

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 “**NerdFlix**” per gestire una piattaforma di streaming online di serie TV. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

SERIETV(NomeSerie, Produttore, Nazione, Genere)

EPISODIO(NomeSerie, Stagione, Episodio, TitoloEpisodio, Anno)

AUDIO(NomeSerie, Stagione, Episodio, Lingua, AdattatoreDialoghi*, Originale)

SOTTOTITOLI(NomeSerie, Stagione, Episodio, Lingua)

Vincoli di integrità referenziale:

EPISODIO(NomeSerie) referencia SERIETV(NomeSerie),

AUDIO(NomeSerie, Stagione, Episodio) referencia EPISODIO(NomeSerie, Stagione, Episodio),

SOTTOTITOLI(NomeSerie, Stagione, Episodio) referencia EPISODIO(NomeSerie, Stagione, Episodio).

“Genere” può assumere i valori “Animazione”, “Dramma”, “Fantascienza” e “Commedia”. “Originale” può assumere i valori True (se l’audio è quello della lingua originale) e False (se si tratta di un doppiaggio). I rimanenti attributi sono autoesplicativi. AdattatoreDialoghi può essere NULL.

A titolo di esempio, la seguente tupla della relazione EPISODIO

<'Stranger Things', 3, 2, 'Capitolo due - Incubi', '2019'>

si riferisce al secondo episodio della terza stagione della serie TV “Stranger Things”. **Non è detto che tutti gli episodi (o che tutte le stagioni) di una serie siano presenti nella base di dati.**

Con riferimento alla base di dati “NerdFlix” esprimere in SQL le seguenti interrogazioni.

Domanda 1 (bassa complessità).

Trovare le serie TV tedesche di genere “Fantascienza” con almeno un episodio sottotitolato in inglese. Mostrare le serie TV in ordine alfabetico crescente dell’attributo Nazione.

Soluzione 1.

```
SELECT DISTINCT se.NomeSerie
FROM serietv se JOIN sottotitoli so ON (se.NomeSerie=so.NomeSerie)
WHERE se.Genere='Fantascienza' AND se.Nazione='Germania' AND so.Lingua='Inglese'
ORDER BY se.Nazione;
```

Domanda 2 (media complessità).

Mostrare le serie TV di cui sono stati prodotte almeno 5 stagioni aventi ciascuna almeno un episodio con audio originale in spagnolo.

Soluzione 2.

```
SELECT e.NomeSerie
FROM episodio e JOIN audio a ON (e.NomeSerie=a.NomeSerie
                                AND e.Stagione=a.Stagione AND e.Episodio=a.Episodio)
WHERE a.Lingua='Spagnolo' AND a.Originale='true'
GROUP BY e.NomeSerie
HAVING COUNT(DISTINCT e.Stagione)>=5;
```

Domanda 3 (alta complessità).

Tra le serie TV che **non** hanno il numero massimo di stagioni presenti nel database, trovare quella (o quelle) con più episodi prodotti (non utilizzare costrutti come LIMIT, ROWNUM, TOP).

Soluzione 3.

```
WITH NumStagioniSerieTV AS (
    SELECT e.NomeSerie, COUNT(distinct e.Stagione) NumStagioni
    FROM episodio e
    GROUP BY e.NomeSerie),
NumEpisodiSerieTVEscluseMaxNumStagioni AS (
    SELECT e.NomeSerie, COUNT(*) NumEpisodiSTV
    FROM episodio e
    WHERE e.NomeSerie IN (SELECT NomeSerie
                          FROM NumStagioniSerieTV
                          WHERE NumStagioni < ANY (SELECT NumStagioni
                                                    FROM NumStagioniSerieTV))

    GROUP BY e.NomeSerie)
SELECT NomeSerie
FROM NumEpisodiSerieTVEscluseMaxNumStagioni
WHERE NumEpisodiSTV = (SELECT MAX(NumEpisodiSTV)
                      FROM NumEpisodiSerieTVEscluseMaxNumStagioni);
```

Esame di Teoria

Domanda 1 (8 punti).

Con riferimento alla base di dati “NerdFlix”:

- A. Esprimere in Algebra Relazionale l'interrogazione
Elencare le serie TV sottotitolate^(*) in tutte le lingue^().**
- B. Esprimere, nel calcolo dei predicati su tuple con dichiarazione di range, la seguente domanda:
Elencare nome e produttore delle serie TV drammatiche per cui esistono sia i sottotitoli^(*) in spagnolo che in inglese^().**

^(*) è sufficiente che esistano i sottotitoli di un solo episodio della serie

^(**) i sottotitoli possono riguardare anche episodi diversi della stessa serie

Soluzione 1.

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

$$\pi_{NomeSerie, Lingua}(sottotitoli) \div \pi_{Lingua}(sottotitoli)$$

B. Una possibile soluzione è la seguente:

$$\{s.NomeSerie, s.Produttore \mid s(SERIETV) \mid s.Genere='Dramma' \wedge \\ \exists st(SOTTOTITOLI)(st.NomeSerie=s.NomeSerie \wedge st.Lingua='Spagnolo') \wedge \\ \exists st'(SOTTOTITOLI)(st'.NomeSerie=s.NomeSerie \wedge st'.Lingua='Inglese')\}$$

Domanda 2 (9 punti).

- A. Riportare la definizione di forma normale di Boyce-Codd (BCNF).
- B. Esprimere la condizione necessaria e sufficiente perché una decomposizione sia senza perdita di informazione.
- C. Dati $R(\text{Matricola}, \text{CorsoLaurea}, \text{CodCorso}, \text{Crediti})$ e $F = \{\text{Matricola} \rightarrow \text{CorsoLaurea}, \text{CodCorso} \rightarrow \text{Crediti}\}$ dire se R è in 3FN motivando la risposta e se non lo è decomporla mostrando tutti i passaggi.

Soluzione 2.

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

- C. Per prima cosa occorre individuare la chiave della relazione R . Si nota immediatamente che una possibile superchiave è $\{\text{Matricola}, \text{CodCorso}\}$. Essa è anche l'unica chiave candidata.

Nessuna delle due d.f. dell'insieme F è quindi in 3FN (Matricola e CodCorso da soli non sono chiave e i due attributi CodCorso e Crediti non sono parte della chiave).

Si cerca quindi l'insieme di copertura minimale di F , che in questo caso coincide con F stesso (non ci sono né attributi estranei né d.f. ridondanti).

Si procede quindi alla normalizzazione, ottenendo le due relazioni:

$R1(\underline{\text{Matricola}}, \text{CorsoLaurea})$

$R2(\underline{\text{CodCorso}}, \text{Crediti})$

a cui si deve aggiungere la relazione contenente la chiave:

$R3(\underline{\text{Matricola}}, \underline{\text{CodCorso}})$

Domanda 3 (8 punti).

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

$CARD(SERIETV) = 1000$

$CARD(EPISODIO) = 20000$

$VAL(Anno, EPISODIO) = 50$

$VAL(Stagione, EPISODIO) = 20$

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:

$\sigma_{Anno='2021' \wedge Genere='Dramma' \wedge Stagione='1'}(episodio \bowtie_{episodio.NomeSerie=serietv.NomeSerie} serietv)$

Soluzione 3.

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

$(\sigma_{Anno='2021' \wedge Stagione='1'}(episodio)) \bowtie_{episodio.NomeSerie=serietv.NomeSerie} (\sigma_{Genere='Dramma'}(serietv))$

Prima dell'ottimizzazione:

- Costo $r_1 = (episodio \bowtie_{episodio.NomeSerie=serietv.NomeSerie} serietv)$: $1000 \times 20000 = 20\,000\,000$.
- Cardinalità $|r_1| = CARD(EPISODIO) = 20000$ (equijoin attraverso la chiave esterna)
- Costo della selezione: $|r_1|$
- Costo totale = $20\,000\,000 + 20\,000 \sim 2 \times 10^7$.

Dopo l'ottimizzazione:

- Costo $\sigma_1 = \sigma_{Anno='2021' \wedge Stagione='1'}(episodio) = CARD(episodio) = 20000$
- Costo $\sigma_2 = \sigma_{Genere='Dramma'}(serietv) = 1000$
- Tuple prodotte dalla selezione $|\sigma_1| = 1/VAL(Anno, EPISODIO) \times 1/VAL(Stagione, EPISODIO) \times CARD(EPISODIO) = 1/50 \times 1/20 \times 20000 = 20$
- Tuple prodotte dalla selezione $|\sigma_2| = 1/VAL(Genere, SERIETV) \times CARD(SERIETV) = 1/4 \times 1000 = 250$
- Costo join $r = \sigma_1 \bowtie_{episodio.NomeSerie=serietv.NomeSerie} \sigma_2 = 20 \times 250 = 5000$.
- Costo totale = $20000 + 1000 + 5000 = 26000 = 2,6 \times 10^4$.

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>;
<T4, START>;
<T4, (t4[D], 8), AS(t4[D], 16)>;
-- checkpoint --;
<T4, COMMIT>;
<T3, BS(t3[C], 3), AS(t3[C], 5)>;
<T3, ABORT>
```

crash!

- A. Indicare il contenuto del record di checkpoint.
- B. Mostrare i passi dell'esecuzione dell'algoritmo di ripristino sul log L.

Soluzione 4.

- A. Il record di checkpoint conterrà la lista delle transazioni non terminate (quindi T1, T3 e T4) con i relativi puntatori ai rispettivi $\langle T_i, \text{START} \rangle$. Contiene inoltre l'OK riguardante la buona riuscita delle operazioni previste dal checkpoint, quindi: $[(T1, p1); (T3, p3); (T4, p4); OK]$.
- B. Si noti che le liste LA e LC sono ricostruite a partire dal record di checkpoint. La lista delle transazioni attive al momento del crash è $LA = \{T1\}$ (T3 è in abort). La lista delle transazioni in commit prima del crash è $LC = \{T4\}$.

Viene effettuato quindi l'UNDO di T1: viene ripristinato a '5' il valore dell'attributo A della tupla t1 (si effettua contestualmente anche un FORCE della pagina). Successivamente, si effettua il REDO di T4: viene confermato il valore '16' dell'attributo D della tupla t4.