SQL JOINS

JOINS

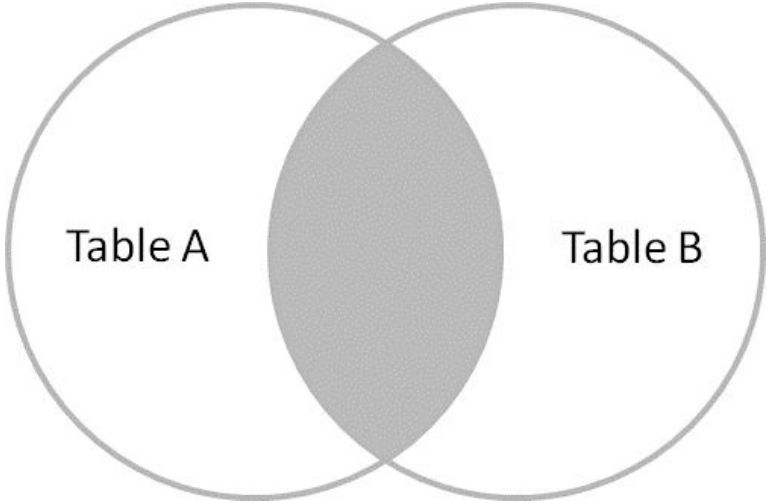
By using joins, you can retrieve data from two or more tables based on logical relationships between the tables. Joins indicate how SQL Server should use data from one table to select the rows in another table.

Types of Joins -

- INNER JOIN
- FULL JOIN
- LEFT JOIN
- RIGHT JOIN
- CROSS JOIN

INNER JOIN

Inner Join returns records that have matching values in both tables. It is also known as a simple join.



INNER JOIN

Syntax - ***SELECT column_name FROM table1 INNER JOIN table2 ON table1.column_name = table2.column_name;***

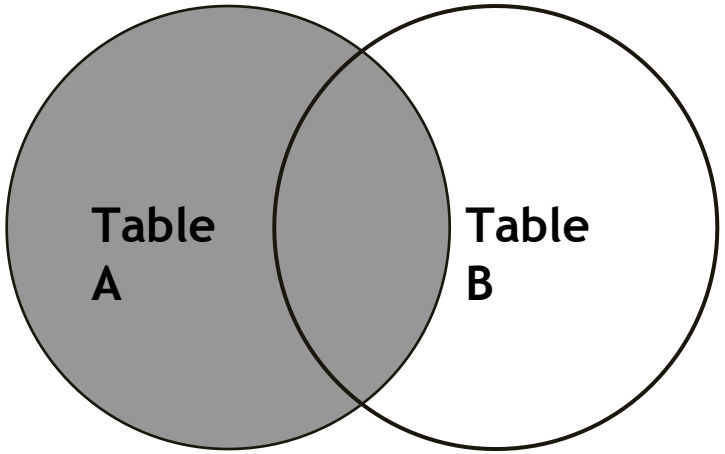
Create inner join of store1 and store2 where Average weekly sales is more than 50000

```
SELECT * FROM store1 INNER JOIN store2 ON store1.Store=Store2.Store WHERE Avg_WS>50000
```

Store	Avg_Temp	Avg_Fuel_P	Avg_CPI	Avg_UnEmp	Store	Dept	Avg_WS
13	50.89	3.30	129.20	6.76	13	38	83485.90
13	50.89	3.30	129.20	6.76	13	72	77119.19
13	50.89	3.30	129.20	6.76	13	91	81272.99
28	67.91	3.63	129.20	12.64	28	2	57751.27
28	67.91	3.63	129.20	12.64	28	90	65285.95
4	60.29	3.24	129.20	5.65	4	93	67815.16

LEFT JOIN

Left Join returns all the records from the left table and the matched records from the right table.



LEFT JOIN

Syntax - *SELECT column_name FROM table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;*

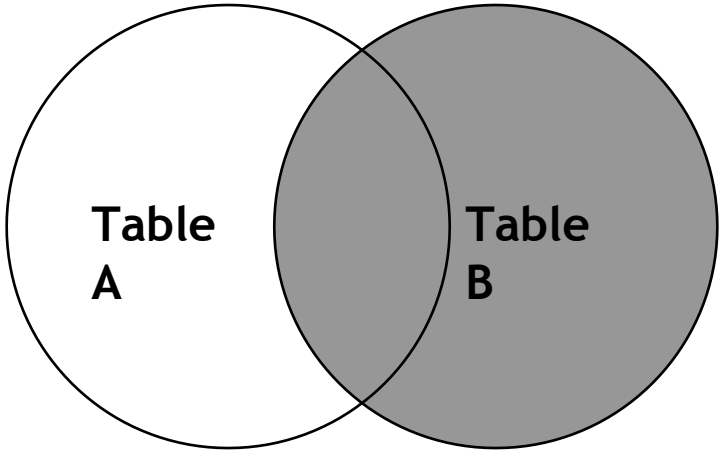
Find the Left Join of store1 and store2 where Average weekly sales is less than 50000

```
SELECT * FROM store1 LEFT JOIN store2 ON store1.Store=Store2.Store WHERE Avg_WS<50000
```

Store	Avg_Temp	Avg_Fuel_P	Avg_CPI	Avg_UnEmp	Store	Dept	Avg_WS
10	70.23	3.60	129.20	8.14	10	14	40044.64
12	67.91	3.63	129.20	12.64	12	79	43270.34
13	50.89	3.30	129.20	6.76	13	4	42563.28
13	50.89	3.30	129.20	6.76	13	13	47698.82
13	50.89	3.30	129.20	6.76	13	23	30216.26
13	50.89	3.30	129.20	6.76	13	81	33276.24
19	49.74	3.63	135.65	7.99	19	9	30645.02
22	52.44	3.48	139.59	7.96	22	8	37236.35
26	41.11	3.48	135.65	7.75	26	40	37448.67

RIGHT JOIN

Right Join returns all the records from the right table and the matched records from the left table.



RIGHT JOIN

Right JOIN returns all records from the right table, and the matching records from the left table.
Syntax - ***SELECT column_name FROM table1 RIGHT JOIN table2 ON table1.column_name = table2.column_name;***

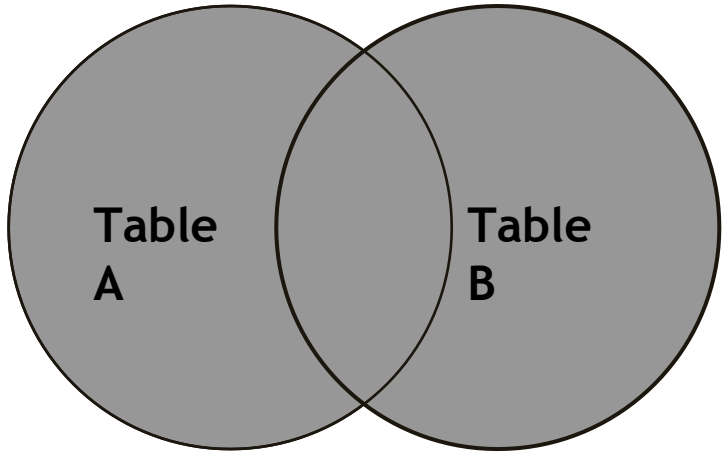
Find the Right Join of store1 and store2 where Average weekly sales is less than 50000

```
SELECT * FROM store1 RIGHT JOIN store2 ON store1.Store=Store2.Store WHERE Avg_WS<50000
```

Store	Avg_Temp	Avg_Fuel_P	Avg_CPI	Avg_UnEmp	Store	Dept	Avg_WS
NULL	NULL	NULL	NULL	NULL	1	8	35718.26
10	70.23	3.60	129.20	8.14	10	14	40044.64
NULL	NULL	NULL	NULL	NULL	11	90	48995.98
12	67.91	3.63	129.20	12.64	12	79	43270.34
13	50.89	3.30	129.20	6.76	13	4	42563.28
13	50.89	3.30	129.20	6.76	13	13	47698.82
13	50.89	3.30	129.20	6.76	13	23	30216.26
13	50.89	3.30	129.20	6.76	13	81	33276.24
NULL	NULL	NULL	NULL	NULL	14	1	30611.78

FULL JOIN

It returns all rows from the LEFT table and the RIGHT table with NULL values in place where the join condition is not met.



FULL JOIN

FULL JOIN also known as FULL OUTER JOIN which returns all records when there is a match in left or right table records.

Syntax - ***SELECT column_name FROM table1 FULL OUTER JOIN table2 ON table1.column_name = table2. column_name where condition;***

Find the store, Average CPI, Average fuel price and Average weekly sales using full Join on store1 and store2 where average Weekly sales is between 10000 to 40000

```
SELECT Store2.Store,
Store1.Avg_CPI,  store1.Avg_Fuel_P,
Store2.Avg_WS  FROM Store1 FULL
OUTER JOIN
store2 ON store1.Store=Store2.Store
WHERE Avg_WS BETWEEN 10000 AND
40000
```

Store	Avg_CPI	Avg_Fuel_P	Avg_WS
1	NULL	NULL	35718.26
13	129.20	3.30	30216.26
13	129.20	3.30	33276.24
14	NULL	NULL	30611.78
14	NULL	NULL	32701.44
14	NULL	NULL	35638.00
19	135.65	3.63	30645.02
22	139.59	3.48	37236.35
26	135.65	3.48	37448.67

JOINS USING GROUPBY AND HAVING CLAUSES

Fetch the Store column from Features table, Dept column and Average of weekly sales for each store from Sales table where average weekly sales is more 60000 using joins.

```
Select F.Store, S.Dept, AVG(S.Weekly_Sales) as
Avg_WS  from Features F
join Sales S
on
F.Store=S.Stor
e
GROUP BY F.Store, S.Dept
HAVING AVG(S.Weekly_Sales)>10000
```

	Store	Dept	Avg_WS
1	24	5	29178.058811
2	28	82	19119.629650
3	28	92	98486.960349
4	32	80	16749.993776
5	32	90	61639.637132
6	45	10	14245.086993
7	20	92	164633.741538
8	1	38	79978.222587
9	1	95	120772.062167
10	26	81	19192.056433

JOINS USING GROUP BY AND ORDER BY CLAUSES

Fetch the Store column from Features table and Average of weekly sales for each store from Sales table where average weekly sales must be displayed in descending order

```
Select F.Store, AVG(S.Weekly_Sales) as
Avg_WS  from Features F
join Sales S
on
F.Store=S.Store
GROUP BY
F.Store
ORDER BY Avg_WS DESC
```

	Store	Avg_WS
1	20	29508.301591
2	4	29161.210414
3	14	28784.851727
4	13	27355.136891
5	2	26898.070031
6	10	26332.303818
7	27	24826.984535
8	6	21913.243623
9	1	21710.543620
10	39	21000.763562