Using Set Operators

Module Overview

- Writing Queries with the UNION Operator
- Using EXCEPT and INTERSECT

Lesson 1: Writing Queries with the UNION Operator

- Interactions Between Sets
- Using the UNION Operator
- Using the UNION ALL Operator
- Demonstration: Using UNION and UNION ALL

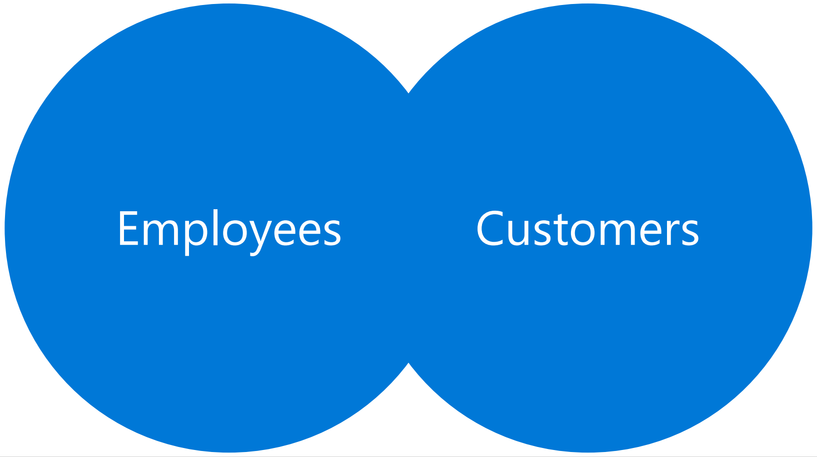
Interactions Between Sets

- The results of two input queries may be further manipulated
- Sets may be combined, compared, or operated against each other
- Both sets must have the same number of compatible columns
- ORDER BY not allowed in input queries, but may be used for result of set operation
- NULLs considered equal when comparing sets

```
<SELECT query_1>
<set_operator>
<SELECT query_2>
[ORDER BY <sort_list>];
```

Using the UNION Operator

- UNION returns a result set of distinct rows combined from both input sets
- Duplicates are removed during query processing (affects performance)



```sql
-- only distinct rows from both queries are returned
SELECT country, region, city FROM HR.Employees
UNION
SELECT country, region, city FROM Sales.Customers;
```

Using the UNION ALL Operator

- UNION ALL returns a result set with all rows from both input sets
- To avoid the performance penalty caused by filtering duplicates, use UNION ALL over UNION whenever requirements allow it

```
-- all rows from both queries will be returned
SELECT country, region, city FROM HR.Employees
UNION ALL
SELECT country, region, city FROM Sales.Customers;
```

Demonstration: Using UNION and UNION ALL

In this demonstration, you will see how to:

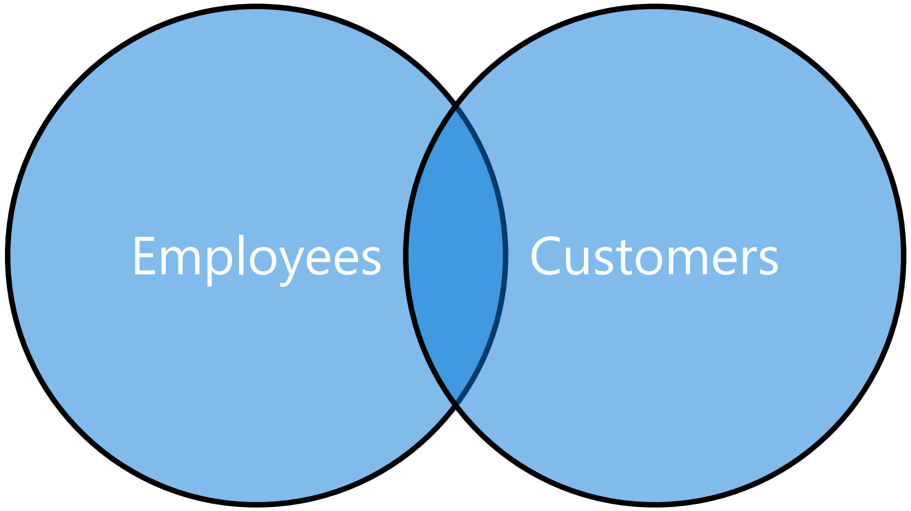
- Use UNION and UNION ALL

Lesson 2: Using EXCEPT and INTERSECT

- Using the INTERSECT Operator
- Using the EXCEPT Operator
- Demonstration: Using EXCEPT and INTERSECT

Using the INTERSECT Operator

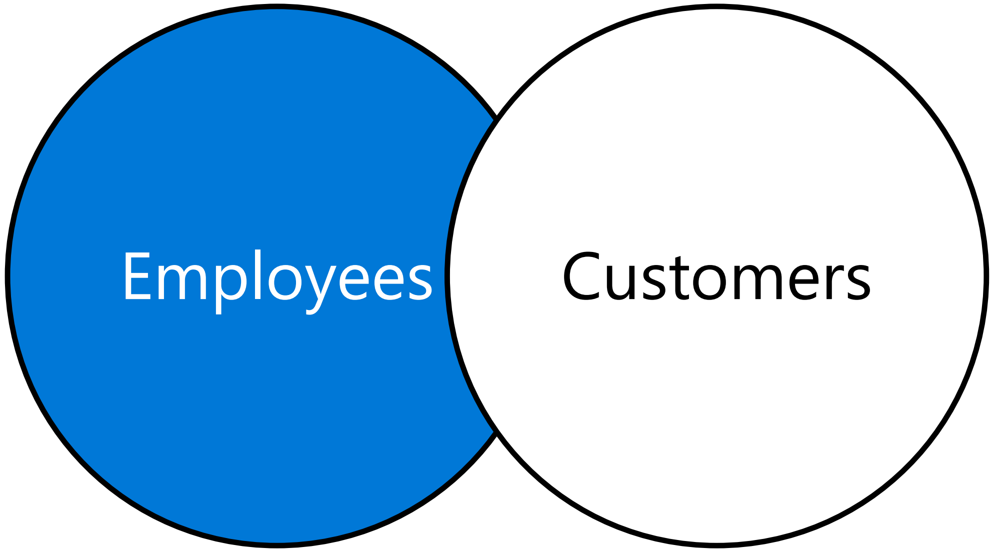
- INTERSECT returns the distinct set of rows that appear in both input result sets



```sql
-- only rows that exist in both queries will be returned
SELECT country, region, city FROM HR.Employees
INTERSECT
SELECT country, region, city FROM Sales.Customers;
```

Using the EXCEPT Operator

- EXCEPT returns only distinct rows that appear in the left set but not the right
  - The order in which sets are specified matters



```
-- only rows from Employees will be returned
SELECT country, region, city FROM HR.Employees
EXCEPT
SELECT country, region, city FROM Sales.Customers;
```

Demonstration: Using EXCEPT and INTERSECT

In this demonstration, you will see how to:

- Use INTERSECT and EXCEPT