Trainee Task: Schema-Based Organization in SQL Project

Title: Training & Certification Platform

Part 1: Explore the Concept

1. What is a schema in SQL Server?

A **schema** in SQL Server is a logical container that groups database objects like tables, views, procedures, etc., under a specific name. It's used to organize and manage access to these objects within a database.

2. How is a schema different from a database?

A **database** is the complete collection of data and objects, including all schemas.

A **schema** is a subset or namespace within a database used to group related objects. You can think of a schema as a folder inside a larger cabinet (the database).

3. List at least two benefits of using schemas in real applications:

 Security Management: You can assign permissions to a schema instead of individual tables, making access control simpler.

• **Organization**: Schemas help logically group related objects (e.g., Sales.Orders, HR.Employees) for better structure and clarity.

4. Can different schemas have the same table name?

Yes, **different schemas can have tables with the same name** because the fully qualified name includes the schema. For example, Sales. Customers and HR. Customers can both exist in the same database.

```
use SchemaDB
CREATE TABLE Employee (
EmployeeID INT PRIMARY KEY,
FullName VARCHAR(100),
Position VARCHAR(50),
HireDate DATE
);
CREATE TABLE Trainer (
TrainerID INT PRIMARY KEY,
FullName VARCHAR(100),
Specialization VARCHAR(100)
);
CREATE TABLE Attendance (
AttendanceID INT PRIMARY KEY,
EmployeeID INT,
Date DATE,
Status VARCHAR(20)
);
CREATE TABLE Course (
CourseID INT PRIMARY KEY,
CourseName VARCHAR(100),
```

```
DurationWeeks INT
);
CREATE TABLE Batch (
BatchID INT PRIMARY KEY,
CourseID INT,
StartDate DATE,
EndDate DATE
);
CREATE TABLE Exam (
ExamID INT PRIMARY KEY,
CourseID INT,
ExamDate DATE
);
CREATE TABLE Result (
ResultID INT PRIMARY KEY,
ExamID INT,
EmployeeID INT,
Score INT
);
CREATE TABLE Client (
ClientID INT PRIMARY KEY,
ClientName VARCHAR(100),
ContactPerson VARCHAR(100)
```

```
);
CREATE TABLE Contract (
ContractID INT PRIMARY KEY,
ClientID INT,
StartDate DATE,
EndDate DATE,
TotalValue DECIMAL(10,2)
);
CREATE TABLE Payment (
PaymentID INT PRIMARY KEY,
ContractID INT,
PaymentDate DATE,
Amount DECIMAL(10,2)
);
INSERT INTO dbo.Employee VALUES (1, 'Ali Al-Harthy', 'Admin', '2022-01-15');
INSERT INTO dbo.Trainer VALUES (101, 'Salim Al-Nabhani', 'Databases');
INSERT INTO dbo. Attendance VALUES (1001, 1, '2024-12-10', 'Present');
INSERT INTO dbo. Course VALUES (201, 'SQL Fundamentals', 4);
INSERT INTO dbo.Batch VALUES (301, 201, '2025-01-10', '2025-02-07');
INSERT INTO dbo.Exam VALUES (401, 201, '2025-02-10');
INSERT INTO dbo.Result VALUES (501, 401, 1, 87);
INSERT INTO dbo.Client VALUES (601, 'TechCorp LLC', 'Hassan Said');
INSERT INTO dbo.Contract VALUES (701, 601, '2025-01-01', '2025-12-31', 10000.00);
```

INSERT INTO dbo.Payment VALUES (801, 701, '2025-02-01', 2000.00);

Part 2: System Overview (Real-Life Use Case)

Department	Schema	Manages
Human Resources	HR	Employees, Trainers, Attendance
Academic Affairs	Academics	Courses, Batches, Exams, Results
Business/Sales	Sales	Clients, Contracts, Payments

Part 3: Create Schemas

CREATE SCHEMA HR;

CREATE SCHEMA Academics;

CREATE SCHEMA Sales;

Part 4 & 5: Transfer Tables to the Appropriate Schemas

-- HR tables

ALTER SCHEMA HR TRANSFER dbo.Employee;

ALTER SCHEMA HR TRANSFER dbo. Trainer;

ALTER SCHEMA HR TRANSFER dbo. Attendance;

-- Academics tables

ALTER SCHEMA Academics TRANSFER dbo.Course;

ALTER SCHEMA Academics TRANSFER dbo.Batch;

ALTER SCHEMA Academics TRANSFER dbo.Exam;

ALTER SCHEMA Academics TRANSFER dbo.Result;

-- Sales tables

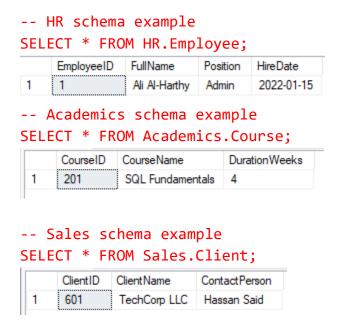
ALTER SCHEMA Sales TRANSFER dbo.Client;

ALTER SCHEMA Sales TRANSFER dbo.Contract;

ALTER SCHEMA Sales TRANSFER dbo.Payment;

Part 6: Validation Practice

1. SELECT queries using schema-qualified names:



2. Query using WHERE clause for filtering:

```
-- Example: Employees hired after 2023-01-01
SELECT * FROM HR.Employee
WHERE HireDate > '2023-01-01';

-- Example: Courses with duration more than 30 days
SELECT * FROM Academics.Course
WHERE DurationWeeks > 30;
```

3. OPTIONAL: Create and test a new table under a schema:

```
-- Create table under Sales schema
CREATE TABLE Sales.Feedback (
```

```
FeedbackID INT PRIMARY KEY,
   ClientID INT,
   Comments TEXT,
   Rating INT,
   FOREIGN KEY (ClientID) REFERENCES Sales.Client(ClientID)
);
-- Insert sample data
INSERT INTO Sales.Feedback VALUES (1, 101, 'Great service', 5);
-- Query inserted data
SELECT * FROM Sales.Feedback;
```

Part 7: Reflection

1. What are the advantages of separating tables into schemas in a large system? Separating tables into schemas improves **organization**, making it easier to manage and understand the structure of large databases. It also helps in **modularizing responsibilities** by department or function.

2. How can schemas support security and access control?

Schemas allow database administrators to **grant or restrict access** to specific groups of users based on roles or departments. For example, HR staff can be granted access only to the HR schema, ensuring **data isolation and confidentiality**.

3. Suggest a new department that could be added to this system and which tables it would manage:

A new department could be "IT Support", managing tables like:

- Tickets
- SupportAgents
- SystemLogs
- HardwareInventory