SQL

Il linguaggio SQL consente di interagire con le basi di dati ed in particolare con i dati.

Questo linguaggio può pensarsi come costituito da quattro parti o sotto-linguaggi specializzati.

1) DDL – **D**ata **D**efinition **L**anguage comprende i comandi SQL che consentono la creazione delle strutture dati e le loro relazioni e la definizione dai dati in esse contenuti
2) DML - **D**ata **M**anipulation **L**anguage comprende tutti i comandi che consentono la manipolazione dei dati in termini di aggiornamento, inserimento, eliminazione e modifica
3) DCL - **D**ata **C**ontrol **L**anguage comprende quei comandi con i quali vengono gestiti i permessi di accesso alle risorse del database
4) QL – **Q**uery **L**anguage – comprende quei comandi che consentono l'interrogazione, il raggruppamento, il conteggio ed in genere l'acquisizione dei dati presenti nel database.

Di ciascuna query possiamo dire che:

- **comando** è il blocco di codice SQL che termina con **;**
- **costrutto** è invece la tipologia di query effettuata (**SELECT**, **CREATE**, **INSERT**, …)
- **clausola** sono i parametri aggiuntivi del costrutto (**FROM**, **WHERE**, **IN**, **ON**, …)

Creare una tabella – CREATE TABLE

L'istruzione per la creazione di una tabella è:

CREATE TABLE <nome della tabella> **(**

<nome campo> <tipo del campo> **(**<eventuale lunghezza>**));**

Oltre all'inserimento dei campi si possono anche specificare la chiave primaria, gli indici, eventuali vincoli ed altre caratteristiche che si vogliono dare alla tabella.

Esempio 1

```
CREATE TABLE persona (
id int(11) NOT NULL,
nome varchar(255) NOT NULL,
cognome varchar(255) NOT NULL,
eta int(11) NOT NULL);
```

Modificare una tabella – ALTER TABLE

L'istruzione per modificare una tabella è:

ALTER TABLE <nome della tabella> **<operazione>**

<eventuale parte da modificare> <nome ed eventuali parametri>**;**

Le più comuni operazioni sono l'aggiunta o l'eliminazione di un campo, di un indice, di una chiave oppure cambiare il nome all'intera tabella.

Esempio 2

Aggiungo la chiave primaria alla tabella dell'**esempio 1**

```
ALTER TABLE persona  ADD PRIMARY KEY (id) ;
```

Esempio 3

Imposto il campo id della tabella dell'**esempio 1** affinché si autoincrementi

```
ALTER TABLE persona MODIFY
id int(11) NOT NULL AUTO_INCREMENT ;
```

Eliminare di una tabella - DROP

Per eliminare del tutto una intera tabella si utilizza l'istruzione:

DROP <nome della tabella>**;**

Esempio 4

Elimino la tabella dell'**esempio 1**

```
DROP persona;
```

Inserire dei dati - INSERT

L'istruzione per inserire delle nuove righe in una tabella è:

INSERT INTO <nome della tabella>**(**<nomi dei campi separati da virgola>**) VALUES (**<valori dei campi separati da virgola>**);**

Esempio 5

Inserisco dati nella tabella dell'**esempio 1**
INSERT INTO persona (nome, cognome, eta) VALUES ("Mario", "Rossi", 21**);**

Eliminare dati - DELETE

L'istruzione per eliminare tutte le righe di una tabella è:

DELETE FROM <nome della tabella>**;**

Se dalla tabella si vogliono eliminare le sole righe che soddisfano una determinata condizione l'istruzione diviene:

DELETE FROM <nome della tabella> **WHERE** <condizione>**;**

Esempio 6

Elimino tutti i dati dalla tabella dell'**esempio 1**
DELETE FROM persona;

Esempio 7

Elimino dei dati specifici dalla tabella dell'**esempio 1**
DELETE FROM persona WHERE nome="Mario";

Aggiornare i dati - UPDATE

L'istruzione per eliminare aggiornare tutte le righe di una tabella è:

UPDATE <nome tabella> **SET** <nome del campo>**=**<valore>**;**

Se della tabella si vogliono aggiornare le sole righe che soddisfano una determinata condizione l'istruzione diviene:

UPDATE <nome tabella> **SET** <nome del campo>**=**<valore>**;**

Esempio 8

UPDATE persona SET nome=”Gianmario” **WHERE id=1;**

Le interrogazioni - SELECT

L’interrogazione di un database per consultare le tabelle ed ottenere dei risultati viene effettuata utilizzando la seguente sintassi:

SELECT <nome dei campi>

FROM <nome tabella>

WHERE <condizione>

GROUP BY <nome campo oppure espressione>

HAVING <condizione>

ORDER BY <nome campo oppure espressione>

Con questo comando si esprimono tre operazioni fondamentali quali:

- selezione
- proiezione
- congiunzione

La selezione in quanto consente di selezionare le righe della tabella che si vogliono cercare.

La proiezione attraverso i campi che si vuol selezionare in quanto consente di fare una estrazione “verticale” dalla tabella.

La congiunzione infine in quanto esprime una relazione tra tabelle che può essere ottenuta tramite appositi costrutti “JOIN”.

Nei seguenti esempi si ha che nella prima query viene enfatizzata la selezione nella seconda query la proiezione e nella terza la congiunzione:

1. SELECT * FROM persone WHERE comune=’ORUNE’

2. SELECT nome, cognome FROM persone WHERE comune='ORUNE'
3. SELECT nome, cognome, telefono, email, cellulare FROM persone, contatti WHERE persone.id = contatti.fk_id_persona

Gli operatori di confronto

Nelle query sono spesso presenti delle espressioni condizionali per i quali si devono utilizzare degli operatori di confronto. La condizione è formata da tre parti:

1. campo oppure espressione
2. operatore di confronto
3. campo oppure espressione oppure costante

OPERATORE	SIGNIFICATO
=	uguale
<>	diverso
<	minore
>	maggiore
<=	minore o uguale
>=	maggiore o uguale
LIKE	confronto stringhe (inizia, contiene, finisce)
BETWEEN	compreso tra due valori

L'operatore LIKE si utilizza utilizzando dei caratteri jolly.

Esistono inoltre due operatori particolari IN ed IS NULL il primo si utilizza quando un campo può assumere un insieme di valori e ci consente di abbreviare la condizione mentre il secondo consente di individuare i campi che assumono un valore nullo.

Ad esempio la query:

SELECT * FROM persona WHERE PROVINCIA='MI' OR PROVINCIA='NA' OR PROVINCIA='RM' OR PROVINCIA='PA'

Usando l'operatore IN diviene:

SELECT * FROM persona WHERE PROVINCIA IN ('MI','NA','RM','PA')

Nelle query si possono utilizzare anche gli operatori matematici per le quattro operazioni, la potenza **^** e la radice quadrata sqr(valore)

Le congiunzioni

La congiunzione (**JOIN**) è l'associazione tra un campo di una tabella e un campo dello stesso tipo di dati in un'altra tabella.

Le congiunzioni esterne (OUTER JOIN)

La congiunzione esterna (**OUTER JOIN**) è la congiunzione in cui tutti i record di una tabella vengono aggiunti al risultato anche se non vi sono valori corrispondenti nella seconda tabella con cui è stata creata la JOIN mentre i record della seconda tabella vengono combinati con quelli della prima solo se ci sono valori corrispondenti nei campi sui quali è stata creata la congiunzione.

La congiunzione esterna può essere:

- **LEFT JOIN** è la congiunzione in cui tutti i record della prima tabella (quella a sinistra) vengono aggiunti al risultato anche se non vi sono valori corrispondenti nella seconda tabella con cui è stata creata la JOIN mentre i record della seconda tabella vengono combinati con quelli della prima solo se ci sono valori corrispondenti nei campi sui quali è stata creata la congiunzione
- **RIGHT JOIN** è la congiunzione in cui tutti i record della seconda tabella (quella a destra) vengono aggiunti al risultato anche se non vi sono valori corrispondenti nella prima tabella con cui è stata creata la JOIN mentre i record della prima tabella vengono combinati con quelli della seconda solo se ci sono valori corrispondenti nei campi sui quali è stata creata la congiunzione
- **FULL JOIN** è la congiunzione in cui tutti i record della prima tabella vengono aggiunti al risultato anche se non vi sono valori corrispondenti nella prima tabella con cui è stata creata la JOIN così come i record della seconda tabella vengono aggiunti anche se non ci sono valori corrispondenti nei campi sui quali è stata creata la congiunzione. La FULL JOIN coincide col prodotto cartesiano

La congiunzione interna (INNER JOIN)

La congiunzione interna (**INNER JOIN**) è la congiunzione in cui vengono aggiunti al risultato i soli record di entrambe le tabelle che hanno valori corrispondenti nei campi sui quali è stata creata la congiunzione.

L'auto congiunzione è una congiunzione che avviene utilizzando una sola tabella che si congiunge con sé stessa.

La congiunzione multipla è una congiunzione che avviene tra più tabelle.

Gli operatori aggregati

Le operazioni di aggregazione consentono di raggruppare più record in modo da poter effettuare calcoli od operazioni specifiche ed a differenza degli operatori matematici agiscono su più record.

I principali operatori sono:

- **AVG** – calcola la media aritmetica
- **COUNT**– conta il numero di record
- **MAX** – calcola il valore massimo
- **MIN** – calcola il valore minimo
- **SUM** – calcola la somma aritmetica
- **STDEV** – calcola la deviazione standard

Se si vuol utilizzare questi operatori si verificano due possibilità:

1. il risultato è costituito solo dal dato ottenuto con l'operatore
2. il risultato non è costituito solo dal dato ottenuto con l'operatore quindi occorre sempre utilizzare la clausola **GROUP BY** visto che si perdono delle componenti in termine di righe.

La clausola GROUP BY serve per raggruppare e per elaborare in modo uniforme diverse righe che nella tabella origine hanno valori uguali per determinate colonne.

Se ad esempio abbiamo una tabella FATTURE(id, PARTITA_IVA, CIFRA, …) per il calcolo del totale delle fatture raggruppate per partita IVA avremo la query:

SELECT PARTITA_IVA, **SUM**(CIFRA) **FROM** FATTURE

GROUP BY PARTITA_IVA

Se poi si vuol conoscere il totale per una specifica partita IVA si deve utilizzare la clausola HAVING come segue:

SELECT PARTITA_IVA, **SUM**(CIFRA) **FROM** FATTURE

GROUP BY PARTITA_IVA **HAVING** PARTITA_IVA=394828221