

Fondamenti di Informatica e Basi di Dati a.a. 2019/2020

DOCENTE: DOTT.SSA VALERIA FIONDA



SQL: DML

Oltre ad i comandi di interrogazione, la parte DML definisce anche le operazioni per la **modifica dell'istanza** della base di dati.

- **insert** → inserisce una o più righe.
- **delete** → cancella una o più righe.
- **update** → aggiorna un attributo o più.

SQL: DML

E' possibile **inserire una riga** esplicitando i valori degli attributi oppure estraendo le righe da altre tabelle del database.

➤ **insert** into NomeTabella
[ListaAttributi] values (ListaValori)

```
INSERT INTO IMPIEGATI(Codice, Nome,  
Cognome, Ufficio) values  
( '8', 'Vittorio', 'Rossi', 'A' )
```

SQL: DML

E' possibile **inserire una riga** esplicitando i valori degli attributi oppure estraendo le righe da altre tabelle del database.

➤ **insert** into NomeTabella
[ListaAttributi] values (ListaValori)

```
INSERT INTO IMPIEGATI(Codice, Nome,  
Cognome) values('8', 'Vittorio'  
, 'Rossi')
```

← Ufficio → non
specificato, NULL
o default

SQLite

Download e installazione

Il software è scaricabile dal seguente link

<https://www.sqlite.org/download.html>

- Una volta scaricato l'eseguibile è sufficiente avviarlo
- Un tutorial su SQLite è scaricabile al seguente link

http://www.laboratoriolibero.com/corsi/jobselect/10/lezione_10_SQLite.pdf

Creare un database

Per creare un database:

```
.open nome_database.db
```

Verrà aperto se esiste oppure creato se non esiste.

Creare una tabella

Per creare una tabella nel database aperto, digitare il corrispondente codice SQL:

```
CREATE TABLE articoli (  
  id INTEGER PRIMARY KEY AUTOINCREMENT, titolo CHAR(200) NOT NULL,  
  categoria VARCHAR(200) NOT NULL,  
  autore VARCHAR(200) NOT NULL, testo TEXT NOT NULL,  
  visite INTEGER NOT NULL  
);
```


Inserire dati in una tabella

Per inserire dei dati in una delle tabelle del database aperto, digitare il corrispondente codice SQL:

```
INSERT INTO articoli (titolo,categoria,autore,testo,visite)  
VALUES ('SQLite è facile','informatica','Antonio','Etc Etc','10');
```

Nel caso si fossero inseriti valori in un campo numerico (INT o FLOAT) bisognerebbe evitare di usare gli apici, altrimenti il valore viene passato non come numero ma come stringa.

Cancellare una tabella

Per cancellare una tabella dal database aperto, digitare il corrispondente codice SQL:

```
DROP TABLE <nome_tabella>;
```

Visualizzare il codice SQL di una base di dati

Per visualizzare il codice sql della base di dati aperta o di una sua tabella, digitare il seguente comando:

```
.schema [<tabella>]
```

SQL: DDL – Esercizio 1

- Scrivere il codice SQL dello schema

ARTICOLI

<u>Codice</u>	Autore	Conferenza	Anno	Formato	Pagine
---------------	--------	------------	------	---------	--------

- Autore deve essere sempre definito.
- Conferenza è una stringa di lunghezza max 8.
- Formato è una stringa di lunghezza 3.
- Il numero di pagine deve essere sempre maggiore di 0, nel caso di omissione si assume pari ad 1.

SQL: DDL – Esercizio 1

```
CREATE TABLE ARTICOLI (  
    CODICE SMALLINT PRIMARY KEY,  
    AUTORE VARCHAR(20) NOT NULL,  
    CONFERENZA VARCHAR(8)  
    FORMATO CHAR(3),  
    ANNO DATE,  
    PAGINE SMALLINT DEFAULT 1  
    CHECK (PAGINE>0)  
);
```

SQL: DDL – Esercizio 1

- Scrivere il codice SQL dello schema

CONFERENZE

<u>Nome</u>	<u>Data</u>	Organizzatore	NumPartecipanti
-------------	-------------	---------------	-----------------

- Vincolo di integrita': ARTICOLO.{Conferenza, Data} → CONFERENZA.{Nome, Data}
- Rimuovendo una conferenza da CONFERENZE, vengono rimossi anche gli articoli corrispondenti

SQL: DDL – Esercizio 1

```
CREATE TABLE CONFERENZE (  
    NOME VARCHAR(8),  
    DATA DATE,  
    ORGANIZZATORE VARCHAR(20),  
    NUMPARTECIPANTI INTEGER CHECK  
    (NUMPARTECIPANTI>0),  
  
    PRIMARY KEY(NOME,DATA)  
);
```

SQL: DDL – Esercizio 1

```
CREATE TABLE ARTICOLI (  
    CODICE SMALLINT PRIMARY KEY,  
    AUTORE VARCHAR(20) NOT NULL,  
    CONFERENZA CHAR(8),  
    FORMATO CHAR(3),  
    DATA DATE,  
    PAGINE SMALLINT DEFAULT 1  
    CHECK (PAGINE>1)  
    FOREIGN KEY (CONFERENZA,DATA)  
    REFERENCES CONFERENZE(NOME, DATA)  
    ON DELETE CASCADE  
);
```


SQL: DDL – Esercizio 1

- Scrivere il codice SQL dello schema

PARTECIPANTE

NomeConf

DataConf

Nome

Cognome

Ruolo

- Vincolo di integrita': PARTECIPANTE.
{NomeConf, DataConf} → CONFERENZA.
{Nome, Data}
- Ogni aggiornamento di {Nome, Data} nella relazione CONFERENZA viene propagato anche alla relazione PARTECIPANTE.

SQL: DDL – Esercizio 1

```
CREATE TABLE PARTECIPANTE (  
    NOMECONF CHAR(8),  
    DATACONF DATE,  
    NOME VARCHAR(20),  
    COGNOME VARCHAR(30),  
    RUOLO CHARACTER(3),  
    PRIMARY KEY (NOMECONF,DATACONF,  
                NOME, COGNOME)  
    FOREIGN KEY (NOMECONF,DATACONF)  
    REFERENCES CONFERENZE(NOME, DATA)  
    ON UPDATE CASCADE  
);
```

SQL: DDL – Esercizio 2

- Definire in SQL i seguenti domini:
1. **Saldo**: intero, compreso tra -200 e 200, valore di default pari a 5.
 2. **NomeUtente**: stringa di max 10 caratteri, valore di default pari al nome dell'utente che esegue le operazioni di inserimento dati.

SQL: DDL – Esercizio 2

➤ Definire in SQL i seguenti domini:

1. **Saldo**: intero, compreso tra -200 e 200, valore di default pari a 5.
2. **NomeUtente**: stringa di max 10 caratteri, valore di default pari al nome dell'utente che esegue le operazioni di inserimento dati.

```
CREATE DOMAIN SALDO AS SMALLINT DEFAULT 5  
CHECK (VALUE<200 AND VALUE> -200)
```

```
CREATE DOMAIN NOMEUTENTE AS VARCHAR(10)  
DEFAULT USER
```

SQL: DDL – Esercizio 3

- Scrivere il codice SQL dello schema seguente:

GIOCATORI

<u>Nome</u>	Cognome	DataNascita	Presenza	Goal	NSquadra
-------------	---------	-------------	----------	------	----------

- Presenza è un intero, valore di default 0.
- Goal è un intero, valore di default 0, e può assumere valori compresi tra 0 e 30.
- Esiste un vincolo di integrità tra GIOCATORI.Nsquadra e SQUADRE.Nome

SQL: DDL – Esercizio 3

- Scrivere il codice SQL dello schema

SQUADRE

<u>Nome</u>	Campionato	Data Fondazione	Città
-------------	------------	-----------------	-------

- Nome è una stringa di lunghezza massima 20 caratteri.
- Campionato è un carattere, e può assumere solo i valori {A,B,C}.
- Città deve essere sempre definito