Primary and Foreign Key

Primary Key (PK): A primary key is a unique column we set in a table to easily identify and locate data in queries.

- A table can have only one primary key, which should be unique and NOT NULL.

Foreign Keys (FK):

- A foreign key is a column used to link two or more tables together.
- A table can have any number of foreign keys, can contain duplicate and NULL values.

Constraints

↳ Constraints are used to specify rules for data in a table.

↳ This ensures the accuracy and reliability of the data in the 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,
    ....
);
```

Commonly used constraints 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.
- FOREIGN KEY: Prevents actions that would destroy links between tables (used to link multiple tables together).
- CHECK – Ensures that the values in a column satisfies a specific condition.
- DEFAULT – Sets a default value for a column if no value is specified.
- CREATE INDEX – Used to create and retrieve data from the database very quickly

Creating Database & Tables

Create Table

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

- Syntax

```
CREATETABLE table-name
(
  column_name1 datatype constraint,
  column_name2 datatype constraint,
  column_name3 datatype constraint,
);
```

- Example

```
CREATE TABLE customer
(
  CustID int8 PRIMARY KEY
  CustName varchar(50) NOT NULL,
  Age int NOT NULL,
  City char(50),
  Salary numeric
);
```

Insert, Update, Delete Values in Table + Alter, Drop & Truncate Table

Insert Values In Table

Syntax

```
INSERT INTO TABLE_NAME
    (column1, column2, column3, .... columnN)
VALUES
    (value1, value2, value3, .... valueN)
```

Example

```
INSERT INTO customer
(CustID, custName, Age, City, Salary)
VALUES
(1, 'Riya', 20, 'Kolkata', 9000),
(2, 'Poiya', 25, 'Delhi', 11000),
(3, 'Amit', 22, 'Mumbai', 6000);
```

Update Values In Table

The UPDATE command is used to update existing rows in a table.

Syntax

```
UPDATE TABLE_NAME
SET "Column-name1" = 'value1', 'Column-name2'=
'value2'
WHERE "ID" = Value'
```

Example

```
UPDATE customer
SET CustName = 'Xam', Age = 32
WHERE CustID = 4;
```

Alter Table

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

- ALTER TABLE - ADD Column Syntax

```
ALTER TABLE table_name
ADD COLUMN column-name;
```

- ALTER TABLE - DROP COLUMN Syntax

```
ALTER TABLE table_name
DROP COLUMN column-name;
```

- ALTER TABLE - ALTER/MODIFY COLUMN Syntax

```
ALTER TABLE table_name
MODIFY COLUMN column-name datatype;
```

Example

ADD Column Syntax: Adding new 'Gender' column to customer table.

```
ALTER TABLE customer
ADD COLUMN Gender varchar(10);
```

ALTER/MODIFY COLUMN Syntax: changing Gender column data type from varchar(10) to char(10)

```
ALTER TABLE customer
ALTER COLUMN Gender char(10);
```

DROP COLUMN Syntax: Deleting Gender column from customer table.

```
ALTER TABLE Customer
DROP COLUMN Gender;
```

Delete Values In Table

The DELETE statement is used to delete existing record in a table.

Syntax

```
DELETE FROM table_name WHERE condition;
```

Example

```
DELETE FROM customer
WHERE CustID = 3;
```

Drop & Truncate Table

The DROP TABLE command deletes a table in the database.

Syntax

```
DROP TABLE table_name;
```

The TRUNCATE TABLE command deletes the data inside a table, but not the table itself.

Syntax

```
TRUNCATE TABLE table_name;
```