

# Introduction to T-SQL Queries




Kathi Kellenberger

Redgate Software



# Kathi Kellenberger

- **Editor and Advocate  
Redgate Software**

-  /Kathikellenberger
-  @aunkathi
- <http://aunkathisql.com>
-  Kathi.Kellenberger@red-gate.com

- Lifelong learner
- Teacher

- 
- Co-leader of Data Platform Women in Technology Group
  - Data Platform MVP

# Agenda

- Class 1
  - Module 1: Introduction
  - Module 2: Simple select statements
  - Module 3: Filtering
- Class 2
  - Module 4: Expressions
  - Module 5: Joining
- Class 3
  - Module 5: Joining (Continued)
  - Module 6: Grouping
- Class 4
  - Module 7: Subqueries
  - Module 8: UNION

# CLASS MATERIALS

- <https://github.com/KathiKellenberger/CoderGirlDataAnalysis>
  - Slides
  - Demos
  - Labs
- Students should install Azure Data Studio and connect to
  - [sqlprojects.com](https://sqlprojects.com),3500
  - Student
  - Madison18\*
  - Instructions will be given in class

# Module 4: Expressions

# What's in an expression?

- Column, really anything
- Column1 + Column2
- Concatenating strings  
    <string1> + <string2>
- Math  
    <number> <operator> <number>
- Lots of built-in functions!

# Functions

- CAST and CONVERT – change a data type
- ISNULL and COALESCE – replace NULL

# Demo: Expressions



# Lab

- Complete Module 4 Lab 1

# String functions

- RTRIM, LTRIM, TRIM remove spaces
- LEFT, RIGHT return a number of characters
- LEN, DATALENGTH return the length
- CHARINDEX find a string
- SUBSTRING return part of a string
- REVERSE returns the string backwards
- UPPER, LOWER returns all upper or lower case
- REPLACE replace part of a string

# Demo: String functions

# Lab

- Complete Module 4 Lab 2

# Working with Dates

- GETDATE, SYSDATETIME returns the server date
- DATEADD adds a time period to a date
- DATEDIFF finds the difference between two dates
- DATENAME, DATEPART returns part of a date
- DAY, MONTH, YEAR returns part of a date
- CONVERT, FORMAT formatting dates

Demo: Working with dates

# Lab

- Complete Module 4 Lab 3

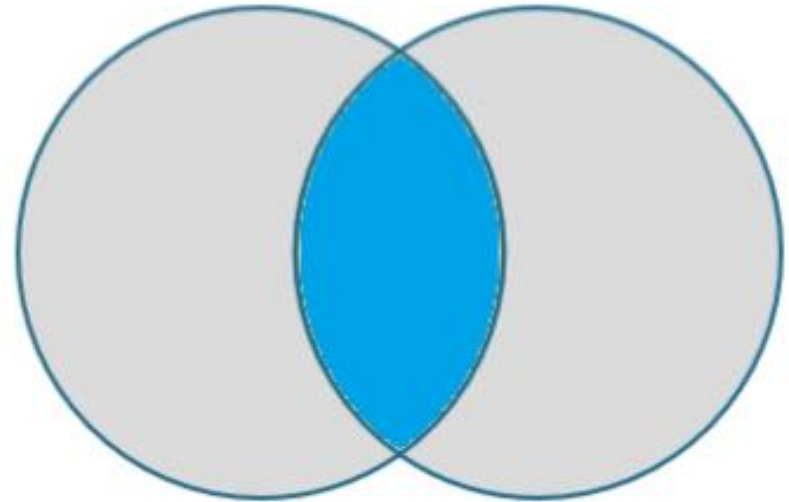
# Module 5: Joining Tables



# INNER JOIN

- The columns from two tables where there is a match on a key
- Syntax

```
SELECT <table1>.<col1>,<table2>.<col2>  
FROM <table1> [INNER] JOIN <table2>  
ON <table1>.<col1> = <table2>.<col1>
```



# Old join syntax: Comma join (Don't use!)

```
SELECT Col1, Col1  
FROM table1, table2  
Where table1.col1 = table2.col1
```

- Used more often by Oracle developers than SQL Server devs

# INNER JOIN

Customer	
CustomerID (Primary Key)	Name
1	John
2	Sharon
3	Dana
4	Fox

Sale		
SaleID (Primary Key)	CustomerID (Foreign Key)	Amt
1	3	100
2	1	200
3	3	75
4	3	90
5	1	100

Query results			
SaleID	CustomerID	Name	Amt
1	3	Dana	100
2	1	John	200
3	3	Dana	75
4	3	Dana	90
5	1	John	100

Demo: INNER JOIN

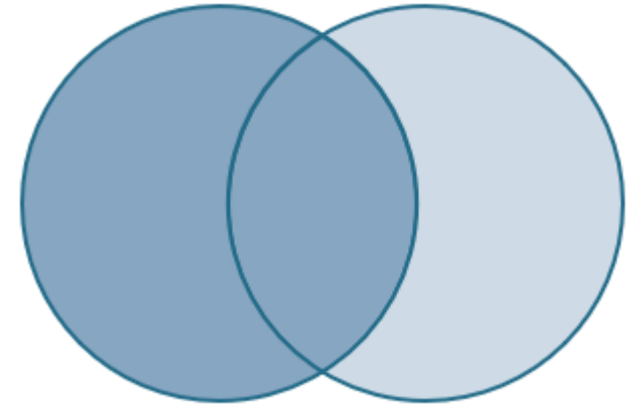
# Lab

- Complete Module 5 Lab 1

# LEFT OUTER JOIN

- All the rows from first table even if they don't match
- Once you start down the left path, continue left
- Syntax

```
SELECT <table1>.<col1>, <table2>.<col2>  
FROM <table1>  
LEFT [OUTER] JOIN <table2>  
ON <table1>.<col1> = <table2>.<col2>
```



# LEFT OUTER JOIN

Customer	
CustomerID	Name
1	John
2	Sharon
3	Dana
4	Fox

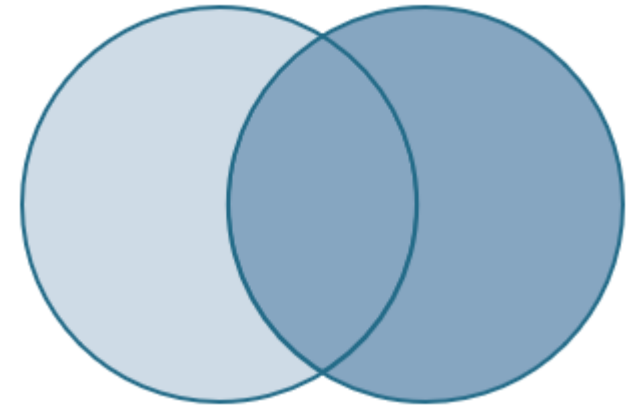
Sale		
SaleID	CustomerID	Amt
1	3	100
2	1	200
3	3	75
4	3	90
5	1	100

Query results			
SaleID	CustomerID	Name	Amt
1	3	Dana	100
2	1	John	200
3	3	Dana	75
4	3	Dana	90
5	1	John	100
NULL	2	Sharon	NULL
NULL	4	Fox	NULL

# RIGHT OUTER JOIN

- All the rows from second table even if they don't match
- Not used as much
- Syntax

```
SELECT <table1>.<col1>, <table2><col2>  
FROM <table1>  
RIGHT [OUTER] JOIN <table2>  
ON <table1>.<col1> = <table2>.<col2>
```

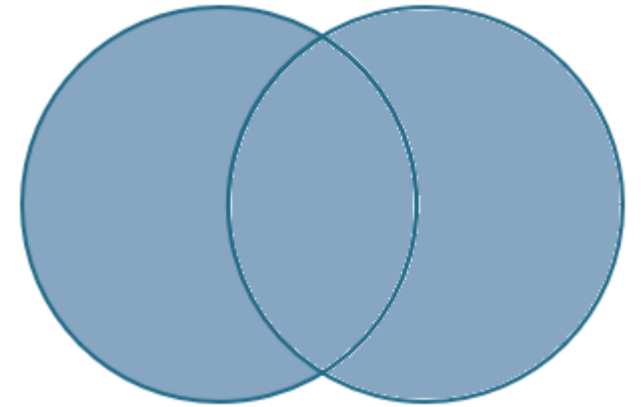




# FULL OUTER JOIN

- All the rows from both tables even if they don't match
- Rarely used
- Syntax

```
SELECT <table1>.<col1>, <table2>.<col2>  
FROM <table1>  
FULL [OUTER] JOIN <table2>  
ON <table1>.<col1> = <table2>.<col2>
```



Demo: OUTER JOIN

# Lab

- Complete Module 5 Lab 2

# LEFT OUTER JOIN with NULL RIGHT Filter

- Use to find rows that don't match
- Filter on a key from the table on the right
- Syntax

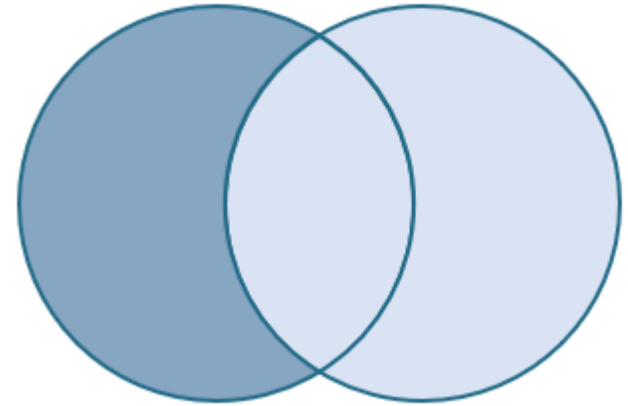
```
SELECT <table1>.<col1>,<table2>.<col2>
```

```
FROM <table1>
```

```
LEFT [OUTER] JOIN <table2>
```

```
ON <table1>.<col1 > = <table2>.<col1>
```

```
WHERE <table2>.<col1> IS NULL
```



# LEFT OUTER JOIN with NULL right table filter

Customer	
CustomerID	Name
1	John
2	Sharon
3	Dana
4	Fox

Sale		
SaleID	CustomerID	Amt
1	3	100
2	1	200
3	3	75
4	3	90
5	1	100

Query results			
SaleID	CustomerID	Name	Amt
NULL	2	Sharon	NULL
NULL	4	Fox	NULL

Demo: OUTER JOIN with FILTER

# Lab

- Complete Module 5 Lab 3