

Esercizio 9 – Query complesse

- × Utilizzare, come per l'esercizio 7, il database creato dallo script `universita.sql`
- × Scrivere ed eseguire le seguenti query **utilizzando l'operatore di join esplicito, le clausole di ordinamento, gli operatori di aggregazione, i raggruppamenti, le interrogazioni nidificate e gli operatori insiemistici** (viene fornito il numero di righe per la soluzione corretta):

1. Stilare una classifica degli studenti (con cognome, nome, matricola e voto) che hanno sostenuto l'esame con codice E3101Q117 con un voto maggiore di 23, ordinandola per voto decrescente [112 righe]

```
SELECT s.Nome, s.Cognome, s.Matricola_st, e.Voto
FROM Studente AS s
  INNER JOIN Esame AS e ON s.Matricola_st = e.Matricola_studente
WHERE e.Codice_corso = "E3101Q117" AND e.Voto > 23
ORDER BY e.Voto DESC;
```

2. Contare il numero di studenti che hanno superato l'esame con codice E3101Q117 con un voto maggiore di 23 [1 riga, 112 studenti]

```
SELECT COUNT(*)
FROM Esame AS e
WHERE e.Codice_corso = "E3101Q117" AND e.Voto > 23;
```

Esercizio 9 – Query complesse

3. Selezionare gli studenti (con nome, cognome e matricola) con la media dei voti degli esami, e ordinarli secondo la media in ordine decrescente [700 righe]

```
SELECT s.Nome, s.Cognome, s.Matricola_st, AVG(e.Voto) AS media
FROM Studente AS s
  INNER JOIN Esame AS e ON s.Matricola_st = e.Matricola_studente
GROUP BY s.Matricola_st
ORDER BY media DESC;
```

Esercizio 9 – Query complesse

4. Selezionare gli studenti (con nome, cognome e matricola) che iniziano con la lettera B e il numero di esami svolti [14 righe]

```
SELECT s.Nome, s.Cognome, s.Matricola_st, COUNT(*) AS n_esami
FROM Studente AS s
  INNER JOIN Esame AS e ON s.Matricola_st = e.Matricola_studente
GROUP BY s.Matricola_st
HAVING s.Nome LIKE "B%";
```

- Alternativa equivalente (gli studenti che iniziano con la lettera B vengono filtrati prima del raggruppamento):

```
SELECT s.Nome, s.Cognome, s.Matricola_st, COUNT(*) AS n_esami
FROM Studente AS s
  INNER JOIN Esame AS e ON s.Matricola_st = e.Matricola_studente
WHERE s.Nome LIKE "B%"
GROUP BY s.Matricola_st;
```

Esercizio 9 – Query complesse

5. Selezionare gli studenti (con nome, cognome e matricola) che hanno svolto l'esame con codice E3101Q020 e/o con codice E3101Q117 [298 righe]

```
(SELECT s.Cognome, s.Nome, s.Matricola_st
FROM Studente AS s JOIN Esame AS e ON e.Matricola_studente = s.Matricola_st
WHERE e.Codice_corso = "E3101Q020")
UNION
(SELECT s1.Cognome, s1.Nome, s1.Matricola_st
FROM Studente AS s1 JOIN Esame AS e1 ON e1.Matricola_studente = s1.Matricola_st
WHERE e1.Codice_corso = "E3101Q117");
```

- Alternativa equivalente:

```
SELECT DISTINCT s.Cognome, s.Nome, s.Matricola_st
FROM Studente AS s INNER JOIN Esame AS e ON s.Matricola_st = e.Matricola_studente
WHERE e.Codice_corso = "E3101Q020" OR e.Codice_corso = "E3101Q117";
```

Esercizio 9 – Query complesse

6. Selezionare gli studenti (con nome, cognome e matricola) che hanno svolto l'esame con codice E3101Q020 e con codice E3101Q117, usando interrogazioni nidificate [41 righe]

- Utilizzando solamente il join:

```
SELECT s.Cognome, s.Nome, s.Matricola_st
FROM Studente AS s
      INNER JOIN Esame AS e ON e.Matricola_studente = s.Matricola_st
      INNER JOIN Esame AS e1 ON e1.Matricola_studente = s.Matricola_st
WHERE e.Codice_corso = "E3101Q020" AND e1.Codice_corso = "E3101Q117";
```

- Utilizzando interrogazioni nidificate:

```
SELECT t1.Cognome, t1.Nome, t1.Matricola_st
FROM (
      SELECT * FROM Studente AS s
            INNER JOIN Esame AS e ON e.Matricola_studente = s.Matricola_st
            WHERE e.Codice_corso = "E3101Q020"
) AS t1
INNER JOIN (
      SELECT * FROM Studente AS s1
            INNER JOIN Esame AS e1 ON e1.Matricola_studente = s1.Matricola_st
            WHERE e1.Codice_corso = "E3101Q117"
) AS t2 ON t1.Matricola_st = t2.Matricola_st;
```

Esercizio 9 – Query complesse

7. Visualizzare i corsi (con codice e nome) di cui lo studente con matricola 1492601 non ha ancora sostenuto l'esame [14 righe]

```
SELECT c.Codice_c, c.Nome
FROM Corso AS c
    INNER JOIN Piano_di_studio AS p ON p.Codice_corso = c.Codice_c
    INNER JOIN Studente AS s ON p.Matricola_st = s.Matricola_st
WHERE s.Matricola_st = "1492601" AND c.Codice_c NOT IN (
    SELECT e1.Codice_corso
    FROM Esame AS e1
        INNER JOIN Studente AS s1 ON e1.Matricola_studente = s1.Matricola_st
    WHERE s1.Matricola_st = "1492601"
);
```

Esercizio 9 – Query complesse

8. Visualizzare i corsi (con codice e nome) il cui esame deve ancora essere sostenuto da almeno uno studente (rispetto al relativo piano di studi) e il numero di studenti che li devono sostenere. Si considerino solo i corsi della laurea triennale e si ordinino i risultati in ordine decrescente per numero di studenti [31 righe]

```
SELECT c.Codice_c, c.Nome, (COUNT(*) - esami_svolti) AS da_sostenere
FROM Corso AS c
    INNER JOIN Piano_di_studio AS p ON p.Codice_corso = c.Codice_c
    INNER JOIN (
        SELECT e1.Codice_corso AS corso, COUNT(*) AS esami_svolti
        FROM Esame AS e1
        GROUP BY e1.Codice_corso
    ) AS es ON c.Codice_c = es.Corso
    INNER JOIN Corso_di_Laurea_corsi AS cl ON cl.Codice_c = c.Codice_c
WHERE cl.idCorso_di_Laurea = (
    SELECT cl1.idCorso_di_Laurea FROM Corso_di_Laurea AS cl1 WHERE cl1.Tipologia = "Triennale"
)
GROUP BY c.Codice_c
ORDER BY da_sostenere DESC;
```

Esercizio 9 – Query complesse

9. Visualizzare i corsi (con codice e nome) con la media dei voti conseguiti dagli studenti nei relativi esami, e il numero di studenti che li hanno sostenuti. Selezionare solo i corsi della laurea magistrale il cui esame è stato superato da almeno 60 studenti. Si ordinino i risultati in ordine decrescente rispetto alla media dei voti [7 righe]

```
SELECT c.Codice_c, c.Nome, AVG(e.Voto) AS media, COUNT(*) AS numero_studenti
FROM Corso AS c
      INNER JOIN Esame AS e ON e.Codice_corso = c.Codice_c
      INNER JOIN Corso_di_Laurea_corsi AS cl ON cl.Codice_c = c.Codice_c
WHERE cl.idCorso_di_Laurea = (
      SELECT cl1.idCorso_di_Laurea
      FROM Corso_di_Laurea AS cl1
      WHERE cl1.Tipologia = "Magistrale"
)
GROUP BY c.Codice_c
HAVING numero_studenti >= 60
ORDER BY media DESC;
```

Esercizio 9 – Query complesse

10. Selezionare il miglior studente in termini di media dei voti (con cognome, nome, matricola, media e numero esami sostenuti) della laurea triennale [1 riga, studente: Accarino Nicola]

```
SELECT t1.Cognome, t1.Nome, t1.Matricola_st, t1.media, t1.esami_sostenuti
FROM (
  SELECT s.Cognome, s.Nome, s.Matricola_st, AVG(e.Voto) AS media, COUNT(*) AS esami_sostenuti
  FROM Studente AS s
    INNER JOIN Esame AS e ON e.Matricola_studente = s.Matricola_st
    INNER JOIN Corso_di_Laurea AS cd1 ON s.Corso_di_laurea = cd1.idCorso_di_Laurea
  WHERE cd1.Tipologia = "Triennale"
  GROUP BY s.Matricola_st
) AS t1
WHERE t1.media = (
  SELECT MAX(t2.media1) FROM (
    SELECT s1.Matricola_st, AVG(e1.Voto) AS media1
    FROM Studente s1
      INNER JOIN Esame AS e1 ON e1.Matricola_studente = s1.Matricola_st
      INNER JOIN Corso_di_Laurea AS cd11 ON s1.Corso_di_laurea = cd11.idCorso_di_Laurea
    WHERE cd11.Tipologia = "Triennale"
    GROUP BY s1.Matricola_st
  ) AS t2
);
```

Esercizio 9 – Query complesse

11. Visualizzare le città di residenza (con nome e regione) e il numero di studenti che provengono da tali città. Ordinare i risultati in ordine decrescente rispetto al numero di studenti [298 righe]

```
SELECT c.Nome, c.Regione, COUNT(*) AS numero_studenti
FROM Citta AS c
      INNER JOIN Studente AS s ON s.Citta_residenza = c.idCitta
GROUP BY c.Nome, c.Regione
ORDER BY numero_studenti DESC;
```

Esercizio 9 – Query complesse

12. Visualizzare gli studenti (con cognome, nome, matricola) della triennale e il numero di crediti che hanno finora ottenuto, in ordine decrescente per numero di crediti ottenuti [531 righe]

- Utilizzando solamente il join:

```
SELECT s.Cognome, s.Nome, s.Matricola_st,  
       SUM(c.Crediti_lezione + c.Crediti_esercitazione) AS crediti_totali  
FROM Studente AS s  
     INNER JOIN Esame AS e ON e.Matricola_studente = s.Matricola_st  
     INNER JOIN Corso AS c ON c.codice_c = e.Codice_corso  
     INNER JOIN Corso_di_Laurea AS cd1 ON s.Corso_di_laurea = cd1.idCorso_di_Laurea  
WHERE cd1.Tipologia = "Triennale" GROUP BY s.Matricola_st  
ORDER BY Crediti_totali DESC;
```

- Utilizzando interrogazioni nidificate:

```
SELECT s.Cognome, s.Nome, s.Matricola_st, SUM(c.Crediti) AS crediti_totali  
FROM Studente AS s  
     INNER JOIN Esame AS e ON e.Matricola_studente = s.Matricola_st  
     INNER JOIN (  
         SELECT c1.Codice_c, c1.Nome, (c1.Crediti_lezione + c1.Crediti_esercitazione) AS crediti  
         FROM Corso AS c1  
     ) AS c ON c.Codice_c = e.Codice_corso  
WHERE s.Corso_di_Laurea = (  
     SELECT c11.idCorso_di_Laurea FROM Corso_di_Laurea c11 WHERE c11.Tipologia = "Triennale"  
)  
GROUP BY s.Matricola_st ORDER BY crediti_totali DESC;
```