Introduction to Databases
DBDMG - Politecnico di Torino
SQL (II) - Solutions

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

   COURSE (<u>CourseCode</u>, CourseName, Year, Semester)
   COURSE SCHEDULE (<u>CourseCode</u>, <u>DayOfWeek</u>, <u>StartTime</u>, EndTime, Room)

   express the following queries in SQL language.

   (a) Find the rooms in which none of the first-year courses has ever been given.

   ```
   SELECT DISTINCT Room
   FROM COUSE-SCHEDULE CS
   WHERE Room NOT IN (SELECT Room
   FROM COURSE-SCHEDULE CS1, COURSE C
   WHERE CS1.CourseCode=C.CourseCode AND Year =1)
   ```

   or

   ```
   SELECT DISTINCT Room
   FROM COUSE-SCHEDULE CS
   WHERE NOT EXISTS (SELECT *
   FROM COURSE-SCHEDULE CS1, COURSE C
   WHERE CS1.CourseCode=C.CourseCode AND Year =1
       AND CS.Room=CS1.Room)
   ```

   (b) Find the codes, the names and the total number of weekly hours of the third-year courses whose total number of weekly hours is greater than 10 and whose schedule spans three different days of the week.

   ```
   SELECT C.CourseCode, CourseName, SUM(EndTime-StartTime)
   FROM COURSE C, COURSE-SCHEDULE CS
   WHERE C.CourseCode=CS.CourseCode AND Year = 3
   GROUP BY C.CourseCode, CourseName
   HAVING SUM(EndTime-StartTime)>10 AND
       COUNT(DISTINCT DayOfWeek)=3
   ```

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

   FLAT (<u>FCode</u>, Address, City, Surface)
   LEASING CONTRACT (<u>LCCode</u>, StartDate, EndDate, PersonName, MonthlyPrice, FCode)

   N.B. The **Surface** is expressed in square meters.
   For contracts that have not yet expired the **EndDate** is **NULL**.

   express the following queries in SQL language.

1

(a) For the cities in which at least 100 contracts have been signed, find the city, the maximum monthly price, the average monthly price, the maximum duration of the leasing contracts, the average duration of the leasing contracts and the total number of signed contracts.

```
SELECT F.City, MAX(L.MonthlyPrice), AVG(L.MonthlyPrice),
       MAX(L.EndDate-L.StartDate),
       AVG(L.EndDate-L.StartDate), COUNT(*)
FROM FLAT F, LEASING-CONTRACT L
WHERE F.FCode=L.Fcode [AND EndDate IS NOT NULL]
GROUP BY F.City
HAVING COUNT (*)>=100
```

(b) Find the names of the people who have never rented any flat with a surface greater than 80 square meters.

```
SELECT DISTINCT PersonName
FROM LEASING_CONTRACT LC
WHERE PersonName NOT IN (
    SELECT PersonName
    FROM LEASING_CONTRACT LC1, FLAT F
    WHERE F.Surface>80 AND LC1.FCode=F.Fcode)
```

or

```
SELECT DISTINCT PersonName
FROM LEASING_CONTRACT LC
WHERE NOT EXIST (
    SELECT *
    FROM LEASING_CONTRACT LC1, FLAT F
    WHERE F.Surface>80 AND LC1.FCode=F.Fcode AND
    LC.PersonName=LC1.PersonName)
```

(c) Find the names of the people who have signed more than two leasing contracts for the same flat (in different periods).

```
SELECT DISTINCT PersonName
FROM LEASING_CONTRACT LC
GROUP BY PersonNamem, FCode
HAVING COUNT(*)>2
```

(d) Find the codes and the addresses of flats in Turin whose monthly leasing price has always been greater than 500 Euro and for which more than 5 contracts have been signed.

```
SELECT FCode, Address
FROM FLAT F
WHERE City='Turin' AND FCode NOT IN (SELECT FCode
                                     FROM LEASING-CONTRACT
                                     WHERE MonthlyPrice<=500)
AND Fcode IN (SELECT FCode
              FROM LEASING-CONTRACT
              GROUP BY Fcode
              HAVING COUNT(*)>5)
```

or

```
    SELECT F.FCode, Address
    FROM FLAT F, LEASING-CONTRACT LC
    WHERE City=   Turin    AND F.FCode NOT IN (SELECT FCode
                                               FROM LEASING-CONTRACT
                                               WHERE MonthlyPrice<=500)
       AND LC.Fcode=F.Fcode
    GROUP BY F.Fcode, Address
    HAVING COUNT(*)>5
```

or

```
    SELECT F.FCode, Address
    FROM FLAT F, LEASING-CONTRACT LC
    WHERE City=   Turin    AND LC.Fcode=F.Fcode
    GROUP BY F.Fcode, Address
    HAVING COUNT(*)>5 AND MIN(MonthlyPrice)>500
```

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

   PERSON (<u>Name</u>, Sex, Age)
   PARENT(<u>ParentName</u>, ChildName)

   express the following queries in SQL language.

   (a) Find the name of each person younger than 10 years old who is an only child.

```
    SELECT DISTINCT Name
    FROM PERSON P, PARENT PA
    WHERE PA.ChildName=P.Name AND Age<10
          AND ParentName IN (SELECT ParentName
                             FROM PARENT
                             GROUP BY ParentName
                             HAVING COUNT(*)=1)
```

   or

```
    SELECT DISTINCT Name
    FROM PERSON P, PARENT PA
    WHERE PA.ChildName=P.Name AND Age<10
          AND ParentName NOT IN (SELECT ParentName
                                 FROM PARENT
                                 GROUP BY ParentName
                                 HAVING COUNT(*)>1)
```

   or

```
    SELECT DISTINCT Name
    FROM PERSON P, PARENT PA
    WHERE PA.ChildName=P.Name AND Age<10
          AND NOT EXISTS (SELECT *
                          FROM PARENT PA1
                          WHERE PA1.ParentName=PA.ParentName
                            AND PA1.ChildName<> PA.ChildName)
```