

### Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati per i voli che vengono effettuati nel mondo:

VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO(TipoAereo, NumPasseggeri, QtaMerci)

A. Restituire il tipo di aereo con il [massimo numero di passeggeri,] tra quelli in uso per voli che partono da Roma (2 punti).

$$C_2 = C_1 \quad , \quad C_1 = \pi_{\text{TIPOAEREO}} \left( \sigma_{\text{CITTÀPART} = \text{"Roma"}} \left( \text{VOLO} \bowtie \text{AEREO} \right) \right)$$

$$\pi_{C_1.\text{TIPOAEREO}} \left( C_1 \bowtie C_1.\text{NUM-P} > C_2.\text{NUM-P} \right)$$

↓

$$\pi_{\text{TIPOAEREO}} \text{ NOT MAX}$$

$$\pi_{\text{TIPOAEREO}} C_1 \bowtie \pi_{\text{TIPOAEREO}} \text{ NOT MAX}$$

### Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati per i voli che vengono effettuati nel mondo:

VOLO(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

AEREO(TipoAereo, NumPasseggeri, QtaMerci)

Nel riquadro, scrivere una query in Standard SQL che restituisce il giorno della settimana in cui ci sono più voli in arrivo a Roma. (2.5 punti)

```
CREATE VIEW Voli_per_giorno (GiornoSett, Num) AS
SELECT GiornoSett, COUNT(*)
FROM Volo
WHERE CittàArr = "Roma"
GROUP BY GiornoSett;

SELECT GiornoSett
FROM Voli_per_giorno
WHERE Num = (SELECT MAX(Num) FROM Voli_per_giorno);
```

### Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati per i voli che vengono effettuati nel mondo:

**VOLO**(IdVolo, GiornoSett, CittàPart, OraPart, CittàArr, OraArr, TipoAereo)

**AEREO**(TipoAereo, NumPasseggeri, QtaMerci)

Scrivere una query in Standard SQL che restituisce, per ogni città C da cui partono almeno 100 voli settimanalmente, restituisce il numero medio di aerei che giornalmente partono da C.

// Raggruppa il numero totale di aerei partendo da C

```
CREATE VIEW N_medio_aereo (CittaPartenza, Numero) AS
SELECT CittaPart, COUNT(*)
FROM Volo
GROUP BY CittaPart
HAVING COUNT(*) ≥ 100 ;
```

```
SELECT CittaPartenza
FROM N_medio_aereo
WHERE Numero = (SELECT AVG(Numero) FROM N_medio_aereo);
```

// O'Lions (Leonidas) - Optimized solution (trust me bro)

```
SELECT CittaPart, Count(*)/7
FROM VOLO
GROUP BY CittaPart
HAVING Count(*) ≥ 100
```

### Esercizio 4: Normalizzazione (6 punti)

Sia data la relazione  $R(A, B, C, D, E, F)$  con copertura ridotta  $G = \{C \rightarrow B, C \rightarrow D, C \rightarrow F, BE \rightarrow C, B \rightarrow A, AD \rightarrow E\}$

- Trovare la/e chiave/i di R, motivando la risposta.
- Quali dipendenze violano la 3NF? Motivare la risposta
- Effettuare una decomposizione in 3NF.
- La decomposizione è anche in BCNF? Motivare la risposta

$$AD^+ = \{A, D, E\}$$

(a.)  $\Rightarrow$  COPERTURA  $\rightarrow C^+ = \{C, A, B, D, E, F\}$   
 $BE^+ = \{C, A, B, D, E, F\}$   
 $C = CHIAVE$   
 $C / BE \rightarrow$  CHIAVI

(b.) 3NF  $X \rightarrow Y$ , X è SUPERCHIAVE  
 $B \rightarrow A$  (VIOLA)

(C)  $\{C \rightarrow B, C \rightarrow D, C \rightarrow F, BD \rightarrow C, B \rightarrow A, AD \rightarrow B\}$

1. DF,  $X \rightarrow A \text{ e } Y \rightarrow B$

$$X = Y \text{ se } X^+ \subseteq Y^+$$

$\{C \rightarrow B, C \rightarrow D, C \rightarrow F, BD \rightarrow C\} / \{B \rightarrow A\} / \{AD \rightarrow B\}$

2. REL.  $\rightarrow R_1(\underline{C}, \underline{B}, D, \underline{F})$   
 $R_2(B, A)$   $\uparrow$   $C/B$   
 $\underline{B}, \underline{AD} / R_3(\underline{AD}, \delta)$

3.  $A(X) \subseteq B(Y) \rightarrow \text{Accorpo}$   
 $X$

4. CHIAVI PARZIALI SIC!  
 $B \rightarrow A$

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

BCNF  $R_2(\underline{B}, A)$   
 $\checkmark$   $R_3(\underline{AD}, \delta)$

BCNF  $\Rightarrow$  HA! DIF. PARZIALI SU  
UNA CHIAVI  
 $\rightarrow$  LEVA RELAZIONE

$R_1(\underline{C}, \underline{B}, D, \underline{E}, F, \textcircled{A}) \rightarrow \{B \rightarrow A\}$

$R_2(\underline{A}, D, b)$

$\Rightarrow$  AGGIUNGI  $R_3(B \rightarrow A) \checkmark$

Data la relazione  $R(\underline{A}, \underline{B}, C, D)$  e  $S(\underline{W}, X, Y, Z)$ , indicato con  $|R|$  e  $|S|$  il numero di tuple di  $R$  e  $S$ . Sapendo che non ci sono chiavi esterne, quale affermazione è vera per l'operazione  $S \bowtie_{A=W} R$  in Algebra Relazionale?

1.  $0 \leq S \bowtie_{A=W} R \leq |R|$
2.  $0 \leq S \bowtie_{A=W} R \leq |R| * |S|$
3.  $0 \leq S \bowtie_{A=W} R \leq |S|$
4.  $0 \leq S \bowtie_{A=W} R \leq \max(|S|, |R|)$

$S \bowtie_{A=W} R \rightarrow \text{TUTTO}$

$S \bowtie_{A=W} R \text{ (NON HA!)} \rightarrow R > S$

$0 \leq S \bowtie_{B=W} R \leq |S|$

NAL. ASSOLUTO

= CARDINALITÀ  
= N. RIGHE DI S

$0 \leq S \bowtie_{C=X} R$

$\max(|S|, |R|)$

POSTERIORE  
ALGEBRA RELAZIONALE

## Esercizio 2 - Capitolo 3

Considerare le relazioni

$R_1(\underline{A}, B, C)$  con cardinalità  $N_1$

$R_2(\underline{D}, E, F)$  con cardinalità  $N_2$

Assumere che sia definito un vincolo di integrità referenziale fra:  
l'attributo  $C$  di  $R_1$  e la chiave  $D$  di  $R_2$

Indicare la cardinalità ( $K$ ) di ciascuno dei seguenti join (specificare l'intervallo nel quale essa può variare):

$$1. K_1 = R_1 \bowtie_{A=D} R_2 \quad 0 \leq |K_1| \leq \min(N_1, N_2)$$

$$2. K_2 = R_1 \bowtie_{C=D} R_2 \quad |K_2| = N_1$$

$$3. K_3 = R_1 \bowtie_{A=F} R_2 \quad 0 \leq |K_3| \leq N_2$$

$$4. K_4 = R_1 \bowtie_{B=E} R_2 \quad 0 \leq |K_4| \leq N_1 \cdot N_2$$

$\bowtie$  Full Join

Sia data la seguente porzione di log fino al guasto:  $[CK(T5, T6), B(T7), U(T7, O6, B6, A6), U(T6, O3, B7, A7), B(T8), I(T8, O5, A5), C(T8), A(T5)]$ . Sapendo che occorre effettuare l'UNDO di  $T5, T6$  e  $T7$  e il REDO di  $T8$ , quale è la prima operazione da effettuare per la ripresa a caldo?

1.  $D(O5)$

2.  $O3=B7$

3.  $O6=B6$

4.  $O5=A5$

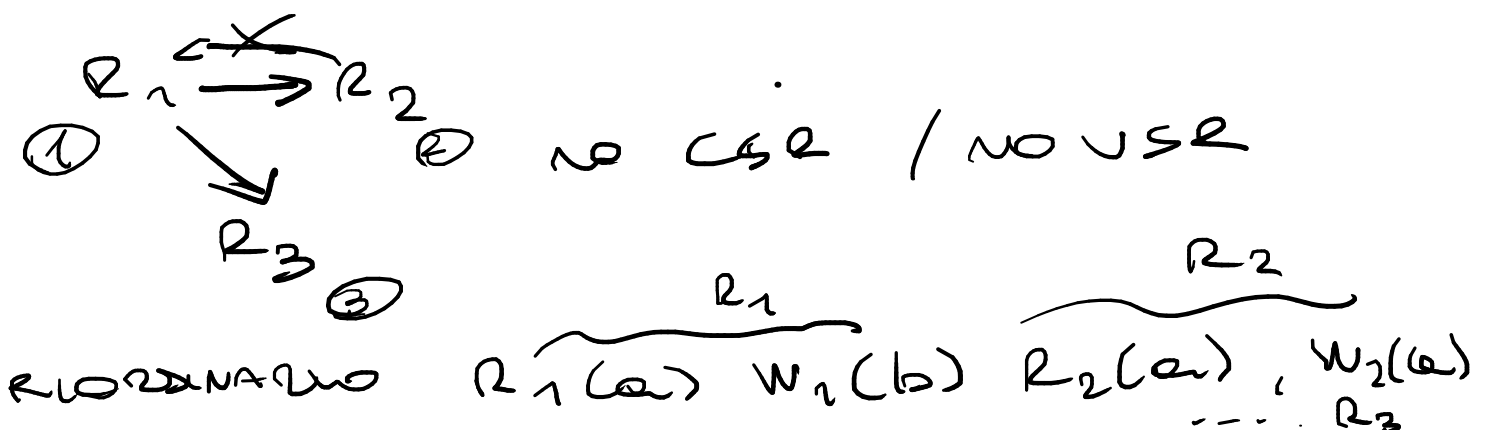
$\rightarrow O3=B7$   $\{UNDO\} \rightarrow COMMIT$   
 $\{REDO\} \Rightarrow CHECK$

$\{T8\} = UNDO$

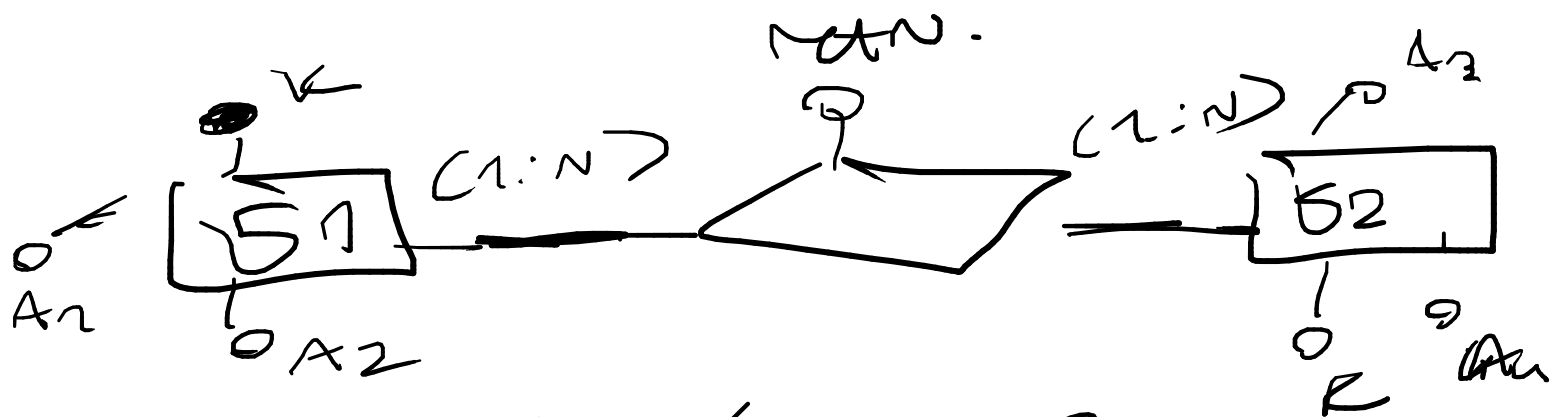
$\{T5, T6\} = REDO$

Si consideri lo schedule  $S = r1(a), r2(b), w1(b), r2(a), w2(a), r3(a), w3(a)$

Indicare se  $S$  è view-serializzabile (VSR) e/o conflict-serializzabile (CSR).







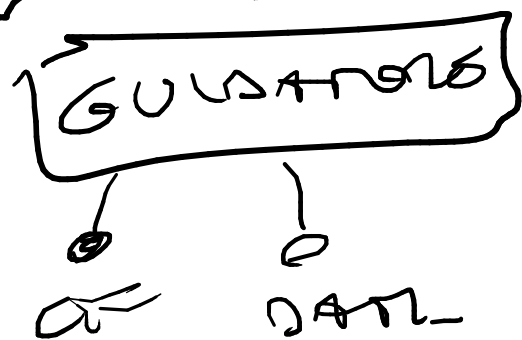
$$S_1 = \underline{K} (A_1, A_2)$$

$$S_2 = \underline{K} (A_3, A_4)$$

$$S_3 = \underline{K_1 / K_2}$$

CLASSICA

normalization



ANALISI

LOGICO: PK / FK

→ AUTO = TARGA / CIL

GOVERNORS

1: N DA ENTREGAS  
(0: N)

SO 1:1

→ NO  
TAB. IN

(A → B)  
FK

SCADA  
LOGCO  
POR

GRUPUS

(A ← B)  
FK

ALABRAS

---

C<sub>1</sub>/C<sub>2</sub>

3 - SQL/  
ALGEBRA

C<sub>1</sub> = CONCORDO

SESTANTOS

✗

10 CONCORDO

10<sup>2</sup>

C<sub>2</sub> = C<sub>1</sub>



$\#$  NOMINATIVO (PERSONA  
 $\bowtie C_1.CF =$   
 $PCF$   
 $[C_1 \bowtie$   
 $C_1.CANTANTE$   
 $\hookrightarrow$   
 $C_2.CANTANTE$   
 $AND$   
 $C_1.CF < C_2.CF$

---

Scrivere una query in standard SQL che restituisce la città in cui è avvenuto il concerto con il numero più alto di spettatori

```

CREATE VIEW Spettatori_per_concerto AS
SELECT ID-Concerto, COUNT(CF) AS Numero_spettatori
FROM Spettatore
GROUP BY ID-Concerto;

```

```

SELECT Città
FROM Concerto C, Spettatori_per_concerto S
WHERE C.ID-Concerto = S.ID-Concerto
AND Numero_spettatori =
(SELECT MAX (Numero_spettatori)
FROM Spettatori_per_concerto));

```

Scrivere una query in standard SQL che, per ogni cantante, restituisce il CF del cantante e il numero medio di spettatori ai suoi concerti



```
SELECT DISTINCT(Cantante), AVG(Numero_spettatori)
FROM Concerto C, Spettatori_per_concerto S
WHERE S.ID-Concerto = C.ID
GROUP BY Cantante;
```

---

①

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

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

$AB/AC \rightarrow CHAU$

②

STRUTTURA RIDOTTA

a.

55 CONDIZIONI SINGOLE

ATTRIBUTI

b.

NO DIP. TRANSITIVE

c. NO DIP. RIDONDATA

$$AB \rightarrow CDG$$

$$AB \rightarrow C /$$

$$AB \rightarrow D$$

$$AB \rightarrow E$$

$$AC \rightarrow BDG$$

$$B \rightarrow C$$

$$AC \rightarrow B /$$

$$AC \rightarrow D$$

$$AC \rightarrow E$$

$$C \rightarrow B$$

$$C \rightarrow D$$

$$B \rightarrow E$$

~~$$AB \rightarrow C$$~~

~~$$AB \rightarrow D$$~~

~~$$AB \rightarrow E$$~~

~~$$AC \rightarrow B$$~~

~~$$AC \rightarrow D$$~~

~~$$AC \rightarrow E$$~~

$$B \rightarrow C$$

$$C \rightarrow B$$

$$C \rightarrow D$$

$$B \rightarrow E$$

TEMERARIO

QUESTÃO

TOGUANO

A<sup>2</sup> RESULTADO

REDUNDANTE

$$\textcircled{C} \quad \underline{\exists \in \mathcal{N}} \Leftrightarrow \underline{X} \in SK$$

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

$$\textcircled{1} \quad X \rightarrow A \mid Y \rightarrow B \text{ use}$$

$$X^+ = Y^+$$

$$B^+ \subseteq C^+ \rightarrow \{B \Rightarrow C, C \Rightarrow B, C \Rightarrow D, B \Rightarrow D\}$$

$$\textcircled{2} \rightarrow Q_1(B, C, D, B)$$

$$\textcircled{3} \quad A(X), B(Y), X \subseteq Y$$

$$\forall B \subseteq A \rightarrow \text{NOT } A$$

$$\textcircled{4} \rightarrow R_2(A, B), R_3(A, C)$$

$$R_1 \rightarrow \{ \underline{B, C}, D, E \} \rightarrow \{ B \rightarrow C, \\ C \rightarrow B,$$

$$R_2 \rightarrow \{ \underline{A}, \underline{B} \}$$

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

$$R_3 \rightarrow \{ \underline{A}, \underline{C} \}$$

solo B, C  $\rightarrow$  NO PERDITA  
SOLTO

ANCORA CON  
A

$\rightarrow$  SI PERDITA

$$R(\underline{A, B}, C, D) \rightarrow AB \rightarrow C \\ C \rightarrow D$$

$$|R| = N + D + \text{ripet}$$

$$\pi_{A, B}(R) \subseteq (R)$$

$$\pi_{A, B}(R) < |R| \rightarrow C$$

1

$$\pi_{A,B}(R) \subset (R) \rightarrow C$$

A/B  
(DA SOLB)

$$\pi_{A,B}(R) = \emptyset \rightarrow D$$

(NO CURANS)

---

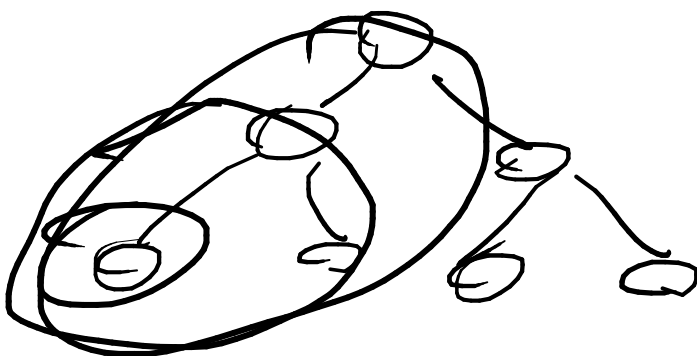
INDEX  $\rightarrow$  HASH

└ B-TREE

(#)  $\rightarrow$  KEY / VALUES

HASH  $\rightarrow O(1) \rightarrow$  KEY

B-TREE  $\rightarrow$  RELAZIONI



ANRAT

$\rightarrow O(\underbrace{m \log(m)})$

$\downarrow$

LF

HASH  $\rightarrow (B, A)$   
                   $\downarrow$              $\downarrow$   
                   $O(1)$      $O(1)$

B + TRUSS  $\rightarrow \underline{[A, B]}$

$S(A, B, C)$

$A=6$  &  $B=7$

SUBSET ~~X~~ FROM S

WHERE  $B \geq 7$



HASH SUB B

Data la relazione e R(A, B, C) la query SELECT \* FROM R ORDER BY C, quale dei seguenti indici velocizza l'esecuzione della query?

- (1) Indice Hash su C
- (2) Indice B+Tree su C
- (3) Indice Hash su A, B, C,
- (4) Indice B+Tree su A, B, C

→  
no warrants

→ ORDER → B-TREE

Si consideri le relazioni R(A, B, C, D) e la seguente query  
SELECT MIN(A) FROM R WHERE B=10

Quale dei seguenti indici garantisce l'efficienza massima?

1. Indice Hash sulla coppia (B,A)
2. Indice Hash sulla coppia (A,B)
3. Indice B-TREE sulla coppia (A,B)
4. Indice B-TREE sulla coppia (B,A)

✓. B+ Tree	mediamente bene per = <del>consigliati per intervalli &lt; &gt;</del> bene per cancellazione o inserimento
B- Tree	mediamente bene per = <del>male per cancellazione o inserimento</del> mediocri in generale
Hash	migliori per uguaglianze male ORDER BY

uguaglianza	Hash
intervallo	B+ Tree
uguaglianza e intervallo	B+ Tree
cancellazione/inserimento	B+ Tree
ORDER BY	B+ Tree



ORDING (WHEN)

→  $z > 4$  ~~000~~  $x = 5$

$(x, z)$

→ ORDING DATA UOMO  
ALPWO

→ SO WHEN  $e$

MIN /  
MAX / ORDERBY  
(AGGREG.)

→ RESCORDERA "WITORS"

SUBJECT  $\approx$  FROM R

WHERE  $B = "10"$

↓

HASH

SUBJECT RUN (A)

WHERE  $B = "10"$

(GETTABLE  $\rightarrow$  SOURCE B-1)

↓

B - TREE

$\left( \begin{array}{l} B^+ - \text{TREE} \\ B^- - \text{TREE} \end{array} \right)$

---

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.

```
SELECT SquadraCasa, SUM(GoalCasa) AS Somma_Goal
FROM Partita
WHERE Stagione = "2023-2024"
GROUP BY SquadraCasa
ORDER BY Somma_Goal;
```

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

```
SELECT L2.Squadra, AVG(GoalCasa)
FROM Partita P, Luogo L1, Luogo L2

WHERE L1.Squadra = P.SquadraTrasferta
AND L2.Squadra = P.SquadraCasa
AND L1.Citta = L2.Citta

GROUP BY L2.Squadra
HAVING COUNT(L2.Squadra) ≥ 10;
```