DATE : 2/9/2022

TOPIC : **RELATING RECORDS WITH TABLES**

**JOINS AGGREGATION**

Joins are used to join one or more columns from two different tables.It produces values by merging together rows from different related tables.

Aggregation function is used to perform calculation on multiple values by looks at many rows and return the result in a single value

EXAMPLE FUNCTIONS,

* AVG() - Returns the average value
* COUNT() - Returns the number of rows
* FIRST() - Returns the first value
* LAST() - Returns the last value
* MAX() - Returns the largest value
* MIN() - Returns the smallest value
* SUM() - Returns the sum

*SELECT COUNT (username) FROM USERS*

It will return the number of usernames present in the user table.

**CONSTRAINTS**

Constraints are defined by applying some certain conditions on the database.

Constraints are,

NOT NULL,DEFAULT,UNIQUE,PRIMARY KEY,FOREIGN KEY,CHECK,CREATE INDEX

EXAMPLE

CREATE TABLE employee

( emp\_id SERIAL PRIMARY KEY,

ename VARCHAR(30) NOT NULL,

job\_desc VARCHAR(20) DEFAULT 'unassigned',

pan VARCHAR(10) UNIQUE,

CHECK (salary>100000));

In the above query,

For ename the constraint given is NOT NULL it means we cannot add any empty NULL values to the ename there must any value present on the column.

For job\_desc the DEFAULT constraint means the value which are present as NULL will be filled DEFAULT by the statement ‘Unassigned’

UNIQUE means it doesn’t allow the duplicate values to present in that column

**TYPES OF JOIN**

* **INNER JOIN**
* **LEFT JOIN**
* **RIGHT JOIN**
* **FULL JOIN**

**INNER JOIN**

The inner join matches the columns without any NULL values.It only return the rows with values.

Example

***SELECT url,username FROM photos JOIN users on users.id = photos.user\_id;***

The inner join examines each row in the first table PHOTOS .It compares the value in the users on users.id with Photos.user\_id if all the values are equal then the inner join creates a new row that contains columns from both tables and add this new row on result.

**LEFT JOIN**

Left outer join do not throw away any rows with NULL in the source table.Instead of throwing it will fill NULL values in the corresponding empty places in another table.The left join starts selecting data from left table.

**Example**

**SELECT url,username,user\_id**

**FROM photos**

**LEFT JOIN users ON users.id = photos.user\_id ;**

The left join examines the data on photos and will display the photos table by matching the corresponding photos.user\_id column and after that it matches the data on users.If there were any NULL values on users table it will not show that values.

**RIGHT JOIN**

It shows all the perfectly matches columns from both tables and also show the tables with NULL values which is mentioned after RIGHT statement. The Right join starts selecting data from Right table.

**Example**

**SELECT url,username,user\_id**

**FROM photos**

**RIGHT JOIN users ON users.id = photos.user\_id ;**

The right join show all the NULL values present in the table USERS.

**FULL JOIN**

The FULL JOIN returns all records when there is a match in left (table1) or right (table2) table records.

**Example**

**SELECT url,username,user\_id**

**FROM photos**

**RIGHT JOIN users ON users.id = photos.user\_id ;**

In full join it will show all the NULL values in both the tables PHOTOS and USERS