LAKSHMI KUMAR
002887298

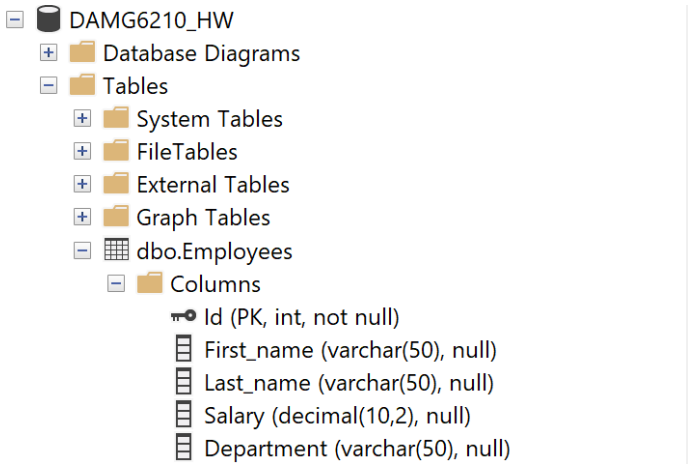
DATA MANAGEMENT AND DATABASE DESIGN
HOMEWORK: WEEK- 8

Answer the following questions on the provided schema for the table and sample record

INSERT statements:
Table definition:

```
CREATE TABLE Employees (
   Id INT PRIMARY KEY,
   First_name VARCHAR (50),
   Last_name VARCHAR (50),
   Salary DECIMAL (10, 2),
   Department VARCHAR (50)
);
```

Sol:



SQLQuery1.sql - L...LKUMAR\laksh (71))*

```sql
CREATE DATABASE DAMG6210_HW;
GO
USE DAMG6210_HW
GO
CREATE TABLE Employees (

Id INT PRIMARY KEY,
First_name VARCHAR(50),
Last_name VARCHAR(50),
Salary DECIMAL(10, 2),
Department VARCHAR(50)
);
```

LAKSHMI KUMAR
002887298

Sample records:
INSERT INTO employees (Id, First_name, Last_name, Salary, Department)
VALUES
(1, 'John', 'Doe', 50000, 'HR'),
(2, 'Jane', 'Smith', 60000, 'Finance'),
(3, 'Alice', 'Johnson', 70000, 'IT'),
(4, 'Bob', 'Williams', 80000, 'IT'),
(5, 'Charlie', 'Brown', 40000, 'HR');

Sol:

SQLQuery1.sql - L...LKUMAR\laksh (71))*

```sql
INSERT INTO Employees (ID, First_name, Last_name, Salary, Department)
VALUES (1, 'John', 'Doe', 50000, 'HR'), (2, 'Jane', 'Smith', 60000, 'Finance'), (3, 'Alice', 'Johnson', 70000, 'IT'),
(4, 'Bob', 'Williams', 80000, 'IT'),(5, 'Charlie', 'Brown', 40000, 'HR');

SELECT * FROM Employees;
```

110 %

Results | Messages

	Id	First_name	Last_name	Salary	Department
1	1	John	Doe	50000.00	HR
2	2	Jane	Smith	60000.00	Finance
3	3	Alice	Johnson	70000.00	IT
4	4	Bob	Williams	80000.00	IT
5	5	Charlie	Brown	40000.00	HR

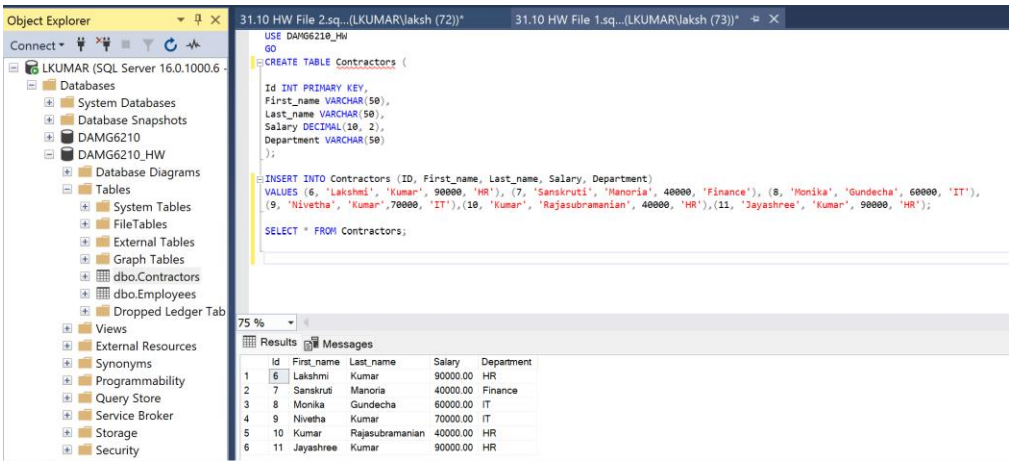
LAKSHMI KUMAR
002887298

QUESTIONS:

1.

Sol:

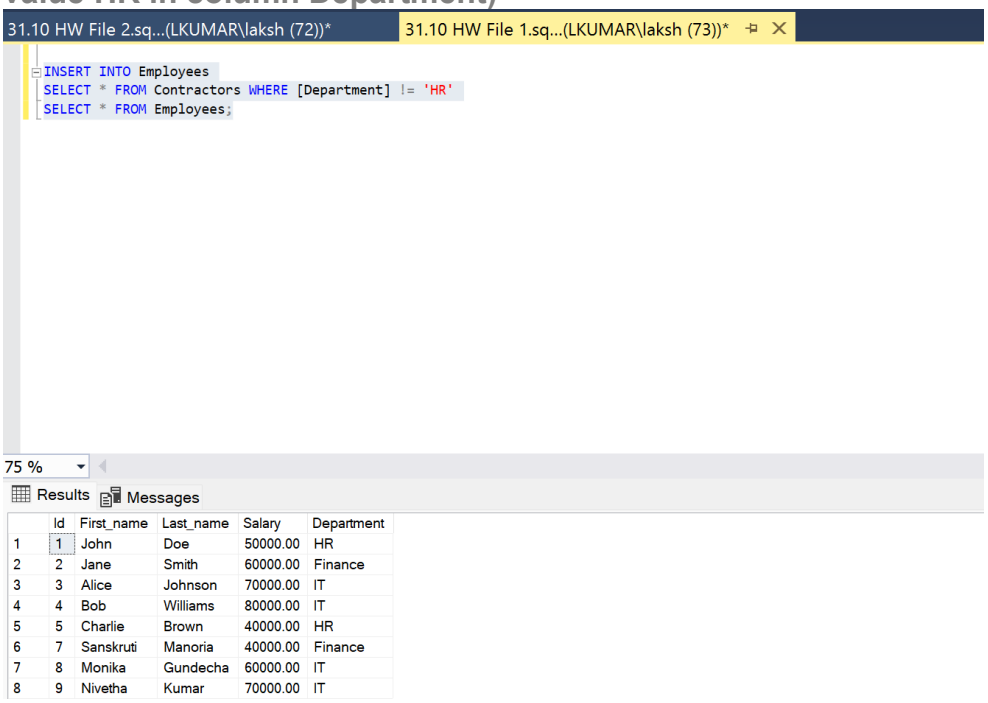
➔ Create a new table “Contractors” and add values into it.



➔ Query to insert data into the `Employees` table from another table `Contractors` with the same structure, excluding those in the 'HR' department.

```
INSERT INTO Employees
SELECT * FROM Contractors WHERE [Department]! = 'HR'
SELECT * FROM Employees;
```

(Only records with Id 7, 8, 9 have been inserted from Contractors to Employees. Records with Id 6, 10, 11 have not been inserted since those records have the value HR in column Department)



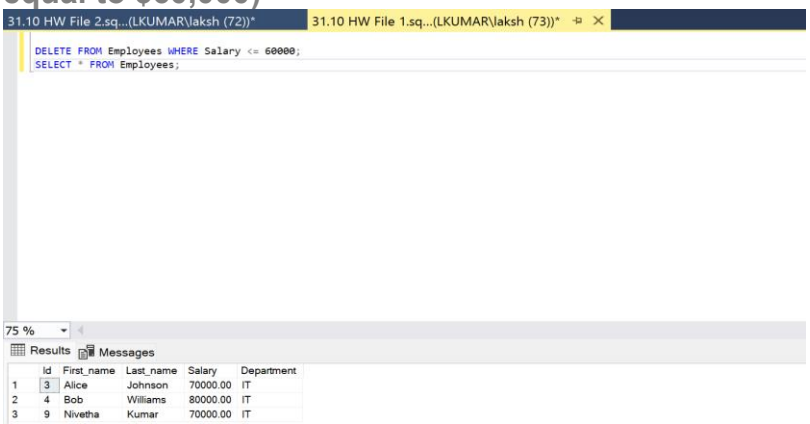
LAKSHMI KUMAR
002887298

2.
Sol:

➔ Query to delete employees from the `Employees` table which have a salary less or equal to $ 60000.

```
DELETE FROM Employees WHERE Salary <= 60000;
SELECT * FROM Employees;
```

(Only records with Id 3, 4, 9 will be displayed from Employees. Records with Id 1,2,5,6,7 have not been displayed since those records have salary less than or equal to $60,000)

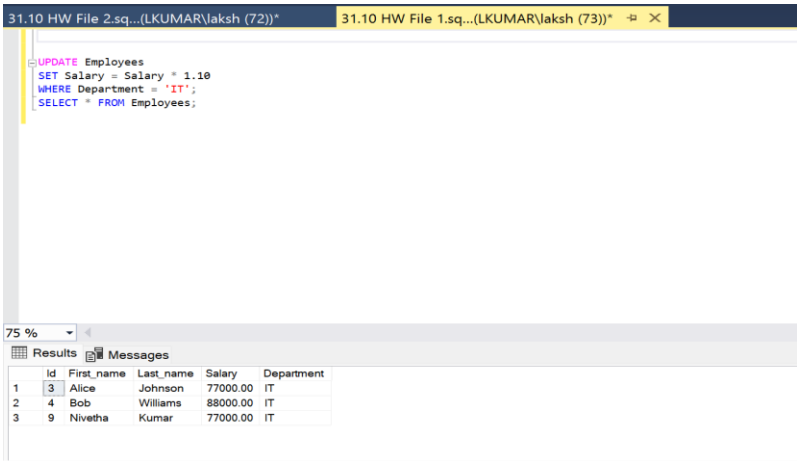


	Id	First_name	Last_name	Salary	Department
1	3	Alice	Johnson	70000.00	IT
2	4	Bob	Williams	80000.00	IT
3	9	Nivetha	Kumar	70000.00	IT

3.
Sol:

➔ Query to update the salary of employees in the 'IT' department by increasing it by 10%

```
UPDATE Employees
SET Salary = Salary * 1.10
WHERE Department = 'IT';
SELECT * FROM Employees;
```



	Id	First_name	Last_name	Salary	Department
1	3	Alice	Johnson	77000.00	IT
2	4	Bob	Williams	88000.00	IT
3	9	Nivetha	Kumar	77000.00	IT

Adding more values for Querying purpose:

31.10 HW File 2.sq...(LKUMAR\laksh (72))* | 31.10 HW File 1.sq...(LKUMAR\laksh (73))*

```sql
INSERT INTO Employees (ID, First_name, Last_name, Salary, Department)
VALUES (6, 'Lakshmi', 'Kumar', 90000, 'HR'), (7, 'Sanskruti', 'Manoria', 40000, 'Finance'), (8, 'Monika', 'Gundecha', 60000, 'IT'),
(10, 'Kumar', 'Rajasubramanian', 40000, 'HR'),(11, 'Jayashree', 'Kumar', 90000, 'HR'), (5, 'Johnson', 'Mathew', 20000, 'HR');
SELECT * FROM Employees;
```

75 %

Results | Messages

	Id	First_name	Last_name	Salary	Department
1	3	Alice	Johnson	77000.00	IT
2	4	Bob	Williams	88000.00	IT
3	5	Johnson	Mathew	20000.00	HR
4	6	Lakshmi	Kumar	90000.00	HR
5	7	Sanskruti	Manoria	40000.00	Finance
6	8	Monika	Gundecha	60000.00	IT
7	9	Nivetha	Kumar	77000.00	IT
8	10	Kumar	Rajasubramanian	40000.00	HR
9	11	Jayashree	Kumar	90000.00	HR

4.

Sol:

➔ Query to display all columns with the top 3 highest salaries from the `Employees` table

```sql
SELECT TOP 3 Salary
FROM Employees
ORDER BY Salary DESC;
```

31.10 HW File 2.sq...(LKUMAR\laksh (72))* | 31.10 HW File 1.sq...(LKUMAR\laksh (73))*

```sql
SELECT TOP 3 Salary
FROM Employees
ORDER BY Salary DESC;
```

75 %

Results | Messages

	Salary
1	90000.00
2	90000.00
3	88000.00

LAKSHMI KUMAR
002887298

➔ Query to display all columns with the top 3 highest salaries from the `Employees` table

```
SELECT TOP 3 *
FROM Employees
ORDER BY Salary DESC;
```

31.10 HW File 2.sq...(LKUMAR\laksh (72))* | 31.10 HW File 1.sq...(LKUMAR\laksh (73))*

```
SELECT TOP 3 *
FROM Employees
ORDER BY Salary DESC;
```

75 %

Results | Messages

	Id	First_name	Last_name	Salary	Department
1	6	Lakshmi	Kumar	90000.00	HR
2	11	Jayashree	Kumar	90000.00	HR
3	4	Bob	Williams	88000.00	IT

5.
Sol:
Query to select employees whose first name starts with 'J' and ends with 'n'

```
SELECT * FROM Employees
WHERE First_name LIKE 'J%n'
```

31.10 HW File 2.sq...(LKUMAR\laksh (72))* | 31.10 HW File 1.sq...(LKUMAR\laksh (73))*

```
SELECT * FROM Employees
WHERE First_name LIKE 'J%n';
```

75 %

Results | Messages

	Id	First_name	Last_name	Salary	Department
1	5	Johnson	Mathew	20000.00	HR

LAKSHMI KUMAR
002887298

6.
Sol:
Query to select employees who are either in the 'HR' department or have a salary greater than 60000, but not both.

```
SELECT * FROM Employees
WHERE (Salary>60000 OR Department='HR') AND NOT (Department = 'HR' AND Salary > 60000);
```

31.10 HW File 1.sq...(LKUMAR\laksh (73))* | 31.10 HW File 2.sq...(LKUMAR\laksh (72))*

```
SELECT * FROM Employees
WHERE (Salary>60000 OR Department='HR') AND NOT (Department = 'HR' AND Salary > 60000);
```

83 %

Results | Messages

	Id	First_name	Last_name	Salary	Department
1	3	Alice	Johnson	77000.00	IT
2	4	Bob	Williams	88000.00	IT
3	5	Johnson	Mathew	20000.00	HR
4	9	Nivetha	Kumar	77000.00	IT
5	10	Kumar	Rajasubramanian	40000.00	HR

7.
Sol:
Query to you get the total salary for each department using the `Employees` table?

```
SELECT Department, SUM(Salary) AS TotalSalary
FROM Employees
GROUP BY Department;
```

31.10 HW File 1.sq...(LKUMAR\laksh (73))* | 31.10 HW File 2.sq...(LKUMAR\laksh (72))*

```
SELECT Department, SUM(Salary) AS TotalSalary
FROM Employees
GROUP BY Department;
```

83 %

Results | Messages

	Department	TotalSalary
1	Finance	40000.00
2	HR	240000.00
3	IT	302000.00

LAKSHMI KUMAR
002887298

8.
Sol:
Query to select employees from the `Employees` table, sorted by department in ascending order, and then by salary in descending order within each department.

```
SELECT * FROM Employees
ORDER BY Department ASC, Salary DESC;
```

31.10 HW File 1.sq...(LKUMAR\laksh (73))* | 31.10 HW File 2.sq...(LKUMAR\laksh (72))*

```
SELECT * FROM Employees
ORDER BY Department ASC, Salary DESC;
```

83 %

Results | Messages

	Id	First_name	Last_name	Salary	Department
1	7	Sanskruti	Manoria	40000.00	Finance
2	6	Lakshmi	Kumar	90000.00	HR
3	11	Jayashree	Kumar	90000.00	HR
4	10	Kumar	Rajasubramanian	40000.00	HR
5	5	Johnson	Mathew	20000.00	HR
6	4	Bob	Williams	88000.00	IT
7	3	Alice	Johnson	77000.00	IT
8	9	Nivetha	Kumar	77000.00	IT
9	8	Monika	Gundecha	60000.00	IT

9.
Sol:
Query to create a view that shows the average salary for each department from the employees` table?

```
CREATE VIEW DepartmentAverageSalaryViewAS
SELECT Department, AVG(Salary) AS AverageSalary
FROM Employees
GROUP BY Department;
SELECT * FROM DepartmentAverageSalaryView;
```

31.10 HW File 1.sq...(LKUMAR\laksh (73))* | 31.10 HW File 2.sq...(LKUMAR\laksh (72))*

```
CREATE VIEW DepartmentAverageSalaryView AS
SELECT Department, AVG(Salary) AS AverageSalary
FROM Employees
GROUP BY Department;
SELECT * FROM DepartmentAverageSalaryView;
```

83 %

Results | Messages

	Department	AverageSalary
1	Finance	40000.000000
2	HR	60000.000000
3	IT	75500.000000

LAKSHMI KUMAR
002887298

10. What is a strategy to optimize a query that frequently retrieves department and average salary columns data from the `employees` table?

Sol:

Strategies to optimize a query that frequently retrieves department and average salary columns data from the Employees table.

- ➔ Maintenance: Ensure indexes are maintained regularly, indexes are not fragmented and also that queries run in a systematic manner.
- ➔ Caching: Implement a cache to store the department and average salary columns data in order to reduce load on the database.
- ➔ Usage of appropriate hardware: Make sure database server has the right type of hardware (i.e.) CPU, Memory, Storage etc.
- ➔ Denormalizing the data: Incase the data wouldn't have to be changed frequently, denormalize the data and store average salary in a separate table directly. In this way, frequent calculations can be eliminated.
- ➔ Indexing: Look out if the table has been indexed properly. Make sure to put an index on the column "Department" in order to speed up grouping and retrieval of data.
- ➔ Partitioning: If the table is extremely large, partition the table. This helps deal with poor performance in case of frequent querying or retrieval.