

Databases I, Winter 2023
Solution Practice Assignment 6

Exercise 6-1 SQL
DML and DDL

Consider the following relational database containing information about multivitamin/multimineral tablets (nutrition supplements). It consists of three relations (tables):

The **Supplement** table contains information about supplements:

Supplement	
<u>Name</u>	Supplier
Centrum	Lederle
Aspirin	Bayer
.	.
.	.
.	.

The **Substance** table contains information about vitamins and minerals:

Substance		
<u>Substance</u>	Daily Value	Unit
Vitamin A	5000	IU
Vitamin C	60	mg
Biotin	300	mcg
Iron	18	mg
.	.	.
.	.	.
.	.	.

The **Content** table contains the composition of each tablet in **Substance**, where *Name* and *Substance* are foreign keys referencing **Supplement** and **Substance** respectively:

Content		
<u>Name</u>	<u>Substance</u>	Quantity
Centrum	Vitamin A	5000
Aspirin	Vitamin A	5000
Centrum	Vitamin C	60
Aspirin	Vitamin C	60
Centrum	Biotin	30
Centrum	Iron	18
Aspirin	Iron	27
.	.	.
.	.	.
.	.	.

Express the following queries in SQL:

- a) List the nutrition supplements that do not contain **Biotin**.

Solution:

```

SELECT C.name
FROM Content C
WHERE NOT EXISTS
(
    SELECT C1.name
    FROM Content C1
    WHERE C1.Substance = 'Biotin' AND C.Name = C1.Name
);

```

Another Solution:

```

(SELECT C1.name
FROM Content C1)
EXCEPT
(SELECT C2.name
FROM Content C2
WHERE C2.Substance = 'Biotin');

```

- b) Display the name and supplier of those nutrition supplements that contain Vitamin C but no other substance.

Solution:

```

SELECT S.Name, S.Supplier
FROM Supplement S, Content C
WHERE S.name = C.name AND NOT EXISTS
(
    (SELECT C.Substance
    FROM Content C
    WHERE C.Name = S.Name)
EXCEPT
SELECT S1.Substance
FROM Substance S1
WHERE S1.Substance = 'Vitamin C'
);

```

**Exercise 6-2 SQL
 Clinic**

Consider the following schema:

```

Doctors(doctor_id, doctor_name, specialization, salary)
Patients(patient_id, patient_full_name)
Appointments(patient_id, doctor_id, date)
             -----

```

Write the following queries in SQL.

- a) Find the doctor names who earn more than all those speacilizing in 'peadiatrics'.

Solution:

```

SELECT D1.doctor_name
FROM Doctors D1
WHERE D1.salary > ALL (SELECT D2.salary

```

```
FROM Doctors D2
WHERE D2.specialization = 'peadiatrics'
```

Another Solution:

```
SELECT doctor_name
FROM Doctors
WHERE salary > (SELECT Max(d1.salary)
                FROM Doctors d1
                WHERE d1.specialization = 'peadiatrics')
```

- b) Find the id of the patient with the greatest number of appointments.

Solution:

```
SELECT patient_id, count(*)
FROM Appointments
GROUP BY patient_id
HAVING count(*) = (SELECT max(X)
                  FROM (SELECT count(*) AS X
                        FROM Appointments A2
                        GROUP BY patient_id))
```

- c) For each specialization, display the doctor name with the biggest number of appointments.

Solution:

```
SELECT Dr.doctor_id,Dr.doctor_name,Dr.specialization
FROM Doctor Dr, Appointments A
WHERE Dr.doctor_id = A.doctor_id
GROUP BY Dr.doctor_id,Dr.doctor_name,Dr.specialization
HAVING Count(*)=(SELECT max(c)
                 FROM(SELECT Count(*)as C
                      FROM Doctor Dr2, Appointments A2
                      WHERE Dr2.doctor_id=A2.doctor_id
                      AND Dr2.specialization=Dr.specialization
                      GROUP BY Dr2.doctor_id) as T);
```

Exercise 6-3 SQL

Consider the following schema (University Database):

```
Professors(PresNr, Name, Title, Office)
Students(IDNr, Name, Semester)
Lectures(LecNr, Name, Hours, Taughtby)
-----
Prerequisites(Predecessor, Successor)
-----
Attended(IDNr,LecNr)
-----
TAs(PersNr, Name, Area of Expertise, Boss)
-----
Examine(IDNr, LecNr, PersNr, Grade)
-----
```

Determine the resulting table of the following SQL query and specify in natural language the statement that corresponds to the query.

```
select s.IdNr, s.Name
from Students s, Examine ex
where s.IdNr = ex.IdNr
      and not exists (select *
                      from Attended a
                      where s.IDNr = a.IDNr
                        and a.LecNr = ex.LecNr);
```

Solution:

List of students who are being examined in a course without attending it.

IDNr	Name
28106	Carnap
25403	Jonas
27550	Schopenhauer

Exercise 6-4 SQL

Write the SQL queries for the following statements.

- a) Names and ID numbers of students who are attending more courses than Carnap.

Solution:

```
SELECT S.Name, S.IdNr, COUNT(*)
FROM Students S, Attended A
WHERE S.IdNr = A.IdNr
GROUP BY S.Name, S.IdNr
HAVING COUNT(*) > (SELECT COUNT(*)
                   FROM Students S2, Attended A2
                   WHERE S2.IdNr = A2.IdNr AND S2.Name = Carnap)
```

- b) Which students attended all courses in which they were examined?

Solution:

```
SELECT E.IdNr
FROM Examine E
EXCEPT
SELECT s.IdNr
FROM Students s, Examine ex
WHERE s.IdNr = ex.IdNr
      AND NOT exists (SELECT *
                      FROM Attended a
                      WHERE s.IDNr = a.IDNr
                        AND a.LecNr = ex.LecNr)
```

University Database:

Professors

PersNr	Name	Title	Office
2125	Socrates	Professor	226
2126	Russel	Professor	232
2127	Kopernikus	Associate Professor	310
2133	Popper	Associate Professor	52
2134	Augustinus	Associate Professor	309
2136	Curie	Professor	36
2137	Kant	Professor	7

Students

IDNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Lectures

LecNr	Name	Hours	Taughtby
5001	Anthropology	4	2137
5041	Ethics	4	2125
5043	Cognitive Science	3	2126
5049	Maieutics	2	2125
4052	Logic	4	2125
5052	Philosophy	3	2126
5216	Bioethics	2	2126
5259	The Vienna Circle	2	2133
5022	Faith and Knowledge	2	2134
4630	Critique of Judgement	4	2137

Prerequisites

Predecessor	Successor
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

Attended

IDNr	LecNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

TAs

PersNr	Name	Area of Expertise	Boss
3002	Platon	Theory of Ideas	2125
3003	Aristoteles	Syllogistic	2125
3004	Wittgenstein	Theory of Language	2126
3005	Rhetikus	Planetry motion	2127
3006	Newton	Kepler's laws	2127
3007	Spinoza	God and Nature	2134

Examine

IDNr	LecNr	PersNr	Grade
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Exercise 6-5

Hotels

```

Continent(ID, name, area)
Countries(ID, name, population, continentID)
-----
Countries.continent_id references Continent.id
Hotels(id, name, address, opened_since, total_rooms, country_id)
-----
Hotels.country_id references Countries.id
Rooms(ID, number, hotels_id)
-----
Rooms.hotels_id references Hotels.id
Guests(ID, name, birth_country_id)
-----
Guests.birth_country_id references Countries.id
Reserve(guest_id, room_id, reservation_date)
-----
Reserve.guest_id references Guest.guest_id
Reserve.room_id references Rooms.room_id

```

- a) Show the details of all the rooms that weren't reserved at all.

Solution:

```

SELECT R.*
FROM Rooms R
WHERE R.id NOT IN (SELECT room_id FROM Reserve)

```

- b) List the chains of hotels (names) that are not in Egypt and at least there is a chain of it in Spain.

Solution:

```

SELECT H.name
FROM Hotels H INNER JOIN Countries C ON C.ID=H.country_id
Where NOT EXISTS(
SELECT *
FROM Hotels H2 INNER JOIN Countries C2 ON H2.country_id=C2.ID
WHERE H2.name = H.name AND c2.name = 'Egypt'
) AND Exists (
SELECT *
FROM Hotels H1 INNER JOIN Countries C1 ON H1.country_id = C1.ID
Where H1.name = H.name AND C1.name = 'Spain'
)

```

- c) List the names of guests who reserved rooms in Hilton only.

Solution:

```

SELECT G.name
FROM Reserve R INNER JOIN Guests G ON G.ID= R.guest_id
                INNER JOIN Rooms R1 ON R1.ID=R.room_id
                INNER JOIN Hotels H ON H.id= R1.hotels_id
Where H.name = 'Hilton' AND
NOT EXISTS( SELECT *
            FROM Reserve RE INNER JOIN Guests GE ON GE.ID= RE.guest_id
            INNER JOIN Rooms RE1 ON RE1.ID=RE.room_id
            INNER JOIN Hotels HE ON HE.id= RE1.hotels_id AND G.ID=GE.ID
            Where HE.name <> 'Hilton'
)

```

- d) Find names and countries of guests who reserved a hotel in a country they were born in.

Solution:

```

SELECT G.name, C.name
FROM Guests G INNER JOIN Countries C ON G.birth_country_id = C.ID
WHERE G.ID IN ( SELECT R.guest_id
                FROM Reserve R INNER JOIN Guests G1 ON G1.ID=R.guest_id
                INNER JOIN Rooms R1 ON R1.ID = R.room_id
                INNER JOIN Hotels H ON H.id = R1.hotels_id
                AND H.country_id=G.birth_country_id
                )

```

- e) List the guest names who made at least two different reservations.

Solution:

```

SELECT G.name
FROM Guests G INNER JOIN Reserve R ON G.ID=R.guest_id
WHERE EXISTS ( SELECT *
                FROM Guests G1 INNER JOIN Reserve R1 ON G1.ID=R1.guest_id
                WHERE G1.ID=G.ID AND R1.room_id<>R.room_id
              )

```

f) Select the name of guests who reserved rooms in all hotel chains.

Solution:

```

SELECT G.name
FROM Guests G
Where NOT EXISTS ( SELECT name FROM Hotels
                   EXCEPT
                   SELECT H.name
                   FROM Hotels H INNER JOIN Rooms R ON R.hotels_id=H.id
                               INNER JOIN Reserve R1 ON R1.room_id=R.ID
                   WHERE R1.guest_id=G.ID
                 )

```

Exercise 6-6

State in English what the following SQL Queries denote.

a)

```

SELECT R1.number,R1.id, H.name, COUNT(*) AS NumberOfReservations
FROM Room R1, Reserve RE1, Hotel H
WHERE RE1.room_id=R1.id AND H.id=R1.hotel_id
GROUP BY R1.number,R1.id, H.name
HAVING COUNT(*)=
    ( SELECT MIN(X)
      FROM (SELECT COUNT(*) AS X, R2.id
            FROM Reserve RE2, Room R2
            WHERE RE2.room_id=R2.id AND R2.hotel_id= H.id
            GROUP BY R2.id)
    )

```

Solution:

For each Hotel it will display the room names, ids and number of reservations. However only the room with lowest number of reservations inside each hotel will be displayed.

b)

```

SELECT R1.number
FROM Room R1, Reserve RE1, Hotel H
WHERE R1.hotel_id=H.id AND R1.number <> 501
    AND H.name='Sheraton' AND R1.id=RE.room_id
    AND RE.guest_id IN
    (
    SELECT RE2.guest_id
    FROM Room R2, Reserve RE2
    WHERE R2.id=RE2.room_id AND R2.number = 501
    )

```

Solution:

Display all the room numbers of the hotel 'Sheraton' reserved by a guest who reserved the room 501. In addition the room 501 will be excluded from selection.