Microsoft

5: Using joins and subqueries

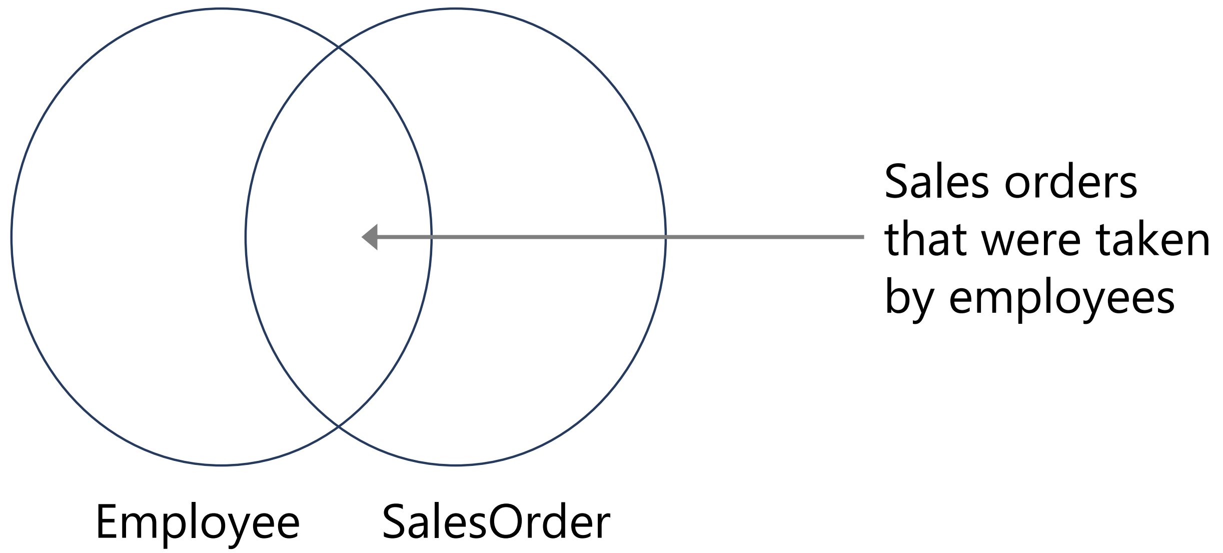
Agenda

- Using joins
- Using subqueries

1: Using joins

Join concepts

It can help to think of the tables as sets in a Venn diagram



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

Join syntax

ANSI SQL-92

- Tables joined by JOIN operator in FROM clause
  - Preferred syntax

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

ANSI SQL-89

- Tables listed in FROM clause with join predicate in WHERE clause
  - Not recommended: can lead to accidental Cartesian products!

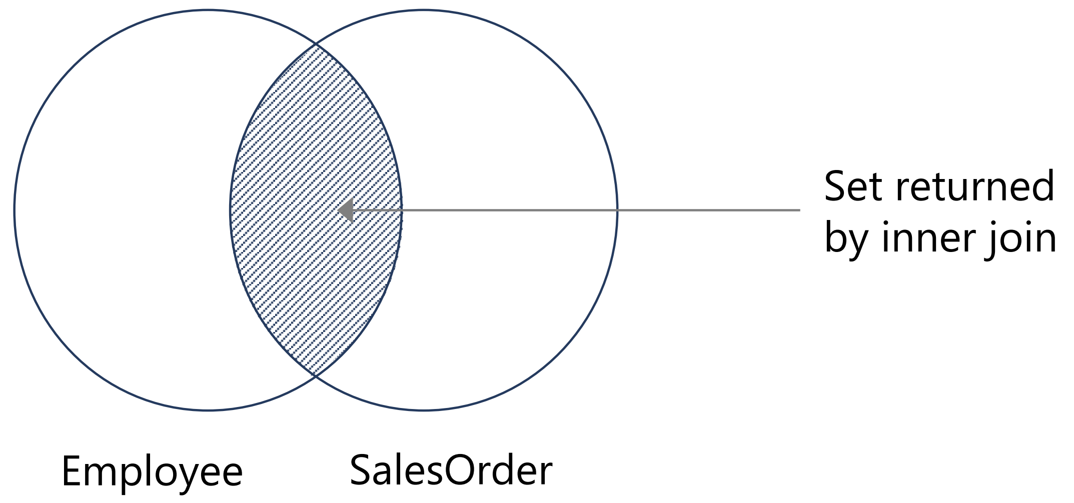
```
SELECT ...
FROM   Table1, Table2
WHERE  <predicate>;
```

Inner joins

Return only rows where a match is found in both input tables

- Match rows based on criteria supplied in the join 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
```

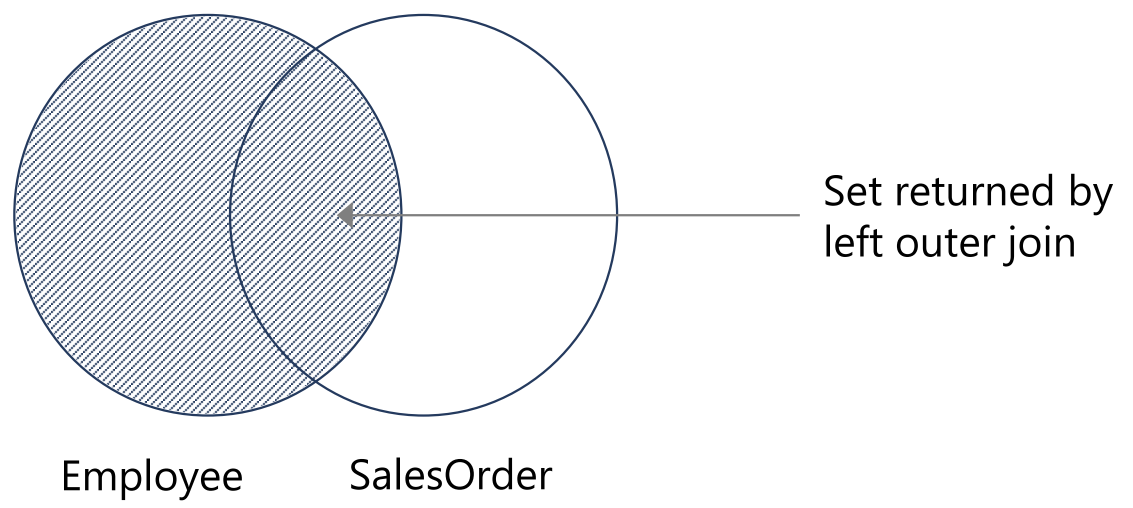


Outer joins

Return all rows from one table and any matching rows from second table

- Outer table's rows are "preserved"
  - Designated with LEFT, RIGHT, FULL keyword
  - All rows from preserved table output to result set
- Matches from inner table retrieved
- NULLs added in places where attributes do not match

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



Cross joins

Combine all rows from both tables

- 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)

Cartesian product output is typically undesired

- Some useful exceptions:
  - Table of numbers
  - Generating data for testing

Employee

EmployeeID	FirstName
1	Dan
2	Aisha

Product

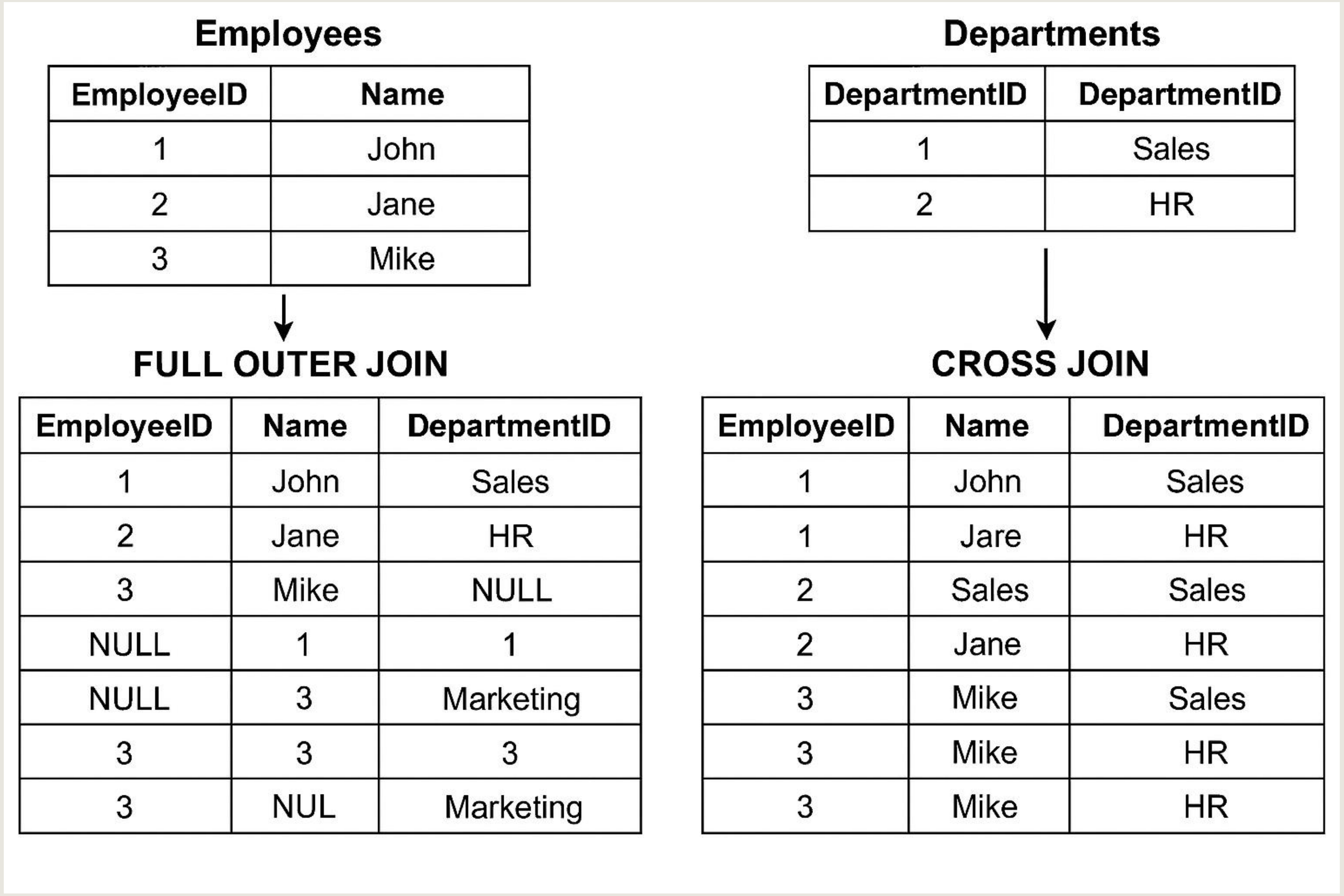
ProductID	Name
1	Widget
2	Gizmo

```
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

FULL OUTER JOIN vs. CROSS JOIN



Employees

EmployeeID	Name
1	John
2	Jane
3	Mike

Departments

DepartmentID	DepartmentID
1	Sales
2	HR

FULL OUTER JOIN

EmployeeID	Name	DepartmentID
1	John	Sales
2	Jane	HR
3	Mike	NULL
NULL	1	1
NULL	3	Marketing
3	3	3
3	NUL	Marketing

CROSS JOIN

EmployeeID	Name	DepartmentID
1	John	Sales
1	Jare	HR
2	Sales	Sales
2	Jane	HR
3	Mike	Sales
3	Mike	HR
3	Mike	HR

Self joins

- Compare rows in a table to other rows in same table
- Create two instances of same table in FROM clause
  - At least one alias required

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

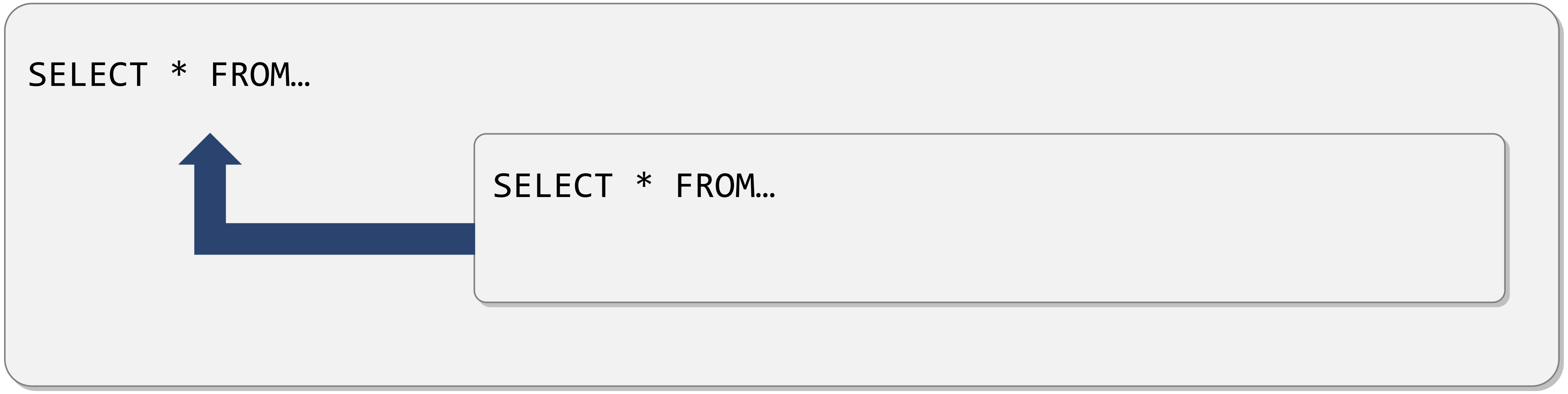
Employee	Manager
Dan	*NULL*
Aisha	Dan
Rosie	Dan
Naomi	Rosie

Lab: Query multiple tables with joins

- https://microsoftlearning.github.io/dp-080-Transact-SQL/Instructions/Labs/03a-joins.html
- Use inner joins
- Use outer joins
- Use a cross join
- Use a self join

Lesson 2: Using subqueries

Introduction to subqueries



Subqueries are nested queries: queries within queries

Results of inner query passed to outer query

- Inner query acts like an expression from perspective of the outer query

Scalar or multi-valued subqueries?

Scalar subquery returns single value to outer query

- Can be used anywhere single-valued expression is used: SELECT, WHERE, and so on

```
SELECT SalesOrderID, ProductID, OrderQty
FROM Sales.SalesOrderDetail
WHERE SalesOrderID =
    (SELECT MAX(SalesOrderID)
    FROM Sales.SalesOrderHeader);
```

Multi-valued subquery returns multiple values as a single column set to the outer query

- Used with IN predicate

```
SELECT CustomerID, SalesOrderID
FROM Sales.SalesOrderHeader
WHERE CustomerID IN (
    SELECT CustomerID
    FROM Sales.Customer
    WHERE CountryRegion = 'Canada');
```

Self-contained or correlated subqueries?

Most subqueries are self-contained and have no connection with the outer query other than passing results to it

Correlated subqueries refer to elements of tables used in outer query

- Dependent on outer query, cannot be executed separately
- Behaves as if inner query is executed once per outer row
- May return scalar value or multiple values

```
SELECT SalesOrderID, CustomerID, OrderDate
FROM SalesLT.SalesOrderHeader AS o1
WHERE SalesOrderID =
     (SELECT MAX(SalesOrderID)
     FROM SalesLT.SalesOrderHeader AS o2
     WHERE o2.CustomerID = o1.CustomerID)
ORDER BY CustomerID, OrderDate;
```

Lab: Use subqueries

- https://microsoftlearning.github.io/dp-080-Transact-SQL/Instructions/Labs/03b-subqueries.html
  - Use simple subqueries
  - Use correlated subqueries

Review

1 **You must return a list of all sales employees that have taken sales orders. Employees who have not taken sales orders should not be included in the results. Which type of join is required?**

[✓ INNER]
☐ LEFT OUTER
☐ FULL OUTER

2 **What does the following query return?**

```
SELECT p.Name, c.Name FROM Store.Product AS p CROSS JOIN Store.Category AS c;
```

☐ Only data rows where the product name is the same as the category name.
☐ Only rows where the product name is not the same as the category name.
[✓ Every combination of product and category name.]

3 **A correlated subquery...**

☐ Returns a single scalar value
☐ Returns multiple columns and rows
[✓ References a value in the outer query]

Microsoft