

Table Creation and Sample Data

1. Create `departments` table:

```
CREATE TABLE departments (  
    dept_id INT PRIMARY KEY,  
    dept_name VARCHAR(100)  
);
```

2. Insert sample data into `departments` table:

```
INSERT INTO departments (dept_id, dept_name) VALUES  
(101, 'Human Resources'),  
(102, 'Finance'),  
(103, 'Engineering'),  
(104, 'Marketing'),  
(105, 'IT Support'),  
(106, 'Logistics'),  
(107, 'Sales'),  
(108, 'Legal'),  
(109, 'Customer Service'),  
(110, 'Research and Development'),  
(111, 'Procurement'),  
(112, 'Security'),  
(113, 'Public Relations'),  
(114, 'Quality Assurance'),  
(115, 'Training');
```

3. 2. Create `employees` table:

```
CREATE TABLE employees (  
    emp_id INT PRIMARY KEY,  
    emp_name VARCHAR(100),  
    dept_id INT,  
    salary DECIMAL(10,2),  
    hire_date DATE  
);
```

4. Insert sample data into `employees` table:

```

INSERT INTO employees (emp_id, emp_name, dept_id, salary, hire_date) VALUES
(1, 'Anand Krishnan', 101, 52000.00, '2020-04-12'),
(2, 'Lekshmi Nair', 102, 60000.00, '2019-08-20'),
(3, 'Rajeev Menon', 103, 75000.00, '2021-03-15'),
(4, 'Sneha Suresh', 104, 48000.00, '2022-11-10'),
(5, 'Vishnu Mohan', 105, 51000.00, '2023-06-01'),
(6, 'Divya Prasad', 101, 46000.00, '2018-12-25'),
(7, 'Hari Varma', 106, 58000.00, '2020-07-14'),
(8, 'Meera Rajan', 107, 54000.00, '2021-09-19'),
(9, 'Arjun Das', 108, 50000.00, '2019-02-05'),
(10, 'Neha Shaji', 102, 61000.00, '2022-03-27'),
(11, 'Sudeep Kumar', 109, 49000.00, '2023-01-09'),
(12, 'Anjali Menon', 110, 47000.00, '2020-10-31'),
(13, 'Jithin Joseph', 111, 53000.00, '2021-06-20'),
(14, 'Revathi Santhosh', 112, 56000.00, '2022-08-15'),
(15, 'Manu Thomas', 103, 62000.00, '2023-04-03');

```

Part A:

1. Display all data from the employees table

SELECT * FROM employees;

The screenshot shows the SQL Developer interface with the following components:

- Navigator:** Shows the database structure with 'MANAGEMENT' and 'INSTANCE' folders.
- SQL Editor:** Contains the SQL script:


```

INSERT INTO employees (emp_id, emp_name, dept_id, salary, hire_date) VALUES
(1, 'Anand', 105, 52000.00, '2020-04-12');

```
- Result Grid:** Displays the results of the SELECT query:

emp_id	emp_name	dept_id	salary	hire_date
1	Anand	105	52000.00	2020-04-12
2	Lekshmi	102	60000.00	2019-08-20
3	Rajeev	103	75000.00	2021-03-15
4	Sneha	104	48000.00	2022-11-10
5	Vishnu	105	51000.00	2023-06-01
6	Shahab	106	46000.00	2018-12-25
7	Hari	106	58000.00	2020-07-14
8	Sagar	101	54000.00	2021-09-19
9	Arjun	105	50000.00	2019-02-05
10	Nithin	103	61000.00	2022-03-27
- Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
7	15:10:56	INSERT INTO employees (emp_id, emp_name, dept_id, salary, hire_date) VALUES (1, 'A...	15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0	0.015 sec
8	15:11:26	SELECT * FROM departments LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
9	15:11:55	SELECT * FROM employees LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.016 sec

2. Show the names of employees who earn more than 50,000

SELECT emp_name FROM employees WHERE salary > 50000;

The screenshot shows the SQL Enterprise Manager interface. The left pane displays the 'MANAGEMENT' tree with 'Server Status' selected. The main pane shows a query window with the following SQL code:

```
64 (15, 'Aparna', 103, 62000.00, '2023-04-03');
65
66 SELECT * FROM departments;
67 SELECT * FROM employees;
68
69 SELECT emp_name FROM employees WHERE salary > 50000;
```

The 'Result Grid' shows the results of the query:

emp_name
Anand
Lekshmi
Rajeev
Vishnu
Hari

The 'Output' pane shows the execution plan and results for the query:

#	Time	Action	Message	Duration / Fetch
9	15:11:55	SELECT * FROM employees LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.016 sec
10	15:15:58	SELECT * FROM departments LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
11	15:17:47	SELECT emp_name FROM employees WHERE salary > 50000 LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

5. 3. Retrieve the list of unique department IDs from the employees table

SELECT DISTINCT dept_id FROM employees;

The screenshot shows the SQL Enterprise Manager interface. The left pane displays the 'MANAGEMENT' tree with 'Server Status' selected. The main pane shows a query window with the following SQL code:

```
67 SELECT * FROM employees;
68
69 SELECT emp_name FROM employees WHERE salary > 50000;
70
71 SELECT DISTINCT dept_id FROM employees;
```

The 'Result Grid' shows the results of the query:

dept_id
105
102
103
104
106

The 'Output' pane shows the execution plan and results for the query:

#	Time	Action	Message	Duration / Fetch
10	15:15:58	SELECT * FROM departments LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
11	15:17:47	SELECT emp_name FROM employees WHERE salary > 50000 LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
12	15:27:03	SELECT DISTINCT dept_id FROM employees LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

4. Display employee names and salaries sorted by salary descending

SELECT emp_name, salary FROM employees ORDER BY salary DESC;

SQL File 8

```

65
66 • SELECT * FROM departments;
67 • SELECT * FROM employees;
68
69 • SELECT emp_name FROM employees WHERE salary > 50000;
70
71 • SELECT DISTINCT dept_id FROM employees;
72
73 • SELECT emp_name, salary FROM employees ORDER BY salary DESC;
74

```

emp_name	salary
Rajeev	75000.00
Aparna	62000.00
Nikhin	61000.00
Lekshmi	60000.00
Hari	58000.00

employees 6

#	Time	Action	Message	Duration / Fetch
11	15:17:47	SELECT emp_name FROM employees WHERE salary > 50000 LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
12	15:27:03	SELECT DISTINCT dept_id FROM employees LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
13	15:27:58	SELECT emp_name, salary FROM employees ORDER BY salary DESC LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec

5. Find all employees hired after January 1, 2020

`SELECT * FROM employees WHERE hire_date > '2020-01-01';`

SQL File 8

```

67 • SELECT * FROM employees;
68
69 • SELECT emp_name FROM employees WHERE salary > 50000;
70
71 • SELECT DISTINCT dept_id FROM employees;
72
73 • SELECT emp_name, salary FROM employees ORDER BY salary DESC;
74
75 • SELECT * FROM employees WHERE hire_date > '2020-01-01';
76

```

emp_id	emp_name	dept_id	salary	hire_date
1	Anand	105	52000.00	2020-04-12
3	Rajeev	103	75000.00	2021-03-15
4	Sneha	104	48000.00	2022-11-10
5	Vishnu	105	51000.00	2023-06-01
7	Hari	106	58000.00	2020-07-14

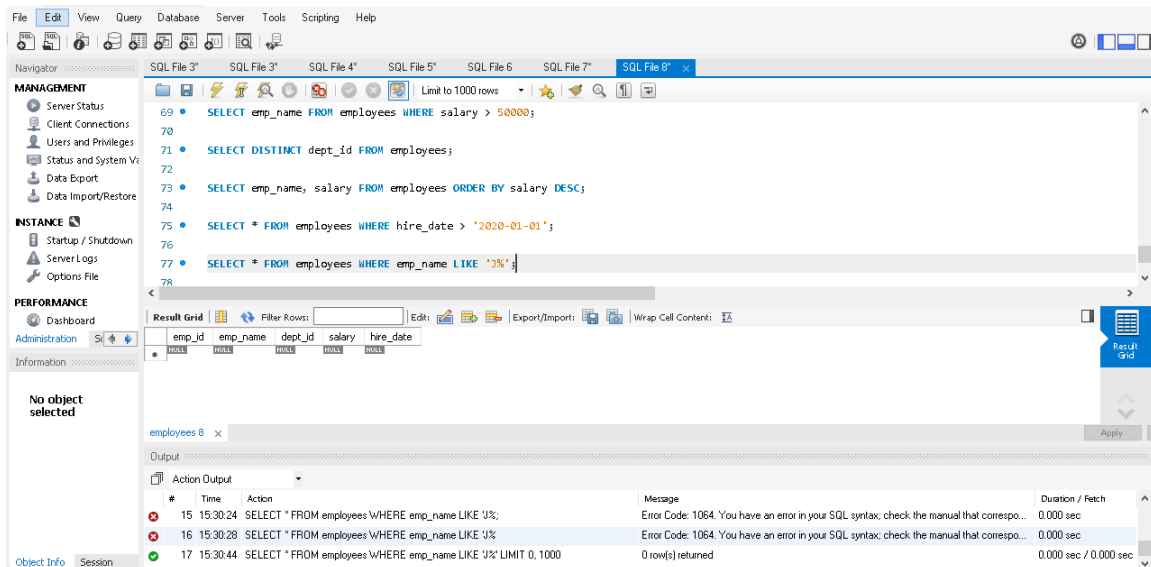
employees 7

#	Time	Action	Message	Duration / Fetch
12	15:27:03	SELECT DISTINCT dept_id FROM employees LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
13	15:27:58	SELECT emp_name, salary FROM employees ORDER BY salary DESC LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec
14	15:29:13	SELECT * FROM employees WHERE hire_date > '2020-01-01' LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec

Part B:

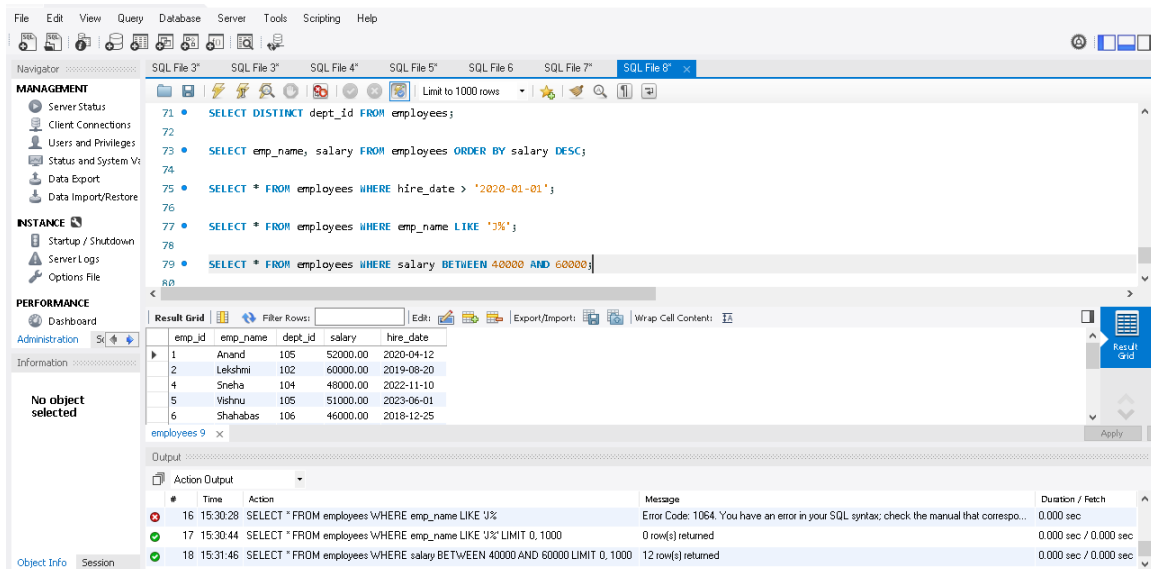
6. Find employees whose names start with 'J'

`SELECT * FROM employees WHERE emp_name LIKE 'J%';`



7. Get all employees with salary between 40,000 and 60,000

SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000;



8. Show employees who do not belong to department ID 102

SELECT * FROM employees WHERE dept_id <> 102;

The screenshot shows the SQL Enterprise Manager interface. The left pane contains the 'Navigator' tree with 'MANAGEMENT' (Server Status, Client Connections, Users and Privileges, Status and System V, Data Export, Data Import/Restore), 'INSTANCE' (Startup / Shutdown, Server Logs, Options File), and 'PERFORMANCE' (Dashboard, Administration, Information). The main pane displays a query window with the following SQL code:

```

73 • SELECT emp_name, salary FROM employees ORDER BY salary DESC;
74
75 • SELECT * FROM employees WHERE hire_date > '2020-01-01';
76
77 • SELECT * FROM employees WHERE emp_name LIKE '%';
78
79 • SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000;
80
81 • SELECT * FROM employees WHERE dept_id <> 102;
82
83

```

The 'Result Grid' shows the following data:

emp_id	emp_name	dept_id	salary	hire_date
1	Anand	105	52000.00	2020-04-12
3	Rajeev	103	75000.00	2021-03-15
4	Sneha	104	48000.00	2022-11-10
5	Vishnu	105	51000.00	2023-06-01
6	Shahabas	106	46000.00	2018-12-25

The 'Output' pane shows the following action output:

#	Time	Action	Message	Duration / Fetch
17	15:30:44	SELECT * FROM employees WHERE emp_name LIKE '%'	0 row(s) returned	0.000 sec / 0.000 sec
18	15:31:46	SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000	12 row(s) returned	0.000 sec / 0.000 sec
19	15:32:29	SELECT * FROM employees WHERE dept_id <> 102	13 row(s) returned	0.000 sec / 0.000 sec

Part C:

9. Find the total number of employees in the company

SELECT COUNT(*) AS total_employees FROM employees;

The screenshot shows the SQL Enterprise Manager interface. The left pane contains the 'Navigator' tree with 'MANAGEMENT' (Server Status, Client Connections, Users and Privileges, Status and System V, Data Export, Data Import/Restore), 'INSTANCE' (Startup / Shutdown, Server Logs, Options File), and 'PERFORMANCE' (Dashboard, Administration, Information). The main pane displays a query window with the following SQL code:

```

75 • SELECT * FROM employees WHERE hire_date > '2020-01-01';
76
77 • SELECT * FROM employees WHERE emp_name LIKE '%';
78
79 • SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000;
80
81 • SELECT * FROM employees WHERE dept_id <> 102;
82
83 • SELECT COUNT(*) AS total_employees FROM employees;
84

```

The 'Result Grid' shows the following data:

total_employees
15

The 'Output' pane shows the following action output:

#	Time	Action	Message	Duration / Fetch
18	15:31:46	SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000	12 row(s) returned	0.000 sec / 0.000 sec
19	15:32:29	SELECT * FROM employees WHERE dept_id <> 102	13 row(s) returned	0.000 sec / 0.000 sec
20	15:33:37	SELECT COUNT(*) AS total_employees FROM employees	1 row(s) returned	0.016 sec / 0.000 sec

10. Show the average salary in each department

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

The screenshot shows the SQL Developer interface. The query window contains the following SQL code:

```

81 SELECT * FROM employees WHERE dept_id <> 102;
82
83 SELECT COUNT(*) AS total_employees FROM employees;
84
85 SELECT dept_id, AVG(salary) AS avg_salary FROM employees GROUP BY dept_id;
86
87
88
89

```

The result grid shows the output of the third query:

dept_id	avg_salary
105	51000.000000
102	54500.000000
103	66000.000000
104	47500.000000
106	53250.000000

The output pane shows the execution of three queries:

#	Time	Action	Message	Duration / Fetch
19	15:32:29	SELECT * FROM employees WHERE dept_id <> 102 LIMIT 0, 1000	13 row(s) returned	0.000 sec / 0.000 sec
20	15:33:37	SELECT COUNT(*) AS total_employees FROM employees LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
21	15:34:20	SELECT dept_id, AVG(salary) AS avg_salary FROM employees GROUP BY dept_id LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

11. Find the highest salary in the employees table

SELECT MAX(salary) AS highest_salary FROM employees;

The screenshot shows the SQL Developer interface. The query window contains the following SQL code:

```

79 SELECT * FROM employees WHERE salary BETWEEN 40000 AND 60000;
80
81 SELECT * FROM employees WHERE dept_id <> 102;
82
83 SELECT COUNT(*) AS total_employees FROM employees;
84
85 SELECT dept_id, AVG(salary) AS avg_salary FROM employees GROUP BY dept_id;
86
87 SELECT MAX(salary) AS highest_salary FROM employees;
88

```

The result grid shows the output of the eighth query:

highest_salary
75000.00

The output pane shows the execution of three queries:

#	Time	Action	Message	Duration / Fetch
20	15:33:37	SELECT COUNT(*) AS total_employees FROM employees LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
21	15:34:20	SELECT dept_id, AVG(salary) AS avg_salary FROM employees GROUP BY dept_id LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
22	15:35:23	SELECT MAX(salary) AS highest_salary FROM employees LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

12. Count how many employees were hired in each year

SELECT YEAR(hire_date) AS hire_year, COUNT(*) AS total_hired FROM employees GROUP BY YEAR(hire_date);

SQL File 3* SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8*

85 • `SELECT dept_id, AVG(salary) AS avg_salary FROM employees GROUP BY dept_id;`

86

87 • `SELECT MAX(salary) AS highest_salary FROM employees;`

88

89 • `SELECT YEAR(hire_date) AS hire_year, COUNT(*) AS total_hired FROM employees GROUP BY YEAR(hire_date);`

90

91

92

93

Result Grid

hire_year	total_hired
2020	3
2019	2
2021	3
2022	3
2023	3

Result 14 x

Output

#	Time	Action	Message	Duration / Fetch
21	15:34:20	SELECT dept_id, AVG(salary) AS avg_salary FROM employees GROUP BY dept_id LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
22	15:35:23	SELECT MAX(salary) AS highest_salary FROM employees LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
23	15:36:24	SELECT YEAR(hire_date) AS hire_year, COUNT(*) AS total_hired FROM employees GROUP BY YEAR(hire_date)	6 row(s) returned	0.000 sec / 0.000 sec

Part D:

13. Display employee names along with their department names

`SELECT e.emp_name, d.dept_name FROM employees e JOIN departments d ON e.dept_id = d.dept_id;`

SQL File 3* SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8*

87 • `SELECT MAX(salary) AS highest_salary FROM employees;`

88

89 • `SELECT YEAR(hire_date) AS hire_year, COUNT(*) AS total_hired FROM employees GROUP BY YEAR(hire_date);`

90

91 • `SELECT e.emp_name, d.dept_name FROM employees e JOIN departments d ON e.dept_id = d.dept_id;`

92

93

94

95

Result Grid

emp_name	dept_name
Anand	IT Support
Lekshmi	Finance
Rajeev	Engineering
Sneha	Marketing
Vishnu	IT Support

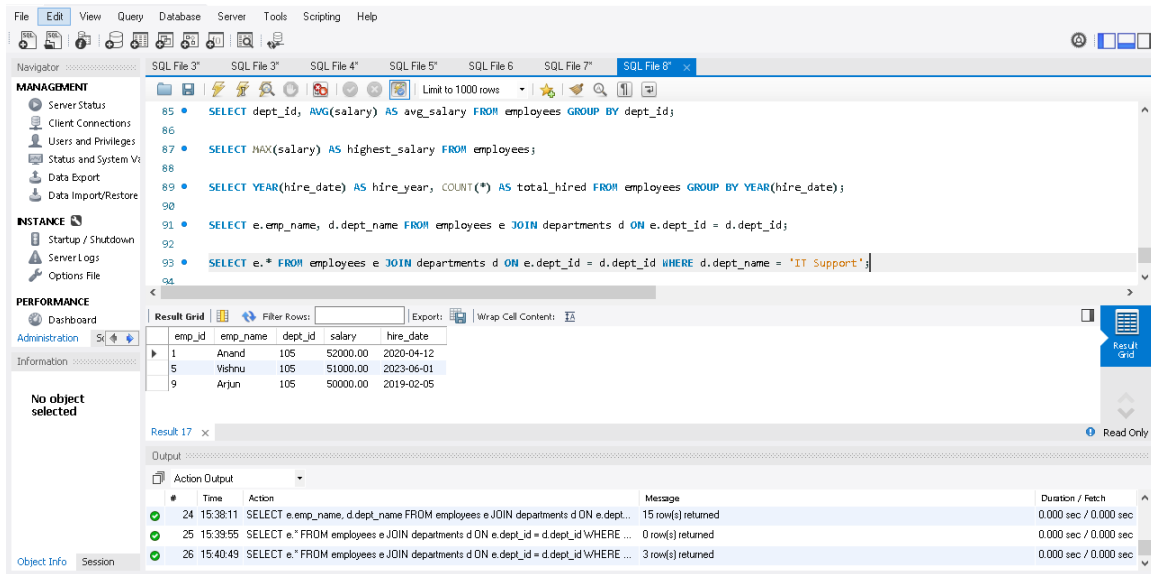
Result 15 x

Output

#	Time	Action	Message	Duration / Fetch
22	15:35:23	SELECT MAX(salary) AS highest_salary FROM employees LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
23	15:36:24	SELECT YEAR(hire_date) AS hire_year, COUNT(*) AS total_hired FROM employees GROUP BY YEAR(hire_date)	6 row(s) returned	0.000 sec / 0.000 sec
24	15:38:11	SELECT e.emp_name, d.dept_name FROM employees e JOIN departments d ON e.dept_id = d.dept_id	15 row(s) returned	0.000 sec / 0.000 sec

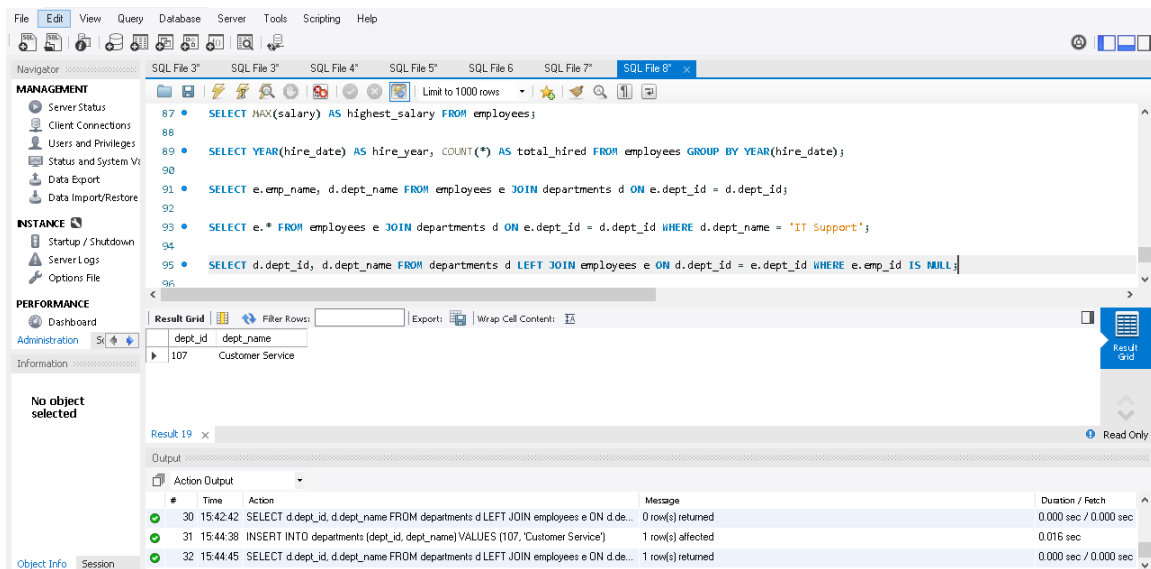
14. Find all employees working in the 'Sales' department

`SELECT e.* FROM employees e JOIN departments d ON e.dept_id = d.dept_id WHERE d.dept_name = 'Sales';`



15. List departments with no employees

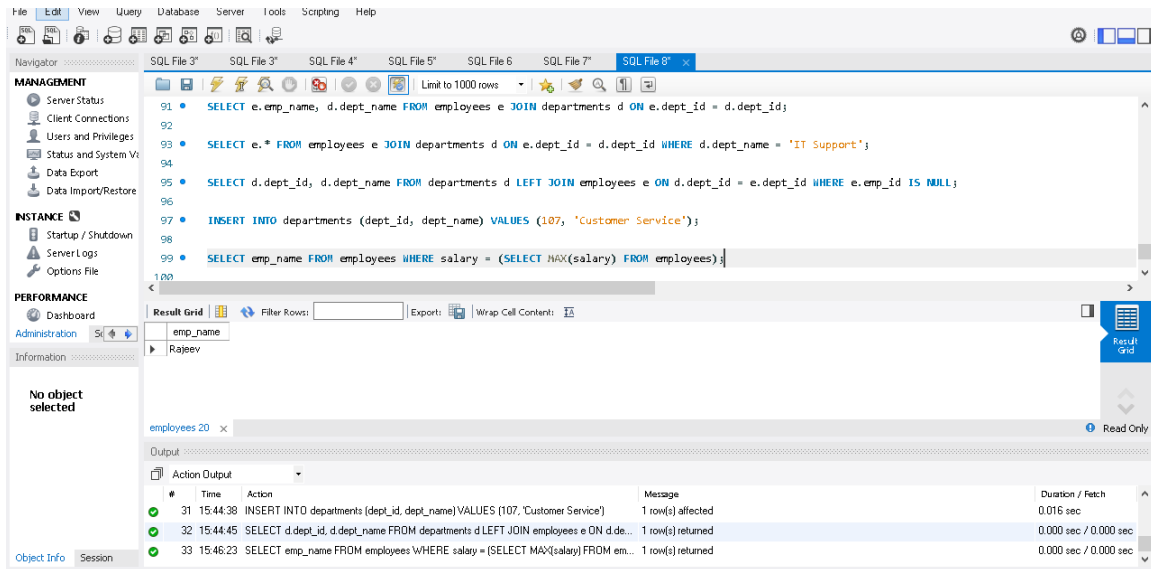
SELECT d.dept_id, d.dept_name FROM departments d LEFT JOIN employees e ON d.dept_id = e.dept_id WHERE e.emp_id IS NULL;



Part E:

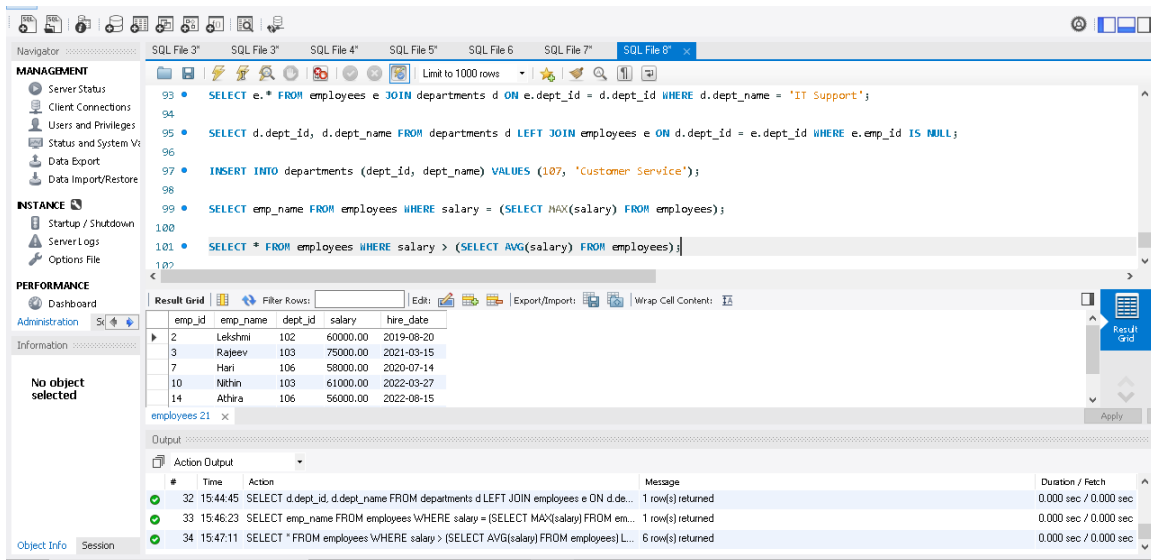
16. Find the name(s) of employee(s) with the highest salary

SELECT emp_name FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);



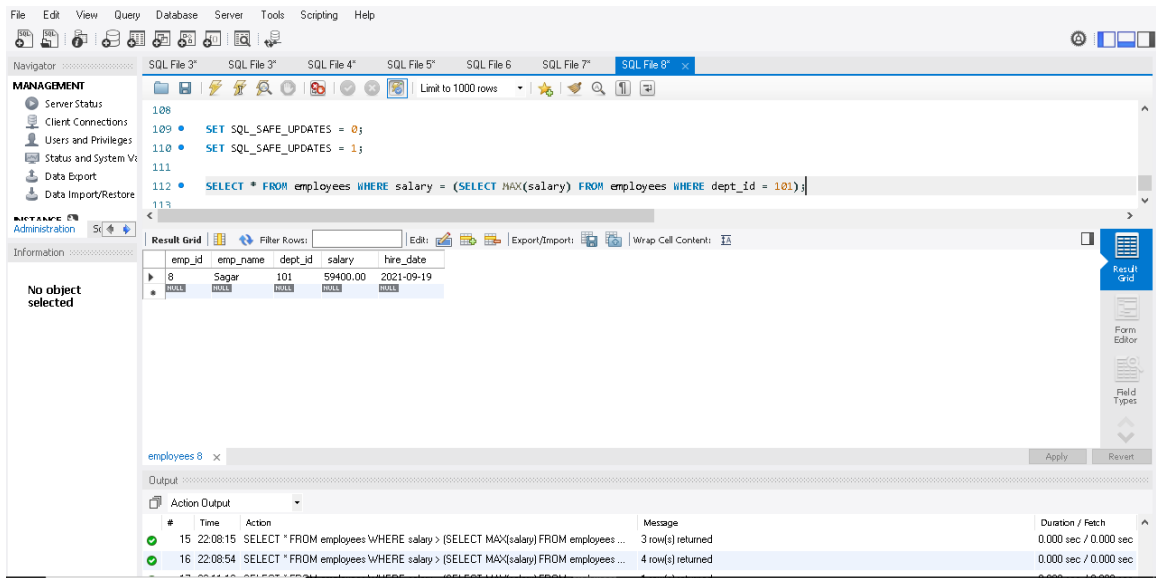
17. List employees whose salary is above the average salary of all employees

SELECT * FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);



6. 18. Show employees who earn more than the maximum salary in department ID 101

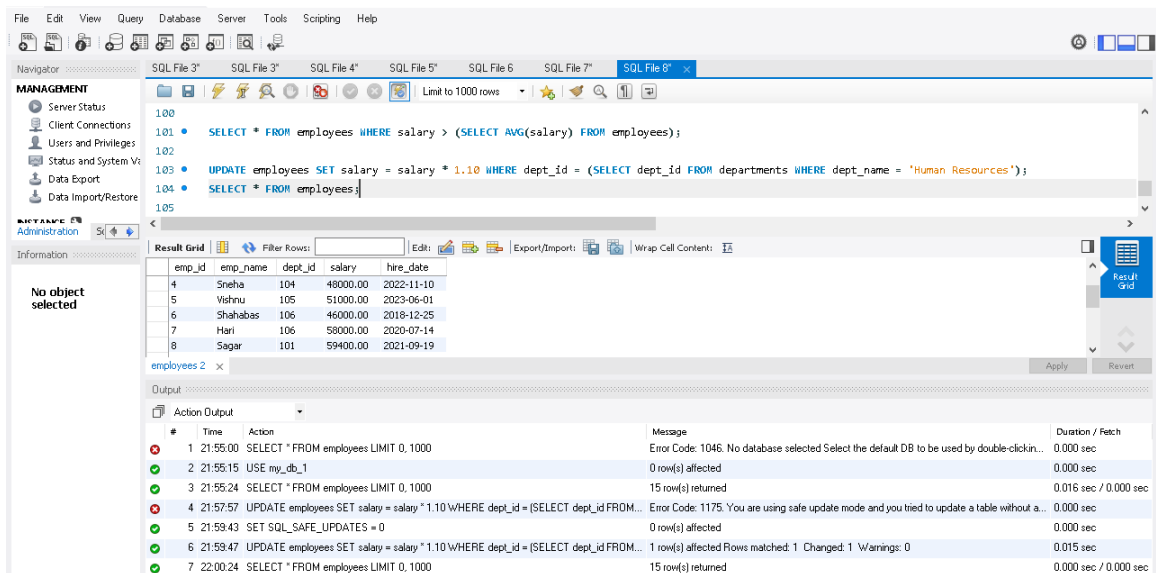
SELECT * FROM employees WHERE salary = (SELECT MAX(salary) FROM employees WHERE dept_id = 101);



Part F:

19. Update the salary by 10% for employees in the 'HR' department (Subquery)

UPDATE employees SET salary = salary * 1.10 WHERE dept_id = (SELECT dept_id FROM departments WHERE dept_name = 'Human Resources');



20. Delete employees who have a salary less than 30,000

DELETE FROM employees WHERE salary < 30000;

File Edit View Query Database Server Tools Scripting Help

SQL File 3" SQL File 3" SQL File 4" SQL File 5" SQL File 6" SQL File 7" SQL File 8" x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Vt
- Data Export
- Data Import/Restore

Administration

Information

No object selected

Limit to 1000 rows

```
103 • UPDATE employees SET salary = salary * 1.10 WHERE dept_id = (SELECT dept_id FROM departments WHERE dept_name = 'Human Resources');
104 • SELECT * FROM employees;
105
106 • DELETE FROM employees WHERE salary < 30000;
107
108
```

Result Grid

emp_id	emp_name	dept_id	salary	hire_date
1	Anand	105	52000.00	2020-04-12
2	Lekshmi	102	60000.00	2019-08-20
3	Rajeev	103	75000.00	2021-03-15
4	Sneha	104	48000.00	2022-11-10
5	Vishnu	105	51000.00	2023-06-01
6	Shahabaz	106	46000.00	2018-12-25
7	Hari	106	58000.00	2020-07-14
8	Sagar	101	59400.00	2021-09-19
9	Arun	105	50000.00	2019-02-05
10	Nithin	103	61000.00	2022-03-27
11	Sachin	102	49000.00	2023-01-09
12	Anjali	104	47000.00	2020-10-31

employees 3 x

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	21:55:00	SELECT * FROM employees LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicki...	0.000 sec
2	21:55:15	USE my_db_1	0 row(s) affected	0.000 sec