

Instituição: Instituto Federal Catarinense - campus Blumenau

Aluno: José Mateus Amaral de Miranda

Matrícula: 2022005197

Curso: Bacharelado em Ciência da Computação

Fase: 04

Matéria: Banco de Dados 2

Data: 19/09/2023

Av3-b

A1)

- a.

```
SELECT first_name, last_name
FROM customer
WHERE customer_id IN (SELECT customer_id FROM payment WHERE amount > 10);
```

∴ Esta consulta retorna os nomes dos clientes que fizeram pagamentos de valor superior a \$10.
- b.

```
SELECT film_id, title
FROM film
WHERE film_id IN (SELECT film_id FROM film_category WHERE category_id = 1);
```

∴ Nesta consulta, estamos selecionando os filmes da categoria 1 usando uma subconsulta que depende da categoria_id da tabela externa.
- c.

```
SELECT title
FROM film
WHERE EXISTS (SELECT * FROM rental WHERE rental.film_id = film.film_id);
```

∴ Esta consulta retorna os títulos dos filmes que foram alugados pelo menos uma vez.
- d.

```
SELECT DISTINCT category_id
FROM film_category;
```

∴ Esta consulta retorna os IDs de categoria únicos da tabela film_category.
- e.

```
SELECT customer.first_name, customer.last_name, payment.amount
FROM customer
NATURAL JOIN payment;
```

∴ Nesta consulta, estamos realizando uma junção natural entre as tabelas customer e payment com base nas colunas comuns.

- f. `SELECT customer.first_name, customer.last_name, payment.amount
FROM customer
LEFT JOIN payment ON customer.customer_id = payment.customer_id;`
∴ Esta consulta retorna todos os clientes, mesmo aqueles que não fizeram nenhum pagamento, usando uma junção LEFT.
- g. `SELECT staff_id, SUM(amount) AS total_amount
FROM payment
GROUP BY staff_id;`
∴ Esta consulta calcula o total de pagamentos feitos por cada funcionário da loja (staff) usando a agregação SUM.
- h. `SELECT category_id, AVG(length) AS average_length
FROM film
GROUP BY category_id;`
∴ Esta consulta calcula a média do comprimento dos filmes para cada categoria.

A2)

```
CREATE VIEW view_customer_payments AS  
SELECT customer.first_name, customer.last_name, SUM(payment.amount) AS  
total_amount  
FROM customer  
LEFT JOIN payment ON customer.customer_id = payment.customer_id  
GROUP BY customer.customer_id;  
∴ Este comando cria uma visão que mostra o total de pagamentos feitos por  
cada cliente.
```

A3)

```
import sqlite3  
  
# Conectar ao banco de dados Sakila  
conn = sqlite3.connect('sakila.db')  
  
# Criar um cursor  
cursor = conn.cursor()  
  
# Executar a consulta na visão  
cursor.execute("SELECT * FROM view_customer_payments")  
  
# Recuperar os registros  
rows = cursor.fetchall()
```

```
# Exibir os registros
for row in rows:
    print(row)
```

```
# Fechar a conexão com o banco de dados
conn.close()
```