Quiz 1 sec E Solution

## -- Task 1: Employee Management Database Operations

**-- 1. Table Creation with Constraints**

CREATE TABLE EmployeeRecords (

EmployeeID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Department VARCHAR(100),

Salary DECIMAL(10,2) CHECK (Salary >= 0)

);

**-- 2. Data Insertion**

INSERT INTO EmployeeRecords (FirstName, LastName, Department, Salary)

VALUES

('John', 'Doe', 'Finance', 50000.00),

('Jane', 'Smith', 'HR', 60000.00),

('Alice', 'Johnson', 'IT', 70000.00),

('YourFirstName', 'YourLastName', 'YourDepartment', YourSalary);

**-- 3. Adding and Deleting Columns with Constraints**

ALTER TABLE EmployeeRecords ADD HireDate DATE;

ALTER TABLE EmployeeRecords DROP COLUMN HireDate;

**-- 4. Employees with Longest Last Names**

SELECT FirstName, LastName

FROM EmployeeRecords

WHERE LEN(LastName) = (SELECT MAX(LEN(LastName)) FROM EmployeeRecords);

**-- 5. Average Salary by Department**

SELECT Department, AVG(Salary) AS AverageSalary

FROM EmployeeRecords

GROUP BY Department;

**-- 6. Salary Increment**

UPDATE EmployeeRecords

SET Salary = Salary \* 1.1

WHERE Department = 'IT';

**-- 7. Employee Count by Department**

SELECT Department, COUNT(\*) AS EmployeeCount

FROM EmployeeRecords

GROUP BY Department;

## -- Task: Creation of Database for a Movie Library

-- Create a new database named "MovieLibraryDB".

CREATE DATABASE IF NOT EXISTS MovieLibraryDB;

-- Use the MovieLibraryDB database.

USE MovieLibraryDB;

-- Create 3 tables:

-- Movies table

CREATE TABLE IF NOT EXISTS Movies (

movie\_id INT PRIMARY KEY,

title VARCHAR(255),

director VARCHAR(255),

release\_year INT,

genre VARCHAR(255)

);

-- Actors table

CREATE TABLE IF NOT EXISTS Actors (

actor\_id INT PRIMARY KEY,

actor\_name VARCHAR(255)

);

-- Movie\_Cast table

CREATE TABLE IF NOT EXISTS Movie\_Cast (

movie\_cast\_id INT PRIMARY KEY,

movie\_id INT,

actor\_id INT,

role VARCHAR(255),

FOREIGN KEY (movie\_id) REFERENCES Movies(movie\_id),

FOREIGN KEY (actor\_id) REFERENCES Actors(actor\_id)

);

-- Insert data into these tables:

-- Movies Table

INSERT INTO Movies (movie\_id, title, director, release\_year, genre)

VALUES

(1, 'Inception', 'Christopher Nolan', 2010, 'Sci-Fi'),

(2, 'The Shawshank Redemption', 'Frank Darabont', 1994, 'Drama'),

(3, 'Pulp Fiction', 'Quentin Tarantino', 1994, 'Crime');

-- Actors Table

INSERT INTO Actors (actor\_id, actor\_name)

VALUES

(1, 'Leonardo DiCaprio'),

(2, 'Morgan Freeman'),

(3, 'Tim Robbins'),

(4, 'John Travolta');

-- Movie\_Cast Table

INSERT INTO Movie\_Cast (movie\_cast\_id, movie\_id, actor\_id, role)

VALUES

(1, 1, 1, 'Cobb'),

(2, 1, 2, 'Red'),

(3, 2, 2, 'Ellis Boyd "Red" Redding'),

(4, 2, 3, 'Andy Dufresne'),

(5, 3, 1, 'Vincent Vega'),

(6, 3, 4, 'Jules Winnfield');

-- Apply the following queries:

**-- Query #1: Retrieve all records from the "Movies" table where the genre is 'Drama'.**

SELECT \* FROM Movies WHERE genre = 'Drama';

**-- Query #2: Retrieve all records from the "Movie\_Cast" table where the role is 'Cobb'.**

SELECT \* FROM Movie\_Cast WHERE role = 'Cobb';

**-- Query #3: Retrieve the actor names and the movies they have acted in from the "Actors" and "Movie\_Cast" tables using joins.**

SELECT Actors.actor\_name, Movies.title

FROM Actors

JOIN Movie\_Cast ON Actors.actor\_id = Movie\_Cast.actor\_id

JOIN Movies ON Movie\_Cast.movie\_id = Movies.movie\_id;

**-- Query #4: Retrieve the average release year of movies.**

SELECT AVG(release\_year) AS average\_release\_year FROM Movies;

**-- Query #5: Retrieve the movie titles and their corresponding actors from the "Movies", "Actors", and "Movie\_Cast" tables using joins.**

SELECT Movies.title, Actors.actor\_name

FROM Movies

JOIN Movie\_Cast ON Movies.movie\_id = Movie\_Cast.movie\_id

JOIN Actors ON Movie\_Cast.actor\_id = Actors.actor\_id;

**-- Query #6: Retrieve all actors who have not been cast in any movie yet, including details from the "Actors" and "Movie\_Cast" tables using joins.**

SELECT Actors.\*

FROM Actors

LEFT JOIN Movie\_Cast ON Actors.actor\_id = Movie\_Cast.actor\_id

WHERE Movie\_Cast.actor\_id IS NULL;

**-- Query #7: Retrieve the top 3 actors who have appeared in the most number of movies, along with the count of movies they have appeared in.**

SELECT TOP 3 Actors.actor\_name, COUNT(Movie\_Cast.movie\_id) AS movie\_count

FROM Actors

JOIN Movie\_Cast ON Actors.actor\_id = Movie\_Cast.actor\_id

GROUP BY Actors.actor\_name

ORDER BY movie\_count DESC;

**-- Query #8: Retrieve the movie titles along with the number of actors who have appeared in each movie, sorted in descending order of the count of actors.**

SELECT Movies.title, COUNT(Movie\_Cast.actor\_id) AS actor\_count

FROM Movies

JOIN Movie\_Cast ON Movies.movie\_id = Movie\_Cast.movie\_id

GROUP BY Movies.title

ORDER BY actor\_count DESC;