

SQL Language: Exercise

1. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

ATHLETE (ACode, AName, ASurname, Nationality, BirthDate)
ATTENDANCE (CCode, ACode, Position, Time)
COMPETITION (CCode, Place, Date, CType)

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

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

- b) Find the countries for which at least 5 athletes born before 1980 compete, each of whom has participated in at least 10 cross-country skiing competitions.

```
SELECT Nationality
FROM ATHLETE
WHERE BirthDate < 1/1/1980 AND ACode IN
      (SELECT ACode
       FROM ATTENDANCE A, COMPETITION C
       WHERE A.CCode = GC.CCode AND C.CType='cross-country skiing'
       GROUP BY ACode
       HAVING COUNT(*) >= 10)
GROUP BY Nationality
HAVING COUNT(*) >= 5
```

2. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

PUBLISHER(PCode, PublisherName, Address, City)

BOOK(BCode, Title, AuthorName, PCode)
 BOOKSHOP(BSCode, BookshopName, Address, City)
 SALE(BCode, BSCode, Date, NoCopies)

- a) Find the name of the bookstores where no books from publishers based in Turin have been sold.

```
SELECT BookshopName
FROM BOOKSHOP B
WHERE BSCode NOT IN
  (SELECT BSCode
   FROM SALE S, BOOK B, PUBLISHER P
   WHERE P.PCode = B.PCode AND S.BCode = B.BCode
   AND City = 'Turin')
```

- b) Find the name of the publishers for which at least 10 publications were sold in 2002 in bookstores in Rome in more than 2,000 copies.

```
SELECT PublisherName
FROM BOOK B, PUBLISHER P
WHERE P.PCode = B.PCode AND PCode NOT IN
  (SELECT PCode
   FROM SALE S, BOOKSHOP BS
   WHERE S.BSCode = BS.BSCode
   AND City = 'Rome'
   AND Date >= 1/1/2002 AND Date <= 31/12/2002
   GROUP BY BCode
   HAVING SUM(NoCopies) > 2000)
GROUP BY P.PCode, PublisherName
HAVING COUNT(*) >= 10
```

3. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

MC_TEST(TestID, Topic, Score)
 STUDENT (StudentID, Name, Address, CityA)
 TEST-RESULT (StudentID, TestID, isCorrectAnswer)

- a) Find the names of students who did not correctly answer **any** math multiple choice test.

```

SELECT DISTINCT Name
FROM STUDENT S, TEST-RESULT T, MC_TEST MC
WHERE S.StudentID = T.StudentID AND MC.TestID = T.TestID
AND Topic = "Math"
AND S.StudentID NOT IN (SELECT S1.StudentID
                        FROM STUDENT S1, TEST-RESULT T1, MC_TEST MC1
                        WHERE S1.StudentID = T1.StudentID
                          AND MC1.TestID = T1.TestID
                          AND Topic = "Math"
                          AND isCorrectAnswer = True)

```

Alternative solution:

```

SELECT DISTINCT Name
FROM STUDENT S, TEST-RESULT T, MC_TEST MC
WHERE S.StudentID = T.StudentID AND MC.TestID = T.TestID
AND Topic = "Math"
AND NOT EXISTS (SELECT *
                FROM STUDENT S1, TEST-RESULT T1, MC_TEST MC1
                WHERE S1.StudentID = T1.StudentID
                  AND MC1.TestID = T1.TestID
                  AND T1.StudentID=T.StudentID
                  AND Topic = "Math"
                  AND isCorrectAnswer = True)

```

- b) Find the names of the students in Turin who achieved the maximum possible score in the math multiple choice test.

```

SELECT Name, SUM(Score)
FROM STUDENT S, TEST-RESULT T, MC_TEST MC
WHERE S.StudentID = T.StudentID AND MC.TestID = T.TestID
AND Topic = "Math" AND isCorrectAnswer=True
GROUP BY S.StudentID, Name
HAVING COUNT(*) = (SELECT COUNT(*)
                  FROM MC_TEST MC1 WHERE Topic = "Math")

```

4. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

AIRCRAFT (SerialNumber, Model, Capacity)

SCHEDULE (Code, Departure, Destination, DepartureTime, ArrivalTime)

FLIGHTS (Code, SerialNumber, Date, NoReservations)

- a) Find out the code and departure time of flights departing from Milan to Naples on 1 October 1993, which still have free seats and whose duration (difference between the time of arrival and the time of departure) is less than the average duration of flights from Milan to Naples.

5. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

MECHANIC(ID, Name)

CAN-REPAIR (ID, FaultType)

HAS-DONE-REPAIR (RCode, ID, LicensePlate, Date, Duration, FaultType)

- a) Find the name of the mechanics who have carried out at least one repair of a fault that they did not know how to repair.
- b) For cars that required repairs carried out by at least 3 different mechanics on the same day, display the car's license plate, the date of repairs and the types of faults that occurred, sorting the result in ascending order of license plate and descending order of date.

6. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

MEETING_ROOM(RCode, NumberSeats, Projector)

BOOKING(RCode, Date, StartTime, EndTime, ECode)

EMPLOYEE(ECode, Name, Surname, BirthDate, City)

- a) View the code and maximum number of seats in projector-equipped rooms that have been booked at least 15 times for meetings starting before 3:00 p.m., but have never been booked for meetings starting after 8:00 p.m.

```
SELECT RCode, NumberSeats
FROM MEETING_ROOM M
WHERE Projector=True AND RCode NOT IN
      (SELECT RCode FROM BOOKING WHERE StartTime > 20:00)
AND RCode IN (SELECT RCode FROM BOOKING
              WHERE StartTime < 15:00
              GROUP BY RCode
              HAVING COUNT(*) >=15)
```

- b) View for each room the room code, the maximum number of seats and the number of reservations considering only the last date on which the room was booked.

7. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

GUIDE (GCode, Name, Surname, Nationality)

TOUR_TYPE (TourTypeCode, Monument, Duration, City)

GROUP (GRCCode, NumberParticipants, Language)

GUIDED_TOUR (GRCCode, Date, StartTime, TourTypeCode, GCode)

- a) Among the monuments for which at least 10 guided tours have been made, view the monument that has been visited by the largest number of people overall.

```
SELECT Monument
FROM TOUR_TYPE TT, GUIDED_TOUR GT, GROUP G
WHERE TT.TourTypeCode = GT. TourTypeCode AND G.GRCCode = GT.GRCCode
GROUP BY Monument
HAVING COUNT(*) >= 10
AND SUM(NumberParticipants) =
      (SELECT MAX(Partial)
       FROM (SELECT SUM(NumberParticipants) AS Partial
             FROM TOUR_TYPE TT1, GUIDED_TOUR GT1, GROUP G1
             WHERE TT1.TourTypeCode = GT1. TourTypeCode
                  AND G.GRCCode = GT1.GRCCode
             GROUP BY Monument
             HAVING COUNT(*)>=10) AS MaxTotal)
```

```
WITH ParticipantsMonuments AS
(SELECT SUM(NumberParticipants) AS Partial
 FROM TOUR_TYPE TT1, GUIDED_TOUR GT1, GROUP G1
 WHERE TT1.TourTypeCode = GT1. TourTypeCode
      AND G.GRCCode = GT1.GRCCode
 GROUP BY Monument
 HAVING COUNT(*)>=10) AS MaxTotal),
```

```
MaxParticipantsMonuments AS
(SELECT MAX(Partial) FROM ParticipantsMonuments)
SELECT Monument
FROM TOUR_TYPE TT, GUIDED_TOUR GT, GROUP G
WHERE TT.TourTypeCode = GT. TourTypeCode AND G.GRCCode = GT.GRCCode
GROUP BY Monument
HAVING COUNT(*) >= 10
AND SUM(NumberParticipants)
```

- b) 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 total duration.

```

SELECT Name, Surname, Date, COUNT(DISTINCT TourTypeCode), SUM(Duration)
FROM GUIDE G, GUIDED_TOUR GT, TOUR_TYPE TP
WHERE G.GCode = GT.GCode AND GT.TourTypeCode = TP.TourTypeCode
AND G.GCode NOT IN (SELECT GT.GCode
                     FROM GUIDED_TOUR GT, GROUP G
                     WHERE GT.GRCode = G.GRCode
                     AND Language = "French")
GROUP BY GCode, Name, Surname, Date

```

8. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

TEENAGER(TaxID, Name, Surname, Birthdate, City)

ACTIVITY(ACode, ActivityName, Description, Category)

SUMMER_CAMP(CCode, CampName, City)

REGISTRATION-FOR-ACTIVITIES-IN-SUMMER-CAMP(TaxID, ACode, CCode, RegistrationDate)

- a) View the first and last name of the teenager who participated in the largest number of summer camps for the activity in the "Tennis" category.

9. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

CUSTOMER (CustomerID, Name)

ACCOUNT(AccountID, Balance, Branch, Country)

CUSTOMER_ACCOUNT (CustomerID, AccountID)

- a) Find all branches that have at least one client holding alone (without co-holders) of a single current account (customer to whom no other current account is in the name).

10. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

TAXPAYER(TaxId, Name, Street, City)

TAX_RETURN(ReturnID, Type, Income)

PRESENTS(TaxId, ReturnID, Date)

- a) Display the tax ID, name, and average income reported from 1990 onwards by taxpayers whom maximum income reported since 1990 is higher than the average income calculated on all tax returns in the database.

11. Given the following relational schema (primary keys are underlined, optional attributes are indicated by “*”)

PERSON (Name, Sex, Age)

PARENT (ParentName, ChildName)

Example table

ParentName	ChildName
Jennifer	Susan
Louis	Susan
Susan	Mike
Kyle	Mike
Susan	Christine
Kyle	Christine

- Find the name of each person younger than 10 years old who is an only child

In our example, the query output should be the following:

R
Susan

```

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)

```

[Note: a parent of an only child has one child, whereas a parent of siblings has more than one child. Here, the nested query finds the list of parents that have more than one child (exclusion condition). The following solution checks the opposite inclusion condition (only parents with exactly one 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)

```

```

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)

```

[Note: in this case the EXISTS operator is used to determine if, given a value of ChildName, there exists at least one sibling. A sibling is defined by a record that has the same value for ParentName, but different value for ChildName.]