Jamal Azad

05/08/2022

SQL Assignment 2

1. Write a query to get all the data of actor.

Query: dvdrental=# **SELECT * FROM actor;**

WORKED: Displayed 200 rows!

Screenshot attached.

2. Write a query to get email and last name of customer.

Query: dvdrental=# **SELECT last_name, email FROM customer;**

WORKED: Displayed 599 rows

Screenshot attached.

3. Write a query to get the title, description and release year of the film.

Query: dvdrental=# **SELECT title, description, release_year FROM film;**

WORKED: Displayed 1000 rows

Screenshot attached.

4. Query city and country id in the city table.

Query: dvdrental=# **SELECT city_id, country_id FROM city;**

WORKED: Displayed 600 rows

Screenshot attached.

5. Write a query to get the amount, payment date and customer id from the customer table.

Query: dvdrental=# **SELECT amount, payment_date, customer_id FROM customer;**

```
ERROR:  column "amount" does not exist
LINE 1: SELECT amount, payment_date, customer_id FROM customer;
```

Finding: customer table does not have column for amount. But the payment table has all of these columns.

WORKED: Displayed 14596 rows

Screenshot attached.

6. Write a query to get all data from the language table.

Query: dvdrental=# **SELECT * FROM language;**

WORKED: Displayed 6 rows!

```
 language_id |         name         |     last_update
-------------+----------------------+---------------------
           1 | English              | 2006-02-15 10:02:19
           2 | Italian              | 2006-02-15 10:02:19
           3 | Japanese             | 2006-02-15 10:02:19
           4 | Mandarin             | 2006-02-15 10:02:19
           5 | French               | 2006-02-15 10:02:19
           6 | German               | 2006-02-15 10:02:19
(6 rows)
```

7. Query all columns for a payment in payment table with customer id 10.

Query: dvdrental=# **SELECT payment FROM payment WHERE customer_id=10;**

WORKED: Displayed 24 rows!

```
                     payment
------------------------------------------------------
(18532,10,1,1801,4.99,"2007-02-16 18:50:19.996577")
(18533,10,1,1995,4.99,"2007-02-17 09:39:40.996577")
(18534,10,2,2222,3.99,"2007-02-18 01:54:49.996577")
(18535,10,1,2814,0.99,"2007-02-19 18:30:25.996577")
(18536,10,1,2865,0.99,"2007-02-19 22:29:21.996577")
(22773,10,2,10671,8.99,"2007-03-01 15:38:25.996577")
(22774,10,2,11289,2.99,"2007-03-02 13:23:26.996577")
(22775,10,1,11405,0.99,"2007-03-02 17:42:05.996577")
(22776,10,2,12031,2.99,"2007-03-17 18:40:01.996577")
(22777,10,2,12400,2.99,"2007-03-18 07:47:38.996577")
(22778,10,2,13316,4.99,"2007-03-19 17:51:56.996577")
(22779,10,2,13917,2.99,"2007-03-20 15:11:54.996577")
(22780,10,1,15370,5.99,"2007-03-22 20:27:55.996577")
(29094,10,2,3790,3.99,"2007-04-06 12:42:11.996577")
(29095,10,2,4042,4.99,"2007-04-07 01:35:06.996577")
(29096,10,1,4255,1.99,"2007-04-07 12:42:39.996577")
(29097,10,1,5038,7.99,"2007-04-09 01:41:18.996577")
(29098,10,2,5068,2.99,"2007-04-09 03:21:44.996577")
(29099,10,1,5444,0.99,"2007-04-09 20:27:23.996577")
(29100,10,1,5905,2.99,"2007-04-10 19:09:35.996577")
(29101,10,1,7738,2.99,"2007-04-28 03:50:08.996577")
(29102,10,2,8001,6.99,"2007-04-28 13:39:21.996577")
(29103,10,2,8188,4.99,"2007-04-28 21:02:38.996577")
(29104,10,1,9935,4.99,"2007-04-30 13:55:33.996577")

(24 rows)
```

8. Query last name and first name of customers in customer table whose first names are "Mary"

Query: dvdrental=# **SELECT last_name, first_name FROM customer WHERE first_name=’Mary’;**

WORKED: Displayed 1 row!

```
   last_name | first_name
-----------+------------
 Smith      | Mary

(1 row)
```

9. Query last name and first name of customers in customer table whose first names are "Mary" and last names are "Smith".

Query: dvdrental=# **SELECT DISTINCT last_name, first_name FROM customer WHERE first_name='Mary' AND last_name='Smith';**

WORKED: Displayed 1 row!

```
 last_name | first_name
-----------+------------
 Smith      | Mary

(1 row)
```

10. Query last name and first name of customers in customer table whose first names are "Susan" or last names are "Jones".

Query: dvdrental=# **SELECT DISTINCT last_name, first_name FROM customer WHERE first_name='Susan' AND last_name='Jones';**

WORKED: Displayed 0 rows!

```
last_name | first_name
-----------+------------
(0 rows)
```

11. Query email of customers in customer table whose first name is "Mar", "Mary" or "Mari".

Query: dvdrental=# **SELECT DISTINCT email FROM customer WHERE first_name = 'Mar' OR first_name='Mary' OR First_name='Mari';**

WORKED: Displayed 1 row!

```
            email
-------------------------------
mary.smith@sakilacustomer.org

(1 row)
```

12. Query last name and first name of customers in customer table whose first names start with "Ma".

Query: dvdrental=# **SELECT last_name, first_name FROM customer WHERE first_name LIKE 'Ma%';**

WORKED: Displayed 31 rows!

Screenshot attached.

13. Write a query to get staff id, first name and username of staff in staff table whose staff id is 10.

Query: dvdrental=# **SELECT staff_id, first_name, username FROM staff WHERE staff_id=10;**

WORKED: Displayed 0 rows!

```
 staff_id | first_name | username
----------+------------+----------

(0 rows)
```

14. Query last name and first name of customers in customer table whose first name starts with the letter "M" and contains 3 to 5 characters.

Query: dvdrental=# **SELECT last_name, first_name FROM customer WHERE first_name LIKE 'M%' AND LENGTH(First_name) >=3 AND LENGTH(first_name) <=5;**

WORKED: Displayed 12 rows!

```
 last_name | first_name
-----------+------------
 Smith     | Mary
 Miller    | Maria
 Turner    | Marie
 Palmer    | Megan
 Holland   | Mabel
 Lambert   | Misty
```

Continued….

```
 Fletcher  | Mae
 Rinehart  | Mark
 Way       | Mike
 Cheatham  | Mario
 Outlaw    | Marc
 Pitt      | Max
(12 rows)
```

15. Query last name and first name of customers in customer table whose first names start with "Bra" and last names are not "Motley".

```
Query: dvdrental=# SELECT last_name, first_name FROM customer WHERE first_name
LIKE '%Bra%' AND last_name NOT LIKE 'Motley';
WORKED: Displayed 3 rows!
last_name | first_name
-----------+------------
 Graves    | Brandy
 Huey      | Brandon
 Mccurdy   | Brad
(3 rows)
```

16. Query store id of stores that have more than 300 customers in customer table.

```
Query: dvdrental=# SELECT Store_id FROM customer GROUP BY store_id HAVING
COUNT(customer) > 300;
WORKED: Displayed 1 row!
 store_id
----------
        1
(1 row)
Query: dvdrental=# SELECT Store_id, COUNT(customer) FROM customer GROUP BY
store_id HAVING COUNT(customer) > 300;
WORKED: Displayed 1 row!
 store_id | count
----------+-------
        1 |   326
(1 row)
```

17. Write a query to select all details of the only customers who have been spending more than 200 in customer table.

```
Query: dvdrental=# SELECT * FROM customer GROUP BY customer.customer_id HAVING
SUM(amount) > 200;
ERROR:  column "amount" does not exist
LINE 1: ...customer GROUP BY customer.customer_id HAVING SUM(amount) > …
```

Alternate option:

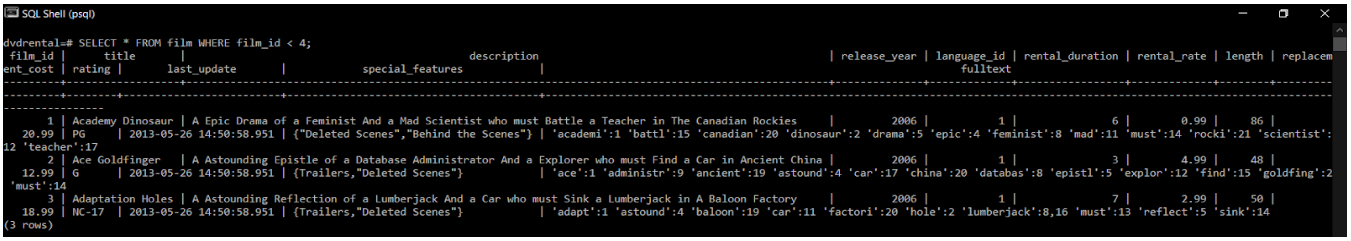
```
Query: dvdrental=# SELECT * FROM payment GROUP BY payment.payment_id HAVING
SUM(amount) > 200;
```

```
payment_id | customer_id | staff_id | rental_id | amount | payment_date
------------+-------------+----------+-----------+--------+--------------
(0 rows)
```

18. Query all columns in film table where the film_id is less than 4.

Query: dvdrental=# **SELECT * FROM film WHERE film_id < 4;**

WORKED: Displayed 3 rows!



```
dvdrental=# SELECT * FROM film WHERE film_id < 4;
 film_id |       title       |                                        description                                        | release_year | language_id | rental_duration | rental_rate | length | replacem
ent_cost | rating |        last_update         |            special_features            |                                        fulltext
---------+-------------------+-------------------------------------------------------------------------------------------+--------------+-------------+-----------------+-------------+--------+---------
---------+--------+----------------------------+----------------------------------------+----------------------------------------------------------------------------------------------------------
----------------
       1 | Academy Dinosaur | A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies      |         2006 |           1 |               6 |        0.99 |     86 |
   20.99 | PG     | 2013-05-26 14:50:58.951 | {"Deleted Scenes","Behind the Scenes"} | 'academi':1 'battl':15 'canadian':20 'dinosaur':2 'drama':5 'epic':4 'feminist':8 'mad':11 'must':14 'rocki':21 'scientist':
12 'teacher':17
       2 | Ace Goldfinger   | A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China |         2006 |           1 |               3 |        4.99 |     48 |
   12.99 | G      | 2013-05-26 14:50:58.951 | {Trailers,"Deleted Scenes"}            | 'ace':1 'administr':9 'ancient':19 'astound':4 'car':17 'china':20 'databas':8 'epistl':5 'explor':12 'find':15 'goldfing':2
 'must':14
       3 | Adaptation Holes | A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory      |         2006 |           1 |               7 |        2.99 |     50 |
   18.99 | NC-17  | 2013-05-26 14:50:58.951 | {Trailers,"Deleted Scenes"}            | 'adapt':1 'astound':4 'baloon':19 'car':11 'factori':20 'hole':2 'lumberjack':8,16 'must':13 'reflect':5 'sink':14
(3 rows)
```

19. Write a query to get all data from address table.

Query: dvdrental=# **SELECT * FROM address;**

WORKED: Displayed 603 rows!

20. Query rental date, customer id and rental id in rental table when rental date is 2005-05-25.

Query: dvdrental=# **SELECT rental_date, customer_id, rental_id FROM rental WHERE rental_date=’%2005-05-25’;**

WORKED: Displayed 0 rows!

```
 rental_date | customer_id | rental_id
-------------+-------------+-----------
(0 rows)
```

21. Query all columns for customers in customer table with store id 2 or customer id 7.

Query: dvdrental=# **SELECT * FROM customer WHERE STORE_ID='2' OR CUSTOMER_ID='7';**

WORKED: Displayed 274 rows!

22. Query all columns for customers in customer table who have spent amount more than $200.

dvdrental=# **SELECT * FROM customer WHERE amount > '200';**

```
ERROR:  column "amount" does not exist
LINE 1: SELECT * FROM customer WHERE amount > '300';
                                     ^
```

23. Query amount and payment_date from payment where the amount paid was less than $2.

Query: dvdrental=# **SELECT amount, payment_date from payment WHERE amount<2;**

WORKED: Displayed 3325 rows!

24. Write a query to get a list of actors with the first name Chris, Cameron, or Cuba.

Query: dvdrental=# **SELECT DISTINCT * FROM actor WHERE first_name = 'Chris' OR first_name= 'Cameron' OR first_name='Cuba';**

WORKED: Displayed 3325 rows!

```
 actor_id | first_name | last_name |        last_update
----------+------------+-----------+------------------------
       24 | Cameron    | Streep    | 2013-05-26 14:47:57.62
      118 | Cuba       | Allen     | 2013-05-26 14:47:57.62
       98 | Chris      | Bridges   | 2013-05-26 14:47:57.62
       15 | Cuba       | Olivier   | 2013-05-26 14:47:57.62
      160 | Chris      | Depp      | 2013-05-26 14:47:57.62
      111 | Cameron    | Zellweger | 2013-05-26 14:47:57.62
       63 | Cameron    | Wray      | 2013-05-26 14:47:57.62
      189 | Cuba       | Birch     | 2013-05-26 14:47:57.62

(8 rows)
```

25. Query last name of customers in customer table whose first names are "John".

Query: dvdrental=# **SELECT last_name FROM customer WHERE first_name = 'John';**

WORKED: Displayed 1 row!

```
 Last_name
------------
 Farnsworth
(1 row)
```

Query: dvdrental=# **SELECT last_name, first_name FROM customer WHERE first_name = 'John';**

WORKED: Displayed 1 row!

```
 last_name  | first_name
------------+------------
 Farnsworth | John
(1 row)
```

26. Write a query to get staff id, first name and username of staff in staff table whose store id is less than 6.

Query: dvdrental=# **SELECT staff_id, first_name, username FROM staff WHERE store_id < 6;**

WORKED: Displayed 2 rows!

```
 staff_id | first_name | username
----------+------------+----------
        1 | Mike       | Mike
        2 | Jon        | Jon

(2 rows)
```

27. Write a query to get release year, rental duration and rental rate of films in film table.

Query: dvdrental=# **SELECT release_year, rental_duration, rental_rate FROM film;**

WORKED: Displayed 1000 rows!

28. Write a query to get city id and country id of country in country table whose name is "New York".

Query: dvdrental=# **SELECT CITY_id, country_id FROM country WHERE city= 'New York';**

```
ERROR:  column "city_id" does not exist
LINE 1: SELECT CITY_id, country_id FROM country WHERE city_name= 'Ne...
```

29. Write a query to get all data of city table.

Query: dvdrental=# **SELECT * from city;**

WORKED: Displayed 600 rows!

30. Write a query to get film id of film in film_category table with category_id 2.

Query: dvdrental=# **SELECT film_id from film_category WHERE category_id=2;**

WORKED: Displayed 66 rows!