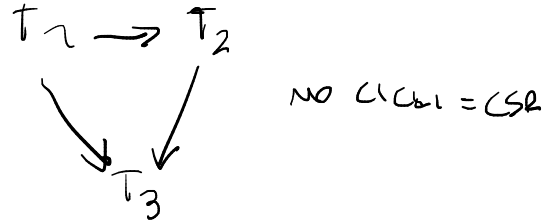


Domanda 3 (2 Punti)

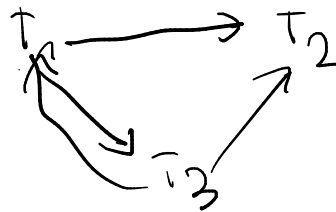
Si consideri lo schedule $S = \{r1(a) \ w2(a) \ r3(a) \ w1(b) \ w3(c) \ r2(d) \ r3(d) \ w3(b) \ w3(e)\}$

Dopo aver calcolato il grafo dei conflitti, indicare se S è conflict-serializzabile (CSR) e/o view-serializzabile (VSR)?

1. R è in CSR **ma non** in VSR
2. R **non** è in CSR **ma** è in VSR
3. R è in CSR ed è in VSR
4. R **non** è in CSR **e non** è VSR



Dato lo schedule $S = r1(x) \ r1(t) \ r2(z) \ w3(x) \ w1(x) \ r1(y) \ w3(t) \ w2(x) \ w1(y)$, indicare se S è view e/o conflict-serializzabile, motivando la risposta. Se S è view e/o conflict-serializzabile, indicare uno schedule seriale che è view e/o conflict-equivalente, motivando la risposta.



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. $K_1 = R_1 \bowtie_{A=D} R_2$ $0 \leq |K_1| \leq \min(N_1, N_2)$ DISTRIBUZIONE? ID → MATCH PARTIAL
 2. $K_2 = R_1 \bowtie_{C=D} R_2$ $|K_2| = N_1 \rightarrow$ DISTRIBUZIONE? ID → MATCH PARTIAL
 3. $K_3 = R_1 \bowtie_{A=F} R_2$ $0 \leq |K_3| \leq N_2$
 4. $K_4 = R_1 \bowtie_{B=E} R_2$ $0 \leq |K_4| \leq N_1 \cdot N_2$ PRODOTTO CARTESIANO (X)
- NON KEY

Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati per le votazioni locali in diverse città italiane:

- **RISULTATO-PARTITO** (Città, Partito, Anno, Percentuale)
- **CANDIDATURA** (Candidato, Anno, Partito, Città, NumVoti)

Una tupla (PD, Padova, 2022, 30) in RISULTATO-PARTITO indica che il partito PD ha preso il 30% dei voti a Padova nel 2022. Una tupla (Giordani, 2022, PD, Padova, 3000) in CANDIDATURA indica che Giordani ha preso 3000 voti a Padova nel 2022 con il PD.

A. Restituire il nome del candidato che si sono presentati una sola volta (cioè in un solo anno) per il PD (2 punti).³

$$C_1 = \pi_{\text{CANDIDATO}} \left(\sigma_{\text{PARTITO} = \text{"PD"}} \left(\text{RISULTATO-PARTITO} \bowtie \text{CANDIDATURA} \right) \right)$$

$$C_2 = C_1$$

$$C_3 = C_1 \setminus \pi_{\text{CANDIDATO}} \left(C_1 \bowtie C_1 \text{ Anno} \neq C_2 \text{ Anno} \right) \quad (2)$$

$C_1 = \sigma_{\text{Partito} = \text{"PD"}} (\text{CANDIDATURA})$

$C_2 = C_1$

$\text{DUE_CANDIDATURE} =$

$\pi_{\text{C1.CANDIDATO}}$

$(C_1 \bowtie_{\text{C1.CANDIDATO} = \text{C2.CANDIDATO AND C1.PARTITO} \neq \text{C2.PARTITO}} C_2)$

$(\pi_{\text{C1.CANDIDATO}} (C_1)) \setminus \text{DUE_CANDIDATURE}$

Nel riquadro, scrivere una query in Standard SQL che, per ogni città con votazioni nel 2022, restituisce il partito che ha ottenuto più voti (2.5 punti)

$\underbrace{\text{IN PARTS VIEW}}_{\text{MAX (PARTS VIEW)}}$

```
CREATE VIEW Top_Voti (Città, Partito, Massimo_voti) AS
SELECT Città, Partito, MAX(Percentuale)
FROM Risultato-Partito RP
WHERE Anno = 2022
GROUP BY Partito, Città;
```

```
SELECT Città, Partito
FROM Top_Voti
WHERE Massimo_Voti = (SELECT MAX(Percentuale)
                      FROM Risultato-Partito));
```

C. Nel riquadro, scrivere una query in Standard SQL che, per ogni città con votazioni nel 2022, restituisce il partito che ha ottenuto più voti (2.5 punti)

```
CREATE VIEW TOP-PERCENTUALE-
CITTA(CITTA,TOP) AS
    SELECT CITTA, MAX(PERCENTUALE)
    FROM RISULTATO-PARTITO
    WHERE ANNO=2022
    GROUP BY CITTA

SELECT CITTA, PARTITO
FROM RISULTATO-PARTITO R,
    TOP-PERCENTUALE-CITTA T
WHERE R.CITTA=T.CITTA
```

"Nel riquadro, scrivere una query in Standard SQL che restituisce, per ogni partito che ha ottenuto almeno il 10% a Padova nel 2022, il numero medio di voti tra tutti i candidati del 2022."

```
CREATE VIEW Candidati_2022 (Partito, Numero_Medio_Voti)
SELECT Partito, AVG (NumVoti)
FROM Risultato-Partito RP
JOIN Candidatura C ON RP.Anno = C.Anno
WHERE Percentuale ≥ 10
AND Città = Padova
AND Anno = 2022;
```

// 0' Lions

```
SELECT PARTITO,AVG(NUMVOTI)
FROM CANDIDATURA C
WHERE ANNO=2022 AND CITTA='Padova'
AND PARTITO IN
    (SELECT PARTITO FROM RISULTATO-PARTITO
    WHERE PERCENTUALE>=10 AND ANNO=2022
    AND CITTA='Padova')
GROUP BY PARTITO
```

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ad una compagnia operante nel campo della cybersecurity, che raccoglie dati su malware, antimalware ed utenti.

Di ogni malware¹ interessa il codice (identificatore), l'anno in cui è apparso ed il nome dei sistemi operativi (almeno uno) in cui può operare. Di ogni antimalware interessa il codice (identificativo), il costo di sviluppo ed i malware (almeno uno) che esso è potenzialmente in grado di rilevare.

Di ogni utente interessa il codice fiscale (identificativo), l'anno di registrazione e gli episodi di infezione di cui è stato vittima, ciascuno con le seguenti caratteristiche: (i) il malware coinvolto, (ii) il danno monetario subito e (iii) la data in cui l'infezione è avvenuta. Si noti che un utente può infettarsi dello stesso malware, ma in giorni diversi.

Di ogni utente interessa anche sapere quali sono stati gli antimalware di cui si è dotato nel tempo mediante abbonamento. In particolare:

- riguardo agli antimalware ai quali l'utente è attualmente abbonato interessa sapere la data di inizio dell'abbonamento.
- riguardo agli abbonamenti scaduti di antimalware, interessa la somma complessivamente pagata dall'utente per l'abbonamento, la data di inizio e la data di fine dello stesso e la ragione della disdetta (se tale ragione è nota).

Per utente nel nostro contesto si intende o una persona o una organizzazione di interesse per la compagnia di cybersecurity. Di ogni persona interessa il sesso, la data di nascita. Di ogni organizzazione interessa il fatturato e la città in cui è ubicata la sede legale

