

Esercizio 1:

Sia dato il seguente schema relazionale relativo alle filiali di una banca presenti sul territorio nazionale:

Filiale(CodiceFiliale, Città, Direttore, TotaleDepositi); *chiavi candidate*: direttore

SiTrovaIn(Città, ~~Regione~~);

CCclienti(CodiceFiscale, *Filiale*, NumeroCC).

Si assuma che ogni filiale sia identificata univocamente dal suo codice e sia caratterizzata dalla città in cui si trova, dal codice fiscale del direttore e dall'ammontare complessivo di denaro depositato presso di essa (attributo *TotaleDepositi*). Si assuma che filiali diverse abbiano direttori diversi. Si assuma anche che la banca possa avere più di una filiale in una data città. Ogni città sia identificata univocamente dal suo nome. Ogni cliente sia identificato unicamente dal suo codice fiscale e possa possedere uno o più conti correnti presso una o più filiali. Ogni conto corrente sia identificato univocamente dal suo numero, sia associato ad una sola filiale e abbia un unico proprietario.

Definire preliminarmente le chiavi primarie, le eventuali altre chiavi candidate e, se ve ne sono, le chiavi esterne delle relazioni date. Successivamente, formulare opportune interrogazioni in algebra relazionale che permettano di determinare (senza usare l'operatore di divisione e usando solo se necessario le funzioni aggregate):

- i clienti che possiedono conti correnti solo in filiali della banca con sede in città della regione Veneto;
- per ogni città con almeno 3 filiali, il numero di filiali con un valore dell'attributo *TotaleDepositi* maggiore di 50000000 di euro;
- i clienti che hanno un conto corrente in tutte le filiali in cui ha un conto corrente il cliente MLNGVN11S11H333P.

a. $CANDIDATI \leftarrow \pi_{CODICEFISC} \left(\left(\sigma_{REGIONE = 'VENETO'} (FILIALE \bowtie SITROVA IN) \right) \bowtie_{FILIALE = CODICEFILIALE} CCCLIENTE \right)$

$NO_GOOD \leftarrow \pi_{CODICEFISC} \left(\left(\sigma_{REGIONE \neq 'VENETO'} (FILIALE \bowtie SITROVA IN) \right) \bowtie_{FILIALE = CODICEFILIALE} CCCLIENTE \right)$

$S \leftarrow CANDIDATI - NO_GOOD$

b. ALMENO 3 $\rightarrow A \times A \times A$

$FILIALE1 (...1,...1,...) \leftarrow FILIALE$

$FILIALE2 (...2,...2,...) \leftarrow FILIALE$

$S \leftarrow \pi_{CITTA} \int_{COUNT(*)} \left(\sigma_{\begin{array}{l} TOTALEDEPOSITI = 50000000 \dots \left(\sigma_{CODICEFILIALE < CODICEFILIALE1} (FILIALE \times FILIALE1 \times FILIALE2) \right) \\ OR \quad TOTALEDEPOSITI1 = \dots \quad AND \quad CITTA = CITTA1 \quad AND \\ OR \quad TOTALEDEPOSITI2 = \dots \quad CODICEFILIALE1 < CODICEFILIALE2 \\ \quad \quad \quad AND \quad CITTA1 = CITTA2 \end{array}} \right)$

c. $CONTCLIENTE \leftarrow \pi_{FILIALE} \left(\sigma_{CODICEFISCALE = 'MLN\dots'} (CCCLIENTI) \right)$

$STATO_DI_FATTO \leftarrow \pi_{CODFISC, FILIALE} (CCCLIENTI)$

$REQUISITI \leftarrow \pi_{CODFISC} (CCCLIENTI) \times CONTCLIENTE$

$NO_GOOD \leftarrow REQUISITI - STATO_DI_FATTO$

$S \leftarrow \pi_{CODFISC} (CCCLIENTE) - \pi_{CODFISC} (NO_GOOD)$

Esercizio 4:

Si considerino le due transazioni seguenti:

```
start transaction; -- T1
  select qta from Articolo where id = 123;
  select qta from Articolo where id = 123;
commit;

start transaction; -- T2
  update Articolo set qta = qta - 1 where id = 123;
commit;
```

Si supponga che T1 e T2 siano sottomesse al sistema simultaneamente. Si spieghi, giustificando la risposta, quali risultati possono produrre le due `select`, nei casi in cui le due transazioni siano eseguite:

1. nel livello `serializable`;
2. nel livello `repeatable read`;
3. nel livello `read committed`;
4. nel livello `read uncommitted`.

nel caso T1 e T2 siano sequenziali (uno dei due ordini) allora le due `select` hanno risultati consistenti per tutti i livelli.
L'unica anomalia è LETTURA INCONSISTENTE ed è possibile solo quando:

T1: `select qta from Articolo...`
T2: `update Articolo set qta = ...`
T1: `select qta from Articolo...`

nel caso di livello: - `READ UNCOMMITTED`: non rilevata → legge valori diversi

- `READ COMMITTED`: non rilevata → legge valori diversi

- `REPEATABLE READ`: la lettura incons. non può avvenire perché 2PL stretto in lettura/scrittura → una delle due transaz. viene messa in attesa finché l'altra non fa commit

se non c'è lo SNAPSHOT

Esercizio 5:

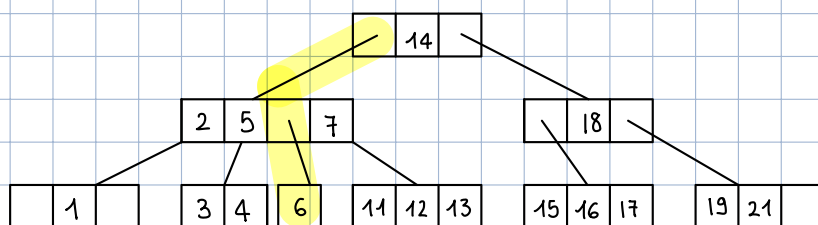
Dato il seguente insieme di chiavi:

2, 1, 18, 17, 21, 14, 13, 7, 6, 15, 12, 11, 3, 19, 5, 4, 16,

mostrare il **B-albero**, con ordine dei nodi interni $p = 4$, ottenuto inserendo un elemento dopo l'altro nell'ordine dato (riportando la sequenza di alberi generata dal processo di inserimento).

Successivamente, si identifichino i nodi del B-albero visitati nella ricerca di rispettivamente: (i) il record contraddistinto dal valore 9 e (ii) il record contraddistinto dal valore 6.

q , ovvero il # puntatori per nodo è: $\lceil \frac{p}{2} \rceil \leq q \leq p$, quindi #valori è $1 \leq \# \leq 3$



cammino 6: nodi 14, 2-5-7, 6

cammino 9 (anche se non c'è): nodi 14, 2-5-7, 11-12-13