

## Aufgabe 2: SQL

Gegeben sind folgende Relationen aus einem Kundenverwaltungssystem:

- 1 Kunde (ID), Vorname, Nachname, PLZ)
- 2 Produkt (GTIN, Bezeichnung, Bruttopreis, MWStSatz)
- 3 Kauf (ID[Kunde], GTIN[Produkt], Datum, Menge)

Verwenden Sie im Folgenden nur Standard-SQL und keine produktspezifischen Erweiterungen. Sie dürfen bei Bedarf Views anlegen. Geben Sie einen Datensatz, also eine Entity, nicht mehrfach aus.

- (a) Schreiben Sie eine SQL-Anweisung, die die Tabelle „Kauf“ anlegt. Gehen Sie davon aus, dass die Tabellen „Kunde“ und „Produkt“ bereits existieren.

**Musterlösung:**

```
1 CREATE TABLE Kauf (  
2     ID INTEGER REFERENCES Kunde(ID),  
3     GTIN INTEGER REFERENCES Produkt(GTIN),  
4     Datum DATE,  
5     Menge INTEGER,  
6     PRIMARY KEY (ID, GTIN, Datum)  
7 );
```

- (b) Schreiben Sie eine SQL-Anweisung, die *Vorname* und *Nachname* aller *Kunden* mit der *Postleitzahl* 20251 ausgibt, absteigend sortiert nach *Nachname* und bei gleichen *Nachnamen*, absteigend nach *Vorname*.

**Musterlösung:**

```
1 SELECT Vorname, Nachname  
2 FROM Kunde  
3 WHERE PLZ = 20251  
4 ORDER BY Nachname DESC, Vorname DESC;
```

- (c) Schreiben Sie eine SQL-Anweisung, die zu jedem Einkauf mit mehr als 10 unterschiedlichen Produkten den *Nachnamen* des *Kunden* und den *Bruttogesamtpreis* des Einkaufs ausgibt. Ein Einkauf ist definiert als Menge aller Produkte, die ein bestimmter Kunde an einem bestimmten Datum kauft.

**Musterlösung:**

```
1 SELECT Nachname, SUM(Bruttopreis * Menge)  
2 FROM Kunde k, Produkt p, Kauf x  
3 WHERE k.ID = x.ID AND p.GTIN = x.GTIN  
4 GROUP BY Datum, Nachname, k.ID  
5 HAVING COUNT (*) > 10;
```

- (d) Schreiben Sie eine SQL-Anweisung, die die *GTINs* aller Produkte ausgibt, die an mindestens einen in der Datenbank enthaltenen PLZ-Bereich noch nie verkauft worden sind. Als in der Datenbank enthaltener PLZ-Bereich gelten alle in der Tabelle „Kunde“ enthaltenen PLZs. Ein Produkt gilt als an einen PLZ-Bereich verkauft, sobald es von mindestens einem Kunden aus diesem PLZ-Bereich gekauft wurde. Produkte, die bisher noch gar nicht verkauft worden sind, müssen nicht berücksichtigt werden.

```

1  SELECT PLZ
2  FROM Kunde k
3  GROUP BY PLZ
4  HAVING NOT EXISTS (
5      SELECT DISTINCT Kauf.GTIN
6      FROM Kauf, Kunde
7      WHERE Kauf.ID = Kunde.ID AND Kunde.PLZ = k.PLZ
8  );

```

#### Musterlösung:

in Mysql gibt es kein EXCEPT

```

1  SELECT GTIN FROM
2  (SELECT GTIN, PLZ
3   FROM Kunde, Produkt)
4  EXCEPT
5  (SELECT DISTINCT x.GTIN, k.PLZ
6   FROM Kunde k, Kauf x
7   WHERE k.ID = x.ID)

```

#### Musterlösung:

```

1  WITH tmp AS (
2      SELECT x.GTIN, k.PLZ
3      FROM Kunde k, Kauf x
4      WHERE x.ID = k.ID
5      GROUP BY x.GTIN, k.PLZ
6  )
7
8  SELECT GTIN
9  FROM tmp
10 WHERE EXISTS
11 (SELECT Kunde.PLZ
12  FROM Kunde LEFT OUTER JOIN tmp
13   ON Kunde.PLZ = tmp.PLZ
14   WHERE tmp.PLZ IS NULL);

```

oder eleganter

```

1  SELECT DISTINCT GTIN
2  FROM
3  (SELECT GTIN, PLZ
4   FROM Kunde, Produkt)
5  EXCEPT
6  (SELECT x.GTIN, k.PLZ
7   FROM Kunde k, Kauf x
8   WHERE x.ID = k.ID
9   GROUP BY x.GTIN, k.PLZ)

```

Statt WITH koennen auch VIEWS erstellt werden ! Eine Konstruktion mit NOT IN sollte auch moeglich sein

- (e) Schreiben Sie eine SQL-Anweisung, die die Top-Ten der am meisten verkauften Produkte ausgibt. Ausgegeben werden sollen der Rang (1 bis 10) und die Bezeichnung des Produkts. Gehen Sie davon aus, dass es keine zwei Produkte mit gleicher Verkaufszahl gibt und verwenden Sie keine produkt-spezifischen Anweisungen wie beispielsweise ROWNUM, TOP oder LIMIT.

#### Musterlösung:

```

1  WITH Gesamtverkauf AS
2  (SELECT k.GTIN, Bezeichnung, SUM (Menge) AS Gesamtmenge)
3  FROM Produkt p, Kauf k
4  WHERE p.GTIN = k.GTIN
5  GROUP BY k.GTIN, Bezeichnung)
6
7  SELECT g1.Bezeichnung, COUNT (*) AS Rang
8  FROM Gesamtverkauf g1, Gesamtverkauf g2
9  WHERE g1.Gesamtmenge <= g2.Gesamtmenge
10 GROUP BY g1.GTIN, g1.Bezeichnung
11 HAVING COUNT (*) <=10
12 ORDER BY Rang;

```

- (f) Schreiben Sie eine SQL-Anweisung, die alle Produkte löscht, die noch nie gekauft wurden.

**Musterlösung:**

```

1  DELETE FROM Produkt
2  WHERE GTIN NOT IN
3  (
4    SELECT DISTINCT GTIN
5    FROM Kauf
6  );

```

```

1  -- sudo mysql < Kundenverwaltungssystem.sql
2  DROP DATABASE IF EXISTS Kundenverwaltungssystem;
3  CREATE DATABASE Kundenverwaltungssystem;
4  USE Kundenverwaltungssystem;
5
6  CREATE TABLE Kunde(
7      ID INTEGER PRIMARY KEY,
8      Vorname VARCHAR(30),
9      Nachname VARCHAR(30),
10     PLZ INTEGER
11 );
12
13 CREATE TABLE Produkt(
14     GTIN INTEGER PRIMARY KEY,
15     Bezeichnung VARCHAR(40),
16     Bruttopreis DOUBLE,
17     MWStSatz INTEGER
18 );
19
20 CREATE TABLE Kauf(
21     ID INTEGER REFERENCES Kunde(ID),
22     GTIN INTEGER REFERENCES Produkt(GTIN),
23     Datum DATE,
24     Menge INTEGER,
25     PRIMARY KEY (ID, GTIN, Datum)
26 );
27
28 INSERT INTO Kunde VALUES (1, 'Max', 'Mustermann',91052);
29 INSERT INTO Kunde VALUES (2, 'Erika', 'Musterfrau',91052);
30 INSERT INTO Kunde VALUES (3, 'Max', 'Meyer',91058);
31 INSERT INTO Kunde VALUES (4, 'Hans', 'Schmidt',91054);
32 INSERT INTO Kunde VALUES (5, 'Eva', 'Müller',91056);
33 INSERT INTO Kunde VALUES (6, 'Hanna', 'Winter',20251);
34 INSERT INTO Kunde VALUES (7, 'Bert', 'Sommer',20251);
35 INSERT INTO Kunde VALUES (8, 'Jakob', 'Sommer',20251);
36

```

```

37 INSERT INTO Produkt VALUES (123, 'Buch', 12.30,19);
38 INSERT INTO Produkt VALUES (124, 'Kaffee', 4.30,7);
39 INSERT INTO Produkt VALUES (125, 'Pullover', 36.40,19);
40 INSERT INTO Produkt VALUES (113, 'Heft', 2.30,19);
41 INSERT INTO Produkt VALUES (023, 'Honig', 3.20,7);
42 INSERT INTO Produkt VALUES (155, 'T-Shirt', 19.30,19);
43 INSERT INTO Produkt VALUES (189, 'Nudeln', 1.30,7);
44 INSERT INTO Produkt VALUES (004, 'Sonnenbrille', 40.60,19);
45 INSERT INTO Produkt VALUES (324, 'Hammer', 22.80,19);
46 INSERT INTO Produkt VALUES (112, 'Topf', 50.20,19);
47 INSERT INTO Produkt VALUES (453, 'Klopapier', 3.30,7);
48 INSERT INTO Produkt VALUES (765, 'Duschgel', 1.89,19);
49 INSERT INTO Produkt VALUES (889, 'Deko', 5.89,19);
50
51 INSERT INTO Kauf VALUES(1,123, '2019-04-11',1);
52 INSERT INTO Kauf VALUES(1,124, '2019-04-11',21);
53 INSERT INTO Kauf VALUES(1,125, '2019-04-11',1);
54 INSERT INTO Kauf VALUES(1,765, '2019-04-11',4);
55 INSERT INTO Kauf VALUES(1,453, '2019-04-11',1);
56 INSERT INTO Kauf VALUES(1,324, '2019-04-11',3);
57 INSERT INTO Kauf VALUES(1,113, '2019-04-11',2);
58 INSERT INTO Kauf VALUES(1,023, '2019-04-11',1);
59 INSERT INTO Kauf VALUES(1,189, '2019-04-11',1);
60 INSERT INTO Kauf VALUES(1,112, '2019-04-11',7);
61 INSERT INTO Kauf VALUES(1,155, '2019-04-11',7);
62 INSERT INTO Kauf VALUES(1,004, '2019-05-11',6);
63 INSERT INTO Kauf VALUES(7,112, '2019-04-11',7);
64 INSERT INTO Kauf VALUES(5,112, '2019-04-11',7);
65 INSERT INTO Kauf VALUES(8,112, '2019-06-23',5);
66 INSERT INTO Kauf VALUES(8,112, '2019-04-12',3);
67 INSERT INTO Kauf VALUES(2,112, '2019-04-23',1);
68 INSERT INTO Kauf VALUES(2,112, '2019-08-11',8);
69 INSERT INTO Kauf VALUES(4,112, '2019-10-10',2);
70 INSERT INTO Kauf VALUES(2,453, '2019-09-24',4);
71 INSERT INTO Kauf VALUES(4,004, '2019-07-30',9);

```