

Lab Aggregation

Let us imagine a scenario where a company organizes an annual employee performance review event. During this event, employees are evaluated based on various criteria such as sales performance, customer feedback, and project completion. The data collected from this event is stored in a database with the following schema:

Tables:

employees

departments

performance_reviews

```
CREATE TABLE employees (  
    employee_id INT PRIMARY KEY,  
    employee_name VARCHAR(100),  
    department_id INT  
);
```

```
CREATE TABLE departments (  
    department_id INT PRIMARY KEY,  
    department_name VARCHAR(100)  
);
```

```
CREATE TABLE performance_reviews (  
    review_id INT PRIMARY KEY,  
    employee_id INT,  
    review_date DATE,  
    sales_performance INT,  
    customer_feedback INT,
```

```
project_completion INT,  
FOREIGN KEY (employee_id) REFERENCES employees(employee_id)  
);
```

```
INSERT INTO departments (department_id, department_name)
```

```
VALUES
```

```
(1, 'Sales'),  
(2, 'Customer Support'),  
(3, 'Development');
```

```
INSERT INTO employees (employee_id, employee_name, department_id)
```

```
VALUES
```

```
(1, 'Jubayer', 1),  
(2, 'Ahmed', 2),  
(3, 'Shawon', 3),  
(4, 'Sham', 1);
```

```
INSERT INTO performance_reviews (review_id, employee_id, review_date, sales_performance,  
customer_feedback, project_completion)
```

```
VALUES
```

```
(1, 1, '2024-01-15', 85, 90, 75),  
(2, 2, '2024-02-10', 70, 95, 80),  
(3, 3, '2024-01-25', 60, 88, 85),  
(4, 4, '2024-03-05', 90, 80, 88),  
(5, 1, '2024-06-15', 87, 92, 76),  
(6, 2, '2024-07-20', 78, 91, 82),  
(7, 3, '2024-08-25', 65, 85, 90),
```

(8, 4, '2024-09-15', 93, 88, 90);

Question 1: Write an SQL query to count the total number of performance reviews conducted.

-- Ans

SELECT COUNT(*) AS total_reviews FROM performance_reviews;

The screenshot shows the HeidiSQL interface with a MySQL database named 'lab03'. The query editor contains the following SQL code:

```
59 SELECT MAX(customer_feedback) AS highest_feedback FROM performance_reviews;
60
61 -- Ans 84
62 SELECT d.department_name, SUM(p.project_completion) AS total_project_completion
63 FROM performance_reviews p
64 JOIN employees e ON e.employee_id = p.employee_id
65 JOIN departments d ON d.department_id = p.department_id
66 GROUP BY d.department_name;
67
68 -- Ans 85
69 SELECT d.department_name,
70        AVG(p.sales_performance) AS avg_sales_performance,
71        AVG(p.customer_feedback) AS avg_customer_feedback,
72        AVG(p.project_completion) AS avg_project_completion
73 FROM performance_reviews p
74 JOIN employees e ON e.employee_id = p.employee_id
75 JOIN departments d ON d.department_id = p.department_id
76 GROUP BY d.department_name;
```

The results pane shows the output of the last query (line 76):

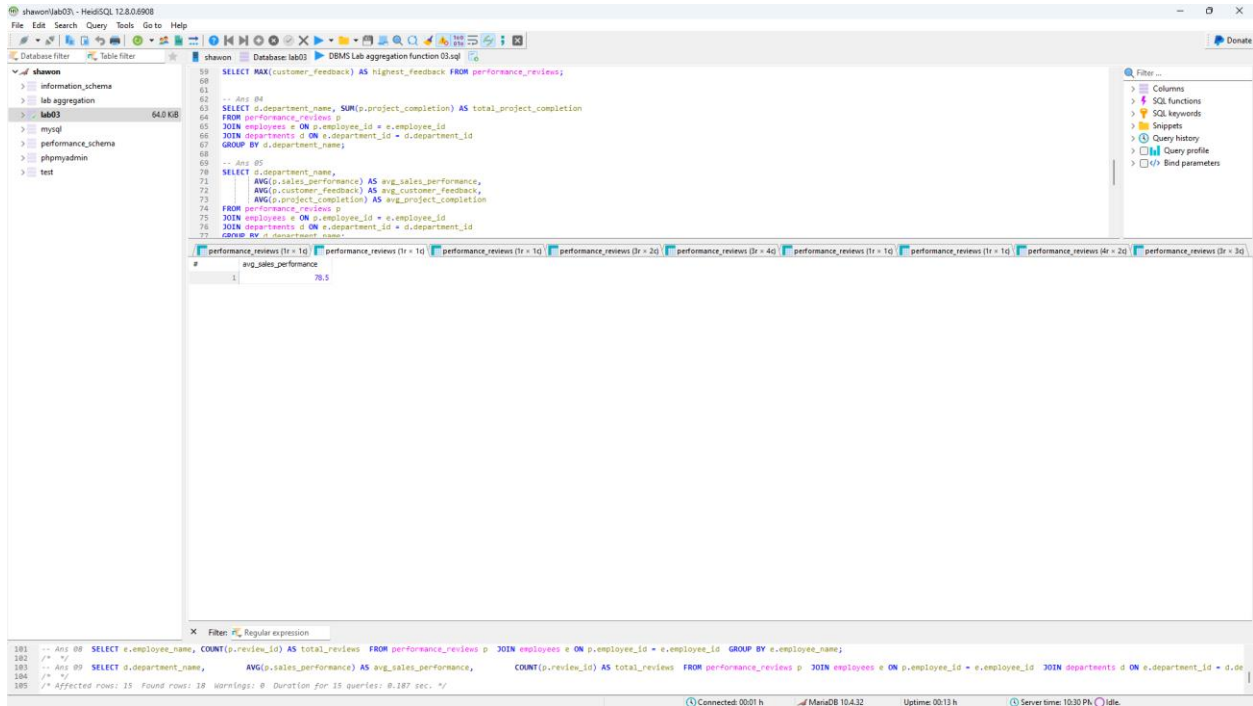
#	total_reviews
1	8

The status bar at the bottom indicates: r105: c1 (2.9 KiB) | Connected: 00:01 h | MariaDB 10.4.32 | Uptime: 00:13 h | Server time: 10:29 PM | Idle.

Question 2: Write an SQL query to calculate the average sales performance score of all employees.

-- Ans 02

SELECT AVG(sales_performance) AS avg_sales_performance FROM performance_reviews;



The screenshot shows the HeidiSQL interface with a MySQL database named 'lab03'. The SQL editor contains a query to calculate the average sales performance score. The results pane shows a single row with the value 78.5.

```
59 SELECT MAX(customer_feedback) AS highest_feedback FROM performance_reviews;
60
61 -- Ans 04
62 SELECT d.department_name, SUM(p.project_completion) AS total_project_completion
63 FROM performance_reviews p
64 JOIN employees e ON e.employee_id = p.employee_id
65 JOIN departments d ON d.department_id = p.department_id
66 GROUP BY d.department_name;
67
68 -- Ans 05
69 SELECT d.department_name,
70        AVG(p.sales_performance) AS avg_sales_performance,
71        AVG(p.customer_feedback) AS avg_customer_feedback,
72        AVG(p.project_completion) AS avg_project_completion
73 FROM performance_reviews p
74 JOIN employees e ON e.employee_id = p.employee_id
75 JOIN departments d ON d.department_id = p.department_id
76 GROUP BY d.department_name;
```

#	avg_sales_performance
1	78.5

101 -- Ans 06 SELECT e.employee_name, COUNT(p.review_id) AS total_reviews FROM performance_reviews p JOIN employees e ON p.employee_id = e.employee_id GROUP BY e.employee_name;
102 /* */
103 -- Ans 07 SELECT d.department_name,
104 AVG(p.sales_performance) AS avg_sales_performance,
105 COUNT(p.review_id) AS total_reviews FROM performance_reviews p JOIN employees e ON p.employee_id = e.employee_id JOIN departments d ON d.department_id = p.department_id
106 /* */
107 /* Affected rows: 15 Found rows: 18 Warnings: 0 Duration for 15 queries: 8.187 sec. */

Connected: 00:01 h MariaDB 10.4.32 Uptime: 00:13 h Server time: 10:30 PM Idle.

Question 3: Write an SQL query to find the highest customer feedback score received by any employee.

-- Ans 03

SELECT MAX(customer_feedback) AS highest_feedback FROM performance_reviews;

The screenshot shows a MySQL IDE window titled 'shawnlab03 - HeidiSQL 12.8.0.8908'. The left sidebar displays a database tree with 'shawnlab03' selected. The main editor contains the following SQL query:

```
59 SELECT MAX(customer_feedback) AS highest_feedback FROM performance_reviews;
60
61 -- Ans 04
62 SELECT d.department_name, SUM(p.project_completion) AS total_project_completion
63 FROM performance_reviews p
64 JOIN employees e ON e.employee_id = p.employee_id
65 JOIN departments d ON d.department_id = p.department_id
66 GROUP BY d.department_name;
67
68 -- Ans 05
69 SELECT d.department_name,
70        AVG(p.sales_performance) AS avg_sales_performance,
71        AVG(p.customer_feedback) AS avg_customer_feedback,
72        AVG(p.project_completion) AS avg_project_completion
73 FROM performance_reviews p
74 JOIN employees e ON e.employee_id = p.employee_id
75 JOIN departments d ON d.department_id = p.department_id
76 GROUP BY d.department_name;
```

The results pane shows a single row with the value 95 for the 'highest_feedback' column.

At the bottom, the status bar indicates 'Connected: 00:02 h', 'MariaDB 10.4.32', 'Uptime: 00:14 h', and 'Server time: 10:30 PM'. The bottom-most status bar shows 'Affected rows: 15 Found rows: 0 Warnings: 0 Duration for 15 queries: 0.187 sec.'

Question 4: Write an SQL query to find the total project completion score for each department.

-- Ans 04

SELECT d.department_name, SUM(p.project_completion) **AS** total_project_completion

FROM performance_reviews p

JOIN employees e **ON** p.employee_id = e.employee_id

JOIN departments d **ON** e.department_id = d.department_id

GROUP BY d.department_name;

The screenshot shows the HeidiSQL interface with the following SQL query and results:

```
59 SELECT MAX(customer_feedback) AS highest_feedback FROM performance_reviews;
60
61 -- Ans 04
62 SELECT d.department_name, SUM(p.project_completion) AS total_project_completion
63 FROM performance_reviews p
64 JOIN employees e ON p.employee_id = e.employee_id
65 JOIN departments d ON e.department_id = d.department_id
66 GROUP BY d.department_name;
67
68 -- Ans 05
69 SELECT d.department_name,
70        AVG(p.sales_performance) AS avg_sales_performance,
71        AVG(p.customer_feedback) AS avg_customer_feedback,
72        AVG(p.project_completion) AS avg_project_completion
73 FROM performance_reviews p
74 JOIN employees e ON p.employee_id = e.employee_id
75 JOIN departments d ON e.department_id = d.department_id
76 GROUP BY d.department_name;
```

The results of the query (Ans 04) are shown in a table:

#	department_name	total_project_completion
1	Customer Support	162
2	Development	175
3	Sales	329

The bottom of the screenshot shows the status bar with the following information: Connected: 00:02 h, MariaDB 10.4.32, Uptime: 00:14 h, Server time: 10:30 PM, Idle.

Question 5: Write an SQL query to find the average sales, customer feedback, and project completion scores for each department.

-- Ans 05

```
SELECT d.department_name,  
  
       AVG(p.sales_performance) AS avg_sales_performance,  
  
       AVG(p.customer_feedback) AS avg_customer_feedback,  
  
       AVG(p.project_completion) AS avg_project_completion  
  
FROM performance_reviews p  
  
JOIN employees e ON p.employee_id = e.employee_id  
  
JOIN departments d ON e.department_id = d.department_id  
  
GROUP BY d.department_name;
```

The screenshot shows the HeidiSQL interface with a query window titled 'Database: lab03' and 'DEMS Lab aggregation function 03.sql'. The query is as follows:

```
-- Ans 04  
SELECT MAX(customer_feedback) AS highest_feedback FROM performance_reviews;  
  
-- Ans 05  
SELECT d.department_name, SUM(p.project_completion) AS total_project_completion  
FROM performance_reviews p  
JOIN employees e ON p.employee_id = e.employee_id  
JOIN departments d ON e.department_id = d.department_id  
GROUP BY d.department_name;  
  
-- Ans 05  
SELECT d.department_name,  
       AVG(p.sales_performance) AS avg_sales_performance,  
       AVG(p.customer_feedback) AS avg_customer_feedback,  
       AVG(p.project_completion) AS avg_project_completion  
FROM performance_reviews p  
JOIN employees e ON p.employee_id = e.employee_id  
JOIN departments d ON e.department_id = d.department_id  
GROUP BY d.department_name;
```

The results pane shows the output of the second query:

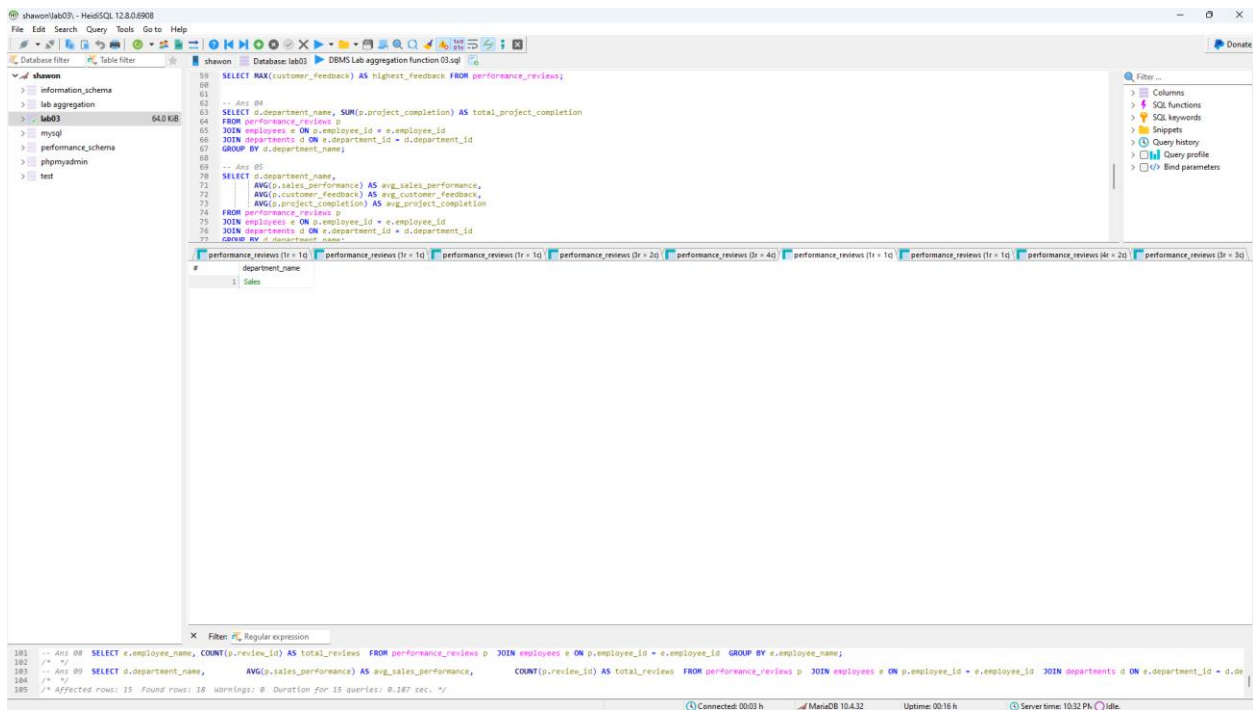
#	department_name	avg_sales_performance	avg_customer_feedback	avg_project_completion
1	Customer Support	74.0	93.0	81.0
2	Development	82.5	86.5	87.5
3	Sales	88.75	87.5	82.25

The status bar at the bottom indicates: 'Connected: 00:02 h', 'MariaDB 10.4.32', 'Uptime: 00:14 h', and 'Server time: 10:31 PM idle'.

Question 6: Write an SQL query to find departments with an average sales performance score greater than 80.

-- Ans 06

```
SELECT d.department_name
FROM performance_reviews p
JOIN employees e ON p.employee_id = e.employee_id
JOIN departments d ON e.department_id = d.department_id
GROUP BY d.department_name
HAVING AVG(p.sales_performance) > 80;
```



The screenshot shows the HeidiSQL interface with a query window. The query is as follows:

```
-- Ans 06
SELECT d.department_name, SUM(p.project_completion) AS total_project_completion
FROM performance_reviews p
JOIN employees e ON p.employee_id = e.employee_id
JOIN departments d ON e.department_id = d.department_id
GROUP BY d.department_name;

-- Ans 07
SELECT d.department_name,
       AVG(p.sales_performance) AS avg_sales_performance,
       AVG(p.customer_feedback) AS avg_customer_feedback,
       AVG(p.project_completion) AS avg_project_completion
FROM performance_reviews p
JOIN employees e ON p.employee_id = e.employee_id
JOIN departments d ON e.department_id = d.department_id
GROUP BY d.department_name;
```

The results pane shows the output of the second query, filtered by the regular expression 'Sales'. The result is a single row for the 'Sales' department:

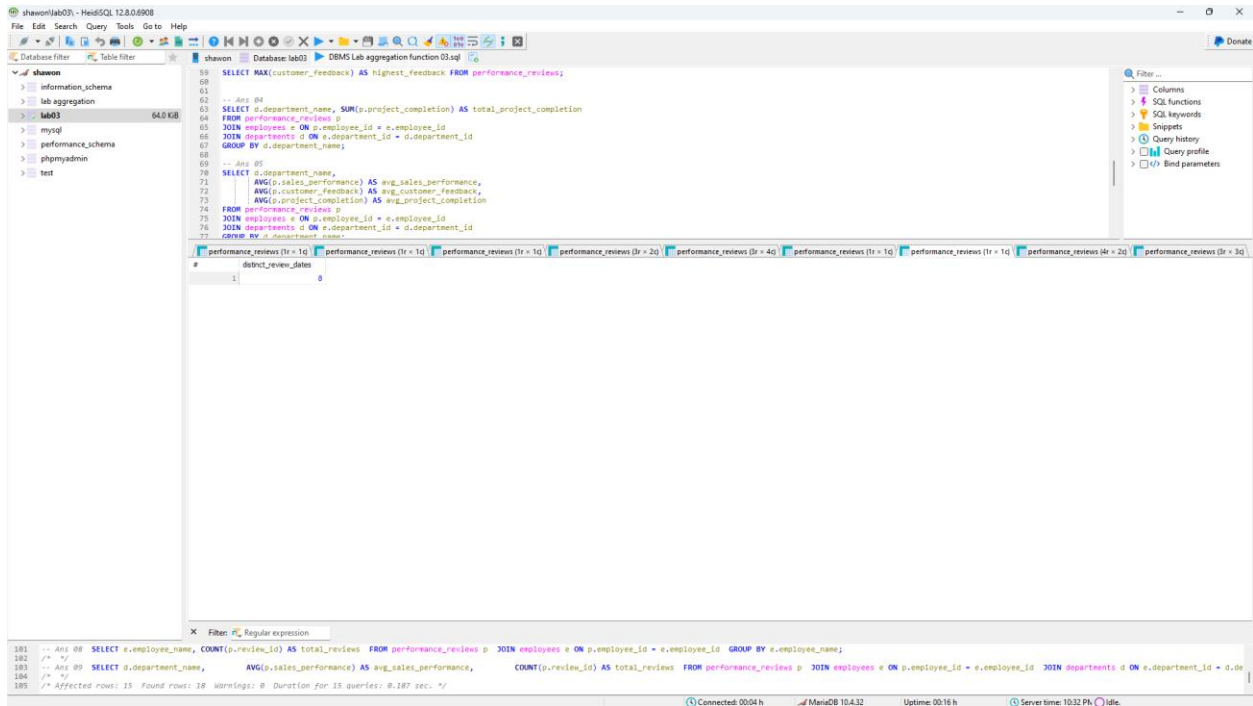
department_name
Sales

The status bar at the bottom indicates: Connected: 00:03 h, MariaDB 10.4.32, Uptime: 00:16 h, Server time: 10:32 PM, 100%.

Question 7: Write an SQL query to count the number of distinct review dates.

-- Ans 07

```
SELECT COUNT(DISTINCT review_date) AS distinct_review_dates  
FROM performance_reviews;
```



Question 8: Write an SQL query to list all employee names along with their total number of reviews.

-- Ans 08

SELECT e.employee_name, **COUNT**(p.review_id) **AS** total_reviews

FROM performance_reviews p

JOIN employees e **ON** p.employee_id = e.employee_id

GROUP BY e.employee_name;

The screenshot shows the HeidiSQL interface with a query window. The query is as follows:

```
-- Ans 08
SELECT e.employee_name, COUNT(p.review_id) AS total_reviews
FROM performance_reviews p
JOIN employees e ON p.employee_id = e.employee_id
GROUP BY e.employee_name;
```

The results window displays the following data:

#	employee_name	total_reviews
1	Alfred	2
2	Subbar	2
3	Shan	2
4	Shawon	2

The status bar at the bottom indicates: Connected: 00:04 h, MariaDB 10.4.32, Uptime: 00:16 h, Server time: 10:33 PM, Idle.

Question 9: Write an SQL query to find the average sales performance and the total number of reviews for each department.

-- Ans 09

```
SELECT d.department_name,  
       AVG(p.sales_performance) AS avg_sales_performance,  
       COUNT(p.review_id) AS total_reviews  
FROM performance_reviews p  
JOIN employees e ON p.employee_id = e.employee_id  
JOIN departments d ON e.department_id = d.department_id  
GROUP BY d.department_name;
```

The screenshot shows the HeidiSQL interface with a query window titled 'DBMS Lab aggregation function 03.sql'. The query is as follows:

```
-- Ans 08  
60 SELECT MAX(customer_feedback) AS highest_feedback FROM performance_reviews;  
61  
62 -- Ans 08  
63 SELECT d.department_name, SUM(p.project_completion) AS total_project_completion  
64 FROM performance_reviews p  
65 JOIN employees e ON p.employee_id = e.employee_id  
66 JOIN departments d ON e.department_id = d.department_id  
67 GROUP BY d.department_name;  
68  
69 -- Ans 09  
70 SELECT d.department_name,  
71        AVG(p.sales_performance) AS avg_sales_performance,  
72        AVG(p.customer_feedback) AS avg_customer_feedback,  
73        AVG(p.project_completion) AS avg_project_completion  
74 FROM performance_reviews p  
75 JOIN employees e ON p.employee_id = e.employee_id  
76 JOIN departments d ON e.department_id = d.department_id  
77 GROUP BY d.department_name;
```

The results pane shows the output for the last query (Ans 09):

#	department_name	avg_sales_performance	total_reviews
1	Customer Support	74.0	2
2	Development	62.5	2
3	Sales	88.75	4

The bottom status bar indicates: Connected: 00:05 h, MariaDB 10.4.32, Uptime: 00:17 h, Server time: 10:33 PM, 1 tab.