

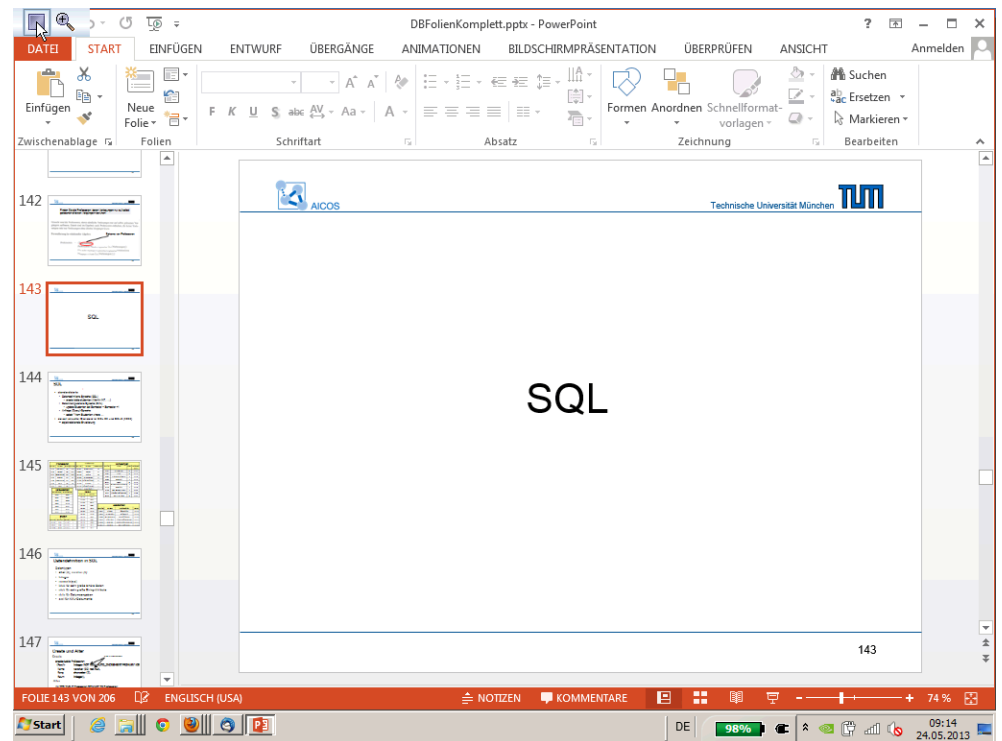
## Script generated by TTT

Title: profile1 (24.05.2013)

Date: Fri May 24 09:14:37 CEST 2013

Duration: 93:30 min

Pages: 64



SQL



## Create und Alter

### Create

```
create table Professoren  
(PersNr integer NOT NULL AUTO_INCREMENT PRIMARY KEY,  
Name varchar (30) not null,  
Rang character (2),  
Raum integer);
```

Integritätsbedingungen

### Alter

- ALTER TABLE** Proessoren **RENAME TO** Professoren;
- ALTER TABLE** Professoren **ADD** Familienstatus varchar(50);
- ALTER TABLE** Professoren **DROP COLUMN** Wohnort;
- ALTER TABLE** Professoren **MODIFY** (Name varchar(50));

•...

## Create und Alter

### Create

```
create table Professoren
(PersNr integer NOT NULL AUTO_INCREMENT PRIMARY KEY,
Name varchar (30) not null,
Rang character (2),
Raum integer);
```

Integritätsbedingungen

### Alter

- ALTER TABLE Proessoren RENAME TO Professoren;
- ALTER TABLE Professoren ADD Familienstatus varchar(50);
- ALTER TABLE Professoren DROP COLUMN Wohnort;
- ALTER TABLE Professoren MODIFY (Name varchar(50));



147

## Create und Alter

### Create

```
create table Professoren
(PersNr integer NOT NULL AUTO_INCREMENT PRIMARY KEY,
Name varchar (30) not null,
Rang character (2),
Raum integer);
```

Integritätsbedingungen

### Alter

- ALTER TABLE Proessoren RENAME TO Professoren;
- ALTER TABLE Professoren ADD Familienstatus varchar(50);
- ALTER TABLE Professoren DROP COLUMN Wohnort;
- ALTER TABLE Professoren MODIFY (Name varchar(50));



147

## Create und Alter

### Create

```
create table Professoren
(PersNr integer NOT NULL AUTO_INCREMENT PRIMARY KEY,
Name varchar (30) not null,
Rang character (2),
Raum integer);
```

Integritätsbedingungen

### Alter

- ALTER TABLE Proessoren RENAME TO Professoren;
- ALTER TABLE Professoren ADD Familienstatus varchar(50);
- ALTER TABLE Professoren DROP COLUMN Wohnort;
- ALTER TABLE Professoren MODIFY (Name varchar(50));



147

## Veränderung am Datenbestand

Einfügen von Tupeln:

**insert into** hören

**select** s.MatrNr, v.VorlNr

**from** Studenten s, Vorlesungen v;

**insert into** Studenten (MatrNr, Name)

**values** (28121, `Archimedes`);



148

## Einfache SQL-Anfrage

```
select  PersNr, Name
from    Professoren
where   Rang= 'C4';
```

PersNr	Name
2125	Sokrates
2126	Russel
2136	Curie
2137	Kant

## Einfache SQL-Anfrage

```
select  PersNr, Name
from    Professoren
where   Rang= 'C4';
```

PersNr	Name
2125	Sokrates
2126	Russel
2136	Curie
2137	Kant

## Einfache SQL-Anfrage

```
select  PersNr, Name
from    Professoren
where   Rang= 'C4';
```

PersNr	Name
2125	Sokrates
2126	Russel
2136	Curie
2137	Kant

## Sortierung von SQL-Anfragen

```
select p.PersNr, p.Name, p.Rang
from Professoren p
order by p.Rang desc, p.Name asc;
```

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

## Sortierung von SQL-Anfragen

**select** p.PersNr, p.Name, p.Rang  
**from** Professoren p  
**order by** p.Rang **desc**, p.Name **asc**;

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

## Sortierung von SQL-Anfragen

**select** p.PersNr, p.Name, p.Rang  
**from** Professoren p  
**order by** p.Rang **desc**, p.Name **asc**;

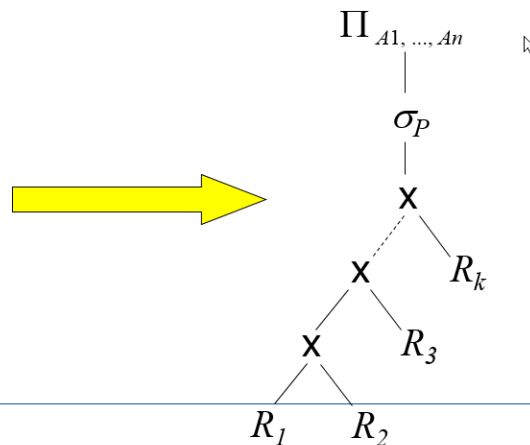
PersNr	Name	Rang
2136	Curie	C4
2137	Kant	C4
2126	Russel	C4
2125	Sokrates	C4
2134	Augustinus	C3
2127	Kopernikus	C3
2133	Popper	C3

## Übersetzung in die relationale Algebra

Allgemein hat eine (ungeschachtelte) SQL-Anfrage die Form:

**select**  $A_1, \dots, A_n$   
**from**  $R_1, \dots, R_k$   
**where**  $P_i$ ;

Übersetzung in die relationale Algebra:  $\Pi_{A_1, \dots, A_n}(\sigma_P(R_1 \times \dots \times R_k))$

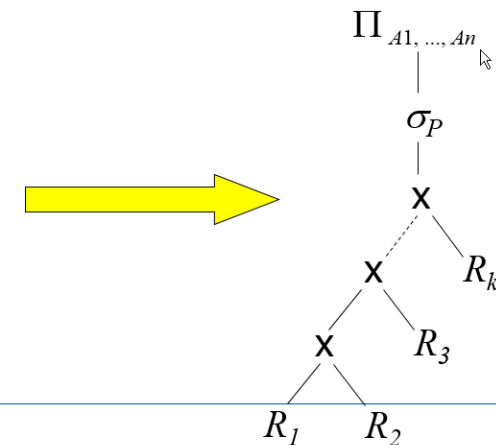


## Übersetzung in die relationale Algebra

Allgemein hat eine (ungeschachtelte) SQL-Anfrage die Form:

**select**  $A_1, \dots, A_n$   
**from**  $R_1, \dots, R_k$   
**where**  $P_i$ ;

Übersetzung in die relationale Algebra:  $\Pi_{A_1, \dots, A_n}(\sigma_P(R_1 \times \dots \times R_k))$



# Anfragen über mehrere Relationen

Welcher Prof liest Mäeutik?

```
select Name, Titel
    from Professoren, Vorlesungen
   where PersNr=gelesenVon and Titel='Mäeutik';
```



$$\Pi_{\text{Name, Titel}} \sigma_{\text{PersNr=gelesenVon} \wedge \text{Titel='Mäeutik'}}(\text{Professoren} \times \text{Vorlesungen})$$

(siehe auch Blatt 3 Aufgabe 1)

# Create und Alter

## Create

```
create table Professoren
(PersNr      integer NOT NULL AUTO_INCREMENT PRIMARY KEY,
 Name       varchar (30) not null,
 Rang       character (2),
 Raum       integer);
```

Integritätsbedingungen

## Alter

- ALTER TABLE Proessoren **RENAME TO** Professoren;
- ALTER TABLE Professoren **ADD** Familienstatus varchar(50);
- ALTER TABLE Professoren **DROP COLUMN** Wohnort;
- ALTER TABLE Professoren **MODIFY** (Name varchar(50));

# Anfragen über mehrere Relationen

Welcher Prof liest Mäeutik?

```
select Name, Titel
    from Professoren, Vorlesungen
   where PersNr=gelesenVon and Titel='Mäeutik';
```



$$\Pi_{\text{Name, Titel}} \sigma_{\text{PersNr=gelesenVon} \wedge \text{Titel='Mäeutik'}}(\text{Professoren} \times \text{Vorlesungen})$$

(siehe auch Blatt 3 Aufgabe 1)

# Anfragen über mehrere Relationen

Welcher Prof liest Mäeutik?

```
select Name, Titel
    from Professoren, Vorlesungen
   where PersNr=gelesenVon and Titel='Mäeutik';
```



$$\Pi_{\text{Name, Titel}} \sigma_{\text{PersNr=gelesenVon} \wedge \text{Titel='Mäeutik'}}(\text{Professoren} \times \text{Vorlesungen})$$

(siehe auch Blatt 3 Aufgabe 1)

PersN	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

↓ Auswahl

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

↓ Projektion

Name	Titel
Sokrates	Mäeutik



PersN	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

↓ Auswahl

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

↓ Projektion

Name	Titel
Sokrates	Mäeutik



PersN	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

↓ Auswahl

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

↓ Projektion

Name	Titel
Sokrates	Mäeutik



PersN	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

↓ Auswahl

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

↓ Projektion

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: (empfohlen --> Einsatz von Tupelvariablen)**

```
select s.Name, v.Titel
from Studenten s, hören h, Vorlesungen v
where s. MatrNr = h. MatrNr and
      h.VorlNr = v.VorlNr
```

## Anfragen über mehrere Relationen

Welcher Prof liest Mäeutik?

```
select Name, Titel
from Professoren, Vorlesungen
where PersNr=gelesenVon and Titel='Mäeutik';
```



$$\Pi_{\text{Name, Titel}} \sigma_{\text{PersNr=gelesenVon} \wedge \text{Titel='Mäeutik'}}(\text{Professoren} \times \text{Vorlesungen})$$

(siehe auch Blatt 3 Aufgabe 1)

## Anfragen über mehrere Relationen

Welcher Prof liest Mäeutik?

```
select Name, Titel
from Professoren, Vorlesungen
where PersNr=gelesenVon and Titel='Mäeutik';
```



$$\Pi_{\text{Name, Titel}} \sigma_{\text{PersNr=gelesenVon} \wedge \text{Titel='Mäeutik'}}(\text{Professoren} \times \text{Vorlesungen})$$

(siehe auch Blatt 3 Aufgabe 1)

## 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: (empfohlen --> Einsatz von Tupelvariablen)**

```
select s.Name, v.Titel
from Studenten s, hören h, Vorlesungen v
where s. MatrNr = h. MatrNr and
      h.VorlNr = v.VorlNr
```

## 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: (empfohlen --> Einsatz von Tupelvariablen)

```
select s.Name, v.Titel
from Studenten s, hören h, Vorlesungen v
where s. MatrNr = h. MatrNr and
      h.VorlNr = v.VorlNr
```

## 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: (empfohlen --> Einsatz von Tupelvariablen)

```
select s.Name, v.Titel
from Studenten s, hören h, Vorlesungen v
where s. MatrNr = h. MatrNr and
      h.VorlNr = v.VorlNr
```

## 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: (empfohlen --> Einsatz von Tupelvariablen)

```
select s.Name, v.Titel
from Studenten s, hören h, Vorlesungen v
where s. MatrNr = h. MatrNr and
      h.VorlNr = v.VorlNr
```

## Mengenoperationen und geschachtelte Anfragen

Mengenoperationen **union**, **intersect**, **minus**

```
( select Name
  from Assistenten )
union
( select Name
  from Professoren);
```



## Existenzquantor exists

Profs die keine Vorlesung halten:

```

select p.Name
from Professoren p
where not exists ( select *
                    from Vorlesungen v
                    where v.gelesenVon = p.PersNr );
    
```

## Existenzquantor exists

Profs die keine Vorlesung halten:

```

select p.Name
from Professoren p
where not exists ( select *
                    from Vorlesungen v
                    where v.gelesenVon = p.PersNr );
    
```

## Existenzquantor exists

Profs die keine Vorlesung halten:

```

select p.Name
from Professoren p
where not exists ( select *
                    from Vorlesungen v
                    where v.gelesenVon = p.PersNr );
    
```

## Existenzquantor exists

Profs die keine Vorlesung halten:

```

select p.Name
from Professoren p
where not exists ( select *
                    from Vorlesungen v
                    where v.gelesenVon = p.PersNr );
    
```

## Existenzquantor exists

Profs die keine Vorlesung halten:

```
select p.Name
from Professoren p
where not exists ( select *
                  from Vorlesungen v
                  where v.gelesenVon = p.PersNr );
```

*Korrelation*

## Mengenvergleich

Profs die keine Vorlesung halten (Variante mit 'in'):

```
select p.Name
from Professoren p
where p.PersNr not in ( select gelesenVon
                      from Vorlesungen );
```

Unkorrelierte Unteranfrage:  
meist effizienter, wird nur  
einmal ausgewertet

## Mengenvergleich

Profs die keine Vorlesung halten (Variante mit 'in'):

```
select p.Name
from Professoren p
where p.PersNr not in ( select gelesenVon
                      from Vorlesungen );
```

Unkorrelierte Unteranfrage:  
meist effizienter, wird nur  
einmal ausgewertet

## Mengenvergleich

Profs die keine Vorlesung halten (Variante mit 'in'):

```
select p.Name
from Professoren p
where p.PersNr not in ( select gelesenVon
                      from Vorlesungen );
```

Unkorrelierte Unteranfrage:  
meist effizienter, wird nur  
einmal ausgewertet

## Mengenvergleich

Profes die keine Vorlesung halten (Variante mit 'in'):

```
select p.Name
from Professoren p
where p.PersNr not in ( select gelesenVon
                        from Vorlesungen );
```

Unkorrelierte Unteranfrage:  
meist effizienter, wird nur  
einmal ausgewertet

## Der Vergleich mit "all"

```
select Name
from Studenten
where Semester >= all ( select Semester
                       from Studenten);
```

effizienter:  
...

```
where Semester >= ( select max(Semester)
                   from Studenten);
```

größer gleich jeder anderen  
Zahl der Menge

Bemerkung (1): neben **all** gibt es auch **any**

## Der Vergleich mit "all"

```
select Name
from Studenten
where Semester >= all ( select Semester
                       from Studenten);
```

effizienter:  
...

```
where Semester >= ( select max(Semester)
                   from Studenten);
```

größer gleich jeder anderen  
Zahl der Menge

Bemerkung (1): neben **all** gibt es auch **any**

## Der Vergleich mit "all"

```
select Name
from Studenten
where Semester >= all ( select Semester
                       from Studenten);
```

effizienter:  
...

```
where Semester >= ( select max(Semester)
                   from Studenten);
```

größer gleich jeder anderen  
Zahl der Menge

Bemerkung (1): neben **all** gibt es auch **any**

## Der Vergleich mit "all"

```
select Name
from Studenten
where Semester >= all ( select Semester
                        from Studenten);
```

größer gleich jeder anderen  
Zahl der Menge

effizienter:

```
...
where Semester >= ( select max(Semester)
                    from Studenten);
```

Bemerkung (1): neben **all** gibt es auch **any**



162

## Der Vergleich mit "all"

```
select Name
from Studenten
where Semester >= all ( select Semester
                        from Studenten);
```

größer gleich jeder anderen  
Zahl der Menge

effizienter:

```
...
where Semester >= ( select max(Semester)
                    from Studenten);
```

Bemerkung (1): neben **all** gibt es auch **any**



162

## Der Vergleich mit "all"

```
select Name
from Studenten
where Semester >= all ( select Semester
                        from Studenten);
```

größer gleich jeder anderen  
Zahl der Menge

effizienter:

```
...
where Semester >= ( select max(Semester)
                    from Studenten);
```

Bemerkung (1): neben **all** gibt es auch **any**



162

## Aggregatfunktion und Gruppierung

Aggregatfunktionen: **avg**, **max**, **min**, **count**, **sum**, **median**

- **select avg(Semester) from Studenten;**

Gruppierung

- **select gelesenVon, sum (SWS)**  
**from Vorlesungen**  
**group by gelesenVon;**



164

## Gruppierung und Aggregation

Bsp.: Zähle pro Semester-Wert die Zahl der betr. Studenten

$\gamma_{\text{Semester}; \text{count}^*}(\text{Studenten})$

$\gamma_{\text{Semester}; \text{count}^*}(\text{Studenten})$	
Semester	count(*)
18	1
12	1
10	1
8	1
6	1
3	1
2	2

## Aggregatfunktion und Gruppierung

Aggregatfunktionen: **avg, max, min, count, sum, median**

- **select avg(Semester) from Studenten;**

Gruppierung

- **select gelesenVon, sum (SWS)**  
**from Vorlesungen**  
**group by gelesenVon;**

## Gruppierung und Aggregation

Bsp.: Zähle pro Semester-Wert die Zahl der betr. Studenten

$\gamma_{\text{Semester}; \text{count}^*}(\text{Studenten})$

$\gamma_{\text{Semester}; \text{count}^*}(\text{Studenten})$	
Semester	count(*)
18	1
12	1
10	1
8	1
6	1
3	1
2	2

## Aggregatfunktion und Gruppierung

Aggregatfunktionen: **avg, max, min, count, sum, median**

- **select avg(Semester) from Studenten;**

Gruppierung

- **select gelesenVon, sum (SWS)**  
**from Vorlesungen**  
**group by gelesenVon;**

# Aggregatfunktion und Gruppierung

Aggregatfunktionen: avg, max, min, count, sum, median

- `select avg(Semester) from Studenten;`

Gruppierung

- `select gelesenVon, sum (SWS)  
from Vorlesungen  
group by gelesenVon;`

Professoren				Studenten			Vorlesungen			
PersNr	Name	Rang	Raum	MatrNr	Name	Semester	VorlNr	Titel	SWS	gelesen Von
2125	Sokrates	C4	226	24002	Xenokrates	18	5001	Grundzüge	4	2137
2126	Russel	C4	232	25403	Jonas	12	5041	Ethik	4	2125
2127	Kopernikus	C3	310	26120	Fichte	10	5043	Erkenntnistheorie	3	2126
2133	Popper	C3	52	26830	Aristoxenos	8	5049	Mäeutik	2	2125
2134	Augustinus	C3	309	27550	Schopenhauer	6	4052	Logik	4	2125
2136	Curie	C4	36	28106	Carnap	3	5052	Wissenschaftstheorie	3	2126
2137	Kant	C4	7	29120	Theophrastos	2	5216	Bioethik	2	2126
				29555	Feuerbach	2	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
25403	5022

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

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

Professoren				Studenten			Vorlesungen			
PersNr	Name	Rang	Raum	MatrNr	Name	Semester	VorlNr	Titel	SWS	gelesen Von
2125	Sokrates	C4	226	24002	Xenokrates	18	5001	Grundzüge	4	2137
2126	Russel	C4	232	25403	Jonas	12	5041	Ethik	4	2125
2127	Kopernikus	C3	310	26120	Fichte	10	5043	Erkenntnistheorie	3	2126
2133	Popper	C3	52	26830	Aristoxenos	8	5049	Mäeutik	2	2125
2134	Augustinus	C3	309	27550	Schopenhauer	6	4052	Logik	4	2125
2136	Curie	C4	36	28106	Carnap	3	5052	Wissenschaftstheorie	3	2126
2137	Kant	C4	7	29120	Theophrastos	2	5216	Bioethik	2	2126
				29555	Feuerbach	2	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
25403	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

# Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

- `select gelesenVon, Name, sum(SWS)  
from Vorlesungen, Professoren  
where gelesenVon = PersNr and Rang = 'C4'  
group by gelesenVon, Name  
having avg(SWS) >= 3;`

## Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

```
• select gelesenVon, Name, sum(SWS)
from Vorlesungen, Professoren
where gelesenVon = PersNr and Rang = 'C4'
group by gelesenVon, Name
having avg(SWS) >= 3;
```

## Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

```
• select gelesenVon, Name, sum(SWS)
from Vorlesungen, Professoren
where gelesenVon = PersNr and Rang = 'C4'
group by gelesenVon, Name
having avg(SWS) >= 3;
```

## Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

```
• select gelesenVon, Name, sum(SWS)
from Vorlesungen, Professoren
where gelesenVon = PersNr and Rang = 'C4'
group by gelesenVon, Name
having avg(SWS) >= 3;
```

## Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

```
• select gelesenVon, Name, sum(SWS)
from Vorlesungen, Professoren
where gelesenVon = PersNr and Rang = 'C4'
group by gelesenVon, Name
having avg(SWS) >= 3;
```

## Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

- **select** gelesenVon, Name, **sum**(SWS)  
**from** Vorlesungen, Professoren  
**where** gelesenVon = PersNr **and** Rang = 'C4'  
**group by** gelesenVon, Name  
**having avg**(SWS) >= 3;

## Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

- **select** gelesenVon, Name, **sum**(SWS)  
**from** Vorlesungen, Professoren  
**where** gelesenVon = PersNr **and** Rang = 'C4'  
**group by** gelesenVon, Name  
**having avg**(SWS) >= 3;

## Aggregatfunktion und Gruppierung

Welcher C4 Professor liest im Durchschnitt lange Vorlesungen?

- **select** gelesenVon, Name, **sum**(SWS)  
**from** Vorlesungen, Professoren  
**where** gelesenVon = PersNr **and** Rang = 'C4'  
**group by** gelesenVon, Name  
**having avg**(SWS) >= 3;

↑

Start typing SQL above  
and let the magic begin.



Firefox

SQL Interface, muehe version

www-db.in.tum.de/~muehe/sql/

```
select * from Studenten;
```

Execute

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

Auto execute  | [load](#) | [save](#) | [uni](#) | [tables](#) | A 120 minute hack by Henrik Mühe (muehe@in.tum.de) | If you encounter a bug, use

Start | DE | 100% | 10:47 24.05.2013