

SQL: DDL



Create

- Creare un database:

```
CREATE DATABASE [IF NOT EXISTS] nomeDatabase;
```

- Creare una tabella:

```
CREATE TABLE [IF NOT EXISTS] nomeTabella (  
    nomeCampo1 TIPO [VINCOLI],  
    nomeCampo2 TIPO [VINCOLI],  
    ...  
);
```

Tipi di dato più usati

➤ Tipi numerici

➤ INT

➤ Varianti: SMALLINT, MEDIUMINT, BIGINT

➤ FLOAT/DOUBLE

➤ DECIMAL(c, d)

➤ c cifre in totale, di queste d decimali

➤ Tipi testuali

➤ VARCHAR(n), per stringhe a lunghezza variabile, massimo n caratteri

➤ CHAR(n), per stringhe a lunghezza fissa n

➤ TEXT, per testo lungo (massimo 65'535 byte)



Tipi di dato più usati

➤ Tipi data/ora

- DATE, solo la data nel formato YYYY-MM-DD
- TIME, solo l'ora nel formato HH:MM:SS
- DATETIME, data e ora (YYYY-MM-DD HH:MM:SS)
- TIMESTAMP

➤ Tipo booleano

- BOOLEAN o BOOL (=0 falso, ≠0 true)



Vincoli principali

- **NOT NULL:** il campo non può contenere NULL
- **UNIQUE:** non possono esistere due record con lo stesso valore
- **PRIMARY KEY:** identifica univocamente ogni record
 - Implica automaticamente i vincoli NOT NULL e UNIQUE
 - Vincolo applicabile su un solo campo
- **DEFAULT:** per specificare un eventuale valore di default



Auto increment

- **AUTO_INCREMENT**: vincolo per generare automaticamente valori numerici progressivi per una colonna
 - Si usa con tipi numerici interi (INT, BIGINT, ecc.)
 - Definito su una colonna con chiave (PRIMARY KEY o UNIQUE)
 - Parte da 1 e incrementa di 1 ad ogni inserimento



Esempio

```
CREATE TABLE IF NOT EXISTS ordini (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    numeroOrdine VARCHAR(20) NOT NULL UNIQUE,  
    cliente INT NOT NULL,  
    prezzoTot DECIMAL(10, 2) NOT NULL,  
    pagato BOOLEAN DEFAULT FALSE,  
    dataOrdine DATETIME DEFAULT CURRENT_TIMESTAMP,  
    dataConsegna DATE,  
    note TEXT  
);
```



Foreign key

- La foreign key vincola a inserire come valore del campo un valore che corrisponde a un valore del campo chiave di un'altra tabella.

```
CREATE TABLE IF NOT EXISTS ordini (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    numeroOrdine VARCHAR(20) NOT NULL UNIQUE,  
    cliente INT NOT NULL,  
    ...  
    FOREIGN KEY (cliente) REFERENCES clienti(id)  
);
```




Foreign key

```
CREATE TABLE IF NOT EXISTS ordini (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    numeroOrdine VARCHAR(20) NOT NULL UNIQUE,  
    cliente INT NOT NULL,  
    ...  
    FOREIGN KEY (cliente) REFERENCES clienti(id)  
);
```

Tabella clienti

id	nome	cognome
1	Mario	Rossi
2	Laura	Bianchi
3	Giulia	Verdi

Tabella ordini

id	numeroOrdine	cliente
1	ORD001	1
2	ORD002	1
3	ORD003	2



Foreign key

- Ma cosa succede se si elimina un cliente?

FOREIGN KEY (cliente) **REFERENCES** clienti(id)

- **Caso 1.** Il cliente non ha fatto ancora ordini: operazione consentita.

Tabella clienti

id	nome	cognome
1	Mario	Rossi
2	Laura	Bianchi
3	Giulia	Verdi

Tabella ordini

id	numeroOrdine	cliente
1	ORD001	1
2	ORD002	1
3	ORD003	2



Foreign key

- Ma cosa succede se si elimina un cliente?

FOREIGN KEY (cliente) **REFERENCES** clienti(id)

- **Caso 2.** Il cliente ha già effettuato un ordine: eliminazione impedita.

Tabella clienti

id	nome	cognome
1	Mario	Rossi
2	Laura	Bianchi
3	Giulia	Verdi



Tabella ordini

id	numeroOrdine	cliente
1	ORD001	1
2	ORD002	1
3	ORD003	2

Foreign key

► Alternativa 1:

**FOREIGN KEY (cliente) REFERENCES clienti(id)
ON DELETE CASCADE**

► Le righe della tabella ordine correlate al cliente vengono eliminate

id	nome	cognome
1	Mario	Rossi
2	Laura	Bianchi
3	Giulia	Verdi



id	numeroOrdine	cliente
1	ORD001	1
2	ORD002	1
3	ORD003	2

Foreign key

- Alternativa 2 (se il campo cliente può essere NULL):

FOREIGN KEY (cliente) **REFERENCES** clienti(id)
ON DELETE SET NULL

- Gli ordini effettuati rimangono, ma senza il riferimento al cliente

Tabella clienti		
id	nome	cognome
1	Mario	Rossi
2	Laura	Bianchi
3	Giulia	Verdi



Tabella ordini		
id	numeroOrdine	cliente
1	ORD001	\neq NULL
2	ORD002	\neq NULL
3	ORD003	2



Foreign key

► Caso di default:

FOREIGN KEY (cliente) **REFERENCES** clienti(id)
ON DELETE RESTRICT --analogamente se omissso

Tabella clienti

id	nome	cognome
1	Mario	Rossi
2	Laura	Bianchi
3	Giulia	Verdi



Tabella ordini

id	numeroOrdine	cliente
1	ORD001	1
2	ORD002	1
3	ORD003	2



Foreign key

- Con l'opzione ON UPDATE è possibile specificare il comportamento in caso l'id del cliente cambi.

```
CREATE TABLE IF NOT EXISTS ordini (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    numeroOrdine VARCHAR(20) NOT NULL UNIQUE,  
    cliente INT NOT NULL,  
    ...  
    FOREIGN KEY (cliente) REFERENCES clienti(id)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE  
);
```



Check

- **Check** è usato per limitare il valore dei dati inseribili in un campo

```
CREATE TABLE IF NOT EXISTS ordini (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    ...  
    prezzoTot DECIMAL(10, 2) NOT NULL CHECK (prezzoTot >= 0),  
    ...  
    dataOrdine DATETIME DEFAULT CURRENT_TIMESTAMP,  
    dataConsegna DATE,  
    ...  
    CONSTRAINT chk_date  
        CHECK (dataConsegna >= dataOrdine)  
);
```




Drop / Truncate

- **Drop**, per eliminare database, tabelle, altro:

```
DROP DATABASE [IF EXISTS] nomeDatabase;
```

```
DROP TABLE [IF EXISTS] nomeTabella;
```

```
DROP USER [IF EXISTS] nomeUtente;
```

- **Truncate**, per eliminare i dati contenuti nella tabella

```
TRUNCATE TABLE nomeTabella;
```



Alter

- Aggiungere una colonna:

```
ALTER TABLE nomeTabella  
ADD nomeColonna tipoDato;
```

- Eliminare una colonna:

```
ALTER TABLE nomeTabella  
DROP COLUMN nomeColonna;
```

- Modificare una colonna:

```
ALTER TABLE nomeTabella  
MODIFY nomeColonna tipoDato vincoli;
```



Alter

- Rinominare una tabella:

```
ALTER TABLE vecchioNome  
RENAME TO nuovoNome;
```

- Rinominare una colonna:

```
ALTER TABLE nomeTabella  
RENAME COLUMN vecchioNome TO nuovoNome;
```



Alter

➤ Aggiungere/eliminare vincoli:

```
ALTER TABLE nomeTabella  
ADD PRIMARY KEY (nomeColonna);
```

```
ALTER TABLE nomeTabella  
DROP PRIMARY KEY;
```

```
ALTER TABLE nomeTabella  
ADD CONSTRAINT nomeVincolo FOREIGN KEY (colonna)  
REFERENCES altraTabella(colonna);
```

```
ALTER TABLE nomeTabella  
DROP CONSTRAINT nomeVincolo;
```