Databases
DBDMG - Politecnico di Torino
SQL (III) – Solutions

Exercise 1. Given the following relations (primary keys are underlined):

```
SPECIALIST_VISIT (VCode, VType)
DOCTOR (DCode, DName)
PATIENT (SSN, PName, BirthDate)
BOOKING (Date, Time, DCode, Room, VCode, SSN)
```

express the following queries in SQL language:

(a) For patients who have booked at least three specialist visits with the same doctor in 2005, show the SSN, name and total number of bookings in November 2005.

```
SELECT P.SSN, PName, COUNT(*)
FROM PATIENT P, BOOKING B
WHERE P.SSN = B.SSN
AND Date >= '2005-11-01' AND Date <= '2005-11-30'
AND P.SSN IN
    (SELECT SSN
     FROM BOOKING
     WHERE Date >= '2005-01-01' AND Date <= '2005-12-31'
     GROUP BY SSN, DCode
     HAVING COUNT(*) >= 3)
GROUP BY P.SSN, PName;
```

(b) Show the SSN and name of the patients born after 1960 who have never booked any cardiologist visits.

```
SELECT SSN, PName
FROM PATIENT
WHERE BirthDate > '1960-12-31'
AND SSN NOT IN
    (SELECT SSN
     FROM BOOKING B, SPECIALIST_VISIT SV
     WHERE B.VCode = SV.VCode
     AND VType = 'Cardiologist');
```

```
SELECT SSN, PName
FROM PATIENT
WHERE BirthDate > '1960-12-31'
AND NOT EXISTS
    (SELECT *
     FROM BOOKING B, SPECIALIST_VISIT SV
     WHERE B.VCode = SV.VCode
     AND VType = 'Cardiologist'
     AND B.SSN = PATIENT.SSN);
```

Exercise 2. Given the following relations (primary keys are underlined):

```
CAR_RENTAL (CRCode, CRName, Address, City, Country)
CAR (NumberPlate, Model, Maker, Category, NumPassengers)
CUSTOMER (SSN, Name, Surname, DrivingLicense, CreditCard)
RENTAL_RESERVATION (RCode, SSN, ReservationDate, CRCode, StartDate, EndDate, Price,
NumberPlate)
```

express the following queries in SQL language:

(a) Show the name and surname of customers who have never made reservations for two Mercedes cars starting on the same day.

```
SELECT Name, Surname
FROM CUSTOMER
WHERE SSN NOT IN
    (SELECT SSN
     FROM RENTAL_RESERVATION RR, CAR C
     WHERE RR.NumberPlate = C.NumberPlate
     AND Maker = 'Mercedes'
     GROUP BY SSN, StartDate
     HAVING COUNT(*) >= 2);
```

```
SELECT Name, Surname
FROM CUSTOMER
WHERE NOT EXISTS
    (SELECT *
     FROM RENTAL_RESERVATION RR, CAR C
     WHERE RR.NumberPlate = C.NumberPlate
     AND Maker = 'Mercedes'
     AND RR.SSN = CUSTOMER.SSN
     GROUP BY SSN, StartDate
     HAVING COUNT(*) >= 2);
```

(b) For car rentals that received at least 30 rental reservation requests during November 2006, show the car rental code and name, and the total number of reservation requests received during the whole year 2006.

```
SELECT CR.CRCode, CRName, COUNT(*)
FROM CAR_RENTAL CR, RENTAL_RESERVATION RR
WHERE CR.CRCode = RR.CRCode
AND ReservationDate >= '2006-01-01' AND ReservationDate <= '2006-12-31'
AND CR.CRCode IN
    (SELECT CRCode
     FROM RENTAL_RESERVATION
     WHERE ReservationDate >= '2006-11-01' AND ReservationDate <= '2006-11-30'
     GROUP BY CRCode
     HAVING COUNT(*) >= 30)
GROUP BY CR.CRCode, CRName;
```

Exercise 3. Given the following relations (primary keys are underlined)

```
DRUG (DCode, Name, ActivePrinciple, Category, Maker)
PHARMACY (PCode, OwnerName, Address, City)
SALE (PCode, DCode, Date, Quantity)
```

express the following queries in SQL language:

(a) Show the name of the owner, address, and city of pharmacies that have never sold paracetamol drugs (`ActivePinciple = 'Paracetamol'`).

```
SELECT OwnerName, Address, City
FROM PHARMACY
WHERE PCode NOT IN
    (SELECT PCode
     FROM SALE S, DRUG D
     WHERE S.DCode = D.DCode
     AND ActivePrinciple = 'Paracetamol');
```

```
SELECT OwnerName, Address, City
FROM PHARMACY
WHERE NOT EXISTS
    (SELECT *
     FROM SALE S, DRUG D
     WHERE S.DCode = D.DCode
     AND ActivePrinciple = 'Paracetamol'
     AND S.PCode = PHARMACY.PCode);
```

(b) For pharmacies that have sold a total quantity of drugs greater than the average quantity of drugs sold by all pharmacies, show the pharmacy code, the name of the owner, and the total quantity of Bayer drugs (`Maker = 'Bayer'`) sold during the whole year 2007.

```
SELECT P.PCode, OwnerName, SUM(Quantity)
FROM PHARMACY P, SALE S, DRUG D
WHERE S.PCode = P.PCode
AND S.DCode = D.DCode
AND Date >= '2007-01-01' AND Date <= '2007-12-31'
AND Maker = 'Bayer'
AND P.PCode IN
    (SELECT PCode
     FROM SALE
     GROUP BY PCode
     HAVING SUM(Quantity) >
         (SELECT AVG(Q)
          FROM
              (SELECT SUM(Quantity) AS Q
               FROM SALE
               GROUP BY PCode) AS T))
GROUP BY P.PCode, OwnerName;
```

```
SELECT P.PCode, OwnerName, SUM(Quantity)
FROM PHARMACY P, SALE S, DRUG D
WHERE S.PCode = P.PCode
AND S.DCode = D.DCode
AND Date >= '2007-01-01' AND Date <= '2007-12-31'
AND Maker = 'Bayer'
AND P.PCode IN
    (SELECT PCode
     FROM SALE
     GROUP BY PCode
     HAVING SUM(Quantity) >
         (SELECT SUM(Q) / COUNT(DISTINCT PCode)
```



```
          FROM SALE))
GROUP BY P.PCode, OwnerName;
;
```

```
SELECT P.PCode, OwnerName, SUM(Quantity)
FROM PHARMACY P, SALE S, DRUG D,
    (SELECT PCode, SUM(Quantity) AS Q
     FROM SALE
     GROUP BY PCode) AS PHARMACY_SALE
WHERE S.PCode = P.PCode
AND S.DCode = D.DCode
AND S.PCode = PHARMACY_SALE.PCode
AND Date >= '2007-01-01' AND Date <= '2007-12-31'
AND Maker = 'Bayer'
AND PHARMACY_SALE.Q >
    (SELECT SUM(Q) / COUNT(DISTINCT PCode)
     FROM SALE)
GROUP BY P.PCode, OwnerName;
;
```

Exercise 4. Given the following relations (primary keys are underlined)

```
BEACH (BeachAddress, BeachCity, Capacity, LifeguardInCharge)
LIFEGUARD (LID, FirstName, LastName, HomeCity, PhoneNumber)
RESCUE (RID, Lifeguard, BeachAddress, BeachCity, BatherSSN, Date, Reason)
BATHER (SSN, FirstName, LastName, DateOfBirth, HomeCity
```

express the following queries in SQL language:

(a) Show the identification code, the first name and the last name of lifeguards living in Ostuni who are not in charge of any beach and have accomplished at least two rescues on the same beach on the same day.

```
SELECT LID, FirstName, LastName
FROM LIFEGUARD
WHERE HomeCity = 'Ostuni'
AND LID NOT IN
    (SELECT LifeguardInCharge
     FROM BEACH)
AND LID IN
    (SELECT Lifeguard
     FROM RESCUE
     GROUP BY Lifeguard, BeachAddress, BeachCity, Date
     HAVING COUNT(*) >= 2);
```

(b) Taking into account only the beaches located in cities that have at least 10 beaches whose capacity is greater than the average capacity of all beaches, show for each of such beaches the address, the city, and the number of distinct lifeguards who have performed rescues on that beach.

```
SELECT B.BeachAddress, B.BeachCity, COUNT(DISTINCT Lifeguard)
FROM BEACH B, RESCUE R
WHERE B.BeachAddress = R.BeachAddress AND B.BeachCity = R.BeachCity
AND BeachCity IN
    (SELECT BeachCity
     FROM BEACH
     WHERE Capacity >
```

```
          (SELECT AVG(Capacity)
           FROM BEACH)
     GROUP BY BeachCity
     HAVING COUNT(*) >= 10)
GROUP BY B.BeachAddress, B.BeachCity;
```

Exercise 5. Given the following relations (primary keys are underlined)

```
ATHLETE (ACode, AName, ASurname, TeamName, Country)
ATTENDANCE (CCode, ACode, Position)
COMPETITION (CCode, CName, CType, Category)
```

express the following queries in SQL language:

(a) Show the code and the name of the athletes who never attended any Super G competitions (`CType = 'Super G'`).

```
SELECT ACode, AName
FROM ATHLETE
WHERE ACode NOT IN
    (SELECT ACode
     FROM ATTENDANCE A, COMPETITION C
     WHERE A.CCode = C.CCode
     AND CType = 'Super G');
```

(b) For each Italian or Spanish athlete who attended at least 10 Super G competitions, show the code of the athlete, the name, the total number of attended competitions, and the best ranking position achieved by the athlete.

```
SELECT ATHLETE.ACode, AName, COUNT(*), MIN(Position)
FROM ATHLETE, ATTENDANCE
WHERE ATTENDANCE.ACode = ATHLETE.ACode
AND ATHLETE.ACode IN
    (SELECT ACode
     FROM ATHLETE ATH, ATTENDANCE ATT, COMPETITION C
     WHERE ATT.ACode = ATH.ACode
     AND ATT.CCode = C.CCode
     AND (Country = 'Italy' OR Country = 'Spain')
     AND CType = 'Super G'
     GROUP BY ACode
     HAVING COUNT(*) > 10)
GROUP BY ATHLETE.ACode, AName;
```

