

Basi di dati II
Prof. Stefano Paraboschi
II prova - 22/12/2016

A. Si ha una base di dati per una banca.

CONTOCORRENTE(Codice, Ammontare, TimestampUltimaOperazione)
VERSAMENTO(CodV, CodConto, AmmontareV, TimestampV)
PRELIEVO(CodP, CodConto, AmmontareP, TimestampP)
COMMISSIONIINTERESSI(CodCI, CodConto, Ammontare, TimestampCI, Causale)

Ogni volta che in seguito all’inserimento di un nuovo prelievo l’ammontare sul conto corrente diventa negativo, devono essere sottratti ulteriori 10 Euro dal conto, inserendo un corrispondente elemento in COMMISSIONIINTERESSI con valore negativo e causale “Commissione scoperto”. Qualora l’ammontare sul conto scenda sotto i -1.000 Euro, vengono sottratti 50 Euro, con causale “Commissione massimo scoperto”. Si assuma la disponibilità di un evento temporale `NewDay` che viene generato dal sistema alla mezzanotte. Ogni mezzanotte si calcolano gli interessi in base alla disponibilità del conto corrente in quell’istante. Se l’ammontare disponibile sul conto è positivo, si carica un interesse pari allo 0,002% (1 su 50.000) dell’ammontare con causale “Interesse”. Se l’ammontare è negativo, si carica un interesse pari allo 0,05% (1 su 2.000) con la stessa causale. Scrivere un sistema di trigger che gestisce questi requisiti, reagendo inizialmente agli inserimenti nelle tabelle VERSAMENTO e PRELIEVO e aggiornando mediante trigger le tabelle CONTOCORRENTE e COMMISSIONIINTERESSI (si assuma la disponibilità di una funzione `GenerateId()` che produce un identificativo per le tuple di COMMISSIONIINTERESSI). Discutere poi la terminazione del sistema.

B. Si hanno queste tabelle:

STUDENTE(Matr, Cognome, Nome, DataNascita, Indirizzo)
ESAME(Matr, CodCorso, Data, Voto)
CORSO(CodCorso, Titolo, Docente)

Tutte le tabelle hanno un’organizzazione primaria sequenziale entry-sequenced. La tabella STUDENTE occupa 1K blocchi e contiene 10K tuple. La tabella ESAME occupa 10K blocchi e contiene 100K tuple. La tabella CORSO occupa 100 blocchi e contiene 1K tuple. Si consideri la seguente query:

```
select *  
from Studente S join Esame E on S.Matr = E.Matr  
      join Corso C E.codCorso = C.CodCorso  
where Cognome = "Rossi" and Titolo = "DB2"
```

La base di dati ha 100 studenti con cognome “Rossi” e 1000 esami di DB2 in totale. Ciascuna delle tabelle dispone di indici con alberi B+ sulla chiave (tutti alberi B+ con fanout pari a 100 e un numero di blocchi foglia pari a un decimo dei blocchi della tabella corrispondente). Si considerino tre scenari. Nel primo, la tabella ESAME ha un solo indice su (Matr, CodCorso). Nel secondo scenario, vi sono due indici per ESAME, per i due diversi ordinamenti degli attributi che compongono la chiave primaria. Nel terzo, oltre ai due indici su ESAME si aggiunge un indice per la tabella STUDENTE sugli attributi (Cognome, Nome, DataNascita). Individuare per ciascuno scenario la migliore strategia di esecuzione, fornendo una stima del numero di accessi richiesti.

C. Confrontare il protocollo di commit Paxos con il protocollo di commit a 2 fasi, mettendo in evidenza vantaggi e svantaggi di uno rispetto all’altro.

Si consideri poi la realizzazione di architetture di replicazione in modalità simmetrica (P2P). L’uso di un protocollo Paxos commit rispetto all’uso di un tradizionale commit a due fasi aumenterebbe la possibilità di adottare una modalità di trasmissione sincrona degli aggiornamenti?