

ASSIGNMENT 10

```
use DAMG6210
```

```
go
```

```
CREATE TABLE employees (  
    id INT PRIMARY KEY identity(1,1),  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    salary DECIMAL(10, 2),  
    department VARCHAR(50)  
);
```

```
INSERT INTO employees (first_name, last_name, salary, department)  
VALUES  
    ('John', 'Doe', 50000, 'HR'),  
    ('Jane', 'Smith', 60000, 'Finance'),  
    ('Alice', 'Johnson', 70000, 'IT'),  
    ('Bob', 'Williams', 80000, 'IT'),  
    ('Charlie', 'Brown', 40000, 'HR');
```

```
[22] 1 select * from employees;
```

SQL

(5 rows affected)

Total execution time: 00:00:00.021



	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

Results grid

```
CREATE TABLE [dbo].[contractors](  
    [id] [int] IDENTITY(1,1) NOT NULL,  
    [first_name] [varchar](50) NULL,  
    [last_name] [varchar](50) NULL,  
    [salary] [decimal](10, 2) NULL,  
    [department] [varchar](50) NULL  
) ON [PRIMARY]
```

```
GO
ALTER TABLE [dbo].[contractors] ADD PRIMARY KEY CLUSTERED
(
    [id] ASC
)
GO
```

```
[26] 1  select * from contractors;
      2
```

SQL

(0 rows affected)

Total execution time: 00:00:00.016



id	first_name	last_name	salary	department
----	------------	-----------	--------	------------

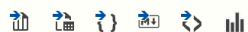
```
INSERT INTO contractors (first_name, last_name, salary, department)
SELECT first_name, last_name, salary, department FROM employees;
```

```
[30] 1  select * from contractors ;
```

SQL

(5 rows affected)

Total execution time: 00:00:00.014



	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	William	80000.00	IT
5	5	Charlie	Brown	40000.00	HR

```
1. INSERT INTO employees (first_name, last_name, salary, department)
SELECT first_name, last_name, salary, department FROM contractors where department !=
'HR';
```

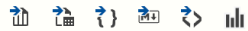
1. Write a query to insert data into the `employees` table from another table `contractors` with the same structure, excluding those in the 'HR' department.

```
[31] 1 SELECT * FROM contractors where department != 'HR';
```

SQL

(3 rows affected)

Total execution time: 00:00:00.048



	id	first_name	last_name	salary	department
1	2	Jane	Smith	60000.00	Finance
2	3	Alice	Johnson	70000.00	IT
3	4	Bob	Williams	80000.00	IT

```
[32] 1 INSERT INTO employees (first_name, last_name, salary, department)
2 SELECT first_name, last_name, salary, department FROM contractors where department != 'HR';
```



SQL

(3 rows affected)

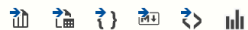
Total execution time: 00:00:00.049

```
[33] 1 select * from employees;
```

SQL

(8 rows affected)

Total execution time: 00:00:00.010



	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
6	6	Jane	Smith	60000.00	Finance
7	7	Alice	Johnson	70000.00	IT
8	8	Bob	Williams	80000.00	IT

- How can you delete employees from the **employees** table who have a salary less or equal to \$60,000?

`delete from employees where salary <= 60000;`

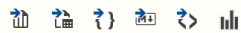
1. How can you delete employees from the **employees** table who have a salary less or equal to \$60,000?

```
[34] 1 select * from employees where salary <= 60000;
```

SQL

(4 rows affected)

Total execution time: 00:00:00.096



	id	first_name	last_name	salary	department
1	1	John	Doe	50000.00	HR
2	2	Jane	Smith	60000.00	Finance
3	5	Charlie	Brown	40000.00	HR
4	6	Jane	Smith	60000.00	Finance

```
[35] 1 delete from employees where salary <= 60000;
```

SQL

(4 rows affected)

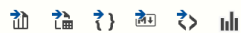
Total execution time: 00:00:00.052

```
1 select * from employees;
```

SQL

(4 rows affected)

Total execution time: 00:00:00.049



	id	first_name	last_name	salary	department
1	3	Alice	Johnson	70000.00	IT
2	4	Bob	Williams	80000.00	IT
3	7	Alice	Johnson	70000.00	IT
4	8	Bob	Williams	80000.00	IT

- ☐ How can you update the salary of employees in the 'IT' department by increasing it by 10%?

```
UPDATE employees
SET salary = salary * 1.10
WHERE department = 'IT';
```

1. How can you update the salary of employees in the 'IT' department by increasing it by 10%?

```
[37] 1 UPDATE employees
      2 SET salary = salary * 1.10
      3 WHERE department = 'IT';
```

SQL

(4 rows affected)

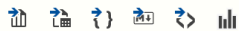
Total execution time: 00:00:00.090

```
1 select * from employees;
```

SQL

(4 rows affected)

Total execution time: 00:00:00.036



	id	first_name	last_name	salary	department
1	3	Alice	Johnson	77000.00	IT
2	4	Bob	Williams	88000.00	IT
3	7	Alice	Johnson	77000.00	IT
4	8	Bob	Williams	88000.00	IT

- ☐ Write a query to select the top 3 highest salaries from the `employees` table.

```
SELECT TOP 3 *
FROM employees
ORDER BY salary DESC;
```

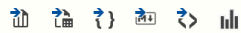
Write a query to select the top 3 highest salaries from the `employees` table.

```
[39]  1  SELECT *
      2  FROM employees
      3  ORDER BY salary DESC;
```

SQL

(4 rows affected)

Total execution time: 00:00:00.033



	id	first_name	last_name	salary	department
1	4	Bob	Williams	88000.00	IT
2	8	Bob	Williams	88000.00	IT
3	3	Alice	Johnson	77000.00	IT
4	7	Alice	Johnson	77000.00	IT

```
[40]  1  SELECT TOP 3 *
      2  FROM employees
      3  ORDER BY salary DESC;
```

SQL

(3 rows affected)

Total execution time: 00:00:00.029



	id	first_name	last_name	salary	department
1	4	Bob	Williams	88000.00	IT
2	8	Bob	Williams	88000.00	IT
3	7	Alice	Johnson	77000.00	IT

☐ How can you select employees whose first name starts with 'J' and ends with 'n'?

```
SELECT *
FROM employees
WHERE first_name LIKE 'J%n';
```

1. How can you select employees whose first name starts with 'J' and ends with 'n'?

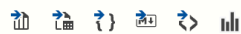
```
1 SELECT *
2 FROM employees
3 WHERE first_name LIKE 'J%n';
```



SQL

(0 rows affected)

Total execution time: 00:00:00.068



id	first_name	last_name	salary	department
----	------------	-----------	--------	------------

and if we want last name to start with 'J' and end with 'n'

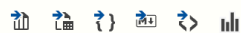
```
[42] 1 SELECT *
      2 FROM employees
      3 WHERE last_name LIKE 'J%n';
```



SQL

(2 rows affected)

Total execution time: 00:00:00.045



	id	first_name	last_name	salary	department
1	3	Alice	Johnson	77000.00	IT
2	7	Alice	Johnson	77000.00	IT

- Write a 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 (department = 'HR' OR salary > 60000)
AND NOT (department = 'HR' AND salary > 60000);
```

1. Write a query to select employees who are either in the 'HR' department or have a salary greater than 60000, but not both.

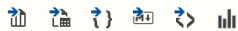
```
[43] 1 SELECT *
      2 FROM employees
      3 WHERE (department = 'HR' OR salary > 60000);
      4
      5
```



SQL

(4 rows affected)

Total execution time: 00:00:00.049



	id	first_name	last_name	salary	department
1	3	Alice	Johnson	77000.00	IT
2	4	Bob	Williams	88000.00	IT
3	7	Alice	Johnson	77000.00	IT
4	8	Bob	Williams	88000.00	IT

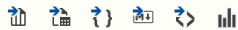
```
[45] 1 SELECT *
      2 FROM employees
      3 WHERE department = 'HR' AND salary > 60000;
```



SQL

(0 rows affected)

Total execution time: 00:00:00.057



id	first_name	last_name	salary	department
----	------------	-----------	--------	------------


```

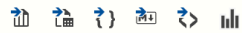
1 SELECT *
2 FROM employees
3 WHERE (department = 'HR' OR salary > 60000)
4 AND NOT (department = 'HR' AND salary > 60000);
5

```

SQL

(4 rows affected)

Total execution time: 00:00:00.032



	id	first_name	last_name	salary	department
1	3	Alice	Johnson	77000.00	IT
2	4	Bob	Williams	88000.00	IT
3	7	Alice	Johnson	77000.00	IT
4	8	Bob	Williams	88000.00	IT

□ How can you get the total salary for each department using the `employees` table?

```

SELECT department, SUM(salary) AS total_salary
FROM employees
GROUP BY department;

```

1. How can you get the total salary for each department using the `employees` table?

```

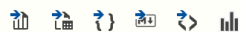
[47] 1 SELECT department, SUM(salary) AS total_salary
      2 FROM employees
      3 GROUP BY department;
      4

```

SQL

(1 row affected)

Total execution time: 00:00:00.046



	department	total_salary
1	IT	330000.00

Since, employees table only have one department, we can try this query on contractors table as well to better understand the output.

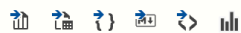
```
[48]  1  SELECT department, SUM(salary) AS total_salary
      2  FROM contractors
      3  GROUP BY department;
      4
```



SQL

(3 rows affected)

Total execution time: 00:00:00.031



	department ▾	total_salary ▾
1	Finance	60000.00
2	HR	90000.00
3	IT	150000.00

- ☐ Write a 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;
```

1. Write a query to select employees from the `employees` table, sorted by department in ascending order, and then by salary in descending order within each department.

```
[49]  1  SELECT *
      2  FROM employees
      3  ORDER BY department ASC, salary DESC;
      4
```



SQL

(4 rows affected)

Total execution time: 00:00:00.016



	id	first_name	last_name	salary	department
1	4	Bob	Williams	88000.00	IT
2	8	Bob	Williams	88000.00	IT
3	3	Alice	Johnson	77000.00	IT
4	7	Alice	Johnson	77000.00	IT

Again, as we have only one department in employees table, we can try this query in contractors table as well.

```
[50]  1  SELECT *
      2  FROM contractors
      3  ORDER BY department ASC, salary DESC;
      4
```



SQL

(5 rows affected)

Total execution time: 00:00:00.025



	id	first_name	last_name	salary	department
1	2	Jane	Smith	60000.00	Finance
2	1	John	Doe	50000.00	HR
3	5	Charlie	Brown	40000.00	HR
4	4	Bob	Williams	80000.00	IT
5	3	Alice	Johnson	70000.00	IT

- ☐ How can you create a view that shows the average salary for each department from the `employees` table?

```
SELECT department, AVG(salary) AS avg_salary
FROM employees
GROUP BY department;
```

```
CREATE VIEW avg_salary_of_department AS
SELECT department, AVG(salary) AS avg_salary
FROM employees
GROUP BY department;
```

1. How can you create a view that shows the average salary for each department from the `employees` table?

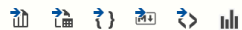
```
[51] 1 SELECT department, AVG(salary) AS avg_salary
      2 FROM employees
      3 GROUP BY department;
```



SQL

(1 row affected)

Total execution time: 00:00:00.059



	department	avg_salary
1	IT	82500.000000

```
[52] 1 CREATE VIEW avg_salary_of_department AS
      2 SELECT department, AVG(salary) AS avg_salary
      3 FROM employees
      4 GROUP BY department;
```



SQL

Commands completed successfully.

Total execution time: 00:00:00.074

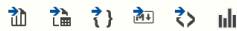
Again, I am trying this on contractors table as it has multiple departments.

```
[53] 1 SELECT department, AVG(salary) AS avg_salary
      2 FROM contractors
      3 GROUP BY department;
```

SQL

(3 rows affected)

Total execution time: 00:00:00.034



	department	avg_salary
1	Finance	60000.000000
2	HR	45000.000000
3	IT	75000.000000

```
[54] 1 CREATE VIEW avg_salary_per_department_contractors AS
      2 SELECT department, AVG(salary) AS avg_salary
      3 FROM employees
      4 GROUP BY department;
```

SQL

Commands completed successfully.

Total execution time: 00:00:00.026

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

To optimize this query, we can create an index on the department column in the employees table. Indexing this column will speed up the retrieval process, especially when selecting data based on the department column. Additionally, caching the view `avg_salary_of_department` can also improve performance, as it pre-calculates average salaries and reduces the need for repeated calculations.

Create an index on the department column:

```
[ 3]  1  CREATE INDEX index_department ON employees(department);
```

SQL

Commands completed successfully.

Total execution time: 00:00:00.161

Create an index on the salary column:

```
[ 4]  1  CREATE INDEX index_salary ON employees(salary);  
      2
```



SQL

Commands completed successfully.

Total execution time: 00:00:00.050

Utilize a index view for pre-computed salaries:

```
[ 6]  1  CREATE VIEW iv_avg_salary_per_department  
      2  AS  
      3  SELECT department, AVG(salary) AS avg_salary  
      4  FROM employees  
      5  GROUP BY department;  
      6
```



SQL

Commands completed successfully.

Total execution time: 00:00:00.050

Regularly analyze query performance:

```
[ 7] 1  SET SHOWPLAN_TEXT ON;
      2  GO
      3
      4  SELECT department, AVG(salary) AS avg_salary
      5  FROM employees
      6  GROUP BY department;
      7
```



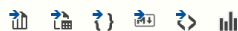
SQL

Commands completed successfully.

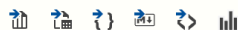
(1 row affected)

(5 rows affected)

Total execution time: 00:00:00.067



	StmtText	▼
1	SELECT department, AVG(salary) AS avg_salary FROM ...	



	StmtText	▼
1	--Compute Scalar(DEFINE:([Expr1002]=CASE WHEN [E...	
2	--Stream Aggregate(GROUP BY:([DAMG6210].[db...	
3	--Nested Loops(Inner Join, OUTER REFER...	
4	--Index Scan(OBJECT:([DAMG6210].[...	
5	--Clustered Index Seek(OBJECT:([D...	