

## SQL Bootcamp- Day 5 Assignment

Query

Query History

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

▼

/\* 1. GROUP BY with WHERE - Orders by Year and Quarter.  
Display order year, quarter, order count, avg freight cost  
only for those orders where freight cost > 100 \*/  
select \* from orders;  
select \* from order\_details;  
  
select  
    extract(year from order\_date) as order\_year,  
    extract(quarter from order\_date) as order\_quarter,  
    count(\*) as order\_count,  
    round(cast(avg(freight) as numeric), 2) as average\_freight\_cost  
from  
    orders  
where  
    freight > 100  
group by  
    order\_year,  
    order\_quarter  
order by  
    order\_year,  
    order\_quarter;

Data Output

Messages

Notifications

≡+

▼

▼

SQL

	order_year numeric	order_quarter numeric	order_count bigint	average_freight_cost numeric
1	1996	3	12	163.84
2	1996	4	22	203.85
3	1997	1	14	232.65
4	1997	2	21	269.55
5	1997	3	26	236.44
6	1997	4	33	206.75
7	1998	1	43	245.48
8	1998	2	16	292.79

Total rows: 8

Query complete 00:00:00.102

```

26  /* 2. GROUP BY with HAVING - High Volume Ship Regions.
27  Display ship region, no of orders in each region,
28  min and max freight cost. Filter regions where no of orders >= 5*/
29
30  select ship_region, count(*) as num_orders, min(freight), max(freight)
31  from orders
32  group by ship_region
33  having count(*) >= 5
34  order by num_orders desc;
35
36

```

Data Output Messages Notifications

	ship_region character varying (15)	num_orders bigint	min real	max real
1	[null]	507	0.02	1007.64
2	SP	49	0.14	890.78
3	RJ	34	2.27	193.37
4	ID	31	8.19	830.75
5	OR	28	0.2	719.78
6	Co. Cork	19	16.74	603.54
7	WA	19	4.56	606.19
8	Táchira	18	2.08	210.19
9	NM	18	8.53	708.95
10	BC	17	0.94	243.73
11	Lara	14	0.12	163.97
12	Québec	13	4.07	379.13
13	Essex	13	3.04	146.32
14	Nueva Esparta	12	2.71	158.44
15	Isle of Wight	10	0.9	154.72
16	AK	10	5.24	257.62
17	WY	9	4.34	195.68

Total rows: 17 Query complete 00:00:00.087

```

37  /* 3. Get all title designations across employees and customers. (UNION & UNION ALL)*/
38  select * from customers;
39  select * from employees;
40
41  select title from employees
42  UNION
43  select contact_title from customers;
44
45  select title from employees
46  UNION ALL
47  select contact_title from customers;
48

```

Data Output Messages Notifications




	title character varying (30)
1	Owner
2	Sales Associate
3	Sales Agent
4	Inside Sales Coordinator
5	Marketing Manager
6	Assistant Sales Agent
7	Assistant Sales Representative
8	Accounting Manager
9	Vice President, Sales
10	Sales Manager
11	Sales Representative
12	Marketing Assistant
13	Owner/Marketing Assistant
14	Order Administrator

Total rows: 14 Query complete 00:00:00.078

Query Query History

```
37  /* 3. Get all title designations across employees and customers. (UNION & UNION ALL) */
38  select * from customers;
39  select * from employees;
40
41  select title from employees
42  UNION
43  select contact_title from customers;
44
45  select title from employees
46  UNION ALL
47  select contact_title from customers;
48
```

Data Output   Messages   Notifications

			SQL	
	title			
	character varying (30)			
1	Sales Representative			
2	Vice President, Sales			
3	Sales Representative			
4	Sales Representative			
5	Sales Manager			
6	Sales Representative			
7	Sales Representative			
8	Inside Sales Coordinator			
9	Sales Representative			
10	Sales Representative			
11	Owner			
12	Owner			
13	Sales Representative			
14	Order Administrator			
15	Sales Representative			
16	Marketing Manager			
17	Owner			
18	Owner			
19	Accounting Manager			
20	Sales Representative			
21	Sales Agent			
22	Marketing Manager			
23	Owner			
24	Sales Associate			
25	Sales Representative			
26	Order Administrator			
27	Owner			
28	Sales Agent			
29	Sales Manager			
Total rows: 100			Query complete 00:00:00.078	

Query

Query History

49

▼

/\* 4. Find categories that have both discontinued and in-stock products.  
(Display category\_id, instock means units\_in\_stock > 0, Intersect)\*/  
--select \* from categories;  
select \* from products;  
  
select category\_id, units\_in\_stock as instock  
from products  
where units\_in\_stock >0  
  
INTERSECT  
  
select category\_id, units\_in\_stock as instock  
from products  
where discontinued = 0  
order by instock;

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	category_id smallint	instock smallint
1	3	3
2	7	4
3	2	4
4	8	5
5	3	6
6	2	6
7	4	9
8	8	10
9	3	10
10	8	11
11	2	13
12	4	14
13	7	15
14	1	15
15	3	15
16	3	17
17	1	17
18	4	19
19	1	20
20	7	20
21	5	21
22	6	21
23	4	22
24	5	22
25	2	24

Total rows: 63

Query complete 00:00:00.110

[Query](#) [Query History](#)

```
65
66  /* 5. Find orders that have no discounted items (Display the order_id, EXCEPT)*/
67  select * from orders;
68  select * from order_details;
69
70  select order_id
71  from orders
72
73  Except
74
75  select distinct order_id
76  from order_details
77  where discount = 1;
78
```

Data Output Messages Notifications

	order_id smallint
1	11038
2	10782
3	10725
4	10423
5	10518
6	10356
7	10963
8	10596
9	10282
10	10658
11	10283
12	10579
13	10693
14	10896
15	10660
16	10253
17	10425
18	10774
19	10615
20	10514
21	10850
22	10276
23	10978
24	10532
25	10340
26	10256
27	10889
28	10294

Total rows: 830      Query complete 00:00:00.080