


2.A

 dvdrental/postgres@PostgreSQL 13 ▾

Query Editor

Query History

```
1 select film_id,title
2 from film inner join inventory using(film_id)
3 full outer join rental using(inventory_id)
4 where return_date is null
5 order by film_id;
```

Data Output

Explain

Messages

Notifications

	<b>film_id</b> [PK] integer	<b>title</b> character varying (255)
1	1	Academy Dinosaur
2	1	Academy Dinosaur
3	2	Ace Goldfinger
4	4	Affair Prejudice
5	5	African Egg
6	13	Ali Forever
7	17	Alone Trip
8	19	Amadeus Holy
9	21	American Circus
10	22	Amistad Midsummer

2.B

The screenshot shows a PostgreSQL query editor interface. The top bar includes tabs for Dashboard, Properties, SQL, Statistics, Dependents, and a connection to 'dvdrental/postgres@PostgreSQL 13'. Below the toolbar, the 'Query Editor' tab is active, displaying the following SQL query:

```
1 select avg(return_date - rental_date) as avg_rental_time
2 from rental inner join inventory using(inventory_id)
3 inner join film_category using(film_id)
4 group by category_id;
```

To the right of the query editor is a 'Scratch Pad' tab. Below the query editor, the 'Data Output' tab is selected, showing the results of the query. The results are displayed in a table with two columns: 'avg\_rental\_time' and 'interval'. The table contains 10 rows of data, each representing a film category and its average rental time.

	avg_rental_time	interval
1	4 days 22:45:54.903225	
2	4 days 25:54:12.73559	
3	4 days 22:48:56.805111	
4	4 days 24:35:17.540453	
5	4 days 28:24:44.167539	
6	4 days 21:14:59.031339	
7	4 days 24:30:40.802348	
8	4 days 24:38:36.821918	
9	4 days 27:34:51.695279	
10	4 days 20:55:51.510917	

## 2.c

dvdfrental/postgres@PostgreSQL 13 ▾

Query Editor

Query History

Scratch Pad

```
1
2 with most_seen(film_id) as
3 (
4   select film_id
5   from film inner join inventory using(film_id)
6   inner join rental using(inventory_id)
7   group by film_id
8   order by count(rental_id) desc
9   limit 7
10 )
11 select actor_id,first_name
12 from actor a inner join film_actor using(actor_id)
13 inner join most_seen using(film_id)
14 order by a.first_name;
```

Data Output

Explain

Messages

Notifications

	actor_id [PK] integer	first_name character varying (45)
3	111	Cameron
4	89	Charlize
5	98	Chris
6	160	Chris
7	61	Christian
8	179	Ed
9	139	Ewan
10	51	Gary
11	151	Geoffrey
12	86	Greg

2.d

Query Editor

Query History

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

```
create view v(name,num_rental,avg_time,num_actor) as
select
title,
(
select count(rental_id)
from film inner join inventory using(film_id)
inner join rental using(inventory_id)
group by film_id
),
(
select avg(return_date - rental_date) as avg_rental_time
from rental inner join inventory using(inventory_id)
group by film_id
),
```

Data Output

Explain

Messages

Notifications

CREATE VIEW

Query returned successfully in 204 msec.