SQL Server Joins

Summary: in this tutorial, you will learn about various SQL Server joins that allow you to combine data from two tables.

In a relational database, data is distributed in multiple logical tables. To get a complete meaningful set of data, you need to query data from these tables using joins. SQL Server supports many kinds of joins, including inner join, left join, right join, full outer join, and cross join. Each join type specifies how SQL Server uses data from one table to select rows in another table.

Let’s set up sample tables for demonstration.

Setting up sample tables

First, create a new schema named hr:

```
CREATE SCHEMA hr;

GO
```

Second, create two new tables named candidates and employees in the hr schema:

```
CREATE TABLE hr.candidates(

   id INT PRIMARY KEY IDENTITY,

   fullname VARCHAR(100) NOT NULL

);


CREATE TABLE hr.employees(

   id INT PRIMARY KEY IDENTITY,

   fullname VARCHAR(100) NOT NULL

);
```

Third, insert some rows into the candidates and employees tables:

```
INSERT INTO

   hr.candidates(fullname)

VALUES

   ('John Doe'),

   ('Lily Bush'),

   ('Peter Drucker'),

   ('Jane Doe');
```

```
INSERT INTO

   hr.employees(fullname)

VALUES

   ('John Doe'),

   ('Jane Doe'),

   ('Michael Scott'),

   ('Jack Sparrow');
```

Let’s call the candidates table the left table and the employees table the right table.

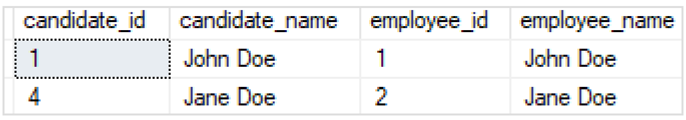
SQL Server Inner Join

Inner join produces a data set that includes rows from the left table, matching rows from the right table.

The following example uses the inner join clause to get the rows from the candidates table that has the corresponding rows with the same values in the fullname column of the employees table:

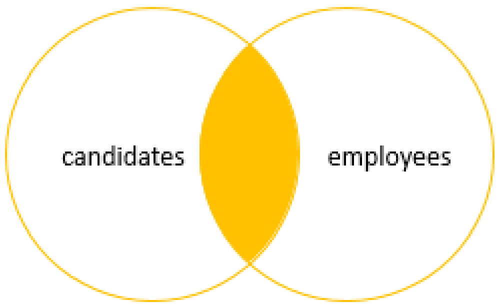
```
SELECT

   c.id candidate_id,

   c.fullname candidate_name,

   e.id employee_id,

   e.fullname employee_name

FROM

   hr.candidates c

   INNER JOIN hr.employees e

      ON e.fullname = c.fullname;
```

Here is the output:



candidate_id	candidate_name	employee_id	employee_name
1	John Doe	1	John Doe
4	Jane Doe	2	Jane Doe

The following Venn diagram illustrates the result of the inner join of two result sets:



SQL Server Left Join

Left join selects data starting from the left table and matching rows in the right table. The left join returns all rows from the left table and the matching rows from the right table. If a row in the left table does not have a matching row in the right table, the columns of the right table will have nulls.

The left join is also known as the left outer join. The outer keyword is optional.

The following statement joins the candidates table with the employees table using left join:

SELECT

c.id candidate_id,

c.fullname candidate_name,

e.id employee_id,

e.fullname employee_name

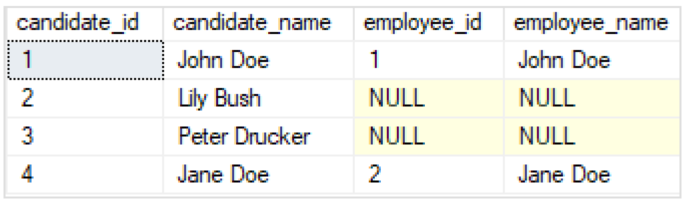
FROM

hr.candidates c

LEFT JOIN hr.employees e

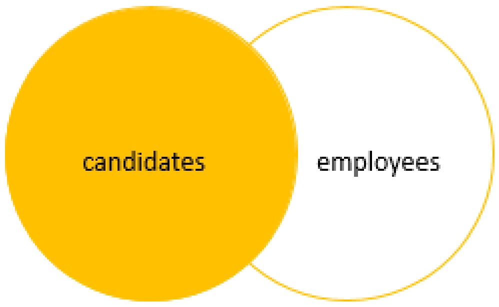
ON e.fullname = c.fullname;Code language: SQL (Structured Query Language) (sql)

Here is the output:



candidate_id	candidate_name	employee_id	employee_name
1	John Doe	1	John Doe
2	Lily Bush	NULL	NULL
3	Peter Drucker	NULL	NULL
4	Jane Doe	2	Jane Doe

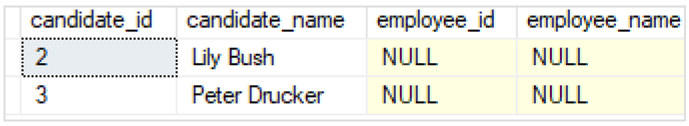
The following Venn diagram illustrates the result of the left join of two result sets:



To get the rows that are available only in the left table but not in the right table, you add a WHERE clause to the above query:

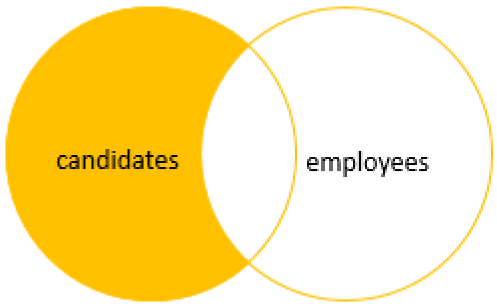
```
SELECT
   c.id candidate_id,
   c.fullname candidate_name,
   e.id employee_id,
   e.fullname employee_name
FROM
   hr.candidates c
   LEFT JOIN hr.employees e
      ON e.fullname = c.fullname
WHERE
   e.id IS NULL;
```

The following picture shows the output:



candidate_id	candidate_name	employee_id	employee_name
2	Lily Bush	NULL	NULL
3	Peter Drucker	NULL	NULL

And the following Venn diagram illustrates the result of the left join that selects rows available only in the left table:



SQL Server Right Join

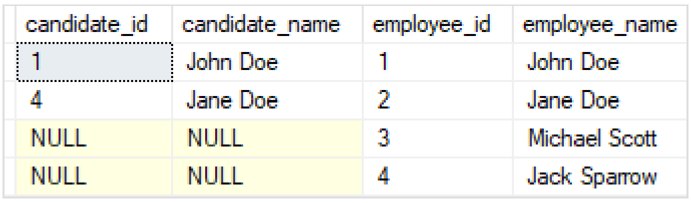
The right join or right outer join selects data starting from the right table. It is a reversed version of the left join.

The right join returns a result set that contains all rows from the right table and the matching rows in the left table. If a row in the right table does not have a matching row in the left table, all columns in the left table will contain nulls.

The following example uses the right join to query rows from candidates and employees tables:

```
SELECT
   c.id candidate_id,
   c.fullname candidate_name,
   e.id employee_id,
   e.fullname employee_name
FROM
   hr.candidates c
   RIGHT JOIN hr.employees e
      ON e.fullname = c.fullname;
```

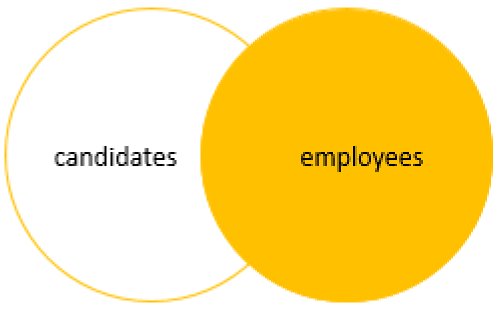
Here is the output:



candidate_id	candidate_name	employee_id	employee_name
1	John Doe	1	John Doe
4	Jane Doe	2	Jane Doe
NULL	NULL	3	Michael Scott
NULL	NULL	4	Jack Sparrow

Notice that all rows from the right table (employees) are included in the result set.

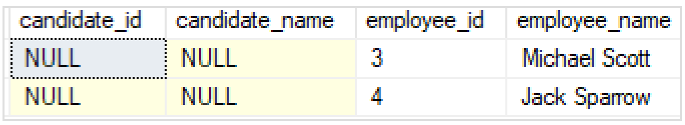
And the Venn diagram illustrates the right join of two result sets:



Similarly, you can get rows that are available only in the right table by adding a WHERE clause to the above query as follows:

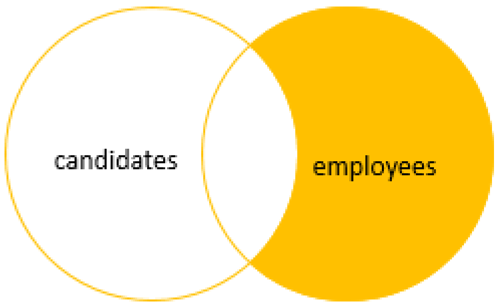
```
SELECT
   c.id candidate_id,
   c.fullname candidate_name,
   e.id employee_id,
   e.fullname employee_name
FROM
   hr.candidates c
   RIGHT JOIN hr.employees e
       ON e.fullname = c.fullname
WHERE
   c.id IS NULL;
```

Here is the output:



candidate_id	candidate_name	employee_id	employee_name
NULL	NULL	3	Michael Scott
NULL	NULL	4	Jack Sparrow

And Venn diagram that illustrates the operation:



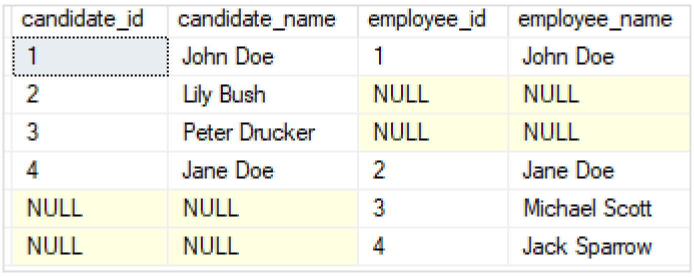
SQL Server full join

The full outer join or full join returns a result set that contains all rows from both left and right tables, with the matching rows from both sides where available. In case there is no match, the missing side will have NULL values.

The following example shows how to perform a full join between the candidates and employees tables:

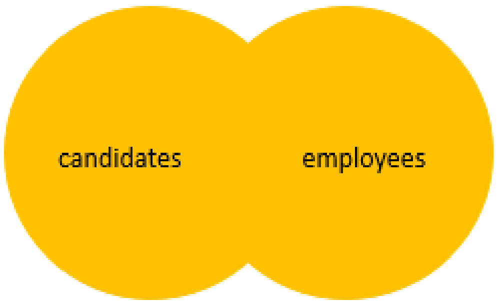
```
SELECT
    c.id candidate_id,
    c.fullname candidate_name,
    e.id employee_id,
    e.fullname employee_name
FROM
    hr.candidates c
    FULL JOIN hr.employees e
        ON e.fullname = c.fullname;
```

Here is the output:



candidate_id	candidate_name	employee_id	employee_name
1	John Doe	1	John Doe
2	Lily Bush	NULL	NULL
3	Peter Drucker	NULL	NULL
4	Jane Doe	2	Jane Doe
NULL	NULL	3	Michael Scott
NULL	NULL	4	Jack Sparrow

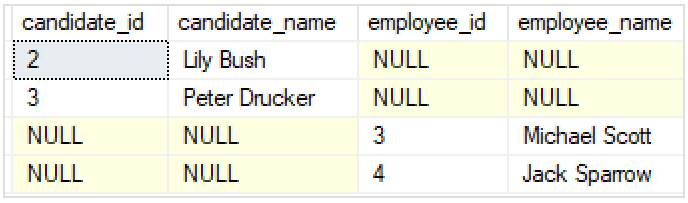
The Venn diagram that illustrates the full outer join:



To select rows that exist either left or right table, you exclude rows that are common to both tables by adding a WHERE clause as shown in the following query:

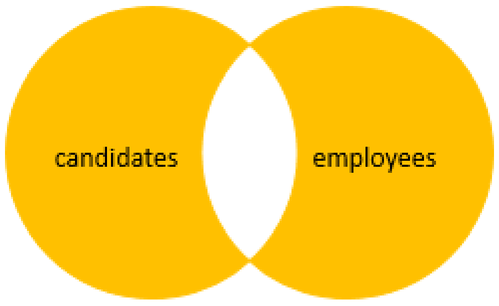
```
SELECT
    c.id candidate_id,
    c.fullname candidate_name,
    e.id employee_id,
    e.fullname employee_name
FROM
    hr.candidates c
    FULL JOIN hr.employees e
        ON e.fullname = c.fullname
WHERE
    c.id IS NULL OR
    e.id IS NULL;
```

Here is the output:



candidate_id	candidate_name	employee_id	employee_name
2	Lily Bush	NULL	NULL
3	Peter Drucker	NULL	NULL
NULL	NULL	3	Michael Scott
NULL	NULL	4	Jack Sparrow

And the Venn diagram illustrates the above operation:



In this tutorial, you have learned various SQL Server joins that combine data from two tables.