**A Project Report on**

**Netflix Database Management System**

****

**Department of Information Technology Faculty of Technology,**

**Dharmsinh Desai University College Road, Nadiad-387001**

**October-2022**

Developed By:

1. Patel Jenish-IT108
2. Patel Megh-IT109
3. Patel Het-IT106

Guided By

**Prof. Archana N. Vyas**

**& Prof. Shweta Jambukia**

**CERTIFICATE**

THIS CERTIFICATE IS TO CERTIFY THAT THE PROJECT

ENTITLED “NETFLIX DATABASE MANAGEMENT SYSTEM” IS ,

A BONAFIDE REPORT OF THE WORK CARRIED OUT BY,

1. Patel Jenish Student ID no:20ITUBS039
2. Patel Megh Student ID no:20ITUON146

3)Patel Het Student ID no:20ITUON154

OF DEPARTMENT OF INFORMATION TECHNOLOGY,

SEMESTER V, UNDER THE GUIDANCE AND SUPERVISION

FOR THE SUBJECT DATABASE MANAGEMENT SYSTEM.

THEY WERE INVOLVED IN PROJECT TRAINING DURING THE ACADEMIC YEAR 2022-2023.



dharaMsinh desai uniVersiTy nadiad

COMMENDATION

We would like to express our heartfelt gratitude to everyone who contributed to the successful completion of our project "Netflix DataBase Management System ".

The success and ultimate conclusion of this project necessitated a great deal of advice and support from a large number of individuals and we are incredibly fortunate to have received it all along with the project's completion.

We owe a debt of appreciation to **Prof. Archana N. Vyas & Prof. Shweta Jambukia**, our project guide, who took an interest in our project work and directed us through it till it was completed by giving all of the required assistance for creating a solid Database System.

We'd also want to express our gratitude to all of our speakers.

Finally, we express our gratitude to all of our friends and colleagues.

**INDEX**

**I ) Certificate**

**II) Commendation**

**1) System Overview 5**

**1.1) Current System & Objective of System**

**1.2) Advantage of System**

**2) E-R Diagram 6**

**3) Schema Diagram 7**

**4) Data Dictionary 8**

**5) Database Implementation 13**

**5.1) Create Schema 13**

**5.2) Insert Data Values 16**

**5.3) Queries DBMS(Based on constructs) 20**

**5.4) PL/SQL Blocks(Views) 23**

**5.5) Functions and Trigger 24**

**5.6) Cursors 26**

**6) Future Enhancement System 27**

**7) Bibliography 28**1.**SYSTEM OVERVIEW**

* 1. **Current System & Objectives**

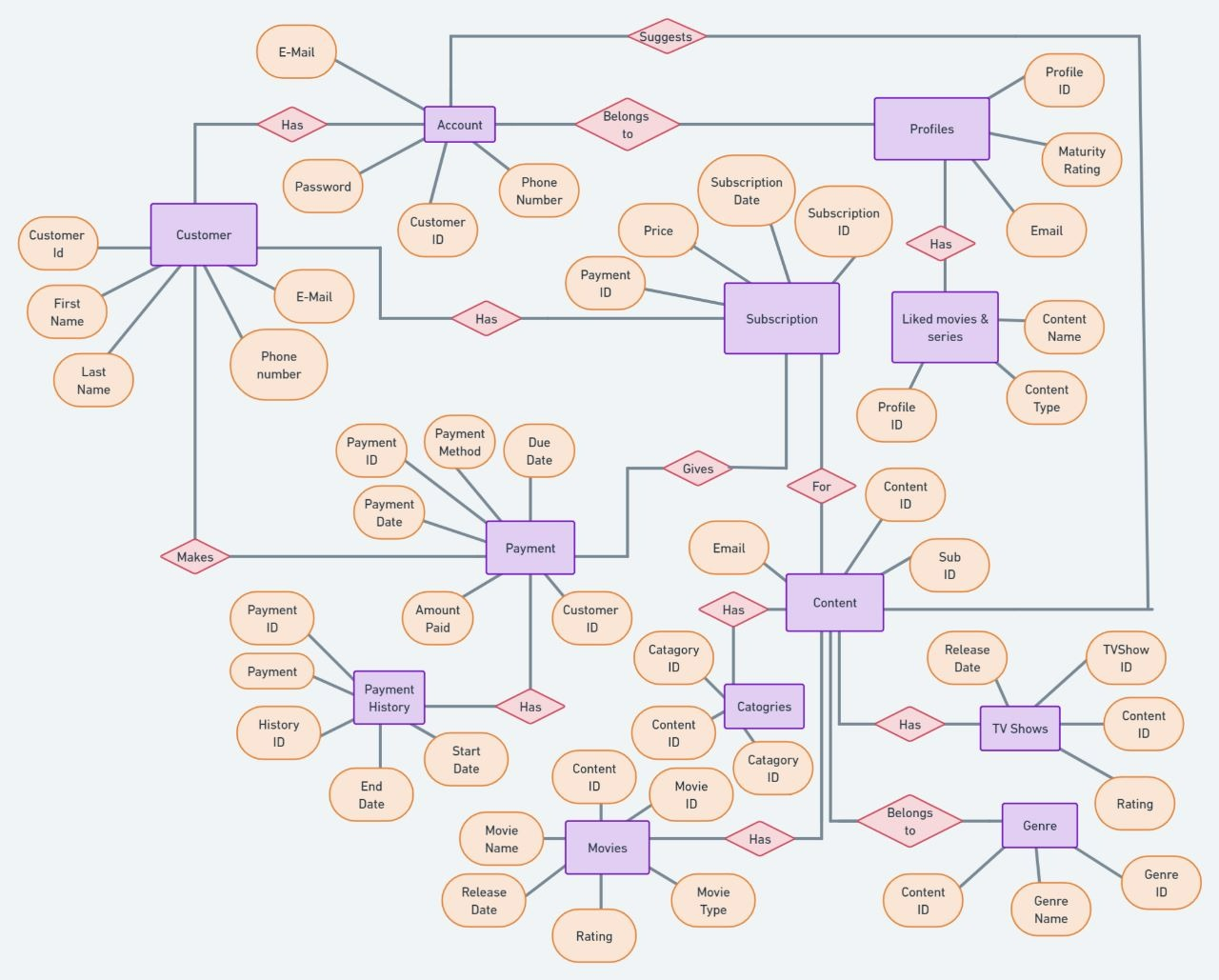
Our database will be designed based on OTT Streaming service

‘NETFLIX’ database management system. Our major focus will be on structuring the data such a way that a customer of a Streaming service and easily access and use our database, And we can easily manage all the data of customers and our content. It will also be done in an effective manner since we don’t want our Database to become redundant.So, We will try to make the database with the highest power possible.

* 1. **Advantage of System**

Netflix is Server and Client-side Application. All customer who has subscription can Watch Movies & Tv-shows, according to their plan they have a different video quality and allowed number of connected devices. Both customer and their payment data and movies and tv-shows which customers are streaming everything is managed very accurately. This type of Database System

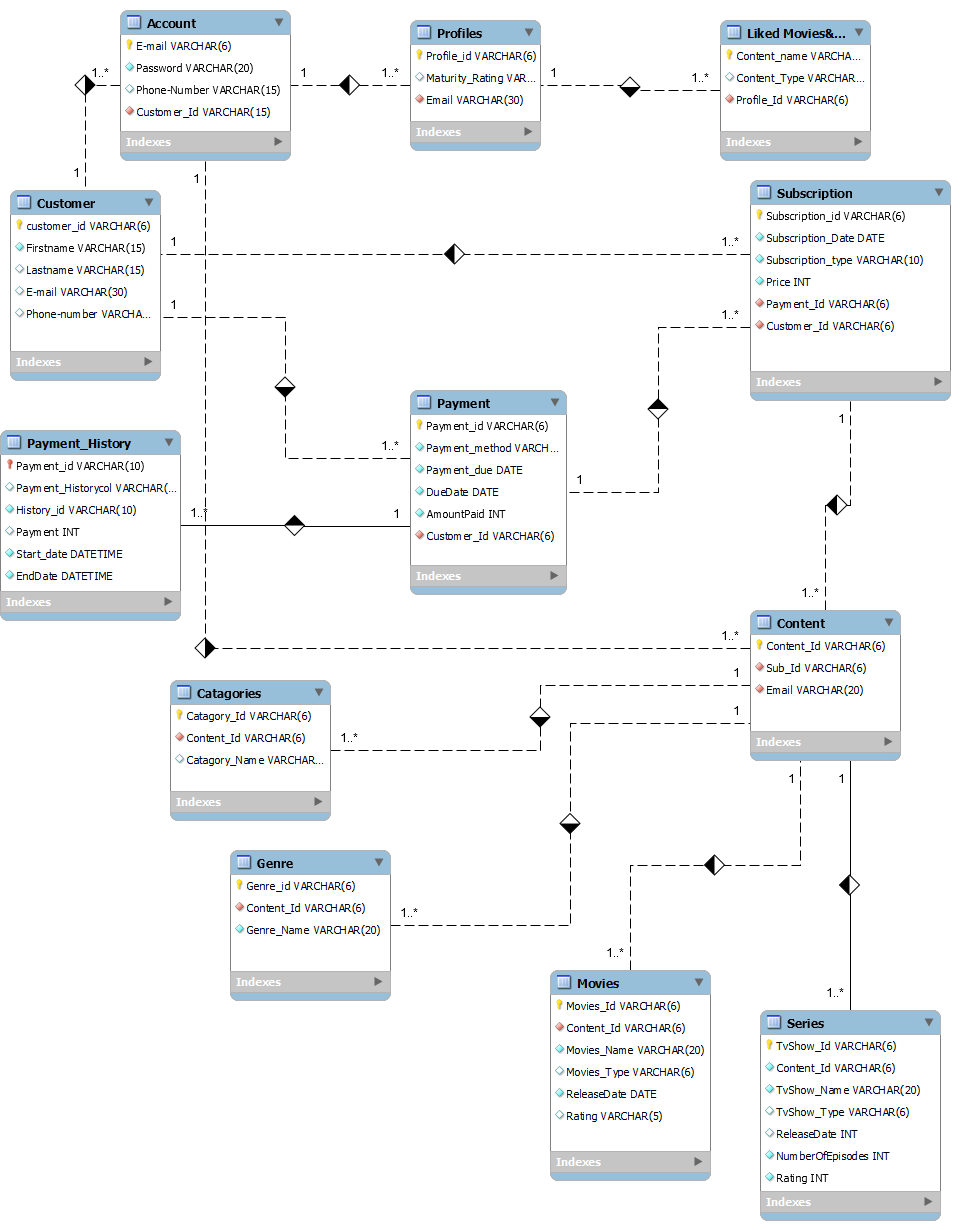
allows Netflix to suggest movies & Tvshows of their like or of type match of what movies they liked in the past.

**2) ER-Entity Relational Diagram**

Link => https://whimsical.com/YW63bK8pU6HZXs7F4h2YoD

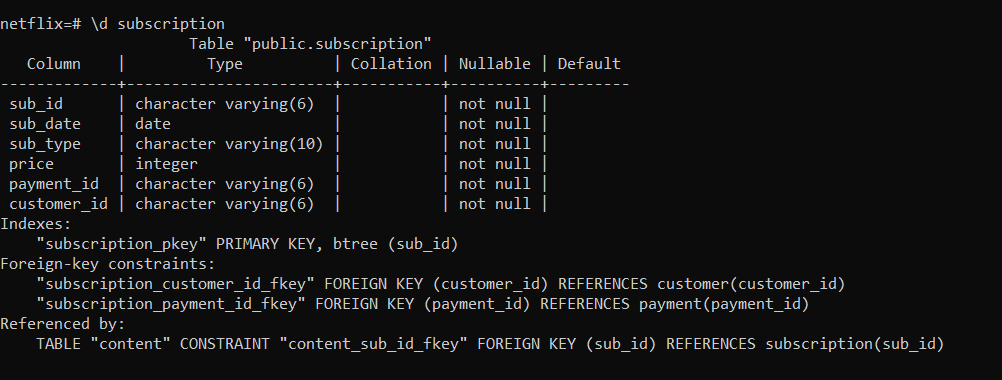
Password: 12345

**3. RELATIONAL SCHEMA**

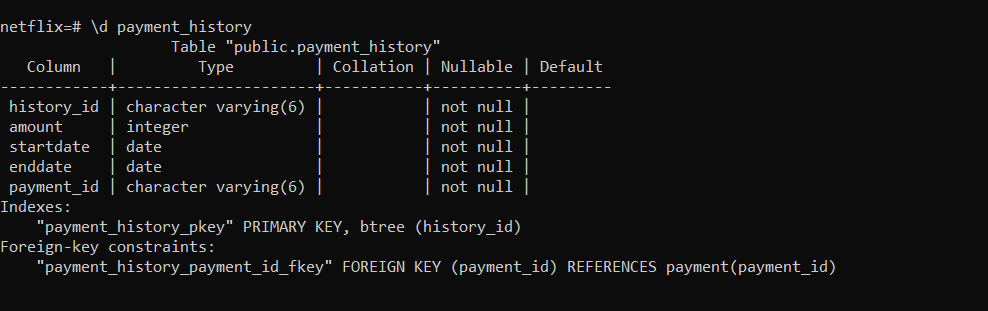


**4. Data Dictionary**

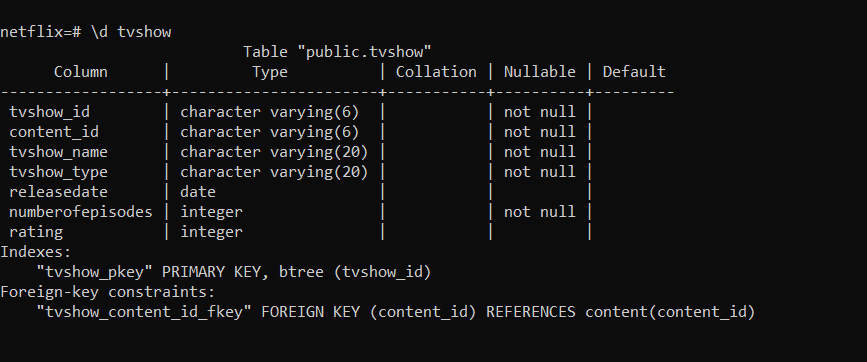
**4.1) Subscription**

****

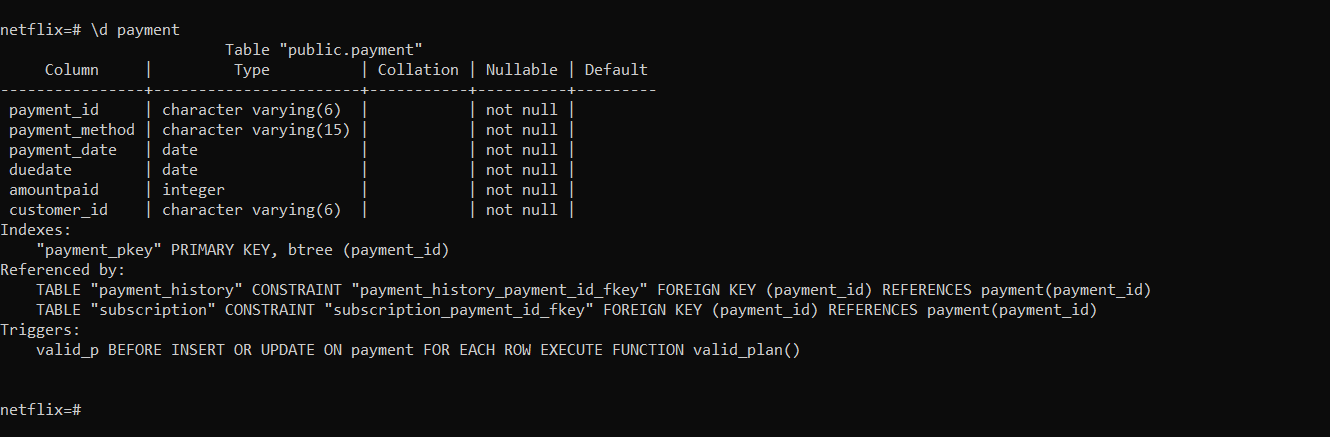
**4.2) Payment History**

****

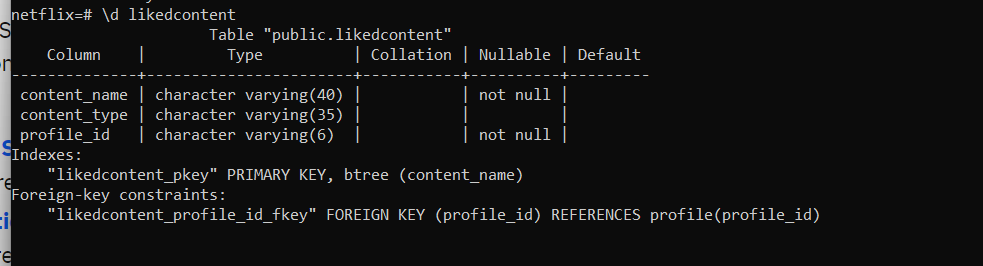
**4.3) TV-Show**

****

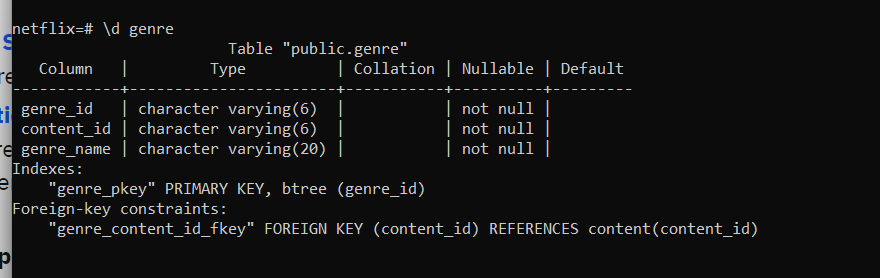
**4.4) Payment**

****

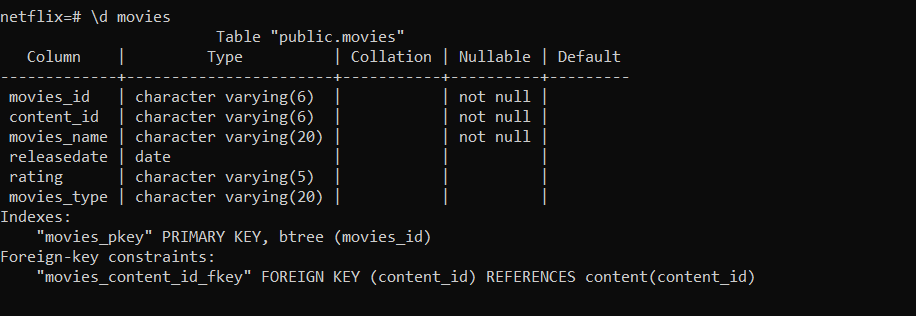
**4.5) Liked Content**

****

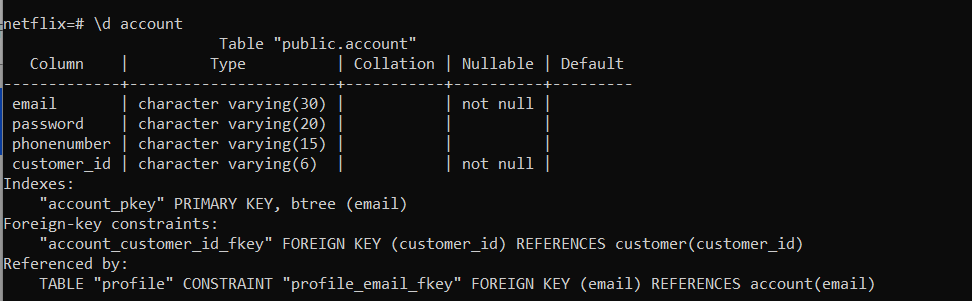
**4.6) Genre**

****

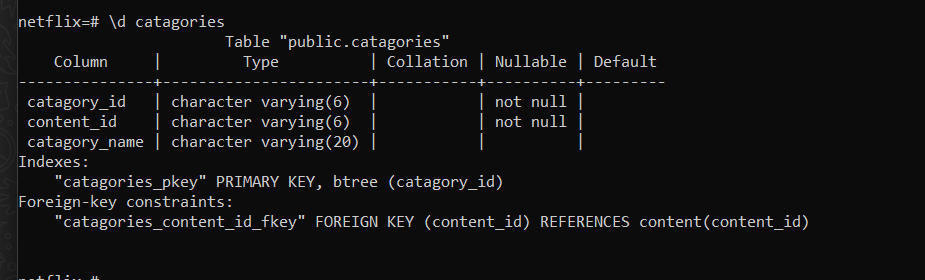
**4.7) Movies**

****

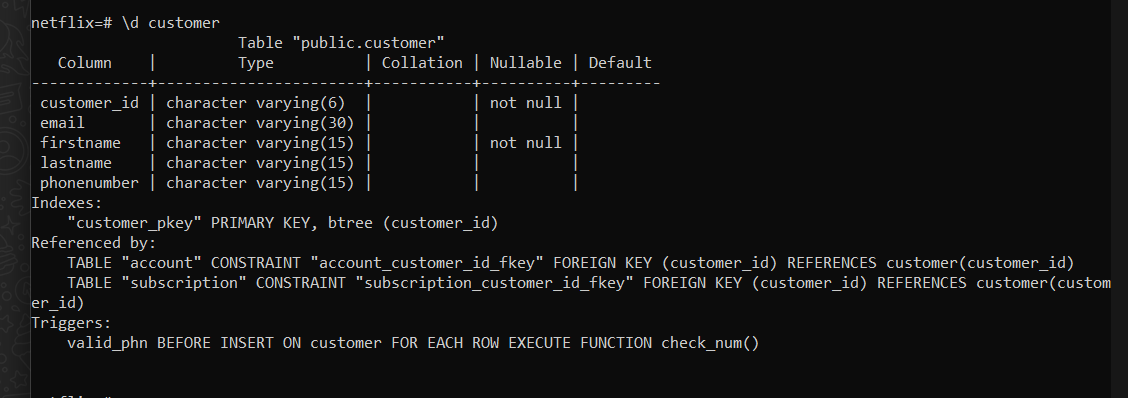
**4.8) Account**

****

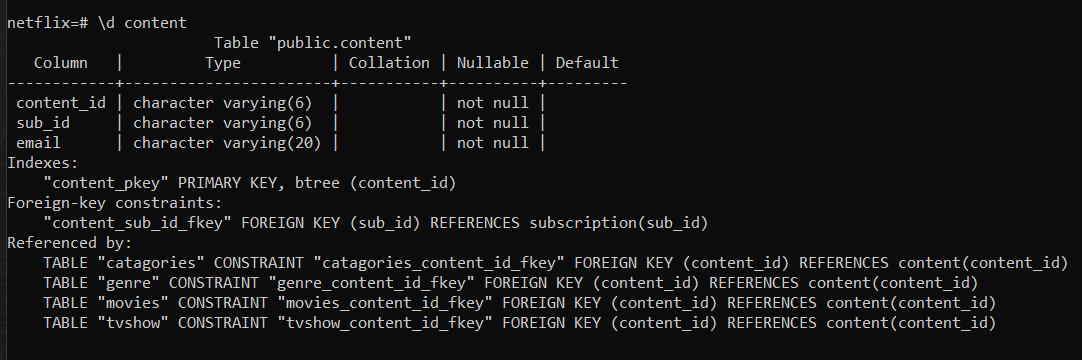
**4.9) Categories**

****

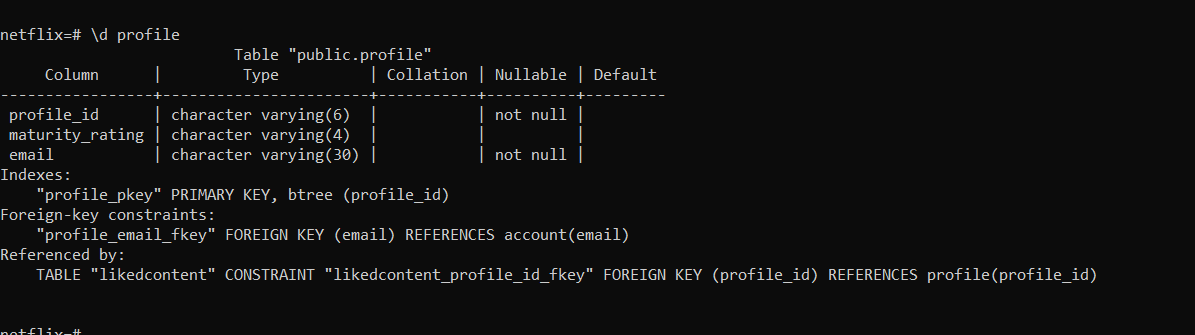
**4.10) Customer**

****

**4.11) content**

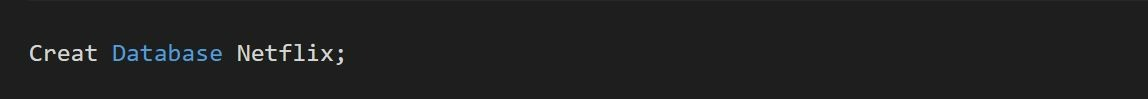
****

**4.12) Profile**

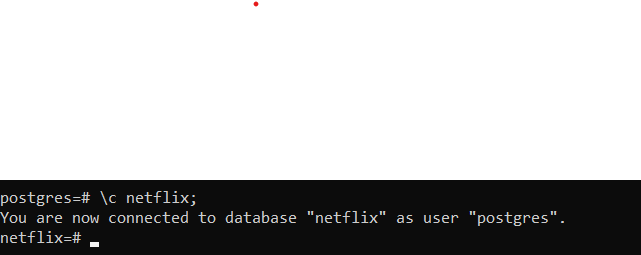
****

**5. Data Implementation**

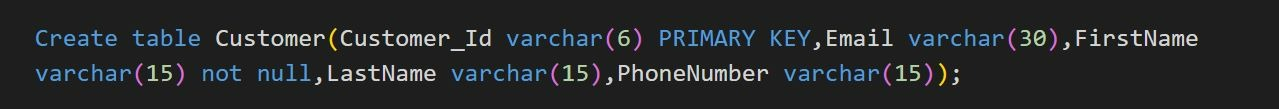
* **First Create Database name ->Netflix**



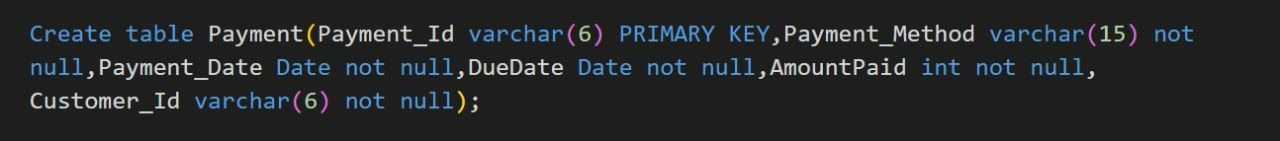
* **Schema => Total 12 Table**
* **Connecting To The ‘Netflix’ Database**



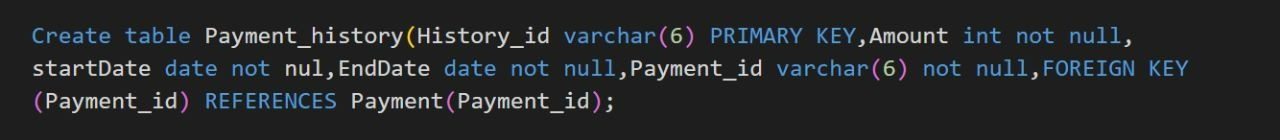
# Customer



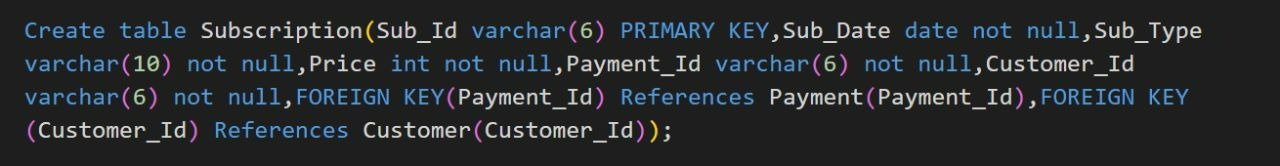
1. Payment

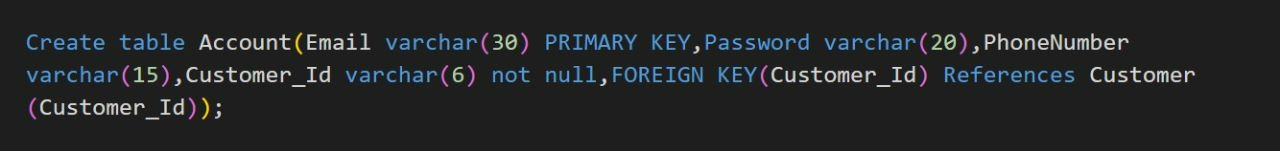


1. Payment History

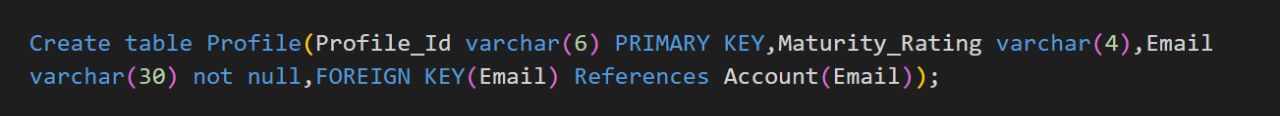


1. Subscription

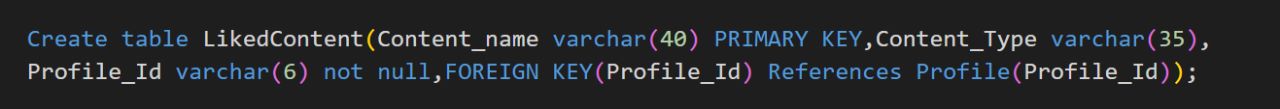
 **5)** Account



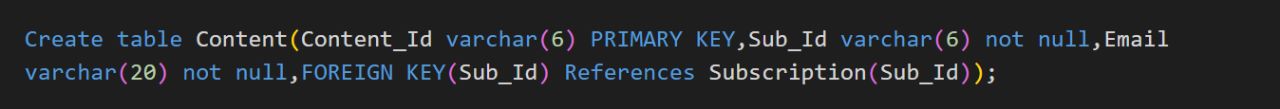
**6)** Profile



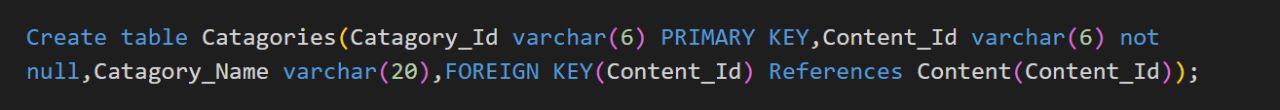
**7)** Liked Content



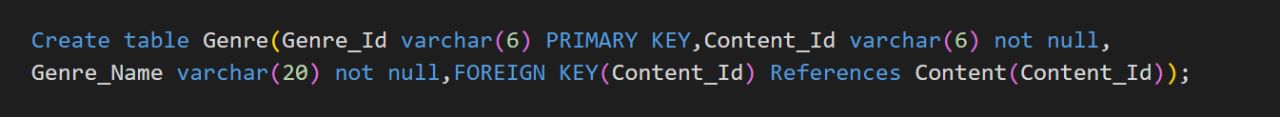
**8)** Content



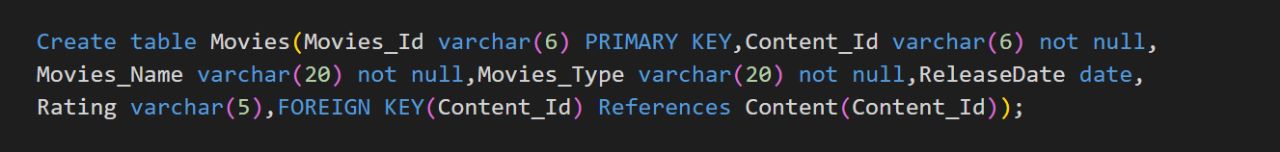
**9)** Categories



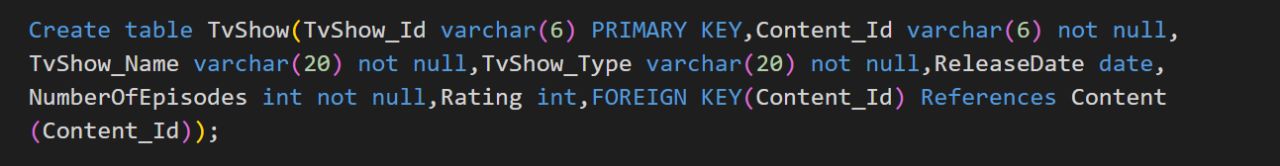
**10)** Genre



**11)** Movies

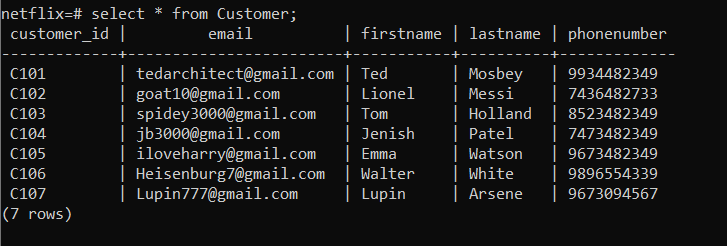


**12)** TV-Show

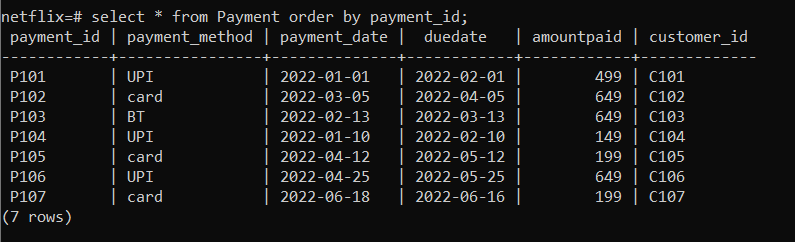


**5.2) Inserting Data Values & Creating Tables**

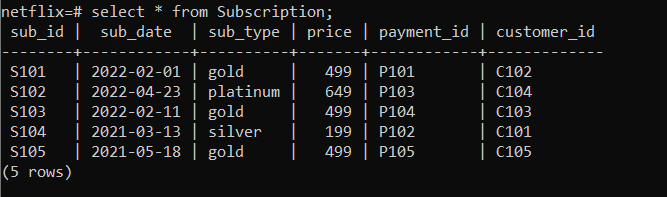
1. **Customer:-**



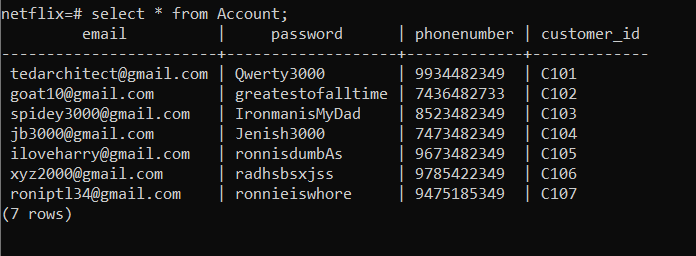
1. **Payment:-**



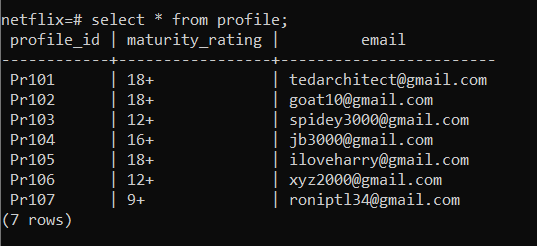
1. **Subscription:-**



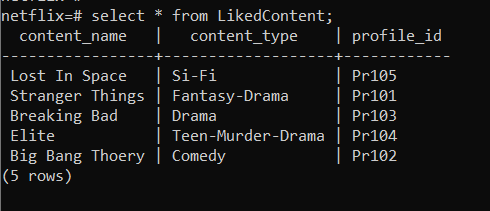
1. **Account:-**



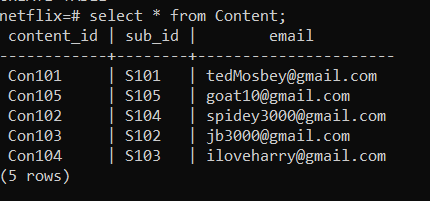
1. **Profile:-**



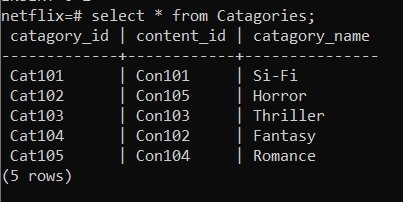
1. **LikedContent**:-



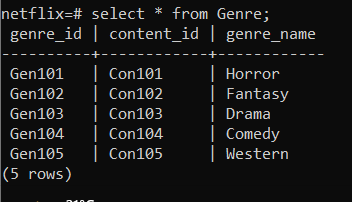
1. **Content**:-



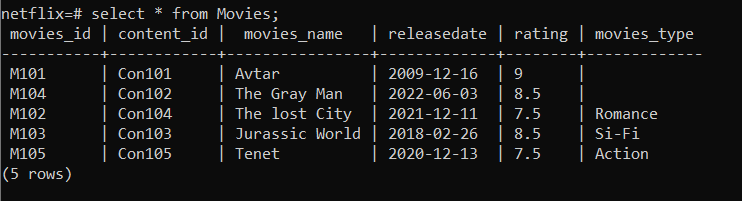
1. **Categories**:-



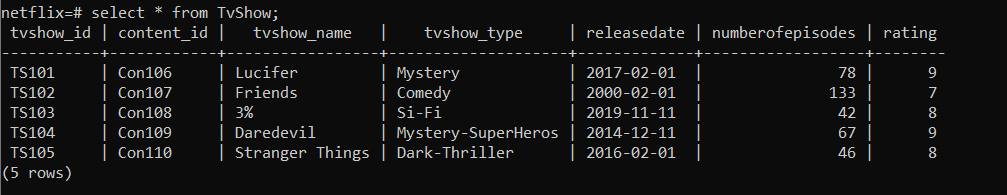
1. **Genre**:-



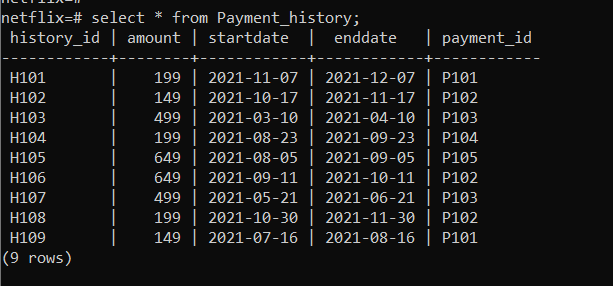
1. **Movies:-**



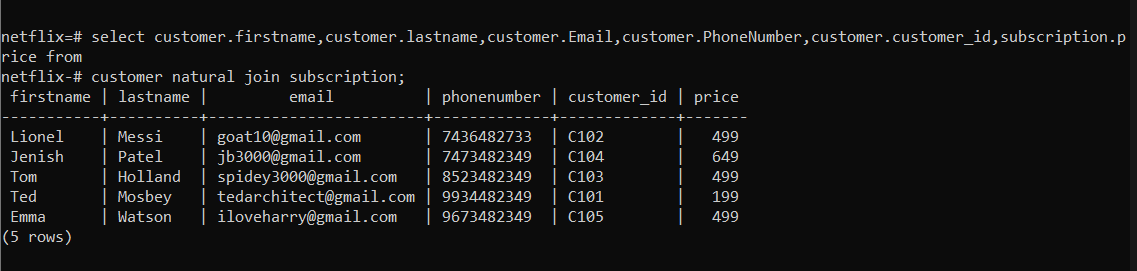
1. **Tvshow:-**

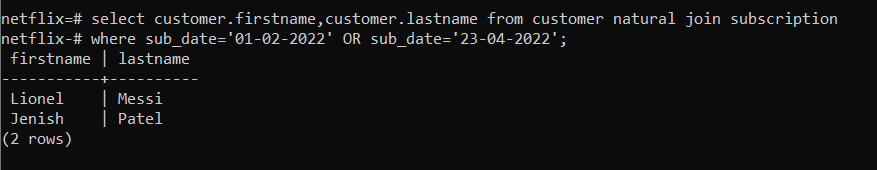


1. **Payment History:-**

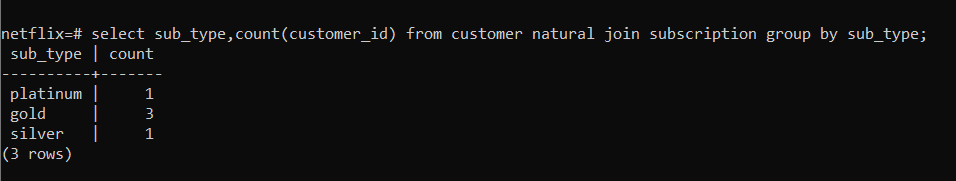


**5.3) Queries using Basic DBMS Constraints , Joins & Subqueries.**

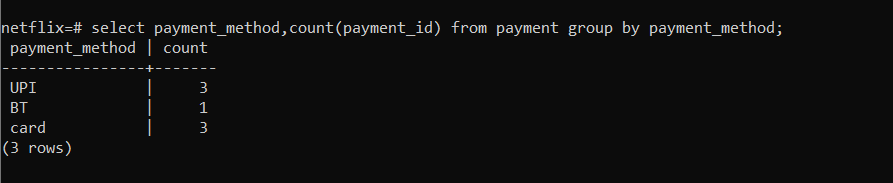
1. Display Customers details who have taken a Subscription with their Subscription plan price.
2. Find the number of customers who have done payment of subscription on specific date and display their names.



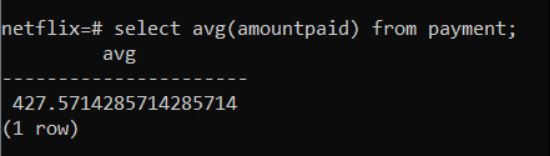
1. Display the number of customers for a specific subscription plan type.



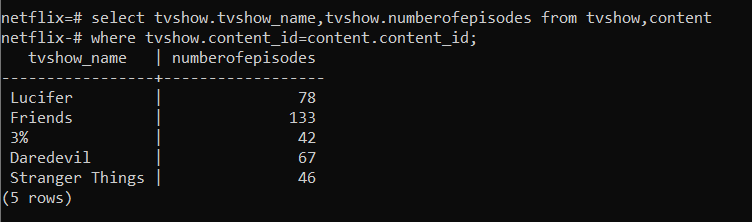
1. Display the count of persons who has done payment with same payment Method.



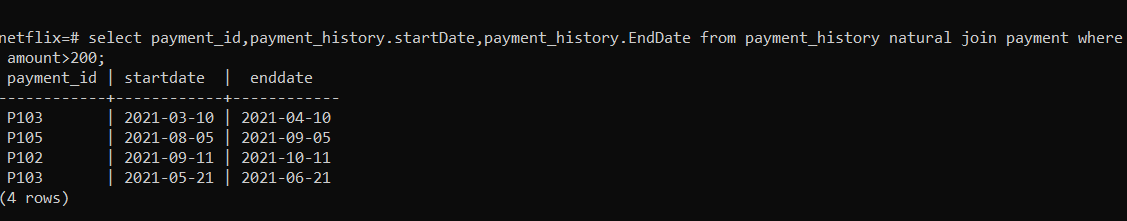
1. Display the Average amount of payment done by customers.

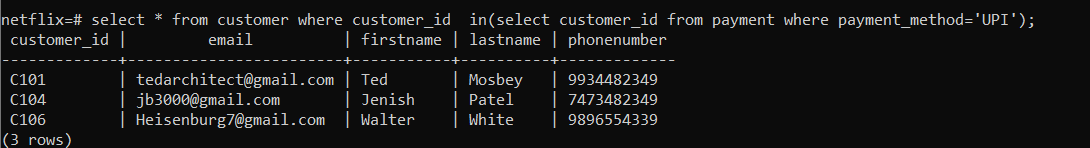


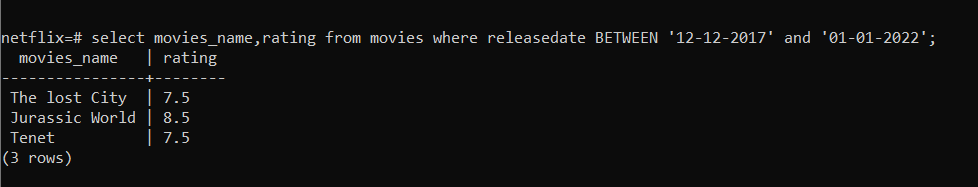
1. Display total number of episodes according to the TVShow.



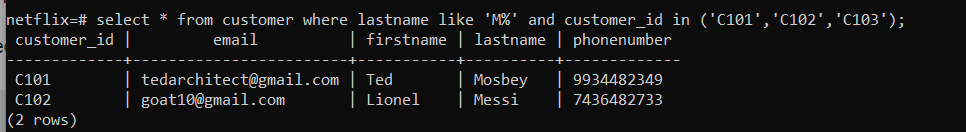
1. Display Start Date & End date of a Subscription plan where payment is done in past with same payment Id, who has paid more than 200.



1. Display details of a Customers who has done payment with ‘UPI’.
2. Display movies Name & Rating of a movies which is released after 2017 & Before 2022 and Has a movie type.



1. Display Customers Details who has last name starting with ‘M’ & customer Id as C101,C102,C103.

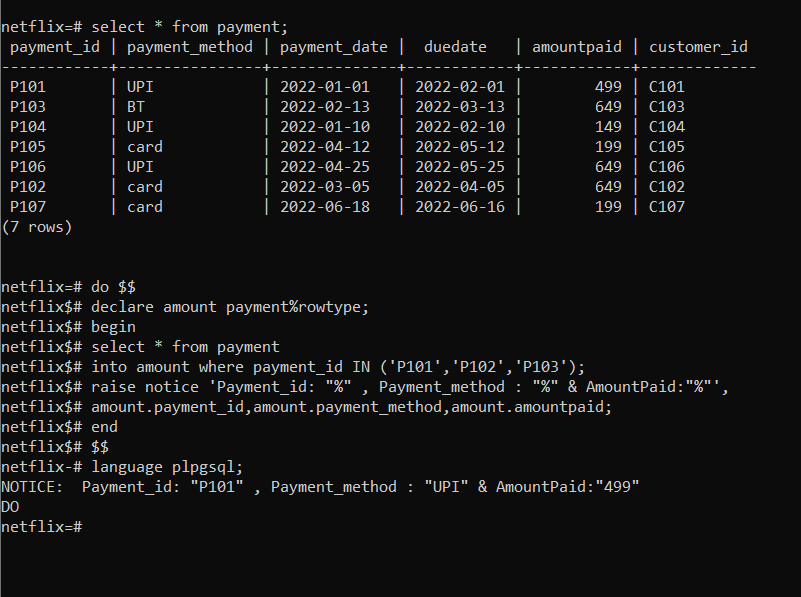


**5.4) PL/SQL Blocks(Views)**

1. **View**

# 

1. RowType

**s**

**5.5) Functions & Triggers**

1. **Create a function & Trigger for invalid entry Of a total numbers of episodes in a TV-Show.**

**FUNCTION:-**

**create or replace function valid\_numofEpi() returns trigger as $num\_episodes$**

**BEGIN**

**if(NEW.numberofEpisodes<0) then**

**raise exception 'This tvshowdont exist if number of episodes is less than 0';**

**end if;**

**return NEW;**

**END;**

**$num\_episodes$**

**LANGUAGE plpgsql;**

**TRIGGER:**

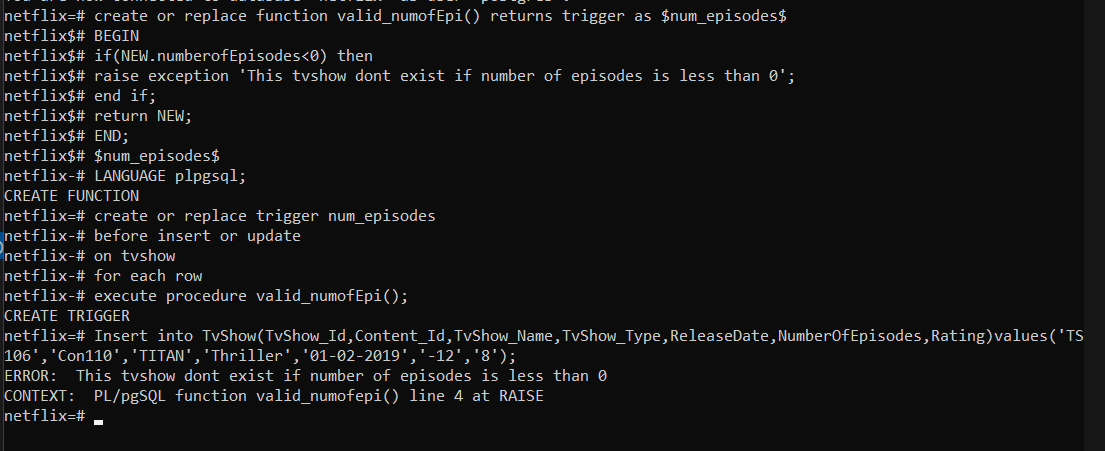
**create or replace trigger num\_episodes**

**before insert or update**

**on tvshow**

**for each row**

**execute procedure valid\_numofEpi();**



**2.) Create a function & Trigger for user make Payment of Invalid amount like 300 because there are no plan with this amount.**

**FUNCTION:-**

**create function valid\_plan() returns trigger as**

**BEGIN**

**if (NEW.amountpaid != 149 or NEW.amountpaid != 199 or NEW.amountpaid != 499 or NEW.amountpaid != 649) then**

**raise exception 'You have selected incorrect plan’;**

**end if;**

**return NEW;**

**END;**

**$$**

**LANGUAGE plpgsql;**

**TRIGGER:-**

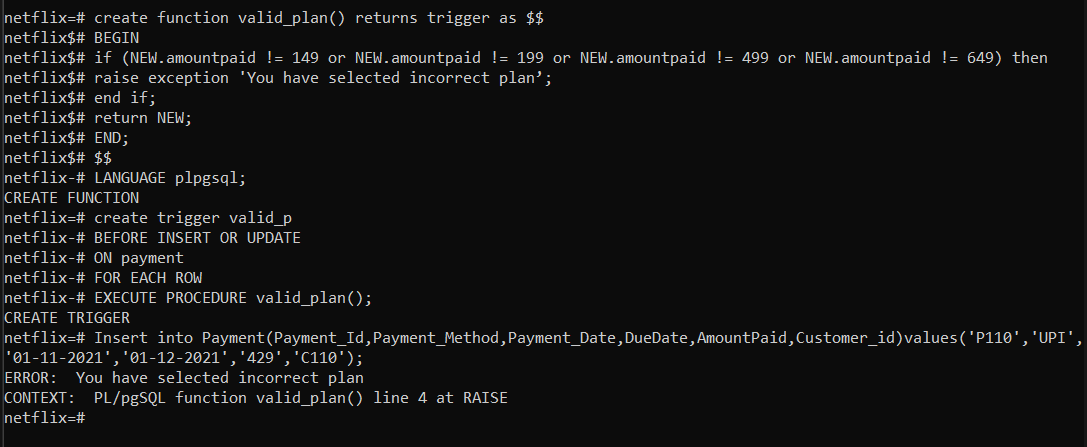
**create trigger valid\_p**

**BEFORE INSERT OR UPDATE**

**ON payment**

**FOR EACH ROW**

**EXECUTE PROCEDURE valid\_plan();**



**5.6) Cursor:-**

1. **Create a Cursor which traverses through a payment table where & payment method is ‘UPI’**.

**CURSOR:-**

**BEGIN;**

**DECLARE mycursor CURSOR for**

**select \* from payment where payment\_method= 'UPI';**

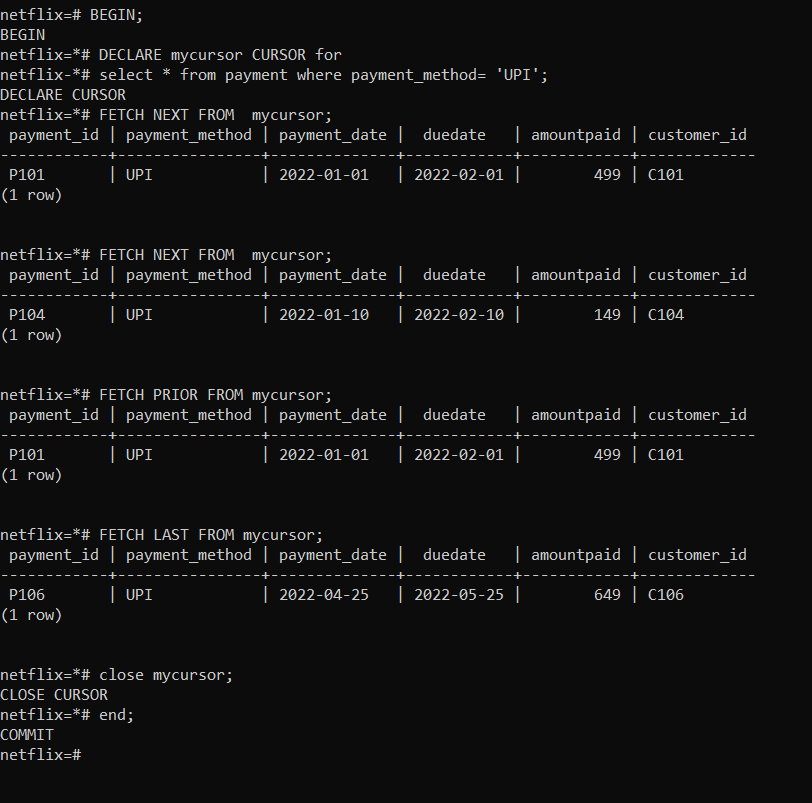
**FETCH NEXT FROM mycursor;**

**FETCH PRIOR FROM mycursor;**

**FETCH LAST FROM mycursor;**

**CLOSE mycursor;**

**end;**



6.) FUTURE ENHANCEMENTS OF THE SYSTEM

* We will design Front-end using React Framework and Develop Back- end in NodeJS.
* Methods and user data input will be a lot easy after the implementation of GUI.
* In the future, we can place the system on the cloud so the maintenance of the data can be reduced.

**7. BIBLIOGRAPHY**

* We created ER-Model on Whimsical and Relational Schema on MySQL WorkBench.
* ER-MODEL -

[**https://whimsical.com/YW63bK8pU6HZXs7F4h2YoD**](https://whimsical.com/YW63bK8pU6HZXs7F4h2YoD)

* For the implementation of this project, we referred to materials shared by Prof. Archana N. Vyas and the following websites and books:

# Book:

Database System Concepts

-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991

# Websites:

* <https://www.w3schools.com/sql/sql_syntax.asp>
* <https://www.tutorialspoint.com/>
* <https://dev.mysql.com/doc/>
* [https://www.geeksforgeeks.org/introduction-of-dbms-database-](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/) [management-system-set-1/](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/)