

SQL Language

Exercises on JOIN, GROUP BY clauses

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

STUDENT (StudID, SName, City)

COURSE(CourseID, CName, TeacherID)

TEACHER (TeacherID, TName)

EXAM(CourseID, StudID, Date, Grade)

- a) Find the student ID and the maximum, minimum and average exam grade for each student

```
SELECT StudID, MAX(Grade), MIN(Grade), AVG(Grade)
FROM EXAM E
GROUP BY StudID
```

```
SELECT S.StudID, MAX(Grade), MIN(Grade), AVG(Grade)
FROM STUDENT S, EXAM E
WHERE S.StudID = E.StudID
GROUP BY StudID
```

- b) Find the student ID, the name, and the maximum, minimum and average exam grade for each student

```
SELECT S.StudID, SName, MAX(Grade), MIN(Grade), AVG(Grade)
FROM STUDENT S, EXAM E
WHERE S.StudID = E.StudID
GROUP BY StudID, SName
```

- c) For each student with an average grade higher than 28, find the studentID, name, and the maximum, minimum and average exam grade for each student

```
SELECT StudID, SName, MAX(Grade), MIN(Grade), AVG(Grade)
```

```
FROM STUDENT S, EXAM E
WHERE S.StudID = E.StudID
GROUP BY StudID, SName
HAVING AVG(Grade) > 28
```

- d) For each student with an average grade higher than 28 and who has had exams in at least 10 different dates, find the student ID, the name and the maximum, minimum and average exam grade for each student

```
SELECT StudID, SName, MAX(Grade), MIN(Grade), AVG(Grade)
FROM STUDENT S, EXAM E
WHERE S.StudID = E.StudID
GROUP BY StudID, SName
HAVING AVG(Grade) > 28 AND COUNT(DISTINCT Date) >10
```

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

PERSON (TaxID, Name, BirthDate)

PRIVATE_LESSON (TaxID, Date, Hour, InstID)

INSTRUCTOR (InstID, NameI)

- a) For each person view the tax code and the number of lessons attended

```
SELECT P.TaxID, COUNT(*)
FROM PERSON P, PRIVATE_LESSON L
WHERE P.TaxID = L.TaxID
GROUP BY P.TaxID
```

- b) For each person view the tax code, the name and the number of lessons attended

```
SELECT P.TaxID, Name, COUNT(*)
FROM PERSON P, PRIVATE_LESSON L
WHERE P.TaxID = L.TaxID
GROUP BY P.TaxID, Name
```

- c) For each person view the tax code, the name, the number of lessons attended and the number of (different) instructors with whom he or she has done lessons

```
SELECT P.TaxID, Name, COUNT(*), COUNT(DISTINCT InstID)
FROM PERSON P, PRIVATE_LESSON L
WHERE P.TaxID = L.TaxID
GROUP BY P.TaxID, Name
```

- d) For each person born after 1970 who has attended at least 5 lessons, view the tax code, the name, the number of lessons attended and the number of (different) instructors with whom he has taken lessons

```
SELECT P.TaxID, Name, COUNT(*), COUNT(DISTINCT InstID)
FROM PERSON P, PRIVATE_LESSON L
WHERE P.TaxID = L.TaxID AND BirthDate > 31/12/1969
GROUP BY P.TaxID, Name
HAVING COUNT(*) > 4
```

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

COURSE (CourseID, CourseName, Year, Semester)

COURSE_SCHEDULE (CourseID, WeekDay, StartTime, EndTime, Room)

- a) Find the IDs, 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. CourseID , CourseName , SUM ( EndTime - StartTime )
FROM COURSE C, COURSE_SCHEDULE CS
WHERE C. CourseID =CS. CourseID AND Year = 3
GROUP BY C. CourseCode , CourseName
HAVING SUM ( EndTime - StartTime ) >10 AND
COUNT ( DISTINCT WeekDay )=3
```

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

ACCOMMODATION (CodeA, Address, City, Area)

LEASE (CodC, StartDate, EndDate*, PersonName, CodeA, MonthlyPrice)

Note: Area expressed in square meters. For current contracts, EndDate is NULL.

- a) Find the name of people who have entered into more than two rental contracts for the same apartment (at different times).

```
SELECT DISTINCT PersonName
FROM LEASE L
GROUP BY PersonName , CodeA
HAVING COUNT (*) >2

SELECT DISTINCT PersonName
FROM LEASE L
GROUP BY PersonName , CodeA
HAVING COUNT (DISTINCT StartDate) >2
```

- b) Find, for cities where at least 100 contracts have been signed, the city, the maximum monthly cost of rents, the average monthly cost of rents, the maximum duration of contracts, the average duration of contracts and the total number of contracts concluded.

```
SELECT A.City , MAX (L. MonthlyPrice ), AVG (L. MonthlyPrice ),  
MAX (L. EndDate -L. StartDate ),  
AVG (L. EndDate -L. StartDate ), COUNT (*)  
FROM ACCOMODATION A, LEASE L  
WHERE A. CodeA =L. CodeA [ AND EndDate IS NOT NULL ]  
GROUP BY A. City  
HAVING COUNT (*) >=100
```

Nested queries (in, not in, exists, not exists)

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

ORCHESTRA (CodeO, NameO, DirectorName, NoElements)

CONCERT(CodeC, Date, CodeO, CodeH, TicketPrice)

HALL (CodeH, Name, City, Capacity)

- a) Find the code and name of the orchestras with more than 30 elements that have given concerts both in Turin and in Milan and have never held concerts in Bologna.

Solution A

```

SELECT CodeO, NameO
FROM ORCHESTRA
WHERE CodeO NOT IN (
    SELECT CodeO
    FROM CONCERT C, HALL H
    WHERE H.CodeH = C.CodeH AND City = 'Bologna')
AND CodeO IN (SELECT CodeO
    FROM CONCERT C1, HALL H1
    WHERE H1.CodeH = C1.CodeH AND City = 'Turin'
    AND CodeO IN (SELECT CodeO
        FROM CONCERT C2, HALL H2
        WHERE H2.CodeH = C2.CodeH AND City = 'Milan' ))
AND NoElements > 30

```

Solution B

```

SELECT CodeO, NameO
FROM ORCHESTRA
WHERE Code NOT IN (
    SELECT CodeO
    FROM CONCERT C, HALL H
    WHERE H.CodeH = C.CodeH AND City = 'Bologna')
AND Code IN (SELECT C1.CodeO
    FROM CONCERT C1, HALL H1, CONCERT C2, HALL2
    WHERE H1.CodeH = C1.CodeH AND C1.City = 'Turin'
    AND H2.CodeH = C2.CodeH AND C2.City = 'Milan'
    AND C1.CodeO = C2.CodeO)

```

AND NoElements > 30

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

COURSE (CourseID, CourseName, Year, Semester)

COURSE_SCHEDULE (CourseID, WeekDay, StartTime, EndTime, Room)

- a) Find rooms where first-year classes were never held.

Solution A

<pre> SELECT DISTINCT Room FROM COUSE_SCHEDULE CS WHERE Room NOT IN (SELECT Room FROM COURSE_SCHEDULE CS1 , COURSE C WHERE CS1.CourseID =C. CourseID AND Year =1) </pre>

Solution B

<pre> SELECT DISTINCT Room FROM COUSE_SCHEDULE CS WHERE NOT EXISTS (SELECT * FROM COURSE_SCHEDULE CS1 , COURSE C WHERE CS1.CourseID =C. CourseID AND Year =1 AND CS. Room = CS1 . Room) </pre>

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

ACCOMMODATION (CodeA, Address, City, Area)

LEASE (CodC, StartDate, EndDate*, PersonName, CodeA, MonthlyRent)

Note: Area expressed in square meters. For current contracts, EndDate is NULL.

- a) Find the name of people who have never rented accommodation with an area of more than 80 square meters.

Solution A

```
SELECT DISTINCT PersonName
FROM LEASE L
WHERE PersonName NOT IN (
SELECT PersonName
FROM LEASE L1 , ACCOMODATION A
WHERE A. Area >80 AND L1.CodeA =A. CodeA)
```

Solution B

```
SELECT DISTINCT PersonName
FROM LEASE L
WHERE NOT EXISTS (
SELECT *
FROM LEASE L1 , ACCOMODATION A
WHERE A.Area >80 AND L1.CodeA =A. CodeA AND
L. PersonName = L1.PersonName )
```


- b) Find the code and address of the apartments in Turin where the monthly fee has always been higher than 500 euros and for which at most 5 rental contracts have been stipulated.

Solution A

```
SELECT CodeA, Address
FROM ACCOMODATION A
WHERE City = 'Turin ' AND CodeA NOT IN ( SELECT CodeA
      FROM LEASE
      WHERE MonthlyPrice <=500)
AND CodeA IN ( SELECT CodeA
      FROM LEASE
      GROUP BY CodeA
      HAVING COUNT (*) >5)
```

Solution B

```
SELECT CodeA, Address
FROM ACCOMODATION A
WHERE City = 'Turin ' AND NOT EXISTS ( SELECT *
FROM LEASE L
WHERE MonthlyPrice <= 500 AND L.CodeA = A.CodeA)
AND CodeA IN ( SELECT CodeA
      FROM LEASE
      GROUP BY CodeA
      HAVING COUNT (*) >5)
```

Solution C

```
SELECT CodeA, Address
FROM ACCOMODATION A, LEASE L
WHERE City = Turin AND L.CodeA = A.CodeA
GROUP BY A.CodeA, Address
HAVING COUNT (*) <= 5 AND MIN ( MonthlyPrice ) >500
```

Correlated queries

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

ACCOMMODATION (CodeA, Address, City, Area)

LEASE (CodC, StartDate, EndDate*, PersonName, CodeA, MonthlyRent)

Note: Area expressed in square meters. For current contracts, EndDate is NULL.

- a) Find the code, address and city of the accommodations that have an area greater than the average area of the accommodations of the cities in which they are located.

```
SELECT CodeA, Address, City
FROM Accomodation AS A1
WHERE Area > (SELECT AVG(Area)
FROM Accomodation as A2
WHERE A2.City = A1.City)
```

1. Sia dato il seguente schema relazionale (le chiavi primarie sono sottolineate, gli attributi opzionali sono indicati con “*”)

AIRCRAFT (SerialNumber, Model, Capacity)

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

FLIGHTS (Code, SerialNumber, Date, NoReservations)

- a) Find the routes (city of departure, city of arrival) that have never been made with a Boeing-747 aircraft.

```
SELECT DISTINCT (Departure, Destination)
FROM SCHEDULE as S1, FLIGHTS as F1
WHERE S1.Code = F1.Code
AND (Departure, Destination) NOT IN (
    SELECT Departure, Destination
    FROM SCHEDULE as S2, FLIGHTS as F2, AIRCRAFT AS A2
    WHERE S2.Code = F2.Code AND A2.SerialNumber = F2.SerialNumber
    AND Model='Boeing-747')
```