# Unit-5

## Creating and Altering Database and Tables

## * Introduction to Sql:

-> Sql is an ANSI(American National Standards Institute) standard computer language for accessing and manipulating database systems.

-> Sql works with database programs like MS-Access, DB2, MS SQL server, Oracle, Sybase etc.

* What can Sql do?
-> Sql can execute queries against a database.
-> Sql can retrieve data from a database.
-> Sql can insert records in a database.
-> Sql can update records in a database.
-> Sql can create new database.
-> Sql can create views in a database.
-> Sql can set permissions on tables, procedures and views.

*DDL(Data Definition Language);
-> It includes changes to the structure of table like creation of tables, altering tables, deleting tables etc. All DDL commands are auto-committed that means it saves all the changes permanently in the database.

* Create command:
-> Create is a DDL Sql command used to create a table or a database in relational database management system
* Creating a Database:
-> To create a database in RDBMs. create command is used.
syntax:

```
create database <DB-name>;
eg: create database test;
```

-> The above command will create a database named Test, which will be an empty schema without any table.
-> To create tables in this newly created database, we can again use the create command.

* Sql Drop Database statement:
-> This is used to drop an existing Sql database.
Syntax:

```
drop database db_name;
drop database test;
```

* Creating a table:
Syntax:

```
create table <table-name>(column_name1 data-type, column2 datatype,.....);
eg: create table student(s_id  INT, name varchar(20), age int);
```

*Sql drop table system:
eg: drop table student;

*Alter Commnad:

-> This is used for altering the table structure , such as :

- to add a column to existing
- to rename existing table
- to change dataype of any column or the modify size.
- to drop a column table from table.

*Alter command: Add a new column:

-> Using alter command, we can add a column to any existing table.

Sysetm:

Alter table table_name ADD( column_name datatype);

eg: Alter table student add ph_no varchar(200);

* Alter command: Add multiple new columns:

-> Using alter command, we can even add multiple new columns to any existing table.

Syntax:

alter table student add column1 datatype, column datatype2;

eg; alter table student add fathername varchar(20), mothername varchar(20);

* Alter command: add column with default value:

-> This command can add a new column to an existing table with a default value too.

syntax: alter table table_name add column_name datatype1 default somevalue;

eg: alter table student add dob date Default '01-Jan-99';

* Alter table- alter/modify column:

-> To change the datatype of a column in a table, use the followng syntax:

- Sql server/Ms-Access:

  syntax: alter table table-name alter column columname datatype;

- Mysql/oracle(prior version 10G):

  syntax:  alter table table-name modify column column_name datatype:

- Oracle 10G and later:

  syntax: alter table tablename modify column-name datatype:

  eg:1: alter table student alter

  a. column name varchar(10) null;
  b. alter table student modify address varchar(100);

* Alter command: rename a column:

-> You can rename an existing column by using alter command.

eg; alter table student rename address to location;

* Alter command: Drop a column:

-> Alter command can also be used to drop or remove column.

Syntax:

alter table student drop column column-name;

eg: alter table student drop column address;

* Truncate command:
-> Truncate command removes all the records from a table. But this command will not destroy the tables's structure.

Syntax:
truncate table table-name;
eg: truncate table student;
*Rename query:
-> is used to set a new name for an existing table.
eg: rename table stuent to students_info;
* Use database;
Syntax: use databasename;

# Sql Constraints:

-> Constraint restricts the values that the table can store.
-> Sql constraints are used to specify rules for data in a table.
-> Constraints can be specified when the table is created with create table statement or after the table is created with the the alter table statement.

## syntax:

Create table table_name( column1 datatype constraint, Column2 datatype constraint,..................);

* The following constraints are commonly used in sql:

a. NOT NULL:
-> Ensures that a column cannot have a null value.
-> The following sql ensures that the "ID","LastName","FirstName" column will not accesss Null values.
eg: create table student( id int not Null, Lastname varchar(30) not null, firstname varchar(30) not Null, age int);

b. Unique:
-> ensures that all values in a column are different.
-> eg: create table persons( id int not null, address varchar(25) not null, unique(id));

c. Primary Key:
-> A combination of a Not Null and unique uniquely identifies each row in a table.
eg: create table persons (id int not null, lastname varchar(25),firstname varchar(25), primary key(id));

d. Foreign Key:
-> Uniquely identifies a row/record in another table.
eg: Create table orders(orderid int not null, orderno int not null, personid int, primary key(orderid),foreign key(personid) references persons(personid));

e. Check:
-> ensures that all values in a column satisfies a specific condition.
-> It is used to restrict the values before inserting into a table.

eg: create table persons(id int not null, lastname varchar(25) not null, firstname varchar(25), age int, check(age>=18);

f. Default:

-> Sets a default value for a column when no value is specified.

eg: create table person( id int not null, lastname varchar(25) not null, firstname varchar(25), age int, city varchar(25) default ‘Janakpur’);

**Atering Constrainsts:**

1. Primary Key:

   -> To create a primary key constraint on the ‘id’ column when the table is already created, use the following sql:

   syntax: alter table <table-name> add constraint <constraint-name> <constraint-type> <column-name>

   eg: alter table persons add primary key(id);

//alter table persons add constraint pk_emp primary key(id);

-> **To drop a primary key constraint**, use the following sql:

Syntax: alter table <table-name> drop constraint <constraint-name>;

eg: alter table persons drop primary key;

2. Unique Constraint:

   -> To create a unique constraint on the “ID” column when the table is already created, use the following Sql:

   eg: alter table persons add unique(id);

   // alter table person add constraint unique_emp unique(email);

   -> To drop a unique constraint, use the following sql:

   alter table persons drop constraint constraint-name;

   eg: alter table person drop constraint unique_emp;

3. Foreign Key:

   -> To create a foriegn key constraint on the “personid” column when the “orders” table is already created, use the following sql:

   eg: alter table orders add foreign key(personid) references persons(personid);

   -> to drop a foreign key constraint, use the following sql:

   alter table orders drop foreign key foreign_key_name;

   // alter table orders drop constraint constraint-name;

4. Check constraint:

   -> To create a check constraint on the “age” column when the table is already created, use the following sql:

   alter table persons add check(age>=18);

   -> To drop a check constraint, use the following sql:

   alter table persoons drop check age;

5. Default constraint:

   -> To create a default constraint on the “city’ column when the table is already created, use the following sql:

   alter table persons modify city default ‘ktm’;

   -> To drop a default constraints, use the following sql:

   alter table persons alter column city drop default;