Questionario 2^ PARZIALE di Basi di Dati

**25 Febbraio 2010**

**(Prof.ssa S. Costantini)**

**COGNOME....................…...........NOME........……...…………......MATR..………...**

# CRITERI DI VALUTAZIONE DELLA PROVA

Ogni parte del questionario del Corso di Basi di Dati è composto nel modo seguente:

* 6 domande chiuse, da 2 punti l’una (totale 12 punti), 0 punti quelle senza risposta;
* 2 domande aperte, obbligatorie, da 4 punti (totale 8 punti), -2 punti per ogni domanda non svolta o gravemente insufficiente o incongruente;
* 2 esercizi, obbligatori, da 5 punti l’uno (totale 10 punti), -3 punti per ogni esercizio non svolto o gravemente insufficiente;

Nella domanda aperta, indicata con A, vengono valutati sia il contenuto che la forma della risposta, con riferimento alla precisione terminologica ed alla completezza. Non sono ammesse risposte generiche: si richiede una risposta precisa e se necessario formale.

Ogni esercizio, indicato con E, deve essere corredato dallo svolgimento e corrisponde ad un punteggio che dipende della correttezza e completezza dello svolgimento. Gli esercizi dove venga fornito soltanto il risultato saranno considerati non svolti.

E’ obbligatorio riportare le soluzioni negli appositi spazi dei fogli allegati (utilizzando eventualmente anche il retro dei fogli). Non si accettano compiti consegnati in brutta copia. Si correggono solo compiti svolti in calligrafia comprensibile.

**Tempo a disposizione: 1 ora e mezza**

1)Si consideri un file di log che contenga la seguente sequenza di record:

DUMP,B(T1),B(T2),D(T1,O1,B1),B(T3),U(T3,O2,B2,A2),B(T4),C(T3),B(T5),I(T2,O3,A3),A(T5)CK(…)

Elencare le transazioni attive al momento del checkpoint (record CK): *T1, T2, T4. Infatti, T3 è iniziata e terminata con commit, T5 è iniziata e terminata con abort.*

2)Data la dipendenza funzionale CF->Nome DataNascita che vale sulla relazione Persona scrivere in notazione matematica cosa significa questa dipendenza per le tuple di questa relazione (utilizzare quantificatori, connettivi e implicazione)

*∀ t1, t2 ∈ Persona t1[CF] = t2[CF] ⇒ t1[Nome DataNascita] = t2[Nome DataNascita]*

3) L'analisi delle ridondanze viene effettuata in fase di:

1. progettazione concettuale;
2. progettazione logica; **X**
3. normalizzazione.

4)Data la tassonomia dei metodi VSR, CSR, 2PL e TS, marcare l’affermazione vera tra le seguenti:

* VSR ⊆ CSR
* CSR ⊆ VSR
* CSR ⊂ VSR **X**

5)Nel Two Phase Locking (2PL), sia **r\_locked** lo stato di una generica risorsa. Una richiesta di locking in scrittura (**w\_lock**) su tale risorsa implica:

* La richiesta di w\_lock viene negata, tuttavia lo stato della risorsa diviene w\_locked
* La richiesta di w\_lock viene negata, lo stato della risorsa rimane r\_locked **X**
* La richiesta di w\_lock viene accettata, lo stato della risorsa diviene w\_locked

6)È noto che determinare la view-equivalenza di due schedule è un problema con complessità lineare, tuttavia nel caso generale la view-equivalenza è non applicabile nella pratica essendo un problema NP-difficile. Questa affermazione è vera, poiché dato un generico schedule S:

* È necessario determinare uno schedule CSR che sia anche VSR
* È necessario permutare in tutti i modi possibili le transazioni di S
* È necessario confrontare tale schedule con tutti gli schedule seriali ottenuti permutando in tutti i modi possibili le transazioni di S. **X**

7) **A** Si dica in sintesi come si eliminano le gerarchie di generalizzazione in fase di ristrutturazione degli schemi E-R, illustrando le soluzioni base e ibride e spiegando per ciascuna soluzione quando si possa ritenere consigliabile.

*Le gerarchie di generalizzazione vanno eliminate perché nel modello relazionale non c’è un costrutto ad esse corrispondenti. La prima soluzione per eliminarle consiste nel sostituire ad ogni freccia figlia-padre una nuova associazione, diciamo per comodità IS\_A, di cardinalità (0,N) da parte del padre (che può avere un numero qualunque di entità figlie) ed 1:1 da parte della figlia, che è identificata esternamente mediante questa associazione. La seconda soluzione prevede di accorpare le figlie nel padre, ottenendo un’unica entità che ha come attributi l’unione degli attributi di padre e figlie, più un attributo di tipo per classificare ciascuna occorrenza. Questa soluzione è opportuna se le figlie hanno pochi attributi (poiché per ciascuna occorrenza gli attributi delle figlie a cui tale occorrenza non appartiene restano nulli) e poche associazioni, che perdono la cardinalità (che per il padre è sempre opzionale) La terza soluzione consiste nell’accorpare il padre in ciascuna delle figlie, replicandone gli attributi. Qui vale il viceversa, ossia è il padre che dovrebbe avere pochi attributi ed associazioni. Sono possibili soluzioni ibride, accorpando nel padre solo parte delle figlie, dove quelle con molti attributi ed associazioni restano, legate al padre da un IS\_A.*

1. **A** Parlare in breve della ripartenza a caldo e a freddo in seguito ad un guasto.

*La ripartenza a caldo avviene dopo un guasto transitorio, come segue: Si cerca ultimo checkpoint nel LOG; Si decidono transazioni da rifare (insieme di REDO) e disfare (insieme di UNDO), dove l’insieme di UNDO contiene l’elenco delle transazioni attive al checkpoint, mentre l’insieme di REDO è inizialmente vuoto. Per creare REDO e completare UNDO, si scorre il LOG: se si trova un record di B(T), la transazione T viene inserita in UNDO; se si trova un record, C(T), la transazione T viene tolta da UNDO e inserita in REDO. Infine, si applicano UNDO e REDO. La ripresa a freddo è necessaria dopo un guasto più serio, come segue: si usa l’ultimo DUMP per ripristinare uno stato stabile della BD; si ripercorre il LOG rifacendo tutte le azioni successive al DUMP; si fa ripresa a caldo dall’ultimo checkpoint.*

1. **E** Si consideri il seguente schema relazionale che annota i noleggi di un insieme di CD ai clienti, ciascuno dei quali noleggia per un certo numero di giorni. Un cliente può noleggiare vari CD nello stesso giorno e ovviamente un CD non può essere noleggiato da più di un cliente. Si assume che alla restituzione di un CD la relativa tupla di noleggio venga rimossa.

CDNoleggi(Codice,TitoloGenere,DataNoleggio,CodiceCliente,NomeCliente,Giorni)

Si indichino esplicitamente una chiave primaria e le dipendenze funzionali significative che è possibile identificare. Si decomponga la tabella in forma normale di Boyce-Codd spiegando perché la decomposizione proposta è senza perdita.

*Dipendenze*

Codice 🡪 Titolo, Genere

CodiceCliente 🡪 NomeCliente

Codice,CodiceCliente,DataNoleggio 🡪 Giorni

*Decomposizione*

CD(Codice,Titolo, Genere)

Cliente(CodiceCliente,NomeCliente)

Noleggio(Codice,CodiceCliente,DataNoleggio,Giorni)

*La decomposizione è senza perdita perché la condizione sufficiente è rispettata, in quanto l’attributo Codice che è comune a CD a Noleggio è chiave di CD, e l’attributo CodiceCliente che è in comune fra Cliente e Noleggio è chiave di Cliente.*

10) Dato il seguente schedule *S*:

**S**: 

Determinare se S è conflict-serializzabile (CSR). Nel caso S sia CSR esplicitare almeno uno schema seriale equivalente ad esso. In caso contrario (S CSR), indicare le operazioni da eliminare, in quantità necessaria minima, per rendere lo schema CSR.



|  |
| --- |
| Soluzione: |
| Determiniamo i conflitti:  C(x) = { ( ; ), ( ; ), ( ; ), ( ; ) }  C(y) = { ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; )}  C(z) = { ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; )}  Adesso costruiamo il grafo:  Il grafo è aciclico ⟶ **S** è conflict-serializzabile.  Uno dei possibili schedule seriali equivalenti è:  : |