Gehad ElSharkawy

Database

Agenda

Aggregate functions

Inner join

Left join

Right join

Full join

Sum function:
returns the total sum of a numeric column.

-
  ```
  SELECT SUM(column_name)
  FROM table_name
  WHERE condition;
  ```

- Example:
- `select sum(price) as product_price`
- `from product`

Average function:
returns the average value of a numeric column.

-
  ```
  SELECT AVG(column_name)
  FROM table_name
  WHERE condition;
  ```

- Example:

-
  ```
  select avg(price) as product_price
  ```

-
  ```
  from product
  ```

- **Note:** NULL values are ignored.

Count function:
returns the number of rows that matches a specified criterion.

-
  ```
  SELECT COUNT(column_name)
  FROM table_name
  WHERE condition;
  ```

- example;:
- `select count(customerid) as c_ID`
- `from customer`

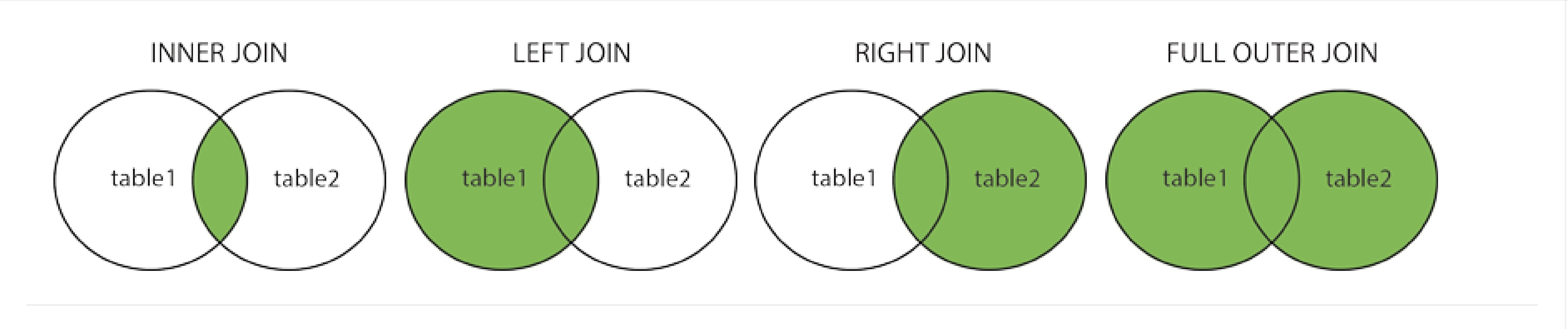
- `select count(phone) as customer_phome`
- `from customer`

- **Note:** NULL values are not counted.

SQL JOIN

- A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Types of SQL JOINs



Types of SQL JOINs

- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table

Syntax

```
SELECT tablename.colname ...etc

FROM table 1

JOIN  table 2

ON table.primaryKey=table.foreignKey;
```

Inner join

- SELECT column_name(s)
- FROM table1
- INNER JOIN table2
- ON table1.column_name = table2.column_name;

Example:

```
SELECT Orders.OrderID, Customer.CustomerN
FROM Orders
INNER JOIN Customer ON Orders.CustomerID =
Customer.CustomerID;
```

Left join

- SELECT *column_name(s)*
  FROM *table1*
  LEFT JOIN *table2*
  ON *table1.column_name* = *table2.column_name*;
- Example:
- SeLECT Customer.CustomerN, Orders.OrderID
- FROM Customer
- LEFT JOIN Orders
- ON Customer.CustomerID = Orders.CustomerID
- Note: The LEFT JOIN keyword returns all records from the left table (Customers), even if there are no matches in the right table (Orders).

Right join

-
  ```
  SELECT column_name(s)
  FROM table1
  RIGHT JOIN table2
  ON table1.column_name = table2.column_name;
  ```
- Example:
- `SELECT Customer.CustomerN, Orders.OrderID`
- `FROM Customer`
- `RIGHT JOIN Orders`
- `ON Customer.CustomerID = Orders.CustomerID`
- Note: The RIGHT JOIN keyword returns all records from the right table (orders), even if there are no matches in the left table (customer).

Full join

-
  ```
  SELECT column_name(s)
  FROM table1
  FULL OUTER JOIN table2
  ON table1.column_name = table2.column_name
  WHERE condition;
  ```
- Example:
- `SELECT Customer.CustomerN, Orders.OrderID`
- `FROM Customer`
- `FULL OUTER JOIN Orders`
- `ON Customer.CustomerID=Orders.CustomerIDORDER BY Customer.CustomerN;`
- Note: The FULL OUTER JOIN keyword returns all matching records from both tables whether the other table matches or not. So, if there are rows in "Customer" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Customer", those rows will be listed as well.

Thank you

Gehad ElSharkawy

Gehad.magdi.mohammed@commerce.Helwan.edu.eg