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

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

   SEMINAR (SCode, STitle, Topic, Duration)
   SPEAKER (S-SSN, SName, BirthDate)
   SEMINAR-CALENDAR (SCode, Date, StartTime, S-SSN, Room)
   EXPERTISE (S-SSN, Topic)

   express the following queries in SQL language.

   (a) Show the code of the seminars for which at least one scheduled presentation was held by the speaker with the highest number of topics of expertise.

```
SELECT SCode
FROM SEMINAR-CALENDER
WHERE S-SSN IN (SELECT S-SSN
                FROM EXPERTISE
                GROUP BY S-SSN
                HAVING COUNT(*)=(SELECT MAX(n)
                                 FROM (SELECT COUNT(*) INTO n
                                 FROM EXPERTISE
                                 GROUP BY S-SSN)AS COUNTER))
```

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

   TEACHER(TCode, TName, TSurname, Department, ResearchGroupName, ResearchArea)
   COURSE(CCode, CName, EnrollingStudent, TCode, Topic)
   CLASSROOM(RoomID, Floor, VideoKit, Seat)
   LECTURE(RoomID, Date, StartHour, EndHour, CCode, AttendingStudent, VideoKit=[yes, no])

   express the following queries in SQL language.

   (a) For each teacher who has taught exclusively courses whose topic is databases, select the code of the teacher and, among her courses, the code of the course for which the average number of students attending the course lectures is the highest.

```
SELECT TCode,Ccode
FROM TEACHER T,COURSE C,LECTURE L
WHERE C.TCode=T.TCode AND L.CCode=C.CCode
  AND TCode NOT IN (SELECT TCode
                    FROM TEACHER T,COURSE C
                    WHERE C.TCode=T.TCode
                      AND Topic<>"Databases")
```

```
GROUP BY (TCode,CCode)
HAVING AVG(AttendingStudent#)=
                (SELECT MAX(x)
                  FROM (SELECT AVG(AttendingStudent#) INTO x
                        FROM LECTURE
                        GROUP BY CCode))
```

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

   TOUR-GUIDE (GuideCode, Name, Surname, Nationality)
   TYPE-OF-TOUR (TourTypeCode, Monument, Duration, City)
   GROUP (GroupCode, NumberOfParticipants, Language)
   GUIDED-TOUR-CARRIED-OUT (GroupCode, Date, StartTime, TourTypeCode, GuideCode)

   express the following queries in SQL language.

   (a) For each tour guide who has never guided a type of tour for French-speaking groups, show name and surname and, for each date, the total number of type of tours guided and their totale duration.

```
SELECT TG.Name, TG.Surname, GTCO.Date, SUM(Duration)
FROM TOUR-GUIDE TG, TYPE-OF-TOUR TT,
     GUIDED-TOUR-CARRIED-OUT GTCO
WHERE TG.GuideCode = GTCO.GuideCode
  AND TT.TourTypeCode = GTCO.TourTypeCode
  AND TG.GuideCode NOT IN
                (SELECT TG.GuideCode
                 FROM GUIDED-TOUR-CARRIED-OUT GTCO2, GROUP G
                 WHERE G.GroupCode = VGE2.GroupCode
                   AND GR.Language = 'French')
GROUP BY TG.GuideCode, G.Name, G.Surname, GTCO.Date
```

   (b) Among the monuments for which at least 10 guided tours have been carried out, show the monuments visited by the highest number of participants.

```
SELECT Monument
FROM TYPE-OF-TOUR TT, GUIDED-TOUR-CARRIED-OUT GT, GROUP G
WHERE TT.TourTypeCode = GT.TourTypeCode
  AND G.GroupCode = GT.GroupCode
GROUP BY Monument
HAVING COUNT(*) >=10
  AND SUM(NumberOfParticipants) =
            (SELECT MAX(TOTPart)
             FROM (SELECT SUM(NumberOfParticipants) AS TOTPart
                   FROM TYPE-OF-TOUR TT,
                        GUIDED-TOUR-CARRIED-OUT GT,
                        GROUP G
                   WHERE TT.TourTypeCode = GT.TourTypeCode
                     AND G.GroupCode = GT.GroupCode
                   GROUP BY Monument
                   HAVING COUNT(*) >= 10) AS TOTM)
```

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

   TEENAGER (SSN, Name, Surname, BirthDate, CityOfResidence, Sex)
   ACTIVITY (ActivityCode, AName, Description, Category)

SUMMER-CAMP (CampCode, CampName, City)
SUBSCRIPTION-TO-ACTIVITY-IN-SUMMER-CAMP (SSN, ActivityCode, CampCode, SubscriptionDate

express the following queries in SQL language.

(a) For each summer camp to which at least 15 teenagers subscribed to carry out activities related to at lest 3 different categories, show the name of the summer camp, the city where the camp takes place, and for each activity carried out in the camp the total number of subscribed teenagers.

```
        SELECT CampName, City, ActivityCode, COUNT(DISTINCT SSN)
        FROM SUMMER-CAMP SC,
             SUBSCRIPTION-TO-ACTIVITY-IN-SUMMER-CAMP STA
        WHERE SC.CampCode = STA.CampCode AND SC.CampCode IN
                                (SELECT CampCode
                                 FROM STA, Acitivity A
                                 WHERE STA.ActivityCode=A.ActivityCode
                                 GROUP BY CampCode
                                 HAVING COUNT(DISTINCT SSN)>=15
                                 AND COUNT(DISTINCT Category) >= 3)
    GROUP BY CE.CampCode, CampName, City, ActivityCode
```

(b) For each teenager, born before 2005, who subscribed to activities organized by at least 5 different summer camps, show name, surname, birth date of the teenager anf the name of each summer camp to which the teenager subscribed to all the different activities organized by the camp.

```
        SELECT Name,Surname,Birthdate,Campname
        FROM TEENAGER T,SUMMER-CAMP S,
             SUBSCRIPTIO-TO-ACTIVITY-IN-SUMMER-CAMP SA
        WHERE T.SSN=SA.SSN
        AND S.CampCode=SA.CampCode AND Birthdate<"01/01/2005)
␣␣␣␣␣␣␣␣AND␣SSN␣IN␣(SELECT␣SSN
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣FROM␣SUBSCRIPTION-TO-ACTIVITY-IN-SUMMER-CAMP␣SA2
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣GROUP␣BY␣SSN
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣HAVING␣COUNT(DISTINCT␣CampCode)>=5)
␣␣␣␣␣␣␣␣GROUP␣BY␣T.SSN,␣S.CampCode,␣Name,␣Surname,␣Birthday,␣Campname
␣␣␣␣␣␣␣␣HAVING␣COUNT(DISTINCT␣ActivityCode)=
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣(SELECT␣COUNT(DISTINCT␣ActivityCode)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣FROM␣SUBSCRIPTION-TO-ACTIVITY-IN-SUMMER-CAMP␣SA3
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣WHERE␣SA3.CampCode=S.CampCode
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣AND␣SA3.SSN=T.SSN␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣GROUP␣BY␣CampCode)
␣␣␣␣␣␣␣␣
```