



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Il Linguaggio SQL DML/DQL

Basi di Dati

Corso di Laurea in Informatica per il Management

Alma Mater Studiorum - Università di Bologna

Prof. Marco Di Felice

Dipartimento di Informatica – Scienza e Ingegneria

marco.difelice3@unibo.it

Il Linguaggio SQL

- **LINGUAGGI** supportati dai **RDBMS**

3. SQL (*Structured Query Language*)

Diverse **versioni** del linguaggio:

- SQL-86 → Costrutti base
- SQL-89 → (**SQL1**) Integrità referenziale
- SQL-92 (**SQL2**) → SQL Interattivo, sistema tipi
- SQL:1999 (**SQL3**) → Modello ad oggetti
- SQL:2003 (**SQL3**) → Nuove parti: SQL/JRT, SQL/XML
- SQL:2006 (**SQL3**) → Estensione di SQL/XML
- SQL:2008 (**SQL3**) → Lievi aggiunte
-

<http://troels.arvin.dk/db/rdbms/>

Il Linguaggio SQL

Oltre ad i costrutti base di SQL2 visti fin qui, esistono molti **costrutti avanzati** (i) definiti in **SQL3** e/o (ii) **dipendenti** dallo specifico **DBMS**.



- **Procedure** (*Stored Procedures*)
- **Trigger**
- **Permessi**

Il Linguaggio SQL

Stored Procedures → Frammenti di codice SQL, con la possibilità di specificare un **nome**, dei **parametri in ingresso** e dei **valori di ritorno**.

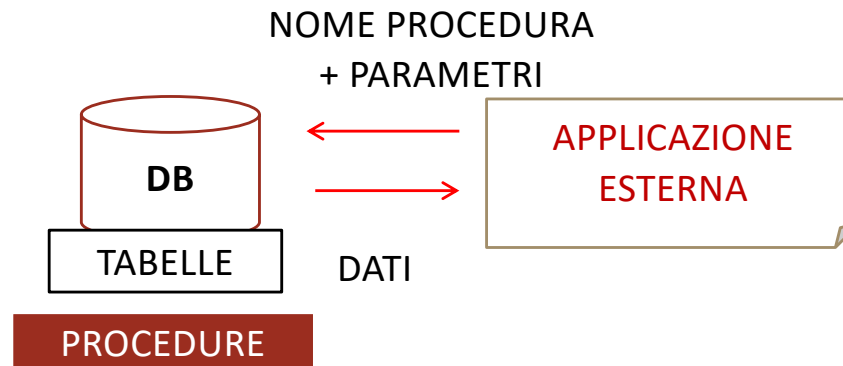
```
Procedure ModificaStipendio (MatricolaN: varchar(20),  
StipendioNew: smallint)  
    update Impiegati  
    set Stipendio=StipendioNew  
    where Matricola=MatricolaN
```

Il Linguaggio SQL



MODELLO con STORED PROCEDURE

- Efficienza
- Maggiore espressività
- Sicurezza



Il Linguaggio SQL

Esempio: **definizione di stored procedure** in **MySQL**:

```
CREATE PROCEDURE myPROC (IN param1 INT, OUT param2 INT)
SELECT COUNT(*) INTO param2
FROM tabella
WHERE name = param1;
```

```
mysql>> CALL myPROC("Test",@variable);
```

Il Linguaggio SQL

Le **estensioni procedurali** consentono di:

- Aggiungere **strutture di controllo** al linguaggio SQL (es. cicli, strutture condizionali if then else, etc).
- Dichiarare **variabili** e tipi di dato **user-defined**.
- Definire **procedure sui dati** avanzate, che sono ritenute “**sicure**” dal DBMS.

Il Linguaggio SQL

Ogni DBMS offre una sua **estensione procedurale**:

- **PL/SQL** → Linguaggio di Oracle Server
- **SQL PL** → Linguaggio di IBM DB2
- **PL/pgSQL** → Linguaggio di PostgreSQL
- **mySQL** → Linguaggio di MySQL

Il Linguaggio SQL

Esempio di costrutti procedurali in PostgreSQL

COSTRUTTO CONDIZIONALE

`if <Condizione> then ... else ... endif`

```
IF user_id <> 0 THEN
    UPDATE USERS
        SET USERS.email=v_email
        WHERE (USERS.user_id = user_id)
ENDIF
```

Il Linguaggio SQL

Esempio di costrutti procedurali in PostgreSQL

COSTRUTTO ITERATIVO

```
WHILE (Expression) LOOP statements END LOOP
```

```
WHILE ncycle>0 LOOP
```

```
    UPDATE SALARY
```

```
    SET SALARY.amount=SALARY.amount - 100
```

```
    ncycle:=ncycle -1;
```

```
END LOOP
```

Il Linguaggio SQL

Oltre ad i costrutti base di SQL2 visti fin qui, esistono molti **costrutti avanzati** (i) definiti in **SQL3** e/o (ii) **dipendenti** dallo specifico **DBMS**.

- **Procedure** (*Stored Procedures*)



- **Trigger**

- **Permessi**

Il Linguaggio SQL

ORDINE

Nome	Codice	Quantita
Xbee Radio Shield	123	3
Arduino Uno Shield	5565	2
Arduino Ethernet	14354	1

Vorrei implementare un **comportamento automatico** del tipo:

- Ogni mese, vengono rimosse tutte le righe presenti dalla tabella ORDINI e spostate nella tabella ORDINI_PENDENTI...

Il Linguaggio SQL

ORDINE			MAGAZZINO		ACQUISTO		
Nome	Codice	Quantita	Codice	Quantita	Codice	Q.a	Data
Xbee Radio Shield	123	3	123	0	123	3	1/2/2012
Arduino Uno Shield	5565	2	5565	2			
Arduino Ethernet	14354	1	14354	1			

Vorrei implementare un **comportamento automatico** del tipo:

- Ogni volta in cui l'utente fa un'ordine, si aggiorna la tabella Magazzino, e, nel caso non vi fossero più prodotti di quel tipo, si aggiorna anche la tabella Acquisti ...

Il Linguaggio SQL

Trigger (o **regole attive**) → meccanismi di gestione della base di dati basati sul paradigma ECA (Evento/Condizione/Azione).

- **Evento:** primitive per la manipolazione dei dati (insert, delete, update)
- **Condizione:** Predicato **booleano**
- **Azione:** sequenza di istruzioni SQL, talvolta procedure SQL specifiche del DBMS.

Il Linguaggio SQL

Trigger (o **regole attive**) → meccanismi di gestione della base di dati basati sul paradigma ECA (Evento/Condizione/Azione).

SINTASSI SQL3

Create trigger Nome

Modo Evento on Tabella

[referencing Referenza]

[for each Livello]

[when (IstruzioneSQL)]

Istruzione/ProceduraSQL

← EVENTO

← CONDIZIONE

← AZIONE

Il Linguaggio SQL

- **Modo** → before/after
- **Evento** → insert/delete/update
- **Referencing** → qui possono essere inserite variabili globali per aumentare l'espressività del trigger...
- **Livello** → row (Il trigger agisce a livello di righe)
statement (Il trigger agisce globalmente a livello di tabella)

Due modalità di esecuzione: **immediata** vs **differita**.

Il Linguaggio SQL

Esempio di Trigger in SQL3

```
CREATE TRIGGER CHECKAUMENTO  
BEFORE UPDATE OF CONTO ON IMPIEGATO  
FOR EACH ROW  
WHEN (NEW.STIPENDIO > OLD.STIPENDIO * 1.2)  
SET NEW.STIPENDIO=OLD.STIPENDIO * 1.2
```

- **Modo** è definito come before.
- **Evento** è definito come update.
- **Livello** è definito come row.

Il Linguaggio SQL

Oltre ad i costrutti base di SQL2 visti fin qui, esistono molti **costrutti avanzati** (i) definiti in **SQL3** e/o (ii) **dipendenti** dallo specifico **DBMS**.

- **Procedure** (*Stored Procedures*)

- **Trigger**



- **Permessi**

Il Linguaggio SQL

SQL2/SQL3 prevede meccanismi di **controllo di accesso** alle risorse dello schema del DB.

Di default, ogni risorsa appartiene all'utente che l'ha definita ... Su ciascuna risorsa sono definiti dei **privilegi (grant)**:

- insert/update/delete → tabelle/viste
- select → tabelle/viste
- references → tabelle/attributi
- usage → domini

Il Linguaggio SQL

Il comando **grant** consente di assegnare **privilegi su una certa risorsa** ad utenti specifici.

```
grant Privilegio on Risorsa/e to Utente/i [with grant option]
```

L'opzione `with grant option` consente di propagare il privilegio ad altri utenti del sistema...

```
grant select on Impiegati to Marco with grant option  
grant delete on Impiegati, Salari to Marco, Michele
```

Il Linguaggio SQL

Il comando **revoke** consente di **revocare privilegi su una certa risorsa** ad utenti specifici.

```
revoke Privilegio on Risorsa/e from Utente/i  
[cascade|restrict]
```

L'opzione **cascade** agisce ricorsivamente sui privilegi eventualmente concessi da quell'utente ...

```
revoke select on Impiegati to Marco cascade  
revoke delete on Impiegati, Salari to Marco, Michele
```

Il Linguaggio SQL

In SQL3 è possibile definire dei **ruoli** per regolare l'accesso alle risorse di un database.

Ruolo → Contenitore di privilegi

ESEMPIO di RUOLO

- Insert su Tabella Impiegati
 - Select su Tabella Retribuzioni
 - Update su Tabella Progetti
-
- Comandi SQL3: **create role/set role**