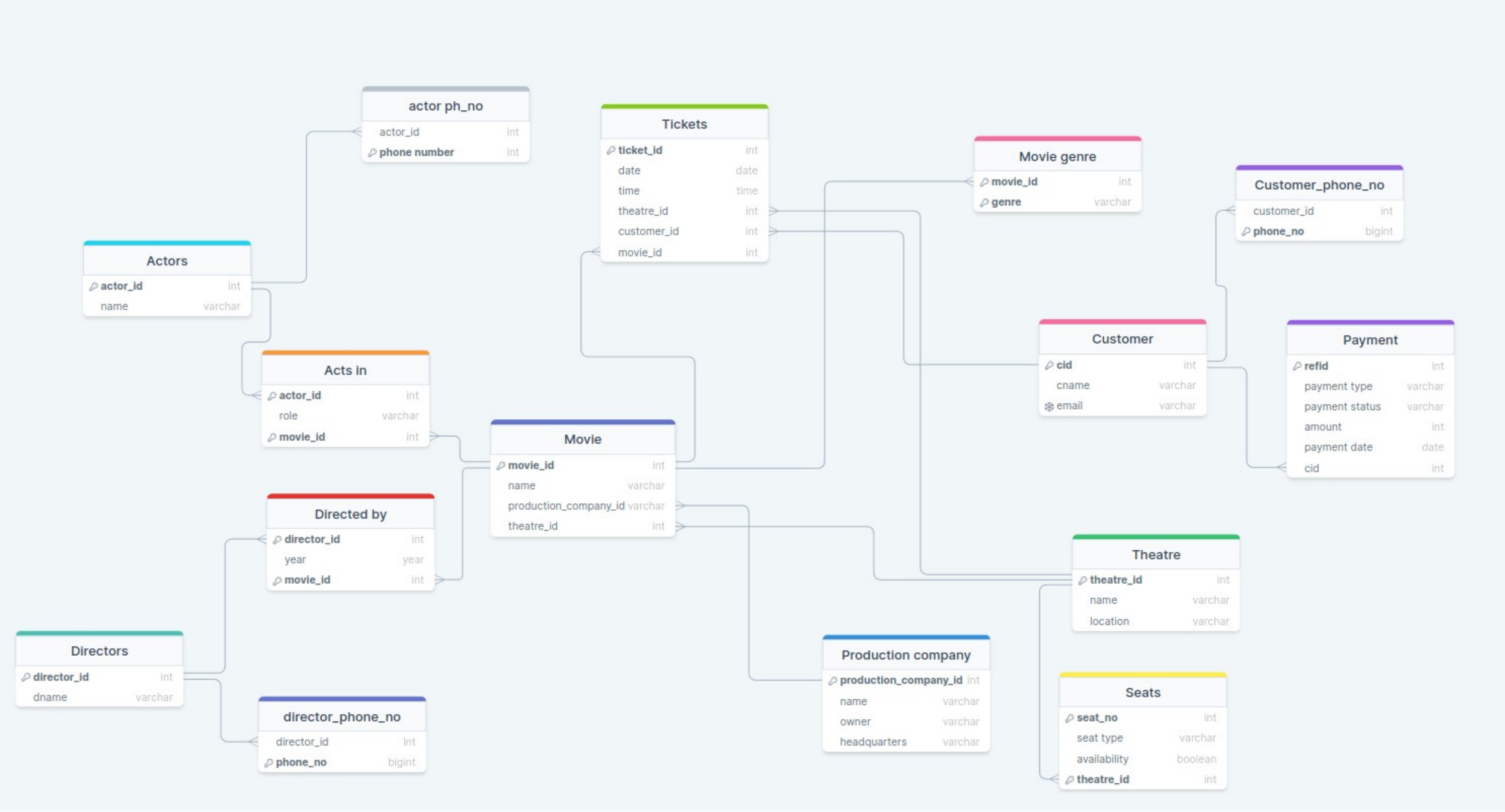
DBMS LAB ASSIGNMENT 3

submitted by
MOHAMED AFTHAB E K
B200719CS

1. Relational diagram



2. Create Tables and Databases

1. Create Database movie_production_company

```
CREATE DATABASE movie_production_company;
```

2. Create table Actors:

```
CREATE TABLE Actors
(
    actor_id INT NOT NULL UNIQUE AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    PRIMARY KEY(actor_id)
);
```

3 .Create table Actor_ph_no

```
CREATE TABLE Actor_phone_no
(
    actor_id INT NOT NULL,
    phone_no BIGINT NOT NULL UNIQUE,
    PRIMARY KEY(phone_no)
);
```

4. Create table Acts_in

```
CREATE TABLE Acts_in
(
    actor_id INT NOT NULL,
    role VARCHAR(20),
    movie_id INT NOT NULL,
    PRIMARY KEY(actor_id,movie_id)
);
```

5.Create table Movies:

```
CREATE TABLE Movies
(
    movie_id INT NOT NULL UNIQUE AUTO_INCREMENT,
    movie_name VARCHAR(50) NOT NULL,
    production_company_id INT NOT NULL,
    theatre_id INT NOT NULL,
    PRIMARY KEY(movie_id)
);
```

6. Create table Directors:

```
CREATE TABLE Directors
(
    director_id INT NOT NULL UNIQUE AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    PRIMARY KEY(director_id)
);
```

7. Create table Director_phone_no:

```
CREATE TABLE Director_phone_no
(
    director_id INT NOT NULL,
    phone_no BIGINT NOT NULL UNIQUE,
    PRIMARY KEY(phone_no)
);
```

8. Create table Directed_by:

```
CREATE TABLE Directed_by
(
    director_id INT NOT NULL,
    release_year YEAR,
    movie_id INT NOT NULL,
    PRIMARY KEY(director_id,movie_id)
);
```

9.Create table Tickets:

```
CREATE TABLE Tickets
(
    ticket_id INT NOT NULL UNIQUE AUTO_INCREMENT,
    ticket_date DATE NOT NULL,
    ticket_time TIME NOT NULL,
    theatre_id INT NOT NULL,
    customer_id INT NOT NULL,
    movie_id INT NOT NULL,
    PRIMARY KEY(ticket_id)
);
```

10. Create table Production_company:

```
CREATE TABLE Production_company
(
    production_company_id INT NOT NULL UNIQUE AUTO_INCREMENT,
    name VARCHAR(25) NOT NULL,
    headquarters VARCHAR(25) NOT NULL,
    owner VARCHAR(25) NOT NULL,
    PRIMARY KEY(production_company_id)
);
```

11. Create table Movie_genre:

```
CREATE TABLE Movie_genre
(
   movie_id INT NOT NULL,
   genre VARCHAR(25) NOT NULL,
   PRIMARY KEY(movie_id,genre)
);
```

12. Create table Customer:

```
CREATE TABLE Customer
(
   customer_id INT NOT NULL UNIQUE AUTO_INCREMENT,
   name VARCHAR(25) NOT NULL,
   email VARCHAR(25) ,
   PRIMARY KEY(customer_id)
);
```

13. Create table Customer_phone_no

```
CREATE TABLE Customer_phone_no
(
   customer_id INT NOT NULL,
   phone_no BIGINT NOT NULL UNIQUE,
   PRIMARY KEY(phone_no)
);
```

14. Create table payment:

```
CREATE TABLE Payment
(
   ref_id INT NOT NULL UNIQUE AUTO_INCREMENT,
   customer_id INT NOT NULL,
   payment_type VARCHAR(10) NOT NULL,
   payment_status VARCHAR(10) NOT NULL,
   payment_date DATE NOT NULL,
   amount INT NOT NULL,
   PRIMARY KEY(ref_id)
);
```

15. Create table theatre

```
CREATE TABLE Theatre
(
    theatre_id INT NOT NULL UNIQUE AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    location VARCHAR(20) NOT NULL,
    PRIMARY KEY(theatre_id)
);
```

16. Create table seats

```
CREATE TABLE Seats
(
    seat_no INT NOT NULL,
    seat_type VARCHAR(20) NOT NULL,
    availability VARCHAR(20) NOT NULL,
    theatre_id INT NOT NULL,
    PRIMARY KEY(seat_no,theatre_id)
);
```

Adding foreign keys by alter table

1. Adding actor_id as foreign key to the following tables

```
ALTER TABLE Actor_phone_no
ADD FOREIGN KEY(actor_id)
REFERENCES Actors(actor_id);

ALTER TABLE Acts_in
ADD FOREIGN KEY(actor_id)
REFERENCES Actors(actor_id);
```

2. Adding movie_id as foreign key to the following tables

```
ALTER TABLE Acts_in
ADD FOREIGN KEY(movie_id)
REFERENCES Movies(movie_id);

ALTER TABLE Directed_by
ADD FOREIGN KEY(movie_id)
REFERENCES Movies(movie_id);
```

```
ALTER TABLE Movie_genre
ADD FOREIGN KEY(movie_id)
REFERENCES Movies(movie_id);

ALTER TABLE Tickets
ADD FOREIGN KEY(movie_id)
REFERENCES Movies(movie_id);
```

3. Adding director_id as foregin key to the following tables

```
ALTER TABLE Director_phone_no
ADD FOREIGN KEY(director_id)
REFERENCES Directors(director_id);

ALTER TABLE Directed_by
ADD FOREIGN KEY(director_id)
REFERENCES Directors(director_id);
```

4. Adding production_company_id as foreign key to the following tables

```
ALTER TABLE Movies
ADD FOREIGN KEY(production_company_id)
REFERENCES Production_company(production_company_id);
```

5. Adding customer_id as foreign key to the following tables

```
ALTER TABLE Tickets
ADD FOREIGN KEY(customer_id)
REFERENCES Customer(customer_id);

ALTER TABLE Customer_phone_no
ADD FOREIGN KEY(customer_id)
REFERENCES Customer(customer_id);

ALTER TABLE Payment
ADD FOREIGN KEY(customer_id)
REFERENCES Customer(customer_id);
```

6. Adding theatre_id as foreign key to the following tables

```
ALTER TABLE Tickets
ADD FOREIGN KEY(theatre_id)
REFERENCES Theatre(theatre_id);

ALTER TABLE Seats
ADD FOREIGN KEY(theatre_id)
REFERENCES Theatre(theatre_id);
```

```
ALTER TABLE Movies
ADD FOREIGN KEY(theatre_id)
REFERENCES Theatre(theatre_id);
```

Inserting values into the database

1. into Actors

```
INSERT INTO `Actors`(`name`)
VALUES
('hritik Roshan'),
('Shahrukh Khan'),
('Vijay Devarakonda'),
('Chiyaan Vikram'),
('Dulquer Salman');
```

2. into Customer

```
INSERT INTO `Customer`(`name`, `email`)
VALUES
('Afthab','afthab@gmail.com'),
('Jithin','jithin@gmail.com'),
('Amal','amal@gmail.com'),
('Milan','milan@gmail.com'),
('Abhay','abhay@gmail.com');
```

3. into Directors

```
INSERT INTO `Directors`(`name`)
VALUES
('Siddharth Anand'),
('Aditya Chopra'),
('Sandeep Reddy Vanga'),
('Lokesh Kanagaraj'),
('Amal Neerad');
```

4. into Theatre

```
INSERT INTO `Theatre`(`name`, `location`)
VALUES
('Kairali','Calicut'),
('Rose','Mukkam'),
('Abhilash','Mukkam'),
('Apsara','Calicut'),
('PVS Film City','Calicut');
```

5. into Production_company

```
INSERT INTO `Production_company`(`name`, `headquarters`,`owner`)
VALUES
('Yash Raj Films','Mumbai','Yash Chopra'),
('Eros International','Mumbai','Kishore Lulla'),
('Bhadrakali Pictures','Vishakpatnam','Pranay Reddy Vanga'),
('Raaj Kamal Films','Calicut','Kamal Hasan'),
('Amal Neerad Productions','Kochi','Amal Neerad');
```

6. into Movies

```
INSERT INTO `Movies`(`movie_name`, `production_company_id`,
`theatre_id`)
VALUES ('WAR','1','1'),
('Dilwale Dulhania Le Jayenge','1','3'),
('Arjun Reddy','3','2'),
('Vikram','4','5'),
('CIA','5','4');
```

7. into Actor_phone_no

```
INSERT INTO `Actor_phone_no`(`actor_id`, `phone_no`)
VALUES
('1','9422334455'),
('2','9422332455'),
('3','8222334458'),
('4','6742334455'),
('1','8322233455'),
('5','9522434455');
```

8. into Acts_in

```
INSERT INTO `Acts_in`(`actor_id`, `role`, `movie_id`)
VALUES
(1,'Male lead','1'),
(2,'Male lead','2'),
(3,'Male lead','3'),
(4,'Major Antagonist','4'),
(5,'Male lead','5');
```

9. into Customer_phone_no

```
INSERT INTO `Customer_phone_no`(`customer_id`, `phone_no`)
VALUES
('1','123346789'),
('2','123414789'),
('3','123432789'),
('4','123456739'),
('5','123456789');
```

10. into Directed_by

```
INSERT INTO `Directed_by`(`director_id`, `release_year`, `movie_id`)
VALUES
('1','2019','1'),
('2','1995','2'),
('3','2017','3'),
('4','2022','4'),
('5','2017','5');
```

11. into Director_phone_no

```
INSERT INTO `Director_phone_no`(`director_id`, `phone_no`)
VALUES
('1','9944334422'),
('2','9942234422'),
('3','9945634422'),
('4','9945434422'),
('5','9944343422');
```

12. into Movie_genre

```
INSERT INTO `Movie_genre`(`movie_id`, `genre`)
VALUES
('1','Action'),
('1','Crime'),
('2','Romance'),
('3','Drama'),
('3','Romance'),
('4','Action'),
('5','Romance');
```

13. into Payment

```
INSERT INTO `Payment`(`customer_id`, `payment_type`,
`payment_status`, `payment_date`, `amount`)
VALUES
('1','UPI','Completed','2019-03-02','220'),
('2','Cash','Completed','1995-03-02','50'),
('3','Debit card','Completed','2017-03-02','110'),
('4','Debit card','Processing','2022-03-02','330'),
('5','UPI','Completed','2017-03-02','110');
```

14. into Seats

```
INSERT INTO `Seats`(`seat_no`, `seat_type`, `availability`, `theatre_id`)
VALUES
('1','Recliner','available','1'),
('5','Regular','booked','2'),
('2','Recliner','available','3'),
('3','Recliner','available','4'),
('2','Recliner','available','5');
```

15. into Tickets

```
INSERT INTO `Tickets`(`ticket_date`, `ticket_time`, `theatre_id`,
`customer_id`, `movie_id`) VALUES
('2017-02-01','12:00:00','1','1','1'),
('1995-02-01','21:00:00','2','2','2'),
('2019-02-01','16:00:00','3','3','3'),
('2022-02-01','14:30:00','4','4','4'),
('2017-08-06','12:00:00','5','5','5');
```

Drop a database

```
CREATE DATABASE test;
DROP DATABASE test;
```

Update an entry

```
-- correcting spelling of Visakapatnam from Vishakapatanam

UPDATE `Production_company` SET `headquarters`='Visakhapatnam'
WHERE production_company_id=3;

--  name capitalization

UPDATE `Actors` SET `name`='Hritik Roshan' WHERE actor_id=1;
```

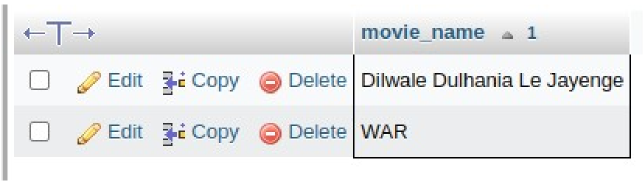
Delete entry

```
-- adding and deleting Santhosh Pandit to Actors

INSERT INTO `Actors`(`name`) VALUES ('Santhosh Pandit');
DELETE FROM `Actors` WHERE name = 'Santhosh Pandit';
```

Exists clause

```
-- checking and listing the movies in the database which were
--  produced by Yash Raj Films


SELECT movie_name
FROM Movies as M
WHERE EXISTS (SELECT movie_name FROM Movies  as M2 WHERE
production_company_id=1 && M.movie_id=M2.movie_id);
```

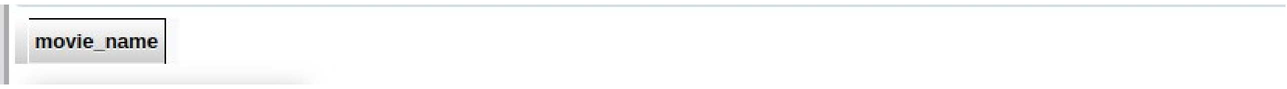
OUTPUT



movie_name ▲ 1
Dilwale Dulhania Le Jayenge
WAR

```
-- checking and listing the movies in the database which were
--  produced by Eros International

SELECT movie_name
FROM Movies as M
WHERE EXISTS (SELECT movie_name FROM Movies  as M2 WHERE
production_company_id=2 && M.movie_id=M2.movie_id);
```

OUTPUT



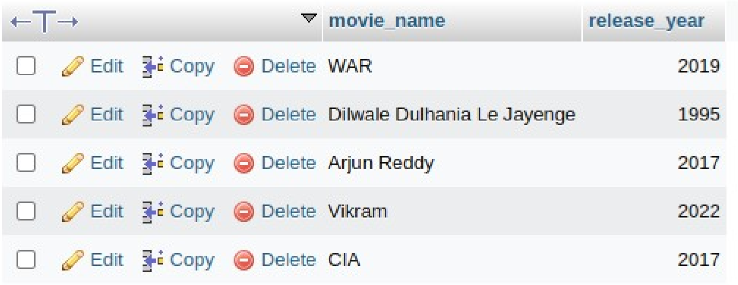
movie_name

View clause

-- creating a view films and their years by combining data from 2 tables
-- i.e Movies and Directed_by

```
CREATE VIEW Movie_year AS
SELECT movie_name,release_year FROM Movies as M, Directed_by AS D
WHERE M.movie_id=D.movie_id;
```

OUTPUT



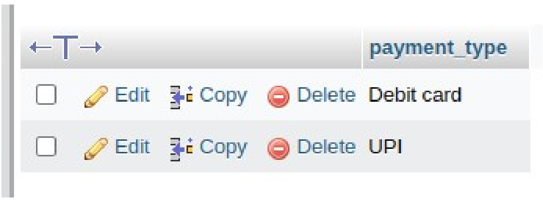
←T→	movie_name	release_year
□ Edit Copy Delete	WAR	2019
□ Edit Copy Delete	Dilwale Dulhania Le Jayenge	1995
□ Edit Copy Delete	Arjun Reddy	2017
□ Edit Copy Delete	Vikram	2022
□ Edit Copy Delete	CIA	2017

GROUP BY AND HAVING CLAUSE

-- listing the payment which is used by more than 1 customer

```
SELECT ALL payment_type FROM `Payment`
GROUP BY payment_type HAVING COUNT(*)>1;
```

OUTPUT



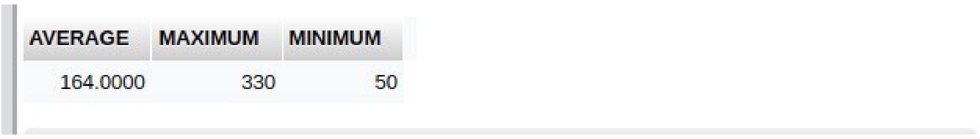
←T→	payment_type
□ Edit Copy Delete	Debit card
□ Edit Copy Delete	UPI

AGGREGATE CLAUSES

-- show maximum, minimum and average payment amounts

```
SELECT AVG(amount) as AVERAGE,MAX(amount) AS
MAXIMUM ,MIN(amount) AS MINIMUM  FROM `Payment`;
```

OUTPUT



AVERAGE	MAXIMUM	MINIMUM
164.0000	330	50

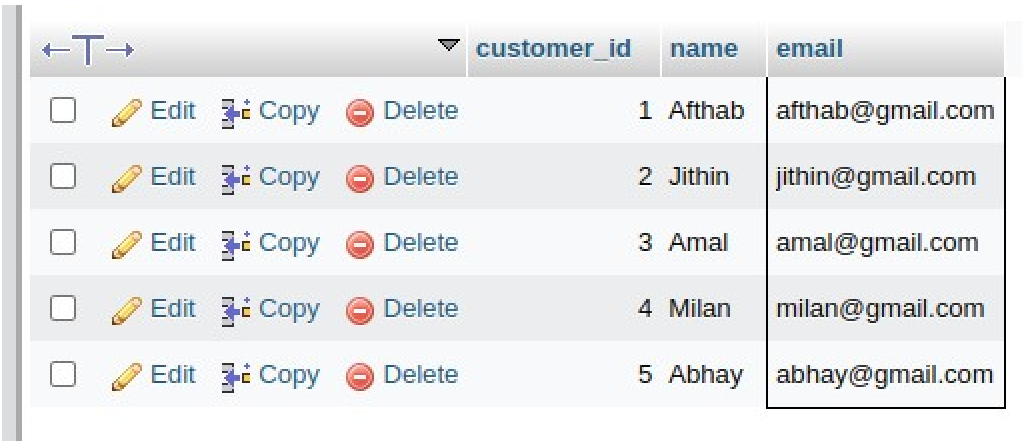
LIKE CLAUSE

-- List all customers with valid emails from Customer table

```
INSERT INTO `Customer`(`name`, `email`) VALUES
('Gautham','gauthamgmailcom');
```

```
SELECT `customer_id`, `name`, `email` FROM `Customer` WHERE email
LIKE '%@%.%';
```

OUTPUT



←T→ ▼	customer_id	name	email
□ Edit Copy Delete	1	Afthab	afthab@gmail.com
□ Edit Copy Delete	2	Jithin	jithin@gmail.com
□ Edit Copy Delete	3	Amal	amal@gmail.com
□ Edit Copy Delete	4	Milan	milan@gmail.com
□ Edit Copy Delete	5	Abhay	abhay@gmail.com