

# Seminar 1-2

## Übungen

# Datenbank

Studenten(MatrnNr, Name, Vorname, Email, Age, Gruppe)

Kurse(KursId, Titel, ECTS)

Enrolled(MatrnNr, KursId, Note)

1. Gebe alle Studenten aus der Gruppe 331 aus.

```
select *  
from Studenten s  
where s.gruppe = 331
```

2. Gebe alle Studenten aus, die in der Gruppe 331 oder 332 sind.

```
select *  
from Studenten s  
where s.gruppe = 331 OR s.gruppe= 332
```

3. Gebe alle Studenten aus, deren Name mit „An“ anfängt.

```
select *  
from Studenten s  
where s.name like 'An%'
```

- Gebe alle Studenten aus, deren Name mit „An“ anfängt und genau vier Buchstaben hat.

```
select *  
from Studenten s  
where s.name like 'An____'
```

4. Gebe alle Studenten aus (MatrNr), die in dem Kurs “Alg1” und “DB1” angemeldet sind.

- Methode I.

```
select e.MatrNr
from Enrolled e
where e.kursId = 'Alg1'
```

**INTERSECT**

```
select MatrNr
from Enrolled e
where e.kursId = 'DB1'
```

- Methode II.

```
select e1.MatrNr
from Enrolled e1, Enrolled e2
where e1.KursId = 'Alg1' and
e2.KursId = 'DB1' and
e1.MatrNr = e2.MatrNr
```

5. Gebe alle Studenten aus mit Alter  $\geq$  allen anderen mit dem Namen 'A...N'(die am ältesten sind)

```
select *  
from Studenten s  
where s.age  $\geq$  ALL (select s2.age  
                     from studenten s2  
                     where s2.name like 'A%N')
```

## 6. Gebe die Emails aller Studenten aus dem Kurs „DB1“ aus. – 3 Methoden

- Methode I.

```
select s.email
from studenten s
where s.matrNr in (SELECT e.MatrNr
                  from enrolled e
                  where e.KursId = 'DB1')
```

- Methode II.

```
select s.email
from studenten s
where exists (select *
             from enrolled e
             where e.MatrNr = s.MatrNr
             and e.KursId = 'DB1')
```



## 6. Gebe die Emails aller Studenten aus dem Kurs „DB1“ aus. – 3 Methoden

- Methode III.

```
select s.email  
from studenten s, enrolled e  
where s.MatrNr = e.MatrNr and e.KursId = 'DB1'
```

## 7. Gebe die Emails, Kurse und Noten aller Studenten aus.

- 2 Methoden (mit INNER JOIN und ohne)
- Methode I.

```
select s.email, e.KursId, e.note
from Studenten s
INNER JOIN Enrolled e
ON s.MatrNr = e.MatrNr
```

- Methode II.

```
select s.email, e.KursId, e.note
from Studenten s, Enrolled e
where s.MatrNr = e.MatrNr
```

8. Gebe die Emails der Studenten und die ECTS der Kurse in denen diese angemeldet sind aus. (mit INNER JOIN)

```
select s.email, k.ECTS
from Studenten s INNER JOIN Enrolled e
on s.MatrNr=e.MatrNr
INNER JOIN Kurse k
on e.KursId = k.KursId
```

## 9. Anzahl von Studenten für jede Altersgruppe deren Namen mit „A“ anfängt

```
SELECT COUNT(*) as StudentenNr, S.age  
FROM Studenten S  
WHERE S.Name like 'A%'  
GROUP BY S.Age
```

10. Für jeden Student, der in wenigstens 2 Kurse angemeldet ist, gebe die Anzahl der Kurse in denen dieser angemeldet ist aus

```
SELECT count(*), S.MatrNr
FROM Studenten S, Enrolled E
WHERE S.MatrNr = E.MatrNr
GROUP BY S.MatrNr
HAVING count(*) > 1
```

# Datenbank

Kunde (Kundeld, Name, Vorname, Stadt, Land, Tel)

Order (OrderNr, Kundeld, OrderDatum, , Preis)

# 11. Anzahl der Kunden in jedem Land mit wenigstens 10 Kunden

```
SELECT COUNT(KundeId) , Land  
FROM Kunde  
GROUP BY Land  
HAVING COUNT(KundeId) >10
```

12. Anzahl der Kunden in jedem Land außer USA,  
mit wenigstens 9 Kunden, sortiert absteigend

```
SELECT COUNT(KundeId) , Land
FROM Kunde
where Land <> 'USA'
GROUP BY Land
Having COUNT(KundeId) >=9
ORDER BY Count(KundeId) DESC
```



## 13. Alle Kunden mit dem Durchschnitt von Orders zwischen 1000 und 1200

```
SELECT AVG(Preis), Name, Vorname  
FROM Order O join Kunde K ON O.KundeId = K.  
KundeId  
GROUP BY Name, Vorname  
HAVING AVG(Preis) BETWEEN 1000 AND 1200
```