Compito di Basi di dati

16 febbraio 2015

Esercizio 1:

Sia dato il seguente schema relazionale:

ARTISTA(CodiceArtista, Nome, Cognome, CittàDiNascita, AnnoDiNascita);

PALAZZO(Nome, Città,AnnoDiCostruzione);

MOSTRA(Artista, Palazzo, Anno);

SI_TROVA_IN(Città, Nazione).

Si assuma che ogni artista sia individuato univocamente da un codice identificativo e sia caratterizzato da nome, cognome, città in cui è nato e anno di nascita. Si assuma, inoltre, che ogni palazzo sia identificato univocamente dal suo nome e sia caratterizzato dalla città in cui si trova e dall'anno di costruzione. Si assuma anche in una città vi possano essere più palazzi utilizzati per mostre. Attraverso la relazione MOSTRA si vuol tener traccia dei palazzi in cui un artista ha esposto le proprie opere. Si assuma che un artista possa organizzare più mostre in uno stesso palazzo, ma solo in anni diversi. Si assuma, infine, che ogni città sia identificata univocamente dal suo nome.

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 e quando necessario le funzioni aggregate):

(i) gli artisti che hanno fatto non più di due mostre in Germania;

(ii) per ogni città con due o più palazzi per esposizioni, determinare il numero di mostre organizzate nell'anno 2014;

(iii) le coppie di artisti (x,y) tali che esistano un palazzo deve il primo ha fatto una mostra e il secondo no e un palazzo dove il secondo ha fatto una mostra e il primo no.

Esercizio 2:

Con riferimento all'Esercizio 1, formulare opportune interrogazioni in SQL che permettano di determinare quanto richiesto (senza usare l'operatore CONTAINS e usando solo se e quando necessario le funzioni aggregate).

Esercizio 3:

Si vuole progettare una base di dati per la gestione di un giardino botanico. Ogni genere di pianta presente nel giardino sia identificato univocamente dal suo nome (ad esempio, ciclamino) ed sia caratterizzato dalla famiglia di appartenenza (ad esempio, il ciclamino appartiene alla famiglia delle primulaceae). Si assuma che vi possano essere più piante dello stesso genere. Ogni pianta di un certo genere presente nel giardino sia identificata da un numero progressivo (ad esempio, ciclamino numero 1, ciclamino numero 2, ..) e sia caratterizzata dalla posizione in cui si trova. Ogni famiglia di piante sia identificata univocamente da un nome e sia caratterizzata da una breve descrizione testuale. Ogni posizione all'interno del giardino botanico in cui è presente una pianta sia caratterizzata da un codice, che la identifica univocamente, e sia caratterizzata da un nome e da una coppia di coordinate spaziali. Il giardino botanico sia gestito da un certo insieme di giardinieri. Ogni giardiniere sia responsabile di un certo insieme di piante. Alcune piante possano essere assegnate a più di un giardiniere. Di ogni giardiniere si registrino giorni e orari di lavoro (il lunedì dalle 9:00 alle 14:00, il martedì dalle 9:00 alle 16:00, il giovedì dalle 11:00 alle 18:00, ..).

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte. Lo schema dovrà essere completato con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione.

Esercizio 4:

Si enuncino le regole dei protocolli 2PL e 2PL stretto e si dimostri che tali protocolli possono dare luogo a situazioni di stallo (*deadlock*). Si dia quindi un esempio di schedule che rispetta il protocollo 2PL ma vìola il protocollo 2PL stretto.

(FACOLTATIVO) Successivamente, date le seguenti definizioni:

```
create table T (x int primary key, y int not null);
insert into T (x,y) values (1,10), (2,20), (3,30);
```

si consideri la seguente esecuzione concorrente di tre transazioni T_1, T_2 e T_3:

Tempo	Sessione 1 (T_1)	Sessione 2 (T_2)	Sessione 3 (T_3)
1	`start transaction;`		
2		`start transaction;`	
3	`select y from T` `where x = 1;`		
4	`select y from T` `where x = 2;`		

5		`update T`
		`set y = y - 10`
		`where x = 3;`	

6	`select y from T` `where x = 3;`		
7		`rollback;`	
8			`start transaction;`

9			`update T`
			`set y = y + 20`
			`where x = 3;`

10			`commit;`
11	`select sum(y) from T;`		
12	`commit;`		

Si descriva in dettaglio il comportamento di tale esecuzione concorrente, in particolare specificando quali valori sono letti da T_1, nel caso in cui le tre transazioni siano eseguite

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

Esercizio 5:

Si consideri un file contenente 40000000 record di dimensione prefissata pari a 200 byte, memorizzati in blocchi di dimensione pari a 4096 byte in modo unspanned. La dimensione del campo chiave primaria V sia 14 byte; la dimensione del puntatore a blocco P sia 6 byte. Si chiede di confrontare fra loro le seguenti soluzioni, in termini di numero medio di accessi a blocco e di dimensione dell'indice.

(a) Ricerca basata su un indice multilivello statico ottenuto a partire da un indice secondario costruito sul campo chiave primaria V.

(b) Ricerca basata su un B-albero, con campo di ricerca il campo chiave primaria V, puntatore ai dati di dimensione pari a 7 byte e puntatore ausiliario di dimensione pari a 6 byte, assumendo che ciascun nodo del B-albero sia pieno al 70%.

(c) Ricerca basata su un B^+-albero, con campo di ricerca il campo chiave primaria V, puntatore ai dati di dimensione pari a 7 byte e puntatore ausiliario di dimensione pari a 6 byte, assumendo che ciascun nodo del B^+-albero sia pieno al 70%.