tel.: 0451 300 5549**
- **Email: nane.kratzke@fh-luebeck.de**

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

2

Datenbank – Server – Client

Wo waren wir nochmal?

Wir sind hier!

Datenbank Server Client

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

3

Zum Nachlesen ...

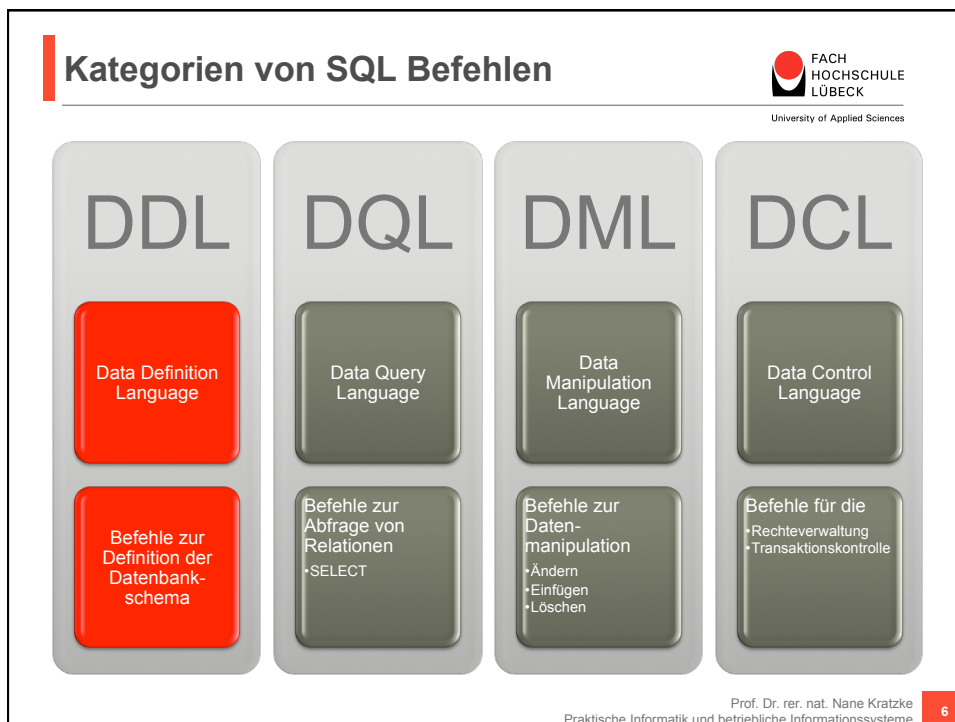
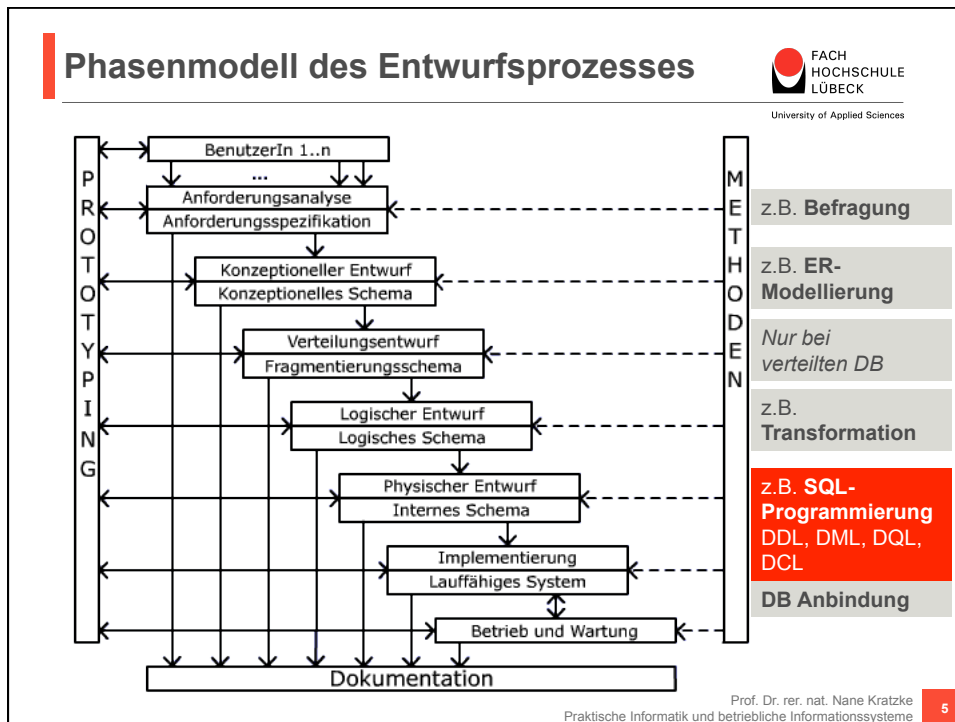
it
Ramez A. Elmasri
Shamkant B. Navathe
Grundlagen von Datenbanksystemen
Bachelorausgabe
2., aktualisierte Auflage

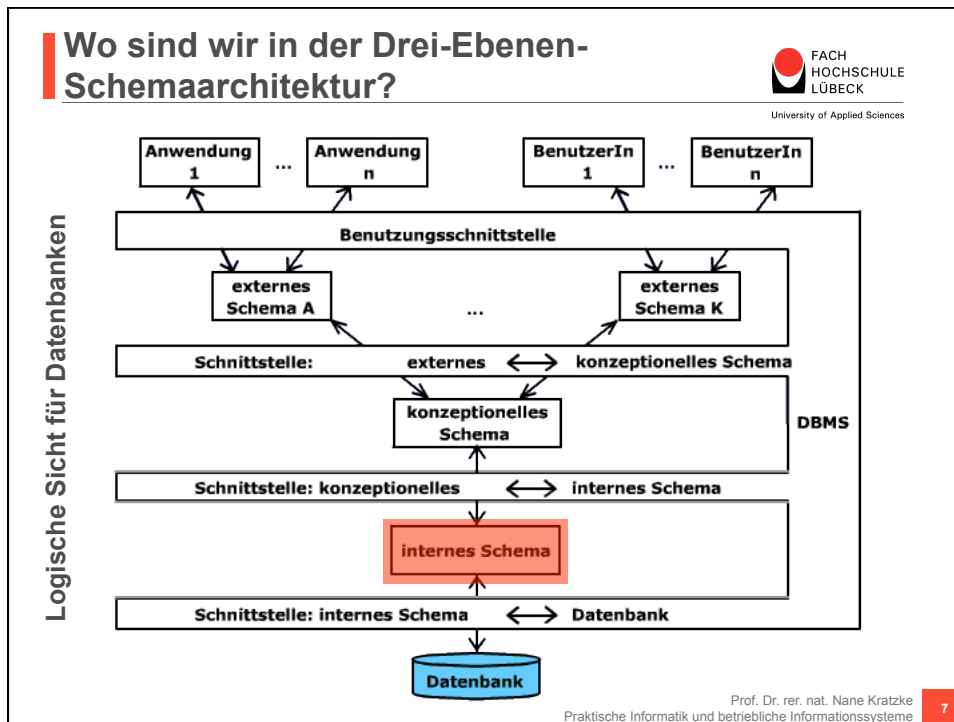
PEARSON
Studium

Kapitel 6
SQL – Der relationale Datenbankstandard

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

4





Data Definition Language

create

The diagram shows three examples of Data Definition Language (DDL) commands:

- Anlegen einer Datenbank** (Creating a database): `CREATE DATABASE Bibliothek;` (here with name *Bibliothek*)
- Anlegen einer Tabelle** (Creating a table): `CREATE TABLE Buch (ISBN VARCHAR(20) NOT NULL, Titel VARCHAR(100), Exemplare NUMBER(3,0) NOT NULL, Leihfrist Frist);` (here with attributes *ISBN, Titel, Exemplare, Leihfrist*)
- Anlegen eines Datentyps** (Creating a domain): `CREATE DOMAIN Frist AS NUMBER(2,0) DEFAULT 30 CONSTRAINT UngueltigeFrist CHECK (VALUE IN (15, 30, 60));` (here with standard value 30 and allowed values 15, 30, 60)

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

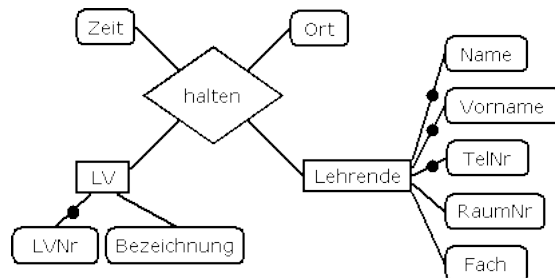
Prof. Dr. rer. nat. Nane Kratzke
Praktische Informatik und betriebliche Informationssysteme

8

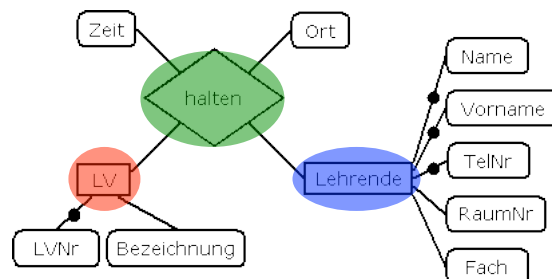
Data Definition Language Erweiterte create table Anweisung

```
CREATE TABLE [Database.]Relname (
  Attributdefinition, ...,
  [PRIMARY KEY (Keys...)],
  [FOREIGN KEY (Attr...) REFERENCES Relname(Keys...)],
  [CONSTRAINT Regelname (CHECK|UNIQUE)-Bedingung]);
```

Veranschaulichung
an folgendem
Beispiel



Transformation des ER-Modells in Relationenschema



LV(LVNr, Bezeichnung)

Lehrende(Name, Vorname, TelNr,
RaumNr, Fach)

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

Übersetzen der Relationenschema in create table Statements

LV(LVNr, Bezeichnung)

```
CREATE TABLE LV (  
LVNr NUMBER(4,0),  
Bezeichnung VARCHAR(32),  
PRIMARY KEY LVNr);
```

Lehrende(Name, Vorname, TelNr, RaumNr, Fach)

```
CREATE TABLE Lehrende (  
Name VARCHAR(32),  
Vorname VARCHAR(32),  
TelNr VARCHAR(32),  
RaumNr VARCHAR(16),  
Fach VARCHAR(32),  
PRIMARY KEY (Name, Vorname, TelNr);
```

Übersetzen der Relationenschema in create table Statements

*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))  
);
```

Weitere DDL Befehle und Klauseln

FACH HOCHSCHULE LÜBECK
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

13

Kategorien von SQL Befehlen

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

DDL	DQL	DML	DCL
Data Definition Language	Data Query Language	Data Manipulation Language	Data Control Language
Befehle zur Definition der Datenbankschema	Befehle zur Abfrage von Relationen •SELECT	Befehle zur Datenmanipulation •Ändern •Einfügen •Löschen	Befehle für die •Rechteverwaltung •Transaktionskontrolle

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

14

Data Query Language SQL SELECT Klausel

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

Abfragen innerhalb von Relationen

DQL

Geschachtelte Abfragen

Abfragen über mehrere Relationen

INNER JOINS
SELF JOINS
OUTER JOINS

Prof. Dr. rer. nat. Nane Kratzke
Praktische Informatik und betriebliche Informationssysteme 15

Abfragen innerhalb von Relationen

FACH HOCHSCHULE LÜBECK
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 16

Abfragen innerhalb von Relationen Beispiel

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

Abfragen über mehrere Relationen INNER JOINS

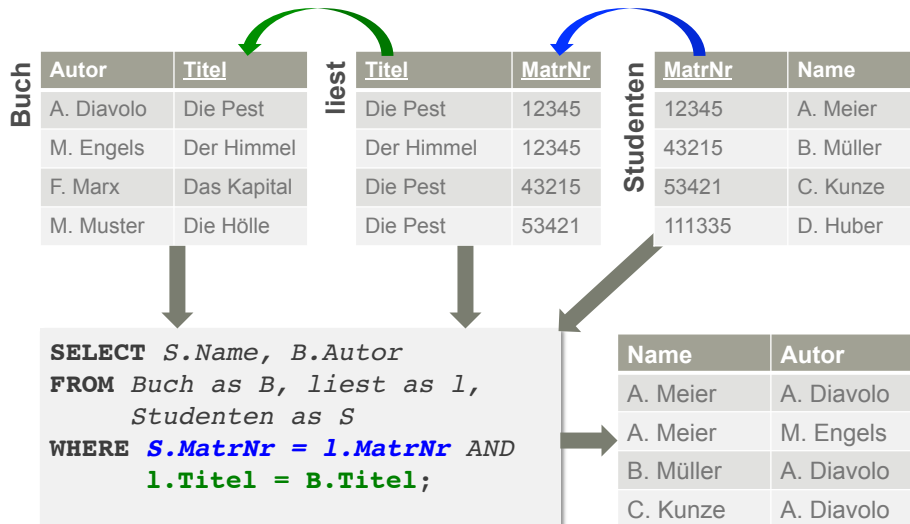
Im Allgemeinen werden bei einer Datenbankabfrage Informationen aus mehreren Relationen benötigt und zusammengestellt.

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

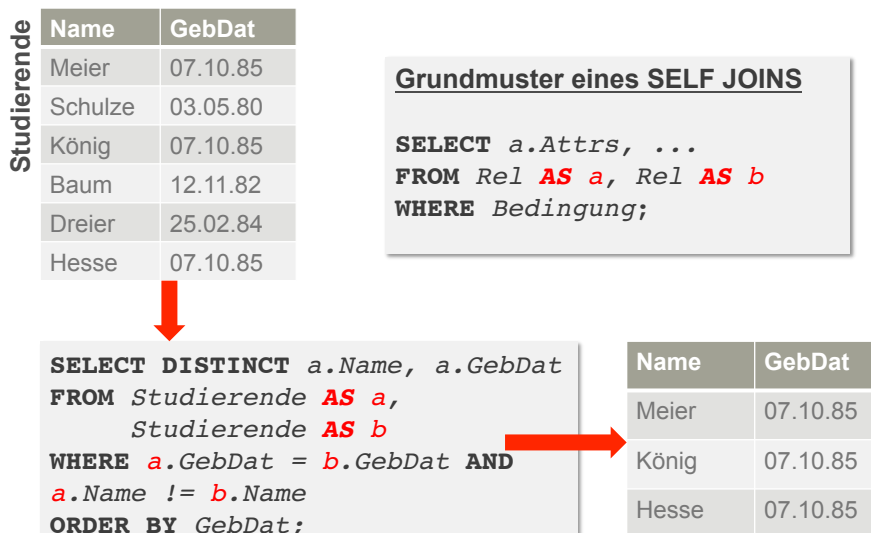
Grundmuster eines INNER JOINS

```
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



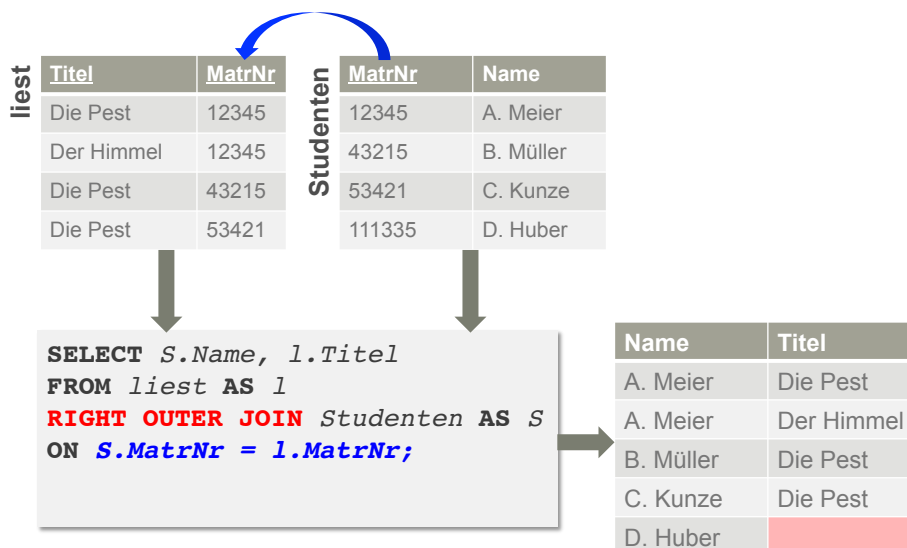
Abfragen über mehrere Relationen OUTER JOINS

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;
```

OUTER JOINS Beispiel RIGHT OUTER JOIN



Geschachtelte Abfragen

Bsp.: Gleicher Geburtstag mit einer Person

FACH HOCHSCHULE LÜBECK
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

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.

Name	GebDat
Meier	07.10.85
König	07.10.85
Hesse	07.10.85

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

Weitere SELECT-Klauseln

FACH HOCHSCHULE LÜBECK
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

Aggregieruns-funktionen


- COUNT, SUM, MAX, MIN, AVG

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



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

Kategorien von SQL Befehlen

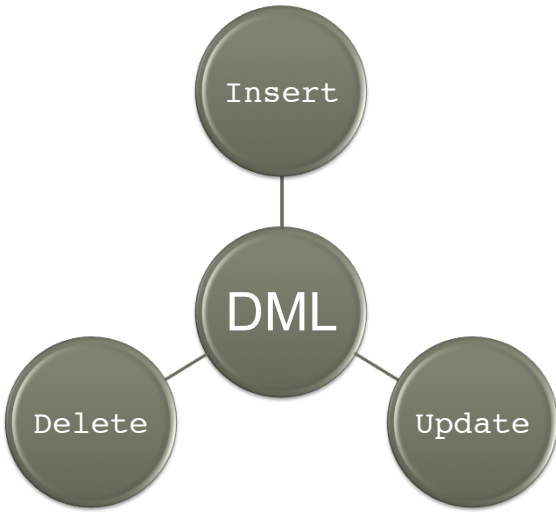



DDL	DQL	DML	DCL
Data Definition Language	Data Query Language	Data Manipulation Language	Data Control Language
Befehle zur Definition der Datenbankschema	Befehle zur Abfrage von Relationen •SELECT	Befehle zur Datenmanipulation •Ändern •Einfügen •Löschen	Befehle für die •Rechteverwaltung •Transaktionskontrolle

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

25

Data Manipulation Language



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

26

INSERT INTO

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

Grundmuster einer INSERT Klausel:

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

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

KNr	Name
1	A. Müller
2	B. Kunze
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;
```

Prof. Dr. rer. nat. Nane Kratzke
Praktische Informatik und betriebliche Informationssysteme 27

UPDATE

FACH HOCHSCHULE LÜBECK
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	Mayer
2	Kunze
3	Mayer

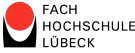
Prof. Dr. rer. nat. Nane Kratzke
Praktische Informatik und betriebliche Informationssysteme 28

DELETE

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 löschenden Datensätze werden durch eine **WHERE** Klausel ausgewählt.



University of Applied Sciences

KNr	Name
1	Müller
2	Kunze
3	Meier

↓

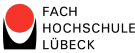
```
DELETE FROM Kunde
WHERE Name LIKE ,M% ';
```

↓

KNr	Name
4	Müller
2	Kunze
3	Meier

Prof. Dr. rer. nat. Nane Kratzke
Praktische Informatik und betriebliche Informationssysteme
29

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
Praktische Informatik und betriebliche Informationssysteme
30

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

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

DCL	TCL
<ul style="list-style-type: none"> • GRANT zur Erteilung von Zugriffsrechten • REVOKE zur Entziehung von Zugriffsrechten 	<ul style="list-style-type: none"> • 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

31

Kategorien von SQL Befehlen

FACH HOCHSCHULE LÜBECK
University of Applied Sciences

DDL	DQL	DML	DCL
Data Definition Language	Data Query Language	Data Manipulation Language	Data Control Language
Befehle zur Definition von Datenbankschemata	Befehle zum Abfragen von Daten (SELECT)	Befehle zur Datenmanipulation • Ändern • Einfügen • Löschen	Befehle für die • Rechteverwaltung • Transaktionskontrolle

Sichten

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

32

Was ist der Unterschied zwischen einer Relation und einer Sicht?

Relation
<ul style="list-style-type: none"> • Speicherstruktur in Datenbank • Speichert physisch Daten

Sicht
<ul style="list-style-type: none"> • 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)

Beispiel für Sichten mittels CREATE VIEW

liest	Titel	MatrNr	Studenten	MatrNr	Name	Name	Titel
	Die Pest	12345		12345	A. Meier	A. Meier	Die Pest
	Der Himmel	12345		43215	B. Müller	A. Meier	Der Himmel
	Die Pest	43215		53421	C. Kunze	B. Müller	Die Pest
	Die Pest	53421		111335	D. Huber	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;
```

Zusammenfassung

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