

Aufgabe 4

Gegeben sind folgende Relationen aus einem Kundenverwaltungssystem:

Kunde : {[ID, Vorname, Nachname, PLZ]}

Produkt : {[GTIN, Bezeichnung, Bruttopreis, MWStSatz)]}

Kauf : {[ID[Kunde], GTIN[Produkt], Datum, Menge]}

```
1 CREATE TABLE Kunde(  
2     ID INTEGER PRIMARY KEY,  
3     Vorname VARCHAR(30),  
4     Nachname VARCHAR(30),  
5     PLZ INTEGER  
6 );  
7  
8 CREATE TABLE Produkt(  
9     GTIN INTEGER PRIMARY KEY,  
10    Bezeichnung VARCHAR(40),  
11    Bruttopreis NUMERIC(7,2),  
12    MWStSatz INTEGER  
13 );  
14  
15 CREATE TABLE Kauf(  
16     ID INTEGER REFERENCES Kunde(ID),  
17     GTIN INTEGER REFERENCES Produkt(GTIN),  
18     Datum DATE,  
19     Menge INTEGER,  
20     PRIMARY KEY (ID, GTIN, Datum)  
21 );  
22  
23 INSERT INTO Kunde VALUES  
24     (1, 'Max', 'Mustermann', 91052),  
25     (2, 'Erika', 'Musterfrau', 91052),  
26     (3, 'Max', 'Meyer', 91058),  
27     (4, 'Hans', 'Schmidt', 91054),  
28     (5, 'Eva', 'Müller', 91056),  
29     (6, 'Hanna', 'Winter', 20251),  
30     (7, 'Bert', 'Sommer', 20251),  
31     (8, 'Jakob', 'Sommer', 20251);  
32  
33 INSERT INTO Produkt VALUES  
34     (123, 'Buch', 12.30, 19),  
35     (124, 'Kaffee', 4.30, 7),  
36     (125, 'Pullover', 36.40, 19),  
37     (113, 'Heft', 2.30, 19),  
38     (023, 'Honig', 3.20, 7),  
39     (155, 'T-Shirt', 19.30, 19),  
40     (189, 'Nudeln', 1.30, 7),  
41     (004, 'Sonnenbrille', 40.60, 19),  
42     (324, 'Hammer', 22.80, 19),  
43     (112, 'Topf', 50.20, 19),  
44     (453, 'Klopapier', 3.30, 7),  
45     (765, 'Duschgel', 1.89, 19),  
46     (889, 'Deko', 5.89, 19);  
47  
48 INSERT INTO Kauf  
49     (ID, GTIN, Datum, Menge)  
50 VALUES  
51     (1, 123, '2019-04-11', 1),
```

```

52 (1, 124, '2019-04-11', 21),
53 (1, 125, '2019-04-11', 1),
54 (1, 765, '2019-04-11', 4),
55 (1, 453, '2019-04-11', 1),
56 (1, 324, '2019-04-11', 3),
57 (1, 113, '2019-04-11', 2),
58 (1, 023, '2019-04-11', 1),
59 (1, 189, '2019-04-11', 1),
60 (1, 112, '2019-04-11', 7),
61 (1, 155, '2019-04-11', 7),
62 (1, 004, '2019-05-11', 6),
63 (7, 112, '2019-04-11', 7),
64 (5, 112, '2019-04-11', 7),
65 (8, 112, '2019-06-23', 5),
66 (8, 112, '2019-04-12', 3),
67 (2, 112, '2019-04-23', 1),
68 (2, 112, '2019-08-11', 8),
69 (4, 112, '2019-10-10', 2),
70 (2, 453, '2019-09-24', 4),
71 (4, 004, '2019-07-30', 9);

```

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.

```

1 CREATE TABLE IF NOT EXISTS 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*.

```

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

```

vorname	nachname
Hanna	Winter
Jakob	Sommer
Bert	Sommer

(3 rows)

- (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.

```

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;

```

nachname	sum
-----+-----	
Mustermann	713.86

(1 row)

- (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.

Die beiden Lösungswege liefern leider unterschiedliche Ergebnisse.

```

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 DISTINCT 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
15 )
16 ORDER BY GTIN;

```

gtin

4
23
112
113
123
124
125
155
189
324
453
765

(12 rows)

```

oder
1  SELECT DISTINCT GTIN FROM (
2      (
3          SELECT GTIN, PLZ
4          FROM Kunde, Produkt
5      )
6      EXCEPT
7      (
8          SELECT x.GTIN, k.PLZ
9          FROM Kunde k, Kauf x
10         WHERE x.ID = k.ID
11         GROUP BY x.GTIN, k.PLZ
12     )
13 ) as tmp
14 ORDER BY GTIN;

gtin
-----
      4
     23
    112
    113
    123
    124
    125
    155
    189
    324
    453
    765
    889
(13 rows)

```

- (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 produktspezifischen Anweisungen wie beispielsweise ROWNUM, TOP oder LIMIT.

```

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
8  SELECT g1.Bezeichnung, COUNT (*) AS Rang
9  FROM Gesamtverkauf g1, Gesamtverkauf g2
10 WHERE g1.Gesamtmenge <= g2.Gesamtmenge
11 GROUP BY g1.GTIN, g1.Bezeichnung
12 HAVING COUNT (*) <= 10
13 ORDER BY Rang;

```

bezeichnung	rang
Topf	1
Kaffee	2
Sonnenbrille	3
T-Shirt	4
Klopapier	5
Duschgel	6
Hammer	7
Heft	8
(8 rows)	

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

```

count
-----
      13
(1 row)

1  SELECT COUNT(*) FROM Produkt;
2
3  DELETE FROM Produkt
4  WHERE GTIN NOT IN
5  (
6    SELECT DISTINCT GTIN
7    FROM Kauf
8  );
9
10 SELECT COUNT(*) FROM Produkt;

count
-----
      12
(1 row)

```