

Dato lo schema

CONTOCORRENTE(id_conto,saldo,data_apertura)
 PRODOTTOFINANZIARIO(id_conto, 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

$R2 = TITOLARECC$

$R3 = TITOLARECC$

$R4 = \pi_{id_conto} (R2 \bowtie_{R2.id_conto=R3.id_conto \wedge R2.id_persona > R3.id_persona} R3)$

$R1 = \pi_{id_conto,saldo} (\sigma_{data_apertura=11/11/12} (CONTOCORRENTE \bowtie R4))$

$R5 = R1$

$R1 - \pi_{R1.id_conto,R1.saldo} (R1 \bowtie_{R1.saldo < R5.saldo} R5)$

- Trovare i conti correnti che non hanno prodotti finanziari aperti

$R1 = \pi_{id_conto} (CONTOCORRENTE)$

$R2 = \pi_{id_conto} (PRODOTTOFINANZIARIO)$

$R1 - R2$

- Trovare le coppie di persone che hanno esattamente gli stessi cc

$R1 = TITOLARECC$

$R2 = TITOLARECC$

$R3 = \pi_{R1.id_persona,R2.id_persona,R1.id_conto} (R1 \bowtie_{R1.id_conto=R2.id_conto \wedge R1.id_persona > R2.id_persona} R2)$

$R4 = \pi_{R1.id_persona,R2.id_persona,R2.id_conto} (R1 \bowtie_{R1.id_conto=R2.id_conto \wedge R1.id_persona > R2.id_persona} R2)$

$R5 = \pi_{R1.id_persona,R2.id_persona,R1.id_conto} (R1 \bowtie_{R1.id_conto <> R2.id_conto \wedge R1.id_persona > R2.id_persona} R2)$
 $- R3$

$R6 = \pi_{R1.id_persona,R2.id_persona,R2.id_conto} (R1 \bowtie_{R1.id_conto <> R2.id_conto \wedge R1.id_persona > R2.id_persona} R2)$
 $- R4$

$\pi_{R1.id_persona,R2.id_persona} (R3) - (\pi_{R1.id_persona,R2.id_persona} (R5) \cup$

$\pi_{R1.id_persona,R2.id_persona} (R6))$

3. Scrivere le seguenti query in SQL

- Trovare il cc con il saldo più alto

```
SELECT id_conto
FROM CONTOCORRENTE CC
WHERE NOT EXISTS (SELECT *
                   FROM CONTOCORRENTE CC1
                   WHERE CC1.saldo > CC.saldo)
```

- Trovare la persona che ha il saldo totale più elevato

```
SELECT id_persona, sum(saldo)
```

```
FROM CONTOCORRENTE NATURAL JOIN TITOLARE CC
GROUP BY id_PERSONA
HAVING sum(saldo) >= ALL
      (SELECT sum(saldo) saldo_tot
       FROM CONTOCORRENTE NATURAL JOIN TITOLARE CC
       GROUP BY id_PERSONA)
```

- Trovare il prodotto finanziario che ha più rate

```
SELECT id_prodotto
FROM PRODOTTOFINANZIARIO PF
WHERE NOT EXISTS (SELECT *
                  FROM PRODOTTOFINANZIARIO PF1
                  WHERE PF1.numero_rate > PF.numero_rate)
```