03 | Querying Multiple Tables with Joins

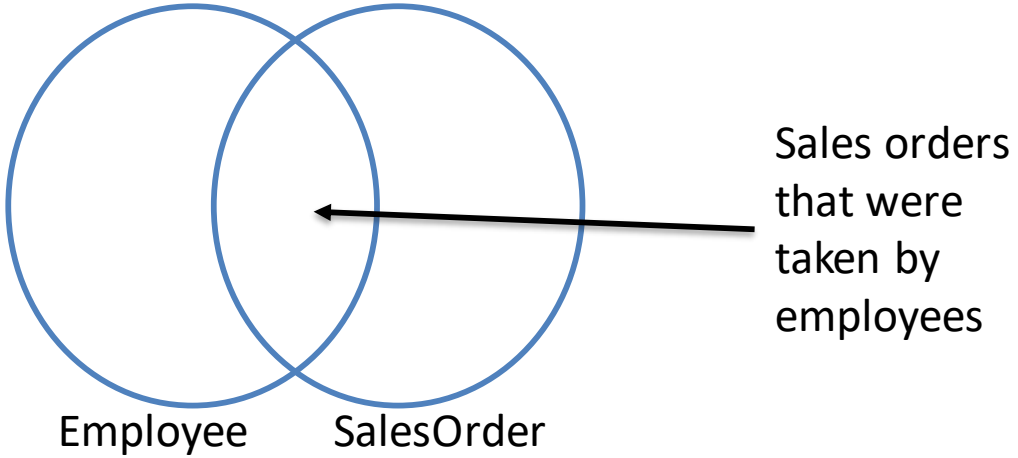
Graeme Malcolm | Senior Content Developer, Microsoft
Geoff Allix | Principal Technologist, Content Master

Module Overview

- Join Concepts
- Join Syntax
- Inner Joins
- Outer Joins
- Cross Joins
- Self Joins

Join Concepts

- Combine rows from multiple tables by specifying matching criteria
  - Usually based on primary key – foreign key relationships
  - For example, return rows that combine data from the **Employee** and **SalesOrder** tables by matching the **Employee.EmployeeID** primary key to the **SalesOrder.EmployeeID** foreign key
- It helps to think of the tables as sets in a Venn diagram



Join Syntax

- ANSI SQL-92
  - Tables joined by JOIN operator in FROM Clause
    - Preferred syntax

```
SELECT ...
FROM   Table1 JOIN Table2
       ON <on_predicate>;
```

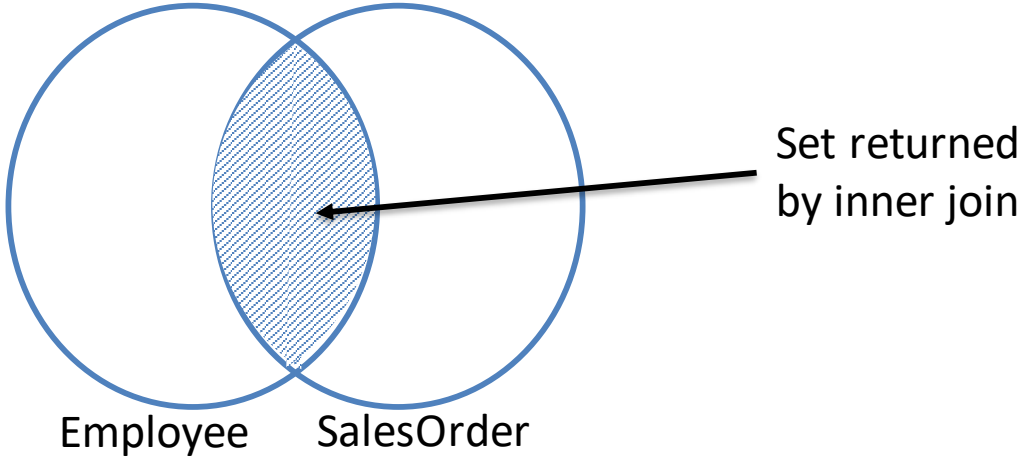
- ANSI SQL-89
  - Tables joined by commas in FROM Clause
    - Not recommended: Accidental Cartesian products!

```
SELECT ...
FROM   Table1, Table2
WHERE  <where_predicate>;
```

Inner Joins

- Return only rows where a match is found in both input tables
- Match rows based on attributes supplied in predicate
- If join predicate operator is =, also known as equi-join

```
SELECT emp.FirstName, ord.Amount
FROM HR.Employee AS emp
[INNER] JOIN Sales.SalesOrder AS ord
ON emp.EmployeeID = ord.EmployeeID
```



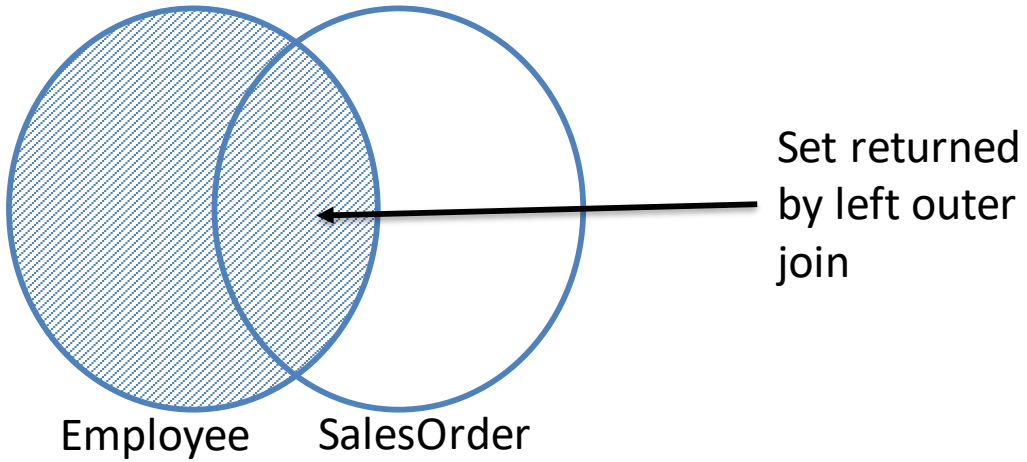
DEMO

Using Inner Joins

Outer Joins

- Return all rows from one table and any matching rows from second table
- One table's rows are "preserved"
  - Designated with LEFT, RIGHT, FULL keyword
  - All rows from preserved table output to result set
- Matches from other table retrieved
- Additional rows added to results for non-matched rows
  - NULLs added in places where attributes do not match
- Example: Return all employees and for those who have taken orders, return the order amount. Employees without matching orders will display NULL for order amount.

```
SELECT emp.FirstName, ord.Amount
FROM HR.Employee AS emp
LEFT [OUTER] JOIN Sales.SalesOrder AS ord
ON emp.EmployeeID = ord.EmployeeID;
```



DEMO

Using Outer Joins

Cross Joins

- Combine each row from first table with each row from second table
- All possible combinations output
- Logical foundation for inner and outer joins
  - Inner join starts with Cartesian product, adds filter
  - Outer join takes Cartesian output, filtered, adds back non-matching rows (with NULL placeholders)
- Due to Cartesian product output, not typically a desired form of join
  - Some useful exceptions:
    - Table of numbers, generating data for testing

Employee	
EmployeeID	FirstName
1	Dan
2	Aisha

Product	
ProductID	Name
1	Widget
2	Gizmo

```
SELECT emp.FirstName, prd.Name
FROM HR.Employee AS emp
CROSS JOIN Production.Product AS prd;
```

Result	
FirstName	Name
Dan	Widget
Dan	Gizmo
Aisha	Widget
Aisha	Gizmo

DEMO

Using Cross Joins

Self Joins

- Compare rows in same table to each other
- Create two instances of same table in FROM clause
  - At least one alias required
- Example: Return all employees and the name of the employee's manager

Employee		
EmployeeID	FirstName	ManagerID
1	Dan	NULL
2	Aisha	1
3	Rosie	1
4	Naomi	3

```
SELECT emp.FirstName AS Employee,
       man.FirstName AS Manager
FROM HR.Employee AS emp
LEFT JOIN HR.Employee AS man
ON emp.ManagerID = man.EmployeeID;
```

Result	
Employee	Manager
Dan	*NULL*
Aisha	Dan
Rosie	Dan
Naomi	Aisha

DEMO

Using Self Joins

Querying Multiple Tables with Joins

- Join Concepts
- Join Syntax
- Inner Joins
- Outer Joins
- Cross Joins
- Self Joins

- Lab: Querying Multiple Tables with Joins

Microsoft