

WINAIM - Screening Assignment - 2023 & 2024 YOP

SQL Assessment

Exercise 1: Employee Management System

Objective: To test the ability to design relational databases, write complex SQL queries, and handle data relationships.

SQL Queries:

1. Write a query to find all employees who have been hired in the last year.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'sakila' and 'sam' databases. The 'sam' database is selected, showing its tables, views, stored procedures, and functions. The main editor window contains two SQL queries. The first query is a simple SELECT statement to find employees hired in the last year. The second query is a more complex JOIN query to calculate the total salary expenditure for each department. The 'Results' tab shows the output of the first query, displaying a table with columns: employee_id, first_name, last_name, department_id, and hire_date. The 'Output' tab shows the execution log for both queries, indicating that the first query returned 0 rows and the second query returned 3 rows.

```
1 -- To Find all employees who have been hired in the last year
2 SELECT *
3 FROM employees
4 WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR;
5
6 -- To Calculate the total salary expenditure for each department
7 SELECT d.department_name, SUM(s.salary) AS total_salary_expenditure
8 FROM employees e
9 JOIN departments d ON e.department_id = d.department_id
```

employee_id	first_name	last_name	department_id	hire_date
3	Moneshwar	V	3	2024-01-05
8	Bala	T	1	2023-07-23
11	Peter	A	4	2024-02-20

employees 2 x

Schema: sam

Output

#	Time	Action	Message	Duration / Fetch
1	21:13:08	SELECT * FROM employees WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR LIMIT ...	0 row(s) returned	0.015 sec / 0.000 sec
2	21:13:57	SELECT * FROM employees WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR LIMIT ...	3 row(s) returned	0.079 sec / 0.000 sec

2. Write a query to calculate the total salary expenditure for each department.

MySQL Workbench

127.0.0.1 (sam1) x unconnected x 127.0.0.1 (sam) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sakila

sam

Tables

Views

Stored Procedures

Functions

Results

WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR;

-- To Calculate the total salary expenditure for each department

SELECT d.department_name, SUM(s.salary) AS total_salary_expenditure

FROM employees e

JOIN departments d ON e.department_id = d.department_id

JOIN salaries s ON e.employee_id = s.employee_id

GROUP BY d.department_name;

Result Grid

department_name	total_salary_expenditure
HR	101000.00
Finance	118000.00
IT	137000.00
Marketing	156000.00
Research and Development	55000.00

Information

Schema: sam

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	21:13:08	SELECT * FROM employees WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR LIMIT ...	0 row(s) returned	0.015 sec / 0.000 sec
2	21:13:57	SELECT * FROM employees WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR LIMIT ...	3 row(s) returned	0.079 sec / 0.000 sec
3	21:14:22	SELECT d.department_name, SUM(s.salary) AS total_salary_expenditure FROM employees e ...	7 row(s) returned	0.078 sec / 0.000 sec

Object Info Session

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

3. Write a query to find the top 5 highest-paid employees along with their department names.

MySQL Workbench

127.0.0.1 (sam1) x unconnected x 127.0.0.1 (sam) x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sakila

sam

Tables

Views

Stored Procedures

Functions

Results

FROM employees e

JOIN departments d ON e.department_id = d.department_id

JOIN (

SELECT employee_id, MAX(salary) AS salary

FROM salaries

GROUP BY employee_id

) s ON e.employee_id = s.employee_id

ORDER BY s.salary DESC

LIMIT 5;

Result Grid

department_name	total_salary_expenditure
HR	101000.00
Finance	118000.00
IT	137000.00
Marketing	156000.00
Research and Development	55000.00

Information

Schema: sam

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	21:13:08	SELECT * FROM employees WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR LIMIT ...	0 row(s) returned	0.015 sec / 0.000 sec
2	21:13:57	SELECT * FROM employees WHERE hire_date >= CURDATE() - INTERVAL 1 YEAR LIMIT ...	3 row(s) returned	0.079 sec / 0.000 sec
3	21:14:22	SELECT d.department_name, SUM(s.salary) AS total_salary_expenditure FROM employees e ...	7 row(s) returned	0.078 sec / 0.000 sec

Object Info Session

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.