

Cognome _____ Nome _____ Matricola _____

Basi di Dati

21-01-2014

PROVA SCRITTA

1 (6 pts)	2 (6 pts)	3 (4 pts)	4 (2 pts)	5 (4 pts)	6 (4 pts)	7 (4 pts)	8 (2 pts)	Somma (32 pts)

Il compito ha la durata di 3 ore, per la compilazione attenersi alle seguenti istruzioni:

- **Nome e cognome devono essere scritti in stampatello sul foglio d'esame e su ogni foglio protocollo da correggere.**
- Il testo d'esame deve essere riconsegnato anche in caso di ritiro dall'esame.
- Riportare le soluzioni sul foglio protocollo in modo chiaro e il numero/lettera dell'esercizio evidente e perfettamente leggibile.
- Non sono ammessi appunti o altro materiale, a chiunque copia, comunichi per qualsiasi ragione con i colleghi, o venga trovato ad usare qualsiasi tipo di note o appunti verrà ritirato l'esame.
- **L'esame si ritiene superato se si raggiunge il punteggio di 18 punti su 32**
- La lode può essere assegnata solo con un punteggio superiore a 30 punti.

1. (6 punti) Si vuole creare una base di dati per la raccolta di micropost da un social network. Ogni micropost ha un testo lungo al più 160 caratteri, se ne registra la data di creazione, e proviene da un utente detto autore. In ogni preciso istante, un utente può pubblicare un unico micropost. Un utente è identificato da un progressivo numerico, possiede uno username lungo al massimo 50 caratteri, e se ne salva la data di registrazione. Un micropost può essere scritto come risposta al micropost di un altro utente. Inoltre, un utente può ripubblicare il micropost di un altro utente. Un micropost ripubblicato, detto citazione, è una copia del micropost originale, pubblicato in risposta all'originale, dove l'autore viene sostituito con l'utente che pubblica la citazione, la data di pubblicazione viene sostituita dalla data in cui avviene la ripubblicazione, ma di cui si conservano le informazioni sull'autore e la data di pubblicazione originale. Se il micropost originale viene eliminato le citazioni e le risposte non vengono eliminate. Le date sono codificate come UNIX timestamp

Si produca:

- (a) lo schema ER della base di dati
- (b) lo schema relazionale della base di dati
- (c) la query SQL per creare la tabella **Micropost**
- (d) e le query necessarie per inserire una tupla **Micropost**.

Si espliciti ogni assunzione necessaria.

2. (6 punti) Una diversa base di dati ha il seguente schema relazionale utile per l'esercizio:

Utente (nome, mese_reg, anno_reg, lingua)

Segue (nome_segue, nome_seguito)

Dell'utente si registra solo il nome utente come chiave, se ne salva separatamente il mese (1, 2, ..., 12) e l'anno di registrazione, oltre alla lingua dell'utente (EN, IT, ...). Un utente può inoltre 'seguire' i micropost di un altro utente, e si registra solo il nome di chi segue e di chi viene seguito.

Esprimere in algebra relazionale (algebra relazione estesa solo se necessario) e SQL le seguenti:

Non sono ammesse viste o tabelle temporanee, ma sono ammesse sotto-query. Le query in algebra relazionale devono essere ottimizzate

- (a) Ritornare **nome**, **anno_reg** e **lingua** per gli iscritti di gennaio degli anni 2003, 2004, 2005, 2006 e 2010
- (b) Ritornare in ordine alfabetico decrescente l'elenco dei nomi degli utenti che seguono utenti iscritti dopo di loro;
- (c) Si assuma di avere il risultato della query precedente in una vista con nome **v1**. Ritornare il numero medio di utenti seguiti dagli utenti iscritti negli anni 2010, 2011, 2012 e che non seguono nessuno iscritto dopo di loro ; deve essere elencata la media di utenti per ogni anno richiesto

3. (4 punti) Utilizzando lo schema dell'esercizio precedente **esprimere in SQL le seguenti query:**

- (a) Eliminare ogni utente iscritto prima del 2005 che non segue e non è seguito da nessuno.
- (b) Ritornare il **nome** degli utenti che seguono gli utenti seguiti dal maggior numero di persone, e gli utenti che seguono utenti che lo seguono, e che seguono utenti che seguono utenti che lo seguono e così via.

4. (2 punti) Considerare lo schema $R(A, B, C)$ con le seguenti istanze:

A	B	C
4	NULL	8
7	2	NULL
NULL	2	NULL
4	NULL	4
7	7	NULL

Definire i risultati delle seguenti query:

- (a) `SELECT A, MIN(B) FROM R GROUP BY A`
- (b) `SELECT COUNT(*), COUNT(A) FROM R`
- (c) `SELECT A, B FROM R WHERE A=C`
- (d) `SELECT B, C FROM R WHERE A<>C`

5. (4 punti) Dati

$R(a,b)$ $T(R)=100$ $V(R,a)=5$ $V(R,b)=10$	$S(b,c)$ $T(S)=200$ $V(S,b)=10$ $V(S,c)=5$
$W(c,d)$ $T(W)=200$ $V(W,c)=5$ $V(W,d)=5$	$U(d,e)$ $T(U)=300$ $V(U,d)=100$ $V(U,e)=50$

- (a) Motivare il modo migliore per valutare $\bowtie (R, S, W, U)$
- (b) Scrivere due espressioni nell'algebra relazionale estesa della seguente query
`SELECT DISTINCT d FROM W, U WHERE W.d = U.d AND U.e = 11`
 Motivare quale delle due presentate sia la migliore

6. (4 punti) Descrivere (in modo informale) gli algoritmi, ed esplicitando le assunzioni necessarie, per calcolare $R \bowtie_B S$ con $B(S) > B(R)$

- (a) Nel caso $B(R) < M$
- (b) Nel caso $B(R) > M$

7. (4 punti) Per ciascuno dei seguenti programmi, indicare se è conflict serializable e, se sì, scrivere una corretta esecuzione seriale equivalente.

- (a) `r2(C); r1(A); w2(B); r1(D); w2(A); w3(B); w1(C); r3(D)`
- (b) `r1(A); w2(B); r3(D); r3(D); w2(A); r3(B); w2(C); r1(D)`

Usare shared, exclusive, e update locks. Descrivere cosa fa ogni parte di un scheduler 2PL con questa sequenza di operazioni.

- (c) `r1(B); r3(A); w1(B); w1(B); w2(B); r2(A); w3(A)`

8. (2 punti) Si consideri il seguente schema relazionale $R(ABCDEF)$ e FD
 $A \rightarrow B$, $CD \rightarrow A$, $BC \rightarrow D$, $AE \rightarrow F$, $CD \rightarrow E$

- (a) Trovare la/le chiave/i per lo schema
- (b) e convertire lo schema in BCNF se non lo fosse