

1/index.php)
Challenges-2 (/sql-exercises/challenges-1/sql-25-query-challenges.php)
.More to come.. ()

SQL Functions and Group by - Exercises, Practice, Solution

Last update on January 06 2026 12:29:58 (UTC/GMT +8 hours)

This resource offers a total of 125 SQL Aggregate Functions problems for practice. It includes 25 main exercises, each accompanied by solutions, detailed explanations, and four related problems.

[SQL Exercise Book](#)

[An Editor is available at the bottom of the page to write and execute the scripts.]

1. Calculate Total Purchase Amount of All Orders

From the following table, write a SQL query to calculate total purchase amount of all orders. Return total purchase amount.

Sample table: orders



70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

```
sum  
17541.18
```

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-1.php)

2. Calculate Average Purchase Amount of All Orders

From the following table, write a SQL query to calculate the average purchase amount of all orders. Return average purchase amount.

Data Aggregation Tools

Sample table: orders



70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

```
avg
1461.765000000000000000
```

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-2.php)

3. Count the Number of Unique Salespeople

From the following table, write a SQL query that counts the number of unique salespeople. Return number of salespeople.

[SQL Exercise Book](#)

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-3.php)

4. Count the Number of Customers

From the following table, write a SQL query to count the number of customers.
Return number of customers.

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidore	Moscow	200	5007
3001	Brad Guzan	London		5005

Sample Output:

count 8

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-4.php)

5. Count Customers with at Least One Grade

From the following table, write a SQL query to determine the number of customers who received at least one grade for their activity.

SQL Exercise Book

3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidore	Moscow	200	5007
3001	Brad Guzan	London		5005

Sample Output:

count
7

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-5.php)

6. Find Maximum Purchase Amount

From the following table, write a SQL query to find the maximum purchase amount.

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-6.php)

7. Find Minimum Purchase Amount

From the following table, write a SQL query to find the minimum purchase amount.

[SQL Exercise Book](#)

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

```
min
65.26
```

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-7.php)

8. Find Highest Customer Grade in Each City

From the following table, write a SQL query to find the highest grade of the customers in each city. Return city, maximum grade.

3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005



Sample Output:

city	max
London	300
Paris	300
New York	200
California	200
Berlin	100
Moscow	200

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-8.php)

PingCAP Powered By TiDB

Download The White Paper

Compare TiDB and MySQL with our expert guide.



Learn More

9. Find Highest Purchase Amount Ordered by Each Customer



ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

customer_id	max
3007	2400.60
3008	250.45
3002	5760.00
3001	270.65
3009	2480.40
3004	1983.43
3003	75.29
3005	948.50

Click me to see the solution with visual presentation ([sql-aggregate-function-exercise-9.php](#))

10. Highest Purchase by Customer on Date

From the following table, write a SQL query to find the highest purchase amount ordered by each customer on a particular date. Return, order date and highest purchase amount.

Sample table: orders

70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

customer_id	ord_date	max
3002	2012-10-05	65.26
3003	2012-08-17	75.29
3005	2012-10-05	150.50
3007	2012-07-27	2400.60
3009	2012-08-17	110.50
3001	2012-09-10	270.65
3002	2012-09-10	5760.00
3005	2012-09-10	948.50
3009	2012-10-10	2480.40
3008	2012-06-27	250.45
3004	2012-10-10	1983.43
3002	2012-04-25	3045.60

Click me to see the solution with visual presentation ([sql-aggregate-function-exercise-10.php](#))

11. Highest Purchase Amount by Salesperson on '2012-08-17'

From the following table, write a SQL query to determine the highest purchase amount made by each salesperson on '2012-08-17'. Return salesperson ID, purchase amount

[SQL Exercise Book](#)

Sample table: orders

70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

salesman_id	max
5003	110.50
5007	75.29

Click me to see the solution with visual presentation ([sql-aggregate-function-exercise-11.php](#))

12. Highest Purchase Amount by Customer on Specific Date

From the following table, write a SQL query to find the highest order (purchase) amount by each customer on a particular order date. Filter the result by highest order (purchase) amount above 2000.00. Return customer id, order date and maximum purchase amount.

Sample table: orders



70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

customer_id	ord_date	max
3007	2012-07-27	2400.60
3002	2012-09-10	5760.00
3009	2012-10-10	2480.40
3002	2012-04-25	3045.60

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-12.php)

13. Max Purchase Amount by Customer and Order Date (2000-6000)

From the following table, write a SQL query to find the maximum order (purchase) amount in the range 2000 - 6000 (Begin and end values are included.) by combination of each customer and order date. Return customer id, order date and maximum purchase amount.

[SQL Exercise Book](#)

Sample table: orders

70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

customer_id	ord_date	max
3007	2012-07-27	2400.60
3002	2012-09-10	5760.00
3009	2012-10-10	2480.40
3002	2012-04-25	3045.60

Discover related topics

[Sql Aggregate Functions Group by Multiple Columns](#) >

[Group by Having Condition in Sql](#) >

[SQL Aggregate Functions Examples](#) >

[How to Aggregate Data in Sql](#) >

[Sql Error Not in Aggregate Function Or Group by Clause](#) >

Click me to see the solution with visual presentation ([sql-aggregate-function-exercise-13.php](#))



Filter the rows for maximum order (purchase) amount is either 2000, 3000, 5760, 6000. Return customer id, order date and maximum purchase amount.

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

customer_id	ord_date	max
3002	2012-09-10	5760.00

Click me to see the solution with visual presentation ([sql-aggregate-function-exercise-14.php](#))

15. Max Purchase Amount for Customers (ID Range 3002-3007)

From the following table, write a SQL query to determine the maximum order amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Return customer id and maximum purchase amount.

[SQL Exercise Book](#)

Sample table: orders

70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

customer_id	max
3002	5760.00
3007	2400.60
3004	1983.43
3003	75.29
3005	948.50

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-15.php)

16. Max Purchase > 1000 for Customers (ID 3002-3007)

From the following table, write a SQL query to find the maximum order (purchase) amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Filter the rows for maximum order (purchase) amount is higher than 1000. Return customer id and maximum purchase amount.

Sample table: orders

70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

customer_id	max
3002	5760.00
3007	2400.60
3004	1983.43

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-16.php)

17. Max Purchase by Salesperson (ID 5003-5008)

From the following table, write a SQL query to determine the maximum order (purchase) amount generated by each salesperson. Filter the rows for the salesperson ID is in the range 5003 and 5008 (Begin and end values are included.). Return salesperson id and maximum purchase amount.

[SQL Exercise Book](#)

Sample table: orders

70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

salesman_id	max
5005	270.65
5003	2480.40
5007	75.29
5006	1983.43

Click me to see the solution with visual presentation (sql-aggregate-function-exercise-17.php)

18. Count Orders on '2012-08-17'

From the following table, write a SQL query to count all the orders generated on '2012-08-17'. Return number of orders.

Sample table: orders

70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

```
count
2
```

Click me to see the solution with visual presentation ([sql-aggregate-function-exercise-18.php](#))

19. Count Salespeople in a City

From the following table, write a SQL query to count the number of salespeople in a city. Return number of salespeople.

[SQL Exercise Book](#)

Sample table: salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

Sample Output:

20. Count Orders by Order Date and Salesperson

From the following table, write a SQL query to count the number of orders based on the combination of each order date and salesperson. Return order date, salesperson id.

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample Output:

ord_date	salesman_id	count
2012-07-27	5001	1
2012-08-17	5007	1
2012-04-25	5001	1
2012-09-10	5002	1
2012-10-05	5002	1
2012-10-10	5003	1
2012-09-10	5005	1
2012-08-17	5003	1
2012-06-27	5002	1
2012-09-10	5001	1
2012-10-05	5001	1
2012-10-10	5006	1

From the following table, write a SQL query to calculate the average product price. Return average product price.

SQL Exercise Book

Sample table: item_mast

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
101	Mother Board	3200.00	15
102	Key Board	450.00	16
103	ZIP drive	250.00	14
104	Speaker	550.00	16
105	Monitor	5000.00	11
106	DVD drive	900.00	12
107	CD drive	800.00	12
108	Printer	2600.00	13
109	Refill cartridge	350.00	13
110	Mouse	250.00	12

Sample Output:

Average Price
1435.0000000000000000

Click me to see the solution with results (sql-aggregate-function-exercise-21.php)

22. Count Products with Price >= 350

From the following table, write a SQL query to count the number of products whose price are higher than or equal to 350. Return number of products.

Sample table: item_mast

105 Monitor	5000.00	11
106 DVD drive	900.00	12
107 CD drive	800.00	12
108 Printer	2600.00	13
109 Refill cartridge	350.00	13
110 Mouse	250.00	12

Sample Output:

Number of Products
8

Click me to see the solution with results (sql-aggregate-function-exercise-22.php)

23. Average Price for Unique Companies

From the following table, write a SQL query to compute the average price for unique companies. Return average price and company id.

[SQL Exercise Book](#)

Sample table: item_mast



105 Monitor	5000.00	11
106 DVD drive	900.00	12
107 CD drive	800.00	12
108 Printer	2600.00	13
109 Refill cartridge	350.00	13
110 Mouse	250.00	12

Sample Output:

Average Price	Company ID
250.000000000000000000	14
650.000000000000000000	12
3200.000000000000000000	15
5000.000000000000000000	11
1475.000000000000000000	13
500.000000000000000000	16

Click me to see the solution with results (sql-aggregate-function-exercise-23.php)

24. Sum of Allotment Amount for All Departments

From the following table, write a SQL query to compute the sum of the allotment amount of all departments. Return sum of the allotment amount.

Sample table: emp_department

DPT_CODE	DPT_NAME	DPT_ALLOTMENT
57	IT	65000
63	Finance	15000
47	HR	240000
27	RD	55000
89	QC	75000