Esercitazione SQL: soluzioni

Livello 1: join e filtri

Richiesta 1

```
SELECT B.titolo, AR.nome, B.durata_secondi
FROM BRANI B, ARTISTI AR
WHERE B.artista = AR.id_artista
  AND B.genere IN ('Rock progressivo', 'Hard rock')
  AND B.durata_secondi > 240
ORDER BY B.durata_secondi DESC;
```

Risultato atteso, 4 righe: Comfortably Numb (Pink Floyd, 382), Wish You Were Here (Pink Floyd, 334), Thunderstruck (AC/DC, 292), Back in Black (AC/DC, 255).

Richiesta 2

```
SELECT S.nickname, B.titolo
FROM STUDENTI S
     JOIN ASCOLTI A ON A.studente = S.id_studente
     JOIN BRANI B   ON A.brano    = B.id_brano
WHERE S.corso_studio = 'Data Science'
  AND A.secondi_ascoltati = B.durata_secondi
ORDER BY S.nickname, B.titolo;
```

Risultato atteso, 18 righe. Le prime: echo con COME VUOI, Comfortably Numb, Don't Stop Me Now, I P' ME, TU P' TE, Non dirgli mai, Thunderstruck. Partenope compare due volte con I P' ME, TU P' TE, perché lo ha ascoltato due volte per intero.

Richiesta 3

```
SELECT S.nickname, B.titolo, AR.nome, A.data_ora
FROM ASCOLTI A
     JOIN STUDENTI S ON A.studente = S.id_studente
     JOIN BRANI B    ON A.brano    = B.id_brano
     JOIN ARTISTI AR ON B.artista  = AR.id_artista
WHERE EXTRACT(YEAR FROM A.data_ora)  = 2026
  AND EXTRACT(MONTH FROM A.data_ora) = 4
ORDER BY A.data_ora;
```

Risultato atteso, 12 righe, dalla prima (aurora, Another Brick in the Wall, Part 2, Pink Floyd, 1 aprile 16:46) all'ultima (rocky, Thunderstruck, AC/DC, 23 aprile 14:04).

Livello 2: aggregazioni

Richiesta 4

```
SELECT genere,
       COUNT(*) AS num_brani,
       ROUND(AVG(durata_secondi), 1) AS durata_media
FROM BRANI
GROUP BY genere
ORDER BY num_brani DESC, genere;
```

genere	num_brani	durata_media
Rock	6	297,8
Hard rock	3	251,7
Pop	3	249,3
Rap	3	189,7
Rock progressivo	3	318,0

Richiesta 5

```
SELECT AR.nome,
       COUNT(DISTINCT A.studente) AS num_studenti,
       SUM(A.secondi_ascoltati)   AS secondi_totali
FROM ARTISTI AR
     JOIN BRANI B   ON B.artista = AR.id_artista
     JOIN ASCOLTI A ON A.brano   = B.id_brano
WHERE AR.tipo_artista = 'GRUPPO'
GROUP BY AR.nome
HAVING COUNT(DISTINCT A.studente) > 5
ORDER BY secondi_totali DESC;
```

nome	num_studenti	secondi_totali
Pink Floyd	6	4162
Dire Straits	7	4094
Queen	8	3347

Gli AC/DC sono esclusi dall'HAVING: hanno 5 studenti distinti.

Richiesta 6

```
SELECT B.titolo,
       ROUND(AVG(A.secondi_ascoltati / B.durata_secondi) * 100, 1) AS
perc_media
FROM BRANI B JOIN ASCOLTI A ON A.brano = B.id_brano
GROUP BY B.id_brano, B.titolo
HAVING AVG(A.secondi_ascoltati / B.durata_secondi) < 0.9
ORDER BY perc_media, B.titolo;
```

titolo	perc_media
Back in Black	82,4
Non dirgli mai	83,3
Another Brick in the Wall, Part 2	85,0
Bohemian Rhapsody	86,6
COME VUOI	87,0

Livello 3: join esterni

Richiesta 7

```
SELECT B.titolo, COUNT(A.id_ascolto) AS num_ascolti
FROM BRANI B LEFT JOIN ASCOLTI A ON A.brano = B.id_brano
GROUP BY B.id_brano, B.titolo
ORDER BY num_ascolti, B.titolo;
```

Risultato atteso, 18 righe. Le prime: DIO LO SA (0), Highway to Hell (2), Money for Nothing (2), Quanti amori (2). L'ultima: I P' ME, TU P' TE (9).

Richiesta 8

```
SELECT S.corso_studio,
       COUNT(DISTINCT S.id_studente) AS num_studenti,
       COUNT(A.id_ascolto)           AS num_ascolti
FROM STUDENTI S LEFT JOIN ASCOLTI A ON A.studente = S.id_studente
GROUP BY S.corso_studio
ORDER BY num_ascolti DESC;
```

corso_studio	num_studenti	num_ascolti
Informatica	4	33

corso_studio	num_studenti	num_ascolti
Data Science	3	20
Ingegneria informatica	3	19
Matematica	1	0

Livello 4: massimi e pari merito

Richiesta 9

```
SELECT B.titolo, AR.nome, B.durata_secondi
FROM BRANI B JOIN ARTISTI AR ON B.artista = AR.id_artista
WHERE B.durata_secondi = (SELECT MAX(durata_secondi) FROM BRANI);
```

Risultato atteso: Money for Nothing (Dire Straits, 505).

Richiesta 10

```
SELECT B.titolo, COUNT(*) AS num_ascolti
FROM BRANI B JOIN ASCOLTI A ON A.brano = B.id_brano
GROUP BY B.id_brano, B.titolo
HAVING COUNT(*) = (SELECT MAX(x)
                   FROM (SELECT COUNT(*) AS x
                         FROM ASCOLTI
                         GROUP BY brano));
```

Risultato atteso: I P′ ME, TU P′ TE, con 9 ascolti. Il massimo è unico.

Richiesta 11

```
SELECT S.nickname, COUNT(*) AS num_ascolti
FROM STUDENTI S JOIN ASCOLTI A ON A.studente = S.id_studente
GROUP BY S.nickname
HAVING COUNT(*) = (SELECT MAX(x)
                   FROM (SELECT COUNT(*) AS x
                         FROM ASCOLTI
                         GROUP BY studente))
ORDER BY S.nickname;
```

Risultato atteso, 2 righe: ada.exe (9) e bytebeat (9). Qui il massimo è raggiunto da due studenti.

Per confronto, la stessa richiesta con ORDER BY num_ascolti DESC e LIMIT 1 (che in Oracle non esiste, ma funziona su SQLite, MySQL e PostgreSQL) restituirebbe una sola riga, perché LIMIT taglia senza sapere che il secondo è a pari merito con il primo.

Livello 5: sottoquery

Richiesta 12

```
SELECT nickname
FROM STUDENTI
WHERE id_studente IN (SELECT A.studente
                      FROM ASCOLTI A
                           JOIN BRANI B    ON A.brano   = B.id_brano
                           JOIN ARTISTI AR ON B.artista = AR.id_artista
                      WHERE AR.paese = 'Italia')
ORDER BY nickname;
```

Risultato atteso, 8 righe: ada.exe, aurora, bytebeat, clef, echo, luna.sql, partenope, synthwave.

Richiesta 13

```
SELECT B.titolo, B.genere, B.durata_secondi
FROM BRANI B
WHERE B.durata_secondi > (SELECT AVG(B2.durata_secondi)
                         FROM BRANI B2
                         WHERE B2.genere = B.genere)
ORDER BY B.genere, B.durata_secondi DESC;
```

titolo	genere	durata_secondi
Thunderstruck	Hard rock	292
Back in Black	Hard rock	255
Non dirgli mai	Pop	270
DIO LO SA	Rap	210
Money for Nothing	Rock	505
Bohemian Rhapsody	Rock	354
Sultans of Swing	Rock	348
Comfortably Numb	Rock progressivo	382

titolo	genere	durata_secondi
Wish You Were Here	Rock progressivo	334

Richiesta 14

```
SELECT S.nickname
FROM STUDENTI S
WHERE NOT EXISTS (SELECT *
                  FROM ASCOLTI A JOIN BRANI B ON A.brano = B.id_brano
                  WHERE A.studente = S.id_studente
                    AND B.genere = 'Rap')
ORDER BY S.nickname;
```

Risultato atteso, 4 righe: clef, floydfan, rocky, silenzio. Silenzio compare perché non ha ascoltato niente.

Richiesta 15

```
SELECT S.nickname
FROM STUDENTI S
WHERE NOT EXISTS (
    SELECT *
    FROM BRANI B JOIN ARTISTI AR ON B.artista = AR.id_artista
    WHERE AR.nome = 'Pink Floyd'
      AND NOT EXISTS (SELECT *
                      FROM ASCOLTI A
                      WHERE A.studente = S.id_studente
                        AND A.brano    = B.id_brano))
ORDER BY S.nickname;
```

Risultato atteso, 2 righe: aurora, floydfan.

Livello 6: sfide

Richiesta 16

```
CREATE OR REPLACE VIEW V_ASCOLTI_DETTAGLIO AS
SELECT S.nickname, S.corso_studio, B.titolo, B.genere,
       AR.nome AS artista, A.data_ora, A.secondi_ascoltati
FROM ASCOLTI A
     JOIN STUDENTI S ON A.studente = S.id_studente
     JOIN BRANI B    ON A.brano    = B.id_brano
     JOIN ARTISTI AR ON B.artista  = AR.id_artista;

SELECT artista,
       COUNT(*) AS num_ascolti,
       SUM(secondi_ascoltati) AS secondi_totali
FROM V_ASCOLTI_DETTAGLIO
WHERE corso_studio = 'Data Science'
GROUP BY artista
ORDER BY num_ascolti DESC, artista;
```

artista	num_ascolti	secondi_totali
Geolier	7	1264
Gigi D'Alessio	5	1108
Queen	3	685
AC/DC	2	412
Dire Straits	2	597
Pink Floyd	1	382

Richiesta 17

```
SELECT S.nickname
FROM STUDENTI S JOIN ASCOLTI A ON A.studente = S.id_studente
     JOIN BRANI B ON A.brano = B.id_brano JOIN ARTISTI AR ON B.artista =
AR.id_artista
WHERE AR.nome = 'Pink Floyd'
MINUS
SELECT S.nickname
FROM STUDENTI S JOIN ASCOLTI A ON A.studente = S.id_studente
     JOIN BRANI B ON A.brano = B.id_brano JOIN ARTISTI AR ON B.artista =
AR.id_artista
WHERE AR.nome = 'Queen';
```

Risultato atteso: luna.sql.

```
SELECT S.nickname
FROM STUDENTI S JOIN ASCOLTI A ON A.studente = S.id_studente
     JOIN BRANI B ON A.brano = B.id_brano JOIN ARTISTI AR ON B.artista =
AR.id_artista
WHERE AR.nome = 'Queen'
INTERSECT
SELECT S.nickname
FROM STUDENTI S JOIN ASCOLTI A ON A.studente = S.id_studente
     JOIN BRANI B ON A.brano = B.id_brano JOIN ARTISTI AR ON B.artista =
AR.id_artista
WHERE AR.nome = 'AC/DC'
ORDER BY 1;
```

Risultato atteso, 5 righe: ada.exe, bytebeat, echo, rocky, synthwave.

Richiesta 18

```
WITH mesi (m) AS (
    SELECT 1 FROM dual
    UNION ALL
    SELECT m + 1 FROM mesi WHERE m < 12
)
SELECT m AS mese, COUNT(A.id_ascolto) AS num_ascolti
FROM mesi LEFT JOIN ASCOLTI A
     ON EXTRACT(MONTH FROM A.data_ora) = m
GROUP BY m
ORDER BY m;
```

Risultato atteso: 12 ascolti in ciascuno dei mesi da 2 a 7 (febbraio a luglio) e 0 negli altri.