2. Constraints

A Constraint is a rule that limits or restricts the type of data that can be entered in a table. In SQL, constraints are used to ensure the integrity of the data in the database.

```
 -- Create the table given below

CREATE TABLE customers (

  id NUMBER(5)  PRIMARY KEY,

  name VARCHAR2(50) NOT NULL,

  email VARCHAR2(100) UNIQUE);
```

i) NOT NULL constraint: This constraint ensures that a column cannot have a NULL value.

--If we try to insert Name attribute with NULL value, we get error message.

```
INSERT INTO customers values(49, '' , 'customer49@example.com');
```

ii) UNIQUE constraint: Ensures that all values in a column are unique (no duplicate values).

```
INSERT INTO customers values(45,'Jeevan','jeevan@example.com');
```

Now, if we try to enter the same email address there will be error message.

This ensures that no two employees can have the same email address.

```
INSERT INTO customers values(46,'Jayanthi','jeevan@example.com');
```

iii) PRIMARY KEY constraint: This constraint ensures that each value in a column is unique and not NULL.

-- id attribute does not allow duplicate values. Here it does not accept value 45 as it already exists.

```
INSERT INTO customers values(45,'Mohan','mohan@example.com');
```

iv) FOREIGN KEY constraint: This constraint creates a link between two tables, where the values in one table must match the values in another table. It Ensures referential integrity by linking a column to the primary key of another table.

```
CREATE TABLE orders (

  id NUMBER(5) PRIMARY KEY,

  customer_id NUMBER(5),

  order_date DATE,

  FOREIGN KEY (id) REFERENCES customers(id));
```

In this example, the "orders" table has a FOREIGN KEY constraint on the "customer_id" column that references the "id" column in the "customers" table. This ensures that the "customer_id" values in the "orders" table match the "id" values in the "customers" table.

example:

```
Insert into orders values(350,425,'15-jan-22');
```

If there is id value with 350 in customers table then we can enter the record otherwise, will not be able to enter this record.

v) CHECK constraint: This constraint ensures that the value in a column meets a specific condition. Here's an example of how to create a table with a CHECK constraint:

```
CREATE TABLE student_grade (

  roll_no  NUMBER(5),

  name VARCHAR2(50),

  grade VARCHAR2(2) CHECK (grade IN ('A', 'B', 'C', 'D')));
```

In this example, the "grade" column is specified with a CHECK constraint that limits the allowed values to 'A', 'B', 'C', 'D' and error message will be displayed if we enter any other values.

```
 Insert into student_grade values(123, 'Jaya','E');
```

vi) DEFAULT constraint: This constraint sets a default value for a column if no value is specified. Here's an example of how to create a table with a DEFAULT constraint:

```
CREATE TABLE Orders (

    Order_ID NUMBER(5) PRIMARY KEY,

    Order_Date DATE DEFAULT CURRENT_DATE );
```

This will set the default value for the "Order_Date" column to the current date whenever a new record is inserted and no value is provided for this column.

```
Insert into orders (order_id) values(75);

select * from orders;
```