Basi Di Dati e di conoscenza

MySQL-Basi di dati attive (Trigger e stored Procedures)
Transazioni

Contenuti della lezione

- Basi di dati Attive
- Stored Procedures
- Trigger
- Transazioni: commit e rollback
- Proprietà acide delle tranzasioni

Basi di dati Attive

- Una base di dati che contiene regole attive (chiamate *trigger o stored procedure*)
- Presentazione:
  - Definizione dei trigger in SQL:1999
  - Definizione dei trigger in DB2 e Oracle
  - Problemi di progetto per applicazioni basate sull'uso dei trigger

Contenuti della lezione

- Basi di dati Attive
- Stored Procedures
- Trigger
- Transazioni: commit e rollback
- Proprietà acide delle tranzasioni

Stored procedure

- Le **stored procedure** sono un insieme di istruzioni SQL precompilate e memorizzate ne
- **Vantaggi**
  - **Efficienza**: Minimizzano il traffico di rete eseguendo le operazioni direttamente sul server.
  - **Sicurezza**: Limitano l'accesso diretto alle tabelle, controllando le operazioni consentite.
  - **Riutilizzabilità**: Possono essere richiamate da più punti dell'applicazione.l database per eseguire operazioni specifiche.

Struttura di una Stored Procedure

- **Intestazione**: Nome, parametri in ingresso, parametri in uscita.
- **Corpo**: Contiene il codice SQL per svolgere l'operazione desiderata.
- **Ritorno**: Se necessario, può contenere valori restituiti o output..

Sintassi

```
CREATE PROCEDURE nome_procedura
    @parametro1 tipo_dato,
    @parametro2 tipo_dato
AS
    -- Corpo della procedura
    -- Inserire qui le istruzioni SQL
GO;
```

Stored Procedure: Esempio creazione

- **Intestazione**: Nome, parametri in ingresso, parametri in uscita.
- **Corpo**: Contiene il codice SQL per svolgere l'operazione desiderata.
- **Ritorno**: Risultati o valori di output, se presenti.

Esempio

```
CREATE PROCEDURE CalcolaMediaVoti
    @corsoId INT
AS
    SELECT AVG(Voto) AS MediaVoti
    FROM Voti
    WHERE CorsoId = @corsoId
GO;
```

Stored Procedure: Esecuzione e modifica

Esecuzione di una Stored Procedure:Corpo:

```
EXEC CalcolaMediaVoti @corsoId = 123
```

Modifica e Rimozione delle Stored Procedure:

```
ALTER PROCEDURE nome_procedura
    @nuovo_parametro tipo_dato
AS
    -- Modifiche alla procedura
GO

DROP PROCEDURE nome_procedura
```

Stored Procedure: Esempio

Esempio: Stored procedure che seleziona i Clienti da una città specifica con un particolare codice postale dalla tabella **`Customers`**.

```
CREATE PROCEDURE SelectAllCustomers @City nvarchar(30), @PostalCode
nvarchar(10)
AS
SELECT * FROM Customers WHERE City = @City AND PostalCode =
@PostalCode
GO;

EXEC SelectAllCustomers @City = 'London', @PostalCode = 'WA1 1DP';
```

Contenuti della lezione

- Basi di dati Attive
- Stored Procedures
- Trigger
- Transazioni: commit e rollback
- Proprietà acide delle tranzasioni

Trigger

- Un **trigger** è una stored procedure in un database che viene automaticamente attivata ogni volta che si verifica un evento speciale nel database.
- Ad esempio, un trigger può essere attivato quando viene inserita una riga in una tabella specifica o quando determinate colonne di una tabella vengono aggiornate.

Idea dei trigger

- **Paradigma**: Evento-Condizione-Azione
  - Quando un evento si verifiva
  - Se la condizione è vera
  - Allora l'azione è eseguita
- **Questo modello consente computazioni reattive**
- Non è il solo tipo di regole:
  - Vincoli di integrità
  - Regole datalog
  - Regole di business
- **Problema**: è difficile realizzare applicazioni complesse

Evento-Condizione-Azione

- **Evento**
  - Normalmente una modifica dello stato del database: **`insert, delete, update`**
  - Quando accade l'evento, il trigger è *attivato*
- **Condizione**
  - Un predicato che identifica se l'azione del trigger deve essere eseguita
  - Quando la condizione viene valutata, il trigger è *considerato*
- **Azione**
  - Una sequenza di update SQL o una procedura
  - Quando l'azione è eseguita anche il trigger è *eseguito*

Sintassi di un trigger

```
CREATE TRIGGER nome_trigger
ON nome_tabella
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    -- Corpo del trigger
    -- Istruzioni da eseguire
END
```

Trigger esempio

```
create trigger AccountMonitor
after update on Account
for each row
when new.Totale > old.Totale
insert values
    (new.NumeroConto,
     new.Totale-old.Totale)
    into Pagamenti
```

Esempio

- Seguono la sintassi e semantica di SQL:1999
- Esempio: gestione salari

```
create trigger ControllaStipendi
after update of Stipendio on Impiegato
for each row
when (new.Stipendio < old.Stipendio * 0.97)
begin
  update Impiegato
  set Stipendio = old.Stipendio * 0.97
  where Matr = new.Matr;
end;
```

Caratteristiche dei trigger

- **Attivazione Automatica:** Un trigger non può essere chiamato direttamente come una stored procedure. Viene automaticamente attivato quando si verifica un evento di modifica dei dati su una tabella, distinguendosi così da una procedura.

- **Differenze chiave tra Trigger e Stored Procedure:**

1. I trigger non possono essere invocati o eseguiti manualmente.
2. I trigger non possono ricevere parametri.
3. All'interno di un trigger non è possibile eseguire il commit o il rollback di una transazione.

Confronto tra Trigger e Stored Procedure

Trigger:

- Attivazione automatica in risposta a eventi di modifica dei dati.
- Non può essere chiamato manualmente.
- Non riceve parametri.
- Non può eseguire commit o rollback di transazioni.

Stored Procedure:

- Richiede una chiamata diretta per essere eseguita.
- Può ricevere parametri per personalizzare l'esecuzione.
- Può eseguire commit o rollback di transazioni.

Contenuti della lezione

- Basi di dati Attive
- Stored Procedures
- Trigger
- Transazioni: commit e rollback
- Proprietà acide delle tranzasioni

Transazione

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

Le transazioni sono … atomiche

- La sequenza di operazioni sulla base di dati viene eseguita per intero o per niente:
  - trasferimento di fondi da un conto A ad un conto B: o si fanno il prelevamento da A e il versamento su B o 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 se restano alla fine allora 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

I risultati delle transazioni sono durevoli

- La conclusione positiva di una transazione corrisponde ad un impegno (in inglese 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)
- 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;
```