

Kapitel 5

Dr. Jérôme Kunegis

SQL: Grundlagen

Lernziele

Kenntnis der Grundkonzepte von SQL

Fähigkeit zur praktischen Anwendung von einfachen SQL-Anweisungen

- Schema-Definitionen, referentieller Integrität
- SELECT
- Änderungsoperationen

SQL: Komponenten der vollständigen DB-Sprache

Datenmanipulation

- Einfügen, Löschen und Ändern von individuellen Tupeln und von Mengen von Tupeln
- Zuweisung von ganzen Relationen

Datendefinition

- Definition von Wertebereichen, Attributen und Relationen
- Definition von verschiedenen Sichten auf Relationen

Datenkontrolle

- Spezifikation von Bedingungen zur Zugriffskontrolle
- Spezifikation von Zusicherungen (Assertions) zur semantischen Integritätskontrolle

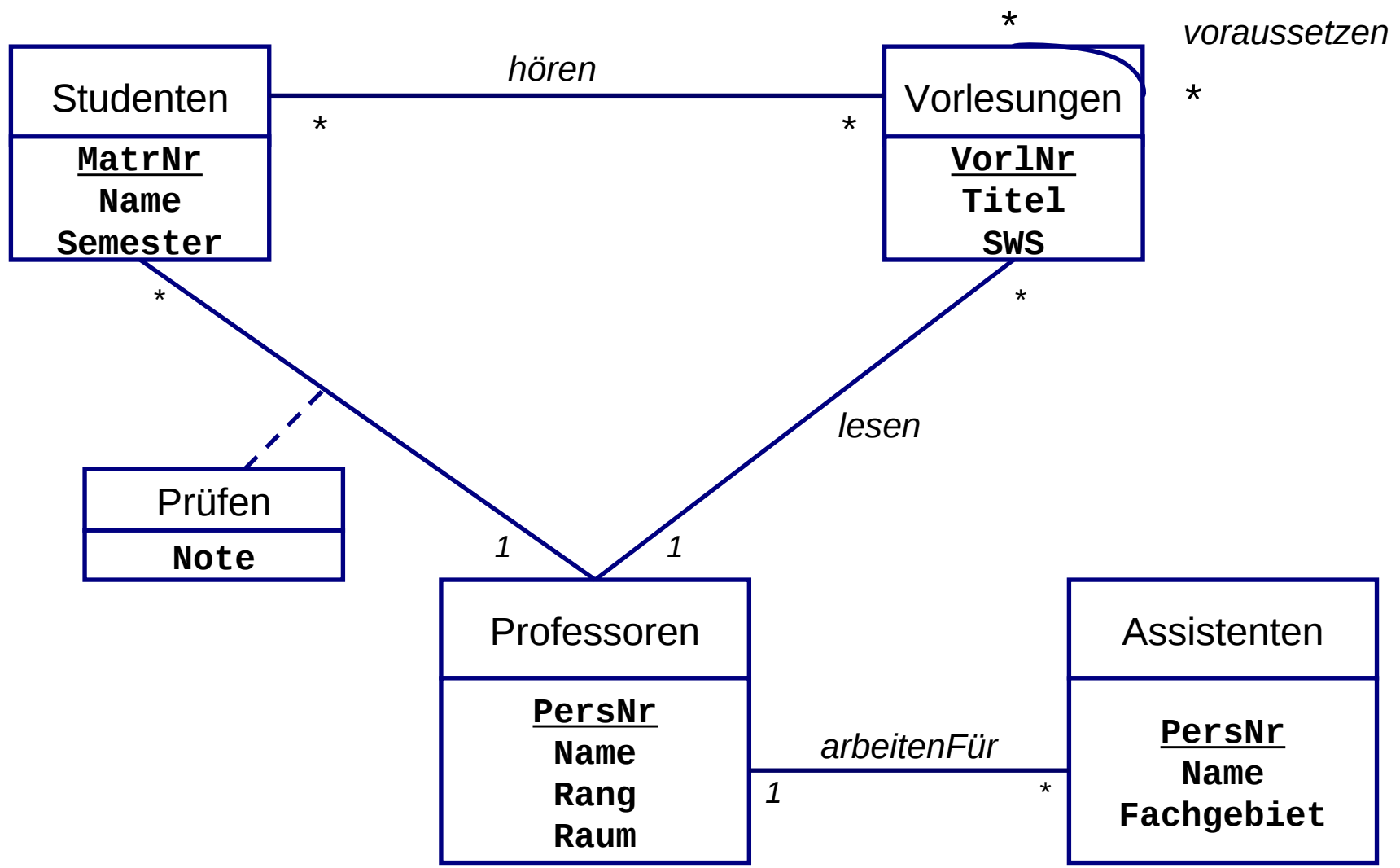
Datenabfragen

- Abbildung von Eingaberelationen durch Auswertung von Bedingungen auf die Ergebnisrelation

Kopplung mit einer Hostsprache (z.B. PHP)

- deskriptive Auswahl von Mengen von Tupeln
- sukzessive Bereitstellung einzelner Tupeln

Beispiel: Universitätsschema



Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	F. Schlegel	2

Vorlesungen				
VorINr	Titel	SWS	gelesenVon	
5001	Grundzüge	4	2137	
5041	Ethik	4	2125	
5043	Erkenntnistheorie	3	2126	
5049	Mäeutik	2	2125	
4052	Logik	4	2125	
5052	Wissenschaftstheorie	3	2126	
5216	Bioethik	2	2126	
5259	Der Wiener Kreis	2	2133	
5022	Glaube und Wissen	2	2134	
4630	Die 3 Kritiken	4	2137	

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören	
MatrNr	VorINr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
29120	5001
29120	5041
29120	5049
25403	5022

Assistenten			
PersINr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

prüfen			
MatrNr	VorINr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2

Relationale Invarianten

Integritätsbedingungen des Relationenmodells

- Primärschlüsselbedingung (Entity-Integrität)
 - Eindeutigkeit des Primärschlüssels
 - keine Nullwerte!
- Referentielle Integrität:
 - Darstellung von Beziehungen durch Fremdschlüssel (foreign key):
Attribut, das in Bezug auf den Primärschlüssel einer anderen (oder derselben) Relation definiert ist
 - zugehöriger Primärschlüssel muss existieren, d.h. zu jedem Wert (ungleich Null) eines Fremdschlüsselattributs einer Relation R2 muss ein gleicher Wert des Primärschlüssels in irgendeinem Tupel von Relation R1 vorhanden sein

Relationale Invarianten (1)

Fremdschlüssel und zugehöriger Primärschlüssel tragen wichtige interrelationale Informationen

- sie sind auf dem gleichen Wertebereich definiert
- sie gestatten die Verknüpfung von Relationen mit Hilfe von Relationenoperationen

Fremdschlüssel

- können Nullwerte aufweisen, wenn sie nicht Teil eines Primärschlüssels sind.
- ein Fremdschlüssel ist „zusammengesetzt“, wenn der zugehörige Primärschlüssel „zusammengesetzt“ ist

Eine Relation kann mehrere Fremdschlüssel besitzen, die die gleiche oder verschiedene Relationen referenzieren

Zyklen sind möglich (***geschlossener referentieller Pfad***)

Eine Relation kann zugleich referenzierende und referenzierte Relation sein (***„self-referencing table“***).

SQL: CREATE TABLE

Assistenten			
PersNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

```
CREATE TABLE Assistenten
```

```
(
```

```
    PersNr INT PRIMARY KEY,
```

```
    Name VARCHAR(100) NOT NULL,
```

```
    Fachgebiet VARCHAR(100),
```

```
    Boss INT FOREIGN KEY (Boss) REFERENCES Professoren(PersNr)
```

```
        ON DELETE SET NULL
```

```
);
```


(Einfache) Datendefinition in SQL

CHAR(*n*) – Fixed length string

VARCHAR(*n*) – String with maximum length *n*

INTEGER – Integer

FLOAT – Floating point number

DATE – Date

BIT(*n*) – Bitstring of length *n*

BLOB – Binary large object

... und viele, viele, viele mehr

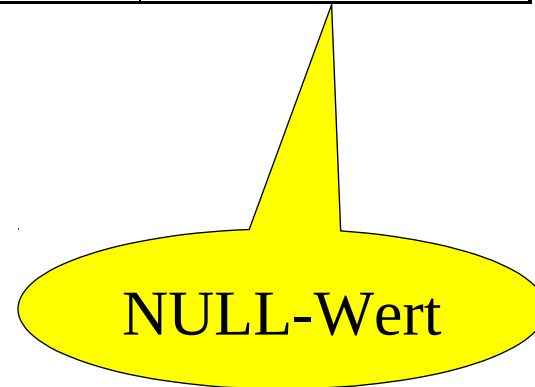
Veränderung am Datenbestand

Einfügen von Tupeln:

```
INSERT INTO Studenten (MatrNr, Name)  
          VALUES (28121, 'Archimedes');
```

```
INSERT INTO hören  
          SELECT MatrNr, VorlNr  
          FROM Studenten, Vorlesungen  
          WHERE Titel= 'Logik';
```

Studenten		
MatrNr	Name	Semester
⋮	⋮	⋮
29120	Theophrastos	2
29555	Feuerbach	2
28121	Archimedes	-



NULL-Wert

Auswertung bei Null-Werten

In arithmetischen Ausdrücken werden Nullwerte propagiert, d.h. sobald ein Operand **NULL** ist, wird auch das Ergebnis **NULL**. Dementsprechend wird z.B. **NULL** + 1 zu **NULL** ausgewertet- aber auch **NULL** * 0 wird zu **NULL** ausgewertet.

SQL hat eine dreiwertige Logik, die nicht nur **TRUE** und **FALSE** kennt, sondern auch einen dritten Wert **UNKNOWN**. Diesen Wert liefern Vergleichsoperationen zurück, wenn mindestens eines ihrer Argumente **NULL** ist. Beispielsweise wertet SQL das Prädikat (*PersNr=...*) immer zu **UNKNOWN** aus, wenn die *PersNr* des betreffenden Tupels den Wert **null** hat.

Logische Ausdrücke werden nach den folgenden Tabellen berechnet:

NOT	
TRUE	FALSE
UNKNOWN	UNKNOWN
FALSE	TRUE

AND	TRUE	UNKNOWN	FALSE
TRUE	TRUE	UNKNOWN	FALSE
UNKNOWN	UNKNOWN	UNKNOWN	FALSE
FALSE	FALSE	FALSE	FALSE

OR	TRUE	UNKNOWN	FALSE
TRUE	TRUE	TRUE	TRUE
UNKNOWN	TRUE	UNKNOWN	UNKNOWN
FALSE	TRUE	UNKNOWN	FALSE

Veränderungen am Datenbestand

Löschen von Tupeln

```
DELETE FROM Studenten  
WHERE Semester > 13;
```

Verändern von Tupeln

```
UPDATE Studenten  
SET Semester = Semester + 1;
```

Einfache SQL-Anfrage

```
SELECT PersNr, Name  
FROM Professoren  
WHERE Rang= 'C4';
```

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

PersNr	Name
2125	Sokrates
2126	Russel
2136	Curie
2137	Kant

Sortierung

```
SELECT PersNr, Name, Rang  
FROM Professoren  
ORDER BY Rang DESC, Name ASC;
```

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

PersNr	Name	Rang
2136	Curie	C4
2137	Kant	C4
2126	Russel	C4
2125	Sokrates	C4
2134	Augustinus	C3
2127	Kopernikus	C3
2133	Popper	C3

Duplikateneeliminierung

```
SELECT DISTINCT Rang  
FROM Professoren;
```

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Rang
C3
C4

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555		2

Vorlesungen			
VorlNr	Titel	SWS	gelesenVon
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022

Assistenten			
PerslNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

prüfen			
MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Anfragen über mehrere Relationen

Welcher Professor liest "Mäeutik"?

```
SELECT Name, Titel  
FROM Professoren, Vorlesungen  
WHERE PersNr = gelesenVon AND Titel =  
    'Mäeutik';
```

Anfragen über mehrere Relationen

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
⋮	⋮	⋮	⋮
2137	Kant	C4	7

Vorlesungen			
VorlNr	Titel	SWS	gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
⋮	⋮	⋮	⋮
5049	Mäeutik	2	2125
⋮	⋮	⋮	⋮
4630	Die 3 Kritiken	4	2137

↙ Kreuzprodukt ↘
×

PersNr	Name	Rang	Raum	VorlNr	Titel	SWS	gelesen Von
2125	Sokrates	C4	226	5001	Grundzüge	4	2137
1225	Sokrates	C4	226	5041	Ethik	4	2125
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2125	Sokrates	C4	226	5049	Mäeutik	2	2125
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2126	Russel	C4	232	5001	Grundzüge	4	2137
2126	Russel	C4	232	5041	Ethik	4	2125
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2137	Kant	C4	7	4630	Die 3 Kritiken	4	2137

↓ Selektion σ

PersNr	Name	Rang	Raum	VorlNr	Titel	SWS	gelesen Von
2125	Sokrates	C4	226	5049	Mäeutik	2	2125

↓ Projektion π

Name	Titel
Sokrates	Mäeutik

Anfragen über mehrere Relationen

Welche Studenten hören welche Vorlesungen?

```
SELECT Name, Titel  
FROM Studenten, hören, Vorlesungen  
WHERE Studenten.MatrNr = hören.MatrNr AND  
        hören.VorlNr = Vorlesungen.VorlNr;
```

Alternativ:

```
SELECT s.Name, v.Titel  
FROM Studenten s, hören h, Vorlesungen v  
WHERE s.MatrNr = h.MatrNr AND  
        h.VorlNr = v.VorlNr;
```

SELECT

s1.Name, s2.Name



FROM

Studenten s1, hören h1, hören h2, Studenten s2

WHERE

h1.VorlNr = h2.VorlNr **AND**

h1.MatrNr = s1.MatrNr **AND**

h2.MatrNr = s2.MatrNr;

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

hören	
MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten		
MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555		2

Vorlesungen			
VorINr	Titel	SWS	gelesenVon
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen	
Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören	
MatrNr	VorINr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022

Assistenten			
PersINr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

prüfen			
MatrNr	VorINr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Mengenoperationen

Mengenoperationen UNION, INTERSECT, MINUS

```
SELECT Name  
FROM Assistenten  
UNION  
SELECT Name  
FROM Professoren;
```

```
SELECT Name  
FROM Assistenten  
MINUS  
SELECT Name  
FROM Professoren;
```

```
SELECT Name  
FROM Assistenten  
INTERSECT  
SELECT Name  
FROM Professoren;
```

Geschachtelte Anfragen

```
SELECT Name  
FROM Professoren  
WHERE PersNr NOT IN ( SELECT gelesenVon  
                        FROM  
Vorlesungen );
```

```
SELECT p.Name  
FROM Professoren p  
WHERE NOT EXISTS ( SELECT gelesenVon  
                   FROM Vorlesungen v  
                   WHERE v.gelesenVon = p.PersNr );
```



...to be continued