

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 “**Presenze**” per gestire le informazioni sulla frequenza delle lezioni da parte di studenti e studentesse. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

STUDENTE(Matricola, Cognome, Nome, DataNascita, TipoStudente*)

INSEGNAMENTO(Codice, Titolo, CFU)

DOCENTE(Matricola, Cognome, Nome, Qualifica)

LEZIONE(Data, Ora, Aula, Insegnamento, Docente)

PARTECIPAZIONE(Data, Ora, Aula, Studente, Gradimento*)

LEZIONE(Insegnamento) referencia INSEGNAMENTO(Codice),

LEZIONE(Docente) referencia DOCENTE(Matricola),

PARTECIPAZIONE(Data, Ora, Aula) referencia LEZIONE(Data, Ora, Aula),

PARTECIPAZIONE(Studente) referencia STUDENTE(Matricola).

“TipoStudente” assume il valore ‘Occupatə’ se lo studente o la studentessa è lavoratore/lavoratrice, altrimenti assume il valore ‘Non occupatə’. Si noti che il simbolo ə (schwa) è si usa per indicare il maschile/femminile.

“Qualifica” può assumere i seguenti valori ‘R’ (Ricercatore/ricercatrice), ‘PA’ (Professore/Professoressa Associatə), ‘PO’ (Professore/Professoressa ordinariə), ‘Contratto’ (docente a contratto).

Gradimento è un numero da 0 a 30 e rappresenta un voto che lo studente/la studentessa ha assegnato alla lezione; la sufficienza è almeno 18.

Tutti gli altri attributi sono autoesplicativi. Gli attributi con l’asterisco possono assumere valore NULL.

Con riferimento alla base di dati proposta, esprimere in SQL le seguenti interrogazioni (commentare ogni soluzione proposta spiegando le varie parti della query).

Domanda 1 (bassa complessità).

Elencare codice e titolo degli insegnamenti con lezioni giudicate insufficienti da studenti e studentesse non occupatə o di tipo sconosciuto. Ordinare i risultati per numero decrescente di CFU e, successivamente, per titolo dell’insegnamento crescente.

Soluzione 1.

```
SELECT DISTINCT i.Codice, i.Titolo
FROM studente s JOIN partecipazione p ON (p.Studente=s.Matricola)
      JOIN lezione l ON (l.Data=p.Data AND l.Ora=p.Ora AND l.Aula=p.Aula)
      JOIN insegnamento i ON (l.Insegnamento=i.Codice)
WHERE Gradimento<18 AND (TipoStudente = 'Non occupatə' OR TipoStudente IS NULL)
ORDER BY i.CFU DESC, i.Titolo ASC;
```

Domanda 2 (media complessità).

Per ogni docente non professore o professoressa, elencare i titoli degli insegnamenti aventi almeno una lezione con più di 50 partecipanti non occupatə.

Soluzione 2.

```
SELECT DISTINCT d.Matricola, i.Titolo
FROM studente s JOIN partecipazione p ON (p.Studente=s.Matricola)
      JOIN lezione l ON (l.Data=p.Data AND l.Ora=p.Ora AND l.Aula=p.Aula)
      JOIN insegnamento i ON (l.Insegnamento=i.Codice)
      JOIN docente d ON (d.Matricola=l.Docente)
WHERE s.TipoStudente='Non occupatø' AND d.Qualifica<>'PO' AND d.Qualifica<>'PA'
GROUP BY d.Matricola, i.Codice, p.Data, p.Ora, p.Aula
HAVING COUNT(*) > 50;
```

Domanda 3 (alta complessità).

Per ogni professoressa o professore ordinario, mostrare il numero di insegnamenti in cui il numero di partecipanti che assegnano un gradimento almeno sufficiente supera sempre il numero di partecipanti che assegnano un gradimento insufficiente.

Soluzione 3.

```
WITH NumPartecipantiSuff AS (
  SELECT Data, Ora, Aula, COUNT(*) AS NumPartSuff
  FROM partecipazione
  WHERE Gradimento>=18
  GROUP BY Data, Ora, Aula
), NumPartecipantiInsuff AS (
  SELECT Data, Ora, Aula, COUNT(*) AS NumPartInsuff
  FROM partecipazione
  WHERE Gradimento<18
  GROUP BY Data, Ora, Aula
)
SELECT d.Matricola, d.Cognome, d.Nome, COUNT(DISTINCT Insegnamento)
FROM docente d JOIN lezione l ON (l.Docente=d.Matricola)
WHERE d.Qualifica='PO' AND l.Insegnamento NOT IN (
  SELECT Insegnamento
  FROM lezione l JOIN NumPartecipantiSuff nps ON
    (npp.Data=l.Data AND npp.Ora=l.Ora AND npp.Aula=l.Aula)
  JOIN NumPartecipantiInsuff npi ON
    (npd.Data=l.Data AND npd.Ora=l.Ora AND npd.Aula=l.Aula)
  WHERE NumPartSuff<=NumPartInsuff)
GROUP BY d.Matricola;
```

Esame di Teoria

Domanda 1 (10 punti).

Con riferimento alla base di dati “**Presenze**” esprimere le seguenti interrogazioni, commentadole:

A. (5 punti) in algebra relazionale:

Elencare le matricole degli studenti/delle studentesse che hanno assegnato un gradimento almeno sufficiente a tutte le lezioni del corso di Basi di Dati.

B. (5 punti) in calcolo relazionale su tuple con dichiarazione di range:

Elencare Titolo e numero di CFU degli insegnamenti tenuti da almeno due docenti.

Soluzione 1.

A. Una possibile soluzione è la seguente:

$$\pi_{Matricola, Data, Ora, Aula} \left(\sigma_{Gradimento \geq 18} (partecipazione \bowtie_{\alpha} lezione \bowtie_{\beta} insegnamento) \right) \\ \div \pi_{Data, Ora, Aula} (\sigma_{Insegnamento = 'BD'} (lezione))$$

dove

$$\alpha = PARTECIPAZIONE.Data = LEZIONE.Data \wedge PARTECIPAZIONE.Ora \\ = LEZIONE.Ora \wedge PARTECIPAZIONE.Aula = LEZIONE.Aula$$

$$\beta = LEZIONE.Insegnamento = INSEGNAMENTO.Codice$$

B. Una possibile soluzione è la seguente:

$$\{i.Titolo, i.CFU \mid i(INSEGNAMENTO) \mid \exists l(LEZIONE)(l.Insegnamento=i.Codice \wedge \\ \exists l'(LEZIONE)(l'.Insegnamento=i.Codice \wedge l.Docente \neq l'.Docente))\}$$

Domanda 2 (10 punti).

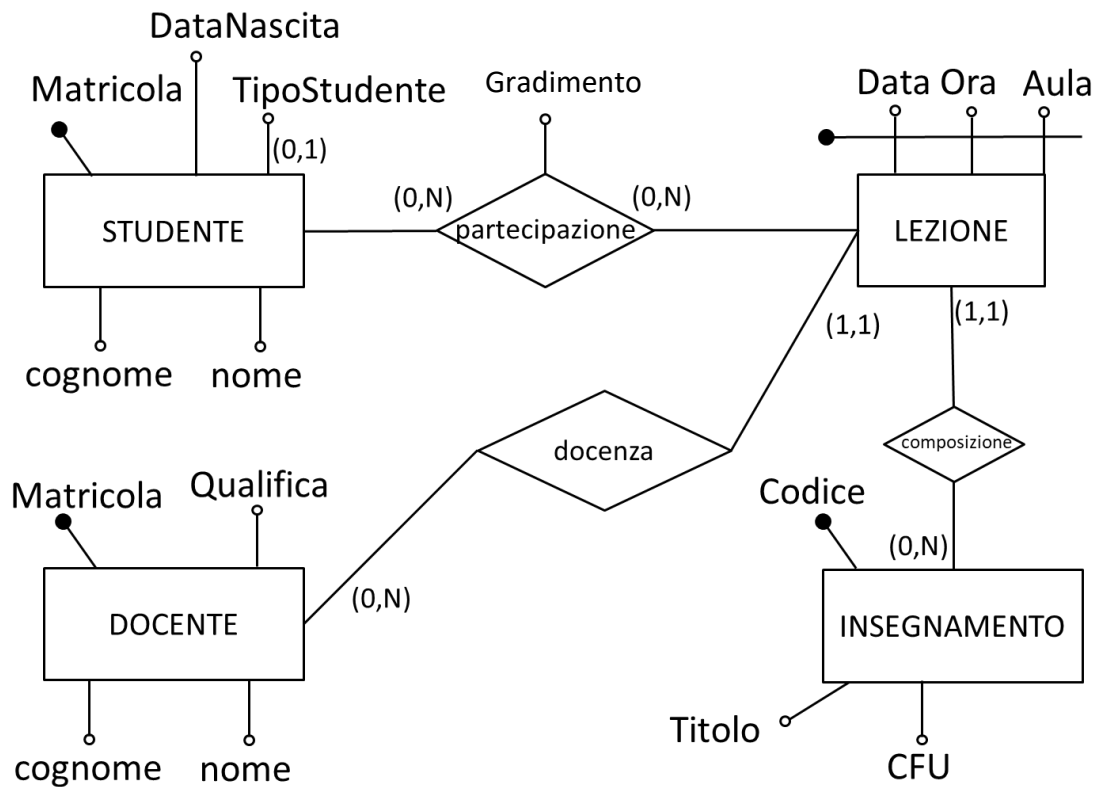
A. Indicare la condizione necessaria e sufficiente affinché una decomposizione di una relazione $R(A)$ in $R_1(A_1)$ e $R_2(A_2)$ sia senza perdita di informazione.

B. Disegnare uno schema Entity-Relationship che potrebbe aver generato lo schema logico della base di dati “Presenze”.

Soluzione 2.

A. Si vedano gli appunti/testo/slide.

B. Una possibile soluzione è la seguente:



Domanda 3 (7 punti).

Con riferimento alla base di dati “**Presenze**” e i seguenti dati quantitativi:

$CARD(DOCENTE) = 100$

$CARD(LEZIONE) = 5000$

$VAL(Insegnamento, LEZIONE) = 50$

$VAL(Aula, LEZIONE) = 10$

disegnare gli alberi sintattici prima e dopo l'ottimizzazione logica e calcolare il numero di tuple “mosse” prima e dopo l'ottimizzazione logica della seguente query:

$\sigma_{(Qualifica \neq 'PO' \wedge Insegnamento='BD' \wedge Aula='D')} (lezione \bowtie_{Docente=Matricola} docente)$

Soluzione 3.

La query ottimizzata dividendo la selezione e portandola verso le foglie è:

$(\sigma_{Qualifica \neq 'PO'} (docente))$

$\bowtie_{Docente=Matricola}$

$(\sigma_{Insegnamento='BD' \wedge Aula='D'} (lezione))$

Prima dell'ottimizzazione:

- Costo $r_1 = (lezione \bowtie_{Docente=Matricola} docente)$: $CARD(LEZIONE) \cdot CARD(DOCENTE) = 5\,000 \cdot 100 = 5 \cdot 10^5$.
- Cardinalità $|r_1| = CARD(LEZIONE) = 5\,000$ (equijoin attraverso la chiave esterna)
- Costo della selezione: $|r_1| = 5\,000$
- Costo totale = $500\,000 + 5\,000 \sim 5 \cdot 10^5$.

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{Qualifica \neq 'PO'} (docente) = CARD(DOCENTE) = 100$

- Costo $\sigma_2 = \sigma_{(\text{Insegnamento}='BD' \wedge \text{Aula}='D')(\text{lezione})} = \text{CARD}(\text{LEZIONE}) = 5\,000$
- Tuple prodotte dalla selezione $|\sigma_1| = (1 - 1/\text{VAL}(\text{Qualifica}, \text{DOCENTE})) \cdot \text{CARD}(\text{DOCENTE}) = (1 - 1/5) \cdot 100 = 80$
- Tuple prodotte dalla selezione $|\sigma_2| = f_{\text{Insegnamento}='BD'} \cdot f_{\text{Aula}='D'} \cdot \text{CARD}(\text{LEZIONE}) = 1/\text{VAL}(\text{Insegnamento}, \text{LEZIONE}) \cdot 1/\text{VAL}(\text{Aula}, \text{LEZIONE}) \cdot \text{CARD}(\text{LEZIONE}) = (1/50) \cdot (1/10) \cdot 5\,000 = 10$
- Costo join $r = \sigma_1 \bowtie_{\text{Docente}=\text{Matricola}} \sigma_2 = 80 \cdot 10 = 800$.
- Costo totale $= 100 + 5\,000 + 800 = 5\,900 \sim 6 \cdot 10^3$.

Domanda 4 (6 punti).

Dire, giustificando la risposta, se la seguente storia è compatibile con il protocollo 2PL (lock a due fasi):

$S = r1(x), r2(x), w1(z), w2(y), w3(x), r1(y), w2(z)$

Soluzione 4.

Non è compatibile perché la transazione T1 sarebbe costretta a riacquisire il lock su y dopo aver rilasciato quello su x per permettere a T3 di acquisire a sua volta il lock su x. Inoltre, T1 non può acquisire prima il lock su y, perché dovrebbe poi lasciarlo a T2.