

## Basi di Dati – Corso B – Appello: 11 settembre 2018

Cognome, Nome \_\_\_\_\_

Matricola \_\_\_\_\_

### Domanda 1.

Considerare la base dati “Cinguettii” contenente dati relativi ad un sito di microblogging con il seguente schema

UTENTE(IDUtente, Nome, Città)

SEGUACE(IDUtente1, IDUtente2, DataS)

CINGUETTIO(IDCinguettio, IDUtente, Testo, DataC, Ricinguettio)

Ogni tupla di SEGUACE rappresenta la seguente informazione: l’utente identificato da IDUtente1 inizia a seguire l’utente identificato da IDUtente2 nella data definita da DataS.

Ogni tupla di CINGUETTIO rappresenta un “micropost”, pubblicato nella data definita da DataC, dall’utente identificato da IDUtente. Se un micropost A cita un altro micropost B, l’attributo Ricinguettio contiene l’identificativo del micropost B, altrimenti è nullo.

Inoltre, valgono i seguenti vincoli di integrità referenziale: IDUtente1 e IDUtente2 in SEGUACE referenziano UTENTE. IDUtente in CINGUETTIO referencia UTENTE. Ricinguettio in CINGUETTIO referencia CINGUETTIO.

Produrre un’espressione in SQL che esprima la domanda:

**“Elencare gli utenti (ID, nome e città) che hanno iniziato a seguire un altro utente dopo averne ricinguettato almeno un micropost”**

### Soluzione 1

```
SELECT *
FROM utente
WHERE IdUtente in
    (SELECT S.IdUtente1
     FROM seguace S JOIN cinguettio C1 ON (S.IdUtente1=C1.IdUtente)
     JOIN cinguettio C2 ON (C1.IdUtente2=C2.IdUtente AND
                          C1.Ricinguettio=C2.IdCinguettio)
     WHERE ( C1.DataC < S.DataS )
    );
```

Altre soluzioni sono possibili.

### Domanda 2.

Facendo riferimento alla base dati “Cinguettii”:

A. Scrivere una espressione in algebra relazionale e in calcolo relazionale che esprima la seguente domanda:

**Elencare gli utenti che seguono solo utenti che non hanno mai pubblicato (né ricinguettato) alcun post.**

B. Dire se è possibile rispondere con una singola espressione in algebra relazionale (ma non è necessario fornire l’eventuale espressione algebrica) alla seguente domanda: **dire se esiste una catena di seguaci che collegano l’utente “@Macron” all’utente “@Tsipras” o viceversa**. Motivare la risposta.

## Soluzione 2

A. Algebra relazionale:

$$\prod_{IDUtente1} (SEGUACE) - \prod_{IDUtente1} (SEGUACE \bowtie_{IDUtente2=IDUtente1} CINGUETTIO)$$

Calcolo relazionale:

$$\{u.* \mid u(UTENTE) \mid \exists s(SEGUACE)(s.IDUtente1=u.IDUtente) \wedge \\ \forall s'(SEGUACE)(s'.IDUtente1 = u.IDUtente) \Rightarrow \neg \exists c(CINGUETTIO)(c.IDUtente=s'.IDUtente2)\}$$

B. Non è possibile in quanto l'algebra relazionale tradizionale non ammette la ricorsione né la chiusura transitiva.

## Domanda 3.

Data una relazione  $R(A,B,C,D,E,F,G)$  e l'insieme delle dipendenze funzionali

$$F = \{AB \rightarrow EF, B \rightarrow G, A \rightarrow CDG, C \rightarrow B, E \rightarrow F\}$$

- A. trovare le chiavi della relazione R
- B. dare la definizione di insieme di copertura minimale
- C. calcolare una copertura minimale di F
- D. dire se R è in 3FN e se non lo è scomporla

## Soluzione 3

- A. L'unica chiave della relazione R è A, infatti  $A^+ = \{A|CDG|B|EF\}$ .
- B. Si vedano gli appunti/slide.
- C. Forma canonica:

$$F' = \{AB \rightarrow E, AB \rightarrow F, B \rightarrow G, A \rightarrow C, A \rightarrow D, A \rightarrow G, C \rightarrow B, E \rightarrow F\}$$

Attributi estranei:

l'unico attributo estraneo è B (per transitività da  $A \rightarrow C$  e  $C \rightarrow B$ . Quindi:

$$F' = \{A \rightarrow E, A \rightarrow F, B \rightarrow G, A \rightarrow C, A \rightarrow D, A \rightarrow G, C \rightarrow B, E \rightarrow F\}$$

Dipendenze ridondanti:

$A \rightarrow F$  è ridondante per transitività ( $A \rightarrow E$  ed  $E \rightarrow F$ )

$A \rightarrow G$  è ridondante per transitività ( $A \rightarrow C$ ,  $C \rightarrow B$  e  $B \rightarrow G$ )

Quindi, la copertura minimale di F è:

$$F' = \{A \rightarrow E, B \rightarrow G, A \rightarrow C, A \rightarrow D, C \rightarrow B, E \rightarrow F\}$$

- D. R non è in 3FN (ad esempio  $B \rightarrow G$  non è né d.f. superchiave né d.f. con attributi primi).

Partiamo dal risultato dell'esercizio C, accorpando le d.f. con stesso antecedente:

$$F' = \{A \rightarrow CDE, B \rightarrow G, C \rightarrow B, E \rightarrow F\}$$

E otteniamo le relazioni:

$R1(\underline{A}CDE)$ ,  $R2(\underline{B}G)$ ,  $R3(\underline{C}B)$  ed  $R4(\underline{E}F)$ .

**Domanda 4.**

- A.** A proposito di gestione della concorrenza descrivere il protocollo di lock a due fasi (2PL) e quello di lock a due fasi stretto.
- B.** La storia  $S = r1(x), r1(y), r2(x), w3(z), w1(y), r2(y), w2(x)$  è compatibile con 2PL stretto? Motivare la risposta.

**Soluzione 4**

Per la prima parte della domanda, si rimanda alle slide o agli appunti.

La storia  $S1$  è compatibile con il 2PL. Infatti, una possibile realizzazione, con i relativi lock e unlock, è la seguente:

$S1 =$  LS1(x), r1(x), LS1(y), r1(y), LS2(x), r2(x), LX3(z), w3(z), UN3(z), LX1(y), w1(y), UN1(x), UN1(y), LS2(y), r2(y), LX2(x), w2(x), UN2(y), UN2(x)

**Domanda 5.**

Si consideri un file di log L con il seguente contenuto in seguito ad un crash:

```
<T1, START>;
<T2, START>;
<T1, BS(t1[A], 5), AS(t1[A], 10)>;
<T2, BS(t2[B], 3), AS(t2[B], 5)>;
<T3, START>;
<T2, COMMIT>;
-- checkpoint --;
<T3, BS(t3[C], 3), AS(t3[C], 5)>;
<T3, ABORT>
```

**crash!**

- A.** Quali politiche di gestione del buffer adotta il DBMS che ha generato il log?
- B.** Indicare il contenuto del record di checkpoint.
- C.** Descrivere l'algoritmo di ripristino corrispondente alle politiche del buffer descritte nella risposta alla domanda A e mostrarne l'esecuzione sul log L.

**Soluzione 5**

**A.** Nel log troviamo sia i before-state che gli after-state. Si può quindi concludere che il DBMS adotta le politiche di gestione del buffer steal/no flush.

**B.** Il record di checkpoint conterrà la lista delle transazioni non terminate (quindi, T1 e T3) con i relativi puntatori ai rispettivi  $\langle Ti, START \rangle$ . Contiene inoltre l'OK riguardante la buona riuscita delle operazioni previste dal checkpoint, quindi:  $[(T1, p1); (T3, p3); OK]$

**C.** Per la descrizione dell'algoritmo si rimanda agli appunti/slide. Si noti che le liste LA e LC sono ricostruite a partire dal record di checkpoint.

La lista delle transazioni attive al momento del crash è  $LA = \{T1\}$  (T3 è in abort). La lista delle transazioni in commit prima del crash è vuota, in quanto le modifiche effettuate dalla transazione T2 sono state rese persistenti dalla procedura di checkpoint.

Viene effettuato quindi l'UNDO di T1: viene ripristinato a '5' il valore dell'attributo A della tupla t1 (si effettua contestualmente anche un FORCE della pagina).