

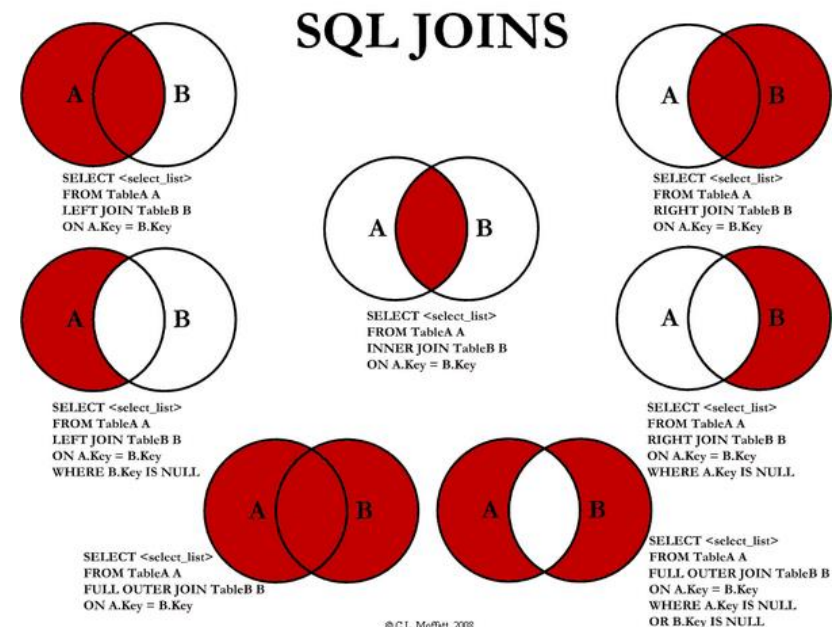
JOINS IN SQL SERVER



By the end of this lecture ✓ Understand about SQL joins in SQL Server

students should be able to:

✓ Using smoothly SQL joins and apply to project



- | **What's SQL Join?**
- | **Inner Join**
- | **Outer Join**
- | **Cross Join**
- | **Self Join**
- | **Excluding JOIN**

Options ?

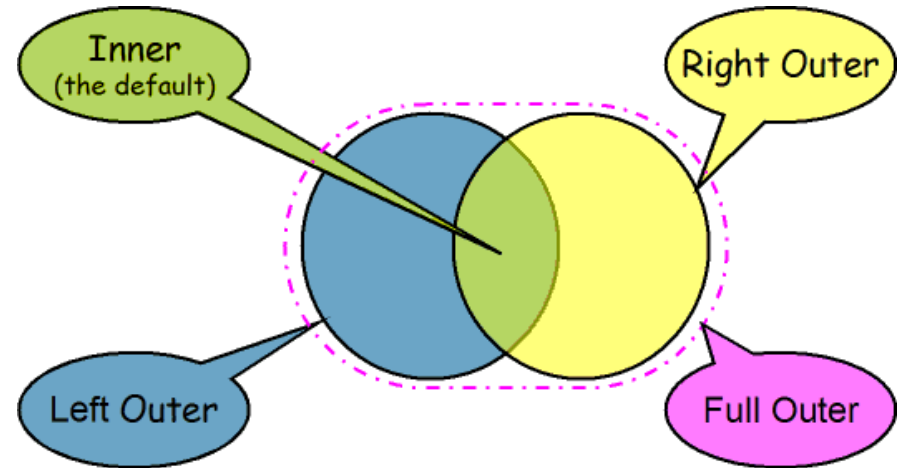


SQL Joins are used to combine rows from two or more tables based on logical relationships between the tables.

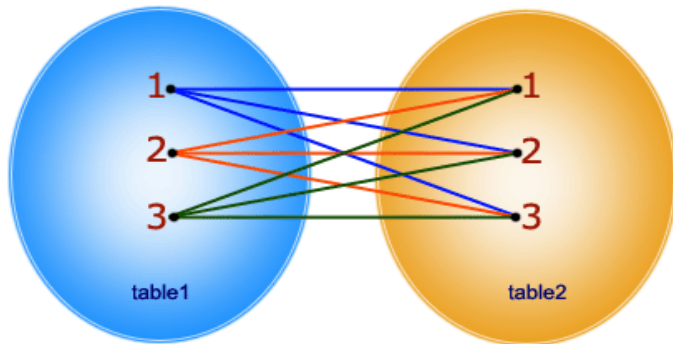


Types of Join in SQL:

- ✓ Inner Join
- ✓ Outer Join
- ✓ Cross Join
- ✓ Self Join



```
SELECT * FROM table1 CROSS JOIN table2;
```



Column Name	Data Type	Nullable	Default	Primary Key
EMP_ID	VARCHAR (5)	No	-	1
EMP_NAME	VARCHAR (20)	Yes	-	-
DT_OF_JOIN	DATE	Yes	-	-
EMP_SUPV	VARCHAR (5)	Yes	-	-
1 - 4				

Constraint	Type	Table
SYS_C004074	C	EMPLOYEE
EMP_ID	P	EMPLOYEE
EMP_SUPV	R	EMPLOYEE

Primary key

Foreign key

Referencing EMP_ID of this table

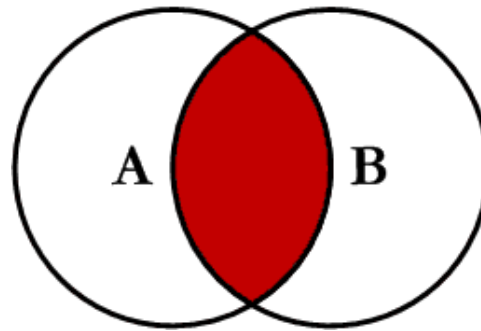
Section 1

INNER JOIN

- ❖ The INNER JOIN selects all rows from both tables as long as there is a match between the columns in both tables.
- ❖ Eliminate the rows that do not match with a row from the other table

✓ **Syntax**

```
SELECT col_names  
FROM Table_A A  
INNER JOIN Table_B B  
ON A.Col1 = B.Col1
```



INNER JOIN (2/2)

❖ Example:

Customer

CustID	CustName	BirthDate	Country
1	Davolio Nancy	12/8/1968	Germany
2	Fuller Andrew	2/19/1952	Mexico
3	Leverling Janet	8/30/1963	Mexico

[Order]

OrderID	CustID	OrderDate	ShipperID
10308	2	2013-09-18	3
10309	3	2013-09-19	1
10310	77	2013-09-20	2

```
SELECT c.CustName, o.OrderID
FROM Customer c
      INNER JOIN [Order] o
      ON c.CustID = o.CustID
ORDER BY c.CustName;
```

❖ Result:

	CustName	OrderID
1	Fuller Andrew	10308
2	Leverling Janet	10309

Section 2

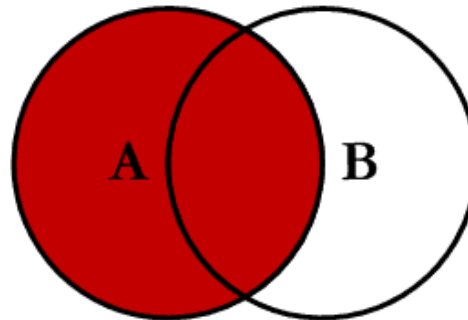
OUTER JOIN

- ❖ **Outer Join:** Return all rows from at least one of the tables mentioned in the FROM clause, as long as those rows meet any WHERE or HAVING search conditions:
 - ✓ **LEFT OUTER JOIN (or LEFT JOIN)**
 - ✓ **RIGHT OUTER JOIN (or RIGHT JOIN)**
 - ✓ **FULL OUTER JOIN (or FULL JOIN)**

- ❖ Return all of the records in the left table (table A) regardless if any of those records has a match in the right table (table B)
 - ✓ In the results where there is no matching condition, the row contains NULL values for the right table's columns.

❖ Syntax

```
SELECT col_names  
FROM Table_A A LEFT JOIN Table_B B  
ON A.Col1 = B.Col1
```



LEFT OUTER JOIN

❖ Example:

Customer

CustID	CustName	BirthDate	Country
1	Davolio Nancy	12/8/1968	Germany
2	Fuller Andrew	2/19/1952	Mexico
3	Leverling Janet	8/30/1963	Mexico

[Order]

OrderID	CustID	OrderDate	ShipperID
10308	2	2013-09-18	3
10309	3	2013-09-19	1
10310	77	2013-09-20	2

```
SELECT c.CustName, o.OrderID  
FROM Customer c LEFT JOIN [Order] o  
ON c.CustID = o.CustID  
ORDER BY c.CustName;
```

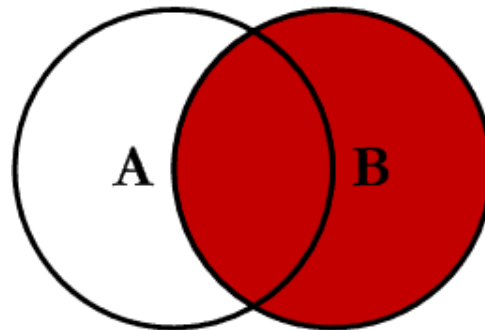
❖ Result:

	CustName	OrderID
1	Davolio Nancy	NULL
2	Fuller Andrew	10308
3	Leverling Janet	10309

- ❖ Return all of the records in the right table (table B) regardless if any of those records have a match in the left table (table A)
 - ✓ In the results where there is no matching condition, the row contains NULL values for the left table's columns.

❖ Syntax

```
SELECT col_names  
FROM Table_A A RIGHT JOIN Table_B B  
ON A.Col1 = B.Col1
```



RIGHT OUTER JOIN

❖ Example:

Customer

CustID	CustName	BirthDate	Country
1	Davolio Nancy	12/8/1968	Germany
2	Fuller Andrew	2/19/1952	Mexico
3	Leverling Janet	8/30/1963	Mexico

[Order]

OrderID	CustID	OrderDate	ShipperID
10308	2	2013-09-18	3
10309	3	2013-09-19	1
10310	77	2013-09-20	2

```
SELECT c.CustName, o.OrderID  
FROM Customer c RIGHT JOIN [Order] o  
ON c.CustID = o.CustID  
ORDER BY c.CustName;
```

❖ Result:

	CustName	OrderID
1	NULL	10310
2	Fuller Andrew	10308
3	Leverling Janet	10309

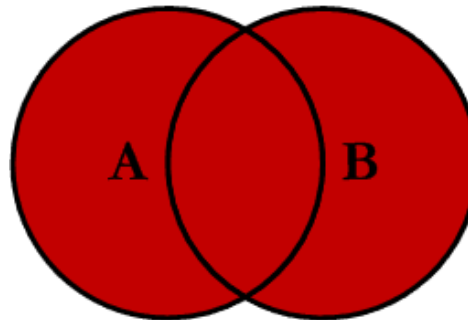
- ❖ Return all of the records from both tables, joining records from the left table (table A) that match records from the right table (table B)

- ❖ **Syntax**

SELECT col_names

FROM Table_A A **FULL JOIN** Table_B B

ON A.Col1 = B.Col1



FULL OUTER JOIN (2/2)

❖ Example:

Customer

CustID	CustName	BirthDate	Country
1	Davolio Nancy	12/8/1968	Germany
2	Fuller Andrew	2/19/1952	Mexico
3	Leverling Janet	8/30/1963	Mexico

[Order]

OrderID	CustID	OrderDate	ShipperID
10308	2	2013-09-18	3
10309	3	2013-09-19	1
10310	77	2013-09-20	2

```
SELECT c.CustName, o.OrderID
FROM Customer c FULL JOIN [Order] o
ON c.CustID = o.CustID
ORDER BY c.CustName;
```

❖ Result:

	CustName	OrderID
1	NULL	10310
2	Davolio Nancy	NULL
3	Fuller Andrew	10308
4	Leverling Janet	10309

Section 3

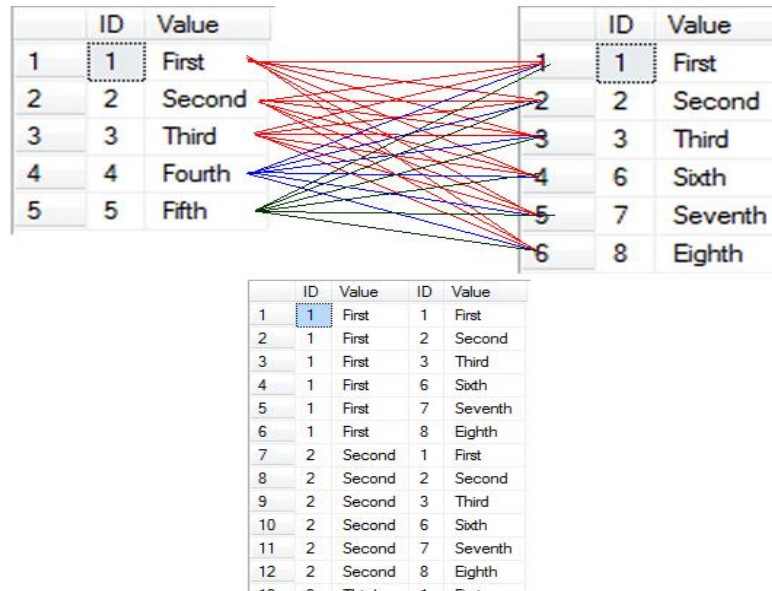
CROSS JOIN

- ❖ Return records that are multiplication of record number from both the tables
 - ✓ No need any condition to join

- ❖ **Syntax**

SELECT col_names

FROM Table_A A **CROSS JOIN** Table_B B



❖ Example:

Customer

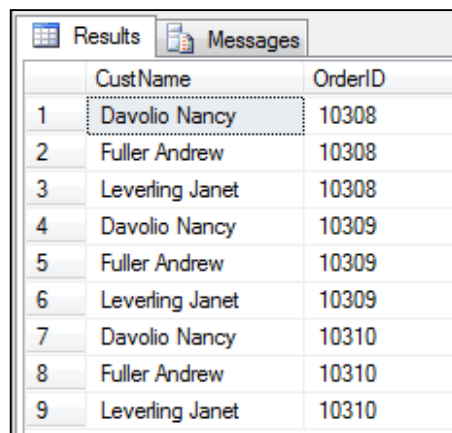
CustID	CustName	BirthDate	Country
1	Davolio Nancy	12/8/1968	Germany
2	Fuller Andrew	2/19/1952	Mexico
3	Leverling Janet	8/30/1963	Mexico

[Order]

OrderID	CustID	OrderDate	ShipperID
10308	2	2013-09-18	3
10309	3	2013-09-19	1
10310	77	2013-09-20	2

```
SELECT c.CustName, o.OrderID  
FROM Customer c CROSS JOIN [Order] o
```

❖ Result:



	CustName	OrderID
1	Davolio Nancy	10308
2	Fuller Andrew	10308
3	Leverling Janet	10308
4	Davolio Nancy	10309
5	Fuller Andrew	10309
6	Leverling Janet	10309
7	Davolio Nancy	10310
8	Fuller Andrew	10310
9	Leverling Janet	10310


Section 4

SELF JOIN

❖ A SELF JOIN is a join of a table to itself. In SELF JOIN, we can use:

- ✓ INNER JOIN
- ✓ OUTER JOIN
- ✓ CROSS JOIN

Column Name	Data Type	Nullable	Default	Primary Key
EMP_ID	VARCHAR (5)	No	-	1
EMP_NAME	VARCHAR (20)	Yes	-	-
DT_OF_JOIN	DATE	Yes	-	-
EMP_SUPV	VARCHAR (5)	Yes	-	-
				1 - 4



Constraint	Type	Table
SYS_C004074	C	EMPLOYEE
EMP_ID	P	EMPLOYEE
EMP_SUPV	R	EMPLOYEE

Primary key

Foreign key
Referencing EMP_ID of this table

❖ Example:

	EMP_ID	EMP_NAME	DT_OF_JOIN	EMP_SUPV
1	10120	Hansen Ola	2013-01-01	NULL
2	10121	Svendson Tove	2013-02-01	10120
3	10122	Pettersen Kari	2013-03-01	10120
4	10123	Alfreds Futterkiste	2013-04-01	10121

```
SELECT  e1.EMP_NAME AS Employee_Name,  
        e2.EMP_NAME AS Manager_Name  
FROM    Employee e1 LEFT JOIN Employee e2  
ON      e1.EMP_SUPV = e2.EMP_ID
```

❖ Result:

	Employee_Name	Manager_Name
1	Hansen Ola	NULL
2	Svendson Tove	Hansen Ola
3	Pettersen Kari	Hansen Ola
4	Alfreds Futterkiste	Svendson Tove

Section 5

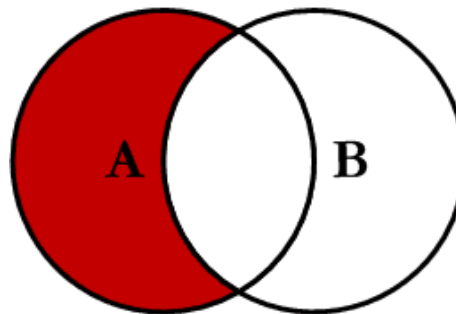
EXCLUDING JOIN

LEFT Excluding JOIN

- ❖ Return all of the records in the left table (table A) that do not match any records in the right table (table B)

- ❖ **Syntax**

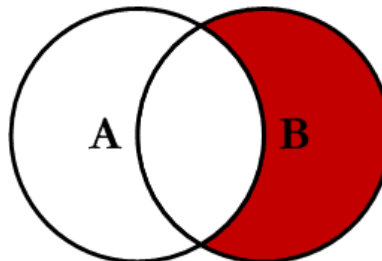
```
SELECT col_names  
FROM Table_A A LEFT JOIN Table_B B  
ON A.Col1 = B.Col1  
WHERE B.Col1 IS NULL
```



- ❖ Returns records in the right table (table B) that do not match any records in the left table (table A)
 - ✓ In the results where there is no matching condition, the row contains NULL values for the right table's columns.

❖ Syntax

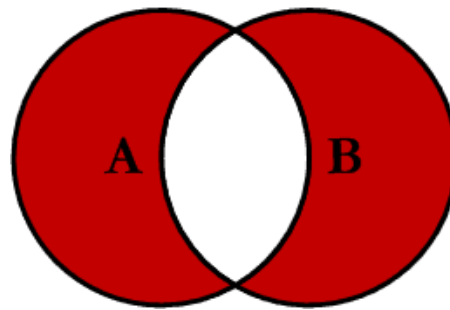
```
SELECT col_names  
FROM Table_A A RIGHT JOIN Table_B B  
ON A.Col1 = B.Col1  
WHERE A.Col1 IS NULL
```



- ❖ Return all of the records in the left table (table A) and all of the records in the right table (table B) that do not match

- ❖ **Syntax**

```
SELECT col_names  
FROM Table_A A  
FULL OUTER JOIN Table_B B  
ON A.Col1 = B.Col1  
WHERE A.Col1 IS NULL OR B.Col1 IS NULL
```



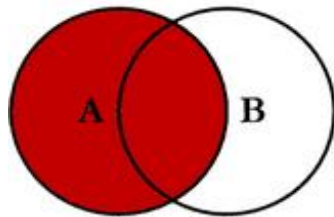
Joining Three or More Tables

- ❖ Since FROM clauses can contain multiple join specifications, this allows many tables to be joined for a single query.

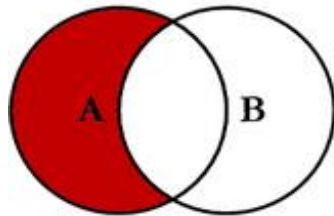
- ❖ **Syntax**

```
SELECT col_names  
FROM Table_A A JOIN Table_B B  
ON A.Col1 = B.Col1 LEFT JOIN Table_C C  
ON B.Col2 = C.Col2  
...
```

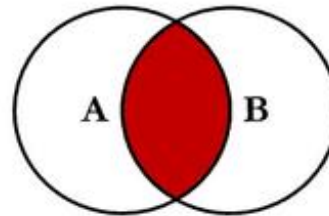
SQL JOINS



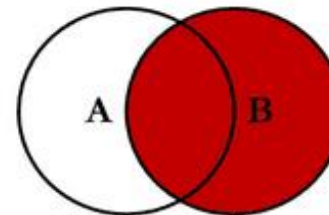
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



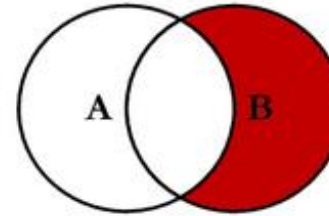
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



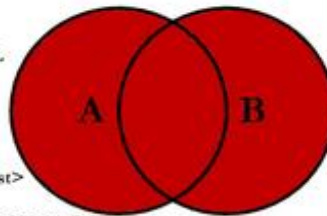
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



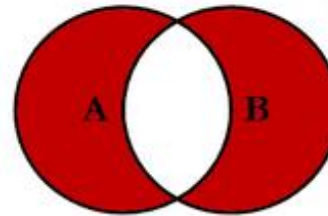
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



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



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

Q&A

Thank you

