

Grundlagen Wirtschaftsinformatik

Sommersemester 2022

Übungsblatt 4

Aufgabe 1 (SQL 1)

Der Datenbankserver enthält u.a. die Datenbank *planetexpress*, in der die Mitarbeiter des inter-galaktischen Lieferdienstes eingepflegt sind. Sie sollen nun ein paar SQL-Befehle benutzen, um die folgenden Abfragen zu beantworten. Nutzen Sie dafür den Zugang die URL

<https://datascience.hs-bochum.de/sql/>

und wählen Sie dort oben links die Datenbank **planetexpress** aus.

1. Selektieren Sie aus der Datenbank alle Mitarbeiter, die in Raum Nr. 2 arbeiten.

```
select * from mitarbeiter where raum = 2
```

2. Finden Sie die Werte der Spalte **Aufgabe** der Mitarbeiter-Tabelle heraus.

```
select distinct aufgabe from mitarbeiter
```

3. Welcher Mitarbeiter ist der Chef? Schreiben Sie einen SQL-Befehl, der den entsprechenden Mitarbeiter ergibt.

```
select * from mitarbeiter where aufgabe = 'Chef'
```

Aufgabe 2 (SQL 2)

Die Tabelle **mitarbeiter_rolle** enthält die Zuordnung von Mitarbeitern zu ihrer Rolle im Unternehmen.

1. Finden Sie alle Mitarbeiter, die zur Rolle **Crew** gehören.

```
select * from mitarbeiter, mitarbeiter_rolle, rolle
where
    mitarbeiter.nr = mitarbeiter_rolle.mitarbeiter
    and mitarbeiter_rolle.rolle = rolle.nr
    and rolle.name = 'Crew'
```

2. Finden Sie alle Mitarbeiter, die zur Rolle **Management** gehören und die zusätzlich in der Etage **Ganz oben** arbeiten.

```
select * from mitarbeiter, mitarbeiter_rolle, rolle, raum
where
    mitarbeiter.nr = mitarbeiter_rolle.mitarbeiter
and mitarbeiter_rolle.rolle = rolle.nr
and mitarbeiter.raum = raum.nr
and rolle.name = 'Management'
and etage = 'Ganz oben'
```

Alternativ mit der JOIN Syntax:

```
select * from mitarbeiter
join mitarbeiter_rolle
    on mitarbeiter.nr = mitarbeiter_rolle.mitarbeiter
join rolle on mitarbeiter_rolle.rolle = rolle.nr
join raum on mitarbeiter.raum = raum.nr

where
    rolle.name = 'Management' and raum.etage = 'Ganz
                                oben'
```

Verwaltungsdatenbank hochschul_db

Für die folgenden Aufgaben nutzen Sie bitte die fiktiven Datenbank **hochschul_db**, die ebenfalls über ein Web-Interface erreicht werden kann. Die Adresse für den Zugang ist: Nutzen Sie dafür den Zugang unter:

https://datascience.hs-bochum.de/sql/db/hochschul_db/

Aufgabe 3

Bei den folgenden Aufgaben geht es zunächst um einfache Abfragen von Zeilen und Spalten.

1. Selektieren Sie alle Einträge (Zeilen und Spalten) der Tabelle *Abschluss*.

```
select * from abschluss
```

2. Selektieren Sie nur die Vornamen und Nachnamen aller Studenten der Datenbank.

```
select vorname, nachname from student
```

3. Selektieren Sie die Matrikelnummer sowie den Vor- und Nachnamen aller Studenten aus dem 3. Hochschulsemester.

```
select matrikelnr, vorname, nachname  
from student  
where semester = 3
```

4. Selektieren Sie die Matrikelnummer, den Vor- und Nachnamen und das Hochschulsemester aller Studenten die sich im 2. oder 3. Hochschulsemester befinden.

```
select matrikelnr, vorname, nachname  
from student  
where semester = 2 or semester = 3
```

Alternative mit dem IN Operator:

```
select matrikelnr, vorname, nachname  
from student  
where semester IN (2, 3)
```

5. Selektieren Sie die Titel sowie die ECTS aller Studiengänge, die zwischen 100 und 200 ECTS benötigen.

```
select * from studiengang  
where ects >= 100 and ects <= 200
```

Alternative mit BETWEEN:

```
select * from studiengang  
where ects between 100 and 200
```

6. Selektieren Sie den Vor- und Nachnamen aller Studenten, deren Nachname mit **S** beginnt.

```
select vorname, nachname from student  
where nachname like 'S%'
```

7. Selektieren Sie die Titel aller Studiengänge, die 120 oder 180 ECTS benötigen. Verwenden Sie für die Formulierung der Abfrage das IN-Prädikat.

```
select title from studiengang  
where ects IN (120, 180)
```

8. Selektieren Sie den Vor- und Nachnamen aller Studenten. Sortieren Sie das Ergebnis einmal aufsteigend nach dem Nachnamen und bei gleichem Nachnamen absteigend nach dem Vornamen.

```
select vorname, nachname from student  
order by nachname asc, vorname desc
```

9. Selektieren Sie den Vor- und Nachnamen aller Studenten, deren Nachname mit **S** nicht jedoch mit **Sch** beginnt und die sich zusätzlich nicht im 3. Semester befinden, sortieren Sie das Ergebnis absteigend nach dem Nachnamen.

Aufgabe 4

1. Selektieren Sie alle Termine, an denen mindestens eine Prüfung stattfindet. Geben Sie jeden Termin nur einmal aus.

```
select distinct datum from pruefung
```

2. Wie viele Prüfungen sind insgesamt aktuell in der Datenbank hinterlegt?

```
select count(id) from pruefung
```

Alternativ:

```
select count(*) from pruefung
```

3. In welchem Hochschulsemester befinden sich die Studenten der Hochschule durchschnittlich / maximal / minimal? Benutzen Sie eine einzelne Abfrage für die Beantwortung der Frage!

```
select avg(semester), max(semester), min(semester)  
from student
```

4. Wie viele Prüfungen finden je Datum statt (Ausgabe: Datum, AnzahlPruefungen)?

```
select datum, count(*) as AnzahlPruefungen  
from pruefung  
group by datum
```

5. Selektieren Sie die Prüfungstage, an denen mehrere Prüfungen stattfinden.

```
select datum from pruefung  
group by datum  
having count(*) > 1
```

6. Selektieren Sie die Menge von Studenten je Hochschulsemester.

```
select semester, count(*) from student  
group by semester
```

7. Selektieren Sie die Studenten, die bereits am längsten studieren (Matrikelnummer, Vorname, Nachname, Hochschulsemester). (Hinweis: Bitte auf die Fragestellung achten, eine Abfrage, die sich auf eine konkrete Semesterzahl bezieht ist falsch!)

```
select matrikelnr, nachname, vorname from student  
where semester = (select max(semester) from student)
```

Aufgabe 5

In den folgenden Aufgaben geht es um Abfragen, die sich über mehrere Tabellen erstrecken.

1. Geben Sie zu jedem Prüfungstermin aus der Tabelle Pruefung den entsprechenden Titel der Vorlesung aus. (Spalten: Datum, Titel). Nutzen Sie für den Abfrage beide vorgestellte Varianten des Joins.

(Es sind also 2 Abfragen zu schreiben, je eine pro Variante)

```
select datum, titel
  from pruefung, vorlesung
 where pruefung.vorlesung = vorlesung.id
```

Alternative mit JOIN:

```
select datum, titel
  from pruefung
 join vorlesung on pruefung.vorlesung = vorlesung.id
```

2. Geben Sie zu jedem Studiengang die Bezeichnung des Abschlusses aus. Benutzen Sie wieder beide Varianten des Joins.

```
select titel, bezeichnung
  from studiengang, abschluss
 where studiengang.abschluss = abschluss.id
```

Alternative mit JOIN:

```
select titel, bezeichnung
  from studiengang
 join abschluss on studiengang.abschluss = abschluss.id
```

3. Selektieren Sie den Titel sowie alle Prüfungstermine, die zur Vorlesung *Grundlagen der Wirtschaftsinformatik* gehören. Nutzen Sie für den Abfrage beide vorgestellte Varianten des Joins.

(Spalten: Vorlesungstitel, Prüfungsdatum)

```
select titel, datum
  from pruefung, vorlesung
 where pruefung.vorlesung = vorlesung.id
       and vorlesung.title = 'Grundlagen der
                               Wirtschaftsinformatik'
```

Alternative mit JOIN:

```
select titel, datum
  from pruefung
 join vorlesung on pruefung.vorlesung = vorlesung.id
 where vorlesung.title = 'Grundlagen der
                               Wirtschaftsinformatik'
```

4. Selektieren Sie die Vornamen und Nachnamen aller Studenten, sowie deren zugehörige Studiengänge samt Titel und eventueller Historie (Hier ist erstmals eine Verknüpfung von 3 Tabellen nötig)

(Spalten: Vorname | Nachname | Studiengang | von | bis).

Nutzen Sie auch hier die Chance, die Abfrage in beiden möglichen Ausprägungen zu formulieren

```
select vorname, nachname, titel, seit, bis
  from studienangstudent, studienang, student

  where
    studienangstudent.studienang = studienang.id
  and studienangstudent.student = student.matrikelnr
```

Alternative mit JOIN:

```
select vorname, nachname, titel, seit, bis
  from studienangstudent

  join studienang
    on studienangstudent.studienang = studienang.id

  join student
    on studienangstudent.student = student.matrikelnr
```