

# SQL 04/09/2024

SQL linguaggio per database relazionale

useremo un DBMS, software per gestire database. Useremo MAMP

MAMP: ambiente server locale gratuito, esegue MySQL

MySQL: rdbms, software libero, conforme agli standard ANSI SQL, ODBC SQL; compatibile con python, moldi "dialetti"

linguaggio standard per gestire database: tabelle di dati

l'interrogazione al databes avviene tramite QUERY: epressioni/richieste

```
SELECT * FROM world.country; /*importa tutto*/
SELECT Code, Name FROM world.country; /* importa colonne Code e
UPDATE
DELETE
INSERT INTO
CREATE DATABASE
ALTER DATABASE
CREATE INDEX
DROP INDEX

CREATE TABLE Students(
    StudentID int PRIMARY KEY,
    FirsstName VARCHAR(50), /*stringa di 50 caratteri*/
    LastName VARCHAR(50)
);
```

Sintassi molto specifica, non è case sensitive per i comandi

Alcuni sistemi dopo query vogliamo ; per inserire più istruzioni

Schema RELAZIONALE: struttura e relazioni tra tabelle

CHIAVI PRIMARIE: attributo/i che identifica univocamente le righe di una tabella; può avere uno o più campi

CHIAVE ESTERNA: permette relazione tra tabelle

```
CREATE TABLE Enrolmentss(  
    EnrolmentID int PRIMARY KEY,  
    Name VARCHAR(50), /*stringa di 50 caratteri*/  
    FOREIGN KEY (StudentiID)  
    FOREIGN KEY (CorsiID)  
);  
-- questo è un commento
```

Operazioni solo per AMMINISTRATORI

```
CREATE DATABASE database_name  
-- questo è un commento
```

SQL non è dinamico, bisogna sempre specificare i TIPI di dato

```
CREATE TABLE Persona(  
    PersonID int PRIMARY KEY  
    Name varchar(224)  
);
```

```
CREATE new_table  
    SELECT PersonID, column2  
    from Persona
```

PRINCIPALI TIPI IN SQL:

```
CHAR(size) --stringa di grandezza fissa(max 225)  
VARCHAR(size) --stringa di grandezza variabile(max 65535)  
MEDIUMTEXT -- fino a 16 777 215 caratteri  
LONGTEXT -- fino a 4 294 967 295 caratteri  
ENUM(val1, val2,...) --lista di valori  
BOOL -- 0 e 1  
INT
```

```
FLOAT
DATE --data in formato YYYY-MM-DD
```

Eliminare tabella:

```
DROP TABLE nome_tabella --elimina la tabella
TRUNCATE TABLE nome_tabella -- elimina i dati ma non la struttura
ALTER TABLE nome_tabella -- modifica struttura
ADD nome_colonna datatype -- aggiunge colonna
```

Possiamo avere VINCOLI

```
CREATE TABLE tabella(
    column1 datatype constraint,
    ...);

--Vincoli più comuni
NOT NULL --mai valori nulli
UNIQUE --dati univoci
PRIMARY (FOREIGN) KEY
CHECK --condizione
DEFAULT -- VALORE DI DEFAULT
CREATE INDEX -- crea un indice per recuperare i dati
AUTO_INCREMENT --aumento automatico

--esempi
CREATE TABLE Persona(
    ID int NOT NULL,
    Age int
    ...);

--si possono aggiungere i vincoli dopo
ALTER TABLE Persona
MODIFY Age int NOT NULL;
CONSTRAINT ctn_name PRIMARY KEY UNIQUE (ID)
```

## Istruzione SELECT:

```
SELECT * FROM table_name; --prende tutte le colonne
SELECT column1, column2 FROM table_name;

SELECT DISTINCT column1
from table_name; --solo elementi univoci

-- filtro condizione
SELECT column1
from table_name
where condition;
--esempio
SELECT * FROM world.country
WHERE Region = 'Antartica';

SELECT name, SurfaceArea FROM world.country
WHERE SurfaceArea > 193;
```

## Condizioni con il WHERE:

```
= --uguale
>
<
!= oppure <> --diverso
BETWEEN 0 and 100 --valori compresi
LIKE 's%' --valori che iniziano per s, ci sono altri pattern
IN(0,100,1000,10000) --valori nell'elenco
--null
WHERE column_name IS NULL
WHERE column_name IS NOT NULL
--operatori logici come python
WHERE condition1 AND condition2;
WHERE condition1 AND (condition2 OR NOT condition3)
```

## Order BY:

```
select colonna1  
from tabella1  
order by column1 ASC|DESC;
```

```
SELECT * FROM word1.country  
ORDER BY column1 ASC
```

```
SELECT * FROM word1.country  
WHERE condition  
GROUP BY column_name(s)
```

```
SELECT Country, COUNT(CustomerID)  
FROM Customers  
group BY Country  
ORDER BY COUNT(CustomerID) DESC
```

INSERT INTO: inserire i dati

```
INSERT INTO table_name(column1, column2)  
VALUES(value1, value2);  
--inserisco in table_name, value1 in column1 e value2 in column2  
INSERT INTO table_name  
VALUES(value1, value2); --devo mettere però tutti e in ordine
```

UPDATE:

```
UPDATE mome_tabella  
SET column1 = value1, column2 = value2  
WHERE condition
```

```
UPDATE Customers  
SET ContactName = 'AlfredSchmidt', City = 'Frankfurt'  
WHERE CustomerID = 1
```

DELETE

```
DELETE FROM nome_tabella  
WHERE condition
```

```
DELETE FROM Customers  
WHERE CustomerName = 'Mario Rossi'
```

SELECT TOP (LIMIT)

```
SELECT column_name  
FROM table_name  
WHERE condition  
LIMIT number --numero limitato di record
```

```
SELECT * FROM Customers  
LIMIT 50 --solo le prime 50 righe trovate
```

ALIAS

Nome temporaneo ad un elemento

```
SELECT column_name AS alias_column_name  
FROM table_name AS alias_table_name;
```

ESERCIZIO 1:

**Scrivete una query SQL che restituisca solo i record dalla tabella "products" con un prezzo superiore a 50.**

```
SELECT productName  
FROM `classicmodels.products`  
WHERE buyPrice > 50
```

ESERCIZIO 2:

**Scrivete una query SQL che restituisca tutti i record dalla tabella "orders" ordinati per data in ordine decrescente**

```
SELECT *  
FROM classicmodels.orders  
ORDER BY orderDate DESC
```

Functions:

```
SELECT MIN(column_name)  
from table_name  
where conftion;
```

```
SELECT MAX(PRICE) AS LargestPrice  
FROM Products;
```

```
SELECT SUM(column_name)|AVG()|  
FROM table_name;
```

WILDCARD CHARACTERS: usati con WHERE e LIKE

```
'%' --zero o più caratteri  
'b1%'  
"_" -- singolo carattere  
'h_t' --(hot, hat, hit)  
-----  
'[]'--ogni carattere nelle quadre  
'^' --rifiuta i caratteri indicati  
'h[^a]t'  
'-' --range  
'c[a-t]t'--tutti i valori compresi  
  
'_r%'-- ha almeno una r come secondo carattere  
  
SELECT * from Customers  
WHERE CustomerName LIKE 'a%'  
  
SELECT *
```

```

FROM table_name
where column_name IN (SELECT STATEMENT);

SELECT *
FROM Customers
WHERE Country in ('Germany'. 'UK')

SELECT *
FROM Customers
WHERE Country IN (SELECT Country FROM Suppliers)

SELECT *
FROM Products
WHERE Price BETWEEN 10 AND 20
AND CategoryID NOT IN (!, 2, 3)

SELECT *
FROM Orders
WHERE OrderDate BETWEEN '1996-07-01' and '1996-07-31'

```

### ESERCIZIO 3

**Scrivete una query SQL che aggiorni il prezzo di tutti i prodotti nella tabella "products" aumentandolo del 10%**

```

UPDATE classicmodels.products
SET buyPrice = buyPrice + 0.1*buyprice;

```

### ESERCIZIO 4

**Scrivete una query SQL che inserisca un nuovo utente nella tabella "customers"**

```

INSERT INTO classicmodels.customers(customerNumber, customerName,
                                     contactFirstName, phone, addressLine1, addressLine2)
VALUES (497, 'Aldo Corporation', 'Perri', 'Aldo', '3454462120',

```

### ESERCIZIO 5



**Scrivete una query SQL che elimini tutti gli ordini nella tabella "orders" con lo stato "Cancelled".**

```
DELETE FROM classicmodels.orderdetails
WHERE orderNumber IN (SELECT orderNumber FROM classicmodels.orders
                       WHERE status = 'Cancelled');

DELETE FROM classicmodels.orders
WHERE status = 'Cancelled';
```