

Learn SQL

Manipulation

[Print Cheat Sheet](#)

Column Constraints

Column constraints are the rules applied to the values of individual columns:

- **PRIMARY KEY** constraint can be used to uniquely identify the row.
- **UNIQUE** columns have a different value for every row.
- **NOT NULL** columns must have a value.
- **DEFAULT** assigns a default value for the column when no value is specified.

There can be only one **PRIMARY KEY** column per table and multiple **UNIQUE** columns.

```
CREATE TABLE student (  
  id INTEGER PRIMARY KEY,  
  name TEXT UNIQUE,  
  grade INTEGER NOT NULL,  
  age INTEGER DEFAULT 10  
);
```

CREATE TABLE Statement

The **CREATE TABLE** statement is used to create a new table in a database. It allows one to specify the name of the table and the name of each column in the table.

```
CREATE TABLE table_name (  
  column_name1 data_type1  
  column_name2 data_type2  
  column_name3 data_type3  
  column_name4 data_type4  
  column_name5 data_type5  
  column_name6 data_type6  
  column_name7 data_type7  
  column_name8 data_type8  
  column_name9 data_type9  
  column_name10 data_type10  
  column_name11 data_type11  
  column_name12 data_type12  
  column_name13 data_type13  
  column_name14 data_type14  
  column_name15 data_type15  
  column_name16 data_type16  
  column_name17 data_type17  
  column_name18 data_type18  
  column_name19 data_type19  
  column_name20 data_type20  
  column_name21 data_type21  
  column_name22 data_type22  
  column_name23 data_type23  
  column_name24 data_type24  
  column_name25 data_type25  
  column_name26 data_type26  
  column_name27 data_type27  
  column_name28 data_type28  
  column_name29 data_type29  
  column_name30 data_type30  
  column_name31 data_type31  
  column_name32 data_type32  
  column_name33 data_type33  
  column_name34 data_type34  
  column_name35 data_type35  
  column_name36 data_type36  
  column_name37 data_type37  
  column_name38 data_type38  
  column_name39 data_type39  
  column_name40 data_type40  
  column_name41 data_type41  
  column_name42 data_type42  
  column_name43 data_type43  
  column_name44 data_type44  
  column_name45 data_type45  
  column_name46 data_type46  
  column_name47 data_type47  
  column_name48 data_type48  
  column_name49 data_type49  
  column_name50 data_type50  
  column_name51 data_type51  
  column_name52 data_type52  
  column_name53 data_type53  
  column_name54 data_type54  
  column_name55 data_type55  
  column_name56 data_type56  
  column_name57 data_type57  
  column_name58 data_type58  
  column_name59 data_type59  
  column_name60 data_type60  
  column_name61 data_type61  
  column_name62 data_type62  
  column_name63 data_type63  
  column_name64 data_type64  
  column_name65 data_type65  
  column_name66 data_type66  
  column_name67 data_type67  
  column_name68 data_type68  
  column_name69 data_type69  
  column_name70 data_type70  
  column_name71 data_type71  
  column_name72 data_type72  
  column_name73 data_type73  
  column_name74 data_type74  
  column_name75 data_type75  
  column_name76 data_type76  
  column_name77 data_type77  
  column_name78 data_type78  
  column_name79 data_type79  
  column_name80 data_type80  
  column_name81 data_type81  
  column_name82 data_type82  
  column_name83 data_type83  
  column_name84 data_type84  
  column_name85 data_type85  
  column_name86 data_type86  
  column_name87 data_type87  
  column_name88 data_type88  
  column_name89 data_type89  
  column_name90 data_type90  
  column_name91 data_type91  
  column_name92 data_type92  
  column_name93 data_type93  
  column_name94 data_type94  
  column_name95 data_type95  
  column_name96 data_type96  
  column_name97 data_type97  
  column_name98 data_type98  
  column_name99 data_type99  
  column_name100 data_type100  
);
```

```
column1 datatype,  
column2 datatype,  
column3 datatype  
);
```

INSERT Statement

The `INSERT INTO` statement is used to add a new record (row) to a table.

It has two forms as shown in the code block:

- Insert values based on the order of the columns in the table.
- Define the columns to insert values into.

```
-- Insert into columns in order:  
INSERT INTO table_name  
VALUES (value1, value2, value3);  
  
-- Insert into columns by name:  
INSERT INTO table_name (column1, column2, column3)  
VALUES (value1, value2, value3);
```

ALTER TABLE Statement

The `ALTER TABLE` statement is used to modify the columns of an existing table.

When combined with the `ADD COLUMN` clause, it is used to add a new column to a table.

```
-- Syntax:  
ALTER TABLE table_name  
ADD column_name datatype;  
  
-- Example:
```

```
ALTER TABLE employees  
ADD first_name TEXT;
```

DELETE Statement

The `DELETE` statement is used to delete records (rows) in a table. This statement does not delete the whole table.

Inside, the `WHERE` clause specifies which record or records that should be deleted. If the `WHERE` clause is omitted, all records will be deleted.

```
DELETE FROM table_name  
WHERE some_column = some_value;
```

UPDATE Statement

The `UPDATE` statement is used to edit records (rows) in a table. It usually includes a `SET` clause that indicates the column to edit and a `WHERE` clause for specifying which record(s) should be updated.

```
UPDATE table_name  
SET column1 = value1, column2 = value2  
WHERE some_column = some_value;
```