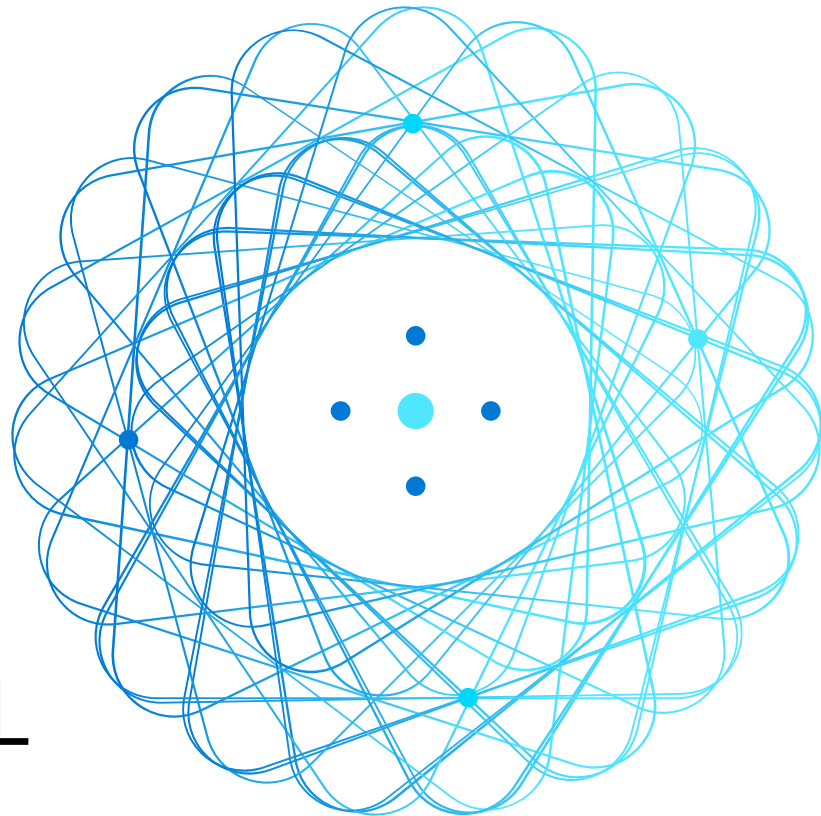


# Course DP-080: Querying Data with Microsoft Transact-SQL



# About This Course

Learn how to write queries using SQL Server and Azure SQL Database

- This course focuses on learning core Transact-SQL syntax used to work with data for reporting and application development
  - Using SELECT to retrieve columns from a table
  - Sorting and filtering query results
  - Using joins and subqueries to retrieve data from multiple tables
  - Using built-in functions, aggregations, and groupings
  - Inserting, updating, and deleting data
- Additional learning materials are available on Microsoft Learn

# Course Agenda

Day 1: Getting Started with Transact-SQL

Day 2: Filtering Query Results and Data type functions

Day 3: Introduction to Join and Union

Day 4: Using Joins and Subqueries

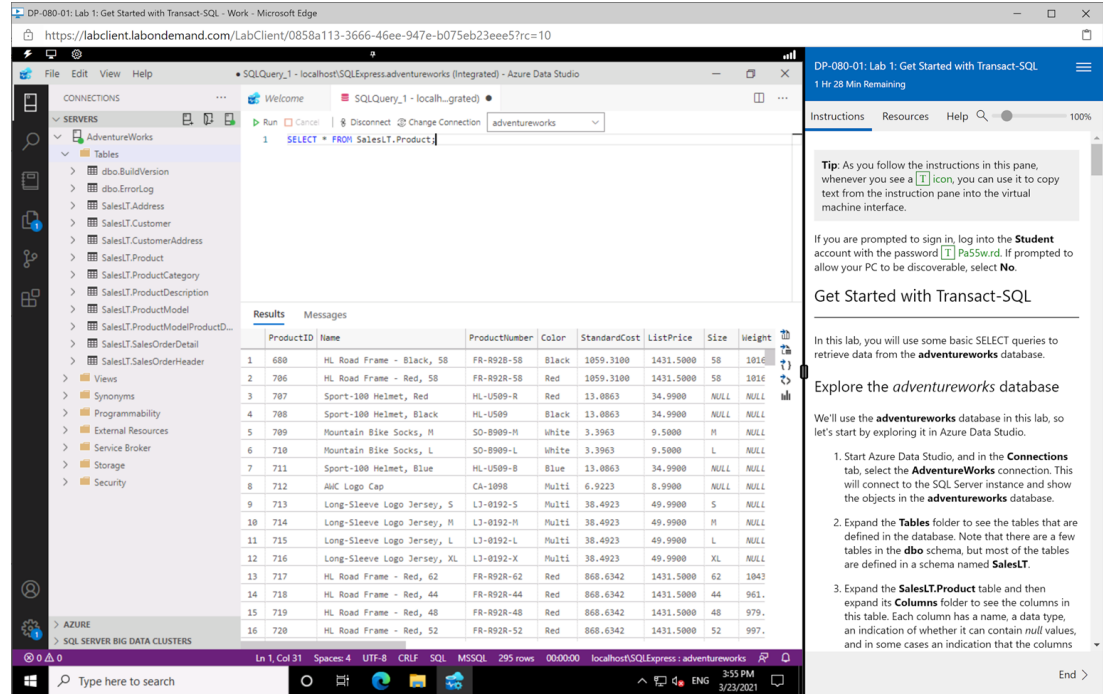
Day 5: Aggregate functions and Group by

Day 6: Practise and course summarize

# Lab Environment

## Hosted Virtual Machine

- Windows 10
- SQL Server Express
- Azure Data Studio



The screenshot shows the Azure Data Studio interface. The top bar indicates the connection is 'localhost\SQLEXPRESS:adventureworks'. The left sidebar shows the 'CONNECTIONS' pane with 'AdventureWorks' selected. The main pane displays a SQL query: `SELECT * FROM SalesLT.Product`. The 'Results' pane shows a table with columns: ProductID, Name, ProductNumber, Color, StandardCost, ListPrice, Size, and Weight. The table contains 16 rows of data.

ProductID	Name	ProductNumber	Color	StandardCost	ListPrice	Size	Weight
1	HL Road Frame - Black, 58	FR-R928-SB	Black	1059.3100	1431.5000	58	1016
2	HL Road Frame - Red, 58	FR-R928-SB	Red	1059.3100	1431.5000	58	1016
3	Sport-100 Helmet, Red	HL-U509-R	Red	13.0863	34.9900	NULL	NULL
4	Sport-100 Helmet, Black	HL-U509	Black	13.0863	34.9900	NULL	NULL
5	Mountain Bike Socks, M	SO-B909-M	White	3.3963	9.5000	M	NULL
6	Mountain Bike Socks, L	SO-B909-L	White	3.3963	9.5000	L	NULL
7	Sport-100 Helmet, Blue	HL-U509-B	Blue	13.0863	34.9900	NULL	NULL
8	AHC Logo Cap	CA-1098	Multi	6.9223	8.9900	NULL	NULL
9	Long-Sleeve Logo Jersey, S	LJ-0192-S	Multi	38.4923	49.9900	S	NULL
10	Long-Sleeve Logo Jersey, M	LJ-0192-M	Multi	38.4923	49.9900	M	NULL
11	Long-Sleeve Logo Jersey, L	LJ-0192-L	Multi	38.4923	49.9900	L	NULL
12	Long-Sleeve Logo Jersey, XL	LJ-0192-X	Multi	38.4923	49.9900	XL	NULL
13	HL Road Frame - Red, 62	FR-R928-62	Red	868.6342	1431.5000	62	1043
14	HL Road Frame - Red, 44	FR-R928-44	Red	868.6342	1431.5000	44	961
15	HL Road Frame - Red, 48	FR-R928-48	Red	868.6342	1431.5000	48	979
16	HL Road Frame - Red, 52	FR-R928-52	Red	868.6342	1431.5000	52	997

The right sidebar contains a 'Get Started with Transact-SQL' section with instructions and a list of steps to explore the database.

**Get Started with Transact-SQL**

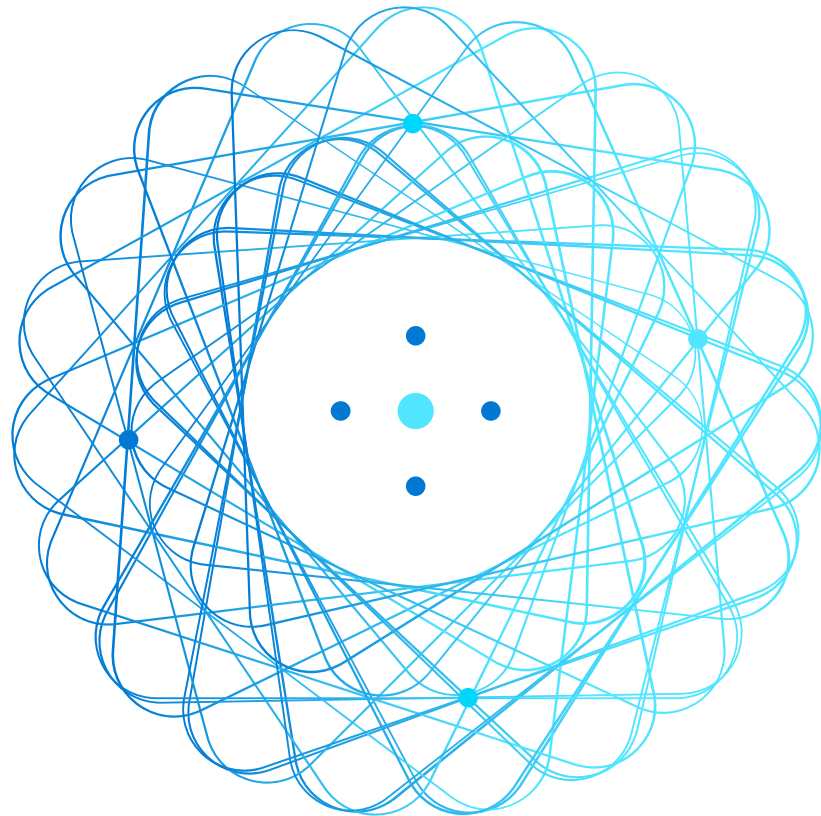
In this lab, you will use some basic SELECT queries to retrieve data from the **adventureworks** database.

**Explore the *adventureworks* database**

We'll use the **adventureworks** database in this lab, so let's start by exploring it in Azure Data Studio.

1. Start Azure Data Studio, and in the **CONNECTIONS** tab, select the **AdventureWorks** connection. This will connect to the SQL Server instance and show the objects in the **adventureworks** database.
2. Expand the **Tables** folder to see the tables that are defined in the database. Note that there are a few tables in the **dbo** schema, but most of the tables are defined in a schema named **SalesLT**.
3. Expand the **SalesLT.Product** table and then expand its **Columns** folder to see the columns in this table. Each column has a name, a data type, an indication of whether it can contain null values, and in some cases an indication that the columns

# Getting Started with Transact-SQL



# Module Agenda



Introduction to Transact-SQL



Using the SELECT Statement



Sorting and Limiting Query Results

# Lesson 1: Introduction to Transact-SQL



# What is Transact-SQL?

## Structured Query Language (SQL)

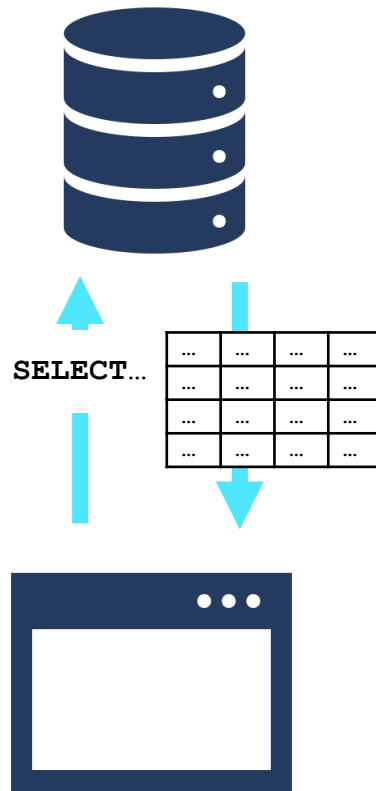
- Developed in the 1970s as a language for querying databases
- Adopted as a standard by ANSI and ISO standards bodies
- Widely used across multiple database systems

## Microsoft's implementation is Transact-SQL

- Often referred to as T-SQL
- Query language for SQL Server (the box product), Azure SQL Database (the cloud platform), and other Microsoft relational database services (RDBMs)

SQL is *declarative*, not *procedural*

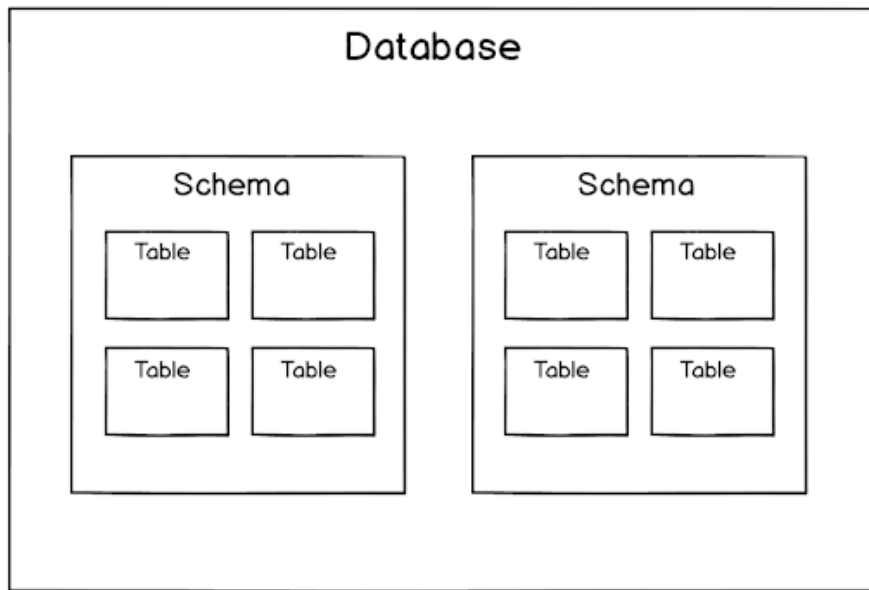
- Describe what you want, don't specify steps





# Main SQL Concept

- Server
- Database
- Schema
- Table
- Column and Row



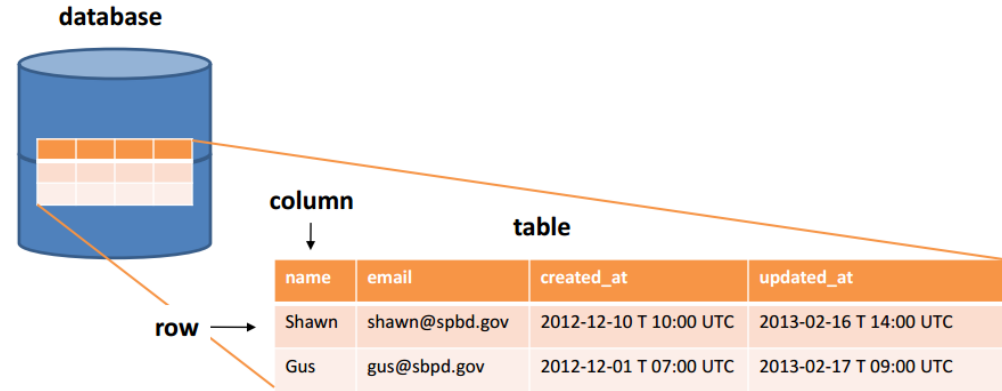
- Data server is the warehouse that hosts your databases
  - In relational databases, data is organized in a hierarchy, like files and folders
- Databases have schemas. Schemas have tables. Tables have columns.

# Schemas

- Schema defines the database structure (how the database is constructed and managed).
- A database can have one or multiple schemas.
- The schema name must be distinct from the name of any existing schema in the current database (i.e. schema name must be unique).
- The dbo schema is the default schema for a newly created database.
- Benefits of using schemas:
  - ✓ Logically organize your DB
  - ✓ Improve security by allowing which users can access which part of your DB

# Tables

- Table is the primary storage object for data in a relational database. A database most often contains one or more tables.
- A table consists of row(s) and column(s), both of which hold the data.
- A table takes up physical space in a database and can be permanent or temporary



# Columns and Rows in a Table

## Columns

- A column (field) is also called an attribute.
- The columns in a table hold specific types of data, such as name, age or address of customers.

## Rows

- A row is a record of data in a database table.
- For example, a row of data in a customer table might consist of a particular customer's identification number, name, address, phone number, fax number, and so on

# Primary Key and Foreign Key

Most relational databases are *normalized*, with relationships defined between tables through *primary* and *foreign* keys

## Primary key

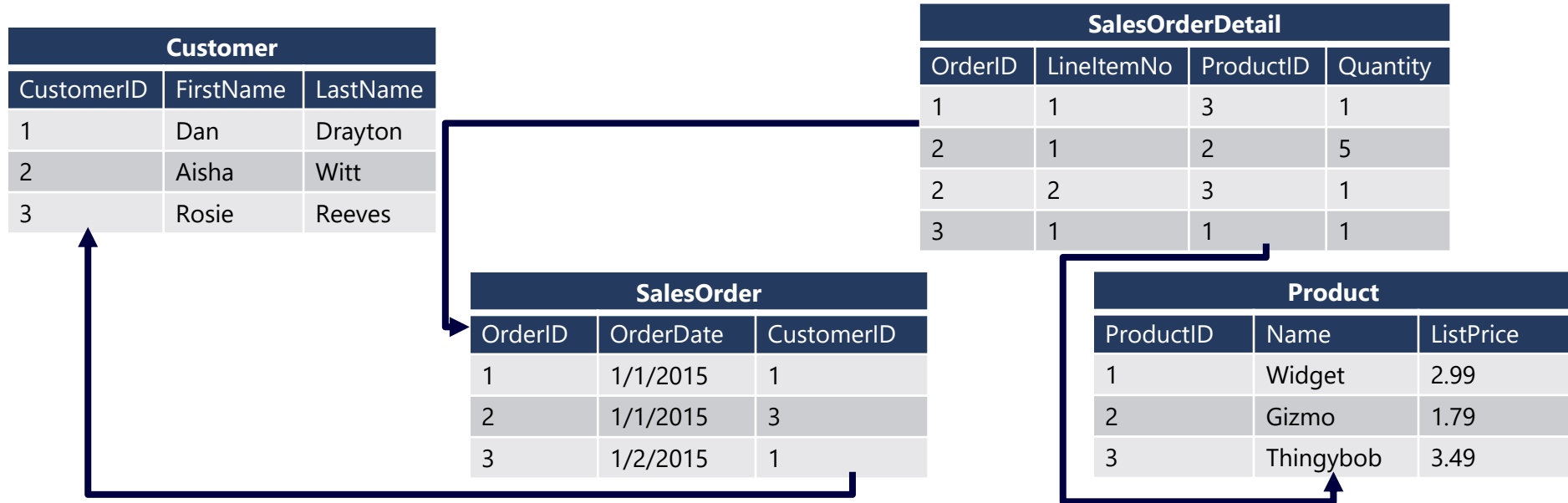
- A primary key is a field (or collection of fields)
- The primary key constraint uniquely identifies each record (row) in a database table.
- Primary keys must contain unique values and cannot contain NULL.
- A table can have only one primary key, which may consist of single or multiple fields (columns)

## Foreign Key

- A foreign key is a key used to link two tables together.
- A foreign key is a field (or collection of fields) in one table that refers to the primary key(s) in another table.

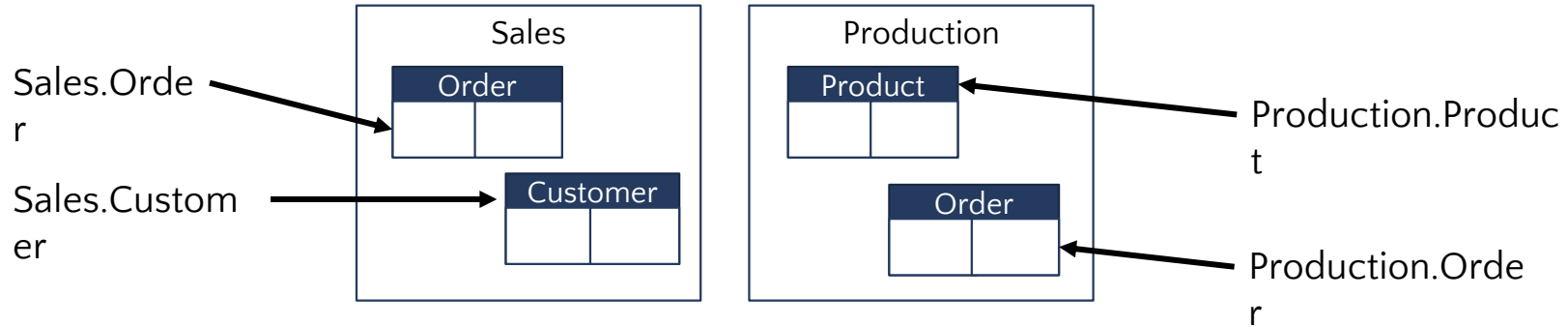
# Primary Key and Foreign Key

Exercise: Determine PK and FK in below tables through existence relationships



# Query Objects name

- Fully-qualified names:  
*[server\_name.][database\_name.][schema\_name.]object\_name*
- Within database context, best practice is to include schema name:  
*schema\_name.object\_name*



# SQL Statement Types

Data Manipulation Language (DML)	Data Definition Language (DDL)	Data Control Language (DCL)
Statements for querying and modifying data: <ul style="list-style-type: none"><li>• SELECT</li><li>• INSERT</li><li>• UPDATE</li><li>• DELETE</li></ul>	Statements for defining database objects: <ul style="list-style-type: none"><li>• CREATE</li><li>• ALTER</li><li>• DROP</li></ul>	Statements for assigning security permissions: <ul style="list-style-type: none"><li>• GRANT</li><li>• REVOKE</li><li>• DENY</li></ul>

A white callout box with a blue border and rounded corners. A blue line points from the box to the 'Data Manipulation Language (DML)' column of the table above.

Focus of this course



## Lesson 2: Using the SELECT Statement



# The SELECT Statement

	Element	Expression	Role
5	SELECT	<select list>	Defines which columns to return
1	FROM	<table source>	Defines table(s) to query
2	WHERE	<search condition>	Filters rows using a predicate
3	GROUP BY	<group by list>	Arranges rows by groups
4	HAVING	<search condition>	Filters groups using a predicate
6	ORDER BY	<order by list>	Sorts the output

```
SELECT ProductLine, COUNT(ProductKey) as NoPro  
duct  
FROM dbo.DimProduct  
WHERE Color IN ('Red', 'Black')  
GROUP BY ProductLine  
HAVING COUNT(ProductKey) > 10  
ORDER BY ProductLine DESC;
```

# Basic SELECT Statement

The SELECT statement is used to select data from a database.

```
SELECT *  
FROM table_name
```

Or

```
SELECT column1, column2, column3  
FROM table_name
```

Try:

```
SELECT EmployeeKey, FirstName, LastName FROM dbo.DimEmployee;
```

```
SELECT * FROM dbo.DimProduct
```

# SQL Comments

Comments are used to explain sections of SQL statements, or to prevent execution of SQL statements.

- Single line comments

```
-- This is a single line comment in SQL  
SELECT ProductKey, Color, ListPrice  
FROM dbo.DimProduct
```

- Multiple-line comments

```
/* This comment can be placed in  
multiple lines in SQL */
```

```
SELECT EmployeeKey /*or even here*/, FirstName, LastName  
FROM dbo.DimEmployee
```

# SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name (only exists for the duration of the query).

Try:

```
SELECT EmployeeKey as 'Ma nhan vien',  
FirstName,  
LastName,  
Gender as 'Gioi tinh'  
FROM dbo.DimEmployee
```

Aliases can be useful when:

- There are more than one table involved in a query (and some of their columns have the same name)
- Functions are used in the query
- Column names are big or not very readable
- Two or more columns are combined together

## Lesson 3: Sorting and Limiting 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, [Name]  
FROM dbo.DimProduct  
ORDER BY Category ASC, ListPrice DESC;
```

# Limiting Results

SELECT TOP Clause: used to specify the number of records to return

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

Try:

```
SELECT TOP 10 * FROM dbo.DimProduct;
```

```
SELECT TOP 10 ProductKey, EnglishProductName, ListPrice  
FROM dbo.DimProduct;
```

```
SELECT TOP 10 PERCENT ProductKey, EnglishProductName, ListPrice  
FROM dbo.DimProduct;
```

```
SELECT TOP 10 ProductKey, EnglishProductName, ListPrice  
FROM dbo.DimProduct  
ORDER BY ListPrice DESC ;
```



# Limiting Results

SELECT DISTINCT Clause: used to return only distinct (unique) records.

```
SELECT DISTINCT column1, column2, column3  
FROM table_name
```

Try:

```
SELECT Title  
FROM dbo.DimEmployee
```

Compare with the result of:

```
SELECT DISTINCT Title  
FROM dbo.DimEmployee
```

