Joins in MS-SQL Server

Soyeb Ghachi

Technical Trainer | Mentor |SME

Overview

Joins

-INNER JOIN

-LEFT JOIN

-RIGHT JOIN

-FULL JOIN

JOINS

Setting up sample tables

```
CREATE TABLE CANDIDATES(
id                              int             IDENTITY(1,1)    PRIMARY KEY,
fullname                        varchar(100)      NOT NULL
);
```

```
CREATE TABLE EMPLOYEES(
id                          int         IDENTITY(1,1) PRIMARY KEY,
fullname varchar(100)         NOT NULL
);
```

```
INSERT INTO CANDIDATES VALUES ('John Doe'),('Lily Bush'),('Peter Drucker'),('Jane Doe');

SELECT * FROM CANDIDATES

INSERT INTO EMPLOYEES VALUES ('John Doe'),('Jane Doe'),('Michael Scott'),('Jack Sparrow');
```

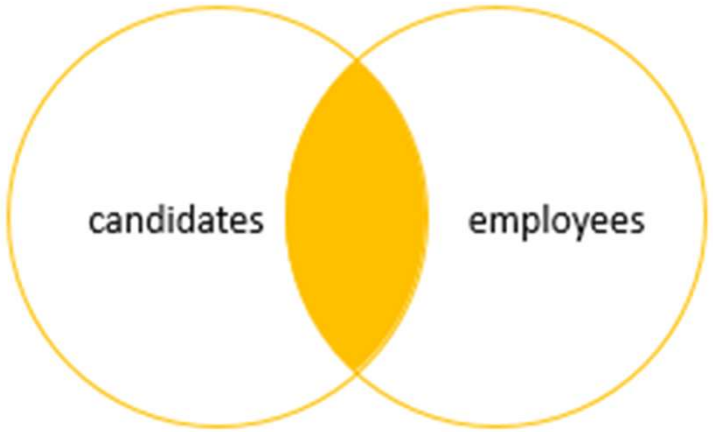
JOINS

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 by using joins.

SQL Server supports many kinds of joins including inner join, left join, right join, full outer join, and cross join.

A) SQL Server Inner Join

Inner Join produces a data set that includes rows from the left table which have matching rows from the right table.




```
SELECT *
FROM table1
INNER JOIN table2
   ON table1.column = table2.column;
```

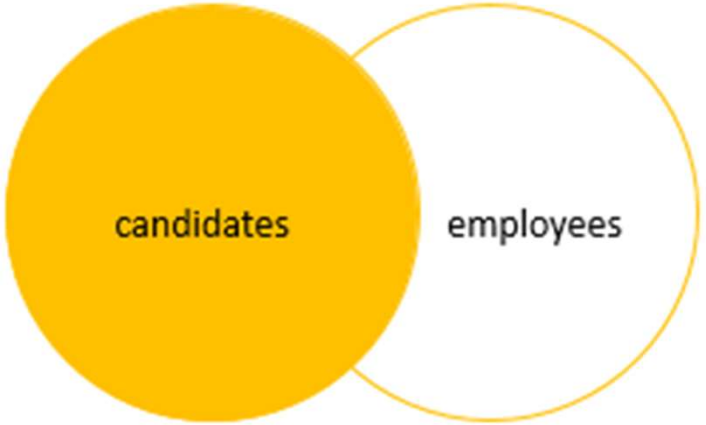
Examples: Inner Join

```
SELECT C.id,C.fullname,E.fullname FROM
CANDIDATES C
INNER JOIN
EMPLOYEES E
ON
C.fullname=E.fullname
```

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

B) 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.




```
SELECT *
FROM table1
LEFT JOIN table2
   ON table1.column = table2.column;
```

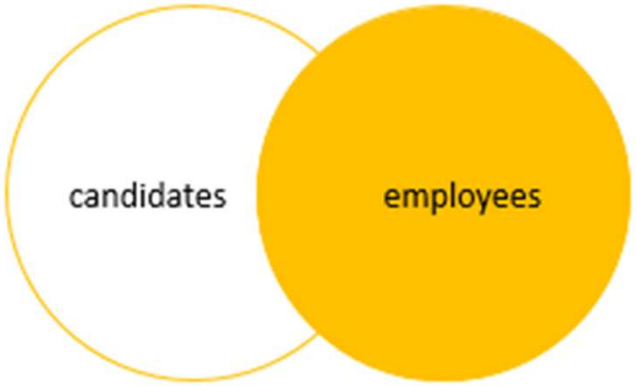
Examples: Left Join

```
SELECT * FROM
CANDIDATES C
LEFT JOIN
EMPLOYEES E
ON
C.fullname=E.fullname
```

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

C) SQL Server Right Join

The right join or right outer join selects data starting from the right table. 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 that does not have a matching row in the left table, all columns in the left table will contain nulls.




```
SELECT *
FROM table1
RIGHT JOIN table2
   ON table1.column = table2.column;
```

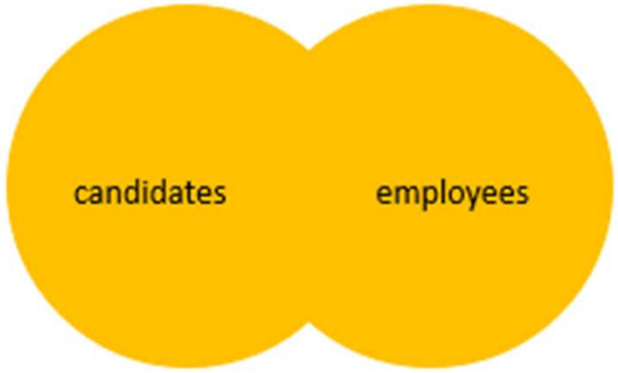
Examples: Right Join

```
SELECT * FROM
CANDIDATES C
RIGHT JOIN
EMPLOYEES E
ON
C.fullname=E.fullname
```

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

C) 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.




```
SELECT *
FROM table1
FULL JOIN table2
   ON table1.column = table2.column;
```

Examples: Full Join

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

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

References

https://learn.microsoft.com/en-us/sql/relational-databases/performance/joins

https://www.w3schools.com/sql/sql_where.asp

https://www.sqlservertutorial.net/sql-server-basics/sql-server-where/

https://www.javatpoint.com/sql-server-joins

https://www.javatpoint.com/sql-server-joins

https://www.guru99.com/sql-server-joins.html

https://learn.microsoft.com/en-us/sql/relational-databases/performance/subqueries