

Matricola	Nome e Cognome	Valutazione

Si consideri il seguente schema relazionale per la gestione di un programma frequent flyer:

Membro(numero\_tessera, nome, cognome, totale\_miglia)

Movimento(id\_movimento, tessera, data, tipologia, miglia)

Itinerario(id, partenza, destinazione, dataora\_partenza, dataora\_arrivo)

Tratta(itinerario, codice\_volo, aeroporto\_partenza, aeroporto\_arrivo, miglia\_base)

ItinerarioMembro(itinerario, membro, classe, completato)

Si consideri che nella tabella ItinerarioMembro, la classe può assumere i valori “prima”, “business”, “economy” e l’attributo completato è un booleano (valori 1/0). Inoltre, nella tabella movimento, la tipologia può essere “entrata” o “uscita”. Le miglia totali di un itinerario si calcolano sommando tutte le miglia delle tratte che compongono l’itinerario. Se il membro vola in prima classe le miglia sono incrementate del 50%; se business class l’incremento è del 30%.

1. Identificare le chiavi primarie ed esterne dello schema [0 corretta, -1 errata]

**2. Algebra**

- Identificare i membri che non hanno fatto alcun volo nel 2020 [3 punti].
- Per ogni membro, identificare le tratte di volo che consentono di accumulare più miglia\_base (indipendentemente dalla classe) [4 punti].

**3. SQL**

- Identificare i membri che hanno volato su itinerari che prevedono il minor numero di tratte, indicando per ogni membro il numero medio di miglia accumulate su tali itinerari [6 punti].
- Identificare gli itinerari più brevi (in termini di date) che consentono di accumulare più miglia. Per ogni itinerario indicare quanti membri hanno prenotato tale itinerario e quanti hanno effettivamente completato l’itinerario [6 punti].

Matricola	Nome e Cognome	Valutazione

2a)

$$\pi_{\text{MemberID}}(\sigma_{\text{MemberID} \neq \text{TrainID}}(\text{Members})) - \left( \sigma_{\text{TrainID} \neq \text{MemberID}} \left( \sigma_{\text{TrainID} = \text{MemberID}} \left( \sigma_{\text{TrainID} \neq \text{MemberID}}(\text{Members}) \right) \right) \right)$$

2b)

R1:  $\pi_{\text{MemberID}, \text{ITimeID}, \text{CostValue}, \text{MightValue}}(\sigma_{\text{MemberID} \neq \text{ITimeID}}(\text{Members}))$

R2:  $\pi_{\text{MemberID}, \text{MightValue}, \text{CostValue}, \text{ITimeID}} \left( \sigma_{\text{MemberID} \neq \text{MightValue}} \left( \sigma_{\text{MemberID} = \text{MightValue}} \left( \sigma_{\text{MemberID} \neq \text{MightValue}}(\text{Members}) \right) \right) \right)$

R1 JOIN R2

3a) CREATE VIEW VI AS  
 SELECT COUNT(\*) MTrain, member, itime, AVG(might) AvgMight  
 FROM ITIME JOIN MEMBERS NATURAL JOIN TRAIN  
 GROUP BY member, itime

SELECT \*

FROM VI

WHERE MTrain < (SELECT MIN(MTrain)

FROM VI)

3b) CREATE VIEW VI AS  
 SELECT DEDIFF(DOA, DOP) Duration, COUNT(member) promotions,  
 SUM(might) might, SUM(cost) cost  
 FROM TRAIN NATURAL JOIN ITIME NATURAL  
 JOIN MEMBERS  
 GROUP BY member, itime

SELECT \*

FROM VI

WHERE NOT EXISTS

(SELECT \* FROM VI

WHERE Duration < V.Duration AND

might > V.might)

4)

Matricola	Nome e Cognome	Valutazione

**4. Trigger**

Implementare un trigger che, quando un membro completa un itinerario, aggiunge un nuovo movimento di “entrata” con la data corrente e il totale delle miglia accumulate in quel volo. Inoltre, il trigger dovrà aggiornare il totale delle miglia del membro [5 punti].

```
CREATE TRIGGER AggiornaMiglia
AFTER UPDATE OF completato ON ItinerarioMembro
FOR EACH ROW
WHEN new.completato = 1 AND old.completato = 0
BEGIN
    DECLARE x FLOAT DEFAULT 0;
    SELECT
        SUM(miglia_base) *
        CASE
            WHEN new.classe = 'prima' THEN 1.5
            WHEN new.classe = 'business' THEN 1.3
            ELSE 1.0
        END
    INTO x
    FROM Tratta WHERE Itinerario = new.Itinerario;

    INSERT INTO MOVIMENTO (tessera, data, tipologia, miglia)
    VALUES (new.membro, CURRENT_DATE, 'entrata', x)

    UPDATE Membro
    SET totale_miglia = totale_miglia + x
    WHERE numero_tessera = new.membro;
END
```

**5. Schedule**

Si consideri il seguente schedule  $r1(x)r1(y)w2(y)r3(y)w2(z)r1(z)w2(z)w1(y)r1(z)w3(y)$

- a. Stabilire se è CSR [3 punti];

No

- b. Se passato ad uno scheduler 2PL ci saranno transazioni che vanno in attesa o deadlock? Se sì elencare quali e motivare la risposta [3 punti].

Si ci sono transazioni in attesa e un deadlock. In particolare  $t_2$  è in attesa che  $t_1$  rilasci il  $r\_lock$  su  $y$ ,  $t_1$  è in attesa che  $t_3$  rilasci il  $r\_lock$  su  $y$ , e  $t_3$  è in attesa che  $t_1$  rilasci il  $r\_lock$  su  $y$ . Quindi c'è una situazione di stallo tra le tre transazioni.