SQL JOINS

- SQL joins are used to query data from two or more tables, based on a relationship between certain columns in these tables.
- The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.
- Tables in a database are often related to each other with keys.
- A primary key is a column (or a combination of columns) with a unique value for each row. Each primary key value must be unique within the table. The purpose is to bind data together, across tables, without repeating all of the data in every table.

- Look at the "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

- Note that the "P_Id" column is the primary key in the "Persons" table. This means that no two rows can have the same P_Id. The P_Id distinguishes two persons even if they have the same name.

- Next, we have the "Orders" table:

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	15

- Note that the "O_Id" column is the primary key in the "Orders" table and that the "P_Id" column refers to the persons in the "Persons" table without using their names.
- Notice that the relationship between the two tables above is the "P_Id" column.

Different SQL JOINs

- JOIN: Return rows when there is at least one match in both tables
- LEFT JOIN: Return all rows from the left table, even if there are no matches in the right table
- RIGHT JOIN: Return all rows from the right table, even if there are no matches in the left table
- FULL JOIN: Return rows when there is a match in one of the tables

INNER JOIN

- The INNER JOIN keyword returns rows when there is at least one match in both tables.
- **SQL INNER JOIN Syntax**
- ```
  SELECT column_name(s)
  FROM table_name1
  INNER JOIN table_name2
  ON
  table_name1.column_name=table_name2.c
  olumn_name
  ```
- **PS:** INNER JOIN is the same as JOIN.
  ```

Inner Join contd.

- The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

- The "Orders" table:

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	15

Inner join contd.

- Now we want to list all the persons with any orders.
- We use the following SELECT statement:
- ```
  SELECT Persons.LastName,
  Persons.FirstName, Orders.OrderNo
  FROM Persons
  INNER JOIN Orders
  ON Persons.P_Id=Orders.P_Id
  ORDER BY Persons.LastName
  ```
- The result-set will look like this:
  ```

Inner Join

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678

The INNER JOIN keyword returns rows when there is at least one match in both tables. If there are rows in "Persons" that do not have matches in "Orders", those rows will NOT be listed.

LEFT JOIN

- Now we want to list all the persons and their orders - if any, from the tables above.
- We use the following SELECT statement:
- ```
  SELECT Persons.LastName,
  Persons.FirstName, Orders.OrderNo
  FROM Persons
  LEFT JOIN Orders
  ON Persons.P_Id=Orders.P_Id
  ORDER BY Persons.LastName
  ```

- The result-set will look like this:
  ```

Left Join Contd.

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678
Svendson	Tove	

The LEFT JOIN keyword returns all the rows from the left table (Persons), even if there are no matches in the right table (Orders).

RIGHT JOIN

- The RIGHT JOIN keyword returns all the rows from the right table (table_name2), even if there are no matches in the left table (table_name1).
- **SQL RIGHT JOIN Syntax**
- ```
  SELECT column_name(s)
  FROM table_name1
  RIGHT JOIN table_name2
  ON
  table_name1.column_name=table_name2.colu
  mn_name
  ```
- **PS:** In some databases RIGHT JOIN is called RIGHT OUTER JOIN.
  ```

Right Join contd.

- Now we want to list all the orders with containing persons - if any, from the tables above.
- We use the following SELECT statement:
- ```
  SELECT Persons.LastName,
  Persons.FirstName, Orders.OrderNo
  FROM Persons
  RIGHT JOIN Orders
  ON Persons.P_Id=Orders.P_Id
  ORDER BY Persons.LastName
  ```

- The result-set will look like this:
  ```

Right Join contd.

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678
		34764

The RIGHT JOIN keyword returns all the rows from the right table (Orders), even if there are no matches in the left table (Persons).

FULL JOIN

- The FULL JOIN keyword return rows when there is a match in one of the tables.
- **SQL FULL JOIN Syntax**
- ```
  SELECT column_name(s)
  FROM table_name1
  FULL JOIN table_name2
  ON
  table_name1.column_name=table_nam
  e2.column_name
  ```
  ```

Full Join contd.

- Now we want to list all the persons and their orders, and all the orders with their persons.
- We use the following SELECT statement:
- ```
  SELECT Persons.LastName,
  Persons.FirstName, Orders.OrderNo
  FROM Persons
  FULL JOIN Orders
  ON Persons.P_Id=Orders.P_Id
  ORDER BY Persons.LastName
  ```
-
- The result-set will look like this:
  ```

Full Join contd.

LastName	FirstName	OrderNo
Hansen	Ola	22456
Hansen	Ola	24562
Pettersen	Kari	77895
Pettersen	Kari	44678
Svendson	Tove	
		34764

The FULL JOIN keyword returns all the rows from the left table (Persons), and all the rows from the right table (Orders). If there are rows in "Persons" that do not have matches in "Orders", or if there are rows in "Orders" that do not have matches in "Persons", those rows will be listed as well.