

Cognome e nome: _____ Matricola: _____ Turno: _____

Riportare sui fogli i seguenti dati: cognome, nome, matricola e turno di laboratorio.

Esame di SQL

Punteggi massimi:

- Domande 1 e 2 svolte perfettamente: 23;
- Domande 1 e 3 svolte perfettamente: 25;
- Domande 2 e 3 svolte perfettamente: 28;
- Domande 1, 2 e 3 svolte perfettamente: 33.

Lo svolgimento corretto di una sola domanda non permette il raggiungimento della sufficienza.

Le seguenti relazioni definiscono una base di dati “**DoveStudio**” per gestire una piattaforma di prenotazione aule studio per gli atenei della città. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

STUDENTE(Matricola, Ateneo, Cognome, Nome, DataNascita, Gruppo*)

AULA(Codice, Nome, Indirizzo, NumPosti)

PRENOTAZIONE(Matricola, Ateneo, Data, Aula, FasciaOraria, Gruppo*)

GRUPPO(NomeGruppo, DataCreazione)

Valgono i seguenti vincoli UNIQUE:

- UNIQUE(Data, Gruppo) in PRENOTAZIONE
- UNIQUE(Matricola, Ateneo, Gruppo) in STUDENTE

Vincoli di integrità referenziale:

STUDENTE(Gruppo) referencia GRUPPO(NomeGruppo)

PRENOTAZIONE(Matricola, Ateneo) referenziano STUDENTE(Matricola, Ateneo),

PRENOTAZIONE(Aula) referencia AULA(Codice),

PRENOTAZIONE(Matricola, Ateneo, Gruppo) referencia STUDENTE(Matricola, Ateneo, Gruppo).

FasciaOraria può assumere i valori “mattino”, “pomeriggio”, “intera giornata”.

Ogni studentessa o studente può prenotare un’aula a proprio nome o a nome di un gruppo di studio. In questo ultimo caso, l’attributo Gruppo conterrà il riferimento al suo gruppo di studio e verranno riservati un numero di posti nell’aula pari al numero di componenti del gruppo. In PRENOTAZIONE ci sarà, però, solo la tupla relativa alla persona che ha effettuato la prenotazione.

Le date rappresentate da stringhe nel formato 'AAAAMMGG'. I rimanenti attributi sono autoesplicativi.

Con riferimento alla base di dati "DoveStudio" esprimere in SQL le seguenti interrogazioni.

Domanda 1 (bassa complessità).

Mostrare i gruppi di studio con componenti di almeno due atenei diversi, creati nel 2022 e che hanno prenotato aule nel 2023.

Soluzione 1.

```
SELECT NomeGruppo
FROM gruppo g JOIN studente s1 ON (NomeGruppo=s1.Gruppo) JOIN studente s2 ON
(NomeGruppo=s2.Gruppo AND s1.Ateneo<>s2.Ateneo)
JOIN prenotazione p ON (NomeGruppo=p.Gruppo)
WHERE DataCreazione LIKE '2022%' AND p.Data LIKE '2023%';
```

Domanda 2 (media complessità).

Elencare le aule (nome e indirizzo) che hanno più di 100 posti con prenotazioni fatte, anche in date diverse, da almeno cinque gruppi di studio diversi. Mostrare anche il numero di prenotazioni effettuate da gruppi per ogni aula elencata. **Non usare sottoquery.**

Soluzione 2.

```
SELECT a.Nome, a.Indirizzo, COUNT(DISTINCT p.Data)
FROM aula a JOIN prenotazione p ON (a.Codice=p.Aula)
WHERE p.Gruppo IS NOT NULL AND a.NumPosti > 100
GROUP BY a.Codice
HAVING COUNT(DISTINCT p.Gruppo)>=5;
```

Domanda 3 (alta complessità).

Mostrare i gruppi di studio in cui **ogni componente** ha effettuato almeno una prenotazione a nome del gruppo, ma solo nel 2020 (cioè non ha effettuato prenotazioni a nome del gruppo in altri anni). **È possibile utilizzare gli operatori insiemistici.**

Soluzione 3.

```
WITH PrenotazioniGruppi2020 AS (
    SELECT *
    FROM prenotazione p
    WHERE p.Data like '2020%' and p.Gruppo IS NOT NULL and NOT EXISTS (
        SELECT *
        FROM prenotazione p1
        WHERE NOT (p1.Data like '2020%') AND p1.Gruppo IS NOT NULL AND
            p1.Ateneo=p.Ateneo AND p1.Matricola=p.Matricola)
),
SELECT DISTINCT p.Gruppo
FROM PrenotazioniGruppi2020 p
WHERE NOT EXISTS (
    SELECT *
    FROM studente s
    WHERE s.Gruppo=p.Gruppo AND NOT EXISTS (
        SELECT *
        FROM PrenotazioniGruppi2020 p1
        WHERE s.Matricola=p1.Matricola AND s.Ateneo=p1.Ateneo
            AND s.Gruppo=p.Gruppo
    )
);
```

Esame di Teoria (rispondere su fogli separati rispetto alla parte di SQL)

Domanda 1 (9 punti).

Con riferimento alla base di dati “DoveStudio”:

- A. (4 punti) Esprimere in Algebra Relazionale l'interrogazione
Elencare le studentesse/gli studenti di UNITO che hanno effettuato solo prenotazioni per il proprio gruppo di studio.
- B. (5 punti) Esprimere, nel calcolo relazionale su tuple con dichiarazione di range, la seguente domanda:
Elencare le studentesse/gli studenti di POLITO che hanno effettuato tutte le prenotazioni per il proprio gruppo di studio.

Soluzione 1.

A. Una possibile soluzione in algebra relazionale è la seguente:

$$\pi_{Matricola, Ateneo}(\sigma_{Ateneo='UNITO'}(prenotazione)) \\ - (\pi_{Matricola, Ateneo}(\sigma_{Gruppo \text{ IS NULL}}(prenotazione)))$$

B. Una possibile soluzione è la seguente:

$$\{p.Ateneo, p.Matricola \mid p(PRENOTAZIONE) \mid p.Ateneo='POLITO' \wedge p.Gruppo \text{ IS NOT NULL} \wedge \\ \neg \exists p'(prenotazione) ((p.Matricola \neq p'.Matricola \vee p.Ateneo \neq p'.Ateneo) \wedge p.Gruppo=p'.Gruppo)\}$$

Domanda 2 (9 punti).

Si consideri lo schema relazionale Catalogo(NomeDitta, IndirizzoDitta, Prodotto, Prezzo) con dipendenze funzionali $F=\{ \text{NomeDitta} \rightarrow \text{IndirizzoDitta}; \text{NomeDitta}, \text{Prodotto} \rightarrow \text{Prezzo}, \text{IndirizzoDitta} \}$

- A. Riportare la definizione di chiusura di un insieme di attributi.
B. Riportare la definizione di insiemi equivalenti di dipendenze funzionali.
C. Dire se la relazione Catalogo è in 3FN motivando la risposta. Se non è in 3FN proporre una decomposizione.

Soluzione 2.

Per i punti A e B, si vedano gli appunti/testo/slide.

C. L'unica chiave è $K=\{\text{NomeDitta}, \text{Prodotto}\}$ (è superchiave perché $K^+=\{\text{NomeDitta}, \text{IndirizzoDitta}, \text{Prodotto}, \text{Prezzo}\}$ ed è minimale perché NomeDitta e Prodotto singolarmente non sono superchiavi). La relazione quindi non è in 3NF: infatti la dipendenza funzionale $\text{NomeDitta} \rightarrow \text{IndirizzoDitta}$ non è né riflessiva, né di tipo superchiave e IndirizzoDitta non è attributo primo.

Ricaviamo una copertura minimale.

La forma canonica è

$$F'=\{\text{NomeDitta} \rightarrow \text{IndirizzoDitta}; \text{NomeDitta}, \text{Prodotto} \rightarrow \text{Prezzo}; \text{NomeDitta}, \text{Prodotto} \rightarrow \text{IndirizzoDitta}\}.$$

Non ci sono attributi estranei nella prima d.f. perché a sinistra c'è un solo attributo, nella seconda d.f. perché Prezzo non è ricavabile senza l'attributo NomeDitta o senza l'attributo Prodotto. Nella terza d.f. Prodotto è attributo estraneo perché $\text{IndirizzoDitta} \in \{\text{NomeDitta}\}^+$. Quindi

$$F''=\{\text{NomeDitta} \rightarrow \text{IndirizzoDitta}; \text{NomeDitta}, \text{Prodotto} \rightarrow \text{Prezzo}; \text{NomeDitta} \rightarrow \text{IndirizzoDitta}\}.$$

La terza d.f. è ovviamente ridondante perché ripetuta. Le altre d.f. non sono ridondanti. Quindi la copertura minimale è $F''' = \{\text{NomeDitta} \rightarrow \text{IndirizzoDitta}; \text{NomeDitta}, \text{Prodotto} \rightarrow \text{Prezzo}\}$.

Una decomposizione possibile derivante da F''' è

$R1(\underline{\text{NomeDitta}}, \text{IndirizzoDitta})$ e $R2(\underline{\text{NomeDitta}}, \text{Prodotto}, \text{Prezzo})$.

Domanda 3 (7 punti).

A. Spiegare in modo succinto ma preciso i soli passi dell'algoritmo di ottimizzazione logica che coinvolgono le selezioni e quali loro proprietà li rendono possibili.

Soluzione 3.

A. Vedere il libro di testo o le slide.

Domanda 4 (8 punti).

Si consideri un file di log L con il seguente contenuto in seguito ad un crash:

```
<T1, START>;
<T2, START>;
<T1, BS(t1[A], 5), AS(t1[A], 10)>;
<T2, BS(t2[B], 3), AS(t2[B], 5)>;
<T3, START>;
<T2, COMMIT>;
-- checkpoint --;
<T3, BS(t3[C], 3), AS(t3[C], 5)>;
<T1, COMMIT>
```

crash!

A. Indicare la differenza tra ripresa a freddo (dump/restore) e ripresa a caldo (ripristino).

B. Descrivere l'algoritmo di ripristino e mostrarne l'esecuzione sul log L

Soluzione 4.

A. Si veda il libro di testo o le slide.

B. L'algoritmo di ripristino recupera la lista di transazioni LC in commit dopo l'ultimo checkpoint e prima del crash (solo T1) e la lista delle transazioni LA attive al momento del crash (quindi solo T3). Viene effettuato quindi l'UNDO di T3: viene ripristinato a '3' il valore dell'attributo C della tupla t3 (si effettua contestualmente anche un FORCE della pagina). Successivamente, il gestore del ripristino esegue il REDO di T1: viene reso persistente il valore 10 dell'attributo A della tupla t1 (con conseguente FORCE della pagina).