

Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati per la registrazione dei voli:

- AEROPORTO(Città, Nazione, NumPiste)
- VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

SOLN

A. Nel riquadro, scrivere una query in Algebra Relazionale che restituisce la/e città da cui partono solo voli internazionali (2 punti).²

$$\pi_{CITTÀ} AEROPORTO = A_1$$

$$A_2 = A_1$$

$$\pi_{CITTÀPART} VOLO$$

$$\pi_{CITTÀPART} A_1 \bowtie A_1 - NAZIONI \rightarrow A_2 - NAZIONI$$

$$A_2$$

$$A_1 = AEROPORTO \bowtie VOLO$$

$$A_2 = A_1$$

$$A_3 = \pi_{CITTÀ} A_1 \bowtie A_1 - NAZIONI$$

$$A_2 - NAZIONI$$

$$AND$$

$$A_1 - NAZIONI = A_2 - NAZIONI$$

$$\pi_{A_1 - CITTÀ} A_3$$

Nel riquadro, scrivere una query SQL che restituisce il tipo di aereo più utilizzato in arrivo nell'aeroporto della città di Roma il giovedì (2.5 punti).

```
CREATE VIEW Best_aereo AS
SELECT TipoAereo, COUNT(*) AS N_utilizzi
FROM Volo V, Aeroporto A
WHERE V.CittàArr = A.Città
AND GiornoSett = "Giovedì"
AND CittàArr = "Roma"
GROUP BY TipoAereo;

SELECT N_utilizzi
FROM Best_aereo
WHERE N_utilizzi = (SELECT MAX(N_utilizzi)
                    FROM Best_aereo));
```

- Nel riquadro, scrivere una query SQL che restituisce l'id e la città di arrivo dell'ultimo volo in partenza il giovedì dall'aeroporto di Napoli (2.5 punti)

```
CREATE VIEW Ultimo_volo (Volo, Arrivo) AS
SELECT IdVolo, CittàArr, MAX(OraPart) AS Ultima_partenza
FROM Volo
WHERE GiornoSett = "Giovedì"
AND CittàPart = "Napoli"
GROUP BY IdVolo, CittàArr;

SELECT Volo, Arrivo
FROM Ultimo_volo
WHERE Ultima_partenza = (SELECT MAX(Ultima_partenza)
                        FROM Ultimo_volo));
```

Sia data la seguente relazione $R(ABCDE)$, con copertura ridotta $G=\{B \rightarrow C, B \rightarrow E, C \rightarrow A \text{ e } C \rightarrow D\}$.

Risolvere i seguenti punti:

- Trovare la/e chiave/i di R , motivando la risposta.
- Effettuare una decomposizione in 3NF ed indicare le chiavi delle relazioni finali ottenute.
- Indicare se la decomposizione ottenuta al punto b è anche in BCNF rispetto all'insieme di dipendenze in G . Motivare la risposta.

②

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

⑤ DSCOMP. LN 3NF

[1.] RELAZIONI $\alpha X \rightarrow A, Y \rightarrow B$
 $\alpha X^t = Y^t$

$\{B \rightarrow C, B \rightarrow D\} \in \{C \rightarrow A, \}$
 $\{C \rightarrow D\}$

[2.] $R_1 (\underline{B}, C, D)$
 $R_2 (C, A, D)$

[3.] $X \subseteq Y \rightarrow \text{UNA SOLA} \rightarrow$
(NOPE)

[4.] SO CITTÀ SENZA
RELAZIONI \rightarrow
AGGIUNGERE!
(NOPE)

[5.] ∇ DISCOMPOSITIONS

$R_1 (\underline{B}, C, D)$ con $B \in C$
 $R_2 (\underline{C}, A, D)$ CATTOLICI...

[C] BCNF? \rightarrow SI

SOLN SONO A PERDITA?



ESORDIA \rightarrow CHIAB

Esercizio 4: Decomposizione in Terza Forma Normale (5 punti)

Sia data la relazione $R(A,B,C,D,E)$ e l'insieme di dipendenze associato $F=\{AB \rightarrow CDE, AC \rightarrow BDE, B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}$:

- Trovare la/e chiave/i di R , motivando la risposta.
- Calcolare la copertura ridotta
- Effettuare una decomposizione in 3NF, indicando le chiavi delle relazioni ottenute
- Indicare se c'è o ci può essere perdita nel join, motivando la risposta.

① $AB^+ \subseteq \{A, B, C, D, E\}$
 $AC^+ \subseteq \{A, C, B, D, E\}$ \rightarrow CHIAVI

$B^+ = \{B, C, D, E\}$

$C^+ = \{C, B, D, E\}$

②

$F = \{AB \rightarrow CDE, AC \rightarrow BDE, B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}$

$AB \rightarrow C$	$AC \rightarrow B$	$B \rightarrow C$
$AB \rightarrow D$	$AC \rightarrow D$	$C \rightarrow B$
$AB \rightarrow E$	$AC \rightarrow E$	$C \rightarrow D$
		$B \rightarrow E$

REDUNDANTS

I passi per calcolare la copertura ridotta di una relazione sono i seguenti:

1. Sostituzione dell'insieme dato con quello equivalente che ha tutti i secondi membri costituiti da singoli attributi;
2. Per ogni dipendenza verifica dell'esistenza di attributi eliminabili dal primo membro;
3. Eliminazione delle dipendenze ridondanti.

① 3NF

$\{B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow \cancel{D}\}$

1. $\{B \rightarrow C, B \rightarrow \cancel{D}\}$

$\{C \rightarrow B, C \rightarrow D\}$

2. $R_1(\underline{B}, C, \cancel{D}) \rightarrow B/C$

$R_2(\underline{C}, B, D)$ ~~CHIAVI~~

3. $\cancel{X} \subseteq Y \rightarrow \underline{no}$

4. SO ~~CHIAVI~~ SONO
~~RELATIONS~~
 \rightarrow AGGIUNGERE

$R_3(\underline{A}, B)$

$R_4(\underline{A}, C)$

5. RELAZIONI CON CHIAVI

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

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

$R_3 (\underline{A}, B)$

$R_4 (\underline{A}, C)$

- POSSIBILE AVERE SOLU? \rightarrow SI

(A NON COMPARE
DAPPERTOVV...)

Si consideri il seguente log:

\rightarrow CK(T2), B(T5), B(T6), U(T5,O5,B5,A5), C(T5), B(T7), U(T7,O6,B6,A6),
B(T8), U(T6,O1,B7,A7), A(T7), C(T6), guasto

Sapendo che occorre fare l'UNDO di T2, T7 e T8, e il REDO di T5 e T6. Quale è la prima operazione da effettuare per la ripresa a caldo?

1. O5=A5
2. O1=A7
3. O1=B7
4. O6=B6

$O_6 = B_6$

- RIPRESA A FREDDO: TROVA CK

(VAL AZZERATO)

5 FAI

RIPRESA
A CALDO

- RIPRESA A CALDO:

(VAL AVANTI 5 RIPRESTINA)

Si consideri la seguente base di dati per la partecipazione di delegati a meeting:

Delegato(IdDelegato, Nome, Cognome, Organizzazione)

Partecipa(IdDelegato, IdMeeting)

Meeting(IdMeeting, Citta, Nazione, Data)

A. Nel riquadro, scrivere una query in Algebra Relazionale che restituisca Nome e Cognome dei delegati che hanno partecipato solo a meeting non italiani (2.5 punti).²

$$D_1 = \sigma_{\text{Nazione} \neq \text{"ITALIA"}} (\text{PARTECIPA})$$

$$D_2 = D_1$$

$$\pi_{\text{nome}, \text{cognome}} D_1 \rightarrow$$

$$\left(\pi_{\text{nome}, \text{cognome}} D_1 \right)$$

$$D_1 \bowtie D_1$$

Si consideri la seguente base di dati per la partecipazione di delegati a meeting:

Delegato(IdDelegato, Nome, Cognome, Organizzazione)

Partecipa(IdDelegato, IdMeeting)

Meeting(IdMeeting, Citta, Nazione, Data)

B. Nel riquadro, scrivere una query in Standard SQL che restituisce, per ogni organizzazione O, il numero di delegati che hanno partecipato a tutti i meeting. (2.5 punti).³

```
/* 0' Lions */
```

```
CREATE DELEGATI_TUTTI_MEETING (IdDelegato) AS
SELECT IdDelegato
FROM PARTECIPA
GROUP BY IdDelegato
HAVING COUNT(*)=(SELECT COUNT(*) FROM MEETING);
```

```
SELECT Organizzazione, COUNT(*)
FROM DELEGATI_TUTTI_MEETING DT, DELEGATO D
WHERE DT.IdDelegato=D.IdDelegato
GROUP BY Organizzazione
```

C. Nel riquadro, scrivere una query in Standard SQL che restituisce, per ogni delegato, la data del primo meeting a cui ha partecipato (2 punti).

```
SELECT IdDelegato, MIN(Data) AS Prima_data
FROM Meeting M, Partecipa P
WHERE M.IdMeeting = P.IdMeeting
GROUP BY IdDelegato;
```

Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati con le relazioni:

- **Fantini** (Nome, Peso, DataNascita)
- **Cavalli** (Nome, AnnoNascita, Scuderia)
- **Gare** (Codice, Nome, Luogo, Data, CavalloVincente, FantinoVincente)

dove $\text{Gare.CavalloVincente} \rightarrow \text{Cavalli.Nome},$
 $\text{Gare.FantinoVincente} \rightarrow \text{Fantini.Nome}$

A. Il nome dei fantini che hanno vinto solamente gare con cavalli della stessa scuderia (2 punti).²

$F_1 = \Pi_{\text{norm}} \text{FANWU} \otimes$
 GARS

~~CAVALLI~~
CAVALLI

CAVALLI.1025
=
GARS.
CAVALLI.1025

$$F_2 = F_1$$

$\Pi_{F_1.1025} F_1$ /

$(\Pi_{F_1.1025} F_1 \quad \text{F1. SCUDERIA} \quad \text{F2. SCUDERIA})$

B. Per ogni fantino che abbia vinto almeno una gara, fornire il nome del fantino ed anche il codice e nome della prima gara che ha vinto (2.5 punti).³

```
SELECT FantinoVincente, Codice, Nome,  
MIN(Data) AS Prima_gara  
FROM Gare G, Fantini F  
WHERE F.Nome = G.FantinoVincente  
GROUP BY FantinoVincente, Codice, Nome;
```

// O' Lions

```
CREATE VIEW PRIMAGARA(FANTINO, DATA) AS  
SELECT FANTINOVINCENTE, MIN(DATA)  
FROM GARE  
GROUP BY FANTINOVINCENTE;
```

```
SELECT FANTINO, P.DATA, G.CODICE  
FROM PRIMAGARA AS P, GARE AS G  
WHERE P.DATA=G.DATA  
AND FANTINOVINCENTE=FANTINO;
```

C. Restituire l'anno di nascita medio dei cavalli con cui ha vinto il fantino più giovane – cioè con la data di nascita più grande (2.5 punti).³

```
SELECT AVG(AnnoNascita), MAX(DataNascita)
FROM Fantini F, Cavalli C, Gare G
WHERE F.Nome = G.FantinoVincente
AND G.CavalloVincente = C.Nome;
```

// Più semplice così..

[AnnoNascita]	[DataNascita]
0	0
1	1
2	2
.....	