

Chapter 7: More SQL (JOIN)

Database Systems CS203

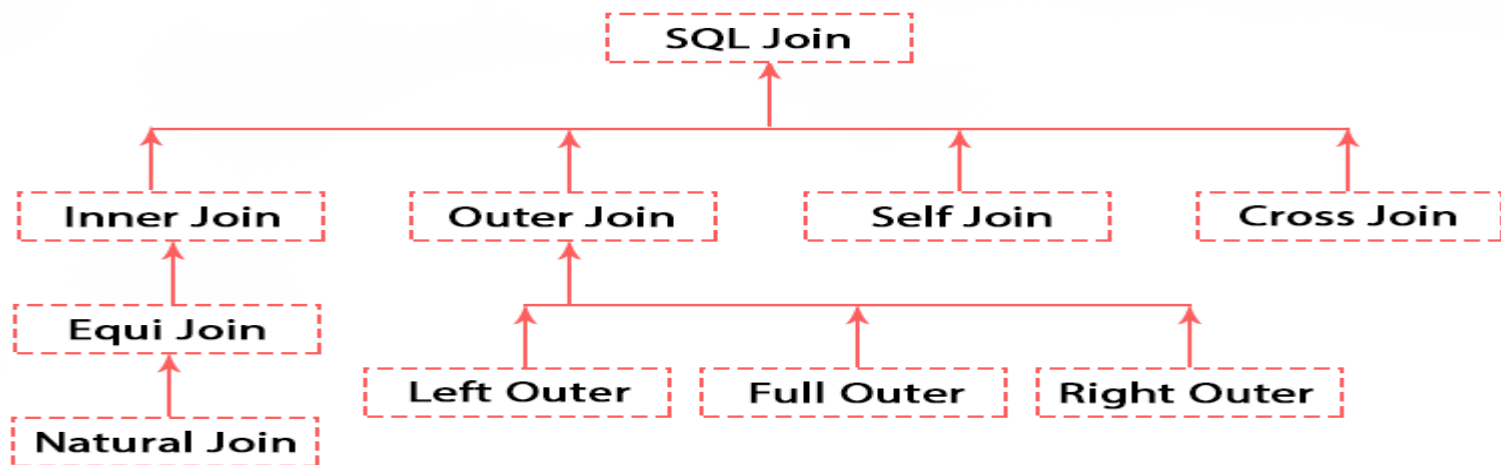


Outline

- JOIN Clause
- Inner Join
- Outer Join
- Cross Join
- Self Join
- Join Vs Subquery

JOIN Clause

- In Relational Database, JOIN is used to combine columns from one or more tables.
- There must be some common identifiers that allow information from multiple tables to be combined easily.



(SQL JOIN TYPES)

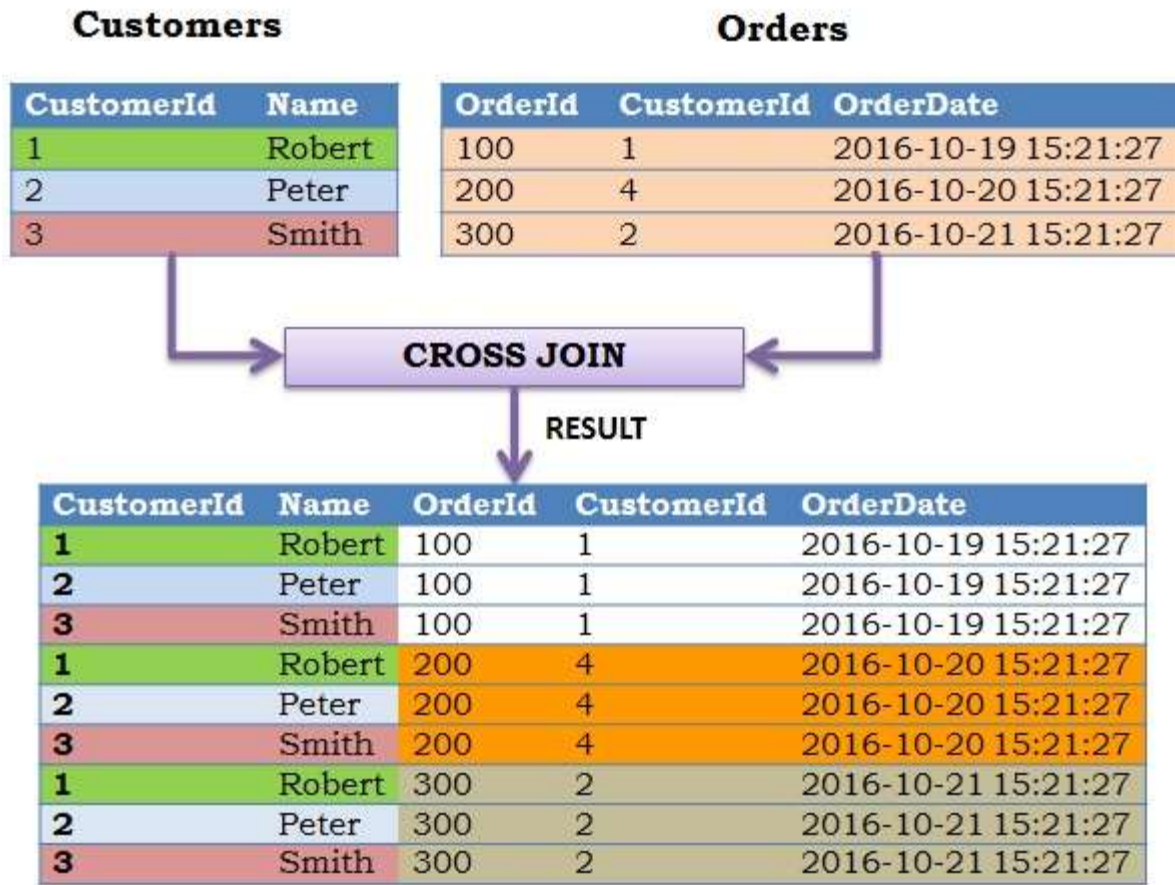
JOIN and ON

- After the FROM statement, we have two new statements: JOIN, which is followed by a table name, and ON, which is followed by a couple column names separated by an equals sign.

```
SELECT employee.LastName, employee.DepartmentID,  
department.DepartmentName  
  
FROM employee  
  
JOIN department  
  
ON employee.DepartmentID = department.DepartmentID
```

CROSS JOIN

CROSS JOIN



- CROSS JOIN returns the Cartesian product of rows from tables in the join.

INNER JOIN

INNER JOIN

Customers

CustomerId	Name
1	Robert
2	Peter
3	Smith

Orders

OrderId	CustomerId	OrderDate
100	1	2016-10-19 15:21:27
200	4	2016-10-20 15:21:27
300	2	2016-10-21 15:21:27

**INNER JOIN on
CustomerId Column**

RESULT

CustomerId	Name	OrderId	CustomerId	OrderDate
1	Robert	100	1	2016-10-19 15:21:27
2	Peter	300	2	2016-10-21 15:21:27

•INNER JOIN returns the matching rows.

NATURAL /INNER/EQUI JOIN

Return all the matching rows of source and target tables.

When there is more than one identical column

```
SELECT first_name, department_name  
FROM employees  
NATURAL JOIN departments;
```

OR

```
SELECT first_name, department_name  
FROM employees  
JOIN DEPARTMENTS
```

```
ON (employees.manager_id =  
departments.manager_id AND employees.de  
partment_id = departments.department_id);
```

OR

```
SELECT first_name, department_name  
FROM employees  
JOIN departments  
USING(manager_id)
```

When there is one identical column.

```
SELECT department_name, city  
FROM departments  
NATURAL  
JOIN locations;
```

OUTER JOIN (LEFT JOIN)

LEFT OUTER JOIN

Customers

CustomerId	Name
1	Robert
2	Peter
3	Smith

Orders

OrderId	CustomerId	OrderDate
100	1	2016-10-19 15:21:27
200	4	2016-10-20 15:21:27
300	2	2016-10-21 15:21:27

LEFT OUTER JOIN on
CustomerId Column

RESULT

CustomerId	Name	OrderId	CustomerId	OrderDate
1	Robert	100	1	2016-10-19 15:21:27
2	Peter	300	2	2016-10-21 15:21:27
3	Smith	NULL	NULL	NULL

As the name suggests Left Outer Join is a form of Outer Join that returns each and every record from the source table and returns only those values from the target table that fulfil the Join condition.

OUTER JOIN (RIGHT JOIN)

RIGHT OUTER JOIN



As the name suggests RIGHT Outer Join is a form of Outer Join that returns each and every record from the source table and returns only those values from the target table that fulfil the Join condition.

OUTER JOIN (FULL JOIN)

FULL OUTER JOIN

Customers

CustomerId	Name
1	Shree
2	Kalpana
3	Basavaraj

Orders

OrderId	CustomerId	OrderDate
100	1	2014-01-29 23:56:57.700
200	4	2014-01-30 23:56:57.700
300	3	2014-01-31 23:56:57.700

FULL OUTER JOIN on
CustomerId Column

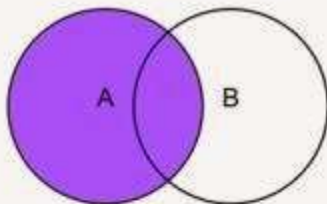
RESULT

CustomerId	Name	OrderId	CustomerId	OrderDate
1	Shree	100	1	2014-01-30 23:48:32.850
2	Kalpana	NULL	NULL	NULL
3	Basavaraj	300	3	2014-02-01 23:48:32.853
NULL	NULL	200	4	2014-01-31 23:48:32.853

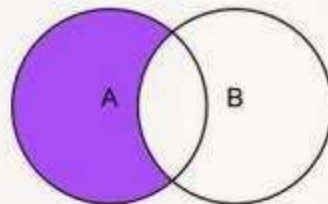
It returns all the rows from both the tables, if there is no matching row in either of the sides then it displays NULL values in the result for that table columns in such rows.

SQL JOINS

Left Join

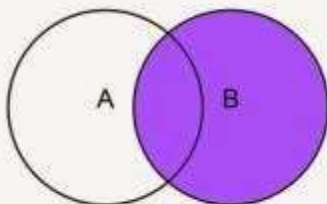


```
SELECT <Field_List>
FROM TableA A LEFT JOIN TableB B
ON A.PKey = B.PKey
```

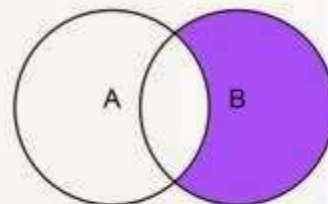


```
SELECT <Field_List>
FROM TableA A LEFT JOIN TableB B
ON A.PKey = B.PKey
WHERE B.PKey IS NULL
```

Right Join

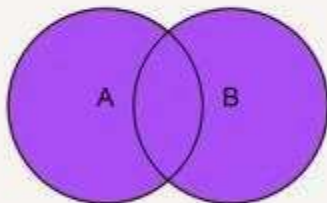


```
SELECT <Field_List>
FROM TableA A RIGHT JOIN TableB B
ON A.PKey = B.PKey
```

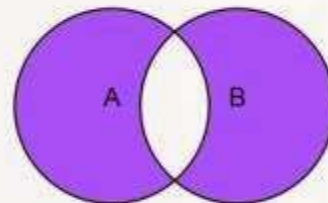


```
SELECT <Field_List>
FROM TableA A RIGHT JOIN TableB B
ON A.PKey = B.PKey
WHERE A.PKey IS NULL
```

Full Join

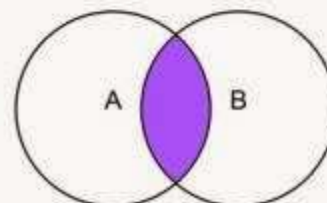


```
SELECT <Field_List>
FROM TableA A FULL OUTER JOIN TableB B
ON A.PKey = B.PKey
```



```
SELECT <Field_List>
FROM TableA A FULL OUTER JOIN TableB B
ON A.PKey = B.PKey
WHERE A.PKey IS NULL
AND B.PKey IS NULL
```

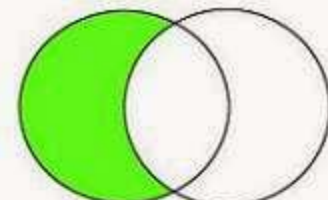
Inner Join



```
SELECT <Field_List>
FROM TableA A INNER JOIN TableB B
ON A.PKey = B.PKey
```

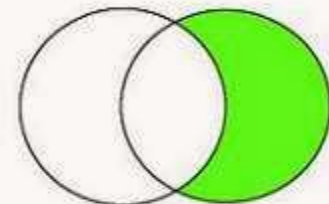
SET OPERATORS

Minus Except



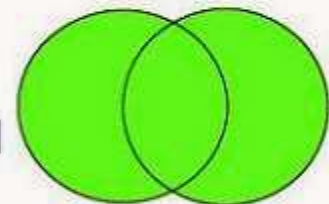
Set 1 Set 2

Minus Except



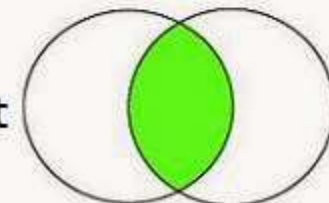
Set 1 Set 2

Union Union All



Set 1 Set 2

Intersect



Set 1 Set 2

JOIN more than TWO tables

Get the first and last names of all analysts whose department is located in Seattle:

```
USE sample;
SELECT emp_fname, emp_lname
      FROM works_on JOIN employee ON works_on.emp_no=employee.emp_no
                        JOIN department ON employee.dept_no=department.dept_no
      AND location = 'Seattle'
      AND job = 'analyst';
```

Reading Assignment

7.2.1 Specifying general constraints as Assertions in SQL

Summary

- JOIN and Types of JOIN