
Esercitazione SQL

17/10/2024



ESERCIZIO 1

SCHEMA 1

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

- PERSONA (CodFisc, NomeP, DataNascita)
 - ISTRUTTORE (CodI, NomeI)
 - LEZIONE_PRIVATA (CodFisc, Data, Ora, CodI)
-

RICHIESTE 1.A, 1.B

- **PERSONA (CodFisc, NomeP, DataNascita)**
- **ISTRUTTORE (CodI, NomeI)**
- **LEZIONE_PRIVATA (CodFisc, Data, Ora, CodI)**

1.A) Per ogni persona visualizzare il codice fiscale e il numero di lezioni frequentate.

1.B) Per ogni persona visualizzare il codice fiscale, il nome e il numero di lezioni frequentate

SOLUZIONE 1.A

```
SELECT CodFisc, COUNT(*)  
FROM LEZIONE_PRIVATA LP  
GROUP BY CodFisc
```

SOLUZIONE 1.B

```
SELECT P.CodFisc, NomeP, COUNT(*)  
FROM LEZIONE_PRIVATA LP, PERSONA P  
WHERE P.CodFisc = LP.CodFisc  
GROUP BY P.CodFisc, NomeP
```

SOLUZIONE 1.A

```
SELECT CodFisc, COUNT(*)  
FROM LEZIONE_PRIVATA LP  
GROUP BY CodFisc
```

SOLUZIONE 1.B

```
SELECT P.CodFisc, NomeI, COUNT(*)  
FROM LEZIONE_PRIVATA LP, PERSONA P  
WHERE P.CodFisc = LP.CodFisc  
GROUP BY P.CodFisc, NomeF
```

SOLUZIONE 1.A

```
SELECT CodFisc, COUNT(*)  
FROM LEZIONE_PRIVATA LP  
GROUP BY CodFisc
```

SOLUZIONE 1.B

```
SELECT P.CodFisc, NomeF, CO  
FROM LEZIONE_PRIVATA LP, P  
WHERE P.CodFisc = LP.CodFisc  
GROUP BY P.CodFisc, NomeF
```

Regola SQL: Colonne nella SELECT e GROUP BY

Quando si utilizza GROUP BY, tutte le colonne elencate nella SELECT devono soddisfare una delle seguenti condizioni:

- 1. Essere presenti nella clausola GROUP BY, perché sono colonne di raggruppamento.**
- 2. Essere usate in una funzione di aggregazione (come MAX, MIN, AVG, COUNT), che restituisce un valore calcolato per ciascun gruppo.**

RICHIESTE 1.C, 1.D

- **PERSONA (CodFisc, NomeP, DataNascita)**
- **ISTRUTTORE (CodI, NomeI)**
- **LEZIONE_PRIVATA (CodFisc, Data, Ora, CodI)**

1.C) Per ogni persona visualizzare il codice fiscale, il nome, il numero di lezioni frequentate e il numero di istruttori (diversi) con cui ha fatto lezione.

1.D) Per ogni persona nata dopo il 1970 che ha frequentato almeno 5 lezioni, visualizzare il codice fiscale, il nome, il numero di lezioni frequentate e il numero di istruttori (diversi) con cui ha fatto lezione

SOLUZIONE 1.C

```
SELECT P.CodFisc, NomeP, COUNT(*), COUNT(DISTINCT CodI)  
FROM LEZIONE_PRIVATA LP, PERSONA P  
WHERE P.CodFisc = LP.CodFisc  
GROUP BY P.CodFisc, Nome
```

SOLUZIONE 1.D

```
SELECT P.CodFisc, NomeP, COUNT(*), COUNT(DISTINCT CodI)  
FROM LEZIONE_PRIVATA LP, PERSONA P  
WHERE P.CodFisc = LP.CodFisc AND DataNascita > 1970  
GROUP BY P.CodFisc, NomeP  
HAVING COUNT(*)>=5
```



ESERCIZIO 2

RICHIESTA 2.A

- **ALLOGGIO (CodA, Indirizzo, Città, Superficie)**
- **CONTRATTO-AFFITTO (CodC, DataInizio, DataFine*, NomePersona, CodA, RettaMensile)**

2.A) Trovare il nome delle persone che non hanno mai affittato alloggi con superficie superiore a 80 metri quadri.

SOLUZIONE 2.A con NOT IN

**SELECT DISTINCT NomePersona
FROM CONTRATTO-AFFITTO**

WHERE NomePersona NOT IN

**(SELECT CA.NomePersona
FROM CONTRATTO-AFFITTO CA, ALLOGGIO A
WHERE A.Superficie > 80 AND CA.CodA = A.CodA)**

SOLUZIONE 2.A con NOT IN

```
SELECT DISTINCT NomePers  
FROM CONTRATTO-AFFITTO  
WHERE NomePersona NOT  
(SELECT CA.Nome  
FROM CONTRATTO-AFFITTO  
WHERE A.Superficie
```

Regola: Uso di NOT IN per Escludere Risultati

Quando si tratta di query in cui si desidera **escludere** determinate righe basandosi su una condizione presente in un'altra tabella, **NOT IN** è spesso la scelta migliore perché permette di selezionare tutti i record che **non appaiono** in un elenco di risultati che soddisfano una determinata condizione.

SOLUZIONE 2.A con NOT EXIST

```
SELECT DISTINCT NomePersona  
FROM CONTRATTO-AFFITTO CA1  
WHERE NOT EXISTS (SELECT *  
FROM CONTRATTO-AFFITTO CA, ALLOGGIO A  
WHERE A.Superficie > 80  
AND CA.CodA = A.CodA  
AND CA.NomePersona= CA1.NomePersona)
```

CONDIZIONE DI CORRELAZIONE

SOLUZIONE 2.A con NOT

```
SELECT DISTINCT NomePersona
FROM CONTRATTO-AFFITTO C
WHERE NOT EXISTS (SELECT *
                   FROM CONTRATTO
                   WHERE A.Superficie < C.Superficie
                   AND CA.CodA = A.CodA
                   AND CA.NomePer
```

CONDIZIONI

Regola: Uso di NOT EXISTS per Verificare l'Assenza di Corrispondenze

NOT EXISTS viene utilizzato quando si desidera **escludere** righe che hanno una relazione con altre tabelle basata su una certa condizione. In altre parole, vuoi ottenere risultati dalla tabella principale solo se **non esiste alcuna riga** nella sottoquery correlata che soddisfi la condizione specificata.

RICHIESTA 2.B

- **ALLOGGIO (CodA, Indirizzo, Città, Superficie)**
- **CONTRATTO-AFFITTO (CodC, DataInizio, DataFine*, NomePersona, CodA, RettaMensile)**

2.B) Trovare il codice e l'indirizzo degli appartamenti di Torino in cui la retta mensile è sempre stata superiore a 500 euro e per cui sono stati stipulati al più 5 contratti di affitto

SOLUZIONE 2.B

```
SELECT A.CodA, Indirizzo  
FROM APPARTAMENTO A, CONTRATTO-AFFITTO CA  
WHERE A.CodA=CA.CodA AND Citta='Torino'  
AND A.CodA NOT IN  
      (SELECT CodA  
      FROM CONTRATTO-AFFITTO  
      WHERE RettaMensile<=500)  
GROUP BY A.CodA, Indirizzo  
HAVING COUNT(*)<=5
```



ESERCIZIO 3

RICHIESTE 3.A

- CORSO (CodCorso, NomeC, Anno, Semestre)
- ORARIO-LEZIONI (CodCorso, GiornoSettimana, Orainizio, OraFine, Aula)

3.A) Trovare codice corso, nome corso e numero totale di ore di lezione settimanali per i corsi del terzo anno per cui il numero complessivo di ore di lezione settimanali è superiore a 10 e le lezioni sono in più di tre giorni diversi della settimana.

SOLUZIONE 3.A

```
SELECT C.CodCorso, NomeC, SUM(OraFine-Orainizio)  
FROM CORSO C, ORARIO-LEZIONI OL  
WHERE C.CodCorso = OL.CodCorso AND Anno = 3  
GROUP BY C.CodCorso, NomeC  
HAVING SUM(OraFine-Orainizio) >10 AND COUNT(DISTINCT  
GiornoSettimana)>3
```
