

# 28-10-2022

## ESERCITAZINE

Si consideri lo schema Marinaio-Barca-Prenotazione

M(idm, nomem, rating, eta)

B(idb, nomeb, colore)

P(idm, idb, data)

1. Trovare i colori della barca prenotata dal marinaio Marco;
2. Stampare gli id dei marinai che hanno un rating di almeno 8 o che hanno prenotato la barca 103;
3. Trovare il nome dei marinai che **non hanno** prenotato barche rosse;
4. Trovare l'id dei marinai che hanno un'età maggiore di 20 e **che non hanno** prenotato una barca rossa;
5. Trovare il nome dei marinai che hanno prenotato **almeno** due barche;
6. Trovare i nome dei marinai che hanno prenotato **tutte** le barche;
7. Trovare il nome dei marinai che hanno prenotato **tutte le barche di nome "BlueFish"**

1.  $\pi_{colore}(\sigma_{nomem='Marco'}((M) \bowtie (P) \bowtie (B)))$

1. SELECT DISTINCT colore FROM M NATURAL JOIN P NATURAL JOIN B WHERE  
nomem='Marco'

2.  $R_1 = \pi_{idm}(\sigma_{rating \geq 8}(M)) \text{ ----> } R_2 \pi_{idm}(\sigma_{idb=103}(P)) \text{ -----> } R_1 \cup R_2$

2. SELECT idm FROM M WHERE rating >= 8 UNION SELECT idm FROM P WHERE idb=103

3.  $R_1 = \pi_{idm}(P \bowtie \sigma_{colore='rosso'}(B)) \text{ ----> } R_2 = \pi_{idm}/R_1 \text{ ---> } \pi_{nomem}(M \bowtie R_2)$

3. SELECT nomem FROM M WHERE NOT EXISTS (SELECT \* FROM P,B WHERE P.idm=M.idm AND  
P.idb=B.idb AND B.colore='rosso')

4. 3.  $R_1 = \pi_{idm}(P \bowtie \sigma_{colore='rosso'}(B)) \text{ ----> } R_2 = \pi_{idm}/R_1 \text{ ---> } \sigma_{eta > 20}(M) \bowtie R_2$

4. SELECT nomem FROM M WHERE NOT EXISTS (SELECT \* FROM P,B WHERE P.idm=M.idm  
AND P.idb=B.idb AND B.colore='rosso') AND eta > 20

5.  $P_1 = P \text{ -----> } R_1 = \pi_{idm}(P \text{ join}_{\{P.idm=P_1.idm, P.idb>P_1.idb\}} P_1) \text{ ---> } \pi(R_1 \text{ join } M)$

5. SELECT nomem FROM M NATURAL JOIN P GROUP BY idm, nomem HAVING COUNT(\*) >= 2

6.  $R_1 = \pi_{idm,idb}(P) \text{ divisione } \pi_{idb}(B) \text{ --> } \pi_{nomem}(M \text{ join } R_1)$

6. SELECT nomem FROM m WHERE NOT EXISTS (SELECT \* FROM B WHERE NOT EXISTS (SELECT  
\* FROM P WHERE P.idm = M.idm AND P.idb = B.idb))

Dato lo schema

CONTOCORRENTE(id\_conto,saldo,data\_apertura)  
PRODOTTOFINANZIARIO(id\_conto,<sup>id</sup>id\_prodotto,data\_stipula,numero\_rate,id\_contraente)  
PERSONA(id\_persona,nome,cognome,data\_nascita)  
TITOLARECC(id\_persona,ID\_conto)  
TRANSAZIONE(id\_contoIN,id\_contoOUT,data,causale,dare\_avere,importo)  
TRANSAZIONEPRODOTTOFINANZIARIO(id\_conto,data,causale ,importo,id\_prodotto)

1. Identificare le chiavi primarie ed esterne dello schema
2. Scrivere le seguenti query in algebra relazionale.
  - Trovare il conto corrente (cc) aperto in data 11/11/2012 con due intestatari e con saldo minimo
  - Trovare i conti correnti che non hanno prodotti finanziari aperti.
  - Trovare le coppie di persone che hanno esattamente gli stessi cc.
3. Scrivere le seguenti query in SQL
  - Trovare il cc con il saldo più alto
  - Trovare la persona che ha il saldo totale più elevato
  - Trovare il prodotto finanziario che ha più rate

2.1. `SELECT id_conto FROM CONTOCORRENTE WHERE saldo = (SELECT MAX(saldo) FROM CONTOCORRENTE)`

2. oppure potevo fare `SELECT id_conto FROM CONTOCORRENTE WHERE saldo >= ALL (SELECT saldo) FROM CONTOCORRENTE)`

2.2 `SELECT T1.id_persona FROM TITOLARECC T1, CONTOCORRENTE C1 WHERE T1.id_conto=C1.id_conto GROUP BY T1.id_persona HAVING SUM(saldo) = (SELECT MAX(S.totale) FROM (SELECT T.id_persona SUM(saldo) AS totale FROM TITOLARECC T, CONTOCORRENTE C WHERE T.id_conto = C.id_conto GROUP BY T.id_persona) AS S)`

2.3 `SELECT (*) FROM PRODOTTOFINANZIARIO WHERE numero_rate>= ALL (SELECT numero_rate FROM PRODOTTOFINANZIARIO)`