**Creating new database**

The CREATE DATABASE statement is used to create a new SQL database.

Syntax

CREATE DATABASE *databasename*;

Example

The following SQL statement creates a database called "testDB":

CREATE DATABASE testDB;

**The SQL DROP DATABASE Statement**

The DROP DATABASE statement is used to drop an existing SQL database.

Syntax

DROP DATABASE ***databasename***;

**Note:** Be careful before dropping a database. Deleting a database will result in loss of complete information stored in the database!

DROP DATABASE Example

The following SQL statement drops the existing database "testDB":

Example

DROP DATABASE testDB;

**The SQL CREATE TABLE Statement**

The CREATE TABLE statement is used to create a new table in a database.

Syntax

CREATE TABLE *table\_name* (

*column1 datatype*,

*column2 datatype*,

*column3 datatype*,

....

);

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar,text, integer, date, etc.).

**The SQL DROP TABLE Statement**

The DROP TABLE statement is used to drop an existing table in a database.

Syntax

DROP TABLE *table\_name*;

**SQL ALTER TABLE Statement**

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

ALTER TABLE *table\_name*

ADD *column\_name datatype*;

ALTER TABLE - DROP COLUMN

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

ALTER TABLE *table\_name*

DROP COLUMN *column\_name*;

## **SQL Create Constraints**

SQL constraints are used to specify rules for data in a table.

Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

### **Syntax**

CREATE TABLE *table\_name* (

*column1 datatype* *constraint*,

*column2 datatype* *constraint*,

*column3 datatype* *constraint*,

....

);

## **SQL Constraints**

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

* **NOT NULL** - Ensures that a column cannot have a NULL value
* **UNIQUE** - Ensures that all values in a column are different
* **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* **FOREIGN KEY** - Uniquely identifies a row/record in another table
* **CHECK** - Ensures that all values in a column satisfies a specific condition
* **DEFAULT** - Sets a default value for a column when no value is specified
* **INDEX** - Used to create and retrieve data from the database very quickly