Questionario TOTALE di Basi di Dati e Sistemi Informativi

**9 Febbraio 2011**

**(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 45 minuti**

1.L’espressione (σB<0 (ΠY (σA>20 (*R*)) è equivalente a ΠY (σA>20 AND B<0 (*R*)) se (spiegare):

**l’attributo B fa parte dell’insieme di attributi Y**

2.Una dipendenza funzionale *Y* → Zsu uno schema *R(X)* dove *Y* è la chiave *degenera* nel vincolo di chiave se: **(c)**

a) *Y ∩ Z = X*;

b) *Y ∪ Z ⊆ X*;

c) *Y ∪ Z = X*.

3.In un Data Wharehouse i dati sono organizzati in: **(b)**

1. tabelle relazionali;
2. tabelle multidimensionali;
3. Data Mart;
4. tabelle “market basket”.

4.Dire come viene considerata l’assenza di informazione nei due approcci seguenti.

* CWA (Closed World Assumption):**informazione negativa**
* OWA (Open World Assumption): **informazione sconosciuta**

5.Il **CAP theorem**, o teorema di **Brewer**, afferma che è **impossibile**, per un sistema distribuito, garantire al tempo stesso: **(iii)**

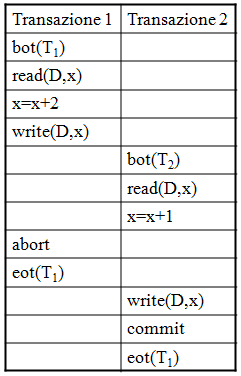
(i) due tra le tre proprietà di [*Consistency*](http://en.wikipedia.org/wiki/Consistency_(database_systems)) , [*Availability*](http://en.wikipedia.org/wiki/Availability), [*Partition Tolerance*](http://en.wikipedia.org/w/index.php?title=Partition_Tolerance&action=edit&redlink=1) ;

(ii) tutte e tre le proprietà di [*Consistency*](http://en.wikipedia.org/wiki/Consistency_(database_systems)) , [*Availability*](http://en.wikipedia.org/wiki/Availability), [*Partition Tolerance*](http://en.wikipedia.org/w/index.php?title=Partition_Tolerance&action=edit&redlink=1) ;

(iii) almeno una delle tre proprietà di [*Consistency*](http://en.wikipedia.org/wiki/Consistency_(database_systems)) , [*Availability*](http://en.wikipedia.org/wiki/Availability), [*Partition Tolerance*](http://en.wikipedia.org/w/index.php?title=Partition_Tolerance&action=edit&redlink=1)

.

6.Date le transazioni t1 e t2, la cui dinamica di esecuzione è riportata nella tabella, sia x=5 il valore all’istante iniziale, ovvero prima dell’inizio dell’esecuzione di t1 e t2.

Dire quale delle seguenti condizioni è valida dopo l’esecuzione delle transazioni t1 e t2: **(a)**

1. Valore finale X=8. Il valore è tuttavia non corretto poiché si è verificato un problema di lettura sporca. Valore corretto X=6
2. Valore finale X=6. Non si verifica alcuna anomalia. Il Il valore finale corrisponde al valore corretto.
3. Valore finale X=7. Il valore tuttavia non è corretto poiché si è verificato un problema di aggiornamento fantasma. Valore corretto X=6.
4. Il valore finale di X non è decidibile, poiché si verifica il problema degli abort in cascata.
5. Valore finale e corretto X=6. Si verifica l’anomalia di aggiornamento fantasma, ma tale anomalia non influisce sulla correttezza.

7.**A** Parlare della frammentazione orizzontale e verticale dei dati in un database distribuito.

*La replica e il partizionamento (o frammentazione) dei dati sono spesso utilizzati (anche insieme) nei database distribuiti per assegnare una parte dei dati a ciascun server.*

*Con il partizionamento orizzontale, il database è suddiviso in due o più database di dimensioni inferiori con lo stesso schema e la stessa struttura, ma con un numero inferiore di righe in ogni tabella. Il partizionamento verticale consiste nel suddividere una o più tabelle in tabelle più piccole con lo stesso numero di righe, ma ciascuna contenente un sottoinsieme di colonne dell'originale. Le interrogazioni devono essere riformulate in termini della frammentazione effettuata, sostituendo al posto della tabella frammentata la sua definizione in termini dei frammenti (come si fa per le viste). Le interrogazioni su frammentazione orizzontale possono richiedere una unione di vari frammenti (se i dati coinvolti non si trovano tutti nel frammento locale) Le interrogazioni su frammentazioni verticali richiedono un join di vari frammenti, il che implica che ciascun frammento deve contenere una chiave comune agli altri mediante la quale effettuare il join (ad esempio, frammentando una tupla relativa ad uno studente ogni frammento dovrà contenere una chiave, tipicamente la matricola).*

8.**A** Descrivere il **2PL strict** in particolare evidenziando la differenza con il **2PL** in relazione alla gestione delle anomalie.

*Nel 2PL è stata introdotta ed è valida l’ipotesi di commit-proiezione, il che rende poco utile tale paradigma nel contesto pratico. Il vincolo aggiuntivo, definito nel complesso come 2PL strict, o locking a due fasi stretto, asserisce che, i lock di una transazione possono essere rilasciati solo dopo aver correttamente effettuato le operazioni di commit/abort.*

*Nella pratica ciò implica che i lock vengono rilasciati solo al termine della transazione, dopo che ciascun dato è stato portato nel suo stato finale. Tramite tale restrizione viene gestita l’anomalia di lettura sporca, possibile nel 2PL.*

1. 9**. E** Si consideri una Base di Dati contenente le tabelle definite dal seguente schema: STUDENTE(Nome,Matr,Media)

ESAME(MatrS:,CodC,Data,Voto)

dove Matr e Media sono la matricola e la media dello studente a nome Nome e dove MatrS è la matricola dello studente che ha sostenuto un dato esame, CodC il codice del relativo Corso, e Data e Voto sono la data in cui è stato sostenuto l’esame e il voto riportato. Definire in Calcolo Relazionale su Tuple un’espressione che identifichi gli studenti che: (i) hanno certamente sostenuto almeno un esame; (ii) che in tutti gli esami sostenuti non hanno mai riportato un voto maggiore o uguale a 27 (non si parla di media, ma di singoli voti); (iii) che però hanno sostenuto l’esame relativo al Corso con codice I203.

{s.\* | s(STUDENTE) | (∃ e1(ESAME) : s.Matr = e1.CodS ∧ e1.CodC = ‘I203’ ) ∧

(~∃ e2(ESAME) : s.Matr = e2.CodS ∧ e2.Voto > 26)}

10.Dato il seguente schedule *S*:

**S**: 

Determinare se S è view-serializzabile

*Per determinare la view-serializzabilità, in questo caso la strada più conveniente da percorrere è sfruttare la tassonomia delle classi di serializzabilità, ed in particolare l’inclusione stretta della classe CSR in VSR, quindi se determiniamo che lo schedule è CSR, allora possiamo affermare che è VSR.*

*Determiniamo i conflitti:*

*C(x) = { ( ; ), ( ; ), ( ; ), ( ; ) }*

*C(y) = { ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; )}*

*C(z) = { ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; ), ( ; )}*

*Adesso costruiamo il grafo:*

*Il grafo è aciclico ⟶* ***S*** *è conflict-serializzabile.*

*Possiamo allora concludere che* ***S*** *è anche view-serializzabile.*