

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

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

```
COURSE (CourseCode, CourseName, Year, Semester)
COURSE_SCHEDULE (CourseCode, DayOfWeek, StartTime, 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 COURSE_SCHEDULE
WHERE Room NOT IN
  (SELECT Room
   FROM COURSE_SCHEDULE CS, COURSE C
   WHERE CS.CourseCode = C.CourseCode
   AND C.Year = 1);
```

```
SELECT DISTINCT Room
FROM COURSE_SCHEDULE
WHERE NOT EXISTS
  (SELECT *
   FROM COURSE_SCHEDULE CS, COURSE C
   WHERE CS.CourseCode = C.CourseCode
   AND C.Year = 1
   AND CS.Room = COURSE_SCHEDULE.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 CourseCode, CourseName, SUM(EndTime - StartTime)
FROM COURSE_SCHEDULE CS, COURSE C
WHERE CS.CourseCode = C.CourseCode
AND C.Year = 3
GROUP BY CourseCode, CourseName
HAVING SUM(EndTime - StartTime) > 10
AND COUNT(DISTINCT DayOfWeek) = 3;
```

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

FLAT (FCode, Address, City, Surface)

LEASING_CONTRACT (LCCode, 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:

- (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 City, MAX(MonthlyPrice), AVG(MonthlyPrice), MAX(EndDate - StartDate),
       AVG(EndDate - StartDate), COUNT(*)
FROM FLAT F, LEASING_CONTRACT LC
WHERE F.FCode = LC.FCode
GROUP BY 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
WHERE PersonName NOT IN
  (SELECT PersonName
   FROM LEASING_CONTRACT LC, FLAT F
   WHERE LC.FCode = F.FCode
   AND Surface > 80);
```

```
SELECT DISTINCT PersonName
FROM LEASING_CONTRACT
WHERE NOT EXISTS
  (SELECT *
   FROM LEASING_CONTRACT LC, FLAT F
   WHERE LC.FCode = F.FCode
   AND Surface > 80
   AND LC.PersonName = LEASING_CONTRACT.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
GROUP BY PersonName, 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€ and for which more than 5 contracts have been signed.

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

```
SELECT FCode, Address
FROM FLAT F, LEASING_CONTRACT LC
WHERE F.FCode = LC.FCode
AND City = 'Turin'
GROUP BY F.FCode, Address
HAVING COUNT(*) > 5 AND MIN(MonthlyPrice)>500;
```

```
SELECT FCode, Address
FROM FLAT F, LEASING_CONTRACT LC
WHERE F.FCode = LC.FCode
AND City = 'Turin'
AND NOT EXISTS
  (SELECT *
   FROM LEASING_CONTRACT
   WHERE MonthlyPrice <= 500
   AND LEASING_CONTRACT.FCode = LC.FCode)
GROUP BY F.FCode, Address
HAVING COUNT(*) > 5;
```

```
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);
```

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

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

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

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