1.

Introduction, Basic Concepts, and DML Commands	
Introduction to Databases	Installation of SQL Server Management Studio
	Introduction to basic database concepts.
	Advantage of DBMS
Keys, Operators & DML Commands	Introduction to RDBMS
	Creating Tables
	Relationship between tables.
	Primary keys, Foreign keys, Unique keys
	SQL operators (Arithmetic, Comparison, Logical)
	DML Commands
	CRUD operations

Session Minutes: Introduction to Databases, DML & CRUD using SSMS

Total Duration: Full Day
Audience: Beginners / Early-stage .NET & SQL learners
Tool: SQL Server Management Studio (SSMS)

| Introduction to Databases

Definition

A **Database** is an organized collection of structured data stored electronically, enabling efficient storage, retrieval, and manipulation.

Real-Life Example

- **E-commerce**: Users, Products, Orders
- **College ERP**: Students, Courses, Exams

Why Databases?

- Data consistency
- Faster access
- Security & backup
- Multi-user access

| DBMS & RDBMS Concepts

DBMS (Database Management System)

Software that manages databases
Examples: MySQL, SQL Server, Oracle

RDBMS (Relational DBMS)

- Data stored in **tables (rows & columns)**
- Relationships between tables
- Uses **SQL**

Advantages of RDBMS

- Data integrity
- Reduced redundancy
- Structured querying
- ACID compliance

| Installing & Exploring SSMS (UI Demo)

SSMS UI Walkthrough

- Object Explorer
- Databases
- Tables
- Query Editor
- Results & Messages window

Best Practices

- Always connect using **Windows Authentication** (local demo)
- Use **meaningful database & table names**

| Creating Database & Tables (Demo)

Demo: Create Database

```
CREATE DATABASE EcommerceDB;
GO
USE EcommerceDB;
```

Create Tables

```
CREATE TABLE Customers (
   CustomerId INT PRIMARY KEY IDENTITY,
   Name VARCHAR(100) NOT NULL,
   Email VARCHAR(100) UNIQUE,
   CreatedDate DATETIME DEFAULT GETDATE()
);

CREATE TABLE Orders (
   OrderId INT PRIMARY KEY IDENTITY,
   CustomerId INT,
   OrderAmount DECIMAL(10,2),
   OrderDate DATETIME DEFAULT GETDATE(),
   FOREIGN KEY (CustomerId) REFERENCES Customers(CustomerId)
);
```

60–80 mins | Keys & Relationships

Primary Key

- Uniquely identifies a row
- Cannot be NULL

Foreign Key

- Creates relationship between tables
- Enforces referential integrity

Unique Key

- Ensures uniqueness
- Allows NULL (once)

Best Practices

- Always define PK
- Use FK to avoid orphan records

| SQL Operators

Arithmetic Operators

+ - * /

Comparison Operators

=, !=, >, <, >=, <=

Logical Operators

AND, OR, NOT

Example

```
SELECT * FROM Orders
WHERE OrderAmount > 1000 AND CustomerId = 1;
```

100–140 mins | DML Commands (CRUD Operations)

INSERT (Create)

```
INSERT INTO Customers (Name, Email)
VALUES ('Amit Sharma', 'amit@gmail.com');
```

SELECT (Read)

```
SELECT * FROM Customers;
```

UPDATE

```
UPDATE Customers
SET Email = 'amit.sharma@gmail.com'
WHERE CustomerId = 1;
```

DELETE

```
DELETE FROM Customers
WHERE CustomerId = 1;
```

| Data Validation & Constraints

Constraints Used

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- DEFAULT

Why Constraints Matter

- Prevent invalid data
- Enforce business rules at DB level

155–170 mins | Real-World Mapping (E-commerce Case)

Business Entity	Table
Customer	Customers
Purchase	Orders
Relationship	Customer → Orders (1:M)

Scenario:
One customer can place multiple orders → enforced using **Foreign Key**

| Best Practices & Wrap-up

SQL Best Practices

- Use SELECT before UPDATE/DELETE

- Avoid SELECT * in production
- Use transactions for critical updates
- Name constraints clearly

Key Takeaways

- Databases store structured data
- SSMS simplifies DB management via UI
- CRUD is the foundation of all applications
- Constraints ensure data quality

Optional Assignment

1. Create Products table
2. Add Price, Stock, IsActive
3. Insert 5 products
4. Update stock for one product
5. Delete inactive products