Anne Melaas and Akwasi Kyei Awuah

**TEAM 15**

A2: Project in Pairs

Hult International Business School

Data Management & SQL - Streaming video platform

Maithily Erande

Due: 01.31.2021

**Table of Contents**

[Flow of System 2](#_Toc63014050)

[Database structure 3](#_Toc63014051)

[Entity-Relational (ER) Model 4](#_Toc63014052)

[10 SQL queries 5](#_Toc63014053)

[1. Which movie genre is the most watched on our platform? 5](#_Toc63014054)

[2. Which series genre is the most watched on our platform? 5](#_Toc63014055)

[3. Who are our top 5 main distributors? 6](#_Toc63014056)

[4. What movie or series is most viewed in kids’ mode? 7](#_Toc63014057)

[5. What is the playtime by producer, by movie and series? 8](#_Toc63014058)

[6. Which country are the majority of the ACTIVE users from? 9](#_Toc63014059)

[7. Which payment plan generates the most revenue? 10](#_Toc63014060)

[8. Identify the least popular movies and series in the last year? 11](#_Toc63014061)

[9. Find the top 3 genres viewed in kids’ mode who are still active users? 13](#_Toc63014062)

[10. Find the most popular genres and average playtime for churning customers to identify potential reasons of leaving and potential reactions to win back the customers. 15](#_Toc63014063)

[2 detailed SQL procedures 17](#_Toc63014064)

[First procedure 17](#_Toc63014065)

[Second procedure 19](#_Toc63014066)

[Appendix: 21](#_Toc63014067)

[Create Table scripts 21](#_Toc63014068)

[Insert scripts: 28](#_Toc63014069)

# Flow of System

The video streaming platform has a worldwide user base which can access a variety of movies and series. The available content includes movies and series from all genres and is frequently updated. Next to third party distributors, self-produced (Netflix) content is offered.

Before watching, users must sign up to the platform. They can choose from different monthly payment plans to get access to the content. A user can cancel their subscription monthly. Furthermore, the user's payment is dependent on their subscription. A kid’s mode can be set on the profile at the “family” profile. Kid´s mode will restrict the visibility of movies and series according to the official maturity rating.

The video streaming platform requests that the new database is easily giving answers to the following business questions:

* *Most popular movie distributor to negotiate a deal for more content to stream on the platform.*
* *Most popular genres for all segments (adults/kids – series/movies) to concisely expand offer.*
* *Least popular movies / series in the last year in order to remove the content and reduce associated costs like licensing right feels.*
* *Revenue created from the prescription model by the active users in the last year for reporting purposes.*
* *Most popular genres and average playtime for churning customers to identify potential reasons of leaving and potential reactions to win back the customers.*
* *Identify the distribution of customers among countries to customize content to specific cultures to see their preferences.*
* *Popularity of self-produced movies and series.*

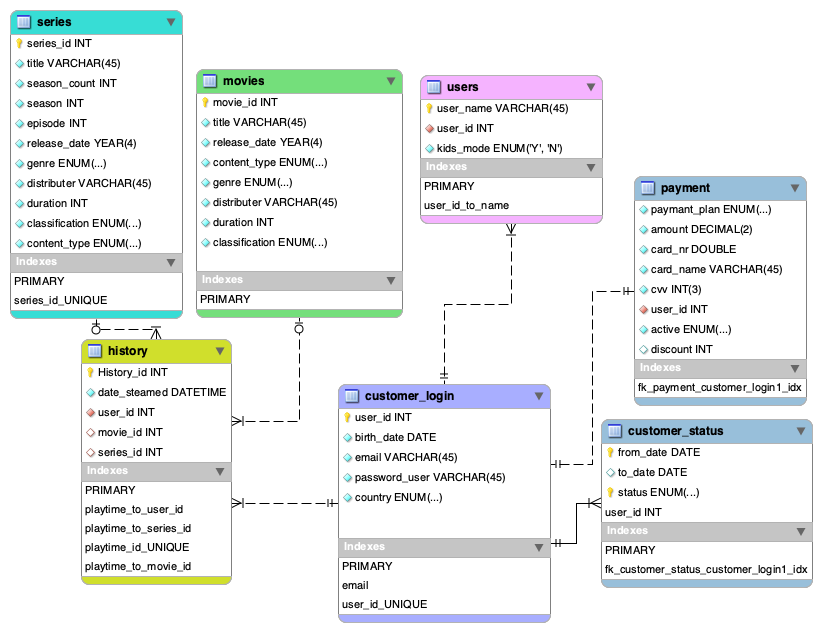
To answer these business questions, user data must relate to the movie and series data. It is important to record the time and frequency a user watches content to identify the popularity of different movies, series, or genres. Furthermore, confidential information such as personal or payment related information should be stored separately to ensure data security. This enables a correct allocation of stream time to the different points of analysis.

# Database structure

The requirements of the business user have been thoroughly analyzed. The following data structure has been created:

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Primary Key | Foreign Key | Description |
| user | user\_name | user\_id | Shows how the customer is and if they have “kids” mode on or not. |
| customer\_Login | user\_id |  | Stores customer personal information. |
| history | history\_id | user\_id, movie\_id, series\_id | An overall storage of all information of what movies and series have been watched of the user. |
| movies | movie\_id |  | Stores all movie information |
| Series | series\_id | content\_type | Stores all series information |
| Payment |  | user\_id | Stores all payment information |
| coustomer\_status | from\_date, status |  | Stores the history of active and inactive users. |

# Entity-Relational (ER) Model



# 10 SQL queries

## 1. Which movie genre is the most watched on our platform?

SELECT

m.genre AS top\_genre\_movie, COUNT(1) AS num\_reprod

FROM

movies AS m

JOIN

history AS h ON m.movie\_id = h.movie\_id

GROUP BY

m.genre

ORDER BY

COUNT(1) DESC

LIMIT 1;

**Answer:**



## 2. Which series genre is the most watched on our platform?

SELECT

s.genre AS top\_genre\_series, COUNT(1) AS num\_reprod

FROM

series AS s

JOIN

history AS h ON s.series\_id = h.series\_id

GROUP BY

s.genre

ORDER BY

COUNT(1) DESC

LIMIT 1;

**Answer:**

****

## 3. Who are our top 5 main distributors?

SELECT

top\_distributors.dis AS distributor, COUNT(1) AS num\_content

FROM

(SELECT

m.distributer AS dis, m.title

FROM

movies AS m

UNION ALL

SELECT

s.distributer AS dis, s.title

FROM

series AS s

GROUP BY

s.title, s.distributer) AS top\_distributors

GROUP BY

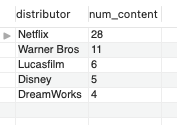
distributor

ORDER BY

num\_content DESC

LIMIT 5;

**Answer:**



## 4. What movie or series is most viewed in kids’ mode?

SELECT

top\_stream.\*

FROM

((SELECT

"Movie", m.title, COUNT(1) AS num\_reprod

FROM

movies AS m

JOIN

history AS h ON m.movie\_id = h.movie\_id

JOIN

users AS u ON h.user\_id = u.user\_id

WHERE

u.kids\_mode = 'Y'

GROUP BY

m.title

ORDER BY

COUNT(1) DESC)

UNION ALL

(SELECT

"Series", s.title, COUNT(1) AS num\_reprod

FROM

series AS s

JOIN

history AS h ON s.series\_id = h.series\_id

JOIN

users AS u ON h.user\_id = u.user\_id

WHERE

u.kids\_mode = 'Y'

GROUP BY

s.title

ORDER BY

COUNT(1) DESC)) AS top\_stream

ORDER BY

top\_stream.num\_reprod DESC

LIMIT 1;

**Answer:**



## 5. What is the playtime by producer, by movie and series?

SELECT

pt.\*

FROM

(SELECT

m.content\_type, m.distributer, SUM(m.duration) AS total\_available\_playtime

FROM

movies AS m

GROUP BY

m.content\_type, m.distributer

UNION ALL

SELECT

s.content\_type, s.distributer, SUM(s.duration) AS total\_available\_playtime

FROM

series AS s

GROUP BY

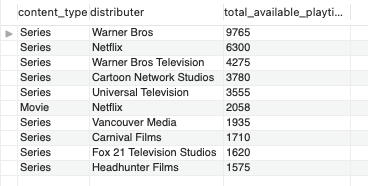
s.content\_type, s.distributer) AS pt

ORDER BY

total\_available\_playtime DESC

LIMIT 10;

**Answer:**



## 6. What country are the majority of the ACTIVE users from?

SELECT

cl.Country, COUNT(1) AS num\_customers

FROM

Customer\_LogIn AS cl

JOIN

customer\_status AS cs ON cl.User\_ID = cs.user\_id

WHERE

cs.status = 'active'

AND cs.to\_date > NOW()

GROUP BY

cl.Country

ORDER BY

num\_customers DESC

LIMIT 1;

**Answer:**



## 7. Which payment plan generates the most revenue?

SELECT

p.paymant\_plan, SUM(p.amount) AS revenue

FROM

payment AS p

JOIN

customer\_status AS cs ON p.user\_id = cs.user\_id

GROUP BY

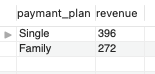
p.paymant\_plan

ORDER BY

revenue DESC

LIMIT 1;

**Answer:**

****

## 8. Identify the least popular movies and series in the last year?

SELECT

low\_stream.\*

FROM

((SELECT

"Movie", m.title, COUNT(1) AS num\_reprod

FROM

movies AS m

JOIN

history AS h ON m.movie\_id = h.movie\_id

WHERE

h.date\_steamed >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY

m.title

ORDER BY

COUNT(1) ASC)

UNION ALL

(SELECT

"Series", s.title, COUNT(1) AS num\_reprod

FROM

series AS s

JOIN

history AS h ON s.series\_id = h.series\_id

WHERE

h.date\_steamed >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR)

GROUP BY

s.title

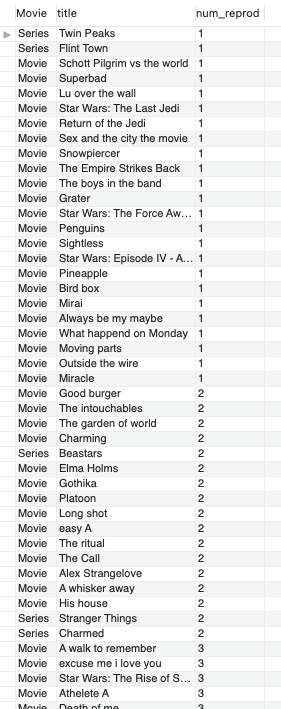
ORDER BY

COUNT(1) ASC)) AS low\_stream

ORDER BY

low\_stream.num\_reprod ASC;

**Answer:**



## 9. Find the top 3 genres viewed in kids’ mode who are still active users?

SELECT

top\_genre.genre, COUNT(1) AS num\_reprod

FROM

((SELECT

m.genre AS genre

FROM

movies AS m

JOIN

history AS h ON m.movie\_id = h.movie\_id

JOIN

users AS u ON h.user\_id = u.user\_id

JOIN

customer\_status AS cs ON u.user\_id = cs.user\_id

WHERE

cs.status = 'active'

AND cs.to\_date > NOW()

AND u.kids\_mode = 'Y')

UNION ALL

(SELECT

s.genre AS genre

FROM

series AS s

JOIN

history AS h ON s.series\_id = h.series\_id

JOIN

users AS u ON h.user\_id = u.user\_id

JOIN

customer\_status AS cs ON u.user\_id = cs.user\_id

WHERE

cs.status = 'active'

AND cs.to\_date > NOW()

AND u.kids\_mode = 'Y')) AS top\_genre

GROUP BY

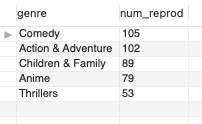
top\_genre.genre

ORDER BY

num\_reprod DESC

LIMIT 5;

**Answer:**



## 10. Find the most popular genres and average playtime for churning customers to identify potential reasons of leaving and potential reactions to win back the customers.

SELECT

top\_genre.genre, COUNT(1) AS num\_reprod, ROUND(AVG(top\_genre.playtime),2) AS avg\_playtime\_min

FROM

(SELECT

m.genre AS genre, m.duration AS playtime

FROM

movies AS m

JOIN

history AS h ON m.movie\_id = h.movie\_id

JOIN

users AS u ON h.user\_id = u.user\_id

JOIN

customer\_status AS cs ON u.user\_id = cs.user\_id

WHERE

cs.status = 'inactive'

UNION ALL

SELECT

s.genre AS genre, s.duration AS playtime

FROM

series AS s

JOIN

history AS h ON s.series\_id = h.series\_id

JOIN

users AS u ON h.user\_id = u.user\_id

JOIN

customer\_status AS cs ON u.user\_id = cs.user\_id

WHERE

cs.status = 'inactive') AS top\_genre

GROUP BY

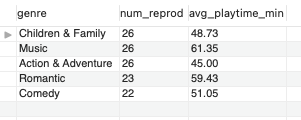
top\_genre.genre

ORDER BY

num\_reprod DESC

LIMIT 5;

**Answer:**



# 2 detailed SQL procedures

## First procedure

**Outline:** Update 25% discount for users who have been active over the past 3 years on their payment plan as a reward of their loyalty to the company.

**Input:** No input parameter.

**Output:** No output parameter.

**Functionality:**

1. Identifying users which have been active without any breaks over the past 3 years
2. For these users, update the payment by applying the 25% discount and show in the bonus column the applied discount value.

**Code of procedure:**

USE `streaming\_database`;

DROP procedure IF EXISTS `p\_discount`;

DELIMITER $$

USE `streaming\_database`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `p\_discount`()

BEGIN

DECLARE cur\_user\_id INT DEFAULT 0;

DECLARE dscnt CURSOR FOR

SELECT

cs.user\_id

FROM

customer\_status AS cs

JOIN

payment AS p ON cs.user\_id = p.user\_id

WHERE

cs.status = 'active'

AND p.discount IS NULL

GROUP BY

cs.user\_id

HAVING

SUM(datediff(cs.to\_date,cs.from\_date)) >= 1096;

OPEN dscnt;

loop\_trigger:LOOP

FETCH dscnt INTO cur\_user\_id;

UPDATE payment

SET discount = 25,

amount = amount \* ((100-discount)/100)

WHERE user\_id = cur\_user\_id;

END LOOP loop\_trigger;

END$$

DELIMITER ;

## Second procedure

**Outline:** The platform welcomes each user with a “Welcome Back” statement. On the user’s birthday change the message to “HAPPY BIRTHDAY” instead.

**Input:** IN in\_User\_ID INT

*Example for the message “Welcome Back”:*

Call b\_day(7, @out\_message);

*Example for the message “HAPPY BIRTHDAY”:*

*Call b\_day(9,@out\_message);*

**Output:** OUT out\_message VARCHAR(255)

*Example for the message “Welcome Back”:*

select @out\_message;



*Example for the message “HAPPY BIRTHDAY”:*

*select @out\_message;*

**

**Functionality:**

1. Identify if it is the user’s birthday
2. If yes, set the welcome message to “HAPPY BIRTHDAY”
3. Else, the welcome message remains “Welcome Back”

**Code of procedure:**

DROP PROCEDURE IF EXISTS b\_day;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `b\_day`(IN in\_User\_ID INT, OUT out\_message VARCHAR(255))

BEGIN

DECLARE bd\_check VARCHAR(5);

SELECT

CONCAT\_WS('-',MONTH(Birth\_date), DAY(Birth\_date)) INTO bd\_check

FROM

Customer\_LogIn

WHERE

User\_ID = in\_User\_ID;

IF bd\_check = CONCAT\_WS('-',MONTH(NOW()), DAY(NOW()))

THEN SET out\_message = "HAPPY BIRTHDAY";

ELSE SET out\_message = "Welcome Back";

END IF;

END $$

DELIMITER ;

# 

# Appendix:

## Create Table scripts

-- MySQL Workbench Forward Engineering

**SET** @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

**SET** @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

**SET** @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema streaming\_database

-- -----------------------------------------------------

**CREATE** **SCHEMA** **IF** **NOT** **EXISTS** `streaming\_database` **DEFAULT** CHARACTER **SET** utf8mb4 **COLLATE** utf8mb4\_0900\_ai\_ci ;

**USE** `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`Customer\_LogIn`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `mydb`.`Customer\_LogIn` (

`User\_ID` INT **NOT** NULL,

`Birth\_date` DATE NULL,

`E\_Mail` VARCHAR(45) NULL,

`Password` VARCHAR(45) NULL,

`Country` VARCHAR(45) NULL,

PRIMARY **KEY** (`User\_ID`))

**ENGINE** = **InnoDB**;

-- -----------------------------------------------------

-- Table `mydb`.`User`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `mydb`.`User` (

`User\_Name` VARCHAR(45) NULL,

`Kids\_Mode` VARCHAR(45) NULL,

`User\_ID` INT **NOT** NULL,

**INDEX** `fk\_User\_Customer\_LogIn1\_idx` (`User\_ID` **ASC**) **VISIBLE**,

PRIMARY **KEY** (`User\_ID`),

**CONSTRAINT** `fk\_User\_Customer\_LogIn1`

**FOREIGN** **KEY** (`User\_ID`)

**REFERENCES** `mydb`.`Customer\_LogIn` (`User\_ID`)

**ON** **DELETE** **NO** **ACTION**

**ON** **UPDATE** **NO** **ACTION**)

**ENGINE** = **InnoDB**;

-- -----------------------------------------------------

-- Table `mydb`.`Content`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `mydb`.`Content` (

`Releasedate` DATE NULL,

`Title` VARCHAR(45) NULL,

`Episode` VARCHAR(45) NULL,

`Genre` ENUM('Action & Adventure', 'Anime', "Children & Family", "Classic", "Comedies", "Documentaries", "Dramas", "Horror", "Music", "Romantic", "Sci-fi & Fantasy", "Sport", "Thrillers", "TV Shows") NULL,

`Content\_ID` VARCHAR(45) **NOT** NULL,

`Distributer` VARCHAR(45) NULL,

`Duration` INT NULL,

`Rating` ENUM("TV-MA", "TV-14", "TV-Y7", "TV-PG", "R", "PG") NULL,

`Season` VARCHAR(45) NULL,

PRIMARY **KEY** (`Content\_ID`))

**ENGINE** = **InnoDB**;

**USE** `streaming\_database` ;

-- -----------------------------------------------------

-- Table `streaming\_database`.`movies`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `streaming\_database`.`movies` (

`movie\_id` INT **NOT** NULL,

`title` VARCHAR(45) **NOT** NULL,

`release\_date` **YEAR**(4) **NOT** NULL,

`content\_type` ENUM('Movie', 'Series') **NOT** NULL,

`genre` ENUM('Action & Adventure', 'Anime', 'Children & Family', 'Classic', 'Comedy', 'Documentary', 'Drama', 'Horror', 'Music', 'Romantic', 'Sci-fi & Fantasy', 'Sport', 'Thrillers', 'TV Shows') **NOT** NULL,

`distributer` VARCHAR(45) **NOT** NULL,

`duration` INT **NOT** NULL,

`classification` ENUM('PG', 'PG-13', 'PG-14', 'R', 'TV-14', 'TV-MA', 'TV-PG', 'TV-Y7') **NOT** NULL,

PRIMARY **KEY** (`movie\_id`))

**ENGINE** = **InnoDB**

**DEFAULT** CHARACTER **SET** = utf8mb4

**COLLATE** = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `streaming\_database`.`series`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `streaming\_database`.`series` (

`series\_id` INT **NOT** NULL AUTO\_INCREMENT,

`title` VARCHAR(45) **NOT** NULL,

`season\_count` INT **NOT** NULL,

`season` INT **NOT** NULL,

`episode` INT **NOT** NULL,

`release\_date` **YEAR**(4) **NOT** NULL,

`genre` ENUM('Action & Adventure', 'Anime', 'Children & Family', 'Classic', 'Comedy', 'Documentary', 'Drama', 'Horror', 'Music', 'Romantic', 'Sci-fi & Fantasy', 'Sport', 'Thrillers', 'TV Shows') **NOT** NULL,

`distributer` VARCHAR(45) **NOT** NULL,

`duration` INT **NOT** NULL,

`classification` ENUM('TV-14', 'TV-MA', 'TV-PG', 'TV-15', 'TV-16', 'TV-17', 'TV-18', 'TV-19', 'TV-20', 'TV-21') **NOT** NULL,

`content\_type` ENUM('Movie', 'Series') **NOT** NULL,

PRIMARY **KEY** (`series\_id`),

**UNIQUE** **INDEX** `series\_id\_UNIQUE` (`series\_id` **ASC**) **VISIBLE**)

**ENGINE** = **InnoDB**

**DEFAULT** CHARACTER **SET** = utf8mb4

**COLLATE** = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `streaming\_database`.`customer\_login`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `streaming\_database`.`customer\_login` (

`user\_id` INT **NOT** NULL AUTO\_INCREMENT,

`birth\_date` DATE **NOT** NULL,

`email` VARCHAR(45) **NOT** NULL,

`password\_user` VARCHAR(45) **NOT** NULL,

`country` ENUM('USA', 'Afghanistan', 'Albania ', 'Andorra ', 'Angola ', 'Antigua ', 'Argentina ', 'Armenia ', 'Australia', 'Brazil', 'Chile', 'China', 'Denmark', 'Georgia', 'Germany', 'Italy', 'Jamaica', 'Norway') **NOT** NULL,

PRIMARY **KEY** (`user\_id`),

**UNIQUE** **INDEX** `email` (`email` **ASC**) **VISIBLE**,

**UNIQUE** **INDEX** `user\_id\_UNIQUE` (`user\_id` **ASC**) **VISIBLE**)

**ENGINE** = **InnoDB**

**DEFAULT** CHARACTER **SET** = utf8mb4

**COLLATE** = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `streaming\_database`.`history`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `streaming\_database`.`history` (

`History\_id` INT **NOT** NULL AUTO\_INCREMENT,

`date\_steamed` DATETIME **NOT** NULL **DEFAULT** **CURRENT\_TIMESTAMP**,

`user\_id` INT **NOT** NULL,

`movie\_id` INT NULL **DEFAULT** NULL,

`series\_id` INT NULL **DEFAULT** NULL,

PRIMARY **KEY** (`History\_id`),

**INDEX** `playtime\_to\_user\_id` (`user\_id` **ASC**) **VISIBLE**,

**INDEX** `playtime\_to\_series\_id` (`series\_id` **ASC**) **VISIBLE**,

**UNIQUE** **INDEX** `playtime\_id\_UNIQUE` (`History\_id` **ASC**) **VISIBLE**,

**INDEX** `playtime\_to\_movie\_id` (`movie\_id` **ASC**) **VISIBLE**,

**CONSTRAINT** `playtime\_to\_movie\_id`

**FOREIGN** **KEY** (`movie\_id`)

**REFERENCES** `streaming\_database`.`movies` (`movie\_id`),

**CONSTRAINT** `playtime\_to\_series\_id`

**FOREIGN** **KEY** (`series\_id`)

**REFERENCES** `streaming\_database`.`series` (`series\_id`),

**CONSTRAINT** `playtime\_to\_user\_id`

**FOREIGN** **KEY** (`user\_id`)

**REFERENCES** `streaming\_database`.`customer\_login` (`user\_id`))

**ENGINE** = **InnoDB**

**DEFAULT** CHARACTER **SET** = utf8mb4

**COLLATE** = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `streaming\_database`.`users`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `streaming\_database`.`users` (

`user\_name` VARCHAR(45) **NOT** NULL,

`user\_id` INT **NOT** NULL,

`kids\_mode` ENUM('Y', 'N') **NOT** NULL,

PRIMARY **KEY** (`user\_name`),

**INDEX** `user\_id\_to\_name` (`user\_id` **ASC**) **VISIBLE**,

**CONSTRAINT** `user\_id\_to\_name`

**FOREIGN** **KEY** (`user\_id`)

**REFERENCES** `streaming\_database`.`customer\_login` (`user\_id`))

**ENGINE** = **InnoDB**

**DEFAULT** CHARACTER **SET** = utf8mb4

**COLLATE** = utf8mb4\_0900\_ai\_ci;

-- -----------------------------------------------------

-- Table `streaming\_database`.`payment`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `streaming\_database`.`payment` (

`paymant\_plan` ENUM("Single", "Family") **NOT** NULL,

`amount` DECIMAL(2) **NOT** NULL,

`card\_nr` **DOUBLE** **NOT** NULL,

`card\_name` VARCHAR(45) **NOT** NULL,

`cvv` INT(3) **NOT** NULL,

`user\_id` INT **NOT** NULL,

`active` ENUM("Active", "Inactive") **NOT** NULL,

`discount` INT NULL,

**INDEX** `fk\_payment\_customer\_login1\_idx` (`user\_id` **ASC**) **VISIBLE**,

**CONSTRAINT** `fk\_payment\_customer\_login1`

**FOREIGN** **KEY** (`user\_id`)

**REFERENCES** `streaming\_database`.`customer\_login` (`user\_id`)

**ON** **DELETE** **NO** **ACTION**

**ON** **UPDATE** **NO** **ACTION**)

**ENGINE** = **InnoDB**;

-- -----------------------------------------------------

-- Table `streaming\_database`.`customer\_status`

-- -----------------------------------------------------

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `streaming\_database`.`customer\_status` (

`from\_date` DATE **NOT** NULL,

`to\_date` DATE NULL,

`status` ENUM("active", "inactive") **NOT** NULL,

`user\_id` INT **NOT** NULL,

PRIMARY **KEY** (`from\_date`, `user\_id`, `status`),

**INDEX** `fk\_customer\_status\_customer\_login1\_idx` (`user\_id` **ASC**) **VISIBLE**,

**CONSTRAINT** `fk\_customer\_status\_customer\_login1`

**FOREIGN** **KEY** (`user\_id`)

**REFERENCES** `streaming\_database`.`customer\_login` (`user\_id`)

**ON** **DELETE** **NO** **ACTION**

**ON** **UPDATE** **NO** **ACTION**)

**ENGINE** = **InnoDB**;

**SET** SQL\_MODE=@OLD\_SQL\_MODE;

**SET** FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

**SET** UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

## Insert scripts:

-- -----------------------------------------------------

-- INSERT SCRIPT 15 out of 50: `customer\_login`

-- -----------------------------------------------------

**INSERT** INTO `customer\_login`

**VALUES**

(1,'1997-09-17','meinkej@icloud.com','nMCAXR6d','USA'),

(2,'1999-09-07','alastair@att.net','7hjnppkA','USA'),

(3,'2001-08-22','rhavyn@verizon.net','3m4MGxhb','Brazil'),

(4,'2001-11-13','martyloo@att.net','DwMtrz8k','China'),

(5,'2002-05-19','iamcal@optonline.net','WTb4ueuu','USA'),

(6,'2003-08-22','arnold@outlook.com','CwLRMD8d','USA'),(7,'2006-08-02','benanov@me.com','XAzfP7KA','Afghanistan'),(8,'2007-02-09','pgottsch@verizon.net','7E4d9RcL','Afghanistan'),(9,'2007-05-13','tjensen@mac.com','S5SsmVkV','Afghanistan'),(10,'2007-08-14','gator@comcast.net','WrUcUrgd','Afghanistan'),(11,'2002-06-06','chrishen@yahoo.com','ETzYrk7b9','Chile'),(12,'1992-09-03','vhbim@gmail.con','2RetP6uJp','Albania'),(13,'1987-06-09','derhi@att.net','u3srCGys7','Albania'),(14,'1989-06-09','furry@yahoo.com','6fWshBRta','Albania'),(15,'1999-12-01','sydy@gmail.com','ETzYrk7b9','Albania')

-- -----------------------------------------------------

-- INSERT SCRIPT 15 out of 53: `customer\_status `

-- -----------------------------------------------------

**INSERT INTO** `customer\_status`

**VALUES**

('2013-06-13','2018-12-04','inactive',49),

('2013-07-12','2021-11-03','active',50),

('2017-09-13','2021-10-12','active',11),

('2017-11-29','2021-11-21','active',8),

('2018-04-12','2019-07-16','inactive',43),

('2018-06-27','2021-08-13','active',7),

('2018-08-21','2022-01-01','active',17),

('2018-08-21','2021-01-01','inactive',29),

('2018-08-22','2021-06-06','active',36),

('2018-08-25','2020-05-17','active',34),

('2018-08-25','2021-05-01','active',44),

('2018-08-26','2021-12-26','active',16),

('2018-08-29','2021-05-08','active',45),

('2018-09-02','2021-11-26','active',14),

('2018-09-02','2021-12-23','active',40),

('2018-09-06','2021-05-22','active',47)

-- -----------------------------------------------------

-- INSERT SCRIPT 15 out of 50: `payment`

-- -----------------------------------------------------

**INSERT** **INTO** `payment`

**VALUES** ('Single',11,1.70712349603675e15,'Jimmy Watson',123,1,'Active',NULL),('Single',11,2.31796106004897e15,'Ryan Lopez',234,2,'Active',NULL),

('Single',11,4.10305515025349e15,'Kathleen Hall',532,3,'Active',NULL),

('Single',11,6.01598554346584e15,'Katherine Morris',532,4,'Active',NULL),

('Single',11,5.13641982865748e15,'Christopher Powell',124,5,'Active',NULL),

('Single',11,2.30436594357103e15,'Terry Evans',399,6,'Active',NULL),

('Single',11,1.91610317248571e15,'Randy Foster',429,7,'Active',NULL),

('Single',11,9.86359610038693e15,'Dennis Jackson',459,8,'Active',NULL),

('Single',11,3.32408368926236e15,'Jacqueline Barnes',489,9,'Active',NULL),

('Single',11,9.63712331743929e15,'Annie Collins',519,10,'Active',NULL),

('Single',11,2.40925122543832e15,'Larry Clark',549,11,'Active',NULL),

('Single',11,8.4850023843731e15,'Thomas King',579,12,'Active',NULL),

('Single',11,9.74436671027786e15,'Peter Hughes',609,13,'Active',NULL),

('Single',11,6.77620711667444e15,'Angela Carter',639,14,'Active',NULL),

('Single',11,9.4323793853526e15,'Carlos Price',669,15,'Active',NULL)

-- -----------------------------------------------------

-- INSERT SCRIPT15 out of 50: `users`

-- -----------------------------------------------------

**INSERT** **INTO** `users`

**VALUES** ('Andre Norris',15,'N'),('ARMY',13,'N'),('Baby shark',7,'Y'),('Becki Hutcherson',26,'Y'),('Black pink',14,'N'),('Bobble',3,'N'),('Bragdol',45,'N'),('Brain',9,'N'),('Brian',1,'Y'),('BTS',11,'N'),('Drubrarg',44,'Y'),('Elizabeth Bell',31,'Y'),('Fun',8,'Y');

-- -----------------------------------------------------

-- INSERT SCRIPT 15 out of 66: `movies`

-- -----------------------------------------------------

**INSERT** **INTO** **`movies` VALUES** (1,'Outside the wire',2021,'Movie','Action & Adventure','Netflix',115,'R'),(2,'What happend on Monday',2017,'Movie','Action & Adventure','Netflix',123,'TV-MA'),(3,'Homefront',2013,'Movie','Action & Adventure','Open Road Films',100,'R'),(4,'Snowpiercer',2013,'Movie','Action & Adventure','CJ Entertainment',126,'R'),(5,'Elma Holms',2020,'Movie','Action & Adventure','Netflix',123,'PG-13'),(6,'Schott Pilgrim vs the world',2010,'Movie','Action & Adventure','Universal Pictures',112,'PG-13'),(7,'A silent voice ',2016,'Movie','Anime','Shochiku',129,'TV-14'),(8,'Mirai',2018,'Movie','Anime','Toho Co Ltd',98,'PG'),(9,'A whisker away',2020,'Movie','Anime','Netflix',105,'TV-PG'),(10,'The garden of world',2013,'Movie','Anime','Toho Co Ltd',45,'TV-14'),(11,'Lu over the wall',2017,'Movie','Anime','Toho Co Ltd',112,'PG'),(12,'Penguins',2014,'Movie','Children & Family','DreamWorks',92,'PG'),(13,'Croods',2013,'Movie','Children & Family','DreamWorks',98,'PG'),(14,'Charming',2021,'Movie','Children & Family','Netflix',85,'TV-Y7'),(15,'Good burger',1997,'Movie','Children & Family','Paramount Pictures',95,'PG')

-- -----------------------------------------------------

-- INSERT SCRIPT 15 out of 1216: `series`

-- -----------------------------------------------------

**INSERT INTO** `series`

**VALUES** (1,'Lost **in** Space',2,2,12,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(2,'Lost **in** Space',2,2,11,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(3,'Lost **in** Space',2,2,10,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(4,'Lost **in** Space',2,2,9,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(5,'Lost **in** Space',2,2,8,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(6,'Lost **in** Space',2,2,7,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(7,'Lost **in** Space',2,2,6,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(8,'Lost **in** Space',2,2,5,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(9,'Lost **in** Space',2,2,4,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(10,'Lost **in** Space',2,2,3,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(11,'Lost **in** Space',2,2,2,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(12,'Lost **in** Space',2,2,1,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(13,'Lost **in** Space',2,1,13,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(14,'Lost **in** Space',2,1,12,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'),(15,'Lost **in** Space',2,1,11,2018,'Sci-fi & Fantasy','Legendary Television',45,'TV-PG','Series'2018,'Classic','Spelling Television',45,'TV-14','Series');

-- -----------------------------------------------------

-- INSERT SCRIPT 15 out of 1384: `history`

-- -----------------------------------------------------

**INSERT INTO** `history` **VALUES** (1,'2013-07-12 00:00:00',49,NULL,732),(2,'2013-07-28 00:00:00',49,43,NULL),(3,'2014-04-01 00:00:00',49,NULL,815),(4,'2014-04-29 00:00:00',49,54,NULL),(5,'2014-08-05 00:00:00',49,NULL,86),(6,'2015-02-05 00:00:00',49,NULL,617),(7,'2015-05-22 00:00:00',49,NULL,817),(8,'2015-09-16 00:00:00',49,NULL,1202),(9,'2015-12-17 00:00:00',49,NULL,840),(10,'2016-03-12 00:00:00',49,NULL,497),(11,'2016-04-19 00:00:00',49,44,NULL),(12,'2016-07-21 00:00:00',49,NULL,666),(13,'2016-09-10 00:00:00',49,NULL,1137),(14,'2016-11-09 00:00:00',49,NULL,39),(15,'2016-12-09 00:00:00',49,NULL,1174)