

## SQL Exercise 2: Aggregate Functions & Grouping

## Dataset: Employees

+  
🏠  
🔍  
📺  
📄  
☁️  
+  
⚙️  
📋

Load sample data with SQL...

2025-04-14 7:47pm

2025-04-15 3:58pm

2025-04-15 5:04pm

2025-04-19 7:27pm

2025-04-19 7:33pm

2025-04-19 10:29pm

2025-04-21 7:43pm

2025-04-03 4:34pm

+ ▼

Databases Worksheets

Search objects ↻

BRIGHTLIGHT\_DB  
CUSTOMERS  
CUSTOMERS\_LARGE  
EMPLOYEE\_DB  
ORDERS  
PRODUCTS  
RETAIL\_SALES  
SALES  
SNOWFLAKE  
SNOWFLAKE\_SAMPLE\_DATA  
SQL\_FUNDAMENTALS

SQL\_FUNDAMENTALS.PUBLIC ⌵ Settings ⌵

```
1 SELECT *
2
3 FROM
4     "SQL_FUNDAMENTALS"."PUBLIC"."EMPLOYEES"
5
```

Code Versions 🔍

Share ➡️ ▼

Results Chart

# ID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY	HIRE_DATE	CITY
1	John	Doe	IT	55 000	2018-06-15	New York
2	Jane	Smith	HR	48 000	2019-07-20	Chicago
3	Mike	Johnson	Finance	60 000	2017-09-30	Los Angeles
4	Sarah	Brown	IT	53 000	2021-03-25	New York
5	David	White	Marketing	52 000	2016-04-10	San Francisco
6	Emily	Davis	IT	62 000	2015-02-14	Chicago
7	Robert	Wilson	Finance	59 000	2019-10-01	Houston
8	Jessica	Moore	HR	51 000	2018-05-22	Los Angeles
9	Daniel	Clark	Marketing	53 000	2022-06-01	Chicago
10	Laura	Hall	IT	50 000	2020-08-10	San Francisco

Query Details ⋮

Query duration 680ms

Rows 10

Query ID 01bbdb89-0000-f5ff-Q...

Show more ▾

ID #

1 10

FIRST\_NAME A

100% filled

L Magongoa

## Question 1 – COUNT () Function

Write a SQL query to find the **total number of employees** in the company.

The screenshot shows a SQL query editor interface. The top bar displays a timeline of recent queries. The left sidebar shows a list of databases, with 'SQL\_FUNDAMENTALS' selected. The main editor area contains the following SQL query:

```
1 SELECT COUNT(*) AS total_number_of_employees
2
3 FROM
4   "SQL_FUNDAMENTALS"."PUBLIC"."EMPLOYEES"
5
```

Below the query editor, the 'Results' tab is active, showing a single row of results:

#	TOTAL_NUMBER_OF_EMPLOYEES
1	10

On the right side, the 'Query Details' panel provides additional information:

- Query duration: 123ms
- Rows: 1
- Query ID: 01bbdb8c-0000-f5f6-0...
- Show more

At the bottom right, a progress bar indicates that the query is 100% filled.

## Question 2 – SUM() Function

Find the total salary paid to all employees in the IT department.

The screenshot displays the Snowflake web interface. At the top, a timeline shows several queries executed on 2025-04-22 at 1:18pm. The active query is 'ACCOUNTADMIN' in the 'COMPUTE\_WH (X-Small)' warehouse. The left sidebar shows a list of databases, with 'EMPLOYEES\_DB' selected. The main editor shows the following SQL query:

```
1 SELECT
2   sum(salary)AS Total_IT_Salary,
3
4 FROM
5   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
6
7 WHERE department = 'IT';
```

Below the query editor, the 'Results' tab is active, showing a single row of results:

#	TOTAL_IT_SALARY
1	220000

On the right, the 'Query Details' panel provides additional information:

- Query duration: 51ms
- Rows: 1
- Query ID: 01bbdbac-0000-15f3-0...
- Show more

At the bottom right, a visual representation of the result is shown as a bar chart labeled 'TOTAL\_IT\_SALARY' with a value of 220000 and a 100% filled bar.

### Question 3 – AVG() Function

Calculate the average salary of employees in the HR department.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 SELECT
2   AVG(salary) AS AVG_HR_SALARY,
3
4 FROM
5   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
6
7 WHERE department = 'HR';
```

The results pane shows the output of the query, which is a single row with the average salary of employees in the HR department.

#	AVG_HR_SALARY
1	49500.000000

## Question 4 – MIN() and MAX() FUNCTIONS

Find the highest and lowest salary in the company.

The screenshot shows the Snowflake SQL interface. The top bar displays several tabs with timestamps, including '2025-03-25 8:25pm', 'Load sample data with SQ...', '2025-04-15 3:58pm', '2025-04-15 5:04pm', '2025-04-21 7:43pm', '2025-04-03 4:34pm', '2025-04-22 1:11pm', and '2025-04-22 1:18pm'. The left sidebar shows a list of databases and schemas, including BRIGHTLIGHT\_DB, CUSTOMERS, CUSTOMERS\_LARGE, EMPLOYEES\_DB, ORDERS, PRODUCTS, RETAIL\_SALES, SALES, SNOWFLAKE, SNOWFLAKE\_SAMPLE\_DATA, and SQL\_FUNDAMENTALS. The main query editor shows the following SQL code:

```
1 SELECT
2   MIN(SALARY) AS LOWEST_SALARY,
3   MAX(salary) AS HIGHEST_SALARY,
4
5 FROM
6   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
7
8
```

The query results are displayed in a table with two columns: # LOWEST\_SALARY and # HIGHEST\_SALARY. The first row shows the values 48000 and 62000 respectively.

	# LOWEST_SALARY	# HIGHEST_SALARY
1	48000	62000

The right sidebar shows the Query Details section, including the Query duration (24ms), Rows (1), and Query ID (01bbdbb6-0000-f5f6-0...). It also displays progress bars for the columns LOWEST\_SALARY and HIGHEST\_SALARY, both showing 100% filled.

## Question 5 – GROUP BY Statement

Group employees by department and display the total salary paid in each department.

The screenshot displays the Snowflake web interface. At the top, a timeline shows several queries executed on 2025-04-22 at 1:18pm. The active query is 'ACCOUNTADMIN - COMPUTE\_WH (X-Small)'. The left sidebar shows a list of databases, with 'EMPLOYEES\_DB' selected. The main editor shows the following SQL query:

```
1 SELECT
2   SUM(SALARY) AS TOTAL_SALARY,
3   department
4
5 FROM
6   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
7 GROUP BY department;
```

Below the query editor, the 'Results' tab is active, showing a table with 4 rows and 2 columns: '# TOTAL\_SALARY' and 'DEPARTMENT'.

	# TOTAL_SALARY	DEPARTMENT
1	220000	IT
2	99000	HR
3	119000	Finance
4	105000	Marketing

On the right side, the 'Query Details' panel shows a query duration of 55ms, 4 rows returned, and a query ID of 01bbdbb8-0000-f5f6-0... Below this, two visualizations are shown: a bar chart for 'TOTAL\_SALARY' and a gauge chart for 'DEPARTMENT'.

L Magongoa

## Question 6 – GROUP BY and COUNT()

Count how many employees work in each city.

The screenshot shows a Snowflake SQL query editor interface. The top bar displays several tabs with timestamps, including '2025-03-25 8:25pm', 'Load sample data with SQ...', '2025-04-14 7:47pm', '2025-04-15 3:58pm', '2025-04-15 5:04pm', '2025-04-19 7:27pm', '2025-04-19 7:33pm', and '2025-04-19 10:29pm'. The left sidebar shows a 'Databases' tab with a search bar and a list of databases: BRIGHTLIGHT\_DB, CUSTOMERS, CUSTOMERS\_LARGE, EMPLOYEE\_DB, ORDERS, PRODUCTS, SALES, SNOWFLAKE, SNOWFLAKE\_SAMPLE\_DATA, and SQL\_FUNDAMENTALS. The main editor area shows a SQL query: 

```
1 SELECT
2   city,
3   COUNT(*) AS total_EMPLOYEES
4
5 FROM
6   "SQL_FUNDAMENTALS"."PUBLIC"."EMPLOYEES"
7 GROUP BY city;
```

 The bottom section displays the 'Results' tab with a table showing the query output. The table has two columns: 'CITY' and 'TOTAL\_EMPLOYEES'. The data rows are: New York (2), Chicago (3), Los Angeles (2), San Francisco (2), and Houston (1). To the right of the table, there is a 'Query Details' panel showing 'Query duration: 50ms', 'Rows: 5', and 'Query ID: 01bbcd03-0000-f445-0...'. Below the table, there are two small bar charts: one for 'CITY' showing a 100% filled bar, and another for 'TOTAL\_EMPLOYEES' showing a bar with a value of 3.

	CITY	TOTAL_EMPLOYEES
1	New York	2
2	Chicago	3
3	Los Angeles	2
4	San Francisco	2
5	Houston	1

## Question 7. GROUP BY and ORDER BY

Write a SQL query to group employees by department, calculate the average salary in each department, and order the results in descending order of average salary.

The screenshot shows a SQL IDE interface with a query editor and a results panel. The query is as follows:

```
1 SELECT
2   avg(SALARY) AS AVG_SALARY,
3   department
4
5 FROM
6   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
7
8 GROUP BY DEPARTMENT
9 ORDER BY AVG (SALARY) DESC;
10
```

The results panel displays the following data:

#	AVG_SALARY	DEPARTMENT
1	59500.000000	Finance
2	55000.000000	IT
3	52500.000000	Marketing
4	49500.000000	HR

Query Details:

- Query duration: 298ms
- Rows: 4
- Query ID: 01bbdbd0-0000-f5f3-0...
- Show more

Visualizations:

- AVG\_SALARY: A bar chart showing the average salary for each department, with values ranging from 49500 to 59500.
- DEPARTMENT: A bar chart showing the count of rows for each department, with 100% filled.



## Question 8. HAVING Clause

Write a SQL query to find departments where the total salary paid exceeds 100,000. (Use GROUP BY and HAVING)

The screenshot shows a SQL query editor interface. The query is as follows:

```
1 SELECT
2   sum(SALARY) AS total_SALARY,
3   department
4
5 FROM
6   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
7
8 GROUP BY DEPARTMENT
9 HAVING SUM(SALARY)>100000;
```

The results table shows the following data:

	# TOTAL_SALARY	DEPARTMENT
1	220000	IT
2	119000	Finance
3	105000	Marketing

Query Details:

- Query duration: 198ms
- Rows: 3
- Query ID: 01bbdbd4-0000-f5f6-0...

Visualizations:

- TOTAL\_SALARY**: A bar chart showing the total salary for each department. The x-axis ranges from 105000 to 220000. The bars for IT and Finance are visible.
- DEPARTMENT**: A gauge chart showing the percentage of rows for each department. The gauge is 100% filled.

## Question 9. Combining GROUP BY, HAVING, and ORDER BY

Write a SQL query to list cities where more than one employee works, ordered by the number of employees in descending order.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query is as follows:

```
1 SELECT
2   CITY,
3   COUNT(*) AS employee_count
4
5
6 FROM
7   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
8
9
10 GROUP BY CITY
11 HAVING COUNT (*) > 1
12 ORDER BY employee_count DESC;
```

The results pane displays a table with two columns: CITY and EMPLOYEE\_COUNT. The data is as follows:

	CITY	EMPLOYEE_COUNT
1	Chicago	3
2	New York	2
3	Los Angeles	2
4	San Francisco	2

Query Details:

- Query duration: 230ms
- Rows: 4
- Query ID: 01bbdbee-0000-f5f6-0...

Visualizations:

- CITY: A horizontal bar chart showing 100% filled for all cities.
- EMPLOYEE\_COUNT: A horizontal bar chart showing the count for each city (2 for New York, Los Angeles, and San Francisco; 3 for Chicago).

## Question 10. Combining Aggregate Functions

Write a SQL query to find the department with the highest average salary.

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
1 SELECT
2   department,
3   AVG(salary) AS avg_salary
4
5 FROM
6   "EMPLOYEES_DB"."PUBLIC"."EMPLOYEES"
7
8 GROUP BY department
9
10 ORDER BY avg_salary DESC
11 LIMIT 1;
```

The results pane displays a table with two columns: DEPARTMENT and AVG\_SALARY. The first row shows 'Finance' with an average salary of 59500.

	DEPARTMENT	AVG_SALARY
1	Finance	59500

Below the table, a summary section shows the following statistics:

	Sum	Average
100% filled	59500	59500