

Übungen SQL - Mengenoperationen

Klaus-Georg Deck

2025-07-24

Einleitung

In diesem Skript findest du typische SQL-Übungen mit Beispielcode und den erwarteten Ergebnistabellen. Ziel ist es, die SQL-Mengenoperationen wie **UNION**, **UNION ALL**, **INTERSECT** und **EXCEPT** (Oracle: auch **MINUS**) zu erlernen und zu vertiefen.

Voraussetzungen

Du verfügst über einen Zugang zu einem Datenbank-Server (Oracle oder Postgres) mit den Tabellen des Bike-Verleih-Szenarios und hast Grundkenntnisse in SQL, etwa indem Du die vorherigen Übungen erfolgreich absolviert hast.

Aufgabe 1: UNION und UNION ALL

a) UNION

Erstelle eine SQL-Anweisung, mit der alle Namen von Kunden (**CUSTOMER**) und Mitarbeitenden (**EMPLO**) ausgegeben werden. Die Ausgabe soll jeden Namen nur einmal enthalten, unabhängig davon, wie häufig er in den einzelnen Tabellen vorkommt. Sortiere die Ausgabe nach Namen.

Lösung:

```
SELECT NAME FROM EMPLO
UNION
SELECT NAME FROM CUSTOMER
ORDER BY NAME;
```

b) UNION ALL

Erstelle eine SQL-Anweisung, die alle Namen enthält, jedoch ein weiteres Attribut **ANZAHL**, das die Häufigkeit des Vorkommens des jeweiligen Namens angibt, unabhängig in welchen der beiden Tabellen er vorkommt.

Erwartet wird das folgende Resultat:

NAME	ANZAHL
Anna	6
Antoine	2
Bea	1
Ben	7
...	...

Lösung:

```
SELECT NAME, COUNT(*) ANZAHL FROM
( SELECT NAME FROM EMPLO
  UNION ALL
  SELECT NAME FROM CUSTOMER
)
GROUP BY NAME
ORDER BY NAME;
```

Aufgabe 2: EXCEPT

In Oracle kann anstelle des **EXCEPT** (Standard-SQL) auch **MINUS** verwendet werden.

a) Einfaches EXCEPT

Welche Namen von Mitarbeitenden gibt es, die bei den Kunden nicht vorkommen?

Lösung:

```
SELECT NAME FROM EMPLO
EXCEPT
SELECT NAME FROM CUSTOMER
```

b) Varianten: GROUP BY, DISTINCT und EXCEPT

Welches Ergebnis liefern jeweils die folgenden SQL-Statements? Begründe Deine Antwort.

```
SELECT COUNTRY, STATE FROM ADDRESS
EXCEPT
SELECT ' ', ' '
ORDER BY COUNTRY, STATE;

SELECT DISTINCT COUNTRY, STATE FROM ADDRESS
ORDER BY COUNTRY, STATE;

SELECT COUNTRY, STATE FROM ADDRESS
GROUP BY COUNTRY, STATE
ORDER BY COUNTRY, STATE;
```

Lösung:

Diese drei Anweisungen liefern das gleiche Resultat, nämlich alle Kombinationen von Land und Kanton (COUNTRY, STATE), die in der Tabelle ADDRESS vorkommen, wobei mehrfach vorkommende Wertkombinationen nur einmal angezeigt werden.

Bei der ersten Anweisung liegt dies daran, dass die Operation **EXCEPT** eindeutige Datensätze liefert. Dass dabei ein Datensatz mit leeren Werten entfernt wird, hat keinen Effekt, ausser den, Eindeutigkeit herzustellen.

c) EXCEPT und JOIN

Welche Mitarbeitenden waren bei keiner Bike-Ausleihe beteiligt (Attribut RESPONSIBLE)? Gebe die ID und Namen der Mitarbeitenden an, aufsteigend sortiert nach ID.

Das folgende Resultat wird erwartet:

ID	NAME
16046	Fabio
16048	Leo
16049	Anna
16050	Felix
16051	Bea
16055	Leon

Lösung:

Im ersten Schritt werden die nicht beteiligten Mitarbeitenden ermittelt, indem man von allen diejenigen ausschliesst, die an Ausleihen beteiligt waren:

```
SELECT ID FROM EMPLO
EXCEPT
SELECT RESPONSIBLE FROM RENTAL
```

Dieser Liste von IDs müssen noch die Namen der Mitarbeitenden ‘angespielt’ werden:

```
SELECT E.ID, E.NAME FROM EMPLO E
JOIN
(SELECT ID FROM EMPLO
EXCEPT
SELECT RESPONSIBLE FROM RENTAL
) EMP_EX ON EMP_EX.ID = E.ID
ORDER BY E.ID
```

Alternativ kann man die nicht beteiligten Mitarbeitenden auch wie folgt erhalten:

```
SELECT E.ID, E.NAME FROM EMPLO E
LEFT JOIN RENTAL R ON E.ID = R.RESPONSIBLE
WHERE R.ID IS NULL
ORDER BY E.ID
```

Aufgabe 3: INTERSECT

a) Einfaches INTERSECT

Welche Namen kommen sowohl bei den Mitarbeitenden als auch bei den Kunden vor? Gib zunächst nur die Namen (ohne Mehrfachvorkommen) sortiert nach Name aus.

Lösung:

```
SELECT NAME FROM CUSTOMER
INTERSECT
SELECT NAME FROM EMPLO
ORDER BY NAME;

--oder mit JOIN anstelle INTERSECT
SELECT DISTINCT E.NAME FROM EMPLO E
JOIN CUSTOMER C ON C.NAME = E.NAME
ORDER BY E.NAME;
```

b) Alternative zu INTERSECT

Jetzt interessieren wir uns auch für die Gesamtanzahl (CT_ALL) der Vorkommen von Namen, die in beiden Tabellen jeweils mindestens einmal vorkommen.

Tipp: Verwende kein **INTERSECT** sondern jeweils eine Anweisung zum Zählen der Vorkommen in den beiden Tabellen und verknüpfe diese mittels **JOIN**.

Erwartet wird das folgende Resultat:

NAME	ANZAHL
Anna	6
Ben	7
Fabio	2
...	...

Lösung:

Zunächst die beiden Anweisungen und dann deren Verknüpfung:

```
SELECT NAME NAMEC, COUNT(*) CTC FROM CUSTOMER GROUP BY NAME;

SELECT NAME NAMEE, COUNT(*) CTE FROM EMPLO GROUP BY NAME;

--nun verknüpft
SELECT NAMEC NAME, CTC + CTE AS CT FROM
(SELECT NAME NAMEC, COUNT(*) CTC FROM CUSTOMER GROUP BY NAME)
JOIN
(SELECT NAME NAMEE, COUNT(*) CTE FROM EMPLO GROUP BY NAME)
ON NAMEC = NAMEE
ORDER BY NAMEC;
```