

SQL Wiederholung

Datenbanktechnologien

Prof. Dr. Ingo Claßen Prof. Dr. Martin Kempa

Hochschule für Technik und Wirtschaft Berlin

Beispieldatenmodell

Verbunde

Aggregation und Gruppierung

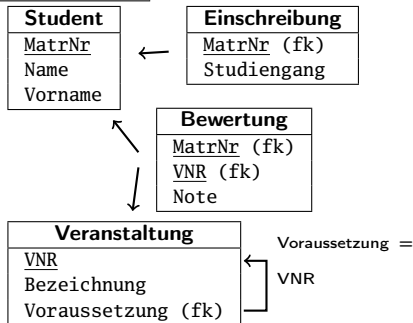
Unterabfragen

Beispieldatenmodell

Student		
<u>MatrNr</u>	Name	Vorname
50101	Svenson	Sven
50102	Clausen	Claus
50103	Jensen	Jens
50104	Hansen	Hans

Einschreibung	
<u>MatrNr</u>	Studiengang
50101	WI
50102	AI
50103	WM

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7



```
alter session set current_schema=ustud;
```

Veranstaltung		
<u>VNR</u>	Bezeichnung	Voraussetzung
99001	Datenmodellierung	
99002	Datenbanktechnologien	99001
99003	DB-Anwendungssysteme	99002

Inner-Join

- ▶ Nur verbundene Datensätze erscheinen im Ergebnis

Student		
<u>MatrNr</u>	Name	Vorname
50101	Svenson	Sven
50102	Clausen	Claus
50103	Jensen	Jens
50104	Hansen	Hans

Einschreibung	
<u>MatrNr</u>	Studiengang
50101	WI
50102	AI
50103	WM

```

select s.MatrNr, Name, Studiengang
from Student s
inner join Einschreibung e on s.MatrNr = e.MatrNr;
  
```

```

MATRNR|NAME      |STUDIENGANG|
-----|-----|-----|
50101|Svenson|WI          |
50102|Clausen|AI          |
50103|Jensen |WM          |
  
```

Left-Outer-Join

- ▶ Alle Zeilen der linken Tabelle erscheinen im Ergebnis
- ▶ Nullwerte in Spalten der rechten Tabelle ohne Verbindung

Student		
<u>MatrNr</u>	Name	Vorname
50101	Svenson	Sven
50102	Clausen	Claus
50103	Jensen	Jens
50104	Hansen	Hans

Einschreibung	
<u>MatrNr</u>	Studiengang
50101	WI
50102	AI
50103	WM

```
select s.MatrNr, Name, Studiengang
from Student s
left outer join Einschreibung e on s.MatrNr = e.MatrNr;
```

```
MATRNR|NAME    |STUDIENGANG|
-----|-----|-----|
50101|Svenson|WI         |
50102|Clausen|AI         |
50103|Jensen |WM         |
50104|Hansen |           |
```

Theta-Join

- ▶ Statt = sind auch andere Vergleichsoperatoren erlaubt, z.B >

Student		
<u>MatrNr</u>	Name	Vorname
50101	Svenson	Sven
50102	Clausen	Claus
50103	Jensen	Jens
50104	Hansen	Hans

Einschreibung	
<u>MatrNr</u>	Studiengang
50101	WI
50102	AI
50103	WM

```

select s.MatrNr, Name, Studiengang
from Student s
      inner join Einschreibung e on s.MatrNr > e.MatrNr;
  
```

```

MATRNR|NAME    |STUDIENGANG|
-----|-----|-----|
50104|Hansen |WM          |
50104|Hansen |AI          |
50104|Hansen |WI          |
50103|Jensen |AI          |
50103|Jensen |WI          |
50102|Clausen|WI          |
  
```

Self-Join

- ▶ Verbindung einer Tabelle mit sich selbst
- ▶ Gleiche Tabelle mit verschiedenen Alias-Namen

Veranstaltung		
<u>VNR</u>	Bezeichnung	Voraussetzung
99001	Datenmodellierung	
99002	Datenbanktechnologien	99001
99003	DB-Anwendungssysteme	99002

```
select v1.VNR, v1.Bezeichnung, v2.Bezeichnung Vorkenntnis
from Veranstaltung v1
      left join Veranstaltung v2 on v1.Voraussetzung = v2.VNR;
```

```
VNR | BEZEICHNUNG | VORKENNTNIS |
----|-----|-----|
99001|Datenmodellierung | |
99002|Datenbanktechnologien|Datenmodellierung |
99003|DB-Anwendungssysteme |Datenbanktechnologien|
```

Aggregation

- ▶ Zusammenfassung mehrerer Zeilen zu einer Zeile
 - ▶ avg – Mittelwert
 - ▶ count – Anzahl Zeilen
 - ▶ max – Maximum
 - ▶ min – Minimum
 - ▶ sum – Summe

Gruppierung

- ▶ Aufteilung Daten einer Tabelle in Gruppen

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```

select VNR, avg(Note) AvgNote,
         max(Note) MaxNote, min(Note) MinNote
from Bewertung
group by VNR
order by VNR;

```

VNR	AVGNOTE	MAXNOTE	MINNOTE
99001	2	2.7	1
99002	2	2.7	1.3
99003	3	3	3

Bedingungen auf Gruppenebene

- ▶ Having-Klausel ist wie Where-Klausel, nur für Gruppen

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```

select
  VNR, avg(Note) AvgNote,
  count(*) AnzBewertungen
from Bewertung
group by VNR
having count(note) > 2;
  
```

```

VNR  |AVGNOTE|ANZBEWERTUNGEN|
-----|-----|-----|
99001|      2|                3|
  
```

Unterabfrage als abgeleitete Tabelle

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```

select VNR, AvgNote, AnzBewertungen
from (
  select VNR,
    count(*) AnzBewertungen,
    avg(NOTE) AvgNote
  from Bewertung
  group by VNR)
where AnzBewertungen > 2;

```

```

VNR |AVGNOTE|ANZBEWERTUNGEN|
-----|-----|-----|
99001|      2|                3|

```

Unterabfrage in Bedingung

- ▶ Degenerierte Tabelle: Eine Spalte, eine Zeile
- ▶ Wird als Wert interpretiert

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```

select *
from Bewertung
where Note <= (
    select avg(Note)
    from Bewertung);
  
```

```

MATRNR|VNR  |NOTE|
-----|-----|----|
 50101|99001|  1|
 50101|99002| 1.3|
  
```

Korrelierte Unterabfrage in Bedingung

- ▶ Zeilen der Unterabfrage beziehen sich auf Zeilen der Oberabfrage
- ▶ Entspricht Schleife:
Auswertung der Unterabfrage für jeden Datensatz der Oberabfrage

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```

select *
from Bewertung b1
where Note = (
    select min(Note)
    from Bewertung b2
    where b2.VNR = b1.VNR);
  
```

```

MATRNR|VNR  |NOTE|
-----|-----|----|
50101|99001|  1|
50101|99002| 1.3|
50102|99003|  3|
  
```

Korrelierte Unterabfrage in Ausgabespalte

- Kombination aggregierter und nicht aggregierter Werte

Bewertung		
MatrNr	VNR	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```

select distinct VNR, Note, (
    select avg(Note)
    from Bewertung b1
    where b1.VNR = b2.VNR) AvgNote
from Bewertung b2
order by VNR;

```

```

VNR | NOTE | AVGNOTE |
-----|-----|-----|
99001| 1 | 2 |
99001| 2.3 | 2 |
99001| 2.7 | 2 |
99002| 1.3 | 2 |
99002| 2.7 | 2 |
99003| 3 | 3 |

```

In-Prädikat

- ▶ Unterabfrage muss Liste von Werten liefern

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

Einschreibung	
<u>MatrNr</u>	Studiengang
50101	WI
50102	AI
50103	WM

```

MATRNR|VNR  |NOTE|
-----|-----|----|
50101|99001|  1|
50101|99002| 1.3|
    
```

```

select *
from Bewertung b
where MatrNr in (
    select Matrnr
    from Einschreibung e
    where
        b.MatrNr = e.MatrNr and
        e.Studiengang = 'WI');
    
```

All-Prädikat

- ▶ Unterabfrage muss Liste von Werten liefern
- ▶ Bedingung muss für alle Werte der Liste gelten

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```

select MatrNr
from Bewertung
where Note <=
    all (select Note
         from Bewertung);
  
```

```

MATRNR |
-----|
50101 |
  
```

Some-Prädikat

- ▶ Unterabfrage muss Liste von Werten liefern
- ▶ Bedingung muss für mindestens einen Wert der Liste gelten

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

Einschreibung	
<u>MatrNr</u>	Studiengang
50101	WI
50102	AI
50103	WM

```

MATRNR|VNR  |NOTE|
-----|-----|----|
50102|99002| 2.7|
50103|99001| 2.7|
    
```

```

select *
from Bewertung b1
where Note = some (
    select Note
    from Bewertung b2
    join Einschreibung e
    on b2.MatrNr = e.MatrNr
where e.Studiengang = 'WM');
    
```


Exists-Prädikat

- ▶ Bedingung trifft zu, wenn Unterabfrage mindestens einen Datensatz liefert. Inhalt des Datensatzes unerheblich

Bewertung		
<u>MatrNr</u>	<u>VNR</u>	Note
50101	99001	1.0
50101	99002	1.3
50102	99001	2.3
50102	99002	2.7
50102	99003	3.0
50103	99001	2.7

```
MATRNR|DNOTE|
-----|-----|
50101| 1.15|
50102| 2.67|
```

```
with
  dnotestud as (
    select MatrNr, avg(Note) as DNote
    from bewertung
    group by MatrNr)
select
  dn1.MatrNr,
  round(dn1.DNote,2) as DNote
from dnotestud dn1
where exists (
  select * from
  dnotestud dn2
  where dn1.DNote < dn2.DNote);
```