

# Il linguaggio SQL

## Tipi di dato predefiniti:

- **char(n)** stringa di “n” caratteri.
- **varchar(n)** stringa di al più “n” caratteri.
- **int** numero intero con segno.
- **smallint** numero intero corto con segno.
- **bigint** numero intero lungo con segno.
- **numeric(n, d)**  
numero decimale di “n” cifre, “d” delle quali dopo la virgola.
- **real** numero reale in semplice precisione.
- **double** numero reale in doppia precisione.
- **float(n)** numero reale con una precisione di almeno “n” cifre decimali.
- **date** data del calendario in formato 'aaaa-mm-gg'.
- **time** ora del giorno in formato 'oo:mm:ss'.
- **timestamp** data e ora in formato 'aaaa-mm-gg oo:mm:ss'.

## **Il comando CREATE TABLE:**

```
CREATE TABLE nome tabella  
    ( definizione di campo  
      [, definizione di campo ] ...  
      [, vincolo d'integrità ] ... );
```

## **Definizione di campo:**

```
nome campo tipo campo [ DEFAULT valore iniziale ]  
    [ NOT NULL ] [ UNIQUE ] [ PRIMARY KEY ]  
    [ REFERENCES nome tabella  
      [ ON DELETE  
        CASCADE | SET DEFAULT | SET NULL ]  
      [ ON UPDATE  
        CASCADE | SET DEFAULT | SET NULL ] ]  
    [ CHECK ( condizione ) ]
```

## Vincoli d'integrità:

- UNIQUE ( nome campo [, nome campo] ...)
- PRIMARY KEY ( nome campo [, nome campo] ...)
- FOREIGN KEY ( nome campo [, nome campo] ...)  
REFERENCES nome tabella  
[ ON DELETE  
CASCADE | SET DEFAULT | SET NULL ]  
[ ON UPDATE  
CASCADE | SET DEFAULT | SET NULL ] ]
- [ CHECK ( condizione ) ]

## I comandi ALTER TABLE e DROP TABLE:

ALTER TABLE nome ADD definizione di campo;

ALTER TABLE nome DROP nome campo;

DROP TABLE nome tabella;

## **Il comando INSERT:**

```
INSERT INTO tabella [ (campo [, campo] ...) ]  
VALUES (valore [, valore] ...);
```

## **Il comando DELETE:**

```
DELETE FROM tabella  
[ WHERE condizione];
```

## **Il comando UPDATE:**

```
UPDATE tabella  
SET campo = valore [, campo = valore] ...  
[ WHERE condizione ];
```

## **Il comando SELECT:**

```
SELECT [ ALL | DISTINCT ]  
      espr [ AS campo] [, espr [ AS campo] ] ...  
FROM espr [ AS alias] [, espr [ AS alias] ] ...  
[ WHERE condizione ]  
[ ORDER BY campo [ ASC | DESC ]  
          [, campo [ ASC | DESC ] ] ...;
```

## Gli operatori relazionali:

< | <= | = | >= | > | <> |

[ NOT ] BETWEEN valore AND valore |

[ NOT ] IN (valore [, valore ] ... ) |

[ NOT ] LIKE pattern |

IS [ NOT ] NULL

## Gli operatori logici:

NOT	unknown	= unknown
-----	---------	-----------

false	AND	unknown	= false
-------	-----	---------	---------

unknown	AND	false	= false
---------	-----	-------	---------

unknown	AND	unknown	= unknown
---------	-----	---------	-----------

unknown	AND	true	= unknown
---------	-----	------	-----------

true	AND	unknown	= unknown
------	-----	---------	-----------

false	OR	unknown	= unknown
-------	----	---------	-----------

unknown	OR	false	= unknown
---------	----	-------	-----------

unknown	OR	unknown	= unknown
---------	----	---------	-----------

unknown	OR	true	= true
---------	----	------	--------

true	OR	unknown	= true
------	----	---------	--------

## Operazioni di manipolazione stringhe:

<u>str</u> <sub>1</sub>    <u>str</u> <sub>2</sub>	concatena “ <u>str</u> <sub>1</sub> ” con “ <u>str</u> <sub>2</sub> ”
LENGTH( <u>str</u> )	ritorna la lunghezza di “ <u>str</u> ”
TRIM( <u>str</u> )	elimina tutti gli spazi all’inizio ed alla fine di “ <u>str</u> ”.
UPPER( <u>str</u> )	converte in maiuscolo tutte le lettere di “ <u>str</u> ”.
LOWER( <u>str</u> )	converte in minuscolo tutte le lettere di “ <u>str</u> ”.

## Operazioni su date e ore:

EXTRACT(field FROM d)

estrae il valore del campo “field” da “d” dove “d” è di tipo DATE, TIME o TIMESTAMP e

“field” = DAY | MONTH | YEAR |  
HOUR | MINUTE | SECOND

## Le funzioni di aggregazione:

- COUNT(espr)    conta il numero di elementi
- AVG(espr)        calcola il valore medio
- MIN(espr)        determina il valore minimo
- MAX(espr)        determina il valore massimo
- SUM(espr)        calcola la somma degli elementi

## Le clausole GROUP BY e HAVING:

```
SELECT [ ALL | DISTINCT ]  
      espr [ AS campo ] [, espr [ AS campo ] ] ...  
FROM espr [ AS alias ] [, espr [ AS alias ] ] ...  
[ WHERE condizione ]  
[ GROUP BY nome [, nome] ...  
[ HAVING condizione ] ]  
[ ORDER BY campo [ ASC | DESC ]  
          [, campo [ ASC | DESC ] ] ...;
```

### **L'operatore NATURAL JOIN:**

espr NATURAL JOIN espr  
[ NATURAL JOIN espr] . . .

### **L'operatore JOIN USING:**

espr JOIN espr USING (campo [,campo] . . .)

### **L'operatore JOIN ON:**

espr JOIN espr ON condizione

### **Gli operatori su insiemi:**

(espr) UNION [ALL] (espr)

(espr) INTERSECT [ALL] (espr)

(espr) EXCEPT [ALL] (espr)



## La clausola WITH:

[ **WITH** tabella [ (campo [, campo] ...) ] **AS** (espr) ]  
SELECT [ ALL | DISTINCT ]  
    espr [ AS campo] [, espr [ AS campo] ] ...  
FROM espr [ AS alias] [, espr [ AS alias] ] ...  
[ WHERE condizione ]  
[ GROUP BY nome [, nome] ...  
[ HAVING condizione ] ]  
[ ORDER BY campo [ ASC | DESC ]  
    [, campo [ ASC | DESC ] ] ...;

## Il comando CREATE VIEW:

**CREATE VIEW** tabella [ (campo [, campo] ...) ]  
**AS** comando select;

## Il comando CREATE TRIGGER:

**CREATE TRIGGER** nome trigger **BEFORE | AFTER**  
**INSERT | DELETE | UPDTE ON** nome tabella  
**WHEN** condizione  
**BEGIN** comando SQL; . . . **END**;

## **Il comando BEGIN TRANSACTION:**

**BEGIN [ TRANSACTION ];**

## **Il comando COMMIT TRANSACTION:**

**END | COMMIT [ TRANSACTION ];**

## **Il comando SAVEPOINT:**

**SAVEPOINT nome savepoint;**

## **Il comando ROLLBACK:**

**ROLLBACK [ TRANSACTION ]**

**[ TO [ SAVEPOINT ] nome savepoint ];**

## **Il comando SET TRANSACTION:**

**SET TRANSACTION ISOLATION LEVEL**

**SERIALIZABLE | REPEATABLE READ |**

**READ COMMITTED | READ UNCOMMITTED**

**[ READ ONLY | READ WRITE ];**