

SQL Data Base

-@meet_kanth

→ The SQL CREATE DATABASE Statement

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

Syntax:

```
CREATE DATABASE databasename;
```

Example:

```
CREATE DATABASE testDB;
```

→ The SQL DROP DATABASE Statement.

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

Syntax:

```
DROP DATA BASE databasename;
```

Example:

```
DROP DATA BASE testDB;
```

→ The SQL BACKUP DATABASE Statement.

The **BACKUP DATABASE** Statement is used in SQL Server to create a full backup of an existing SQL database.

Syntax :

```
BACK UP DATABASE databasename  
TO DISK = 'file path' ;
```

→ The SQL BACKUP WITH DIFFERENTIAL Statement.

The differential backup only backs up the parts of the database that have changed since the last full database backup.

Syntax :

```
BACKUP DATABASE databasename  
TO DISK = 'file path'  
WITH DIFFERENTIAL;
```

BACKUP DATABASE example:

```
BACKUP DATABASE test DB  
TO DISK = 'D:\backups\testDB.bak' ;
```

BACKUP WITH DIFFERENTIAL Example:

```
BACKUP DATABASE test DB  
TO DISK = 'D:\backups\test DB.bak'  
WITH DIFFERENTIAL;
```

→ 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 data type,  
    column2 data type,  
    column3 data type,  
    .....  
);
```

→ The SQL DROP TABLE Statement.

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

Syntax:

```
DROP TABLE table_name;
```

Example:

```
DROP TABLE Shippers;
```

→ SQL TRUNCATE TABLE

The **TRUNCATE TABLE** statement is used to delete the data inside a table, but not the table itself.

Syntax:

```
TRUNCATE TABLE table_name;
```

→ SQL ALTER TABLE Statement.

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

→ ALTER TABLE - ADD Column Syntax:

```
ALTER TABLE table_name  
ADD column_name data_type;
```

→ ALTER TABLE - DROP COLUMN Syntax:

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

→ SQL Create constraints

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 data type constraint,  
    column2 data type constraint,  
    column3 data type constraint,  
    .....  
);
```


→ SQL NOT NULL Constraint.

The **NOT NULL** constraint enforces a column to NOT accept NULL values.

Example :

```
CREATE TABLE Persons (  
    ID int NOT NULL  
    Last Name varchar(255) NOT NULL,  
    First Name Varchar(255) NOT NULL,  
    Age int  
);
```

→ SQL CREATE INDEX Statement.

The **CREATE INDEX** Statement is used to create Indexes in tables.

→ CREATE INDEX Syntax

```
CREATE INDEX Index_name  
ON table_name (Column1, column2, ....);
```

Example :

```
CREATE INDEX idx_lastname  
ON Persons (LastName);
```

→ CREATE UNIQUE INDEX Syntax:

```
CREATE UNIQUE INDEX Index_name  
ON table_name (column1, column2, ....);
```

→ DROP INDEX Statement

The **DROP INDEX** Statement is used to delete an index in a table.

MS Access:

```
DROP INDEX Index_name ON table_name;
```

SQL SERVER:

```
DROP INDEX table_name.index_name;
```

DB2/Oracle:

```
DROP INDEX Index_name;
```

My SQL:

```
ALTER TABLE table_name
```

```
DROP INDEX index_name;
```