

## Linguaggio SQL: Esercizi

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

WORKSHOP (WSID, Name, Address, City)

VEHICLE (LicensePlate, Model, Brand, Category, Power, YearRegistration, TaxCode)

CUSTOMER (TaxCode, Name, Surname, BirthDate, Address, City)

OVERHAUL (LicensePlate, WSID, Date, Cost)

For workshops that have serviced at least 200 different vehicles registered to people born between 1970 and 1980, display the name and address of the workshop that carried out the most overhauls (including all overhauls) among workshops located in the same city. Also view the total cost of workshop inspections and the number of different vehicle models overhauled.

Possible solutions (Note that other solutions are possible as well)

```
WITH WORKSHOPS_200 AS (
SELECT WSID
FROM OVERHAUL O, VEHICLE V, CUSTOMER C
WHERE O.licensePlate = V.licensePlate AND V.TaxCode = C.TaxCode
AND BirthDate > 1/1/1970 AND BirthDate < 31/12/1980
GROUP BY WSID
COUNT(DISTINCT licensePlate) >= 200)

SELECT W.Name, W.Address, SUM(Cost), COUNT(DISTINCT Model)
FROM OVERHAUL O, WORKSHOP W, VEHICLE V
WHERE W.WID = O.WID AND V.licensePlate = O.licensePlate
AND W.WID IN (
        SELECT WID FROM WORKSHOPS_200
)
GROUP BY W.WID, W.Name, W.Address
HAVING COUNT(*) = (
        SELECT MAX(NumServices)
        FROM (
                SELECT City, COUNT(*) As NumServices
                FROM OVERHAUL O2, WORKSHOP W2
                WHERE O2.WID = W2.WID
                AND W2.City = W.City          <-correlation condition
                GROUP BY W2.WID, City
        ) AS NumServicesCity
        WHERE NumServicesCity.City = W.City   <- correlation condition
```

```

)
SELECT (O. WSID ), W.Name, W.Address, SUM(Cost), COUNT(DISTINCT Model)

FROM OFFICINA O, REVISIONE R, VEICOLO V

FROM OVERHAUL O, WORKSHOP W, VEHICLE V
WHERE W.WID = O.WID AND V.LicensePlate = O.LicensePlate
AND OID IN

    (SELECT WSID
     FROM OVERHAUL O, VEHICLE V, CUSTOMER C
     WHERE O.LicensePlate = V.LicensePlate AND V.TaxCode = C.TaxCode
     AND BirthDate > 1/1/1970 AND BirthDate < 31/12/1980
     GROUP BY WSID
     COUNT(DISTINCT LicensePlate) >= 200
    )

GROUP BY W.WID, W.Name, W.Address
HAVING COUNT(*) = (
    SELECT MAX(NumServices)
    FROM (
        SELECT City, COUNT(*) As NumServices
        FROM OVERHAUL O3, WORKSHOP W3
        WHERE O3.WID = W3.WID
        AND W.City = W3.City          <-correlation condition
        GROUP BY W2.WID, City
    ) AS NumServicesCity
    WHERE NumServicesCity.City = W.City  <- correlation condition

```

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

TECHNICIAN (ID, Name, Surname, BirthDate, Gender, Type)  
INTERVENTION (IntID, Name, Description, HourlyCost)  
BUILDING (BuildingID, Address, City, Province, Region, Type)  
PERFORM\_INTERVENTION (ID, IntID, Date, BuildingID, Duration)

Considering only the buildings located in the province of Turin, view the date in March 2022 in which the highest number of interventions was carried out in the buildings considered.

```

WITH BUILDING_TURIN AS
(SELECT BuildingID
 FROM BUILDING
 WHERE Province='Turin'
 )

WITH NUM_INTERVENTIONS_DATE AS
(SELECT Date, COUNT(*) As NumInt
 FROM PERFORM_INTERVENTION
 WHERE Date >= 1/3/2022 AND Date < 1/04/2022
 AND Building IN (SELECT BuildingID FROM BUILDING_TURIN)
 GROUP BY Date)

SELECT Date
FROM NUM_INTERVENTIONS_DATE
WHERE NumInt = (SELECT MAX(NumInt)
 FROM NUM_INTERVENTIONS )

```

```

SELECT Date
FROM PERFORM_INTERVENTION P, BUILDING B
WHERE P.BuildingID = B.BuildingID AND
Date >= 1/3/2022 AND Date < 1/04/2022
AND Province = 'Turin'
GROUP BY Date
HAVING COUNT(*) = (
SELECT MAX(NumInt) As
FROM
(SELECT COUNT(*) As NumInt
 FROM PERFORM_INTERVENTION P2, BUILDING B2
 WHERE P2.BuildingID = B2.BuildingID AND
 AND Province = 'Turin'
 GROUP BY Date)
)

```

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

LOCATION (LocID, Name, City, Region, CapacityMax)

EVENT (EvID, Title, Type)

EDITION (EvID, Date, LocID, NumberParticipants)

Among the events for which editions have been organized in at least 3 different cities, view the title of the event in which the largest number of people participated overall (considering all editions of the event).

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

FILM (CodF, Title, ReleaseDate, Genre, DurationMinutes)

CINEMA (CodC, Nome, Indirizzo, Città)

HALL (CodC, HallNumber, Capacity)

SCREENING (CodC, HallNumber, Date, StartTime, EndTime, CodF)

Display the title of each film that has a shorter duration than the average duration of films belonging to the same genre, and that has been screened a number of times greater than the average number of screenings of films belonging to the same genre.

**Solution 1**

```
SELECT Title
FROM FILM F, SCREENING S
WHERE S.CodF=F.CodF
AND DurationMinutes <
  (SELECT AVG(DurationMinutes)
   FROM FILM F2
   WHERE F2.Genre =F.Genre)
GROUP BY F.CodF, Title, Genre
HAVING COUNT(*) > (SELECT AVG(N)
FROM (SELECT Genre, COUNT (*) AS N
      FROM SCREENING S2, FILM F3
      WHERE S2.CodF=F3.CodF
      GROUP BY F3.CodF, Genre) AS SG
WHERE SG.Genre=F.Genre)
```

**Solution 2 (CTE)**

```
WITH SCREENING-FILM AS
  (SELECT F.CodF, Title, Genre, COUNT (*) AS N
   FROM FILM F, SCREENING S
   WHERE S.CodF=F.CodF
```

```
GROUP BY F.CodF, Genre)
```

```
DURATION-GENRE AS
```

```
(SELECT Genre, AVG(DurationMinutes) AS AvgDuration
```

```
FROM FILM F
```

```
GROUP BY Genre)
```

```
SCREENING-GENRE AS (
```

```
SELECT Genre, AVG(N) AS AvgGenre
```

```
FROM SCREENING-FILM
```

```
GROUP BY Genre
```

```
)
```

```
SELECT Title
```

```
FROM SCREENING-FILM P, DURATION-GENRE D, SCREENING-GENRE G
```

```
WHERE D.Genre=F.Genre
```

```
AND G.Genre=P.Genre
```

```
AND DurationMinutes < AvgDuration
```

```
AND N > AvgGenre
```