

Compito di Basi di dati

7 febbraio 2020

Esercizio 1:

Sia dato il seguente schema relazionale relativo a film e attori:

FILM(CodiceFilm, Titolo, Regista, Anno);

ATTORI(CodiceAttore, Nome, Cognome, Sesso, DataNascita, Nazionalità);

INTERPRETAZIONE(Film, Attore, Ruolo).

Si assuma che ogni film sia identificato univocamente da un codice e sia caratterizzato da titolo, regista e anno di uscita. Per semplicità, si assuma che ogni film sia diretto da un unico regista e ogni regista sia identificato univocamente dal suo cognome. Si ammetta la possibilità che vi siano film diversi con lo stesso titolo (questo è il caso, ad esempio, dei remake), ma si escluda la possibilità che due film con lo stesso titolo escano lo stesso anno. Si assuma che la base di dati non contenga film privi di attori (ad esempio, film di animazione), ossia che in ogni film reciti almeno un attore.

Ogni attore sia identificato univocamente da un codice e sia caratterizzato da nome, cognome, sesso, data di nascita e nazionalità. Si assuma che più attori possano recitare in un dato film e che un attore possa recitare in più film. Infine, si assuma che, in ogni film, un attore possa svolgere solo un ruolo.

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 necessario le funzioni aggregate):

- l'attore (gli attori se più di uno) che hanno recitato nel maggior numero di film;
- con riferimento ai soli film del regista Quentin Tarantino, gli attori che hanno recitato assieme in al più un film;
- gli attori che hanno recitato solo in film del regista Martin Scorsese, ma non in tutti.

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 realizzare una base di dati per la gestione di informazioni relative ai corsi di formazione professionale organizzati da una data azienda per i suoi dipendenti sulla base del seguente insieme di requisiti.

- Ogni corso è caratterizzato da un nome e da un insieme (che può essere vuoto) di corsi propedeutici. Ogni corso è identificato univocamente dal suo nome. Un dato corso può essere frequentato solo se sono già stati frequentati tutti i relativi corsi propedeutici.
- L'azienda può offrire più di una edizione di ogni corso in tempi e/o sedi diversi. Ogni edizione di un corso è caratterizzata da una data di inizio e da una data di fine, da una durata, da una sede, da un insieme di docenti (uno o più) e da un insieme di studenti (impiegati dell'azienda), ed è identificata univocamente dalla data di inizio e dalla sede, ossia edizioni diverse di un corso possono iniziare nella stessa data, ma in sedi diverse, o essere tenute nella stessa sede, ma iniziando in date diverse.
- Ogni docente è caratterizzato da un insieme di competenze, da un recapito e da uno o più numeri di telefono. Ogni studente è caratterizzato da un livello di preparazione iniziale.

- Di ogni impiegato dell'azienda registriamo i dati personali, i compiti, il titolo di studio di più alto livello e le eventuali esperienze di lavoro precedenti. Per ogni esperienza di lavoro precedente, si vuole tener traccia del tipo di lavoro svolto e della sua durata.

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. Si definiscano anche eventuali regole di gestione (regole di derivazione e vincoli di integrità) necessarie per codificare alcuni dei requisiti attesi del sistema.

Esercizio 4:

Si consideri un file contenente 100.000.000 record di dimensione prefissata pari a 500 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.

- Ricerca basata su un indice (primario) costruito sul campo chiave primaria V .
- Ricerca basata su un indice multilivello statico ottenuto a partire dall'indice primario.
- 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%.

Esercizio 5:

Si consideri la seguente istanza di basi di dati:

Università(nome, numeroRicercatori) = {(Udine, 100), (Trento, 200), (Verona, 300)}

Statistiche(data, totaleRicercatori) = {(2020-02-07, 0)}

Si considerino le seguenti due transazioni SQL: T1 è una transazione che calcola il numero totale di ricercatori (sommando i ricercatori tutte le università) e T2 è una transazione che sposta un ricercatore da UniTN a UniUD.

T1 (calcola il numero totale di ricercatori)

```
start transaction;
set transaction isolation level ...;
update Statistiche set tot_ricercatori = tot_ricercatori + UniVR.num_ricercatori
  from (select num_ricercatori from Università where nome='Verona') as UniVR;

update Statistiche set tot_ricercatori = tot_ricercatori + UniTN.num_ricercatori
  from (select num_ricercatori from Università where nome='Trento') as UniTN;

update Statistiche set tot_ricercatori = tot_ricercatori + UniUD.num_ricercatori
  from (select num_ricercatori from Università where nome='Udine') as UniUD;
commit;
```

T2 (sposta un ricercatore da UniTN a UniUD)

```
start transaction;
set transaction isolation level ...;
update Università set num_ricercatori = num_ricercatori - 1
  where nome='Trento';
update Università set num_ricercatori = num_ricercatori + 1
  where nome='Udine';
commit;
```

Per ciascuno dei 4 livelli di isolamento, indicare tutti i possibili risultati dell'esecuzione delle transazioni T1 e T2, e, per ognuno di essi, fornire e illustrare almeno uno schedule che produce tale risultato.