

Lab Report-02

Course Title : DATABASE MANAGEMENT SYSTEM

Course Code : CSEC321

Submitted By

Name : Robaaed Afred Tanik

ID No : 2213081048

Batch : 55

Submitted To

Md. Harun Ar Rashid

Senior Lecturer, Dept. CSE

Uttara University

Date Of Submission : 06.11.2024

Lab Report No- 02

Task No 01:

The screenshot shows the HeidiSQL interface with the following SQL queries:

```
71 (3, 'Tablet', 3),
72 (4, 'Monitor', 2),
73 (5, 'Keyboard', 4),
74 (6, 'Mouse', 4),
75 (7, 'Printer', 5),
76 (8, 'Router', 5),
77 (9, 'Speaker', 6),
78 (10, 'Headphone', 6),
79 (11, 'Smartwatch', 7);
80
81 -- Task 1:
82 SELECT * FROM employees;
83
84 SELECT * FROM employees WHERE salary > 50000;
85
86 -- Task 2:
87
88 SELECT * FROM customers WHERE customer_name LIKE 'A%';
89
90 SELECT * FROM customers WHERE customer_name LIKE 'XAX';
91
92 SELECT * FROM customers WHERE customer_name LIKE 'XA';
93
94 -- Task 3:
95 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
96
97 -- Task 4:
98 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
99
100 -- Task 5:
101 SELECT * FROM products WHERE category_id IN (1, 2, 3);
102
103
104
```

The results of the queries are displayed below:

employees (9 x 4)

#	employee_id	name	department_id	salary
5	7	Zahangir Alam	1	62,000.0
6	8	Ahmad Ullah	2	70,000.0
7	9	Ripon Sarkar	3	45,000.0
8	10	Rakib Sarkar	2	52,000.0
9	11	Sibal Hossen	1	94,000.0

customers (13 x 2)

#	customer_id	customer_name
1	1	Ahmad Ullah
2	3	Ahmad Ullah
3	5	Ahmad Ullah
4	6	Ahmad Ullah
5	8	Ahmad Ullah

1152 SELECT * FROM employees WHERE salary > 50000;
1153 /* */
1154 /* Affected rows: 0 Found rows: 9 Warnings: 0 Duration for 1 query: 0.000 sec. */

Task No 02:

The screenshot shows the HeidiSQL interface with the following SQL queries:

```
71 (3, 'Tablet', 3),
72 (4, 'Monitor', 2),
73 (5, 'Keyboard', 4),
74 (6, 'Mouse', 4),
75 (7, 'Printer', 5),
76 (8, 'Router', 5),
77 (9, 'Speaker', 6),
78 (10, 'Headphone', 6),
79 (11, 'Smartwatch', 7);
80
81 -- Task 1:
82 SELECT * FROM employees;
83
84 SELECT * FROM employees WHERE salary > 50000;
85
86 -- Task 2:
87
88 SELECT * FROM customers WHERE customer_name LIKE 'A%';
89
90 SELECT * FROM customers WHERE customer_name LIKE 'XAX';
91
92 SELECT * FROM customers WHERE customer_name LIKE 'XA';
93
94 -- Task 3:
95 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
96
97 -- Task 4:
98 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
99
100 -- Task 5:
101 SELECT * FROM customers WHERE customer_name LIKE 'XA';
102
103
104
```

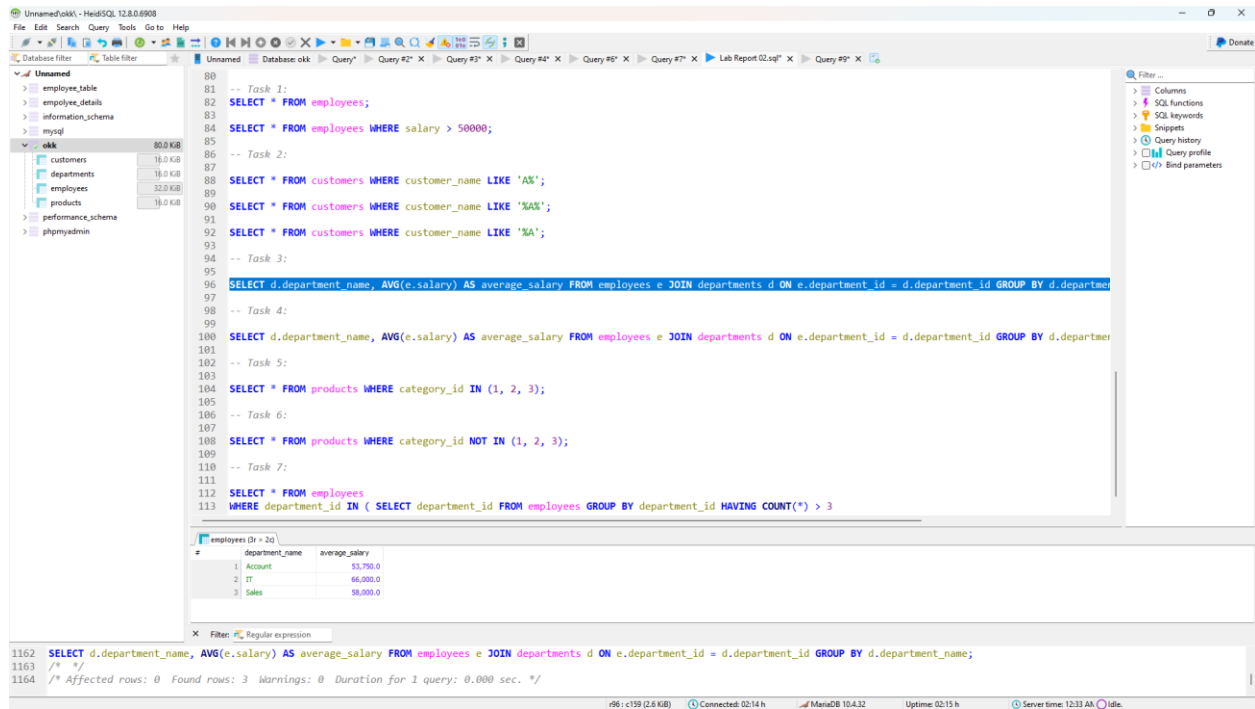
The results of the queries are displayed below:

customers (13 x 2)

#	customer_id	customer_name
1	1	Ahmad Ullah
2	3	Ahmad Ullah
3	5	Ahmad Ullah
4	6	Ahmad Ullah
5	8	Ahmad Ullah

1159 SELECT * FROM customers WHERE customer_name LIKE 'XA';
1160 /* */
1161 /* Affected rows: 0 Found rows: 13 Warnings: 0 Duration for 3 queries: 0.000 sec. */

Task No 03:



HeidiSQL 12.8.0.6908 interface showing SQL queries for Task No 03. The queries are as follows:

```
-- Task 1:
SELECT * FROM employees;

SELECT * FROM employees WHERE salary > 50000;

-- Task 2:
SELECT * FROM customers WHERE customer_name LIKE 'AS';
SELECT * FROM customers WHERE customer_name LIKE '%AS';
SELECT * FROM customers WHERE customer_name LIKE '%A';

-- Task 3:
SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_name;

-- Task 4:
SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_name;

-- Task 5:
SELECT * FROM products WHERE category_id IN (1, 2, 3);

-- Task 6:
SELECT * FROM products WHERE category_id NOT IN (1, 2, 3);

-- Task 7:
SELECT * FROM employees
WHERE department_id IN ( SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 3 )
```

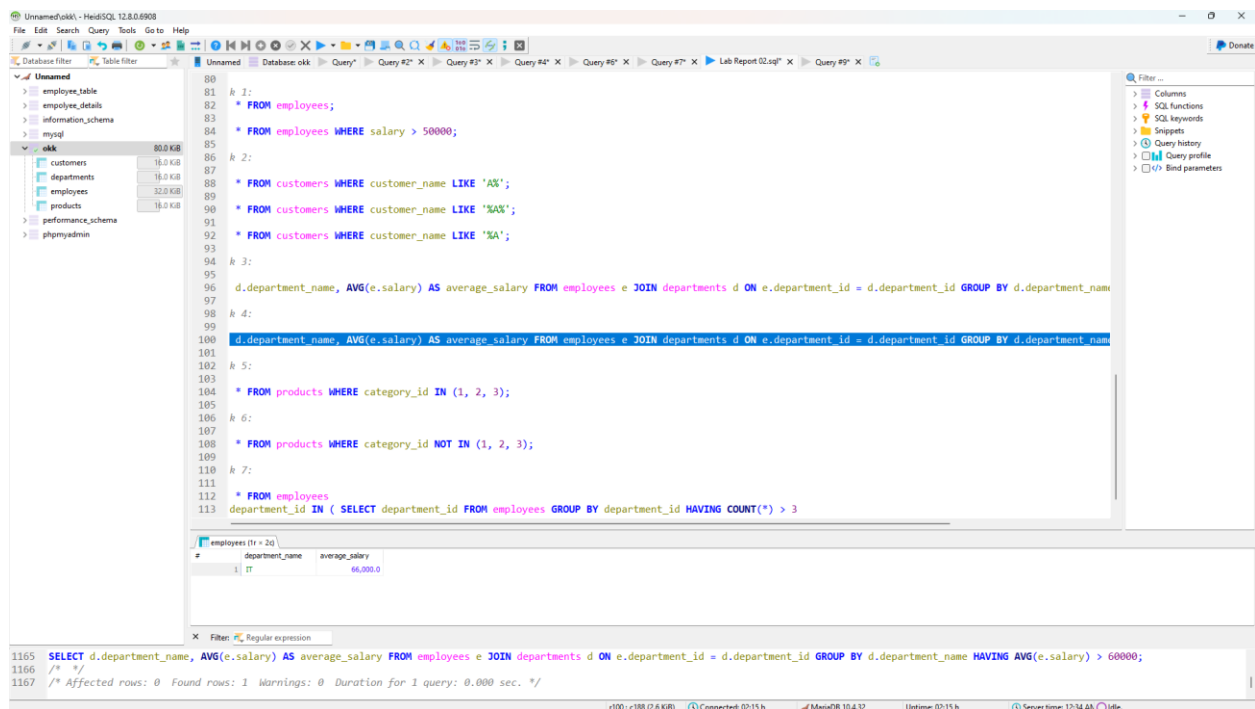
The results of the queries are displayed in the bottom panel:

#	department_name	average_salary
1	Account	53,750.0
2	IT	66,000.0
3	Sales	58,000.0

Filter: Regular expression

1162 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_name;
1163 /* */
1164 /* Affected rows: 0 Found rows: 3 Warnings: 0 Duration for 1 query: 0.000 sec. */

Task No 04:



HeidiSQL 12.8.0.6908 interface showing SQL queries for Task No 04. The queries are as follows:

```
-- Task 1:
SELECT * FROM employees;

SELECT * FROM employees WHERE salary > 50000;

-- Task 2:
SELECT * FROM customers WHERE customer_name LIKE 'AS';
SELECT * FROM customers WHERE customer_name LIKE '%AS';
SELECT * FROM customers WHERE customer_name LIKE '%A';

-- Task 3:
SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_name;

-- Task 4:
SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_name;

-- Task 5:
SELECT * FROM products WHERE category_id IN (1, 2, 3);

-- Task 6:
SELECT * FROM products WHERE category_id NOT IN (1, 2, 3);

-- Task 7:
SELECT * FROM employees
WHERE department_id IN ( SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 3 )
```

The results of the queries are displayed in the bottom panel:

#	department_name	average_salary
1	IT	66,000.0

Filter: Regular expression

1165 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_name HAVING AVG(e.salary) > 60000;
1166 /* */
1167 /* Affected rows: 0 Found rows: 1 Warnings: 0 Duration for 1 query: 0.000 sec. */

Task No 05:

The screenshot shows the HeidiSQL interface with the following SQL queries:

```
82 SELECT * FROM employees;
83
84 SELECT * FROM employees WHERE salary > 50000;
85
86 -- Task 2:
87
88 SELECT * FROM customers WHERE customer_name LIKE 'A%';
89
90 SELECT * FROM customers WHERE customer_name LIKE '%A%';
91
92 SELECT * FROM customers WHERE customer_name LIKE '%A';
93
94 -- Task 3:
95
96 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
97
98 -- Task 4:
99
100 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
101
102 -- Task 5:
103
104 SELECT * FROM products WHERE category_id IN (1, 2, 3);
105
106 -- Task 6:
107
108 SELECT * FROM products WHERE category_id NOT IN (1, 2, 3);
109
110 -- Task 7:
111
112 SELECT * FROM employees
113 WHERE department_id IN ( SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 3
114 );
115
```

The results pane shows the output of the query `SELECT * FROM products WHERE category_id IN (1, 2, 3);`:

#	product_id	product_name	category_id
1	1	Laptop	1
2	2	Mobile Phone	2
3	3	Tablet	3
4	4	Monitor	2

The status bar at the bottom indicates: r104: c1 (2.6 KiB) Connected: 02:16 h MariaDB 10.4.32 Uptime: 02:16 h Server time: 12:34 AA Idle.

Task No 06:

The screenshot shows the HeidiSQL interface with the following SQL queries:

```
82 SELECT * FROM employees;
83
84 SELECT * FROM employees WHERE salary > 50000;
85
86 -- Task 2:
87
88 SELECT * FROM customers WHERE customer_name LIKE 'A%';
89
90 SELECT * FROM customers WHERE customer_name LIKE '%A%';
91
92 SELECT * FROM customers WHERE customer_name LIKE '%A';
93
94 -- Task 3:
95
96 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
97
98 -- Task 4:
99
100 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
101
102 -- Task 5:
103
104 SELECT * FROM products WHERE category_id IN (1, 2, 3);
105
106 -- Task 6:
107
108 SELECT * FROM products WHERE category_id NOT IN (1, 2, 3);
109
110 -- Task 7:
111
112 SELECT * FROM employees
113 WHERE department_id IN ( SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 3
114 );
115
```

The results pane shows the output of the query `SELECT * FROM products WHERE category_id NOT IN (1, 2, 3);`:

#	product_id	product_name	category_id
1	5	Keyboard	4
2	6	Mouse	4
3	7	Printer	5
4	8	Router	5
5	9	Speaker	6

The status bar at the bottom indicates: r108: c1 (2.6 KiB) Connected: 02:16 h MariaDB 10.4.32 Uptime: 02:16 h Server time: 12:35 AA Idle.

Task No 07:

The screenshot shows the HeidiSQL 12.8.0.6903 interface. The left sidebar displays a database tree with the following structure:

- Unnamed
 - employee_table
 - employee_details
 - information_schema
 - mysql
- ekk (80.0 KiB)
 - customers (18.0 KiB)
 - departments (18.0 KiB)
 - employees (32.0 KiB)
 - products (18.0 KiB)
 - performance_schema
 - phpmyadmin

The main query editor contains the following SQL code:

```
82 SELECT * FROM employees;
83
84 SELECT * FROM employees WHERE salary > 50000;
85
86 -- Task 2:
87
88 SELECT * FROM customers WHERE customer_name LIKE 'A%';
89
90 SELECT * FROM customers WHERE customer_name LIKE '%A%';
91
92 SELECT * FROM customers WHERE customer_name LIKE '%A';
93
94 -- Task 3:
95
96 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
97
98 -- Task 4:
99
100 SELECT d.department_name, AVG(e.salary) AS average_salary FROM employees e JOIN departments d ON e.department_id = d.department_id GROUP BY d.department_id;
101
102 -- Task 5:
103
104 SELECT * FROM products WHERE category_id IN (1, 2, 3);
105
106 -- Task 6:
107
108 SELECT * FROM products WHERE category_id NOT IN (1, 2, 3);
109
110 -- Task 7:
111
112 SELECT * FROM employees
113 WHERE department_id IN ( SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 3 );
114
115
```

The result set for the last query is displayed in a table with 5 columns: employee_id, name, department_id, salary, and a filter icon. The table contains 5 rows of data:

#	employee_id	name	department_id	salary
1	1	Noor Hassan	1	55,000.0
2	2	Tamim Rahman	2	48,000.0
3	3	Fahar Shahan	1	41,000.0
4	5	Shoyab Ahmed	2	45,000.0
5	7	Zahangir Alam	1	62,000.0

The status bar at the bottom shows: 1174 SELECT * FROM employees WHERE department_id IN (SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 3);
1175 /* */
1176 /* Affected rows: 0 Found rows: 8 Warnings: 0 Duration for 1 query: 0.000 sec. */