

```

-- 7.1.a) Anzahl der Kunden pro Land; Ergebnisspalten: Land,
Anzahl_Kunden
SELECT Land, COUNT(KundenNr) AS Anzahl_Kunden
FROM Kunden
GROUP BY Land;

-- b) Anzahl der Kunden pro Land für die Länder, die mindestens 5 Kunden
haben, absteigend sortiert nach der Anzahl; Ergebnisspalten: Land,
Anzahl_Kunden
SELECT Land, COUNT(KundenNr) AS Anzahl_Kunden
FROM Kunden
GROUP BY Land
HAVING Anzahl_Kunden >= 5
ORDER BY Anzahl_Kunden DESC;

-- c) Anzahl der verschiedenen (Kunden-) Orte pro Land, sortiert nach
Land; Ergebnisspalten: Land, Anzahl_Orte
-- Achtung: Es müssen andere Zahlen rauskommen, als bei Anfrage nach der
Zahl der Kunden pro Land!
SELECT Land, COUNT(DISTINCT Ort) AS Anzahl_Orte
FROM Kunden
GROUP BY Land
ORDER BY Land;

-- d) Gesamtbestellmenge und -umsatz pro Artikel für alle Artikel im
Maßstab 1:18, aufsteigend sortiert nach der Menge; Ergebnisspalten:
ArtikelNr, Artikelname, Menge, Umsatz
-- Hinweis: Auch Artikel ohne Auftragspositionen sollen ausgewiesen
werden, d.h. hier ist Menge = 0 und Umsatz = 0. Nullwerte einer
numerischen Spalte können mit folgendem Ausdruck in
-- numerische Werte umgewandelt werden: "COALESCE(<column>, 0)".
SELECT Artikel.ArtikelNr, Artikel.Artikelname,
       COALESCE(SUM(Auftragspositionen.Bestellmenge), 0) AS Menge,
       COALESCE(SUM(Auftragspositionen.Bestellmenge *
Auftragspositionen.Verkaufspreis), 0) AS Umsatz
FROM Artikel
LEFT JOIN Auftragspositionen ON Artikel.ArtikelNr =
Auftragspositionen.ArtikelNr
WHERE Artikel.Massstab = '1:18'
GROUP BY Artikel.ArtikelNr, Artikel.Artikelname
ORDER BY Menge ASC;

-- e) Wie viele verschiedene Artikel (nicht die Bestellmenge, sondern die
verschiedenen Artikel) wurden jeweils von Kunden aus Deutschland,
Frankreich und England bestellt?
-- Ergebnisspalten: Land, AnzahlArtikel
SELECT Kunden.Land, COUNT(DISTINCT Auftragspositionen.ArtikelNr) AS
AnzahlArtikel
FROM Kunden
LEFT JOIN Auftraege ON Kunden.KundenNr = Auftraege.KundenNr
JOIN Auftragspositionen ON Auftraege.AuftragsNr =
Auftragspositionen.AuftragsNr
WHERE Kunden.Land IN ('Deutschland', 'Frankreich', 'England')
GROUP BY Kunden.Land;

-- f) Wie hoch ist der Gesamtumsatz und -gewinn pro Kunde für alle Kunden
mit einem Gesamtumsatz von mehr als 100.000€?
-- Ergebnisspalten: KundenNr, Firma, Umsatz, Gewinn, AnzahlAuftraege;
Ausgabe aufsteigend sortiert nach Umsatz;

```

-- Hinweis: Für den Umsatz benötigen Sie den Verkaufspreis und die Bestellmenge, für den Gewinn noch zusätzlich den Einkaufspreis des Artikels.

```
SELECT
    Kunden.KundenNr,
    Kunden.Firma,
    SUM(Auftragspositionen.Bestellmenge *
Auftragspositionen.Verkaufspreis) AS Umsatz,
    SUM((Auftragspositionen.Verkaufspreis - Artikel.Einkaufspreis) *
Auftragspositionen.Bestellmenge) AS Gewinn,
    COUNT(Auftraege.AuftragsNr) AS AnzahlAuftraege
FROM Kunden
JOIN Auftraege ON Kunden.KundenNr = Auftraege.KundenNr
JOIN Auftragspositionen ON Auftraege.AuftragsNr =
Auftragspositionen.AuftragsNr
JOIN Artikel ON Auftragspositionen.ArtikelNr = Artikel.ArtikelNr
GROUP BY Kunden.KundenNr, Kunden.Firma
HAVING SUM(Auftragspositionen.Bestellmenge *
Auftragspositionen.Verkaufspreis) > 100000
ORDER BY Umsatz ASC;
```

-- 2.a) Legen Sie eine View "ArtikelUmsatz" an, die Bestellmenge und Gesamtumsatz pro Artikel ermittelt, d.h. sie soll folgende Spalten enthalten: ArtikelNr, Bestellmenge, Umsatz

```
CREATE VIEW ArtikelUmsatz AS
SELECT
    Artikel.ArtikelNr,
    COALESCE(SUM(Auftragspositionen.Bestellmenge), 0) AS
Bestellmengegesamt,
    COALESCE(SUM(Auftragspositionen.Bestellmenge *
Auftragspositionen.Verkaufspreis), 0) AS Umsatz
FROM Artikel
LEFT JOIN Auftragspositionen ON Artikel.ArtikelNr =
Auftragspositionen.ArtikelNr
GROUP BY Artikel.ArtikelNr;
```

-- b) Formulieren Sie die Anfrage aus Teilaufgabe 1 d) unter Verwendung der View aus 2a) erneut! Ist noch eine Aggregation erforderlich?

-- Ja ist hier erforderlich, um die Gesamtbestellmenge und den Gesamtumsatz pro Artikel zu berechnen

```
CREATE VIEW MengeUmsatz1_18 AS
SELECT
    Artikel.ArtikelNr,
    Artikel.Artikelname,
    COALESCE(SUM(Auftragspositionen.Bestellmenge), 0) AS Menge,
    COALESCE(SUM(Auftragspositionen.Bestellmenge *
Auftragspositionen.Verkaufspreis), 0) AS Umsatz
FROM Artikel
JOIN Auftragspositionen ON Artikel.ArtikelNr =
Auftragspositionen.ArtikelNr
WHERE Artikel.Massstab = '1:18'
GROUP BY Artikel.ArtikelNr, Artikel.Artikelname
ORDER BY Menge ASC;
```

-- 3. Es gibt im Datenbestand viele Aufträge, die keine Auftragspositionen haben. Das darf an sich nicht vorkommen.
-- a) Warum lässt sich das nicht durch eine referenzielle Integritätsbedingung sicherstellen?

-- Referenzielle Integrität stellt sicher, dass Fremdschlüssel in einer Tabelle auf gültige Primärschlüssel in einer anderen Tabelle verweisen. Das bedeutet:

-- Jede Auftragsposition muss zu einem existierenden Auftrag gehören. Es verhindert jedoch nicht, dass ein Auftrag ohne Auftragspositionen existiert.

-- Die referenzielle Integrität kontrolliert nur, dass die Beziehung in eine Richtung gültig ist (Auftragspositionen zu Aufträgen), aber nicht in die andere Richtung (Aufträge müssen Auftragspositionen haben).

-- Um sicherzustellen, dass jeder Auftrag mindestens eine Auftragsposition hat, reichen referenzielle Integritätsregeln nicht aus.

-- b) Lassen Sie sich diese Aufträge mit Hilfe einer SELECT-Anweisung mit Unterabfrage in der WHERE Klausel ausgeben.

```
SELECT *  
FROM Auftraege  
WHERE AuftragsNr NOT IN (SELECT DISTINCT AuftragsNr FROM  
Auftragspositionen);
```

-- c) Verfassen Sie die gleiche Abfrage nun mit einem Join!

```
SELECT Auftraege.*  
FROM Auftraege  
LEFT JOIN Auftragspositionen ON Auftraege.AuftragsNr =  
Auftragspositionen.AuftragsNr  
WHERE Auftragspositionen.AuftragsNr IS NULL;
```