# Introduction to T-SQL Queries – Module 4

Kathi Kellenberger
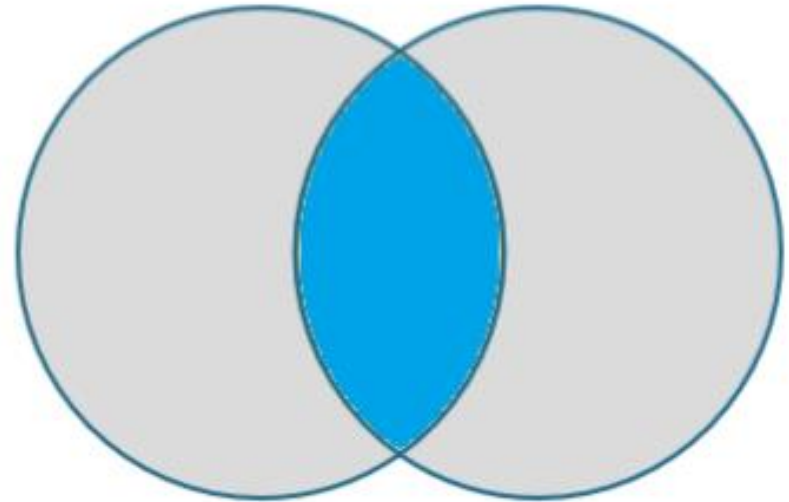
Redgate Software

# 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

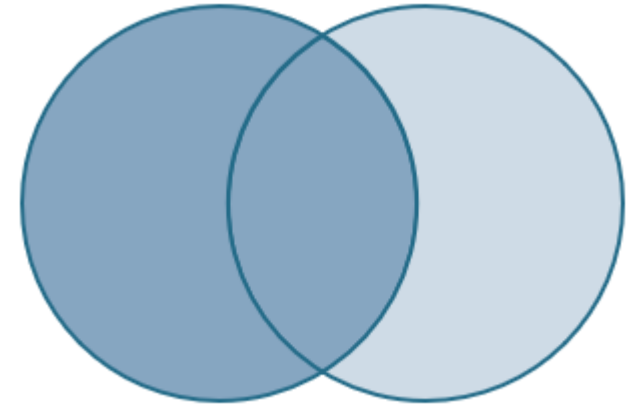| SaleID | CustomerID | Name | Amt |
|---|---|---|---|
| 1 | 3 | Dana | 100 |
| 2 | 1 | John | 200 |
| 3 | 3 | Dana | 75 |
| 4 | 3 | Dana | 90 |
| 5 | 1 | John | 100 |

# 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>
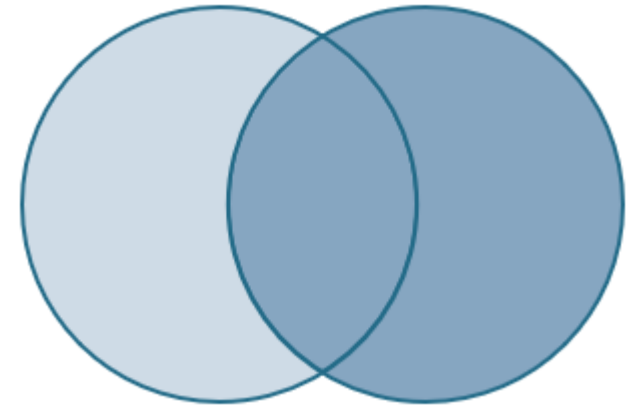
# LEFT OUTER JOIN

## 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>
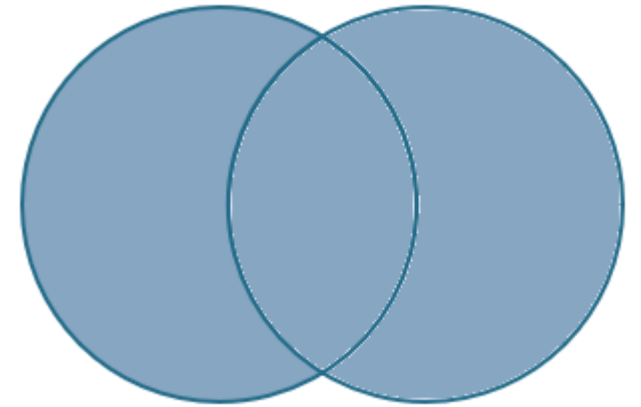
    FROM <table1>

    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
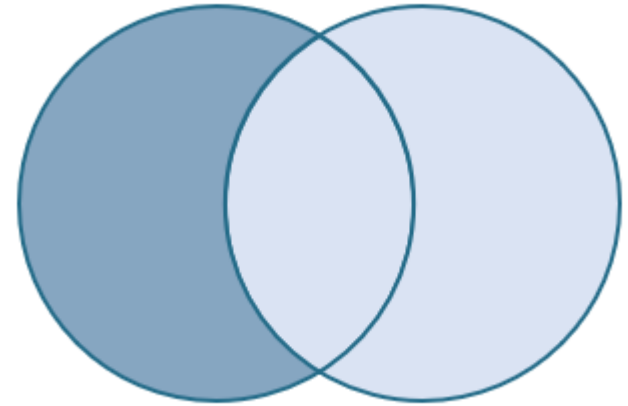- Filter on a key from the table on the right
- Syntax

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