SQL: Concetti Avanzati

Fallucchi Francesca
fallucchi@info.uniroma2.it
http://art.uniroma2.it/fallucchi/
a.a. 2010/2011

Vincoli di integrità generici

- Specifica di vincoli di ennupla (e anche vincoli più complessi, non sempre supportati)

check (Condizione)

Check, esempio

```
create table Impiegato
(Matricola character(6) primary key,
Cognome character(20) not null,
Nome character(20) not null,
Sesso character not null
Stipendio integer,
Ritenute integer,
Netto integer,
Superiore character(6))
```

Check, esempio

```
create table Impiegato
(Matricola character(6) primary key,
Cognome character(20) not null,
Nome character(20) not null,
Sesso character not null check (sesso in ('M','F'))
Stipendio integer,
Ritenute integer,
Netto integer,
Superiore character(6),
check (Netto = Stipendio - Ritenute ) )
```

Asserzioni

- Specifica vincoli a livello di schema
- Esempi

```
create assertion NomeAss
    check ( Condizione )

create assertion AlmenoUnoStudente
    check (1 <= (    select count(*)
                     from Studente ))
```

Viste

```
create view NomeVista [ ( ListaAttributi ) ] as SelectSQL
[ with [ local | cascaded ] check option ]

create view ImpiegatiAmmin
    (Nome, Cognome, Stipendio) as
  SELECT Nome, Cognome, Stipendio
  FROM Impiegato
  WHERE Dipart = 'Amministrazione' and
    Stipendio > 10
```

Interrogazioni sulle viste

- Possono fare riferimento alle viste come se fossero relazioni di base

```
SELECT * from ImpiegatiAmmin
```

- equivale a (e viene eseguita come)

```
SELECT Nome, Cognome, Stipendio
FROM Impiegato
WHERE Dipart = 'Amministrazione'
        and Stipendio > 10
```

Aggiornamenti sulle viste

- Ammessi (di solito) solo su viste definite su una sola relazione
- Alcune verifiche possono essere imposte

Check Option

```
create view ImpiegatiAmminPoveri as
SELECT *
FROM ImpiegatiAmmin
WHERE Stipendio < 50
with check option
```

- check option permette modifiche, ma solo a condizione che la ennupla continui ad appartenere alla vista (non posso modificare lo stipendio portandolo a 60)

Aggiornamento su Vista

```
create view ImpiegatiAmminPoveri as
  SELECT *
  FROM ImpiegatiAmmin
  WHERE Stipendio < 50
  WITH local check option

update ImpiegatiAmminPoveri
  set stipendio = 8
  where nome = 'Paola'
```

Un'interrogazione non standard

- Interrogazione scorretta

```
SELECT avg(count(distinct Ufficio))
FROM Impiegato
GROUP BY Dipart
```

- Con una vista

```
create view DipartUffici(NomeDip,NroUffici) as
    SELECT Dipart, count(distinct Ufficio)
    FROM Impiegato
    GROUP BY Dipart

  SELECT avg(NroUffici )
  FROM DipartUffici
```

Ancora sulle viste

- Estrarre il dipartimento caratterizzato dal massimo della somma degli studendi

```
SELECT Dipart
FROM Impiegato
GROUP BY Dipart
HAVING sum(Stipendio) >= all
                (SELECT sum(Stipendio)
                 FROM Impiegato
                 GROUP BY Dipart)
```

- La nidificazione nella having non è ammessa in alcuni sistemi

Soluzione con le viste

```
create view BudgetStipendi(Dip,TotaleStipendi) as
  select Dipart, sum(Stipendio)
  from Impiegato
  group by Dipart

select Dip
from BudgetStipendi
where TotaleStipendi =(select max(TotaleStipendi)
                       from BudgetStipendi)
```

Transazione

- Insieme di operazioni da considerare indivisibile ("atomico"), corretto anche in presenza di concorrenza e con effetti definitivi
- Proprietà ("ACID"):
  - Atomicità
  - Consistenza
  - Isolamento
  - Persistenza

Le transazioni sono … atomiche

- La sequenza di operazioni sulla base di dati:
  - viene eseguita per intero
  - non viene eseguita affatto:
- Esempio: trasferimento di fondi da un conto A ad un conto B:
  - si fanno il prelevamento da A e il versamento su B
  - nessuno dei due

Le transazioni sono … consistenti

- Al termine dell'esecuzione di una transazione, i vincoli di integrità debbono essere soddisfatti
- "Durante" l'esecuzione ci possono essere violazioni, ma alla fine:
  - sono risolte
  - oppure la transazione deve essere annullata per intero ("abortita")

Le transazioni sono … isolate

- L'effetto di transazioni concorrenti deve essere coerente (ad esempio "equivalente" all'esecuzione separata)
  - se due assegni emessi sullo stesso conto corrente vengono incassati contemporaneamente si deve evitare di trascurarne uno

Le transazioni sono … durevoli

- La conclusione positiva di una transazione corrisponde ad un impegno (commit) a mantenere traccia del risultato in modo definitivo, anche in presenza di guasti e di esecuzione concorrente

Transazioni in SQL

- Una transazione inizia al primo comando SQL dopo la "connessione" alla base di dati oppure alla conclusione di una precedente transazione
- lo standard indica anche un comando `start transaction`, non obbligatorio, e quindi non previsto in molti sistemi

Transazioni in SQL

- Conclusione di una transazione
  - commit [work]: le operazioni specificate a partire dall'inizio della transazione vengono eseguite sulla base di dati
  - rollback [work]: si rinuncia all'esecuzione delle operazioni specificate dopo l'inizio della transazione
- Molti sistemi prevedono una modalità autocommit, in cui ogni operazione forma una transazione

Una transazione in SQL

```
start transaction (opzionale)
update ContoCorrente
  set Saldo = Saldo - 10
  where NumeroConto = 12345 ;
update ContoCorrente
  set Saldo = Saldo + 10
  where NumeroConto = 55555 ;
commit work;
```

Procedure e Trigger

- procedure e trigger permettono di implementare funzioni e controlli sui dati, direttamente nella base di dati
- Il codice SQL è arricchito con istruzioni procedurali
- La sintassi non è standardizzata, per cui ogni DBMS può presentare proprie personalizzazioni

Procedure

- Le procedure permettono di creare uno strato di astrazione sulla rappresentazione delle informazioni strutturate nel database
- Si possono sollevare le applicazioni da interventi troppo di dettaglio sulla struttura dei dati (e.g. verifiche di consistenza, aggiornamento di informazioni, ecc..)
- Le stored procedure sono più efficienti di un equivalente programma client

Istruzioni

- Istruzioni dichiarative: sono le istruzioni SQL con eventuali estensioni per la gestione di variabili
- Istruzioni procedurali: costrutti analoghi a quelli normalmente usati nei linguaggi di programmazione
  - If-then-else, while-do, ecc

Create Procedure

```
CREATE PROCEDURE name
   [(param <datatype> [, param <datatype> ...])]
   [RETURNS (param <datatype> [, param <datatype> ...])]
   AS <procedure_body> [terminatore]

<procedure_body> = [<variable_declaration_list> <block>

<variable_declaration_list> =
            DECLARE VARIABLE var <datatype>;
            [DECLARE VARIABLE var <datatype>; ...]
<block> = BEGIN <compound_statement>
            [<compound_statement> ...]
          END
```

Esempio

```
procedure AssegnaCitta(:Dip varchar(20),
                       :Citta varchar(20))
   update Dipartimento
   set Città = :Citta
   where Nome= :Dip;
```

Trigger

- I trigger sono procedure attivabili automaticamente al verificarsi di uno specifico evento
- Gli eventi gestibili con trigger sono: inserimenti, aggiornamenti e cancellazioni in una tabella
- Bisogna fare attenzione affinché un trigger non attivi altri trigger in cascata con il rischio di generare dei loop

Create Trigger

CREATE TRIGGER name FOR tabella {BEFORE | AFTER } {INSERT | DELETE | UPDATE} AS <trigger_body>

- Il trigger verrà eseguito "prima" o "dopo" di inserimento/aggiornamento/cancellazione sulla tabella citata nella clausola FOR
- Il corpo del trigger può contenere una qualunque istruzione definita anche per le procedure
- Un trigger NON può ritornare valori

Esempio

```
create trigger ImpiegatiSenzaDip
after insert into Impiegati
for each row
when (new.Dipart is null)
update Impiegati
      set Dipart ='Nuovi arrivati'
      where Matr= new.Matr
```

Invocazioni

- Una procedura può essere richiamata da un'altra procedura o da un trigger

  EXECUTE PROCEDURE procedura [(parametro tipo, ...)] [RETURING_VALUES (risultato)]

- Se la procedura restituisce dei valori essi vanno assegnati ad altrettante variabili
- ovviamente i trigger non possono essere invocati essi sono attivati automaticamente dal DBMS