

-Vincoli supportati da postgre?

In sql è possibile definire dei vincoli di integrità sui dati. Vengono implicitamente specificati quando un attributo viene associato al corrispondente dominio: i valori dell'attributo devono appartenere al relativo dominio.

I vincoli supportati sono la chiave primaria (non può essere null), le chiavi esterne, le chiavi candidate (unique). Si può poi usare il check nella definizione di una tabella, usare le asserzioni oppure usare i trigger (le regole attive).

-Gestione transazioni, file di log?

Le transazioni sono sequenze atomiche di operazioni corrette anche in presenza di concorrenza e con effetti definitivi. Termina con un abort e successivo roll back o con un commit. Viene tutto salvato nei file di log che permettono di gestire l'abort e in caso roll back del DBMS.

Il sistema del DB tiene traccia sul disco dei vecchi valori di ogni dato su cui viene fatta una transazione. Queste informazioni vengono salvate sui file di log, che vengono usati per ripristinare i vecchi valori nel caso una transazione non vada a buon fine. Sembra che la transazione non sia mai avvenuta.

-Per fare undo e redo che informazioni devo avere nel file di log? Cosa deve esserci memorizzato?

Ogni modifica del DB fatta da una transazione viene salvata nei log. Salviamo l'identificatore della transazione che effettua la modifica, l'identificatore dell'oggetto data che sta venendo modificato e sia il vecchio valore che il nuovo valore di quell'oggetto dato.

Questo permette di effettuare sia il redo che l'undo.

I log hanno i seguenti dati: identificatore transazione, identificatore oggetto-data, vecchio valore, nuovo valore.

-Checkpoints

Quando il sistema va in crash, bisogna consultare i file log per determinare quali transazione vanno rifatte e quali devono essere tolte. Visto che cercare tutte queste informazioni nei file di log è pesante per il sistema e richiede molto tempo usiamo dei checkpoint. In un determinato momento del tempo viene eseguita una sorta di backup durante la quale non si possono effettuare transazioni e da in output un file su disco con tutte le ultime modifiche e lo stato del database.

-Proprietà acide?

1. Atomicità: ogni transazione viene trattata come una singola unità. O fallisce completamente, o ha successo per intero.
2. Consistenza: si assicura che una transazione possa portare il DB da un "punto valido"/valid state ad un altro, mantenendo invariati del DB: constraints, triggers, chiavi ecc che non vengono modificate. Previene una corruzione del database da una transazione illegale. Non garantisce comunque che una transazione sia corretta.
3. Isolamento: Le transazioni spesso vengono eseguite concorrenti fra loro. L'isolamento garantisce che le transazioni concorrente lascino il database nello stesso stato in cui il DB sarebbe se le transazioni fossero avvenute in modo sequenziale.
4. Durabilità: garantisce che una volta che una transazione abbia fatto commit, rimarrà committed anche nel caso di un fallimento di sistema. Significa che le transazioni completate vengono salvate in memoria non volatile.

-Domande riguardo ai check su tabelle esterne

-Differenze tra asserzione e check? quando scegliere uno o l'altro?

Le asserzioni sono pezzi di SQL che si assicurano che una condizione sia soddisfatta, oppure fermano l'azione in corso dal modificare il database. Le asserzioni non modificano i dati, controllano solamente certe condizioni. Le asserzioni non sono linkate a tables specifiche nel DB o a eventi specifici.

I check, sono dei semplici check che sono relativi ad una tabella specifica o ad una vista. Andrebbe riscritto diverse volte se si volesse usare lo stesso check su più tabelle.

- trigger? scopo?

I trigger sono dei pezzi di codice SQL che vengono eseguiti o prima o dopo un update, inserimento o cancellazione in un database.

Un esempio di trigger sarebbe "prima di fare un update sul record di un cliente, salva una copia del record."

I trigger sono potenti perché controllano una condizione, ma possono anche modificare i dati.

I trigger inoltre sono linkati a tabelle specifiche oppure event.

-A cosa servono i vincoli?

Un DB non deve solo memorizzare dati, deve anche garantire la loro correttezza. Per farlo, il linguaggio SQL fornisce i vincoli di integrità, regole che vengono applicate alle tabelle e che vincolano i valori che possono essere inseriti. E' possibile applicare vincoli su singole colonne o anche su più tabelle. La loro integrità impone delle restrizioni sui valori assunti da colonne e tabelle.

-Come funzionano gli abort delle transazioni?

La transazione inizia come "attiva" e poi come "partially committed". Dopo di che abbiamo un "failed" dopo la scoperta che la normale esecuzione non può più procedere. Si passa alla fase "aborted". Una transazione aborted non deve avere nessun effetto sul sistema e tutti i cambi che la transazione ha fatto devono essere ripristinati. Si va a recuperare dai file di log lo stato precedente dei dati e effettua il roll back allo stato precedente.

-Fare un trigger che controlla che un tecnico non possa fare più di 5 controlli al giorno.

```
CREATE TRIGGER no5controlli
ON controlli
AFTER INSERT
AS
Declare @numero_controlli int
IF EXISTS (select numero_controlli from controlli where numero_controlli > 5)
BEGIN
RAISERROR ('max controlli')
ROLLBACK TRANSACTION;
END;
```

-Cos'è il protocollo 2PL?

Il two-phase locking è un metodo di controllo di concorrenza che garantisce serializzabilità. Il protocollo utilizza dei locks, applicati dalla transazione a dei dati, che possono bloccare altre transazione dall'accesso ad uno stesso dato durante una transazione.

I lock sono messi e tolti in due fasi: la fase di espansione, dove i lock sono acquisiti e non rilasciati. La fase di shrink, dove i lock sono rilasciati e non acquisiti.

Evita fondamentalmente che due transazioni concorrenti si modifichino dei file a vicenda.

Evita soprattutto situazioni di deadlock.

-Possono verificarsi deadlock, il sistema cosa fa per risolvere queste situazioni?

Se si va in stato di deadlock, nessuna delle due supposte transazioni può più procedere. Bisogna quindi che il sistema faccia il roll back di una delle due transazioni. Quando una delle due transazione è stata rollata, gli oggetti dato che erano bloccati dall'altra transazioni sono sbloccati, e resi disponibili alla prima transazione per continuare con l'esecuzione.

Si usa sistema di lock -> vedi troppi / pochi lock inconsistenza.

-Quando vengono verificati i vincoli di check?

I vincoli di check di norma vengono inseriti nel CREATE TABLE di una tabella. Di conseguenza i vincoli di check vengono verificati durante l'aggiunta di dati alla tabella. Esempio, vincolo di età per aggiungere una persona alla tabella "adulti".

-Come funziona il controllo della concorrenza con le multiversioni?

E' un metodo di controllo della concorrenza comunemente usato nei sistemi di gestione delle basi di dati per fornire access concorrente alle basi di dati e nei linguaggi di progr per implementare una memoria transazionale.

Una base di dati implementa gli aggiornamenti selezionando il dato vecchio come obsoleto e aggiungendo la versione nuova del dato, archiviando più copie dello stesso dato.

-Cos'è uno schedule? E uno scheduler ?

E' una sequenza di operazioni di ingresso/uscita relative ad un dato insieme di transazioni concorrenti. Fondamentalmente uno schedule S1 è una sequenza tipo S1:r1(x) r2(z) w1(x) w2(z)... Le operazioni compaiono nelle schedule nell'ordine temporale di esecuzione sulla base di dati.

Controlla la concorrenza delle transazioni. Tiene traccia di tutte le operazioni eseguite sulla base di dati dalle transazioni e decide se accettare o rifiutare le operazioni che vengono via via richieste dalle transazioni.

-Quando uno scheduler è serializzabile?

Uno schedule S si dice seriale se per ogni transazione t, tutte le azioni di t compaiono in S in sequenza, senza essere inframmezzate da azioni di altre transazioni.

-Cosa sono schema e istanza di un database?

Istanza è l'attuale contenuto dei dati, lo schema è quello che io ho stabilito prima durante la create table. Alter table quindi non va bene per modificare l'istanza.

Si dice che una transazione è recoverable se le viene impedito di eseguire commit prima che tutte le transazioni, che scrivono valori letti da T eseguano commit o abortiscano.

- perché funziona il protocollo basato su timestamp?

- query: estrai aerei dove sono stati effettuati 100 controlli
SELECT codice_aereo FROM aerei WHERE (controlli >= 100) ;

- quando si usa il before o after?
Nei trigger

- tipi di fallimenti sul file di log

- se la visita medica è scaduta devi impedire l'inserimento del controllore
- procedura per il recovery

- elencare le memorie con cui il DBMS lavora
- perché solo il log è in memoria stabile?

- Trigger: quando inseriamo un punteggio, in automatico dobbiamo inserire che il controllo programma sia questo mese.
- Esempio di schedule che presenta un deadlock.

Nuove domande:

1. **Schema visite mediche:**

```
create table visite (
  medico varchar(50) not null references medici(codice)
  paziente int not null references pazienti(codice_ssn)
  data date not null,
  diagnosi varchar(500) not null,
  medicinale int references medicinali(codici)
  Primary key(medico, paziente, data)
);
```

2. **Quanti sono i medici che hanno effettuato almeno una visita nel 2020:**

```
select distinct codice, nome, cognome from medici, visite where (visite.medico=medici.codice
and date in "1/1/2020");
```

```
select count(distinct visite.medico) from visite v, medici
Where m.codice = v.medico AND
Extract year from (data) = '2020'
);
```

3. **Per ogni medico in quanti giorni del 2020 ha fatto una visita**

```
select medico, count (distinct data)
where extract year from data = '2020'
group by medico
;
```

4. **Modificare tab visite, tutte le visite registrate in data 4/9/2020 -> 3/9/2020**

```
update visite(
set data='2020-09-03' where data ='2020-09-04'
);
```

5. **Cosa è un trigger?**

6. **Nel modello ER abbiamo generalizzazione e specializzazione. Si può usare per intendere un insieme come sotto insieme**

7. **Cosa significa che le transazioni sono isolate fra di loro + acide + atomicità**

8. **Create table pubblicazioni(**

```
...
);
```

9. **Pubblicazioni (codice titolo) che nel 2020 hanno venduto almeno 1000 copie**

```
Select p.codice, p.titolo, sum(copie_vendute) from pubblicazioni p, vendite v where
(v.pubblicazione = p.codice AND
v.data = '2020' //AND v.copie_vendute >= 1000
```

);

Group by p.codice, p.titolo
Having sum(copie_vendute)>=1000
Order by sum(copie_vendute) desc;

10. Eliminare le librerie per le quali non si è avuta nessuna vendita

Delete from librerie
Where codice not in (select libreria from vendite);

11. Alg relazionale project su attribuito nazione di nazione su autori

Proj_ nazione(autori) -> in sql -> select distinct nazione from autori;

12. Codifica con modello logico concettuale la tabella visite e medicinali (portalo in ER e specifica la loro associazione)

13. Cosa si fa nei DBMS per implementare il requisito di atomicità?

Si usa il 2pl / checkpoint / scrivo sui file di log -> quanta roba c'è nei file di log?

14. Cosa si intende quando si parla di diritti di accesso degli utenti nei DB?

15. Algebra relazionale, alcuni operatori sono insiemistici, quali?

Unione, intersezione, differenza,

16. Condizioni soddisfatte per unione in alg rel?

Unisco due relazioni R1 e R2 presenti contemporaneamente in R1 e R2 -> ris R1

17. Per ogni giocatore quante volte è stato convocato ai mondiali

Select cognome_nome, id count (id) from giocatori g, convocati c where (c.giocatore = g.id group by cognome_nome, id);

18. Vogliamo inserire in squadre le stesse nazioni e lo stesso allenatore che avevano partecipato all'edizione precedente (2018)

Insert into squadre(nome, cognome_allenatore)

Select n.nome, s.cognome_nome_allenatore from nazioni n, squadre s, edizioni e where (e.anno = '2022')

19. Differenza fra union sql e union algebra relazionale

20. Come viene gestito il valore null

21. Vincolo check (data > 1920), posso inserire una persona con data nulla ?

22. Che succede nel DBMS per implementare l'isolamento

Mio esame

Quali sono i proprietari di auto (cod fiscale e nome) per le quali si ha avuto almeno 1 sinistro nel 2019

Select distinct codice_fiscale, nome from proprietari p, sinistro s, auto a, AutoCoinvolte au where (s.data between '2019-01-01' and '2019-12-31' AND a.proprietario = p.codice_fiscale AND au.auto = a.targa AND s.codice=au.sinistro);

Vediamo tutti i codici di sinistro che hanno coinvolto almeno 5 auto

Select sinistro from autocoinvolte au group by sinistro having (count (auto) >= 5);

Come funziona il controllo dei vincoli quando ci sono di mezzo valori nulli

Danno*

Check danno => 200

Schema er per sinistri stradali

Modificare l'importo del danno di tutte le auto coinvolte nel sinistro 1313 per fare in modo che
Tutte quelle che avevano importo del danno null ora lo hanno a 1000 euro

```
Update autocoinvolte (  
Set importo_danno = 1000 where (sinistro = 1313 AND importo_danno is null)  
);
```