### FLEXTV DATABASE SYSTEM

### **OVERVIEW**

The database utilized for this project was built using two tables: the customer table and the payment table, each of which has several columns. It contains statistics for a month's worth of subscriber information from several African countries.

The aim of this project is to understand the overall sales performance of this company in the month of June 2023.

### PROBLEM STATEMENT

Flextv, an online movie streaming service, has launched its website and mobile application for viewings in the latter week of May 2023.

The company had provided a 20% membership discount to the first consumers who subscribed within the next 30 days of June 2023.

Setup Database - Postgre SQL and PGAdin console was used for this project.

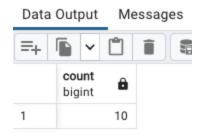
**SQL** concepts that are used:

- 1. Basic operations: INSERT, UPDATE, DELETE, SELECT, JOIN
- 2. Clauses: WHERE, AND, GROUP BY, CASE etc
- 3. Operations on tables: CREATE, ALTER, Foreign Key Constraints
- 4. Indexes

### QUESTIONS

1. How many customers subscribed in the month of June?

SELECT COUNT(\*) FROM customer;

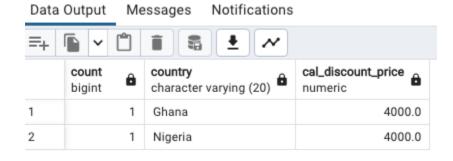


2. How many customers subscribed with #4,000 and above and what country?

SELECT COUNT(customer\_id),country,discount\_price AS cal\_discount\_price FROM payment

WHERE discount\_price >=4000

GROUP BY cal\_discount\_price, country;



3. List the first five customers to subscribe . Return customers name, email, country, and payment date.

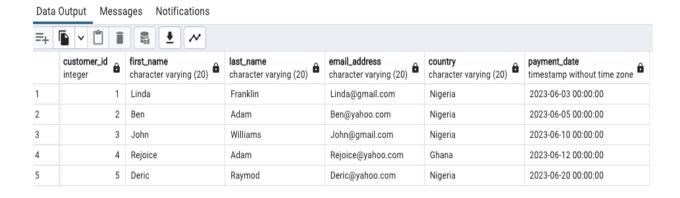
SELECT DISTINCT (customer\_id),first\_name,last\_name,email\_address,Country,payment\_date

FROM customer

LEFT JOIN payment ON payment.customer\_id = customer. id

ORDER BY customer\_id ASC

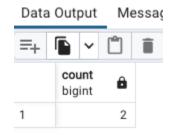
LIMIT 5;



## 4. How many subscribers are from Ghana?

SELECT COUNT(\*) FROM payment

WHERE country LIKE '%Ghana';



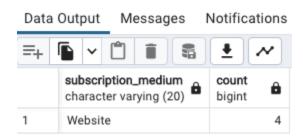
## 5. What medium was least used for payment?

SELECT DISTINCT subscription\_medium,COUNT(\*) FROM payment

GROUP BY subscription\_medium

ORDER BY COUNT ASC

LIMIT 1;



## 6. List customer ids with minimum discount price.

SELECT customer\_id,discount\_price FROM payment

WHERE discount\_price = (SELECT MIN(discount\_price) from payment);

SELECT customer\_id,discount\_price FROM payment

WHERE discount\_price = (SELECT MIN(discount\_price) from payment);

Data Output Messages Notification		
=+		
	customer_id integer	discount_price numeric
1	1	2800.0
2	3	2800.0
3	5	2800.0
4	7	2800.0
5	8	2800.0
6	10	2800.0

# 7. What is the status of payment as 'Low', 'Standard', and 'High'?

ALTER TABLE payment RENAME discount\_price to d\_price;

SELECT d\_price,

CASE

WHEN d\_price <=2800 then 'L'

WHEN d\_price =4000 then 'H'

ELSE 'S'

**END** 

AS price\_status

## FROM payment;

Data Output Messages Notifi		
=+	~ °	
	d_price numeric	price_status text
1	2800.0	L
2	4000.0	Н
3	2800.0	L
4	3360.0	S
5	2800.0	L
6	4000.0	Н
7	2800.0	L
8	2800.0	L
9	3360.0	S
10	2800.0	L