

Si consideri la seguente base di dati con le relazioni:

- **Programma** (Codice, NomeProgramma, Rete, Tipologia) che, ad ogni codice di programma televisivo, associa il nome del programma, la rete che lo trasmette e la tipologia.
- **Artista** (CProgramma, NomeArtista, compenso) che memorizza gli artisti che hanno partecipato ai diversi programmi, con i rispettivi compensi.

con chiave esterna Artisti.CProgramma → Programmi.Codice.

Si noti che ci possono essere due programmi con lo stesso nome ma codice diverso.

- B. Scrivere una query in Standard SQL che, per ogni codice programma, restituisce il nome del programma e la somma dei compensi per gli artisti del programma. (2.5 punti).

STUDENTE(Matricola, Nominativo, DataNascita, CorsoLaurea)

CORSO(Id-Corso, NomeCorso, Crediti, NomeDocente)

LIBRETTO(Id-Corso, Matricola, Voto, Lode)

- A. Restituire le matricole degli studenti iscritti a Informatica che hanno preso almeno due lodi (2 punti).²

$S_1 = \sigma_{\text{CorsoLaurea} = \text{"Informatica"}} \text{STUDENTE} \bowtie \text{LIBRETTO}$

$S_2 = S_1$

$\pi_{S_1.Matricola} S_1 \bowtie S_1 \text{ ID-Corso} <> S_2 \text{ ID-Corso}$
 AND
 $S_1.Matricola = S_2.Matricola$
 AND
 $S_1.Lode = \text{TRUE} \text{ AND } S_2.Lode = \text{TRUE}$

$C1 =$

$\pi_{\text{Libretto.Id-Corso}, \text{Libretto.Matricola}}$
 $(\sigma_{\text{CorsoLaurea} = \text{"Informatica"}}(\text{STUDENTE}) \bowtie$
 $\sigma_{\text{Lode} = \text{TRUE}}(\text{LIBRETTO}))$

$C2 = C1$

$\pi_{C1.Matricola}$
 $(C1 \bowtie_{C1.Matricola = C2.Matricola \text{ AND } C1.Id-Corso <> C2.Id-Corso} C2)$

SELECT SUM(compenso)

AS TOTALE_ARTISTI,

CPROGRAMMA,

NAMEPROGRAMMA

FROM ARTISTA A

INNER JOIN PROGRAMMA

ON P.CODICE =

A.CPROGRAMMA

GROUP BY NAMEPROGRAMMA

CPROGRAMMA

STUDENTE(Matricola, Nominativo, DataNascita, CorsoLaurea)

CORSO(Id-Corso, NomeCorso, Crediti, NomeDocente)

LIBRETTO(Id-Corso, Matricola, Voto, Lode)

B. Nel riquadro, scrivere una query in Standard SQL che restituisce la matricola e il nominativo degli studenti che hanno sostenuto esami per meno di 120 crediti. (2.5 punti).

```
SELECT MATRICOLA, NOMINATIVO
FROM STUDENTE S, LIBRETTO L, CORSO C
WHERE S.MATRICOLA = L.MATRICOLA
AND C.ID-CORSO = L.ID-CORSO
GROUP BY MATRICOLA, NOMINATIVO
HAVING SUM(CREDITI) < 120;
```

```
SELECT MATRICOLA, NOMINATIVO
FROM STUDENTE
WHERE MATRICOLA IN
    (SELECT MATRICOLA
     FROM LIBRETTO L, CORSO C
     WHERE L.ID-CORSO=C.ID-CORSO
     GROUP BY MATRICOLA
     HAVING SUM(CREDITI)<120)
```

STUDENTE(Matricola, Nominativo, DataNascita, CorsoLaurea)

CORSO(Id-Corso, NomeCorso, Crediti, NomeDocente)

LIBRETTO(Id-Corso, Matricola, Voto, Lode)

```
CREATE VIEW MODA_VOTO
AS SELECT ID-CORSO,
AVG(VOTO) AS MODA
```

A. Restituire le matricole degli studenti iscritti a Informatica che hanno preso almeno due lodi (2 punti).²

C. Nel riquadro, scrivere una query in Standard SQL che restituisce, il corso con la media voto più alta, tra tutti i corsi con almeno 6 crediti (2.5 punti).

```
SELECT CORSO FROM
MODA_VOTO WHERE
```

```
MODA = (SELECT MAX(MODA)
FROM MODA_VOTO)
```

```
FROM CORSO C,
LIBRETTO L
WHERE
C.ID-CORSO =
L.ID-CORSO
AND
CREDITI ≥ 6
GROUP BY ID-CORSO
```

Esercizio 4: Ripresa a freddo (6 punti)

Si consideri la seguente porzione di log fino al guasto hardware: DUMP, B(T5), B(T6), U(T5, O1, B1, A1), CK(T5, T6), B(T7), U(T7, O6, B6, A6), U(T6, O3, B7, A7), B(T8), I(T8, O5, A5), C(T8), A(T5). Descrivere i passi e le operazioni necessarie per la relativa ripresa a freddo.

Si ripristina il sistema di gestione di basi di dati da un sistema di massa terziario (nastri magnetici, Blu-ray ottici, ecc.)

① DOPO DUMP → ROLAND

$$O1 = B1, O6 = B6, O3 = B7, O5 = A5$$

② UNDO → (CLEAN) $O6 = B6, O3 = B7 = O1 = B1$

③ UNDO → (CLEAN) $O5 = A5$

Si consideri una relazione $R(X, Y)$ con un indice B-Tree sull'attributo X.

Assumendo che ogni nodo del B-Tree può contenere 100 entry e la relazione R contenga 1000 tuple, quale è la profondità del B-Tree?

1. Profondità 1
2. Profondità 2
3. Profondità 3
4. Profondità 4

$$10^1 = 10 < 100$$

$$10^2 = 100 \leq 100$$

RIGHT

$$d \leq \log_{100} 1000$$

$$d \leq \log_{100} 1000$$

$$100^? = 1000 ?$$

Si consideri una relazione $R(A, B, C)$ con un indice B-Tree sull'attributo A. Assumendo che ogni nodo del B-Tree può contenere 50 entry e la relazione R contiene 2000 tuple, quale è la profondità del B-Tree?

1. Profondità 1
2. Profondità 2
3. Profondità 3
4. Profondità 4

$$\log_{50} 2000$$

$$50^? = 2000 ?$$

$$50^1 = 50 < 2000$$

$$50^2 = 2550 > 2000 \rightarrow \text{proponiamo } 2$$

Si consideri la seguente relazione $R(A,B,C,D)$ con l'insieme delle dipendenze funzionali $\{A \rightarrow B, B \rightarrow C, A \rightarrow D\}$. Quale delle seguenti affermazioni è vera?

1. La relazione è in 3NF ed anche è in BCNF.
2. La relazione è in 3NF ma non è in BCNF.
3. La relazione non è in 3NF e non è in BCNF.
4. La relazione non è in 3NF ma è in BCNF.

$$A^+ = \{A, B, C, D\} = \text{chiave}$$

$$B^+ = \{B, C\} \quad 3FN \quad (X^+ \rightarrow Y) \text{ DEF A SO. SUPERCURANS}$$

$$\underline{B^+ \subseteq A^+} \rightarrow \text{NO 3FN}$$

$$R_1 = (\underline{A}, B, C, D)$$

$$R_1 = (\underline{A}, C, D)$$

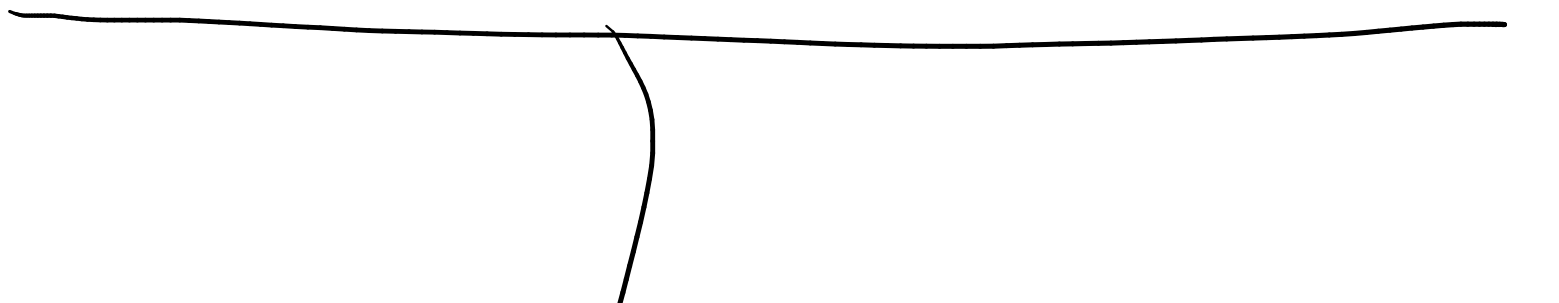
$$R_2 = (\underline{B}, C)$$

$$\underline{\text{BCNF}} \rightarrow B?$$

$$\begin{aligned} B^+ &= \{B, A\} \\ C^+ &= \{C, D\} \\ A^+ &= \{A, B, C\} \\ AB^+ &= \{A, B, C, D\} \end{aligned}$$

$$\rightarrow \left[\begin{array}{l} B \rightarrow A, C \rightarrow D, C \rightarrow B \\ AB \rightarrow C \end{array} \right]$$

$[BCNF]$? \rightarrow tutti siano!



Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati per la gestione dei campionati di calcio di Serie A, dove una stagione è rappresentata dalla stringa Anno-Inizio-AnnoFine (per es. "2023-2024"):

- Partita(SquadraCasa, SquadraTrasferta, Stagione, GoalCasa, GoalTrasferta)
- Luogo(Squadra, Città)

A. Nel riquadro, scrivere una query in Algebra Relazionale che restituisca la squadra che ha fatto più goal alla Roma, quando giocava in casa, nella stagione 2023-2024 (2.5 punti).²

$P_1 = \text{PARTITA}$
 $P_2 = P_1$
 $\pi_{P_1.SQUADRA}$
 $P_1 \rightarrow (\sigma_{\text{STAGIONE} = "2023-2024"})$
 $(\pi_{P_1.SQUADRA} \cap P_2)$
 $\text{AND } SQUADRA \text{ TRASFERITA} = "ROMA"$
 $(P_1 \bowtie P_2)$
 $P_1.GOALCASA < P_2.GOALCASA$
 $\text{AND } P_1.SQUADRA \text{ CASA}$
 $\hookrightarrow P_2.SQUADRA \text{ CASA}$

B. Scrivere una query in SQL Standard che restituisce, per ogni squadra **s** della stagione 2023-2024, la coppia (**s,g**) per indicare che la squadra **s** ha realizzato **g** goal in casa nella stagione. Ordinare poi le coppie per valori di **g** decrescente (2 punti).

- Partita(SquadraCasa, SquadraTrasferta, Stagione, GoalCasa, GoalTrasferta)
- Luogo(Squadra, Città)

SELECT SQUADRA CASA AS S, SUM(GOALCASA) AS G
 FROM PARTITA
 WHERE STAGIONE = "2023-2024"
 GROUP BY S
 ORDER BY G DESC;

- Partita(SquadraCasa, SquadraTrasferta, Stagione, GoalCasa, GoalTrasferta)
- Luogo(Squadra, Citta)

C. Scrivere una query in SQL Standard che restituisce, per ogni squadra **s** che ha giocato almeno 10 derby come squadra di casa (cioè partite tra squadre della stessa città), il numero medio di goal segnati da **s** nei derby in cui **s** era la squadra di casa (2.5 punti).

```

CREATING VIEW DERBY_S AS
SELECT SUM (GOALCASA) AS NUM- GOAL, SQUADRA CASA
FROM PARTITA P, LUOGO L1, LUOGO L2      COUNT (*) AS
WHERE P. SQUADRA CASA = L. SQUADRA      N- DERBY
AND L1. CITTÀ = L2. CITTÀ
AND L1. SQUADRA <> L2. SQUADRA
GROUP BY SQUADRA CASA
HAVING COUNT (*) >= 10;

```

```

SELECT NUM- GOAL
FROM DERBY_S
WHERE NUM- GOAL = (SELECT AVG (NUM- GOAL)
FROM DERBY_S);

```