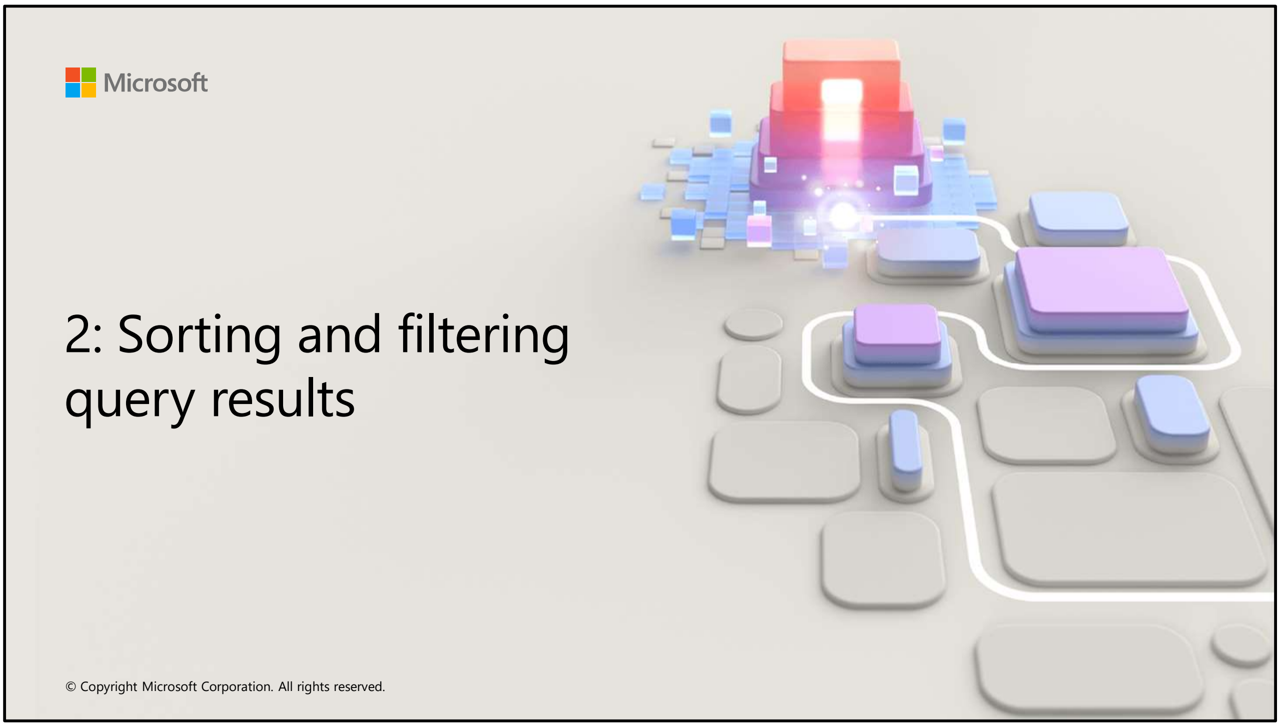
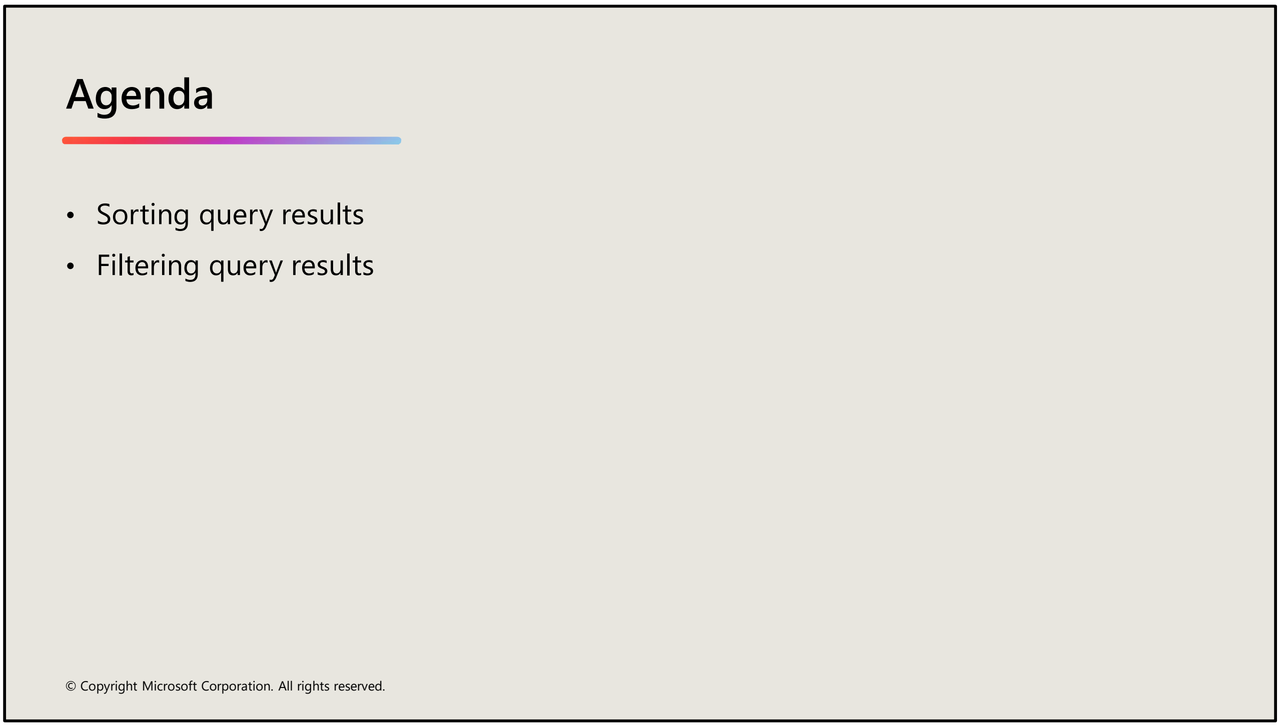


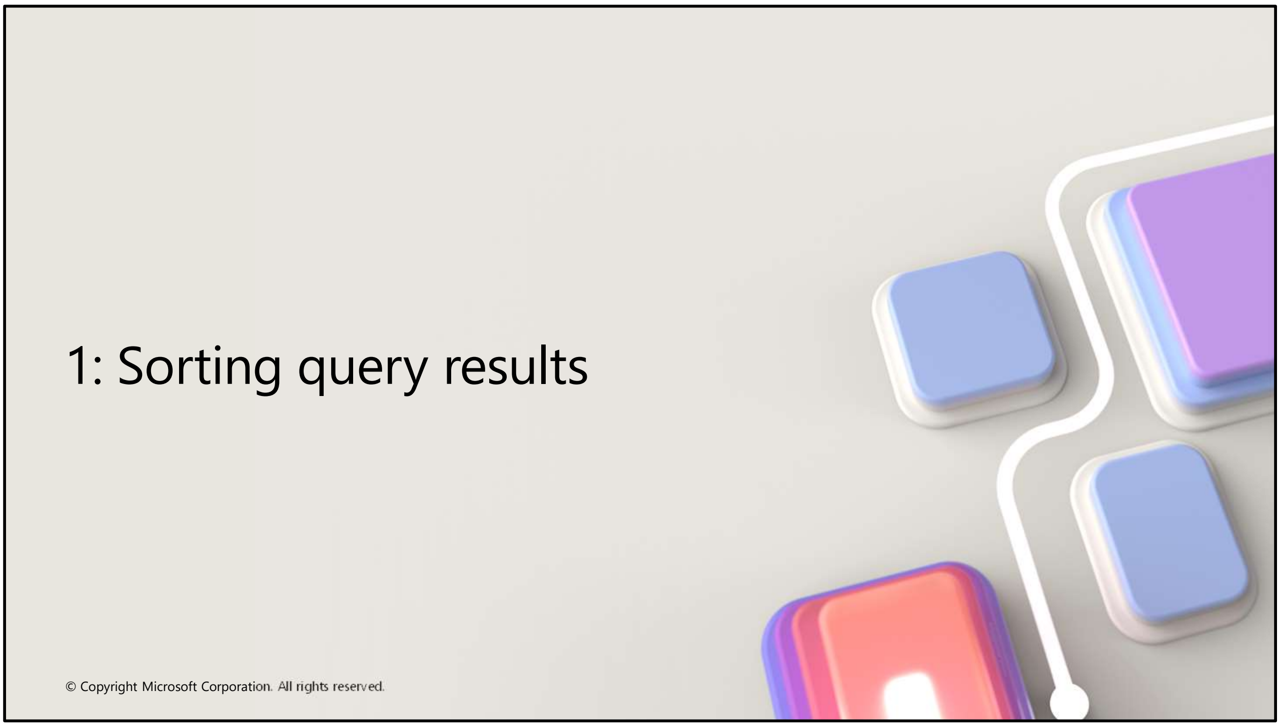
Microsoft

2: Sorting and filtering query results

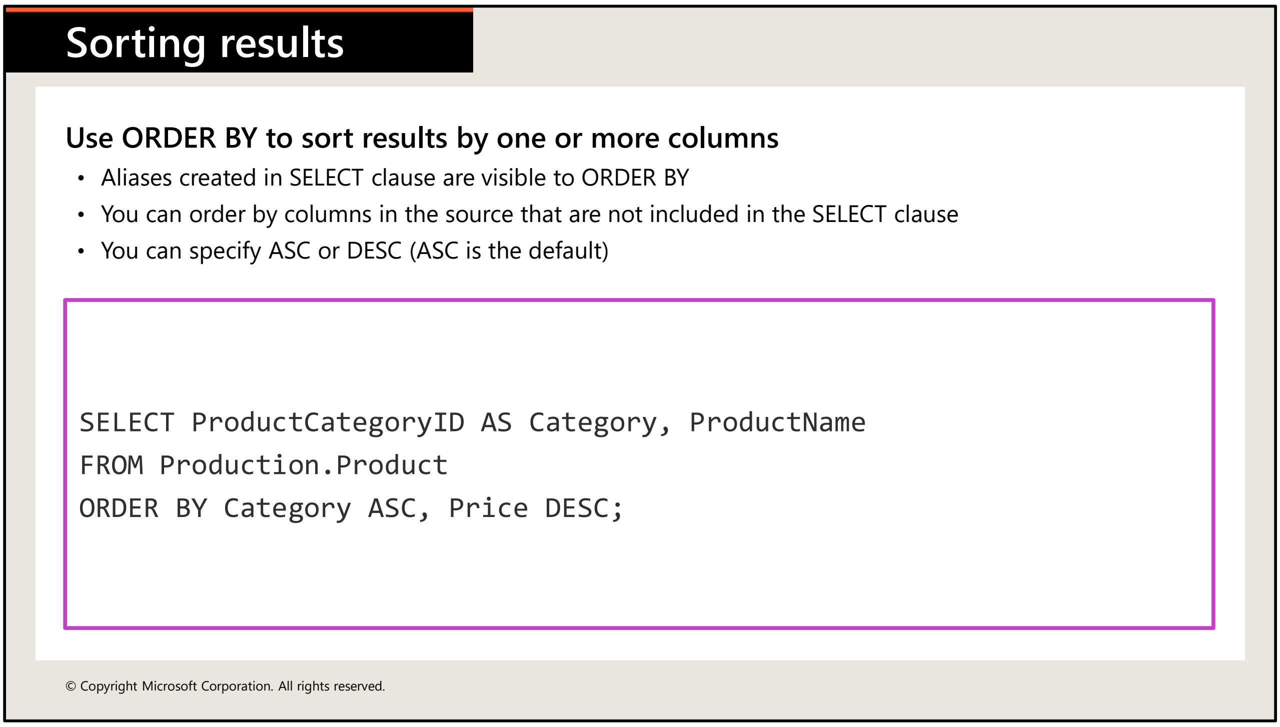


Agenda

- Sorting query results
- Filtering query results



1: Sorting query results

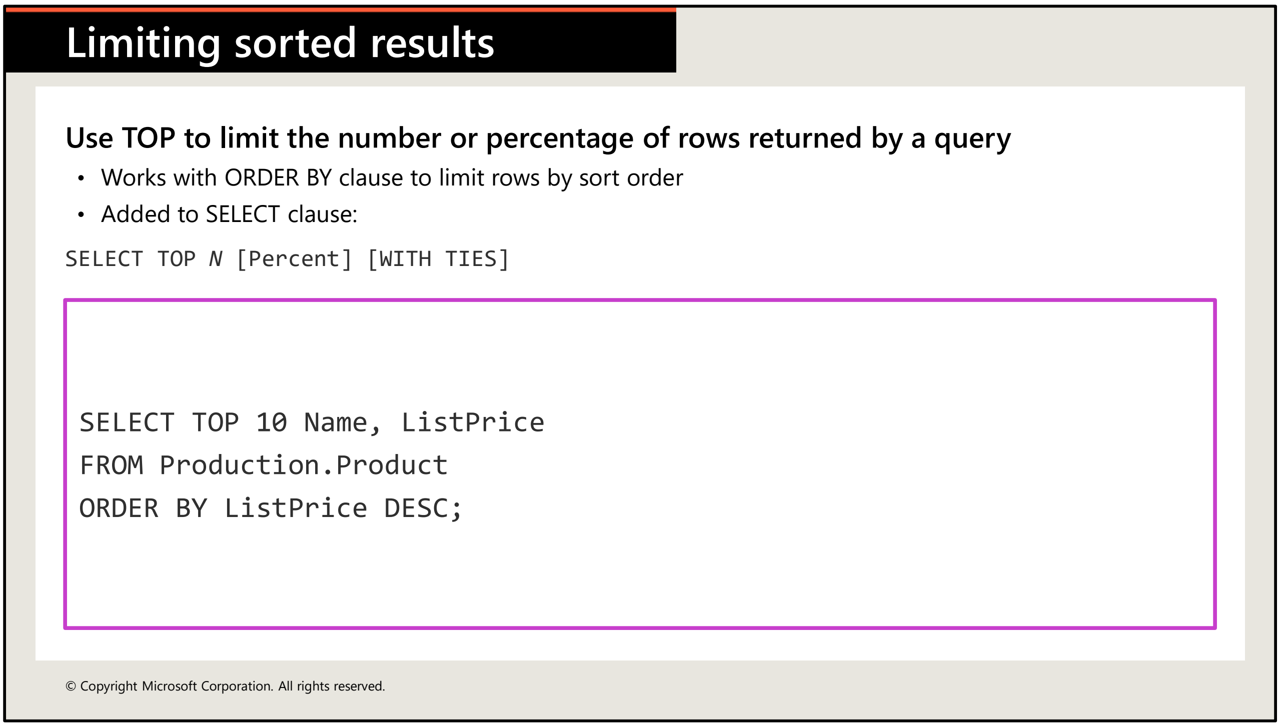


Sorting results

Use ORDER BY to sort results by one or more columns

- Aliases created in SELECT clause are visible to ORDER BY
- You can order by columns in the source that are not included in the SELECT clause
- You can specify ASC or DESC (ASC is the default)

```
SELECT ProductCategoryID AS Category, ProductName
FROM Production.Product
ORDER BY Category ASC, Price DESC;
```



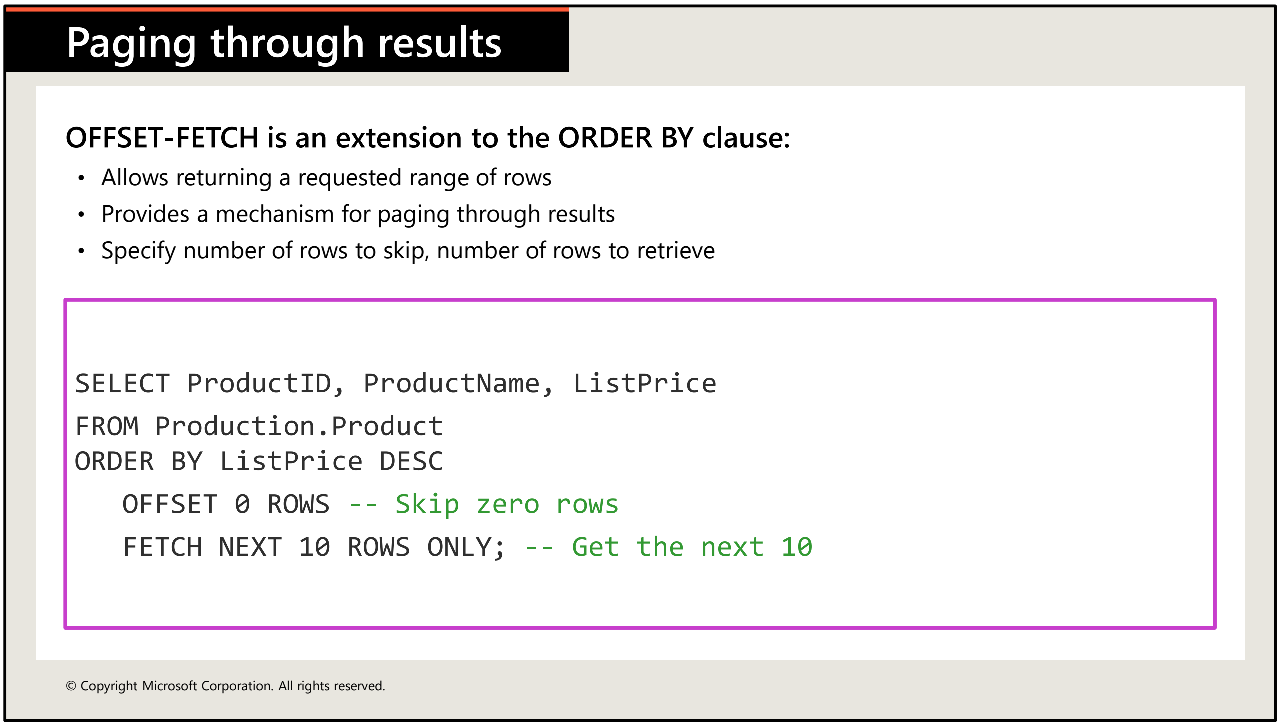
Limiting sorted results

Use TOP to limit the number or percentage of rows returned by a query

- Works with ORDER BY clause to limit rows by sort order
- Added to SELECT clause:

```
SELECT TOP N [Percent] [WITH TIES]
```

```
SELECT TOP 10 Name, ListPrice
FROM Production.Product
ORDER BY ListPrice DESC;
```

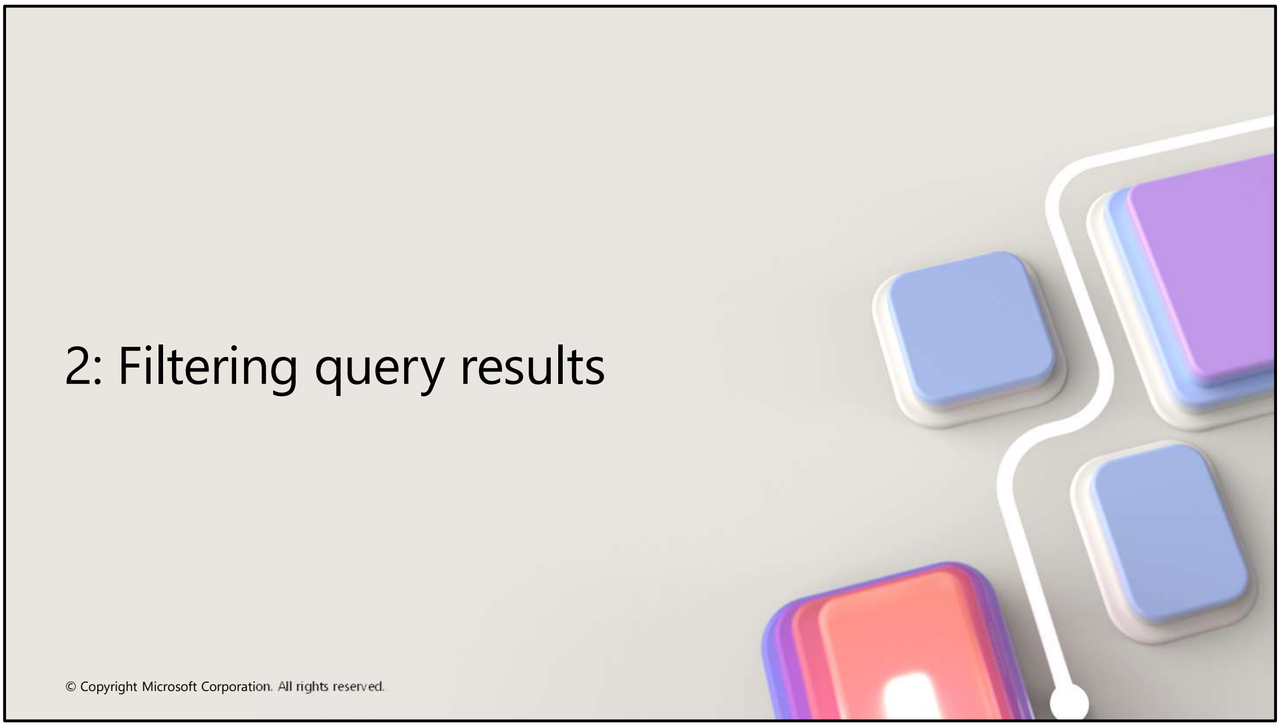


Paging through results

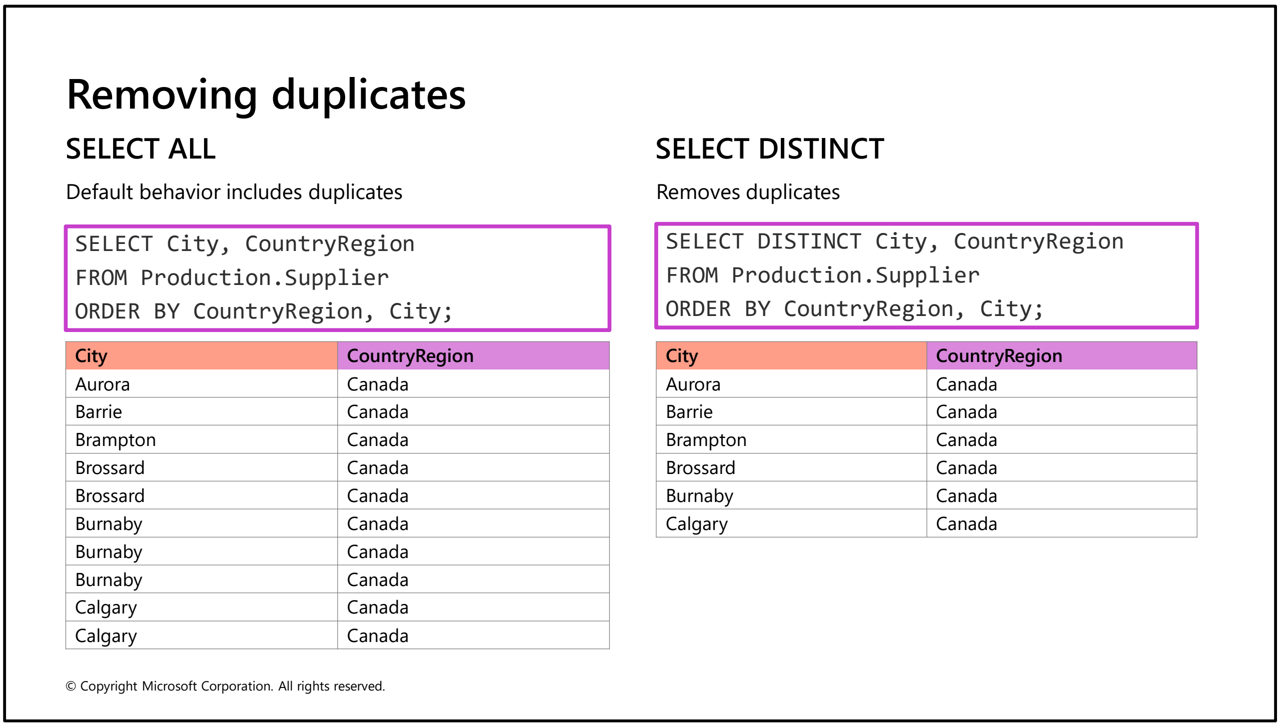
OFFSET-FETCH is an extension to the ORDER BY clause:

- Allows returning a requested range of rows
- Provides a mechanism for paging through results
- Specify number of rows to skip, number of rows to retrieve

```
SELECT ProductID, ProductName, ListPrice
FROM Production.Product
ORDER BY ListPrice DESC
   OFFSET 0 ROWS -- Skip zero rows
   FETCH NEXT 10 ROWS ONLY; -- Get the next 10
```



2: Filtering query results



Removing duplicates

SELECT ALL

Default behavior includes duplicates

```
SELECT City, CountryRegion
FROM Production.Supplier
ORDER BY CountryRegion, City;
```

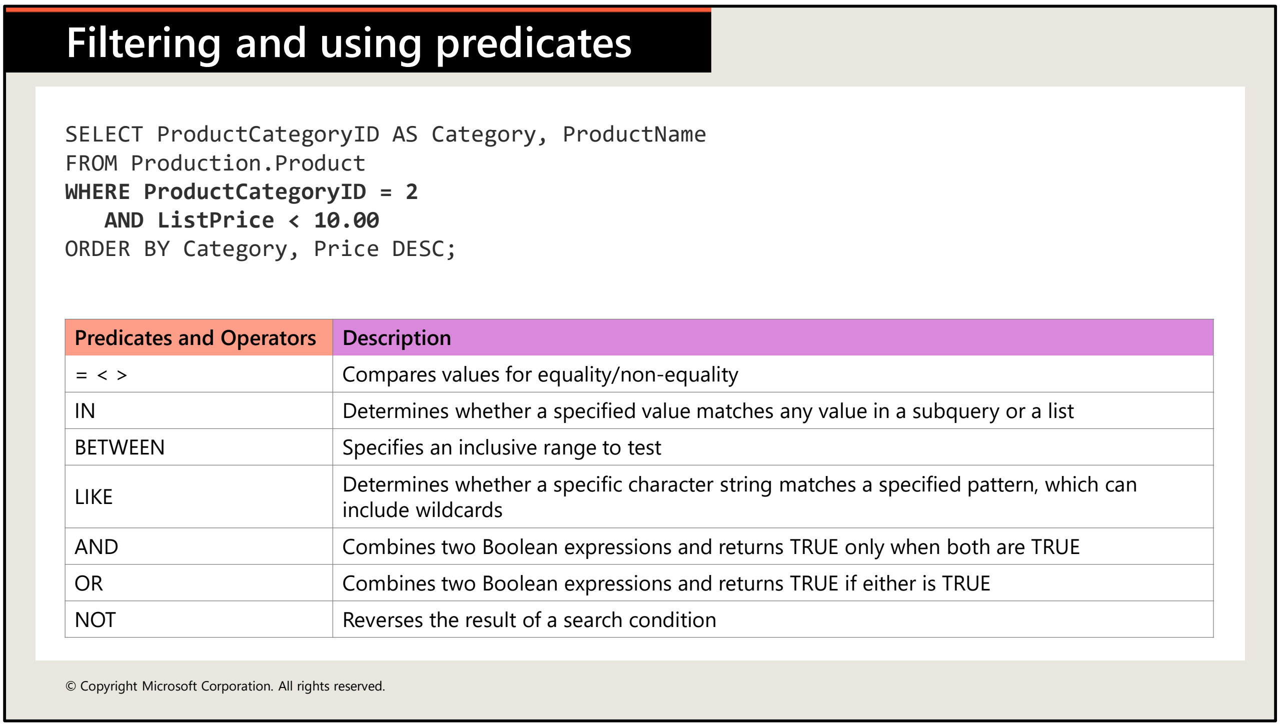
City	CountryRegion
Aurora	Canada
Barrie	Canada
Brampton	Canada
Brossard	Canada
Brossard	Canada
Burnaby	Canada
Burnaby	Canada
Burnaby	Canada
Calgary	Canada
Calgary	Canada

SELECT DISTINCT

Removes duplicates

```
SELECT DISTINCT City, CountryRegion
FROM Production.Supplier
ORDER BY CountryRegion, City;
```

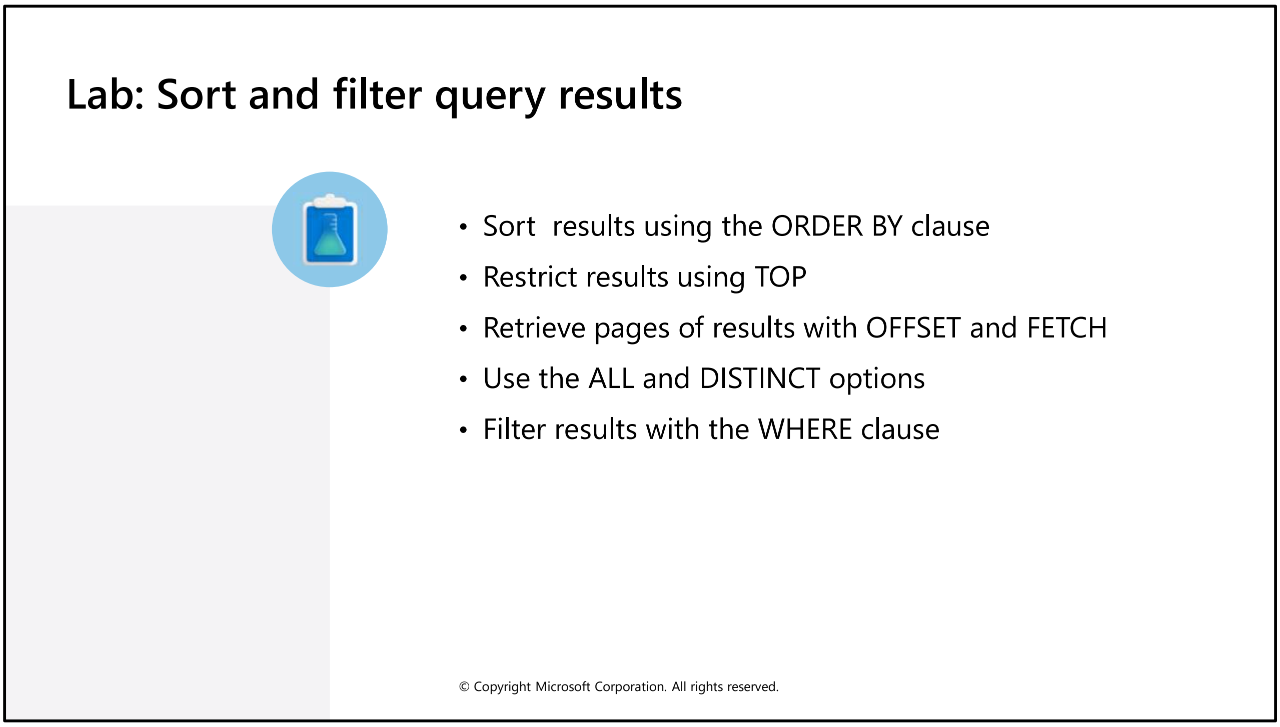
City	CountryRegion
Aurora	Canada
Barrie	Canada
Brampton	Canada
Brossard	Canada
Burnaby	Canada
Calgary	Canada



Filtering and using predicates

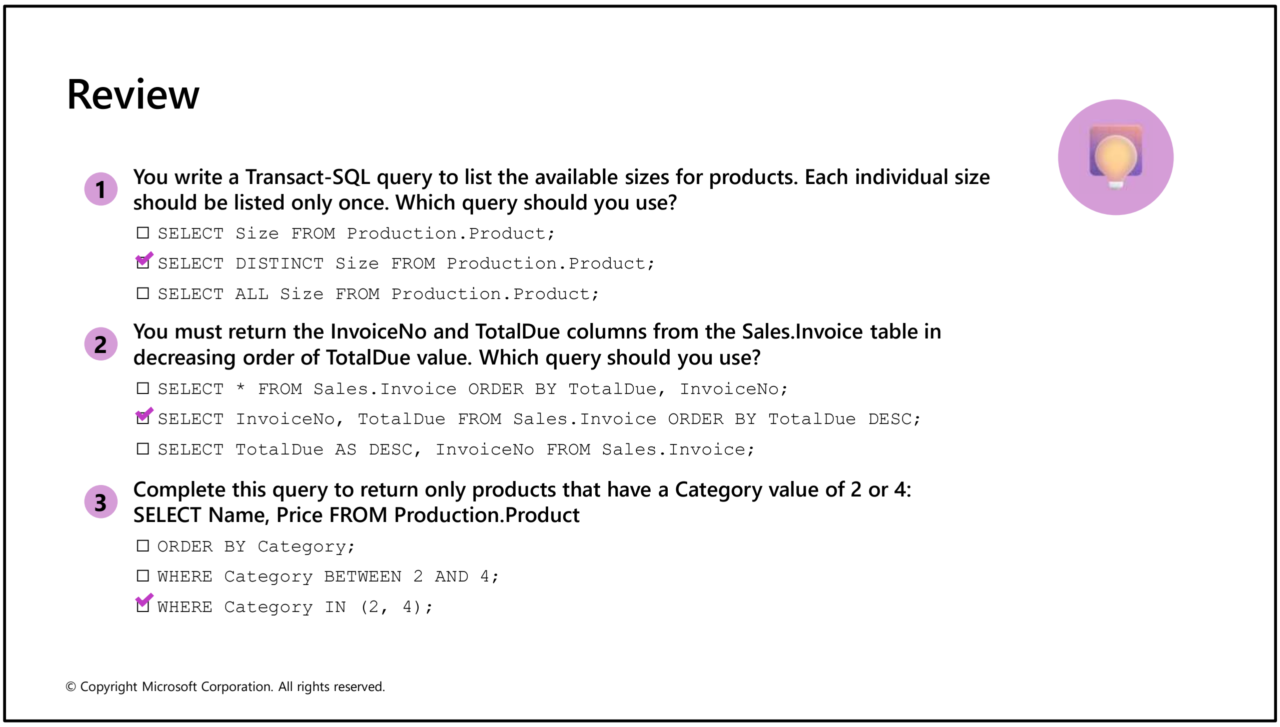
```
SELECT ProductCategoryID AS Category, ProductName
FROM Production.Product
WHERE ProductCategoryID = 2
   AND ListPrice < 10.00
ORDER BY Category, Price DESC;
```

Predicates and Operators	Description
= < >	Compares values for equality/non-equality
IN	Determines whether a specified value matches any value in a subquery or a list
BETWEEN	Specifies an inclusive range to test
LIKE	Determines whether a specific character string matches a specified pattern, which can include wildcards
AND	Combines two Boolean expressions and returns TRUE only when both are TRUE
OR	Combines two Boolean expressions and returns TRUE if either is TRUE
NOT	Reverses the result of a search condition



Lab: Sort and filter query results

- Sort results using the ORDER BY clause
- Restrict results using TOP
- Retrieve pages of results with OFFSET and FETCH
- Use the ALL and DISTINCT options
- Filter results with the WHERE clause



Review

1 You write a Transact-SQL query to list the available sizes for products. Each individual size should be listed only once. Which query should you use?

□ `SELECT Size FROM Production.Product;`

☑ `SELECT DISTINCT Size FROM Production.Product;`

□ `SELECT ALL Size FROM Production.Product;`

2 You must return the InvoiceNo and TotalDue columns from the Sales.Invoice table in decreasing order of TotalDue value. Which query should you use?

□ `SELECT * FROM Sales.Invoice ORDER BY TotalDue, InvoiceNo;`

☑ `SELECT InvoiceNo, TotalDue FROM Sales.Invoice ORDER BY TotalDue DESC;`

□ `SELECT TotalDue AS DESC, InvoiceNo FROM Sales.Invoice;`

3 Complete this query to return only products that have a Category value of 2 or 4:
SELECT Name, Price FROM Production.Product

□ `ORDER BY Category;`

□ `WHERE Category BETWEEN 2 AND 4;`

☑ `WHERE Category IN (2, 4);`

Use the slide animation to reveal the correct answers.

Microsoft