**DATABASE MGMT & APPLICATIONS PROJECT**

**NETFLIX DATABASE**

Report Summary

The Netflix entertainment corporation, which provides online media streaming, is the subject of this project report. This report analyses a business model and the organization's actions from a business aspect, as well as the company's history and business rules. It also describes Netflix's offerings as well as the target audience. In addition, this paper defines the company's important business models in terms of many perspectives provided as business rules.

The DDL (Data Definition Language) and DML (Data Manipulation Language) programming languages are discussed in this paper to assist in the building of a database for the company.

Index

Introduction 4

Requirements 5

Entity-Relationship Diagram 6

Relational Model 7

Model Description 9

MySQL Implementation 12

**Table Insertions** 14

**Tables:** 18

**SQL Queries** 22

**Conclusion** 27

Introduction

Netflix has a global subscriber base of over 200 million people who watch movies and TV shows online. Customers can watch as many shows as they like as long as they have an internet connection and a monthly subscription that starts at $10. It also creates original material and licenses feature films and television shows for streaming.

This project is about creating a Netflix database. It is a well-known online streaming service. The business is based in California and operates from the same location. The company's main business is its subscription-based streaming service, which allows users to watch a collection of films and television shows, including those produced in-house, online.

This report gives a quick summary.

Requirements

This application mostly contains information about users/customers, as well as information about tv series and movies. The following are the business rules and cardinalities for the database.

1. Customer ID, name, email, address, and phone number are all stored in the user account.
2. The subscription ID, subscription type, price, plan information, subscription date, and expiration date are all included in the subscription. A subscription can have many customers, a customer should have one subscription.
3. Payment made by the customer includes a payment ID, payment method, amount, payment date, and card information.
4. Each movie or tv series has a title, a year of release, a genre(s), an IMDB rating, a director's name, a producer's name, and a runtime.
5. Each video should be a movie or a television series. If the video is a TV series, the number of seasons is recorded; otherwise, only the title and length of the film are recorded.
6. This platform has a unique feature known as Xray. It includes information about the actors and actresses who appeared in the film, as well as information about the film itself, including any accolades received.
7. This platform will establish a profile for a player (actor or actress) who has been in a film. Each player is assigned a unique profile ID, as well as their name, birth date, birthplace, and the name of the character they played in previous movies.
8. Xray also shows whether the movie or the character has received any awards. The award contains the name of the award and the year in which it was given out. If it is given for a movie, it contains the title of the film; if it is given for a player, it contains the name of the player.
9. A single movie or tv series can feature one or many casts, while a single cast can appear in multiple films.
10. Every movie or tv series must be included in any subscription. Since Netflix does not provide any free shows to those who do not have a Netflix account.

Entity-Relationship Diagram

An entity-relationship diagram depicts the relationships between entity sets maintained in a database (ERD). In this context, an entity is an object or a data component. The term "entity set" refers to a group of comparable entities. These entities can have properties that define their qualities.

Diagram

Description automatically generated

Relational Model

Diagram

Description automatically generated

Model Description (Metadata)

We construct metadata for the tables listed in the logical representation model now that the ER Diagram and Logical Representation model are complete. We begin by creating a table with no foreign key associations. The metadata tables that describe the attribute information are listed below.

**Content table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| content\_id | Int |  | Yes | No | Primary Key |
| title | Varchar | 30 | No | No |  |
| release\_date | Date |  | No | No |  |
| genre | Varchar | 30 | No | No |  |
| Imdb\_rating | Float |  | No | Yes |  |
| director\_name | Varchar | 30 | No | Yes |  |
| producer\_name | Varchar | 30 | No | Yes |  |
| run\_time | Int |  | No | No |  |
| tv\_shows | Varchar | 30 | No | No |  |
| movies | Varchar | 30 | No | No |  |

**Customer table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| cust\_id | Int |  | Yes | No | Primary Key |
| cust\_fname | Varchar | 30 | No | No |  |
| cust\_lname | Varchar | 30 | No | No |  |
| customer\_email | Varchar | 100 | No | No |  |
| address | Varchar | 30 | No | No |  |
| city | Varchar | 20 | No | No |  |
| State | varchar | 20 | No | No |  |
| Zipcode | varchar | 10 | No | No |  |
| state\_code | Varchar | 20 | No | Yes |  |
| customer\_phone | Double |  | No | No |  |
| Plan\_id | Int |  | No | No | Foreign Key |
| subscription\_id | Double |  | No | No | Foreign Key |

**Actor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| actor\_id | Int | 11 | Yes | No | Primary Key |
| actor\_fname | Varchar | 50 | No | No |  |
| actor\_lname | Varchar | 50 | No | No |  |
| date | Date | Default | No | Yes |  |
| awards | Varchar | 50 | No | Yes | Foreign Key |

**Award**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| award\_type | Varchar | 100 | No | No | Primary Key |
| award\_name | Varchar | 100 | No | No |  |

**Movie**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| movie\_id | Int |  | Yes | No | Primary Key |
| Mname | Varchar | 100 | No | No |  |
| Mduration | Int |  | No | No |  |
| mdirector\_name | Varchar | 30 | No | Yes |  |
| mproducer\_name | Varchar | 30 | No | Yes |  |
| maward | Varchar | 30 | No | Yes | Foreign Key |
| award\_year | Date |  | No | Yes |  |
| mcontentid | Int |  | No | No | Foreign Key |

**Payment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| payment\_id | Varchar | 30 | No | No | Primary Key |
| payment\_method | Varchar | 20 | No | No |  |
| billing\_date | Date |  | No | No |  |
| amount | Float |  | No | No |  |
| cust\_id | Int |  | No | No | Foreign Key |
| subscription\_id | Double |  | No | No | Foreign Key |

**Subscription**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| subscription\_id | Double |  | Yes | No | Primary Key |
| subscription\_type | Varchar | 100 | No | No | Foreign Key |
| price | Double |  | No | No |  |
| start\_date | Date |  | No | No |  |
| expiry\_date | Date |  | No | No |  |

**Subscription Plans**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| plan\_id | Int |  | No | No | Primary Key |
| plan\_name | Varchar | 100 | No | No | Unique Key |
| plan\_price | Varchar | 100 | No | Yes |  |
| plan\_vedio\_quality | Varchar | 100 | No | Yes |  |

**TV Shows**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | Data Type | Size | Auto | Null | Constraint |
| show\_id | Int |  | Yes | No | Primary Key |
| show­\_name | Varchar | 100 | No | No |  |
| total\_seasons | Int |  | No | Yes |  |
| total\_episodes | varchar | 30 | No | No |  |
| episode\_duration | Int |  | No | Yes |  |
| show\_director | Varchar | 30 | No | Yes |  |
| show\_producer | Varchar | 30 | No | Yes |  |
| show\_award | Varchar | 30 | No | Yes | Foreign key |
| show\_award\_year | Date |  | No | Yes |  |
| show\_content\_id | Int |  | No | Yes | Foreign key |

MySQL Implementation

We can begin with SQL queries once the metadata for all ten tables has been created. We'll begin by creating the database and tables for the same.

Data Definition Language (DDL) queries

Creating the database tables using DDL queries in MYSQL,

**/\* Creating the Database \*/**

CREATE DATABASE Netflix\_Database;

**/\* Using the created Netflix Database \*/**

USE Netflix\_Database;

**/\* Creating user table \*/**

create table customer(

CustID int(20) not null AUTO\_INCREMENT,

cust\_fname varchar(30) not null,

cust\_lname varchar(30) not null,

customer\_email varchar(100) not null,

address varchar(30) not null,

city varchar(20) not null,

state varchar(20) not null,

zipcode varchar(10) not null,

state\_code varchar(20) ,

customer\_phone double not null,

PlanID int(10) not null,

subscription\_id double not null,

CONSTRAINT customer\_pk PRIMARY KEY (CustID),

CONSTRAINT customer\_fk FOREIGN KEY (PlanID) REFERENCES Subscription\_Plans(PlanID) ON UPDATE CASCADE ON DELETE CASCADE,

constraint customer\_fk2 foreign key(subscription\_id) references subscription(subscription\_id) ON UPDATE CASCADE ON DELETE CASCADE);

**/\* Creating content table \*/**

create table content(

content\_id int(20) not null AUTO\_INCREMENT,

title varchar(30) not null,

release\_date date not null,

genre varchar(30) not null,

imdb\_rating float(10) not null,

director\_name varchar(30) ,

producer\_name varchar(30) ,

run\_time int(5) not null,

tvshows varchar(30) not null,

movies varchar(30) not null,

CONSTRAINT content\_pk PRIMARY KEY (content\_id));

**/\* Creating Movie Table \*/**

CREATE TABLE movie (

MovieID int(10) NOT NULL AUTO\_INCREMENT,

Mname varchar(100) NOT NULL,

Mduration int NOT NULL,

mdirector\_name varchar(30) DEFAULT NULL,

mproducer\_name varchar(30) DEFAULT NULL,

maward varchar(30) DEFAULT NULL,

award\_year date DEFAULT NULL,

mcontentid int(20)

CONSTRAINT movie\_pk PRIMARY KEY (MovieID),

CONSTRAINT movie\_fk FOREIGN KEY (maward) REFERENCES Awards (award\_type)

ON DELETE CASCADE ON UPDATE CASCADE,

constraint movie\_fk2 foreign key(mcontentid) references content(content\_id)

ON DELETE CASCADE ON UPDATE CASCADE);

/\* Creating Subscription Table \*/

create table subscription(

subscription\_id double not null AUTO\_INCREMENT,

subscription\_type varchar(100) not null,

price double not null,

start\_date date not null,

expiry\_date date not null,

CONSTRAINT subscription\_pk PRIMARY KEY (subscription\_id));

alter table subscription modify subscription\_type varchar(100) not null;

alter table subscription add CONSTRAINT subscription\_fk foreign key(subscription\_type)

references subscription\_plans(PlanName)

on update cascade on delete cascade);

/\* Creating Subscription Plans table \*/

CREATE TABLE Subscription\_Plans  (

PlanID INT(10) NOT NULL,

PlanName VARCHAR(100) NOT NULL unique,

PlanPrice VARCHAR(100) NOT NULL,

PlanVideoQuality VARCHAR(100) NOT NULL,

CONSTRAINT Subscription\_Plans\_pk PRIMARY KEY (PlanID));

/\* Creating Awards table \*/

create table Awards(

award\_type varchar(100) not null,

award\_name varchar(100) not null,

constraint awards\_pk primary key (award\_type));

/\* Creating TV Shows table\*/

create table TVShows(

showID int(10) not null auto\_increment,

showname varchar(100) not null,

total\_seasons int(10) ,

total\_episodes varchar(30) not null,

episode\_duration int(10),

show\_director varchar(30),

show\_producer varchar(30),

show\_award varchar(30),

show\_award\_year date,

showcontentid int(20),

constraint tvshows\_pk primary key (showID),

constraint tvshows\_fk foreign key (show\_award) references Awards(award\_type)

ON UPDATE CASCADE ON DELETE CASCADE,

constraint tvshows\_fk2 foreign key(showcontentid) references content(content\_id)

ON UPDATE CASCADE ON DELETE CASCADE);

/\* Creating Payment table \*/

create table payment(

payment\_id varchar(30) not null ,

payment\_method varchar(20) not null,

billing\_date date not null,

amount float(10) not null,

custid int(20) not null,

subscription\_id double not null,

CONSTRAINT payment\_pk PRIMARY KEY (payment\_id),

CONSTRAINT payment\_fk1 FOREIGN KEY (custid) REFERENCES customer (custid)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT payment\_fk2 FOREIGN KEY (subscription\_id) REFERENCES subscription (subscription\_id)

ON DELETE CASCADE ON UPDATE CASCADE);

/\* Creating Actor table \*/

create table actor(

actor\_id double not null auto\_increment,

actor\_fname varchar(50) not null,

actor\_lname varchar(50) not null,

DOB date,

awards varchar(50),

constraint actor\_pk primary key (actor\_id),

CONSTRAINT actor\_fk1 FOREIGN KEY (awards) REFERENCES awards (award\_type)

ON DELETE CASCADE ON UPDATE CASCADE);

**Table Insertions**

#Inserting values into subscription plans table

insert into subscription\_plans(planid, planname, planprice, planvideoquality) values

(1, 'Basic', '9.99', '480p'),

(2,'Standard','15.49','1080p'),

(3,'Premium','19.99','4K UHD');

#Inserting values into subscription table

insert into subscription(subscription\_type, price, start\_date,expiry\_date) values

('Standard','19.99','2021-01-02','2021-02-01'),

('Basic','9.99','2021-05-31','2021-06-30'),

('Premium','19.99','2021-02-21','2021-03-23'),

('Premium','19.99','2021-12-17','2022-01-16'),

('Premium','19.99','2022-01-27','2022-02-26'),

('Basic','9.99','2022-02-13','2021-03-15'),

('Standard','15.49','2021-10-21','2021-11-20'),

('Basic','9.99','2021-12-17','2022-01-16'),

('Basic','9.99','2022-03-21','2022-04-20'),

('Premium','19.99','2022-04-30','2022-05-29'),

('Basic','9.99','2021-07-21','2021-08-20'),

('Basic','9.99','2021-02-21','2021-03-23'),

('Standard','15.49','2021-01-02','2021-02-01'),

('Premium','19.99','2022-01-26','2022-02-25'),

('Premium','19.99','2022-02-13','2021-03-15'),

('Standard','15.49','2021-09-21','2021-10-20'),

('Standard','15.49','2021-12-17','2022-01-16'),

('Basic','9.99','2022-03-21','2022-04-20'),

('Basic','9.99','2022-03-31','2022-04-30');

#Inserting into Payment table

insert into payment(payment\_method,billing\_date,amount,CustID,subscription\_id) values

('Visa','2021-01-02','19.99',1000000001,1),

('Master','2021-05-31','9.99',1000000002,2),

('Discover','2021-02-21','19.99',1000000003,3),

('Visa','2021-12-17','19.99',1000000004,4),

('Master','2022-01-27','19.99',1000000005,5),

('Discover','2022-02-13','9.99',1000000006,6),

('Visa','2021-10-21','15.49',1000000007,7),

('Master','2021-12-17','9.99',1000000008,8),

('Discover','2022-03-21','9.99',1000000009,9),

('Visa','2022-04-30','19.99',1000000010,10),

('Master','2022-04-30','19.99',1000000011,11),

('Discover','2021-07-21','9.99',1000000012,12),

('Visa','2021-02-21','9.99',1000000013,13),

('Master','2021-01-02','19.99',1000000014,14),

('Discover','2022-01-26','19.99',1000000015,15),

('Visa','2022-02-13','19.99',1000000016,16),

('Master','2021-09-21','15.49',1000000017,17),

('Discover','2021-12-17','15.49',1000000018,18),

('Visa','2022-03-21','9.99',1000000019,19),

('Master','2022-03-31','9.99',1000000020,20);

#Inserting into Awards table

insert into awards(award\_type,award\_name) values

('best\_actor', 'Best Actor'),

('best\_actress', 'Best Actress'),

('best\_director','Best Director'),

('best\_music', 'Best Music Director'),

('best\_screenplay', 'Best TV Show Actress'),

('best\_effects','Best Visual Effects'),

('best\_bol', 'Best Background Original Score'),

('best\_cdesign', 'Best Costume Design'),

('best\_aff','Best Animated Feature Film'),

('best\_editing','Best Edting'),

('best\_FLF','Best Foreign Language Film')

#Inserting into Customer Table

insert into customer(cust\_fname,cust\_lname,address ,customer\_email,customer\_phone,city,state,zipcode,PlanID,subscription\_id) values

/\*( 'Abhinay', 'Parasa', '1610 Holly Circle', 'aparasa@gmail.com', '9259996571', 'Pleasanton', 'California',94566, 3, 1)\*/

( 'Krushi', 'Padamati', '40 Salisbury Way', 'kpadamati@gmail.com', '6413255770', 'Hayward', 'California',94542, 1, 2),

( 'Dileep',  'Kumar', '8 Janacourt', 'edkumar@yahoo.com', '5413255230',  'Fremont', 'California',94536, 1, 3),

( 'Kirsti', 'Gwillym', '8 OxfordCrossing', 'KirstiG@outlook.com', '5013259930', 'London', 'United Kingdom',5211, 2, 4),

( 'Terrence', 'Lewis', '5 DwereyPark', 'Terryrl@yahoo.com', '9112345876' , 'Scotsdale', 'Arizona',85054, 3, 5),

( 'Lazaro',  'Croster', '94542 Carlos Bee Blvd', 'Lazacross@yahoo.com', '3413259930', 'Hayward', 'California',94542, 3, 6),

( 'Hari',  'Rao', '333 ManhattanSt', 'hariprasad@yahoo.com', '2313259930', 'Queens', 'New York',11426, 2, 7),

( 'Naina', 'Reddy', '881 Fremont ln', 'nainareddy@gmail.com', '5123252113', 'Sunnyvale', 'California',94085, 2, 8),

( 'Samhitha', 'Reddy', '511 Newtown', 'reddysamhitha@yahoo.com', '2453259975', 'Jacksonville', 'Florida',32034, 1, 9),

( 'Anne',  'Watson', '1630 Holly Cir ', 'Annehw@yahoo.com', '8913259233', 'Pleasanton', 'California',94566, 1, 10),

('Iwette', 'David', '30 Autumnpark ', 'Iwetteb@yahoo.com', '7613253450', 'Seattle', 'Washington',98063, 2, 11),

('Paul', 'Hawkins', 'Melrosejunction', 'Stephenhawk@yahoo.com', '5012259321', 'Montreal', 'Quebec','H0HH9X', 1, 12),

('Olympie', 'Golborn', '22 Jasperst', 'Olympus@gmail.com', '6752593442', 'OlympGlb', 'Alabama',35242, 3, 13),

('Sam', 'Mane',   '912 BStreet,', 'Manesam@yahoo.com', '5589399391', 'San Jose', 'California',94088, 3, 14),

('Danna',  'Chatterjee', '65 Larrypoint', 'chatterjee@yahoo.com', '3213257938', 'Anchorage', 'Alaska',99501, 3, 15),

('Devon',  'Bollis', 'IndianSt,', 'dheerick@yahoo.com', '5338259956', 'Austin', 'Texas','73301', 1, 16),

('Val',  'Champken', '33 FlorenceStreet', 'champkenval@yahoo.com', '4453251976', 'Berlin', 'Germany',14199, 2, 17),

('Melanie', 'Clinton', 'Nottinghill', 'melaniec@yahoo.com', '5313256630', 'Brooklyn', 'New York', 11207, 1, 18),

('Rowela',  'Johnson', 'Elak lane,', 'johnsonowela@yahoo.com', '5119859965', 'Cleveland', 'Ohio',44101, 1, 19),

('Pooja', 'Agarwal', '221 Fremont', 'Poojaagarwal@yahoo.com', '4323257632', 'Fremont', 'California',94536, 3, 20);

#Inserting into Actor table

insert into actor(actor\_fname, actor\_lname, DOB) values

('Amber','Heard','1986-04-22'),

('Jason','Momoa','1979-08-01'),

('Channing','Tatum','1980-04-26'),

('Alexandra','Daddario','1986-03-16'),

('Angelina','Jolie','1975-06-04'),

('Kristen','Stewart','1990-04-09'),

('Tom','Cruise','1962-07-03'),

('Kevin','Hart','1979-07-06'),

('Chadwick','Boseman','1976-11-29'),

('Tom','Holland','1996-06-01'),

('Tom','Hiddleston','1981-02-09'),

('Jennifer','Aniston','1969-02-11');

#Insert into Movie table

insert into movie(Mname, Mduration) values

('The Adam Project','2022-03-09','140'),

('The Social Network','120'),

('Spectre','135'),

('Rambo','91'),

('83','160'),('Rambo: Last Blood','89'),

('Sanju','158'),

('Veronica','105'),

('The Conjuring','111'),

('Alive','98'),

('Annabelle Creation','109'),

('The Silence','90'),

('The Ritual','94');

#Inserting data into TV Shows table

insert into tvshows( showname, total\_seasons, total\_episodes, episode\_duration, show\_director, Show\_producer,

show\_award, show\_award\_year, showcontentid) values

('Ozark', 3 , 44, '60', 'Adam', 'Ben', 'best\_actor', '1997-01-12', 100014),

('Space Force', 1, 3, '30', 'Rock', 'Hill', 'best\_director', '2020-02-11', 100015),

('Stranger Things','2', 10 , '40', 'Jazero', 'Laura', 'best\_music', '2011-03-09', 10005),

('The Vampire Dairies','10', 30, '75', 'Robin', 'Christine', 'best\_aff', '2012-04-10', 100019),

('Flash', 5 , 50, '25', 'Bill', 'Tim', 'best\_actor', '2022-06-16', 100020),

('Arrow', 7 , 70, '35', 'Obama', 'Cook', 'best\_music', '2000-11-19', 100021),

('Money Heist', 3 , 30, '40', 'Trump', 'Brook', 'best\_actress', '1998-12-12', 10006),

('Lucifer', 6 , 60, '60', 'Treudeu', 'Modi', 'best\_effects', '2015-01-31', 10009),

('Squid Game', 9, 25, '70', 'Michel', 'Jagan', 'best\_bol', '2018-05-30', 100022);

#Inserting data into content table

insert into content(title,release\_Date,genre,imdb\_rating,director\_name,producer\_name,run\_time,tvshows,movies) values

('The Adam Project','2022-03-09','Adventure','6.7','Bill','John','140','N','Y'),

('The Social Network','2010-12-11','History','7.8','Stefan','Peterson','120','N','Y'),

('Spectre','2018-01-21','Sci-fi','7','Andy','Peterman','135','N','Y'),

('Rambo','2009-11-30','Action','6.9','Robert','Joseph','91','N','Y'),

('Stranger Things','2010-12-11','Horror','8.4','Jazero', 'Laura','25','Y','N'),

('Money Heist','2019-04-29','Thirller','7.9','Trump', 'Brook','48','Y','N'),

('Veronica','2020-10-12','Horror','6.1','Klark','Andrew','105','N','Y'),

('The Conjuring','2017-07-19','Horror','6.1','Bruce','Alfred','111','N','Y'),

('Lucifer','2016-11-30','Drama','8','Treudeu', 'Modi',93,'Y','N'),

('Alive','2013-02-25','Adventure','5.9','Jarvis','Richard','98','N','Y'),

('Annabelle Creation','2017-08-19','Horror','6.7','Connor','David','109','N','Y'),

('The Silence','2018-05-21','Horror','6.7','Cook','Warner','90','N','Y'),

('The Ritual','2018-11-29','Horror','7.1','Mitchell','Stark','94','N','Y'),

('Ozark','2015-07-21','Drama','7','Adam', 'Ben',44,'Y','N'),

('Space Force','2018-09-21','Comedy','7.8','Rock','Hill', 17,'Y','N'),

('83','2015-07-21','Drama','6.3','Ranveer','Shahrukh','160','N','Y'),

('Rambo: Last Blood','2013-01-31','Action','5.8','Robert','Daemon','89','N','Y'),

('Sanju','2020-11-21','Drama','6.9','Aamir','Karan','158','N','Y'),

('The Vampire Dairies','2008-11-11','Horror','7.8','Robin', 'Christine','171','Y','N'),

('Flash','2007-11-28','Drama','7.5','Bill', 'Tim','151','Y','N'),

('Arrow','2008-12-11','Drama','7.3','Obama', 'Cook','170','Y','N'),

('Squid Game','2021-03-11','Thriller','8.1','Michel', 'Jagan',9,'Y','N');

**Tables:**

**Customer Table**

Graphical user interface, text, table

Description automatically generated

**Content Table**

Table

Description automatically generated

**Movie Table**

Table

Description automatically generated

**Subscription Table**

Table

Description automatically generated

**Subscription Plans Table**

Table

Description automatically generated

**Awards Table**

Graphical user interface, application, table

Description automatically generated

**TV Shows Table**

Graphical user interface

Description automatically generated

**Payment Table**

Table

Description automatically generated

**Actor Table**

Table

Description automatically generated

**SQL Queries**

**Query 1:** In this query we combined customer, payment, subscription, subscription plans tables to show all the customers and their subscriptions on Netflix

#Cross joning 4 tables

SELECT \* FROM

customer

JOIN

payment

JOIN

subscription

JOIN

subscription\_plans;

**Table

Description automatically generated**

**Query 2:** In this query we used a view to sort out all the California users who are subscribed to basic plan.

#Create view for California customers having basic plan

CREATE VIEW CaliBasic AS

SELECT

c.custid, c.cust\_fname, c.cust\_lname, s.subscription\_type

FROM

customer c

JOIN

subscription s ON c.subscription\_id = s.subscription\_id

JOIN

subscription\_plans sp ON s.subscription\_type = sp.planname

WHERE

c.state = 'California'

AND sp.planname = 'Basic';

select \* from calibasic;

**Table

Description automatically generated**

**Query 3:** In this query we created a view to see all the users who are not from California.

#Creating a view to see all users having all type subscription that are not from California

CREATE VIEW NonCaliUsers AS

SELECT

c.custid,

c.cust\_fname,

c.cust\_lname,

c.state,

s.subscription\_type

FROM

customer c

JOIN

subscription s ON c.subscription\_id = s.subscription\_id

JOIN

subscription\_plans sp ON s.subscription\_type = sp.planname

WHERE

c.state != 'California';

select \* from NonCaliUsers;

**Table, Excel

Description automatically generated**

**Query 4:** In this query we created a stored procedure which gives the customer details ordered by customers first name.

#creating a stored procedure

DELIMITER //

CREATE PROCEDURE payment()

BEGIN

sELECT c.cust\_fname,c.cust\_lname,p.payment\_id, p.custid, p.amount, p.subscription\_id FROM

payment p join customer c on p.custid=c.custid

WHERE p.amount >'10'

ORDER BY p.amount ;

END//

DELIMITER ;

call payment;

**Table

Description automatically generated**

**Query 5:** In this query we wrote a stored procedure to get content that has IMDB rating more than 6.5

#Stored Procedure

DELIMITER //

CREATE PROCEDURE ContentRating()

BEGIN

SELECT \* from content

WHERE imdb\_rating>'6.5'

ORDER BY imdb\_rating ;

END//

DELIMITER ;

call ContentRating;

**Table

Description automatically generated**

**Query 6:** Query to fetch subscription date less than '2021-10-01'

#Join query to get subscription date less than 2021-10-01

SELECT

c.custid,

c.cust\_fname,

c.state,

c.zipcode,

s.subscription\_type,

s.price,

s.Start\_Date

FROM

customer c

JOIN

subscription s ON c.subscription\_id = s.subscription\_id

JOIN

payment p ON s.subscription\_id = p.subscription\_id

WHERE

p.billing\_date < '2021-10-01';

**Graphical user interface, application, table

Description automatically generated**

**Query 7:** Query to fetch customers having basic plan using join

#Query to fetch the customers who has Basic plan

SELECT

c.custid,

c.cust\_fname,

c.customer\_phone,

p.payment\_method,

p.billing\_date,

p.amount,

sp.planname

FROM

customer c

JOIN

payment p ON c.custid = p.custid

JOIN

subscription s ON p.subscription\_id = s.subscription\_id

JOIN

subscription\_plans sp ON s.subscription\_type = sp.planname

WHERE

planname = 'Basic';

**Table

Description automatically generatedConclusion**

We created a Netflix database that tracks and records customer transactions, as well as the content that is included in subscription type.

In our case, we created a database that records transactional data. That has to do with CRUD operations—creating, reading, updating, and deleting records. The contents are movies, TV Shows from different countries and different genres. We analyzed the business requirements and updated the entity relationships based on them. This database management system is an enhanced ER model with supertype and subtype entities.

Additionally, using MYSQL workbench and SQL queries such as DDL and DML, we built the database design and determined the answers to certain research questions. We developed a better grasp of database design and how it might be applied in real-life scenarios while working on this project.