Database Management and Applications

HULU Media Application

Project Report Submitted by:

Group 5:

Damoder Reddy Pannala (lz7285)

Nishtha Hiteshkumar Ranpara (gg9226)

Sri Samhitha Pallelamudi (ti6266)

Prathamesh Rajiv Bhople (yv5392)

CONTENTS

I. SUMMARY

HULU is an American Streaming Platform. It offers a library of Films and Television Series as well as HULU original content. Hulu is one of th e leading premium streaming services offering live and on-demand TV and movies. This report provides a brief overview of a business model and the organization's activities from a business standpoint.

II. INTRODUCTION

HULU is an American Streaming Platform. It offers a library of Films and Television Series as well as HULU original content. The application was launched on October 29th, 2007. It is currently available in United States only. In 2010, HULU became the first streaming service to add “Plus” to its name when it launched a subscription service. In 2017, the company launched HULU with Live TV – an over-the-top IPTV service featuring linear television service. In the fourth quarter of 2021, HULU has 43.8 million subscribers.

III. BUSINESS REQUIREMENTS

- HULU application classifies the media type into TV shows and Movies, along with it also streams premiums and originals.
- Every media type has a unique MediaTypeID identifier which will uniquely identify the media as tv show, movies, premiums, or originals.
- Also, we can store the information like genre, country, Censor rating etc. for every media type.
- Each media type can be either a movie or tv show but cannot be both at a same time.
- Information like TV show number of seasons or length of a movie is stored.
- Originals can have free as well as paid media types. However, premiums will have only paid media types.
- Each User have unique login id as UserID along with login password.
- Every user can either be a subscriber or a free user. Subscriber can access paid as well as free media, but free user cannot access paid media.
- Subscription plan details for each user can be recorded such as plan type, plan price, video quality subscribed.
- Every subscription payment details are recorded where unique TransactionID will identify every payment of plan made by each user.

IV. CARDINALITY CONSTRAINTS

Following are the cardinality constraints for all the entities

- The HULU application can have one or more of their originals but, every originals is exclusive to HULU application.
- The HULU application can have one or more of their premiums contents but, every premium content is streamed on only HULU application.
- Every user must have one login to the application, but the application can have zero or multiple users logged in.
- Every user must have a subscription plan. Also, every subscription plan can be subscribed by one or more users.
- User can have multiple transactions of payment for same plan if user decides to upgrade the same. But each transaction should be performed by one user login.

V. ENTITY RELATIONSHIP (ER) DIAGRAM

Below is the ER Diagram for HULU Media Application,

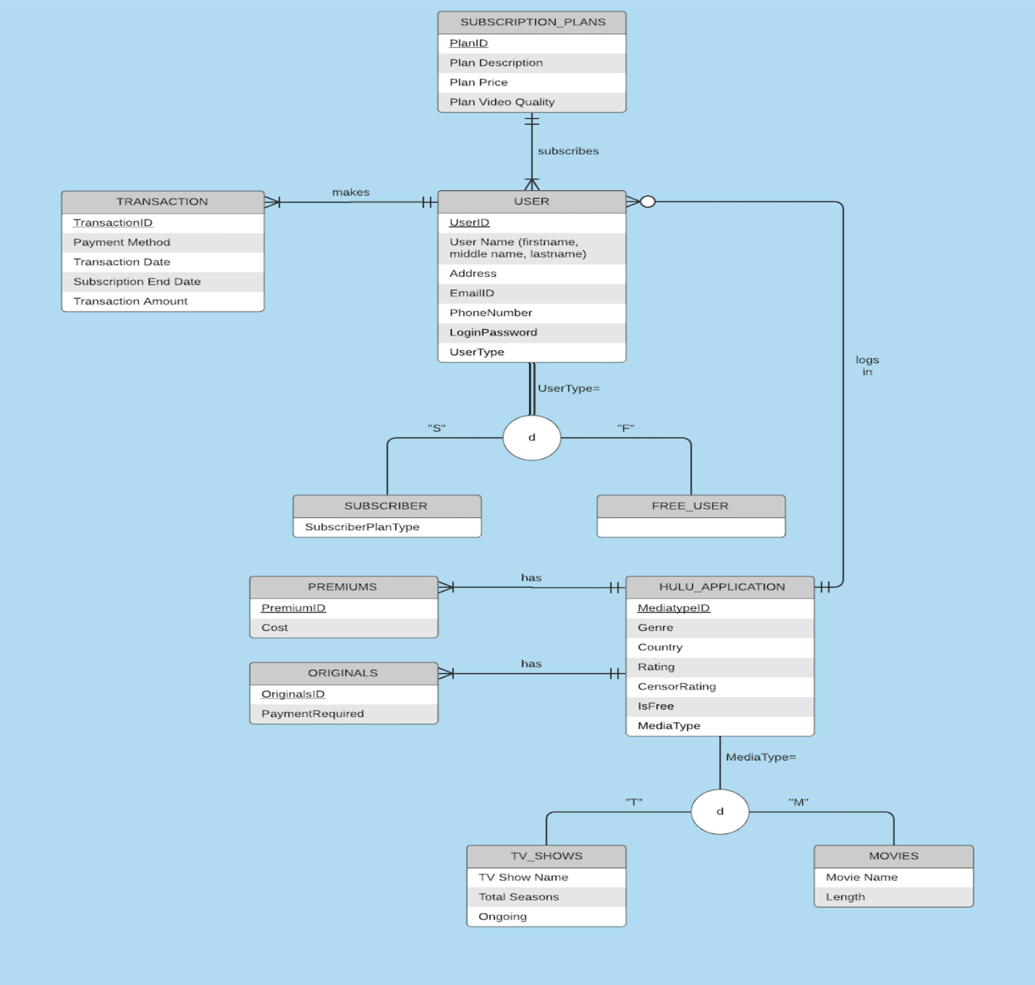


Figure 1. ER Diagram for HULU Media Application

VI. LOGICAL RELATIONSHIP MODEL

Below is the Logical Relationship model developed using ER Diagram in Figure 1,

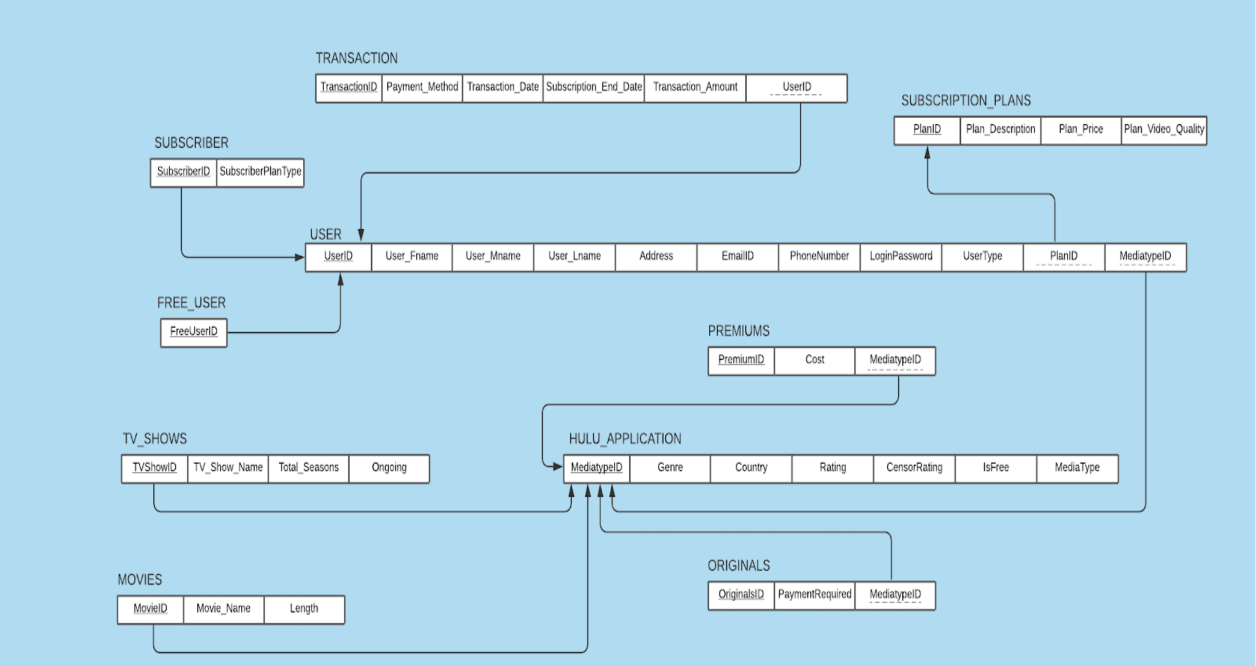


Figure 2. Logical Relationship Model for HULU Media Application

VII. MODEL DESCRIPTION

Now that the ER Diagram and Logical Representation model is ready, we create the metadata for the tables mentioned in the logical representation model. We first create the table that does not have foreign key relationships. Following are the metadata tables describing the attribute information,

TV_Shows_T table

Attributes	Data Type	Size	Auto	Null	Constraint
tvshowid	Int	10		no	primary
tv_show_name	varchar	100			
total_seasons	Int	10			
ongoing	varchar	100			

Movies_T table

Attributes	Data Type	Size	Auto	Null	Constraint
movieid	Int	10		no	primary
movie_name	varchar	100			
movie_length	Int	10			

Subscription_Plans_T table

Attributes	Data Type	Size	Auto	Null	Constraint
planid	int	10		no	primary
plan_description	varchar	100			
plan_price	varchar	100			
plan_video_quality	varchar	100			

HULU_Application_T table

Attributes	Data Type	Size	Auto	Null	Constraint
mediatypeid	Int	10		no	primary
genre	varchar	100			
country	varchar	100			
rating	varchar	100			
censorrating	varchar	100			
isfree	Char	15			
mediatype	varchar	100			

User_T table

Attributes	Data Type	Size	Auto	Null	Constraint
userid	int	10		no	primary
user_fname	varchar	100			
user_mname	varchar	100			
user_lname	varchar	100			
address	varchar	100			
emailed	varchar	100			
phonenumber	varchar	100			
loginpassword	varchar	100			
usertype	varchar	100			
planid	int	10			foreign
mediatypeid	int	10			foreign

Free_User_T table

Attributes	Data Type	Size	Auto	Null	Constraint
freeuserid	int	10		no	Part of primary, foreign

Subscriber_T table

Attributes	Data Type	Size	Auto	Null	Constraint
subscriberid	int	10		no	Part of primary, foreign
subscriberplantype	varchar	20		no	

Transaction_T table

Attributes	Data Type	Size	Auto	Null	Constraint
transactionid	int	10		no	primary
payment_method	varchar	100			
transaction_date	date				
subscription_end_date	date				
transaction_amount	double				
userid	int	10			foreign

Premiums_T table

Attributes	Data Type	Size	Auto	Null	Constraint
premiumid	int	10		no	primary
cost	varchar	100			
mediatypeid	int	10			foreign

Originals_T table

Attributes	Data Type	Size	Auto	Null	Constraint
originals_id	int	10		no	primary
paymentrequired	char	100			
mediatypeid	int	10			foreign

VIII. MYSQL IMPLEMENTATION

Once the metadata for all the 10 tables is created, we can go ahead with SQL queries. We will start with creating the database and tables in the same.

I). Data Definition Language (DDL) queries

Creating the database tables using DDL queries in MYSQL,

```
/* Creating the Database */
CREATE DATABASE HULU;

/* Using the created HULU Database */
USE HULU;

/* DDL Statements */
/* Table TV_Shows_T */
CREATE TABLE TV_Shows_T (
TVShowID int(10) NOT NULL,
TV_Show_Name varchar(100),
Total_Seasons int(10),
Ongoing varchar(100),
CONSTRAINT TV_Shows_pk PRIMARY KEY (TVShowID))
ENGINE = INNODB;

/* Table Movies_T */
CREATE TABLE Movies_T  (
MovieID int(10) NOT NULL,
Movie_Name varchar(100),
Movie_Length int(10),
CONSTRAINT Movies_pk PRIMARY KEY (MovieID))
ENGINE = INNODB;
```

```
/* Table Subscription_Plans_T */
CREATE TABLE Subscription_Plans_T  (
PlanID int(10) NOT NULL,
Plan_Description varchar(100),
Plan_Price varchar(100),
Plan_Video_Quality varchar(100),
CONSTRAINT Subscription_Plans_pk PRIMARY KEY (PlanID))
ENGINE = INNODB;

/* Table HULU_Application_T */
CREATE TABLE HULU_Application_T  (
MediaTypeID int(10) NOT NULL,
Genre varchar(100),
Country varchar(100),
Rating varchar(100),
CensorRating varchar(100),
IsFree char(15),
MediaType varchar(100),
CONSTRAINT HULU_Application_pk PRIMARY KEY (MediaTypeID))
ENGINE = INNODB;

/* Table User_T */
CREATE TABLE User_T (
UserId int(10) NOT NULL,
User_Fname varchar(100),
User_Mname varchar(100),
User_Lname varchar(100),
Address varchar(100),
EmailID varchar(100),
PhoneNumber varchar(100),
LoginPassword varchar(100),
UserType varchar(100),
PlanId int(10),
MediaTypeID int(10),
CONSTRAINT User_pk PRIMARY KEY (UserId),
CONSTRAINT User_fk FOREIGN KEY (PlanId) REFERENCES Subscription_Plans_T(PlanId) ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT User_fk1 FOREIGN KEY (MediaTypeID) REFERENCES HULU_Application_T(MediaTypeID) ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;


/* Table FreeUser_T */
CREATE TABLE FreeUser_T  (
FreeUserID int(10) NOT NULL,
CONSTRAINT FreeUser_pk PRIMARY KEY (FreeUserID),
CONSTRAINT FreeUser_fk FOREIGN KEY (FreeUserID) REFERENCES User_T (UserID) ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;


/* Table Subscriber_T */
CREATE TABLE Subscriber_T (
Subscriber_ID int(10) NOT NULL,
SubscriberPlanType varchar(20) NOT NULL,
CONSTRAINT Subscriber_pk PRIMARY KEY (Subscriber_ID),
CONSTRAINT Subscriber_fk FOREIGN KEY (Subscriber_ID) REFERENCES User_T (UserID) ON UPDATE CASCADE ON DELETE CASCADE)
ENGINE = INNODB;


/* Table Transaction_T */
CREATE TABLE Transaction_T (
TransactionID int(10) NOT NULL,
Payment_Method varchar(100),

Transaction_Date varchar(100),

Subscription_End_Date varchar(100),

Transaction_Amount double,

UserID int(10),

CONSTRAINT Transaction_pk PRIMARY KEY (TransactionID),

CONSTRAINT Transaction_fk FOREIGN KEY (UserID) REFERENCES User_T(UserID) ON
UPDATE CASCADE ON DELETE CASCADE)

ENGINE = INNODB;


/* Table Premiums_T */

CREATE TABLE Premiums_T (

Premium_ID int(10) NOT NULL,

Cost varchar(100),

MediaTypeID int(10),

CONSTRAINT Premiums_pk PRIMARY KEY (Premium_ID),

CONSTRAINT Premiums_fk FOREIGN KEY (MediaTypeID) REFERENCES
HULU_Application_T(MediaTypeID) ON UPDATE CASCADE ON DELETE CASCADE)

ENGINE = INNODB;


/* Table Originals_T */

CREATE TABLE Originals_T (

Originals_ID int(10) NOT NULL,

PaymentRequired char(100),

MediaTypeID int(10),

CONSTRAINT Originals_pk PRIMARY KEY (Originals_ID),

CONSTRAINT Originals_fk FOREIGN KEY (MediaTypeID) REFERENCES
HULU_Application_T(MediaTypeID) ON UPDATE CASCADE ON DELETE CASCADE)

ENGINE = INNODB;
```

II). Data Manipulation Language (DML) queries

After executing DDL statements and database tables are created, we will insert the data into each tables. Following are the INSERT Statements for the same,

```
INSERT INTO TV_Shows_T (TVShowID, Tv_Show_Name, Total_Seasons, Ongoing)

VALUES

(101, 'Bakers Dozen', '1','Yes'),

(102, 'Insecure','5','No'),

(103, 'Dexter', '8', 'No'),

(104, 'Yellowjackets','1','Yes'),

(105,'Teen Titans Go','6','No'),

(106,'The Great','2','Yes'),

(107,'Family Guy','20','No'),

(108,'Ancient Aliens','14','No'),

(109,'Succession','3','No'),

(110,'P-Valley','1','No'),

(111,'Marvels Hit Monkey','1','Yes'),

(112,'Godfather of Harlem','2','Yes'),

(113,'Powerbook II','2','No'),

(114,'Wrong Man','2','No'),

(115,'Pen-15','2','Yes'),

(116,'Samurai Champloo', '1','Yes'),

(117,'Staged','2','Yes'),

(118,'Power','6','No'),

(119,'Black-ish','7','No'),

(120,'The Hardy Boys','1','Yes');
```

```
INSERT INTO Movies_T (MovieID, Movie_Name, Movie_Length)

VALUES

(201,'DeadPool 2','2h'),

(202, 'elf','1h 37m'),
(203,'PIG','1h 32m'),
(204,'The Humans','1h 48m'),
(205,'Norbit','1h 43m'),
(206,'Caged','1h 21m'),
(207,'Last Holiday','1h 52m'),
(208,'Bad Boys for life','2h 4m'),
(209,'Jumanji: The next level','2h 3m'),
(210,'Grown Ups 2','1h 41m'),
(211,'Queen Bees','1h 40m'),
(212,'Jingle All The Way','1h 20m'),
(213,'Iron Man 3','2h 10m'),
(214,'Boss Baby','1h 37m'),
(215,'The Holiday','2h 18m'),
(216,'The Marksman','1h 48m'),
(217,'Shrek 2','1h 33m'),
(218,'Rush Hour','1h 38m'),
(219,'The Misfits','1h 34m'),
(220,'Christmas Cupid','1h 25m');

INSERT INTO subscription_plans_T (planid, plan_description, plan_price, plan_video_quality)
VALUES
(1001, 'Monthly Hulu 30 day free trial', 6.99, 'standard'),
(1002, 'Monthly Hulu (no ads) 30 day free trial', 12.99, 'HD'),
(1003, 'Monthly Hulu + Live TV 7 day free trial', 64.99, '4K'),
(1004, 'Monthly Hulu, Disney+ and ESPN+ Bundle', 13.99, 'standard'),
(1005, 'Monthly Hulu, Disney+ and ESPN+ Bundle (no ads)', 19.99, 'HD'),
(1006, 'Monthly Hulu + Live TV, Disney+ and ESPN+ Bundle', 72.99, '4K');

INSERT INTO hulu_application_T (mediatypeid, genre, country, rating, censorrating, isfree,
mediatype) VALUES
```

```
(101,'Sitcom','USA','6.8','G','N','T'),
(102,'Drama','USA','7.9','PG-13','N','T'),
(103,'Mystery','USA','8.6','R','N','T'),
(104,'Drama','Canada','7.9','R','N','T'),
(105,'Action','USA','5.3','G','N','T'),
(106,'Historical','USA','8.1','PG','Y','T'),
(107,'Comedy','USA','8.1','G','Y','T'),
(108,'Documentary','UK','7.2','G','N','T'),
(109,'Comedy','USA','8.7','R','Y','T'),
(110,'Drama','USA','7.1','PG-13','N','T'),
(111,'Anime','USA','8.0','G','Y','T'),
(112,'Drama','USA','8.0','R','Y','T'),
(113,'Crime','UK','6.9','R','N','T'),
(114,'Crime','UK','7.1','R','Y','T'),
(115,'Comedy','Canada','8.0','PG','Y','T'),
(116,'Anime','Japan','8.5','PG-13','N','T'),
(117,'Action','USA','8.6','R','N','T'),
(118,'Crime','USA','8.1','R','N','T'),
(119,'Sitcom','USA','7.1','PG-13','N','T'),
(120,'Drama','UK','7.1','R','Y','T'),
(121,'Action','USA','7.7','R','Y','M'),
(122,'Comedy','USA','7.0','G','N','M'),
(123,'Drama','USA','6.9','R','N','M'),
(124,'Drama','USA','6.1','PG-13','Y','M'),
(125,'Comedy','USA','4.1','R','N','M'),
(126,'Crime','USA','3.6','PG-13','Y','M'),
(127,'Comedy','UK','6.5','G','N','M'),
(128,'Action','USA','6.6','PG-13','Y','M'),
(129,'Action','USA','6.7','G','N','M'),
(130,'Comedy','USA','5.4','PG','N','M'),
(131,'Comedy','UK','6.1','G','Y','M'),

(132,'Comedy','USA','5.7','G','N','M'),

(133,'Action','USA','7.1','G','Y','M'),

(134,'Comedy','USA','6.3','G','Y','M'),

(135,'Comedy','USA','6.9','PG','N','M'),

(136,'Action','USA','5.6','PG-13','N','M'),

(137,'Anime','USA','7.2','G','Y','M'),

(138,'Action','USA','6.5','PG-13','Y','M'),

(139,'Action','UK','4.2','R','N','M'),

(140,'Comedy','USA','5.6','G','Y','M');


INSERT INTO User_T (UserID, User_Fname, User_Mname, User_Lname, Address, EmailID,
PhoneNumber, LoginPassword, UserType, planid, mediatypeid) VALUES

(1, 'Jessica', 'Eva', 'Parker', '912 BStreet California', 'jevaparker@yahoo.com', 3413255430,
'Jessp', 'FreeUser', 1002, 101),

(2, 'Laurie', 'H', 'Paul', '27eagenalley Texas', 'lpaul@gmail.com', 6413255770, 'Laupau',
'FreeUser', 1003, 102),

(3, 'Faye', 'Evans', 'DSouza', '8 Janacourt Idaho', 'evansfaye@yahoo.com', 5413255230,  'Fayep',
'FreeUser', 1002, 103),

(4, 'Kirsti', 'N', 'Gwillym', '8 OxfordCrossing UK', 'KirstiG@outlook.com', 5013259930, 'gkirsti',
'Subscriber', 1004, 108),

(5, 'Terrence', 'Roger', 'Lewis', '5 DwereyPark Arizona', 'Terryrl@yahoo.com', 9112345876 ,
'Terrman', 'FreeUser', 1003, 104),

(6, 'Lazaro', 'W', 'Croster', '94542 Carlos Bee Blvd California', 'Lazacross@yahoo.com',
3413259930, 'Lazac', 'FreeUser', 1002, 105),

(7, 'Hari', 'P', 'Rao', '333 ManhattanSt New York', 'hariprasad@yahoo.com', 2313259930, 'Harir',
'FreeUser', 1003, 106),

(8, 'Naina', 'Ram', 'Reddy', '881 Sunnyvale California', 'nainareddy@gmail.com', 5123252113,
'Reddyn', 'Subscriber', 1004, 107),

(9, 'Samhitha', 'Chakri', 'Pallelamudi', '511 Newtown Florida', 'samhithap@yahoo.com',
2453259975, 'Sammy', 'FreeUser', 1001, 109),

(10, 'Anne', 'Holmes', 'Watson', '12wellingStreet Misisippi', 'Annehw@yahoo.com', 8913259233,
'Holmes', 'FreeUser', 1001, 110),
(11, 'Iwette', 'B', 'David', '30 Autumnpark Washington', 'Iwetteb@yahoo.com', 7613253450,
'David', 'Subscriber', 1005, 111),

(12, 'Paul', 'Steph', 'Hawkins', 'Melrosejunction Canada', 'Stephenhawk@yahoo.com',
5012259321, 'Hawkin', 'Subscriber', 1006, 112),

(13, 'Olympie', 'H', 'Golborn', '22 Jasperst Alabama', 'Olympus@gmail.com', 6752593442,
'OlympGlb', 'Subscriber', 1006, 113),

(14, 'Sam', 'Aaron', 'Manekshaw', '912 BStreet California', 'Manesam@yahoo.com', 5589399391,
'Samm', 'Subscriber', 1004, 114),

(15, 'Danna', 'Roy', 'Chatterjee', '65 Larrypoint Alaska', 'chatterjee@yahoo.com', 3213257938,
'Chatterg', 'Subscriber', 1006, 115),

(16, 'Devon', 'Heerick', 'Bollis', 'IndianSt Texas', 'dheerick@yahoo.com', 5338259956, 'Devonh',
'Subscriber', 1005, 116),

(17, 'Val', 'K' ,'Champken', '33 FlorenceStreet UK', 'champkenval@yahoo.com', 4453251976,
'valc','Subscriber', 1006, 117),

(18, 'Melanie', 'Marie', 'Clinton', 'Nottinghill New York', 'melaniec@yahoo.com', 5313256630,
'Clinton', 'Subscriber', 1006, 118),

(19, 'Rowela', 'Emma', 'Johnson', 'Elak lane Ohio', 'johnsonowela@yahoo.com', 5119859965,
'Johrow', 'Subscriber', 1004, 119),

(20, 'Pooja', 'Pankaj', 'Agarwal', '221 Freemont California', 'Poojaagarwal@yahoo.com',
4323257632, 'Poojaa', 'FreeUser', 1001, 120);


INSERT INTO Subscriber_T (Subscriber_ID, SubscriberPlanType) VALUES

(4, '$13.99 Plan'),

(8, '$13.99 Plan'),

(11, '$19.99 Plan'),

(12, '$72.99 Plan'),

(13, '$72.99 Plan'),

(14, '$13.99 Plan'),

(15, '$72.99 Plan'),

(16, '$19.99 Plan'),

(17, '$72.99 Plan'),

(18, '$72.99 Plan'),

(19, '$13.99 Plan');
```

```
INSERT INTO FreeUser_T(FreeuserID) VALUES (1),(2),(3),(5),(6),(7),(9),(10),(20);

INSERT INTO transaction_t
(transactionID,payment_method,transaction_date,subscription_End_Date,transaction_amount,userid) VALUES
(101,'Credit','2021-11-01','2021-11-31','6.99',1),
(102,'Debit','2021-11-01','2022-11-30','69.99',2),
(103,'Credit','2021-11-05','2022-12-04','12.99',3),
(104,'Credit','2021-11-05','2022-11-04','69.99',4),
(105,'Debit','2021-11-01','2021-11-31','12.99',5),
(106,'ACH','2021-11-03','2021-12-02','6.99',6),
(107,'ACH','2021-11-04','2021-12-03','12.99',7),
(108,'Credit','2021-11-06','2021-12-05','6.99',8),
(109,'ACH','2021-12-06','2022-01-05','6.99',9),
(110,'Debit','2021-12-06','2022-01-05','6.99',10),
(111,'Credit','2021-12-06','2022-01-05','64.99',11),
(112,'Debit','2021-12-06','2022-01-05','64.99',12),
(113,'Credit','2021-12-06','2022-01-05','64.99',13),
(114,'ACH','2021-12-07','2022-01-06','70.99',14),
(115,'ACH','2021-12-08','2022-01-07','6.99',15),
(116,'ACH','2021-12-06','2022-01-05','12.99',16),
(117,'Credit','2021-12-09','2022-01-08','6.99',17),
(118,'Credit','2021-12-06','2022-01-05','70.99',18),
(119,'Credit','2021-12-06','2022-01-05','12.99',19),
(120,'Debit','2021-12-08','2022-01-07','64.99',20);

INSERT INTO originals_t (Originals_ID, PaymentRequired, MediaTypeID) VALUES
(900, "Y",101),
(901, "Y",102),
(902, "N",103),
(903, "N",104),
(904, "Y",105),
(905, "Y",106),
(906, "N",107),
(907, "Y",108),
(908, "N",109),
(910, "N",110),
(911, "Y",111),
(912, "N",112),
(913, "Y",113),
(914, "N",114),
(915, "Y",115),
(916, "Y",116),
(917, "N",117),
(918, "Y",118),
(919, "N",119),
(920, "Y",120);

INSERT INTO premiums_T (Premium_id, Cost, MediaTypeId) VALUES
(1,4.99,136),
(2,5.99,122),
(3,9.99,116),
(4,2.99,117),
(5,4.99,123),
(6,3.99,118),
(7,9.99,132),
(8,5.99,104),
(9,8.99,129),
```

(10,7.99,102),

(11,8.99,127),

(12,3.99,139),

(13,9.99,101),

(14,6.99,135),

(15,2.99,103),

(16,8.99,119),

(17,5.99,125),

(18,8.99,113),

(19,3.99,108),

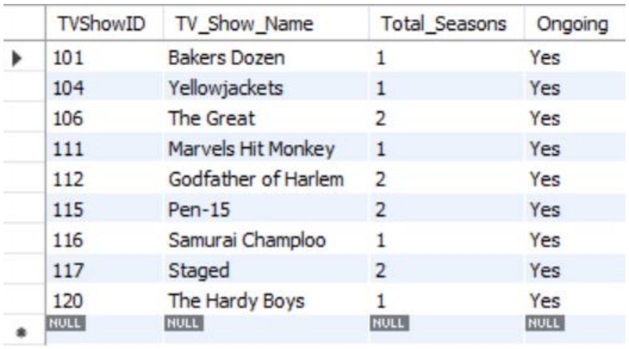
(20,7.99,130);

IX. RESEARCH QUESTIONS

After inserting the data in all the tables, we are going to answer some research questions using SQL queries in mysql.

Query 1 - Displaying Tv shows information (Tv show id, tv show name, and total seasons) for the TV shows which are ongoing.

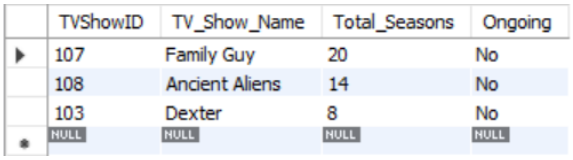
```
SELECT * FROM TV_Shows_T WHERE Ongoing = 'Yes';
```



	TVShowID	TV_Show_Name	Total_Seasons	Ongoing
▶	101	Bakers Dozen	1	Yes
	104	Yellowjackets	1	Yes
	106	The Great	2	Yes
	111	Marvels Hit Monkey	1	Yes
	112	Godfather of Harlem	2	Yes
	115	Pen-15	2	Yes
	116	Samurai Champloo	1	Yes
	117	Staged	2	Yes
	120	The Hardy Boys	1	Yes
*	NULL	NULL	NULL	NULL

Query 2 - Displaying top 3 Tv shows information (Tv show id, tv show name, and total seasons) for the TV shows which have highest number of seasons in total.

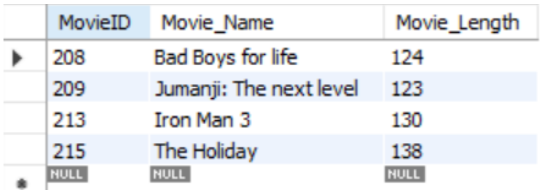
```
SELECT * FROM TV_Shows_T ORDER BY Total_Seasons DESC LIMIT 3;
```



	TVShowID	TV_Show_Name	Total_Seasons	Ongoing
▶	107	Family Guy	20	No
	108	Ancient Aliens	14	No
	103	Dexter	8	No
*	NULL	NULL	NULL	NULL

Query 3 - Displaying Movies information (movie id, movie name, and movie length) for movies which runs for more than 2 hours (120 minutes).

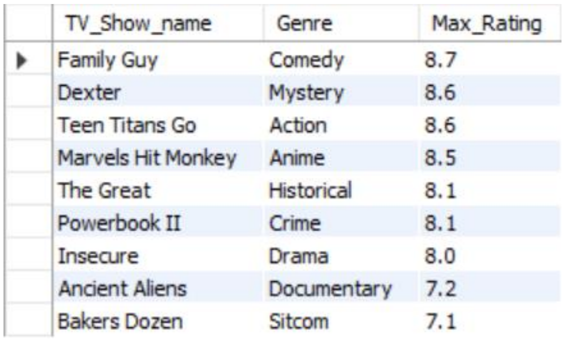
SELECT * FROM Movies_T WHERE movie_length > 120;



	MovieID	Movie_Name	Movie_Length
▶	208	Bad Boys for life	124
	209	Jumanji: The next level	123
	213	Iron Man 3	130
	215	The Holiday	138
*	NULL	NULL	NULL

Query 4 - Displaying TV Show name for each genre with max rating (Using inner join).

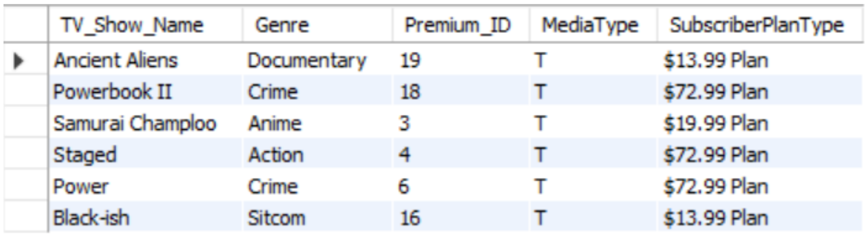
SELECT TS.TV_Show_name, HA.genre, MAX(HA.Rating) AS Max_Rating FROM TV_Shows_T TS INNER JOIN Hulu_Application_T HA ON TS.TVShowID = HA.MediaTypeID GROUP BY HA.Genre ORDER BY MAX(HA.Rating) DESC;



	TV_Show_name	Genre	Max_Rating
▶	Family Guy	Comedy	8.7
	Dexter	Mystery	8.6
	Teen Titans Go	Action	8.6
	Marvels Hit Monkey	Anime	8.5
	The Great	Historical	8.1
	Powerbook II	Crime	8.1
	Insecure	Drama	8.0
	Ancient Aliens	Documentary	7.2
	Bakers Dozen	Sitcom	7.1

Query 5 – Which Premium TV shows and movies belongs to subscription plan of '$72.99 Plan' and '$13.99 Plan'.

SELECT TV_Shows_T.TV_Show_Name AS TV_Show_Name, Hulu_Application_T.Genre, Premiums_T.Premium_ID, Hulu_Application_T.MediaType, Subscriber_T.SubscriberPlanType FROM Subscriber_T INNER JOIN User_T INNER JOIN Hulu_Application_T INNER JOIN Premiums_T INNER JOIN TV_Shows_T ON Subscriber_T.Subscriber_ID = User_T.UserID AND Hulu_Application_T.mediatypeid = User_T.mediatypeid AND Hulu_Application_T.mediatypeid = Premiums_T.MediatypeID AND TV_Shows_T.TVshowID = Hulu_Application_T.mediatypeid;



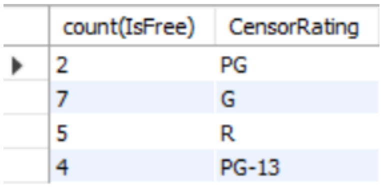
TV_Show_Name	Genre	Premium_ID	MediaType	SubscriberPlanType
Ancient Aliens	Documentary	19	T	$13.99 Plan
Powerbook II	Crime	18	T	$72.99 Plan
Samurai Champloo	Anime	3	T	$19.99 Plan
Staged	Action	4	T	$72.99 Plan
Power	Crime	6	T	$72.99 Plan
Black-ish	Sitcom	16	T	$13.99 Plan

Query 6 - Displaying number of free movies of each censor rating (**using view**).

CREATE VIEW FreeMedia_view AS

SELECT count(IsFree), CensorRating FROM Hulu_Application_T WHERE IsFree='Y' GROUP BY CensorRating;

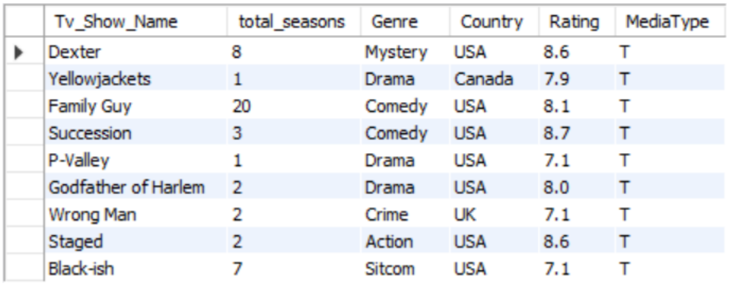
SELECT * FROM FreeMedia_view;



count(IsFree)	CensorRating
2	PG
7	G
5	R
4	PG-13

Query 7 – List all Hulu Originals tv shows and its related data which doesn’t require payment.

SELECT Tv_Show_Name, total_seasons ,Genre, Country, Rating, MediaType FROM Originals_T o INNER JOIN Hulu_Application_T hApp INNER JOIN TV_Shows_T tv ON hApp.mediatypeID = o.MediaTypeID AND tv.TvshowID = hApp.MediaTypeID WHERE PaymentRequired = 'N';



Tv_Show_Name	total_seasons	Genre	Country	Rating	MediaType
Dexter	8	Mystery	USA	8.6	T
Yellowjackets	1	Drama	Canada	7.9	T
Family Guy	20	Comedy	USA	8.1	T
Succession	3	Comedy	USA	8.7	T
P-Valley	1	Drama	USA	7.1	T
Godfather of Harlem	2	Drama	USA	8.0	T
Wrong Man	2	Crime	UK	7.1	T
Staged	2	Action	USA	8.6	T
Black-ish	7	Sitcom	USA	7.1	T

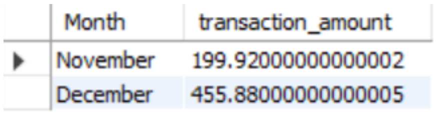
Query 8 – Create a view for the generated revenue with running total

CREATE VIEW totalRevenueByMonths_v AS

SELECT monthname(transaction_date) as Month , sum(transaction_amount) as transaction_amount

FROM transaction_t GROUP BY Month;

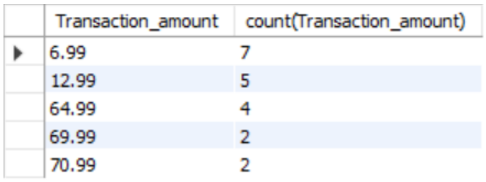
SELECT * FROM totalRevenueByMonths_v;



Month	transaction_amount
November	199.92000000000002
December	455.88000000000005

Query 9 – Count the number of transactions by grouping plans accordingly

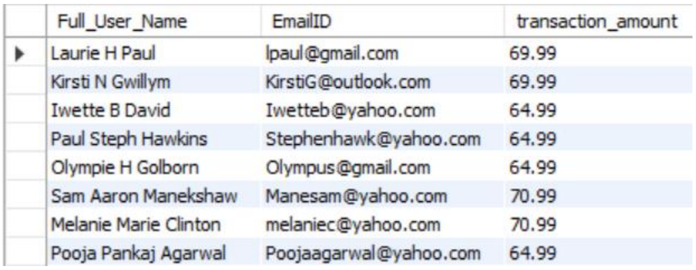
```
SELECT Transaction_amount, count(Transaction_amount) FROM transaction_T GROUP BY
Transaction_amount ORDER BY count(Transaction_amount) DESC;
```



Transaction_amount	count(Transaction_amount)
6.99	7
12.99	5
64.99	4
69.99	2
70.99	2

Query 10 – Displaying customer information who has transaction amount greater than or equal to $64.99

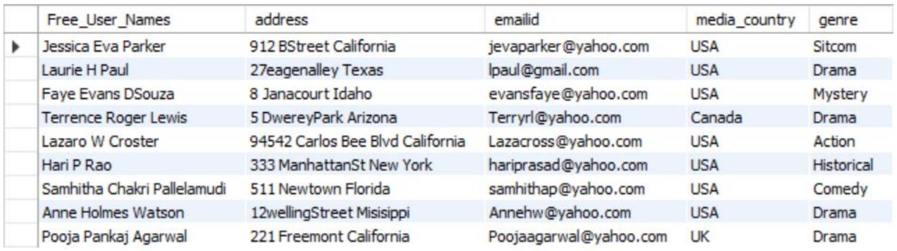
```
SELECT Concat(u.User_Fname,' ',u.User_mname,' ',u.User_lname) as Full_User_Name,
u.EmailID, t.transaction_amount from user_T u INNER JOIN transaction_t t ON t.userid =
u.userid where t.transaction_amount >= 64.99;
```



Full_User_Name	EmailID	transaction_amount
Laurie H Paul	lpaul@gmail.com	69.99
Kirsti N Gwillym	KirstiG@outlook.com	69.99
Iwette B David	Iwetteb@yahoo.com	64.99
Paul Steph Hawkins	Stephenhawk@yahoo.com	64.99
Olympie H Golborn	Olympus@gmail.com	64.99
Sam Aaron Manekshaw	Manesam@yahoo.com	70.99
Melanie Marie Clinton	melaniec@yahoo.com	70.99
Pooja Pankaj Agarwal	Poojaagarwal@yahoo.com	64.99

Query 11 – Displaying Free Users from USA who are watching content which is created in similar or different countries.

```
SELECT Concat(u.User_Fname,' ',u.User_mname,' ',u.User_lname) AS Free_User_Names,
u.address ,ha.country AS media_country, ha.genre FROM FreeUser_T f INNER JOIN User_T u
INNER JOIN Hulu_Application_T ha ON f.FreeuserID = u.UserID AND ha.mediatypeid =
u.mediatypeid;
```



Free_User_Names	address	emailid	media_country	genre
Jessica Eva Parker	912 BStreet California	jevaparker@yahoo.com	USA	Sitcom
Laurie H Paul	27eagenalley Texas	lpaul@gmail.com	USA	Drama
Faye Evans DSouza	8 Janacourt Idaho	evansfaye@yahoo.com	USA	Mystery
Terrence Roger Lewis	5 DwereyPark Arizona	Terryrl@yahoo.com	Canada	Drama
Lazaro W Croster	94542 Carlos Bee Blvd California	Lazacross@yahoo.com	USA	Action
Hari P Rao	333 ManhattanSt New York	hariprasad@yahoo.com	USA	Historical
Samhitha Chakri Pallelamudi	511 Newtown Florida	samhithap@yahoo.com	USA	Comedy
Anne Holmes Watson	12wellingStreet Misisippi	Annehw@yahoo.com	USA	Drama
Pooja Pankaj Agarwal	221 Freemont California	Poojaagarwal@yahoo.com	UK	Drama

IX. CONCLUSION

In conclusion, we have created a database management system that records and tracks user transaction along with subscription plans and contents of the Hulu application. The contents are Movies, Tv Shows, Premiums and Hulu originals from different countries and genres. We analyzed the business requirements, decided on cardinality constraints for each entity based on the same business requirements. Furthermore, we designed an Entity-Relationship diagram along with its logical representation model. We utilized. The database management system we designed is an enhanced ER model with supertype and subtype entities.

Additionally, we implemented the database design with the help of SQL queries like DDL and DML using MYSQL workbench and determined the solutions for certain research questions. While doing this project we also gained deeper understanding on database design and how it can be implemented in real life situations.