

SQL Assignment 1

Note : Queries to look at the exact table have been included to just understand the table and columns available.

Q1. Do we have actors in the actor table that share the full name and if yes display those shared names.

Answer: Yes, we do. The shared name is Susan Davis.

1. Required query

SELECT first_name, last_name FROM actor

GROUP BY first_name, last_name

HAVING COUNT(actor_id) >= 2

The screenshot shows a SQL query editor with two queries. The first query is highlighted in blue and returns the result 'Susan Davis'. The second query is highlighted in purple and returns the result 'Susan Davis'.

```
1 SELECT first_name, last_name FROM actor
2 GROUP BY first_name, last_name
3 HAVING COUNT(actor_id) >= 2
4
5 SELECT * FROM actor
6 WHERE first_name = 'Susan' AND last_name = 'Davis';
7
8 SELECT first_name, last_name, COUNT(actor_id) FROM actor
9 GROUP BY first_name, last_name
10 HAVING COUNT(actor_id) >= 2;
11
12
13
14
```

	first_name character varying (45)	last_name character varying (45)
1	Susan	Davis

2. Just validating the findings.

SELECT * FROM actor

WHERE first_name = 'Susan' AND last_name = 'Davis';

```

5 SELECT * FROM actor
6 WHERE first_name = 'Susan' AND last_name = 'Davis';
7
8 SELECT first_name, last_name, COUNT(actor_id) FROM actor
9 GROUP BY first_name, last_name
10 HAVING COUNT(actor_id) >= 2;
11
12
13
14

```

ins

ws

	actor_id [PK] integer	first_name character varying (45)	last_name character varying (45)	last_update timestamp without time zone
1	101	Susan	Davis	2013-05-26 14:47:57.62
2	110	Susan	Davis	2013-05-26 14:47:57.62

3. A more organised answer.

SELECT first_name, last_name, COUNT(actor_id) FROM actor

GROUP BY first_name, last_name

HAVING COUNT(actor_id) >= 2;

```

8 SELECT first_name, last_name, COUNT(actor_id) FROM actor
9 GROUP BY first_name, last_name
10 HAVING COUNT(actor_id) >= 2;
11
12
13
14

```

s

s

	first_name character varying (45)	last_name character varying (45)	count bigint
1	Susan	Davis	2

Q2. Return the customer IDs of customers who have spent at least \$110 with the staff member who has an ID of 2.

```
SELECT * FROM payment;
```

2. Required answer

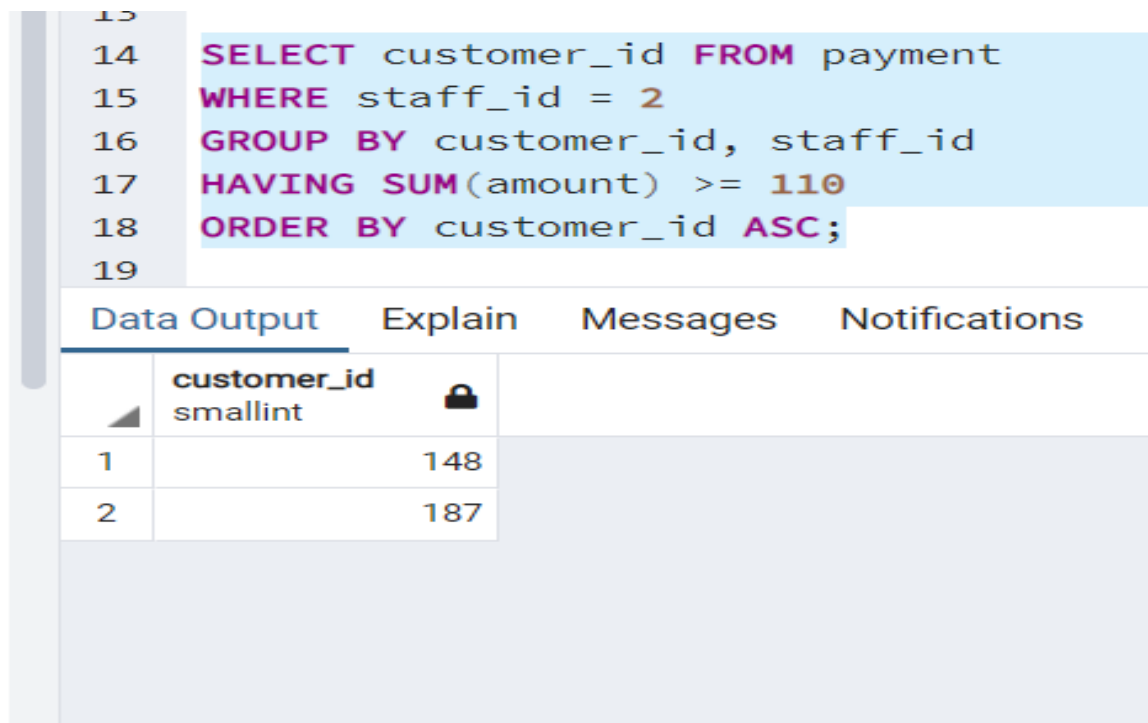
```
SELECT customer_id FROM payment
```

```
WHERE staff_id = 2
```

```
GROUP BY customer_id, staff_id
```

```
HAVING SUM(amount) >= 110
```

```
ORDER BY customer_id ASC;
```



```
13
14 SELECT customer_id FROM payment
15 WHERE staff_id = 2
16 GROUP BY customer_id, staff_id
17 HAVING SUM(amount) >= 110
18 ORDER BY customer_id ASC;
19
```

	customer_id	smallint
1		148
2		187

3. A better, organised answer.

```
SELECT customer_id, staff_id, SUM(amount) FROM payment
```

```
WHERE staff_id = 2
```

```
GROUP BY customer_id, staff_id
```

```
HAVING SUM(amount) >= 110
```

```
ORDER BY customer_id ASC;
```

```

19
20 SELECT customer_id,staff_id, SUM(amount) FROM payment
21 WHERE staff_id = 2
22 GROUP BY customer_id, staff_id
23 HAVING SUM(amount) >= 110
24 ORDER BY customer_id ASC;
25

```

Data Output Explain Messages Notifications

	customer_id smallint	staff_id smallint	sum numeric
1	148	2	110.78
2	187	2	110.81

Q3. How many films begin with the letter J?

Answer: 20 Films.

SELECT * FROM film;

SELECT COUNT(film_id) FROM film

WHERE title ILIKE 'j%';

```

28 SELECT COUNT(film_id) FROM film
29 WHERE title ILIKE 'J%';
30

```

Data Output Explain Messages Notifications

	count bigint
1	20

Q4. What customer has the highest customer ID number whose name starts with an 'E' and has an address ID lower than 500?

Answer : Eddie Tomlin whose customer_id is 434 fulfils the given criteria.

```
SELECT * FROM customer;
```

```
SELECT customer_id, first_name, last_name FROM customer
```

```
WHERE first_name LIKE 'E%' AND address_id < 500
```

```
ORDER BY customer_id DESC
```

```
LIMIT 1;
```

```
31 SELECT * FROM customer;
```

```
32
```

```
33 SELECT customer_id, first_name, last_name FROM customer
```




```
34 WHERE first_name LIKE 'E%' AND address_id < 500
```

```
35 ORDER BY customer_id DESC
```

```
36 LIMIT 1;
```

```
37
```

Data Output Explain Messages Notifications

	 customer_id [PK] integer	 first_name character varying (45)	 last_name character varying (45)	
1	434	Eddie	Tomlin	

Q5. How many films have the word Truman somewhere in the title?

Answer: 5 Films.

```
SELECT * FROM film;
```

```
SELECT COUNT(film_id) FROM film
```

```
WHERE title ILIKE '%Truman%';
```

```

38 SELECT * FROM film;
39
40 SELECT COUNT(film_id) FROM film
41 WHERE title ILIKE '%Truman%';
42

```

Data Output Explain Messages Notifications

	count	
	bigint	
1	5	

Q6. Display the total amount paid by all customers in the payment table.

Answer: Total amount paid was \$61312.04

SELECT * FROM payment;

SELECT SUM(amount) FROM payment;

```

44
45 SELECT SUM(amount) FROM payment;
46

```

Data Output Explain Messages Notifications

	sum	
	numeric	
1	61312.04	

'S

Q7. Display the total amount paid by each customer in the payment table.

SELECT * FROM payment;

```

SELECT customer_id,COUNT(customer_id) as Total_Transactions,SUM(amount) as
total_amount_paid

FROM payment

GROUP BY customer_id

ORDER BY SUM(amount) DESC;

```

48

49 SELECT customer_id,COUNT(customer_id) as Total_Transactions,SUM(amount) as total_amount_paid

50 FROM payment

51 GROUP BY customer_id

52 ORDER BY SUM(amount) DESC;

53

54

Data Output Explain Messages Notifications

	customer_id smallint	total_transactions bigint	total_amount_paid numeric	
1	148	45	211.55	
2	526	42	208.58	
3	178	39	194.61	
4	137	38	191.62	
5	144	40	189.60	
6	459	37	183.63	
7	181	33	167.67	
8	410	38	167.62	
9	236	39	166.61	
10	403	33	162.67	

Te

Q8. What is the highest total_payment done.

Answer \$211.5 , assuming this refers to highest total payment made by a single person.

```

SELECT * FROM payment;

```

```

SELECT customer_id,COUNT(customer_id) as Total_Transactions,SUM(amount) as
total_amount_paid

FROM payment

GROUP BY customer_id

ORDER BY SUM(amount) DESC

LIMIT 1;

```

```

54
55 SELECT * FROM payment;
56
57 SELECT customer_id,COUNT(customer_id) as Total_Transactions,SUM(amount) as total_amount_paid
58 FROM payment
59 GROUP BY customer_id
60 ORDER BY SUM(amount) DESC
61 LIMIT 1;
62

```

	customer_id smallint	total_transactions bigint	total_amount_paid numeric
1	148	45	211.55

Q9. Which customers have not rented any movies so far.

SELECT * FROM customer;

SELECT customer_id FROM customer

WHERE active = 0;

```

63 SELECT * FROM customer;
64
65 SELECT customer_id FROM customer
66 WHERE active = 0;
67

```

	customer_id [PK] integer
1	16
2	64
3	124
4	169
5	241
6	271
7	315
8	368
9	406
10	446

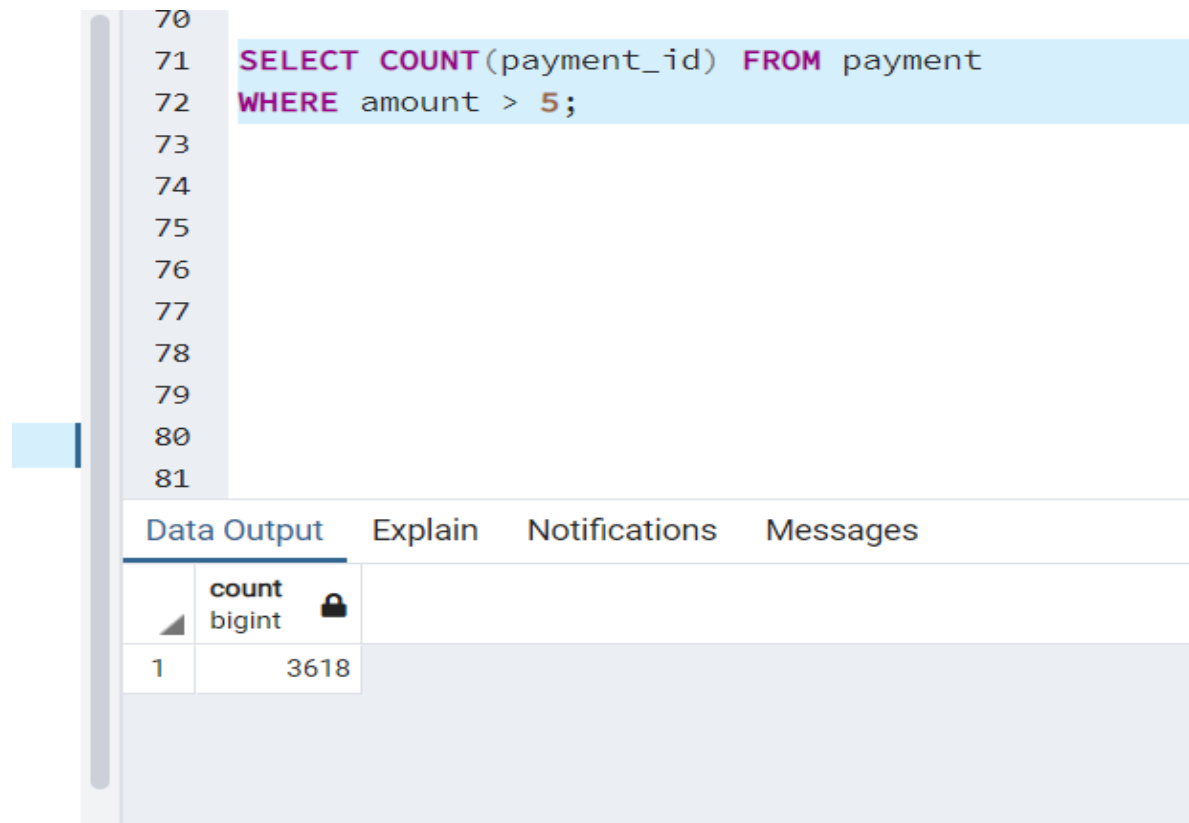
Q10. How many payment transactions were greater than \$5.00?

Answer: 3618 transactions

SELECT * FROM payment;

SELECT COUNT(payment_id) FROM payment

WHERE amount > 5;



The screenshot shows a SQL query editor with a line number column on the left (70-81) and a query text area. The query is: `SELECT COUNT(payment_id) FROM payment WHERE amount > 5;`. Below the query editor, there are four tabs: "Data Output", "Explain", "Notifications", and "Messages". The "Data Output" tab is selected, showing a table with two columns: "count" and "bigint". The first row of data shows the value "3618".

	count	bigint
1	3618	