Back to the Basics: T-SQL 101

Data Platform Discovery Day

DEBORAH MELKIN

APRIL 29, 2020

The Basics About Me

- Worked as a DBA for 20 years:
  - Mainly work with SQL Server but have worked with Oracle, MySQL, Informix and PostgreSQL
- NESQL board member
- User Group & SQL Sat speaker
- IDERA ACE Class of 2020
- Speaker Idol 2019 Winner
- Random facts:
  - I'm the alto section leader in my choir.
  - I go to bluegrass jams regularly.
  - I've been learning guitar and now mandolin.
  - I am a bit of a musical theater geek.

Goals 101

- Understand **what** the different parts are which make up queries.
- Begin to think about how these different parts work together to answer not only what are they doing but **why** that query.
- Learn 1 new thing.

Data Platform Discovery Day

Types of SQL Statements

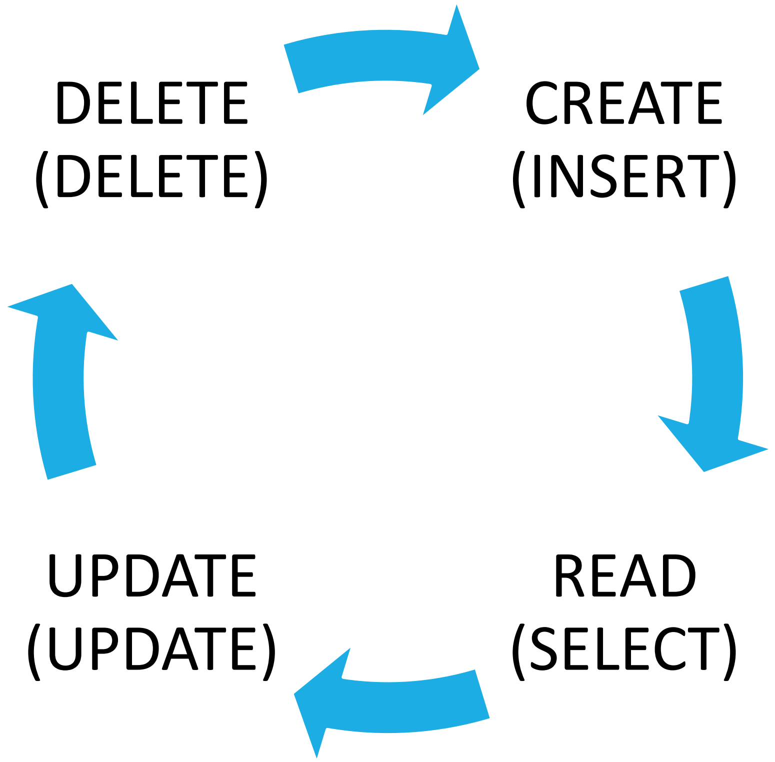
- Data Definition Language (DDL)
  - Statements that create, modify, or delete database objects

- Data Manipulation Language (DML)
  - Statements that modify data
  - May also be referred to as CRUD operations

This is our focus!

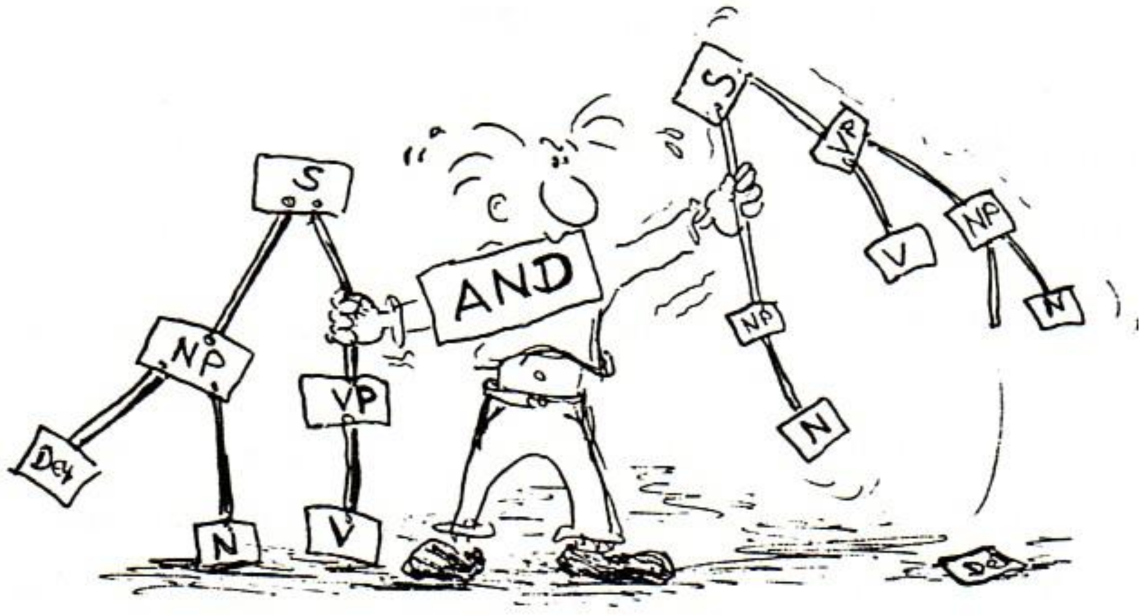
Breaking down the CRUD

- CREATE = INSERT
- READ = SELECT
- UPDATE = UPDATE
- DELETE = DELETE



The CRUD operations in action.

SQL Syntax

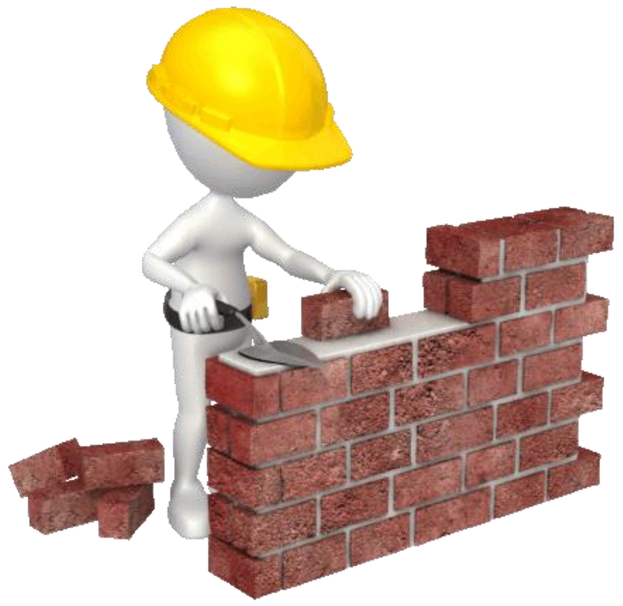


Data Platform Discovery Day

The Basic Statement: SELECT

- The SELECT statement is the basis for the other DML statements.

- You will write more SELECT statements than any other type of SQL statement.



SELECT Statement

```
SELECT <column list>

FROM <table(s)>

WHERE <where condition>
```

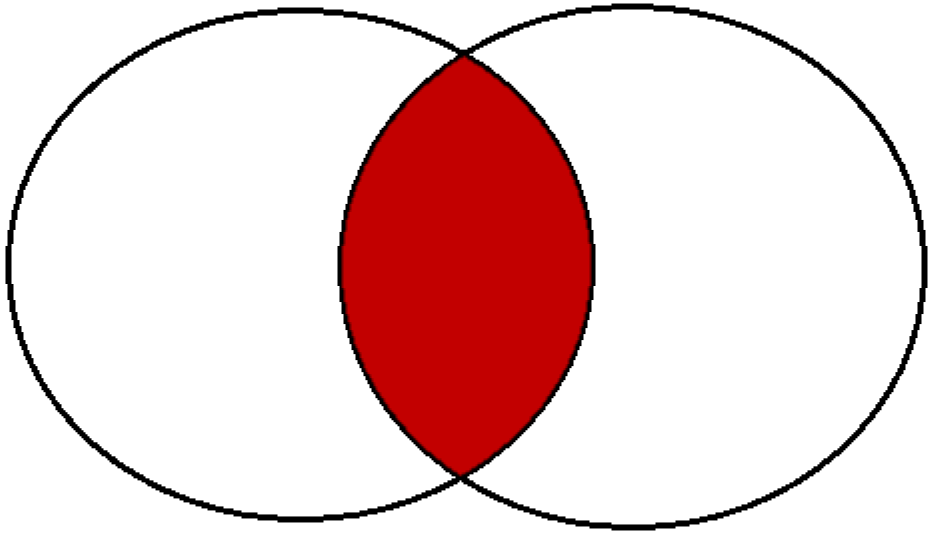
Limits the columns that are being returned

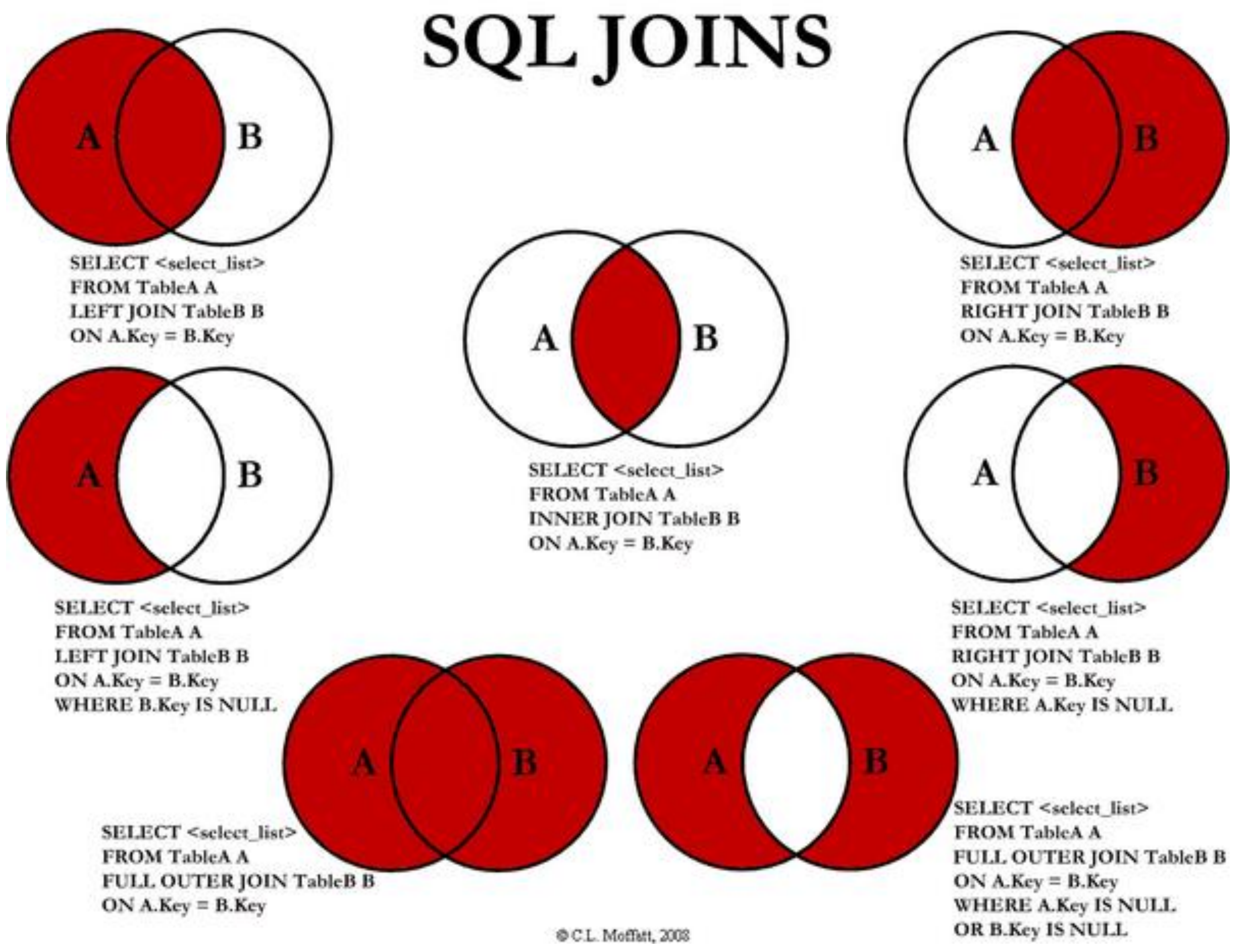
States the table\tables being working with

Limits the rows that are being returned

Tables and JOINs

- A JOIN defines the related column or columns between two tables.
- Types of Joins
  - INNER
  - OUTER
    - LEFT
    - RIGHT
    - FULL
  - CROSS JOIN





SQL JOINS

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

```
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
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

```
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
```

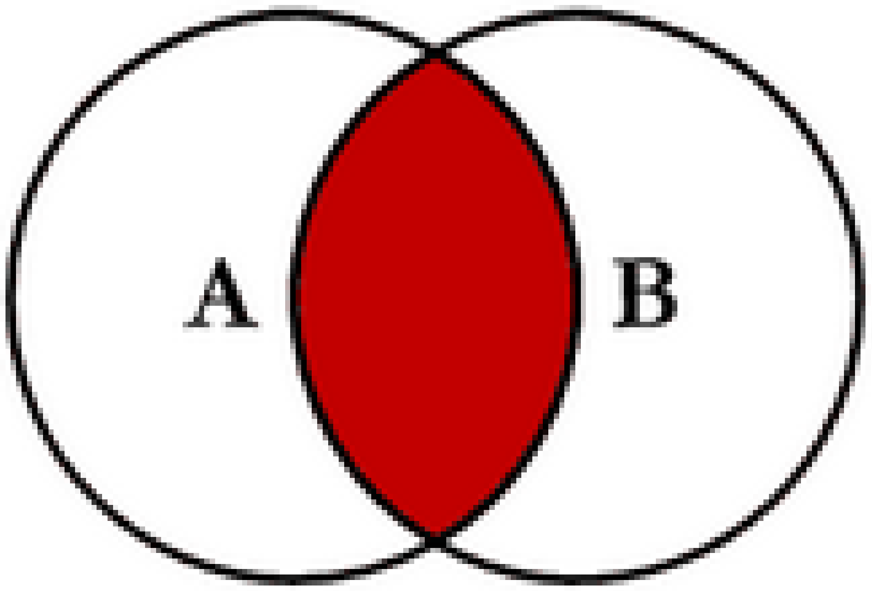
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Image from: https://www.codeproject.com/kb/database/visual_sql_joins.aspx

Data Platform Discovery Day

INNER JOIN

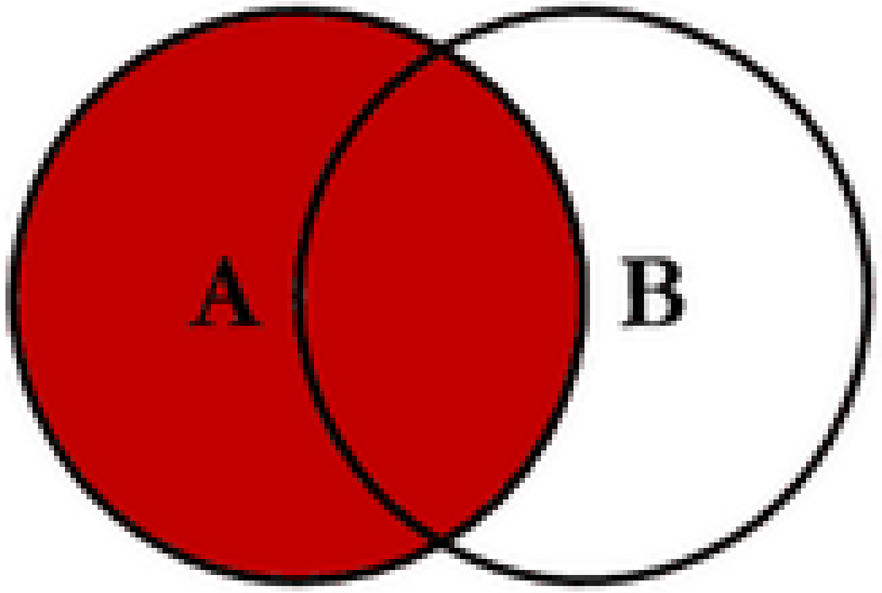
- Rows are returned when there are records in both tables based on the JOIN criteria.



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

LEFT OUTER JOIN

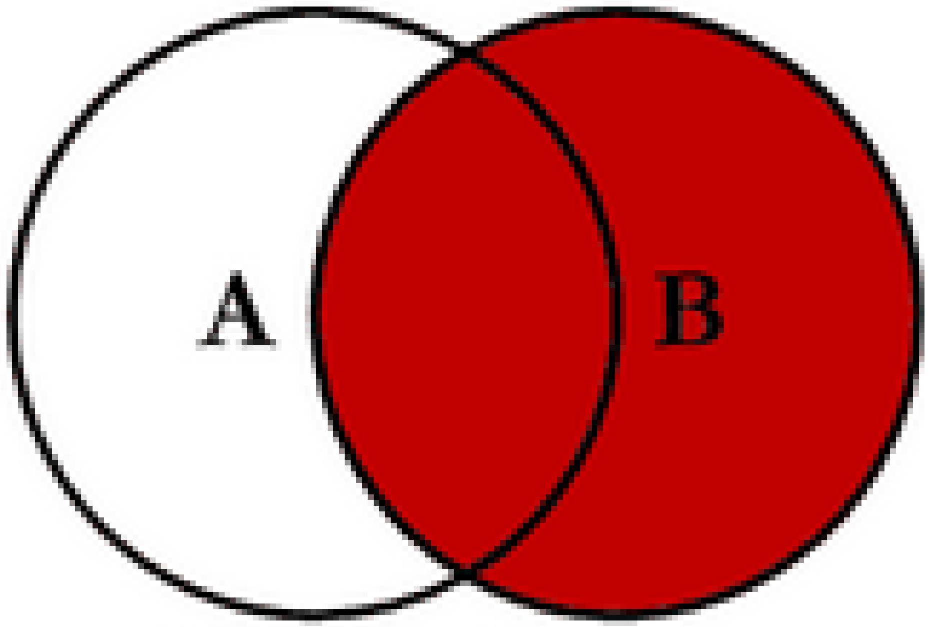
- Records are returned from the “Left” table and matching records from the “Right” table



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

RIGHT OUTER JOIN

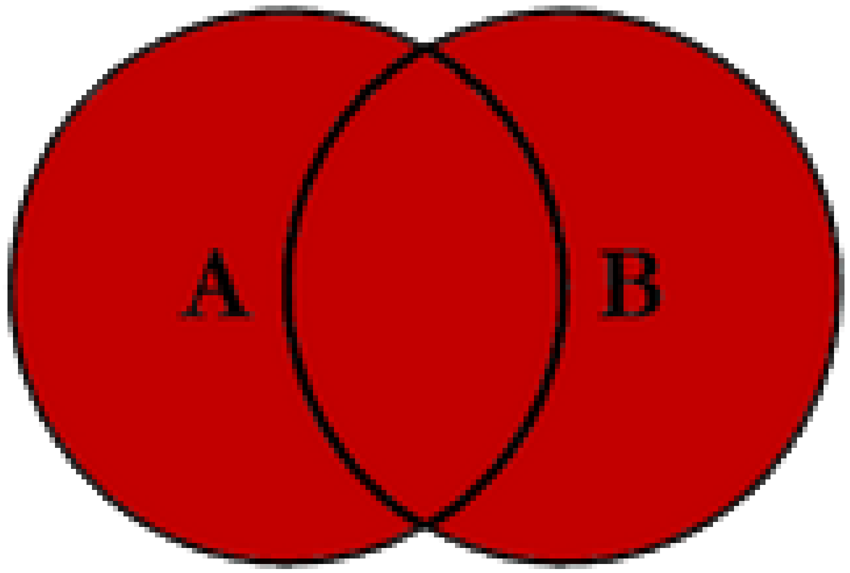
- Records are returned from the “Right” table and matching records from the “Left” table



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

FULL OUTER JOIN

- Returns records from the “Left” table and “Right” table, regardless whether there is a match

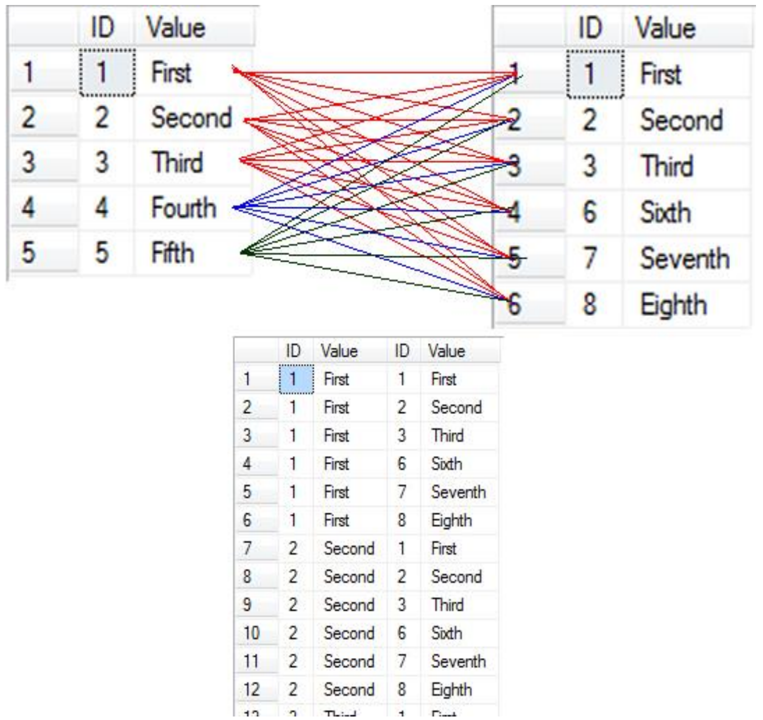


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

CROSS JOIN

- Returns a record set for each combination of records from both tables.

- Also known as a Cartesian JOIN

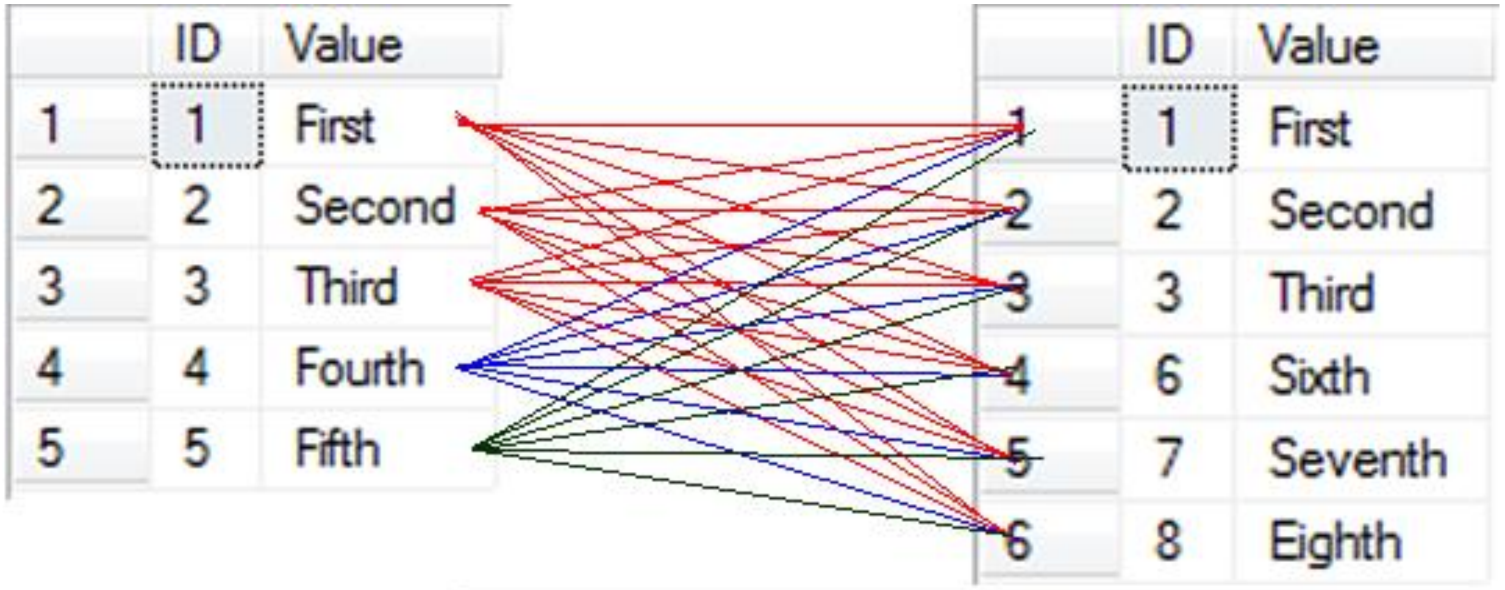


	ID	Value
1	1	First
2	2	Second
3	3	Third
4	4	Fourth
5	5	Fifth

	ID	Value
1	1	First
2	2	Second
3	3	Third
4	6	Sixth
5	7	Seventh
6	8	Eighth

	ID	Value	ID	Value
1	1	First	1	First
2	1	First	2	Second
3	1	First	3	Third
4	1	First	6	Sixth
5	1	First	7	Seventh
6	1	First	8	Eighth
7	2	Second	1	First
8	2	Second	2	Second
9	2	Second	3	Third
10	2	Second	6	Sixth
11	2	Second	7	Seventh
12	2	Second	8	Eighth

Data Platform Discovery Day

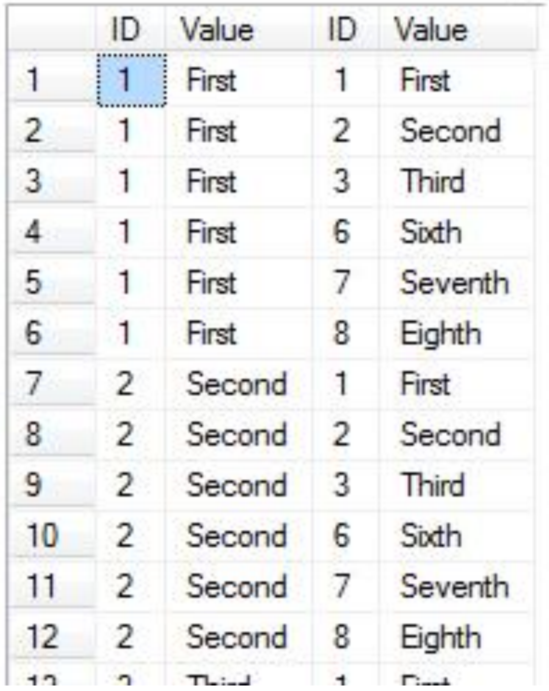


	ID	Value
1	1	First
2	2	Second
3	3	Third
4	4	Fourth
5	5	Fifth

	ID	Value
1	1	First
2	2	Second
3	3	Third
4	6	Sixth
5	7	Seventh
6	8	Eighth

CROSS JOIN:

a closer look



	ID	Value	ID	Value
1	1	First	1	First
2	1	First	2	Second
3	1	First	3	Third
4	1	First	6	Sixth
5	1	First	7	Seventh
6	1	First	8	Eighth
7	2	Second	1	First
8	2	Second	2	Second
9	2	Second	3	Third
10	2	Second	6	Sixth
11	2	Second	7	Seventh
12	2	Second	8	Eighth

Image from: https://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/

Other parts of the SELECT statement

```
SELECT <column list>
FROM  <table(s)>
WHERE <where condition>
GROUP BY <column list>
HAVING <where condition>
ORDER BY <column list>
```

The level to roll up the aggregations

WHERE conditions for aggregates

Sort order for the result set

SELECT Demos

INSERT Statement

ASSUMES ALL VALUES ARE KNOWN

```
INSERT INTO <table>
     (<column list>)
VALUES (<values>)
```

DIRECTS THE OUTPUT OF THE SELECT

```
INSERT INTO <table>
     (<column list>)
SELECT <column list>
FROM <table(s)>
WHERE <where condition>
```

UPDATE Statement

CAN BE USED BY ALL SQL DATABASES	T-SQL SPECIFIC
`UPDATE <table>`	`UPDATE <table or alias>`
`SET <column> = '<value>'`	`SET <column> = '<value>'`
`WHERE <where condition>`	`FROM <table(s)>` `WHERE <where condition>`

DELETE Statement

CAN BE USED BY ALL SQL DATABASES	T-SQL SPECIFIC
`DELETE`	`DELETE <table or alias>`

`FROM <table>`
`WHERE <where condition>`
	`FROM <table(s)>`
	`WHERE <where condition>`

INSERT/ UPDATE/ DELETE Demos

Practice, Practice, Practice

- Set up a local test database.
- Start simple and work up to complex.
- Look up examples and use those to learn.
- Read some books.
  - *SQL in a Nutshell* by Kevin Kline, Daniel Kline and Brand Hunt
  - *T-SQL Fundamentals* by Itzik Ben-Gan

- Go to your local user group and PASS Virtual User Group meetings.
- Keep going to Virtual Conferences, SQL Saturdays, and other events like these.

Have More Questions? Let me know!

Email: dgmelkin@gmail.com

Twitter: @dgmelkin

Blog: DebtheDBA.wordpress.com

Thanks for coming!