

# PVS

## Übungsblatt 9

Luca Krüger  
luca.krueger@uni-ulm.de

zum 20.06.2018

### Aufgabe 1

1. Alle Lieferanten, die nicht aus München kommen.

```
SELECT LiefName,LiefStadt
FROM lieferanten
WHERE NOT LiefStadt='Mnchen'
```

2. Alle Teilenummern der Liefert-Tabelle. Achten Sie darauf dass jede Nummer nur einmal aufgeführt wird.

```
SELECT TeileID AS TeileNummer
FROM liefert
GROUP BY TeileID
```

3. Farbe, TeileNr und Preis aus der Preisliste-Tabelle. Es sollen nur die Einträge ausgegeben werden, bei denen der Preis zwischen 300 und 430 liegt.

```
SELECT Farbe,TeileID,Preis
FROM preisliste
WHERE Preis<430 AND Preis>300
```

4. Alle Bestellungen vor dem 15.02.2009 und nach dem 23.02.2009.

```
SELECT *
FROM bestellungen
WHERE BestDatum>'15.02.2009' AND BestDatum<'23.02.2009'
```

5. Alle Informationen zu Kunden, die aus einer Stadt kommen, die mit einem S beginnt.

```
SELECT *
FROM kunden
WHERE KdName LIKE 's%'
```

6. Den Preis des jeweiligen Teils in der Teile-Tabelle mit ihrer Nummer und Farbe.

```
SELECT TeileID,Farbe,Preis
FROM teile NATURAL JOIN preisliste
```

7. Das Gehalt aller Mitarbeiter mit Funktion Montage II aus Ulm; aufsteigend geordnet.

```
SELECT Gehalt
FROM mitarbeiter
WHERE Funktion='Montage II'
ORDER BY Gehalt
```

8. Alle Informationen zu Aufträgen sowie das berechnete Skontodatum. Das Skontodatum liegt 3 Monate nach Auftragsdatum. Hinweis: Abweichung vom Skript! datum + INTERVAL n MONTH anstatt datum + n months!

```
SELECT AuftrNr,KdNr,AuftrDatum, AuftrDatum + INTERVAL 3
        MONTH AS SkontoDatum
From auftraege
```

9. Alle Aufträge mit Nummer und Datum, die an einem geraden Tagesdatum getätigt wurden (z.B. 02.xx, 04.xx, 06.xx . . .).

```
SELECT AuftrNr,AuftrDatum
FROM auftraege
WHERE day(AuftrDatum) % 2 = 0
```

10. Geben Sie eine Bestandsbewertung für die vorrätigen Teile in der Teilerelation. Geben sie dazu die Attribute zu den Teilen aus sowie eine Bewertung nach folgendem Schema: Wenn der Bestand größer als der Mindestbestand ist, geben sie Überbestand aus, wenn beide gleich sind Mindestbestand, wenn zu wenig Teile da sind Mangel.

```
SELECT TeileID,Farbe,KalkKosten,Bestand,MinBestand,
(CASE
    WHEN Bestand>MinBestand THEN 'Ueberbestand'
    WHEN Bestand<MinBestand THEN 'Mangel'
    WHEN Bestand=MinBestand THEN 'Mindestbestand'
END)
AS Bestandsbewertung
FROM teile
```

## Aufgabe 2

1. Alle Abteilungen mit den Namen und Adressen ihrer Manager.

```
SELECT a.AbtNr,a.AbtName,a.MgrPersNr,m.Name,m.Wohnort
FROM abteilungen AS a LEFT OUTER JOIN mitarbeiter AS m ON
      a.MgrPersNr = m.PersNr
```

2. Alle Teile die von einem Lieferanten aus Ulm geliefert werden, der eine Bewertung besser als 0 erhalten hat. Geben Sie jeweils den Namen und die Bewertung mit aus.

```
SELECT liefert.TeileID,lieferanten.LiefName,lieferanten.
      Bewertung
FROM liefert LEFT OUTER JOIN lieferanten ON liefert.
      LiefNr=lieferanten.LiefNr
WHERE lieferanten.LiefStadt = 'Ulm' AND lieferanten.
      Bewertung > 0
```

3. Alle Auftragspositionen mit ihrer Auftrags- und Teilenummer, Anzahl und Farbe, die vom Kunden Dehling in einem Auftrag geordert wurden.

```
SELECT afpos.AuftrNr,afpos.TeileID,afpos.Anzahl,afpos.
      Farbe
FROM auftragspos AS afpos
LEFT OUTER JOIN auftraege AS af
ON af.AuftrNr = afpos.AuftrNr
LEFT OUTER JOIN kunden AS k
ON af.KdNr=k.KdNr
WHERE k.KdName = 'Dehling'
```

4. Zu allen Lieferanten die Anzahl der Bestellungen.

```
SELECT l.LiefNr,l.LiefName,l.LiefStadt,COUNT(*) AS Anzahl
FROM bestellungen As b
JOIN lieferanten AS l ON b.LiefNr=l.LiefNr
GROUP BY l.LiefNr
```

5. Alle Kunden die mehr als eine Bestellung aufgegeben haben.

```
SELECT k.KdNr,k.KdName,Count(*) AS AnzahlBestellungen
FROM auftraege AS a
LEFT OUTER JOIN kunden AS k ON a.KdNr=k.KdNr
GROUP BY k.KdNr
HAVING Count(*) >1
```

6. Das durchschnittliche Gehalt aller Mitarbeiter aus der Abteilung Lager.

```

SELECT AVG(Gehalt) AS AverageGehalt
FROM mitarbeiter
WHERE mitarbeiter.AbtNr = 'LG'

```

7. Den Mitarbeiter mit dem höchsten Gehalt mit allen seinen Attributen. Mitarbeiter die Manager einer Abteilung sind, sind hierbei ausgeschlossen.

```

SELECT *
FROM mitarbeiter
WHERE gehalt=
( SELECT MAX(mitarbeiter.Gehalt)
  FROM mitarbeiter
  WHERE mitarbeiter.PersNr NOT IN
    ( SELECT abteilungen.MgrPersNr
      FROM abteilungen
    )
)

```

8. Die Kosten eines jeden Auftrags mit Betrag und Auftragsnummer, aufsteigend nach Preis geordnet.

```

SELECT auftragspos.AuftrNr,SUM(DISTINCT preisliste.Preis)
  AS Auftragskosten
FROM auftragspos LEFT OUTER JOIN preisliste ON
  auftragspos.TeileID=preisliste.TeileID
GROUP BY auftragspos.AuftrNr
ORDER BY Auftragskosten ASC

```

9. Zu allen Aufträgen die Attribute und die Anzahl der Auftragspositionen.

```

SELECT A.AuftrNr,A.KdNr,A.AuftrDatum,COUNT(*) AS
  AnzahlPos
FROM auftragspos RIGHT OUTER JOIN auftraege AS A ON
  auftragspos.AuftrNr=A.AuftrNr
GROUP BY auftragspos.AuftrNr

```

10. Alle Kunden mit einem zweiten Kunden in derselben Stadt. Geben Sie diese Kunden paarweise mit Namen und Stadt aus, wobei jedes Paar nur einmal ausgegeben werden soll. Für Städte mit mehr als zwei Kunden werden entsprechend alle möglichen Paare ausgegeben. Ordnen Sie die Ergebnisse nach Städten.

```

SELECT k1.KdName,k2.KdName
FROM kunden AS k1 INNER JOIN kunden AS k2 ON k1.KdStadt=
  k2.KdStadt AND k1.KdName > k2.KdName

```

11. Alle Teile die billiger geliefert werden können als in den Kalkulationskosten veranschlagt. Geben Sie Lieferantenummer, Teilenummer, Farbe, kalkulierter und Lieferantenpreis sowie die Ersparnis aus.

```
SELECT l.LiefNr,t.TeileID,t.Farbe,t.KalkKosten AS
      Kalkuliert, l.Preis AS LiefPreis, t.KalkKosten-l.
      Preis AS Erparniss
FROM teile AS t INNER JOIN liefert AS l ON l.TeileID=t.
      TeileID AND l.Preis<t.KalkKosten
GROUP BY t.TeileID,t.Farbe
```

12. Geben Sie alle Lieferanten namentlich aus, die nicht gleichzeitig Kunden bei uns sind, einmal unter Verwendung eines Left Outer Join und einmal durch eine korrelierte Sub-Query.

```
SELECT lieferanten.LiefName
FROM lieferanten LEFT OUTER JOIN kunden ON lieferanten.
      LiefName =kunden.KdName
WHERE kunden.KdName IS NULL
```

```
SELECT lieferanten.LiefName
FROM lieferanten
WHERE lieferanten.LiefName NOT IN (SELECT kunden.KdName
      FROM Kunden)
```

13. Bestimmen Sie alle Teile mit Nummer und Farbe die zur Zeit in keiner Auftragsposition auftauchen. Stellen sie die Anfrage auf zwei verschiedene Arten: Einmal als korrelierte und einmal als unkorrelierte Subquery.

```
SELECT t.TeileID,t.Farbe
FROM teile AS t
WHERE NOT EXISTS(SELECT a.TeileID
      FROM auftragspos AS a
      WHERE a.TeileID=t.TeileID)
```

```
SELECT t.TeileID,t.Farbe
FROM teile AS t
WHERE t.TeileID NOT IN (SELECT a.TeileID
      FROM auftragspos AS a)
```