

SECTION 21

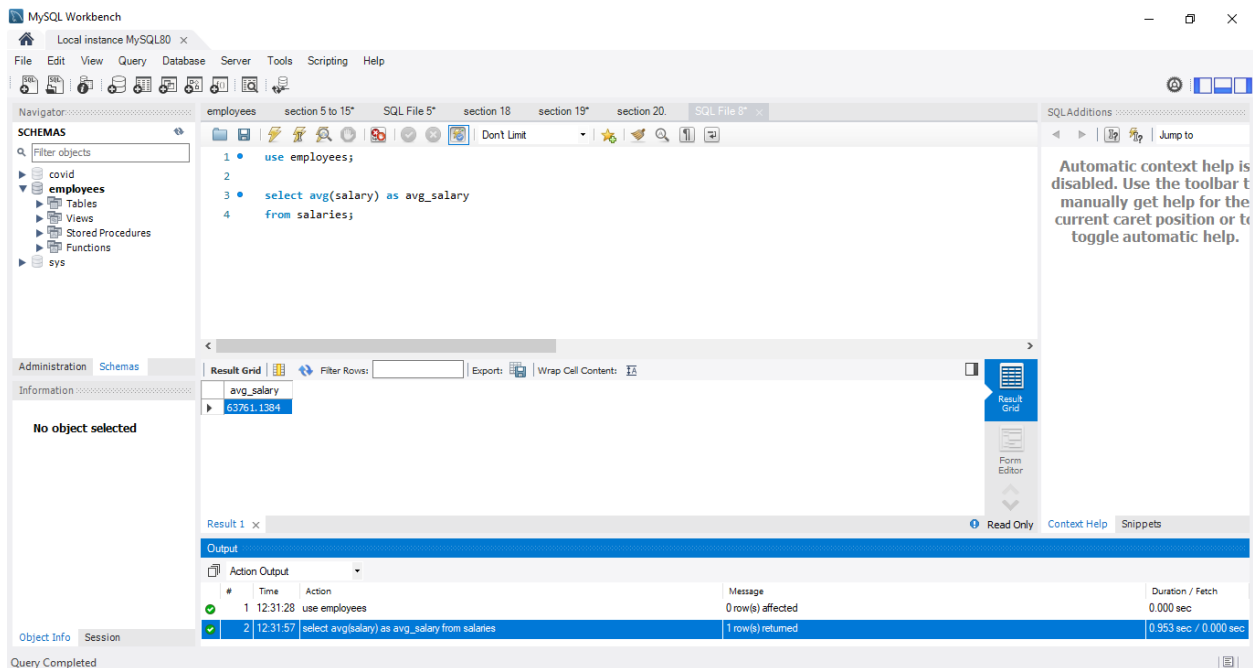
In this lesson, we explored Common Table Expressions (CTEs).

Our first question is how many salary contracts signed by female employees have been valued above the all-time average contract salary value of the company?

To obtain the answer, we need two datasets

1. A list of all contracts signed by female employees from the company's history
2. A single all-time average value

So, after running our query for 2, we got:



MySQL Workbench interface showing a query execution. The query is as follows:

```
1 use employees;
2
3 select avg(salary) as avg_salary
4 from salaries;
```

The result grid displays the average salary value: 63761.1384.

The output pane shows the execution steps and their durations:

#	Time	Action	Message	Duration / Fetch
1	12:31:28	use employees	0 row(s) affected	0.000 sec
2	12:31:57	select avg(salary) as avg_salary from salaries	1 row(s) returned	0.953 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: employees section 5 to 15* SQL File 5* section 18 section 19* section 20. SQL File 8*

SCHMAS

Filter objects

covid

employees

Tables

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

No object selected

Result Grid

no_f_salary_above_avg	total_no_of_salary_contracts
171179	387475

Result 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	13:38:15	with cte as (select avg(salary) as avg_salary from salaries) select sum(case when s.salary > c.avg_salary then 1 else 0 end) as no_f_salary_above_avg, count(s.salary) as total_no_of_salary_contracts from salaries s join employees e on s.emp_no = e.emp_no and e.gender = 'F'	Error Code: 1630. FUNCTION employees.sum does not exist. Check the 'Function Name Par...	0.281 sec
4	13:40:46	with cte as (select avg(salary) as avg_salary from salaries) select sum(case when s.salary > ...	1 row(s) returned	2.968 sec / 0.000 sec

Object Info Session

Query Completed

Exercise #1:

Use a CTE (a Common Table Expression) and a SUM() function in the SELECT statement in a query to find out how many male employees have never signed a contract with a salary value higher than or equal to the all-time company salary average.

Solution:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: employees section 5 to 15* SQL File 5* section 18 section 19* section 20. SQL File 8*

SCHMAS

Filter objects

covid

employees

Tables

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

No object selected

Result Grid

no_salaries_below_avg	no_of_salary_contracts
324417	579856

Result 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
4	13:40:46	with cte as (select avg(salary) as avg_salary from salaries) select sum(case when s.salary > ...	1 row(s) returned	2.968 sec / 0.000 sec
5	13:47:03	WITH cte AS (SELECT AVG(salary) AS avg_salary FROM salaries) SELECT SUM(CASE W...	1 row(s) returned	3.235 sec / 0.000 sec

Object Info Session

Query Completed

Exercise #2:

Use a CTE (a Common Table Expression) and (at least one) COUNT() function in the SELECT statement of a query to find out how many male employees have never signed a contract with a salary value higher than or equal to the all-time company salary average.

Solution:

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
27
28 WITH cte AS (
29   SELECT AVG(salary) AS avg_salary FROM salaries)
30 SELECT
31   COUNT(CASE WHEN s.salary < c.avg_salary THEN s.salary ELSE NULL END) AS no_salaries_below_avg_w_count,
32   COUNT(s.salary) AS no_of_salary_contracts
33 FROM salaries s
34 JOIN employees e ON s.emp_no = e.emp_no AND e.gender = 'M'
35 JOIN cte c;
```

The Results pane shows the following data:

no_salaries_below_avg_w_count	no_of_salary_contracts
324417	579856

The Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
5	13:47:03	WITH cte AS (SELECT AVG(salary) AS avg_salary FROM salaries) SELECT SUM(CASE W...	1 row(s) returned	3.235 sec / 0.000 sec
6	13:48:36	WITH cte AS (SELECT AVG(salary) AS avg_salary FROM salaries) SELECT COUNT(CASE ...	1 row(s) returned	2.453 sec / 0.000 sec

Exercise #3:

Use MySQL joins (and don't use a Common Table Expression) in a query to find out how many male employees have never signed a contract with a salary value higher than or equal to the all-time company salary average (i.e. to obtain the same result as in the previous exercise).

Solution:

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query in the 'SQL File 6*' tab. The query is as follows:

```

37
38
39 • SELECT
40     SUM(CASE WHEN s.salary < a.avg_salary THEN 1 ELSE 0 END) AS no_salaries_below_avg,
41     COUNT(s.salary) AS no_of_salary_contracts
42 FROM (SELECT AVG(salary) AS avg_salary FROM salaries s) a
43 JOIN salaries s
44 JOIN employees e ON e.emp_no = s.emp_no AND e.gender = 'M';
45
46
47
48

```

The 'Result Grid' shows the following data:

no_salaries_below_avg	no_of_salary_contracts
324417	579856

The 'Output' pane shows the execution details:

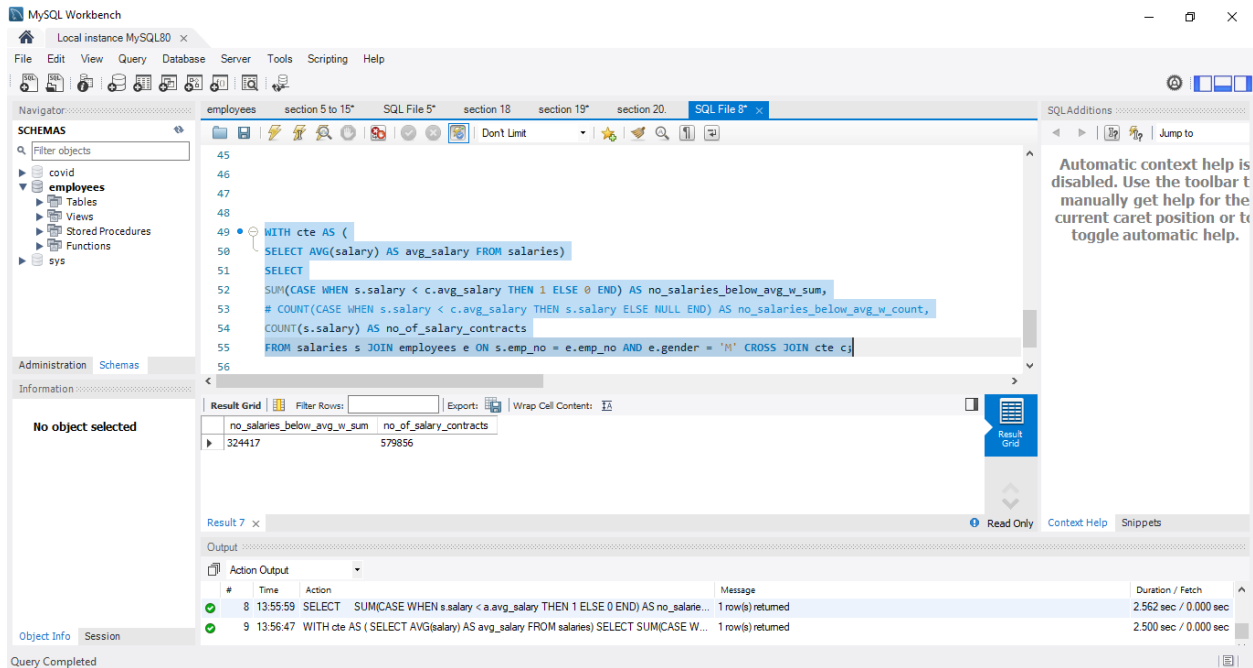
#	Time	Action	Message	Duration / Fetch
7	13:55:15	WITH cte AS (SELECT AVG(salary) AS avg_salary FROM salaries) SELECT COUNT(CASE ...	1 row(s) returned	2.437 sec / 0.000 sec
8	13:55:59	SELECT SUM(CASE WHEN s.salary < a.avg_salary THEN 1 ELSE 0 END) AS no_salaries...	1 row(s) returned	2.562 sec / 0.000 sec

The 'Object Info' pane shows 'No object selected'. The 'Session' pane shows 'Query Completed'.

Exercise #4:

Use a cross join in a query to find out how many male employees have never signed a contract with a salary value higher than or equal to the all-time company salary average (i.e. to obtain the same result as in the previous exercise).

Solution:



Using multiple Subclauses in a WITH clause

Our task is: How many female employees' highest contract salary values were higher than all-time company salary average (across all genders)?

To solve the problem, we will have two sub clauses

1. A CTE computing all-time average
2. A CTE to obtain a list of containing highest contract salary values of all female employees

Then, we will compare the salary values and count the no. of occurrences when the salary value is higher than the average.

So we can get all time average by simple select statement:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: employees section 5 to 15* SQL File 5* section 18 section 19* section 20. section 21* %

SCHEMAS

Filter objects

covid

employees

Tables

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

No object selected

Result Grid

avg_salary
63761.1384

Result 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	12:04:46	select avg(salary) as avg_salary from salaries	1 row(s) returned	0.485 sec / 0.000 sec

Object Info Session

Query Completed

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

Now, for point 2, we will do the following:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: employees section 5 to 15* SQL File 5* section 18 section 19* section 20. section 21* %

SCHEMAS

Filter objects

covid

employees

Tables

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

No object selected

Result Grid

emp_no	highest_salary
10002	72527
10006	60098
10007	80070
10009	94443
10010	80324

Result 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	12:06:30	SELECT AVG(salary) AS avg_salary FROM salaries	1 row(s) returned	0.516 sec / 0.000 sec
3	12:08:44	select s.emp_no, max(s.salary) as highest_salary from salaries s join employees e on e.emp_no = s.emp_no and e.gender = 'F'	40832 row(s) returned	0.109 sec / 2.609 sec

Object Info Session

Query Completed

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

Next, we will implement the CTE statement:

MySQL Workbench interface showing a SQL query and its result. The query is as follows:

```

67 with cte1 as (
68   SELECT AVG(salary) AS avg_salary FROM salaries),
69   cte2 as (
70     select s.emp_no, max(s.salary) as highest_salary
71     from salaries s
72     join employees e on e.emp_no = s.emp_no and e.gender = 'F'
73     group by s.emp_no)
74   select sum(case when c2.highest_salary > c1.avg_salary then 1 else 0 end) as f_highest_salary_above_avg
75   from employees e
76   join cte2 c2 on c2.emp_no = e.emp_no
77   cross join cte1 c1;
78

```

The result grid shows the following data:

Result 4
f_highest_salary_above_avg
24242

The output pane shows the following message:

```

# Time Action Message Duration / Fetch
3 12:08:44 select s.emp_no, max(s.salary) as highest_salary from salaries s join employees e on e.emp_n... 40832 row(s) returned 0.109 sec / 2.609 sec
4 12:22:21 with cte1 as ( SELECT AVG(salary) AS avg_salary FROM salaries), cte2 as ( select s.emp_n... 1 row(s) returned 3.531 sec / 0.000 sec

```

The result we obtain shows that 24,242 female employees have highest valued salary contracts than the all-time company average.

But, we also need to know the total number of female employee contracts stored in the database to interpret our results in a much better way. For this, we will add count statement to our last select query as shown below:

MySQL Workbench interface showing the same SQL query with an additional count statement. The query is as follows:

```

67 with cte1 as (
68   SELECT AVG(salary) AS avg_salary FROM salaries),
69   cte2 as (
70     select s.emp_no, max(s.salary) as highest_salary
71     from salaries s
72     join employees e on e.emp_no = s.emp_no and e.gender = 'F'
73     group by s.emp_no)
74   select sum(case when c2.highest_salary > c1.avg_salary then 1 else 0 end) as f_highest_salary_above_avg,
75         count(e.emp_no) as total_no_of_female_contracts
76   from employees e
77   join cte2 c2 on c2.emp_no = e.emp_no
78   cross join cte1 c1;

```

The result grid shows the following data:

Result 5	
f_highest_salary_above_avg	total_no_of_female_contracts
24242	40832

The output pane shows the following message:

```

# Time Action Message Duration / Fetch
4 12:22:21 with cte1 as ( SELECT AVG(salary) AS avg_salary FROM salaries), cte2 as ( select s.emp_n... 1 row(s) returned 3.531 sec / 0.000 sec
5 12:32:55 with cte1 as ( SELECT AVG(salary) AS avg_salary FROM salaries), cte2 as ( select s.emp_n... 1 row(s) returned 3.515 sec / 0.000 sec

```

Thus, we have 40,832. Now, what if we want to know the result in percentage?

MySQL Workbench interface showing a SQL query and its results. The query is as follows:

```

68 SELECT AVG(salary) AS avg_salary FROM salaries),
69 cte2 as (
70 select s.emp_no, max(s.salary) as highest_salary
71 from salaries s
72 join employees e on e.emp_no = s.emp_no and e.gender = 'F'
73 group by s.emp_no)
74 select sum(case when c2.highest_salary > c1.avg_salary then 1 else 0 end) as f_highest_salary_above_avg,
75 count(e.emp_no) as total_no_of_female_employees,
76 (sum(case when c2.highest_salary > c1.avg_salary then 1 else 0 end) / count(e.emp_no))*100
77 from employees e
78 join cte2 c2 on c2.emp_no = e.emp_no
79 cross join cte1 c1;

```

The result grid shows the following data:

f_highest_salary_above_avg	total_no_of_female_employees	(sum(case when c2.highest_salary > c1.avg_salary then 1 else 0 end) / count(e.emp_no))*100
24242	40832	59.3701

The output pane shows the execution of the query, with messages indicating the number of rows returned for each step.

We can round off our value and make the field easier to read by adding alias as well as using concat.

MySQL Workbench interface showing a modified SQL query and its results. The query is as follows:

```

70 SELECT AVG(salary) AS avg_salary FROM salaries),
71 cte2 as (
72 select s.emp_no, max(s.salary) as highest_salary
73 from salaries s
74 join employees e on e.emp_no = s.emp_no and e.gender = 'F'
75 group by s.emp_no)
76 select sum(case when c2.highest_salary > c1.avg_salary then 1 else 0 end) as f_highest_salary_above_avg,
77 count(e.emp_no) as total_no_of_female_employees,
78 concat(round((sum(case when c2.highest_salary > c1.avg_salary then 1 else 0 end) / count(e.emp_no))*100, 2), '%') as
79 percentage
80 join cte2 c2 on c2.emp_no = e.emp_no

```

The result grid shows the following data:

f_highest_salary_above_avg	total_no_of_female_employees	percentage
24242	40832	59.37%

The output pane shows the execution of the query, with messages indicating the number of rows returned for each step.

Exercise #1:

Use two common table expressions and a SUM() function in the SELECT statement of a query to obtain the number of male employees whose highest salaries have been below the all-time average.

Solution:

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
85 SELECT AVG(salary) AS avg_salary FROM salaries),
86 cte2 AS (
87 SELECT s.emp_no, MAX(s.salary) AS max_salary
88 FROM salaries s
89 JOIN employees e ON e.emp_no = s.emp_no AND e.gender = 'M'
90 GROUP BY s.emp_no)
91 SELECT
92 SUM(CASE WHEN c2.max_salary < c1.avg_salary THEN 1 ELSE 0 END) AS highest_salaries_below_avg
93 FROM employees e
94 JOIN cte2 c2 ON c2.emp_no = e.emp_no
95 JOIN cte1 c1;
```

The Results tab shows the following result set:

highest_salaries_below_avg
24753

The Output tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
13	12:52:26	with cte1 as (SELECT AVG(salary) AS avg_salary FROM salaries), cte2 as (select s.emp_no,...	1 row(s) returned	3.593 sec / 0.000 sec
14	13:00:56	WITH cte1 AS (SELECT AVG(salary) AS avg_salary FROM salaries), cte2 AS (SELECT s.e...	1 row(s) returned	3.657 sec / 0.000 sec

Exercise #2:

Use two common table expressions and a COUNT() function in the SELECT statement of a query to obtain the number of male employees whose highest salaries have been below the all-time average.

Solution:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'employees' selected. The main editor contains a SQL query using Common Table Expressions (CTEs). The query is as follows:

```
99 WITH cte_avg_salary AS (  
100 SELECT AVG(salary) AS avg_salary FROM salaries),  
101 cte_m_highest_salary AS (  
102 SELECT s.emp_no, MAX(s.salary) AS max_salary  
103 FROM salaries s JOIN employees e ON e.emp_no = s.emp_no AND e.gender = 'M'  
104 GROUP BY s.emp_no)  
105 SELECT  
106 COUNT(CASE WHEN c2.max_salary < c1.avg_salary THEN c2.max_salary ELSE NULL END) AS max_salary  
107 FROM employees e  
108 JOIN cte_m_highest_salary c2 ON c2.emp_no = e.emp_no  
109 JOIN cte_avg_salary c1;
```

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

max_salary
24753

The 'Output' pane at the bottom shows the execution log with two messages:

#	Time	Action	Message	Duration / Fetch
14	13:00:56	WITH cte1 AS (SELECT AVG(salary) AS avg_salary FROM salaries), cte2 AS (SELECT s.e...	1 row(s) returned	3.657 sec / 0.000 sec
15	13:04:06	WITH cte_avg_salary AS (SELECT AVG(salary) AS avg_salary FROM salaries), cte_m_high...	1 row(s) returned	4.140 sec / 0.000 sec

Exercise #3:

Does the result from the previous exercise change if you used the Common Table Expression (CTE) for the male employees' highest salaries in a FROM clause, as opposed to in a join?

Solution:

No, the result remained the same.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: employees section 5 to 15* SQL File 5* section 18 section 19* section 20 section 21* x

SCHEMAS

- covid
- employees
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

Administration Schemas

Information

No object selected

```
112 WITH cte_avg_salary AS (  
113 SELECT AVG(salary) AS avg_salary FROM salaries),  
114 cte_m_highest_salary AS (  
115 SELECT s.emp_no, MAX(s.salary) AS max_salary  
116 FROM salaries s JOIN employees e ON e.emp_no = s.emp_no AND e.gender = 'M'  
117 GROUP BY s.emp_no)  
118 SELECT  
119 COUNT(CASE WHEN c2.max_salary < c1.avg_salary THEN c2.max_salary ELSE NULL END) AS max_salary  
120 FROM cte_m_highest_salary c2  
121 JOIN cte_avg_salary c1;  
122
```

Result Grid

max_salary
24753

Result 13 x

Read Only Context Help Snippets

Output

#	Time	Action	Message	Duration / Fetch
15	13:04:06	WITH cte_avg_salary AS (SELECT AVG(salary) AS avg_salary FROM salaries), cte_m_high...	1 row(s) returned	4.140 sec / 0.000 sec
16	13:06:41	WITH cte_avg_salary AS (SELECT AVG(salary) AS avg_salary FROM salaries), cte_m_high...	1 row(s) returned	3.516 sec / 0.000 sec

Object Info Session

Query Completed

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