Contents

1. Create a Database: 2
2. Use a database: 2
3. Delete a Database: 2
4. Create a Table in Database: 2
5. Drop a Table: 3
6. Truncate a Table: 3
7. SQL Constraints 3
    a. NOT NULL Constraint 3
    b. UNIQUE Constraint 4
    c. Primary Key Constraint 5
    d. Foreign Key Constraint 6
    e. Check Constraint 7
    f. Default Constraint 8
    g. Drop a Constraint 8
8. Add a New Column 8
9. Drop a Column 8
10. Modify a Column 8

1. Create a Database:

In order add data in our database, we first need to create our database. The syntax for creating database is as follows:

```
Create database databasename;
```

Example:

```
Create database myDB;
```

2. Use a database:

To tell the DBMS about the database in which we want to create database tables or add data, we need to use the following syntax after creating our database:

```
Use databasename;
```

3. Delete a Database:

We can delete a database by using the following syntax:

```
Drop database databasename
```

Example:

```
Drop database myDB;
```

The above statement will delete all tables of myDB and their data.

4. Create a Table in Database:

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

```
Create Table table_name (
  column1 datatype,
  column2 datatype,
  column3 datatype,
 ....
);
```

Example:

```
Create Table Student
(

    rollNumber int,
    name nvarchar(100),
    cnic nvarchar(100),
    cgpa float
 )
```

To learn about datatypes in SQL Server visit the following link, and see datatypes under SQL SERVER Data types:

https://www.w3schools.com/sql/sql_datatypes.asp

5. Drop a Table:

A table can be drop using the following syntax: (Dropping a table will delete the table data as well as table definition)

```
Drop Table tableName
```

Example:

```
Drop Table Student
```

6. Truncate a Table:

A table can be truncated using the following syntax: (Truncating a table will only delete the data in the table and not table definition)

```
Truncate Table tableName
```

Example:

```
Truncate Table Student
```

7. SQL Constraints

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

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.

a. NOT NULL Constraint

By default, a column can hold NULL values. The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

Add NOT NULL Constraint Using Create Table:

```
Create Table Student
(
 rollNumber int NOT NULL,
 name nvarchar(50) NOT NULL,
 cgpa float NULL
)
```

Add NOT NULL Constraint Using ALTER Table:

```
Alter Table Student Alter column rollNumber int NOT NULL
```

b. UNIQUE Constraint

The UNIQUE constraint ensures that all values in a column are different.

Add Unique Constraint Using Create Table

```
Create Table Student
(
 rollNumber int NOT NULL,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL unique,
 cgpa float NULL
)
```

Unique constraint can also be defined as follows:

```
Create Table Student
(
 rollNumber int NOT NULL,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL,
 cgpa float NULL
 unique (cnic)
```

Add Unique Constraint Using Alter Table

```
Alter Table Student add constraint UNIQUE_CONSTRAINT_STUDENT_CNIC Unique (cnic)
```

Unique Constraint on a Combination of Columns

Unique constraint can also be applied on a combination of columns:

```
Create Table Student
(
 rollNumber int NOT NULL,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL,
 cgpa float NULL,
 unique (cnic, name)
)
```

(OR)

```
Alter Table Student add constraint UNIQUE_CONSTRAINT_STD_CNIC_NAME Unique (cnic, name)
```

c. Primary Key Constraint

The primary key constraint uniquely identifies each row in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only one primary key, which may consist of single or multiple columns.

Add Primary Key Constraint Using Create Table

```
Create Table Student
(
 rollNumber int primary key,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL,
 cgpa float NULL
)
```

(OR)

```
Create Table Student
(
rollNumber int,
name nvarchar(50) NOT NULL,
cnic nvarchar(50) NOT NULL,
cgpa float NULL
primary key (rollNumber)
)
```

Add Primary Key Constraint Using Alter Table

To add primary key using alter table statement, the column must have not null constraint, otherwise the not null constraint must first be defined.

```
Alter Table Student alter column rollNumber int NOT NULL
Alter Table Student add constraint PK_STUDENT primary key (rollNumber)
```

Composite Primary Key

A primary key involving more than one column is called composite primary key.

Add composite Primary Key Using Alter Table

```
Create Table Student
(
 rollNumber int,
 name nvarchar(50),
 cnic nvarchar(50),
 cgpa float,

 primary key (rollNumber, cnic)
)
```

Add Composite Primary Key Using Alter Table

```
Alter Table Student add constraint PK_STUDENT primary key (rollNumber, cnic)
```

d. Foreign Key Constraint

A foreign key is a column (or a collection of columns) in one table that refers to the primary key in another table. The value of the foreign key comes from the table in which it is the primary key.

Add Foreign Key Constraint Using Create Table

```
CREATE TABLE Student (
    rollNumber int PRIMARY KEY,
    deptId int FOREIGN KEY REFERENCES Department(departmentId)
);
```

(OR)

```
CREATE TABLE Student(
    rollNumber int,
    deptId int,
    PRIMARY KEY (rollNumber),
    FOREIGN KEY (deptId) REFERENCES Department(departmentId)
);
```

Add Foreign Key Constraint Using Alter Table:

```
Alter Table Student add constraint FK_STUDENT foreign key (deptId) references
Department (departmentId)
```

Cascade/No Action/Set NULL:
We can define some actions for a foreign key value whenever there is a change (update or delete) in the primary key value.

If we want to update the foreign key value when the corresponding primary key value is updated, and delete the foreign key value whenever the corresponding primary key value is deleted; we use **cascade** option.

If we want to set the foreign key value to null whenever the corresponding primary key value is updated or deleted, then we use **set NULL** option.

If we want to force the foreign key to remain unchanged, we will use **No Action** option. If No Action is specified, then the primary key value will not be allowed to delete or update in the referenced table if there is a corresponding value of foreign key in the referencing table.

Example:

```
Create Table Student
(
 rollNumber int primary key,
 deptId int,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL,
)
```

```
    Create Table Department
    (

        departmentId int primary key,
        departmentName nvarchar(50)
    )

alter table Student add constraint FK_STUDENT foreign key (deptId) references Department
(departmentId) on delete No Action on update Cascade
```

Composite Foreign key:
A primary key which is composite in the referenced table will form a composite foreign key in the referencing table (all attributes of the composite primary key must be part of the foreign key in such a case).

e. Check Constraint

The check constraint is used to limit the value range that can be placed in a column.

Add Check Constraint Using Create Table

```
Create Table Student
(
 rollNumber int,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL,
 cgpa float NULL check (cgpa>=0 AND cgpa<=4),
)
```

(OR)

```
Create Table Student
(
 rollNumber int,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL,
 cgpa float NULL,
 check (cgpa>=0 AND cgpa<=4)
)
```

Add Check Constraint Using Alter Table

```
alter table Student add Constraint STUDENT_CHECK_CGPA check (cgpa>=0 AND cgpa<=4)
```

Check Constraint can be applied to relate the value of a column with another column:

```
Create Table StudentMarks
(
 rollNumber int,
 obtainedMarks float,
 maximumMarks float,
 check (obtainedMarks >=0 AND obtainedMarks<= maximumMarks)
)
```

f. Default Constraint

Default constraint is used to assign a default value to a column when no value is specified by the user during data insertion.

Add Default Constraint Using Create Table

```
Create Table Student
(
 rollNumber int,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL,
 cgpa float default 0,
 check (cgpa>=0 AND cgpa<=4)
)
```

Add Default Constraint Using Create Table

```
alter table Student add constraint DEFAULT_SUDENT_CGPA default 0 for CGPA
```

g. Drop a Constraint

```
alter table Student drop constraint DEFAULT_SUDENT_CGPA
```

8. Add a New Column

```
Create Table Student
(
 rollNumber int,
 name nvarchar(50) NOT NULL,
 cnic nvarchar(50) NOT NULL
)

alter table Student add cgpa float NOT NULL
```

9. Drop a Column

```
alter table Student drop column cnic
```

10. Modify a Column

```
Create Table Student
(
  rollNumber int,
  name nvarchar(50) NOT NULL,
  cnic nvarchar(50) NOT NULL,
)
alter table  Student alter column rollNumber varchar(100) NOT NULL
```