Comprehensive SQL Server Tutorial

DBMS (Database Management System)

A DBMS is software that lets you define, create, maintain and control access to databases. It abstracts away low-level file operations and offers a declarative language (SQL) to work with data.

Core Responsibilities:

1. Data Definition (DDL) – create/alter/drop schemas, tables, indexes.
2. Data Manipulation (DML) – insert, update, delete, and query data.
3. Transaction Management – ensure ACID properties for reliability.
4. Concurrency Control – manage simultaneous access without conflicts.
5. Security & Authorization – grant/revoke privileges.

When & Why Use:

- Handling more data than flat files.
- Need reliable concurrent access and backups.

Example:

```
CREATE TABLE Products (
ProductID INT IDENTITY(1,1) PRIMARY KEY,
Name VARCHAR(100) NOT NULL,
Price DECIMAL(10,2) NOT NULL,
CreatedDate DATETIME DEFAULT GETDATE()
);
```

Best Practices:

- meaningful names;
- define primary keys;
- use defaults.

Interview Tip: explain ACID vs BASE, and implications of violating each.

Entity

An entity is anything you need to store information about—a “thing” with attributes.

Entity Type vs Instance:

- Entity Type = class (e.g., Customer)
- Entity Instance = a row in the Customers table.

When & Why Use:

- During design, model real-world objects.
- Helps normalization to avoid redundancy.

Example ER to SQL:

```
CREATE TABLE Customer (
CustomerID INT PRIMARY KEY IDENTITY(1,1),
FirstName VARCHAR(50) NOT NULL,
LastName VARCHAR(50) NOT NULL,
Email VARCHAR(100) UNIQUE NOT NULL
);
```

Interview Tip: decide entity vs attribute: if independent lifecycle, it's an entity.