Databases

Week 4 - JOINS

SQL Joins

- SQL joins are used 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.
- 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.

SQL Joins

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: “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.

SQL Joins

Now Consider 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 Persons and Orders table is the "P_Id" column.

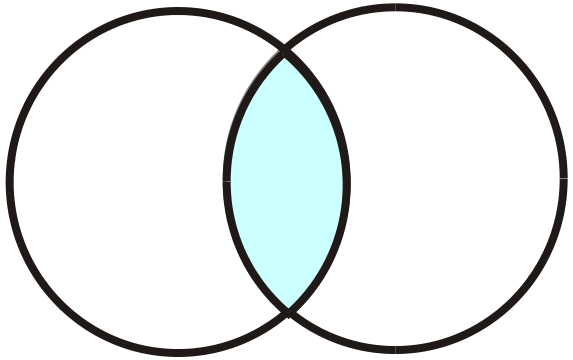
Different SQL JOINs

Before we continue with examples, we will list the types of JOIN you can use, and the differences between them.

- **JOIN (Also Called Inner Join)**: Return rows when there is at least one match in both tables
- **LEFT JOIN(Also called LEFT OUTER JOIN):** Return all rows from the left table, even if there are no matches in the right table
- **RIGHT JOIN(Also called Right OUTER 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

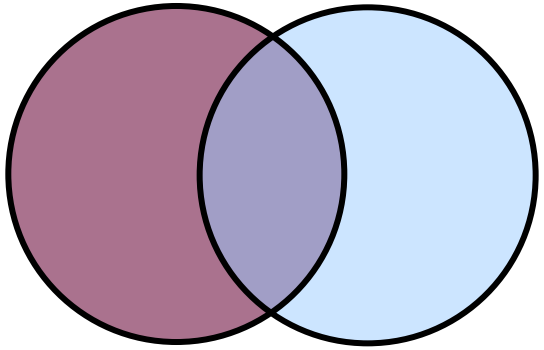
Types of Joins

- Inner joins
  - return only matching rows

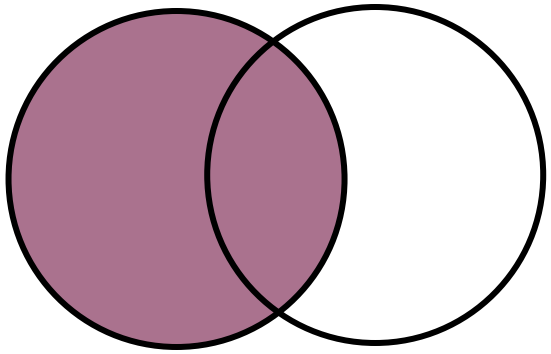


Types of Joins

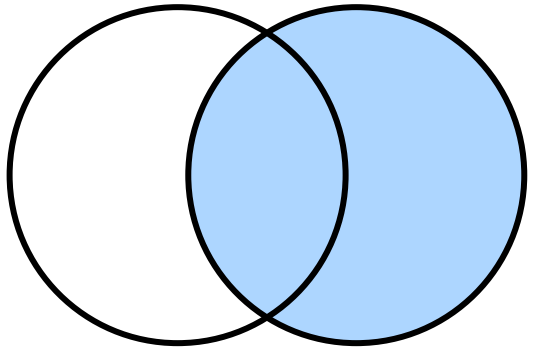
- Outer joins
  - return all matching rows, plus nonmatching rows from one or both tables
  - can be performed on only two tables or views at a time.



Full



Left



Right

SQL INNER JOIN

The INNER JOIN keyword return 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.column_name
```

SQL INNER JOIN

Using the Person and Orders Tables 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 INNER JOIN keyword return rows when there is **at least one match in both tables.**

SQL INNER JOIN

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

"Orders" table:

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

The result-set will look like this:

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.

SQL LEFT JOIN

- The LEFT JOIN (also called LEFT OUTER JOIN) keyword returns all rows from the left table (table_name1), even if there are no matches in the right table (table_name2).

SQL LEFT JOIN Syntax

```
SELECT column_name(s)
        FROM table_name1
        LEFT JOIN table_name2
        ON table_name1.column_name = table_name2.column_name
```

SQL LEFT JOIN

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

SQL LEFT JOIN

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

"Orders" table:

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

The result-set will look like this:

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

SQL RIGHT JOIN

- The RIGHT JOIN (Sometimes called RIGHT OUTER 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.column_name
```

SQL RIGHT JOIN

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 RIGHT keyword returns all the rows from the right table even if there are no matches in the left table.

SQL RIGHT JOIN

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

"Orders" table:

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

The result-set will look like this:

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

SQL 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_name2.column_name
```

SQL FULL JOIN

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 FULL JOIN keyword returns all the rows from the left table (Persons), and all the rows from the right table (Orders).

SQL FULL JOIN

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

"Orders" table:

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

The result-set will look like this:

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

Another SQL join

Run this example

- This example shows how many prescriptions the audiologist made

```
select count(*) from prescriptions
inner join doctors on
prescriptions.doc_id = doctors.doc_id
where speciality = "Audiologist";
```