

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY



OTT Database Management System

Submitted By:

Aashish Kumar

2021UCS1675

Problem Statement

In Amazon Prime Database, there will be several entities and one of them is for ADMIN only which contains there “user_id” and “password” and has relation with various other entities which allows them to add or edit.

There will be entity ACCOUNTING which contains the data related to transaction made by any user and price of shows or movies which can be purchased by users if they want.

In entity CLIENT, if a new user wanted to sign up they can do it by sharing name, password, address, age, user_name and if they wanted to subscribe they will need to fill PaymentOption, after this they will redirected to search portal where they can search what they wanted to watch like movies and shows.

In both SHOWS and MOVIES there will be title, rating, description, image and other attributes like who was director of the show and about the actor details.

Objectives

To develop a database that:

- Satisfies rule of normalization
- Captures all required information for video streaming platform.

SYSTEM REQUIREMENTS:

H/W Requirement Core:

i5 processor

-> 2GB Ram

-> 20GB of hard disk space in terminal machines

-> 1TB hard disk space in Server Machine

S/W Requirement:

-> Windows 7 or above operating system

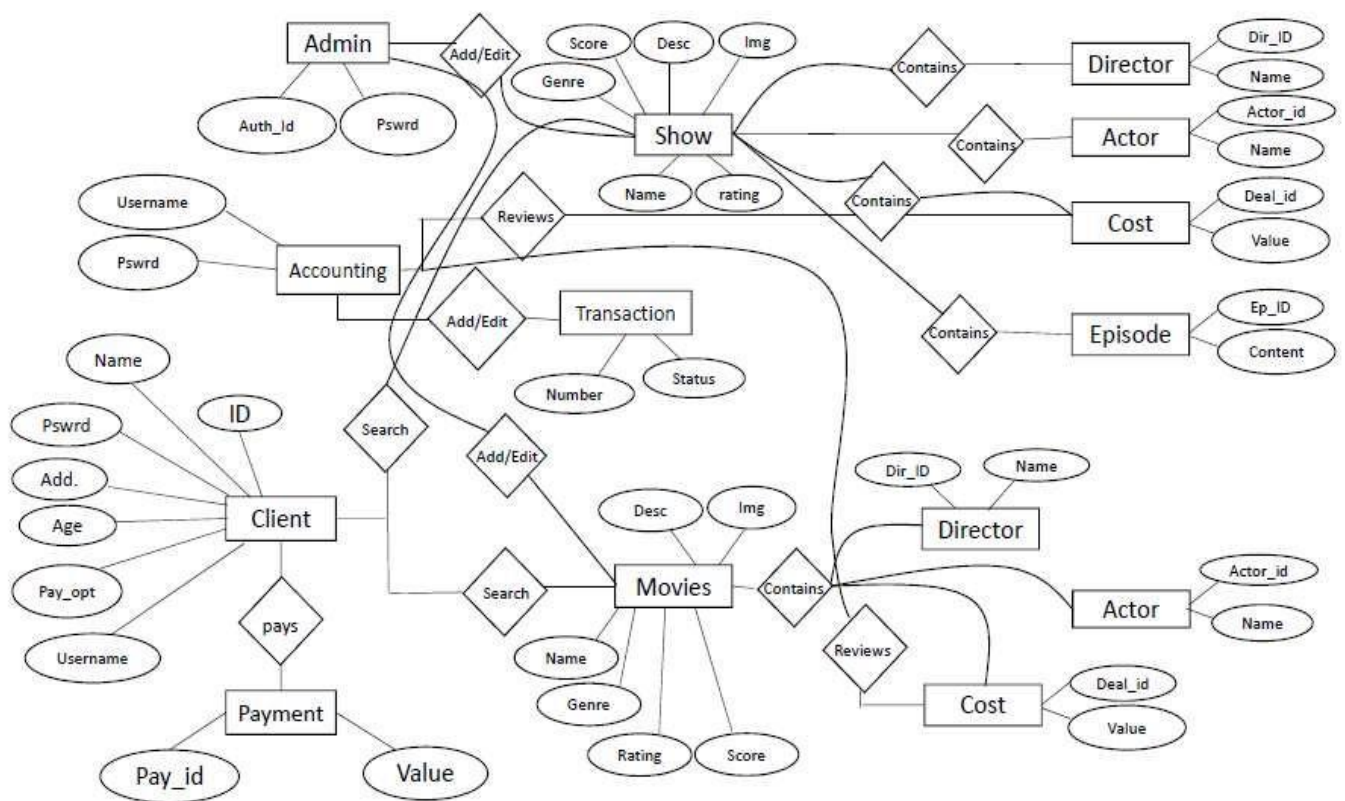
-> VS Code

-> Python 3.11

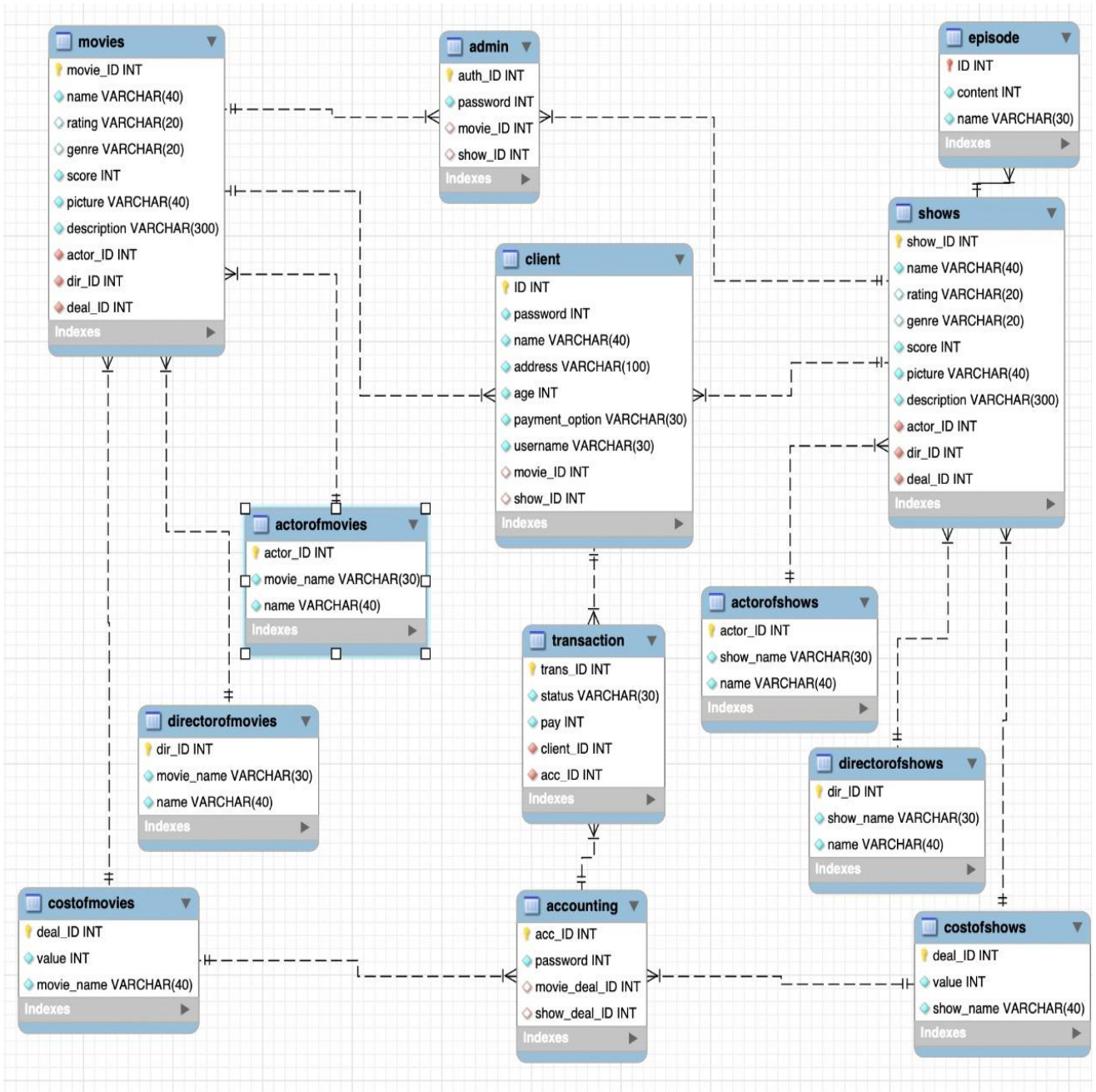
-> Xampp server

-> Command Prompt

ER-Diagram



Relational Schema



Creating Tables

v Table for ‘movies’

```
CREATE TABLE movies (  
  movie_ID int NOT NULL,  
  name varchar(40) NOT NULL,  
  rating varchar(20) DEFAULT NULL,  
  genre varchar(20) DEFAULT NULL,  
  score int NOT NULL,  
  picture varchar(40) NOT NULL,  
  description varchar(300) NOT NULL,  
  actor_ID int NOT NULL,  
  dir_ID int NOT NULL,  
  deal_ID int NOT NULL,  
  PRIMARY KEY (movie_ID),  
  FOREIGN KEY (actor_ID) REFERENCES actorofmovies (actor_ID) ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (dir_ID) REFERENCES directorofmovies (dir_ID) ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (deal_ID) REFERENCES costofmovies (deal_ID) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

v Table for ‘shows’

```
CREATE TABLE shows (  
  show_ID int NOT NULL,  
  name varchar(40) NOT NULL,  
  rating varchar(20) DEFAULT NULL,  
  genre varchar(20) DEFAULT NULL,  
  score int NOT NULL,  
  picture varchar(40) NOT NULL,  
  description varchar(300) NOT NULL,  
  actor_ID int NOT NULL,  
  dir_ID int NOT NULL,  
  deal_ID int NOT NULL,  
  PRIMARY KEY (show_ID),  
  FOREIGN KEY (actor_ID) REFERENCES actorofshows (actor_ID) ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (dir_ID) REFERENCES directorofshows (dir_ID) ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (deal_ID) REFERENCES costofshows (deal_ID) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

#

#Table for ‘director’, ‘actor’, ‘cost’ of movies :

```

CREATE TABLE actorofmovies (
  actor_ID int NOT NULL,
  movie_name varchar(30) NOT NULL,
  name varchar(40) NOT NULL,
  PRIMARY KEY (actor_ID)
);
CREATE TABLE costofmovies (
  deal_ID int NOT NULL,
  value int NOT NULL,
  movie_name varchar(40) NOT NULL,
  PRIMARY KEY (deal_ID)
) ;
CREATE TABLE directorofmovies (
  dir_ID int NOT NULL,
  movie_name varchar(30) NOT NULL,
  name varchar(40) NOT NULL,
  PRIMARY KEY (dir_ID)
) ;

```

Table for 'actor', 'director', 'cost' of shows :


```

CREATE TABLE actorofshows (
    actor_ID int NOT NULL,
    show_name varchar(30) NOT NULL,
    name varchar(40) NOT NULL,
    PRIMARY KEY (actor_ID)
);

CREATE TABLE costofshows (
    deal_ID int NOT NULL,
    value int NOT NULL,
    show_name varchar(40) NOT NULL,
    PRIMARY KEY (deal_ID)
);

CREATE TABLE directorofshows (
    dir_ID int NOT NULL,
    show_name varchar(30) NOT NULL,
    name varchar(40) NOT NULL,
    PRIMARY KEY (dir_ID)
);

```

v Table for 'client'

```

CREATE TABLE client (
    ID int NOT NULL,
    password int NOT NULL,
    name varchar(40) NOT NULL,
    address varchar(100) NOT NULL,
    age int NOT NULL,
    payment_option varchar(30) NOT NULL,
    username varchar(30) NOT NULL,
    movie_ID int,
    show_ID int,
    PRIMARY KEY (ID),
    FOREIGN KEY (movie_ID) REFERENCES movies (movie_ID) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (show_ID) REFERENCES shows (show_ID) ON DELETE CASCADE ON UPDATE CASCADE
);

```

v Table for 'admin' :


```

CREATE TABLE admin(
    auth_ID int NOT NULL,
    password int NOT NULL,
    movie_ID int,
    show_ID int,
    PRIMARY KEY (auth_ID),
    FOREIGN KEY (movie_ID) REFERENCES movies (movie_ID) on update cascade on delete cascade,
    FOREIGN KEY (show_ID) REFERENCES shows (show_ID) on update cascade on delete cascade
);

```

v Table for ‘accounting’ :

```

CREATE TABLE accounting (
    acc_ID int NOT NULL,
    password int NOT NULL,
    movie_deal_ID int,
    show_deal_ID int,
    PRIMARY KEY (acc_ID),
    FOREIGN KEY (movie_deal_ID) REFERENCES costofmovies (deal_ID) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (show_deal_ID) REFERENCES costofshows (deal_ID) ON DELETE CASCADE ON UPDATE CASCADE
);

```

v Table for ‘episode’

```
• ⊖ CREATE TABLE episode (  
    ID int NOT NULL,  
    content int NOT NULL,  
    name varchar(30) NOT NULL,  
    PRIMARY KEY (ID),  
    FOREIGN KEY (ID) REFERENCES shows (show_ID)  
);
```

v Table for 'transaction'

```
CREATE TABLE transaction (  
    trans_ID int NOT NULL,  
    status varchar(30) NOT NULL,  
    pay int not null,  
    client_ID int NOT NULL,  
    acc_ID int NOT NULL,  
    PRIMARY KEY (trans_ID),  
    FOREIGN KEY (client_ID) REFERENCES client(ID) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (acc_ID) REFERENCES accounting (acc_ID) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Tables After Inserting Values

v Table for 'movies' :

movie_ID	name	rating	genre	score	picture	description	actor_ID	dir_ID	deal_ID
1	3 idiots	U	COMEDY/DRAMA	9	3idiots.jpg	In college Farhan	101	101	101
2	KGF	U	drama/action	10	RRR.jpg	Best movie	102	102	102
3	Dear zindagi	U	COMEDY/DRAMA	7	dearzindagi.jpg	The plot centers on a budding cinematography ...	103	103	103
4	Drishyam	A	COMEDY/MYSTERY	10	drishyam.jpg	Desperate measures are taken by a man who tr...	104	104	104
5	PK	U	ROMANCE/DRAMA	8	pk.jpg	PK is a movie about ideas of stranger in the city...	105	105	105
6	URI	A	CRIME/DRAMA	8	uri.jpg	Indian army speical forces execute a covert op...	106	106	106
7	Ra-one	U	DRAMA/ACTION	7	raone.jpg	An immensely powerful virtual character that is ...	107	107	107
8	Dangal	A	DRAMA	9	dangal.jpg	Mahavir singh phogat and his two wrestler daug...	108	108	108
9	Barfi	U	COMEDY	8	barfi.jpg	set in the 1970s in a pretty corner of india,barfi...	109	109	109
10	Oh-My-God	U	DRAMA	8	ohmygod.jpg	Oh My God is a comedy drama.The story of a s...	110	110	110
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

v Table ‘shows’

movie_ID	name	rating	genre	score	picture	description	actor_ID	dir_ID	deal_ID
1	3 idiots	U	COMEDY/DRAMA	9	3idiots.jpg	In college Farhan	101	101	101
2	KGF	U	drama/action	10	RRR.jpg	Best movie	102	102	102
3	Dear zindagi	U	COMEDY/DRAMA	7	dearzindagi.jpg	The plot centers on a budding cinematography ...	103	103	103
4	Drishyam	A	COMEDY/MYSTERY	10	drishyam.jpg	Desperate measures are taken by a man who tr...	104	104	104
5	PK	U	ROMANCE/DRAMA	8	pk.jpg	PK is a movie about ideas of stranger in the city...	105	105	105
6	URI	A	CRIME/DRAMA	8	uri.jpg	Indian army speical forces execute a covert op...	106	106	106
7	Ra-one	U	DRAMA/ACTION	7	raone.jpg	An immensely powerful virtual character that is ...	107	107	107
8	Dangal	A	DRAMA	9	dangal.jpg	Mahavir singh phogat and his two wrestler daug...	108	108	108
9	Barfi	U	COMEDY	8	barfi.jpg	set in the 1970s in a pretty corner of india,barfi...	109	109	109
10	Oh-My-God	U	DRAMA	8	ohmygod.jpg	Oh My God is a comedy drama.The story of a s...	110	110	110
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

v Table ‘director’ of movies

dir_ID	movie_name	name
101	3 idiots	Rajkumar hirani
102	KGF	Prashanth Neel
103	Dear zindagi	Gauri shinde
104	Drishyam	Jeetu Joseph
105	PK	Rajkumar hirani
106	URI	Aditya Dhar
107	Ra-one	Anubhav-Sinha
108	Dangal	Nitest-Tiwari
109	Barfi	Anurag-Basu
110	Oh-My-God	Umesh-Shukla
NULL	NULL	NULL

v Table 'director' of shows

dir_ID	show_name	name
101	1899	Baran Bo Odar
102	Squid Game	Dong-hyuk
103	Stranger things	Duffer-Brothers
104	Dark	Baran Bo Odar
105	Peaky Blinders	Otto-Bathurst
106	Money Heist	Alejandro
107	Manifest	Michael-Smith
108	The Witcher	Tomasz-Baginski
109	Cursed	Sean-Ellis
110	Enola Holmes	Harry-Bradbeer
NULL	NULL	NULL

v Table 'actor' of movies

actor_ID	movie_name	name
101	3 idiots	aamir khan
102	KGF	Naveen kumar
103	Dear zindagi	Alia bhat
104	Drishyam	Ajay devgun
105	PK	Anoushka Sharma
106	URI	Vicky Kaushal
107	Ra-one	Shah-Rukh-Khan
108	Dangal	Aamir-Khan
109	Barfi	Ranbir-Kapoor
110	Oh-My-God	Akshay-Kumar
NULL	NULL	NULL

v Table 'actor' of shows

actor_ID	show_name	name
101	1899	Andreas Pietschmann
102	Squid Game	Lee
103	Stranger things	Milly-Bobby-Brown
104	Dark	Louis Hoffman
105	Peaky Blinders	Cillian Murphy
106	Money Heist	Asa Butterfield
107	Manifest	Melissa-Roxburgh
108	The Witcher	Henry-Cavill
109	Cursed	Katherine-Langford
110	Enola Holmes	Milly-Bobby-Brown
NULL	NULL	NULL

v Table 'client'

ID	password	name	address	age	payment_option	username	movie_ID	show_ID
1	123	ayush	uttam nagar	20	debit	ayush_0	1	NULL
2	123	lalit	dwarka	19	credit	lalit_0	NULL	NULL
3	345	ashmeet	pahar ganj	25	cash	ashmeet_0	2	4
4	678	bhavya	vikar puri	23	debit	bhavya_0	4	1
5	157	abhishek	tilak nagar	27	credit	abhishek_0	5	6
6	826	sam	karol bagh	21	cash	sam_0	3	3
7	157	aman	mumbai	27	credit	sam_0	10	1
8	811	rahul	kolkata	21	cash	rahul_0	1	7
9	787	shiv	noida	23	debit	shiv_0	6	5
10	109	ram	gurugram	27	credit	ram_0	8	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

v Table 'admin'

auth_ID	password	movie_ID	show_ID
1	123	1	NULL
3	345	3	NULL
4	456	4	NULL
5	567	5	NULL
6	678	6	NULL
7	123	NULL	1
8	234	NULL	2
9	345	NULL	3
10	456	NULL	4
11	567	NULL	5
12	678	NULL	6
13	233	7	NULL
14	764	8	NULL
15	147	9	NULL
16	876	10	NULL
17	139	NULL	7
18	498	NULL	8
19	510	NULL	9
20	618	NULL	10
NULL	NULL	NULL	NULL

v Table ‘accounting’

acc_ID	password	movie_deal_ID	show_deal_ID
1	123	101	NULL
2	234	NULL	NULL
3	345	102	104
4	678	104	101
5	157	105	106
6	826	103	103
7	953	110	101
8	824	101	107
9	685	106	105
10	818	108	103
NULL	NULL	NULL	NULL

v Table ‘episode’

ID	content	name
1	9	1899
2	10	Squid Game
3	34	Stranger things
4	24	Dark
5	36	Peaky Blinders
6	40	Money Heist
7	52	Manifest
8	20	The Witcher
9	10	Cursed
10	16	Enola Holmes
NULL	NULL	NULL

v Table ‘transaction’

trans_ID	status	pay	client_ID	acc_ID
1	NO	55	1	1
2	NO	0	2	2
3	YES	133	3	3
4	NO	100	4	4
5	YES	99	5	5
6	NO	232	6	6
7	YES	102	7	7
8	NO	121	8	8
9	YES	91	9	9
10	NO	528	10	10
NULL	NULL	NULL	NULL	NULL

v Table 'costofshows'

deal_ID	value	show_name
101	62	1899
102	122	Squid Game
103	188	Stranger Things
104	89	Dark
105	66	Peaky Blinders
106	32	Money Heist
107	66	Manifest
108	220	The Witcher
109	122	Cursed
110	155	Enola Holmes
NULL	NULL	NULL

v Table 'costofmovies'

deal_ID	value	movie_name
101	55	3 Idiots
102	80	KGF
103	44	Dear zindagi
104	38	Drishyam
105	67	PK
106	25	URI
107	144	Ra-one
108	340	Dangal
109	55	Barfi
110	40	Oh-My-God
NULL	NULL	NULL

Queries

Query 1:

Write a query to show all movies present on OTT.


```
select * from movies where name in(select movie_name from costofmovies where deal_ID=101);
```

Query 5:

No. Of users who's transaction is not completed

```
select count(trans_ID) from transaction where status ='NO';
```

	count(trans_ID)
▶	6

Query 6:

Find amount paid by the client name 'ayush'

	pay
▶	55

```
select pay from transaction where trans_ID in (select ID  
from client where name='ayush');
```

Query 7:

Find the details of show whose value is minimum.

```
select * from shows where name in (select show_name from costofshows where value=(select  
MIN(value) from costofshows));
```

	show_ID	name	rating	genre	score	picture	description	actor_ID	dir_ID	deal_ID	
	6	Money Heist	U/A 15+	CRIME/DRAMA	8	moneyheist.jpg	This crime drama, which interlaces social comm...	106	106	106	
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Query 8:

Find movie directed by Jeetu Joseph.

```
select * from movies where dir_ID in(select dir_ID from directorofmovies where  
name='Jeetu Joseph');
```


Query 13:

Show name ,id and age of person who bought 10th movie and 1st show.

```
select name, ID, age from client where movie_ID=10 and show_ID=1;
```

	name	ID	age
▶	aman	7	27
	NULL	NULL	NULL

Query 14:

Trigger statement to calculate the total amount paid by customers till now

```
create trigger total_amount before insert on transaction for each row set
    @sum=@sum+transaction.pay;
select @sum as total_amount;
```

	total_amou...
▶	3700

Query 15:

Adding client view

```
-- client view
create view client_view as ID,name,address,username from client;
-- altering client's view by adding password,movie_ID
alter view client_view as select ID,address,username payment_option from client;
```