

Final Lab Report

Database Systems

Name:-

* Syed Hassan As’ad (SP21-BAI-023)
* Muhammad Ali Zafar (SP21-BAI-016)

Class:-

BSAI-5A

Submitted To:-

Dr. Madiha Yousaf

Task 1: Scope document

Identify the topic of your final project and explain various modules which will be present in the DBMS.

Topic: “Cinema Management System”

Abstract:

Cinema Management System deals with all the tasks that are required by a cinema to operate successfully. There will be two main parts of our system. One part will be used by the administration and other will be used by the public for booking their tickets. The administration section covers tasks like add employees, edit employee details. The admin add will also have a section which will enable the administration to add a movie, cancel a show or change the show timings. It will also have section to block and unblock the users. The other part will be for the users. It will have two sections. The guest section will allow the users to find out about the shows but to book a ticket they have to login to the system through their username and password. We will be using database for data retrieval and data storage. All these functionalities will be included in our Cinema Management System.

1. Introduction:

Traditionally, it is observed that people must wait in long lines to book their ticket which is tiresome. Also, if someone has bought a ticket, he cannot cancel it. The manager must keep many books to maintain the record of his employees. If any one of the books is lost, valuable data related to that system is lost forever. Our System will help automate the activities related to the cinema. It will cover different activities performed by the users which will provide enhanced techniques for maintaining the data up to date resulting in greater efficiency. The system will provide real life understanding of movie-ticket booking and activities performed by different users in cinema management system. In short, our system will enhance the procedure of ticket booking and related activities of the users and the administration.

2. Proposed System:

Our System will provide information about the movies and shows currently running in the cinema. It will allow the customers to book a ticket in a show. It will allow the user to cancel their bookings. It will allow users to create their profile for logging in the system. It will also facilitate the administration section in many ways. The admin will be able to see the number of tickets sold in a show by entering the show ID. It will allow to add new employees in the system. It will also help to edit the details of the employees. It will have a section to add a show. It will also have edit a show and cancel a show section. There will be refreshment section which will allow the user to see the available refreshment items in the cinema. Users will also be able to add a coupon or discount when booking their ticket.

3. Advantages/Benefits of Proposed System:

The advantages of our system are as follows:

• User can book a ticket

• User can cancel a ticket

• User can view the available shows

• User can view shows in a movie

• Employee can add show

• Employee can remove show

• Employee can add movie in a show

• Employee can remove a movie from system

• Admin can add employees

• Admin can add a hall

• Admin can check total tickets booked in a movie

• Admin can remove a hall

4. Scope:

The project will be used as Cinema Management System and will help the administration, employees and the customers in managing the system efficiently. The project will have three ends customer, employee and admin end. On the main login page, there will be an option to login according to your role. Admin will be allowed to manage employees including add or remove employees and managing employee details. The admin will also be allowed to manage the halls in the cinema. The employee end will also allow the employees to manage shows which will include add or remove shows, manage shows timings and dates, or remove any show in case of cancellation. The employees will also be able to add movies in a certain show and add movies generally in the system. The customer end will allow the customers to view details of a certain show or movies by title or genre. The customer should be able to select a particular show and book its tickets. Customer will also be able to cancel the booking by entering their ticket ID. The system will have a notifications section which will allow the customers to see new available shows, cancellation of a show or new available movies.

5. Modules:

Following are the modules for Cinema Management System:

• User Profiling

• Ticket Booking / Cancellation

• Hall / Show Management

• Movie Management

• Refreshments

• Notifications

• Discount

• Feedback

• Tickets Management

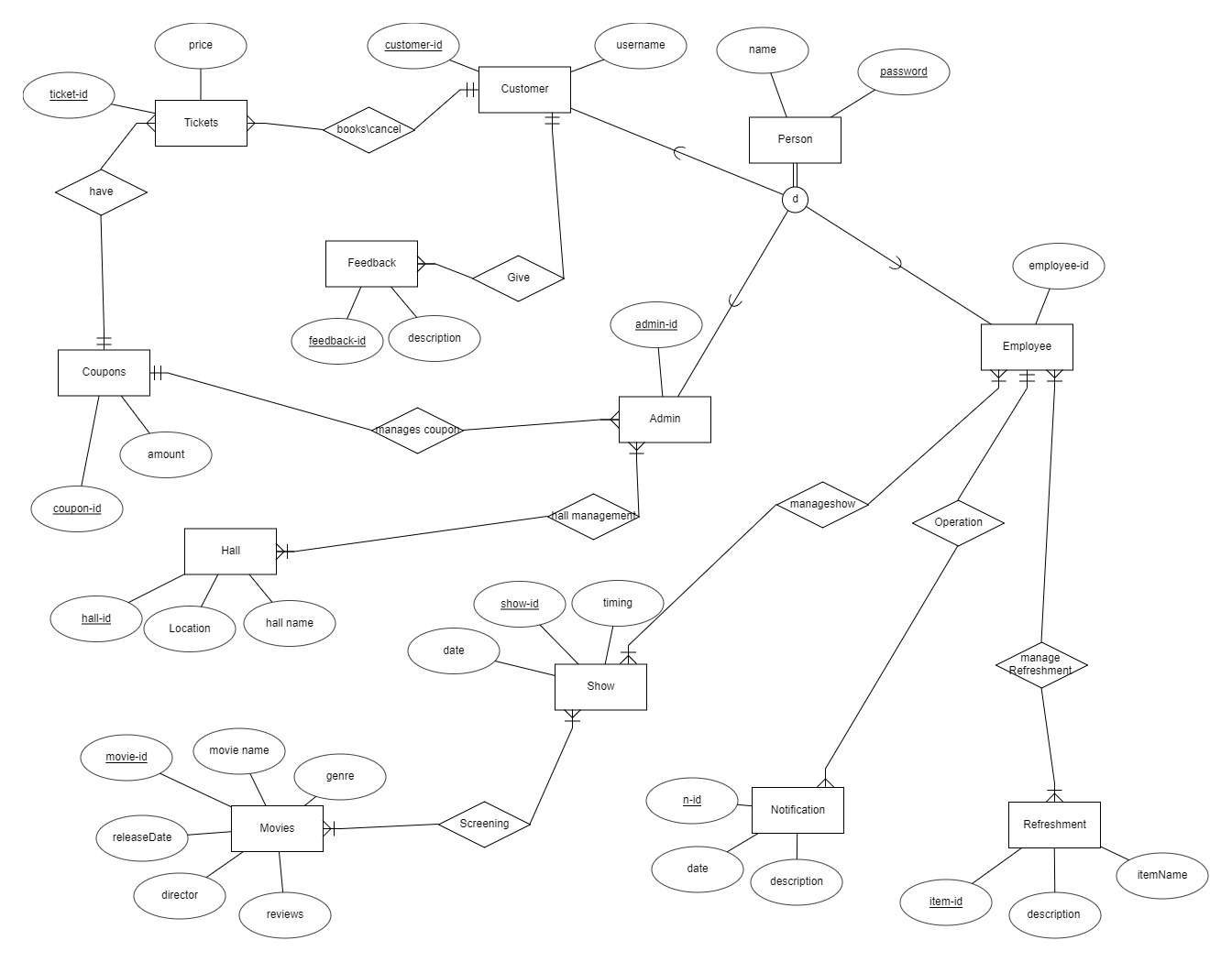
Task 2: Creating DB for your project

Implement your proposed DB using the questions given in lab- assignment.

Task 3: lab mid-term

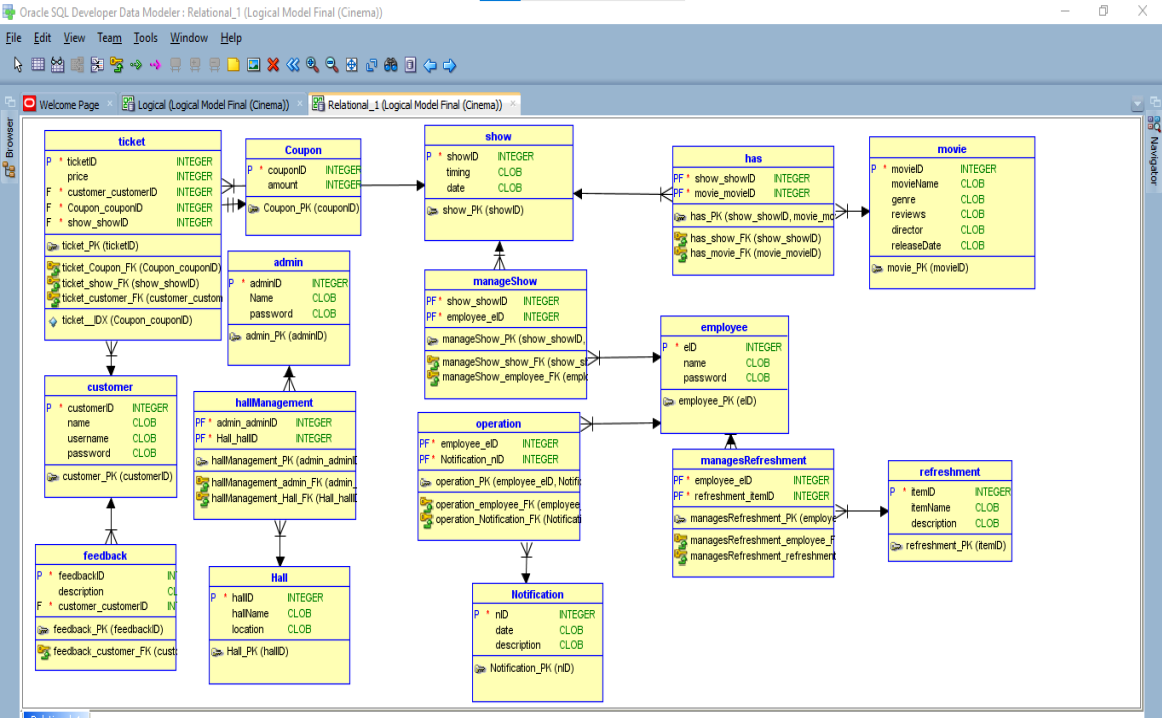
Part-I: Design and develop your own project EER diagram, create the same using a modeling tool of your choice.

Enhanced ER Diagram:





Relational Schema:



Part-II: Export the DDL script, corresponding to the relational schema of the EERD created in Part-I, to Oracle Database System.

DDL Script:

CREATE TABLE admin

( adminid INT NOT NULL,

name VARCHAR(10),

password VARCHAR(10),

PRIMARY KEY (adminid));

CREATE TABLE coupon

(couponid INT NOT NULL,

amount INTEGER,

PRIMARY KEY ( couponid ));

CREATE TABLE customer

(customerid INT NOT NULL,

 name VARCHAR(10),

username VARCHAR(10),

password VARCHAR(10),

PRIMARY KEY ( customerid ));

CREATE TABLE employee

( eid INT NOT NULL,

 name VARCHAR (10),

password VARCHAR(10),

PRIMARY KEY ( eid ));

CREATE TABLE feedback (

feedbackid INT NOT NULL,

description VARCHAR(10),

customer\_customerid INT NOT NULL,

PRIMARY KEY ( feedbackid ),

FOREIGN KEY ( customer\_customerid ) REFERENCES customer( customer\_customerid ));

CREATE TABLE hall (

hallid INT NOT NULL,

hallname VARCHAR(10),

location VARCHAR(10),

PRIMARY KEY ( hallid ));

CREATE TABLE hallmanagement (

admin\_adminid INT NOT NULL,

hall\_hallid INT NOT NULL,

PRIMARY KEY ( admin\_adminid, hall\_hallid ),

FOREIGN KEY ( admin\_adminid ) REFERENCES admin ( adminid ),

FOREIGN KEY ( hall\_hallid ) REFERENCES hall ( hallid ));

CREATE TABLE screening (

show\_showid INT NOT NULL,

movie\_movieid INT NOT NULL,

PRIMARY KEY ( show\_showid, movie\_movieid ),

FOREIGN KEY ( movie\_movieid ) REFERENCES movie ( movieid ),

FOREIGN KEY ( show\_showid ) REFERENCES show ( showid ));

CREATE TABLE managecoupons

( admin\_adminid INTEGER NOT NULL,

coupon\_couponid INT NOT NULL,

PRIMARY KEY ( admin\_adminid, coupon\_couponid ),

FOREIGN KEY ( admin\_adminid ) REFERENCES admin ( adminid ),

FOREIGN KEY ( coupon\_couponid ) REFERENCES coupon ( couponid ));

CREATE TABLE manages (

admin\_adminid INT NOT NULL,

employee\_eid INT NOT NULL,

PRIMARY KEY ( admin\_adminid, employee\_eid ),

FOREIGN KEY ( admin\_adminid ) REFERENCES admin ( adminid ),

FOREIGN KEY ( employee\_eid ) REFERENCES employee ( eid ));

CREATE TABLE manageshow (

show\_showid INT NOT NULL,

employee\_eid INT NOT NULL,

PRIMARY KEY ( show\_showid, employee\_eid ),

FOREIGN KEY ( employee\_eid ) REFERENCES employee ( eid ),

FOREIGN KEY ( show\_showid ) REFERENCES show ( showid ));

CREATE TABLE managesv1 (

employee\_eid INT NOT NULL,

refreshment\_itemid INTEGER NOT NULL,

PRIMARY KEY ( employee\_eid, refreshment\_itemid ),

FOREIGN KEY ( employee\_eid ) REFERENCES employee ( eid ),

FOREIGN KEY ( refreshment\_itemid ) REFERENCES refreshment ( itemid ));

CREATE TABLE movie ( movieid INTEGER NOT NULL, moviename CLOB,

genre VARCHAR(10),

reviews VARCHAR(10),

director VARCHAR(10),

releasedate VARCHAR(10),

PRIMARY KEY ( movieid ));

CREATE TABLE notification

( nid INTEGER NOT NULL,

 "date" VARCHAR(10),

 description VARCHAR(10),

PRIMARY KEY ( nid ));

CREATE TABLE operation (

    employee\_eid INTEGER NOT NULL,

    notification\_nid INTEGER NOT NULL,

    PRIMARY KEY (employee\_eid, notification\_nid),

    FOREIGN KEY (employee\_eid) REFERENCES employee (eid),

    FOREIGN KEY (notification\_nid) REFERENCES notification (nid)

);

CREATE TABLE refreshment ( itemid INT NOT NULL,

itemname VARCHAR(10),

description VARCHAR(10),

PRIMARY KEY ( itemid ));

CREATE TABLE show (

showid INT NOT NULL,

timing varchar(10),

"date" VARCHAR (10),

PRIMARY KEY ( showid ));

CREATE TABLE ticket (

ticketid INTEGER NOT NULL,

price INTEGER,

customer\_customerid INTEGER NOT NULL,

coupon\_couponid INTEGER NOT NULL,

show\_showid INTEGER NOT NULL ,

PRIMARY KEY ( ticketid ),

FOREIGN KEY ( coupon\_couponid ) REFERENCES coupon ( couponid ),

FOREIGN KEY ( customer\_customerid ) REFERENCES customer ( customerid ),

FOREIGN KEY ( show\_showid ) REFERENCES show ( showid ));

Part-III: Implement the EER constraints 1-1, 1-many, many to many, “U”, “O”, “d”, “Partial participation”, “Total participation” , multi-attribute, super-class, sub-class, multiple-inheritance ( Can Java language ).

Part-IV: Insert a few meaningful tuples in the resultant relations

INSERT INTO admin (adminid, name, password) VALUES (1, 'John Doe', 'admin123');

INSERT INTO admin (adminid, name, password) VALUES (2, 'Jane Smith', 'password456');

INSERT INTO admin (adminid, name, password) VALUES (3, 'Sarah Adams', 'admin987');

INSERT INTO admin (adminid, name, password) VALUES (4, 'David Wilson', 'passadmin123');

INSERT INTO admin (adminid, name, password) VALUES (5, 'Karen Thompson', 'adminpass456');

INSERT INTO admin (adminid, name, password) VALUES (6, 'Michael Smith', 'admin13');

INSERT INTO admin (adminid, name, password) VALUES (7, 'Jessica Johnson', 'password436');

INSERT INTO admin (adminid, name, password) VALUES (8, 'Andrew Wilson', 'admin654');

INSERT INTO admin (adminid, name, password) VALUES (9, 'Emma Thompson', 'passadmin907');

INSERT INTO admin (adminid, name, password) VALUES (10, 'Christopher Davis', 'adminpass107');



INSERT INTO coupon (couponid, amount) VALUES (1, 10);

INSERT INTO coupon (couponid, amount) VALUES (2, 20);

INSERT INTO coupon (couponid, amount) VALUES (3, 15);

INSERT INTO coupon (couponid, amount) VALUES (4, 25);

INSERT INTO coupon (couponid, amount) VALUES (5, 30);

INSERT INTO coupon (couponid, amount) VALUES (6, 5);

INSERT INTO coupon (couponid, amount) VALUES (7, 10);

INSERT INTO coupon (couponid, amount) VALUES (8, 15);

INSERT INTO coupon (couponid, amount) VALUES (9, 20);

INSERT INTO coupon (couponid, amount) VALUES (10, 25);

INSERT INTO coupon (couponid, amount) VALUES (11, 10);

INSERT INTO coupon (couponid, amount) VALUES (12, 20);

INSERT INTO coupon (couponid, amount) VALUES (13, 15);

INSERT INTO coupon (couponid, amount) VALUES (14, 25);

INSERT INTO coupon (couponid, amount) VALUES (15, 30);

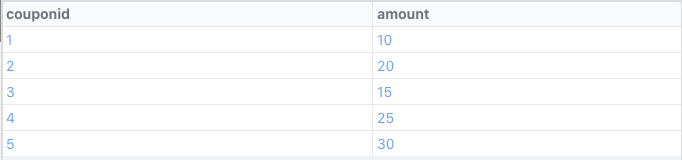
INSERT INTO coupon (couponid, amount) VALUES (16, 5);

INSERT INTO coupon (couponid, amount) VALUES (17, 10);

INSERT INTO coupon (couponid, amount) VALUES (18, 15);

INSERT INTO coupon (couponid, amount) VALUES (19, 20);

INSERT INTO coupon (couponid, amount) VALUES (20, 25);



INSERT INTO customer (customerid, name, username, password) VALUES (1, 'Alice Johnson', 'alice23', 'pass789');

INSERT INTO customer (customerid, name, username, password) VALUES (2, 'Bob Thompson', 'bob88', 'secureuwd');

INSERT INTO customer (customerid, name, username, password) VALUES (3, 'Mark Davis', 'mark85', 'customer763');

INSERT INTO customer (customerid, name, username, password) VALUES (4, 'Emily Wilson', 'emily28', 'passcustomer463');

INSERT INTO customer (customerid, name, username, password) VALUES (5, 'Samuel Roberts', 'sam98', 'customerpass456');

INSERT INTO customer (customerid, name, username, password) VALUES (6, 'Sophie Wilson', 'sophie34', 'pass73');

INSERT INTO customer (customerid, name, username, password) VALUES (7, 'Henry Roberts', 'henry56', 'securepwd');

INSERT INTO customer (customerid, name, username, password) VALUES (8, 'Grace Davis', 'grace72', 'customer23');

INSERT INTO customer (customerid, name, username, password) VALUES (9, 'Sebastian Thompson', 'seb89', 'passcustomer46');

INSERT INTO customer (customerid, name, username, password) VALUES (10, 'Lily Adams', 'lily99', 'customerpass79');

INSERT INTO customer (customerid, name, username, password) VALUES (11, 'Jackson Wilson', 'jackson77', 'pass12');

INSERT INTO customer (customerid, name, username, password) VALUES (12, 'Victoria Davis', 'victoria44', 'securewd');

INSERT INTO customer (customerid, name, username, password) VALUES (13, 'Ryan Thompson', 'ryan78', 'customer23');

INSERT INTO customer (customerid, name, username, password) VALUES (14, 'Zoe Adams', 'zoe22', 'passcustomer456');

INSERT INTO customer (customerid, name, username, password) VALUES (15, 'Christopher Johnson', 'chris33', 'customerpass89');

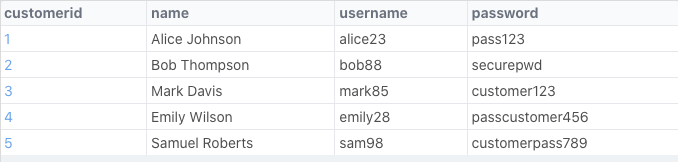
INSERT INTO customer (customerid, name, username, password) VALUES (16, 'Nora Wilson', 'nora12', 'pass123');

INSERT INTO customer (customerid, name, username, password) VALUES (17, 'Elijah Davis', 'elijah90', 'securepwd');

INSERT INTO customer (customerid, name, username, password) VALUES (18, 'Avery Thompson', 'avery46', 'customer123');

INSERT INTO customer (customerid, name, username, password) VALUES (19, 'Hannah Adams', 'hannah65', 'passcustomer456');

INSERT INTO customer (customerid, name, username, password) VALUES (20, 'Benjamin Johnson', 'ben88', 'customerpass789');



INSERT INTO employee (eid, name, password) VALUES (1, 'Michael Brown', 'emp367');

INSERT INTO employee (eid, name, password) VALUES (2, 'Emily Davis', 'emp789');

INSERT INTO employee (eid, name, password) VALUES (3, 'Jennifer Johnson', 'emp987');

INSERT INTO employee (eid, name, password) VALUES (4, 'Robert Smith', 'employeepass321');

INSERT INTO employee (eid, name, password) VALUES (5, 'Laura Davis', 'emp456');

INSERT INTO employee (eid, name, password) VALUES (6, 'Alexander Wilson', 'emp654');

INSERT INTO employee (eid, name, password) VALUES (7, 'Charlotte Davis', 'emp978');

INSERT INTO employee (eid, name, password) VALUES (8, 'Gabriel Johnson', 'emp123');

INSERT INTO employee (eid, name, password) VALUES (9, 'Scarlett Roberts', 'employeepass987');

INSERT INTO employee (eid, name, password) VALUES (10, 'Henry Davis', 'emp785');

INSERT INTO employee (eid, name, password) VALUES (11, 'Zoey Johnson', 'emp799');

INSERT INTO employee (eid, name, password) VALUES (12, 'Carter Adams', 'emp947');

INSERT INTO employee (eid, name, password) VALUES (13, 'Mila Thompson', 'employeepass237');

INSERT INTO employee (eid, name, password) VALUES (14, 'Liam Davis', 'emp496');

INSERT INTO employee (eid, name, password) VALUES (15, 'Harper Roberts', 'emp729');

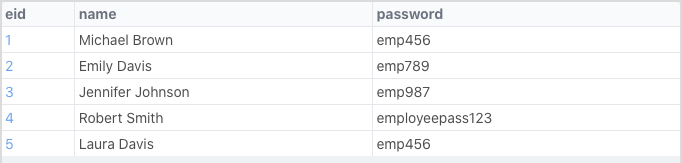
INSERT INTO employee (eid, name, password) VALUES (16, 'Lucas Johnson', 'emp907');

INSERT INTO employee (eid, name, password) VALUES (17, 'Aria Adams', 'employeepass923');

INSERT INTO employee (eid, name, password) VALUES (18, 'Jackson Thompson', 'emp102');

INSERT INTO employee (eid, name, password) VALUES (19, 'Evelyn Davis', 'emp105');

INSERT INTO employee (eid, name, password) VALUES (20, 'Caleb Roberts', 'emp621');



INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (1, 'Great service!', 1);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (2, 'Could be better.', 2);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (3, 'The seats were uncomfortable.', 3);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (4, 'Excellent customer service!', 4);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (5, 'The sound quality needs improvement.', 5);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (6, 'Loved the movie!', 6);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (7, 'Average experience.', 7);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (8, 'Great service!', 8);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (9, 'The food was delicious.', 9);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (10, 'The staff was friendly.', 10);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (11, 'Movie was disappointing.', 11);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (12, 'Could be better.', 12);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (13, 'Service was slow.', 13);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (14, 'Loved the ambiance.', 14);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (15, 'Movie was thrilling!', 15);

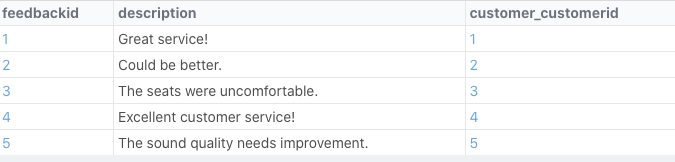
INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (16, 'Could improve cleanliness.', 16);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (17, 'Theater seats were uncomfortable.', 17);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (18, 'Good value for money.', 18);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (19, 'Not worth the price.', 19);

INSERT INTO feedback (feedbackid, description, customer\_customerid) VALUES (20, 'Highly recommended!', 20);



INSERT INTO hall (hallid, hallname, location) VALUES (1, 'Hall A', 'New York');

INSERT INTO hall (hallid, hallname, location) VALUES (2, 'Hall B', ' Phoenix');

INSERT INTO hall (hallid, hallname, location) VALUES (3, 'Hall C', ' Atlanta ');

INSERT INTO hall (hallid, hallname, location) VALUES (4, 'Hall D', 'San Francisco');

INSERT INTO hall (hallid, hallname, location) VALUES (5, 'Hall E', 'Miami');

INSERT INTO hall (hallid, hallname, location) VALUES (6, 'Hall F', 'Seattle');

INSERT INTO hall (hallid, hallname, location) VALUES (7, 'Hall G', 'Houston');

INSERT INTO hall (hallid, hallname, location) VALUES (8, 'Hall H', 'Boston');

INSERT INTO hall (hallid, hallname, location) VALUES (9, 'Hall I', 'Atlanta');

INSERT INTO hall (hallid, hallname, location) VALUES (10, 'Hall J', 'Denver');

INSERT INTO hall (hallid, hallname, location) VALUES (11, 'Hall K', 'Dallas');

INSERT INTO hall (hallid, hallname, location) VALUES (12, 'Hall L', 'Phoenix');

INSERT INTO hall (hallid, hallname, location) VALUES (13, 'Hall M', 'Las Vegas');

INSERT INTO hall (hallid, hallname, location) VALUES (14, 'Hall N', ' Austin ');

INSERT INTO hall (hallid, hallname, location) VALUES (15, 'Hall O', ' Seattle ');

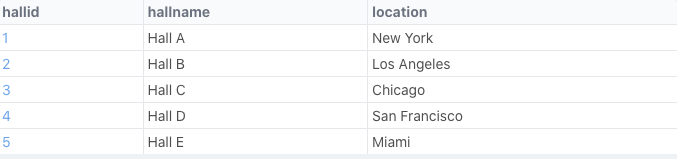
INSERT INTO hall (hallid, hallname, location) VALUES (16, 'Hall P', 'Washington');

INSERT INTO hall (hallid, hallname, location) VALUES (17, 'Hall Q', 'Philadelphia');

INSERT INTO hall (hallid, hallname, location) VALUES (18, 'Hall R', 'Austin');

INSERT INTO hall (hallid, hallname, location) VALUES (19, 'Hall S', 'Miami');

INSERT INTO hall (hallid, hallname, location) VALUES (20, 'Hall T', 'San Antonio');



INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (1, 1);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (2, 2);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (3, 3);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (4, 4);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (5, 5);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (6, 6);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (7, 7);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (8, 8);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (9, 9);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (10, 10);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (11, 11);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (12, 12);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (13, 13);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (14, 14);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (15, 15);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (16, 16);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (17, 17);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (18, 18);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (19, 19);

INSERT INTO hallmanagement (admin\_adminid, hall\_hallid) VALUES (20, 20);



INSERT INTO screening (show\_showid, movie\_movieid) VALUES (1, 1);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (2, 2);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (3, 3);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (4, 4);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (5, 5);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (6, 6);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (7, 7);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (8, 8);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (9, 9);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (10, 10);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (11, 11);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (12, 12);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (13, 13);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (14, 14);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (15, 15);

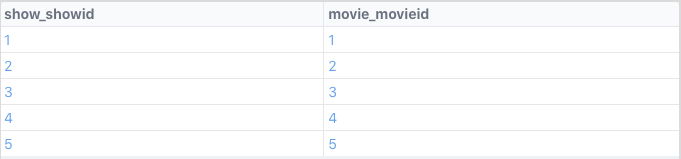
INSERT INTO screening (show\_showid, movie\_movieid) VALUES (16, 16);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (17, 17);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (18, 18);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (19, 19);

INSERT INTO screening (show\_showid, movie\_movieid) VALUES (20, 20);



INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (1, 1);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (2, 2);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (3, 3);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (4, 4);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (5, 5);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (6, 6);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (7, 7);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (8, 8);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (9, 9);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (10, 10);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (11, 11);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (12, 12);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (13, 13);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (14, 14);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (15, 15);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (16, 16);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (17, 17);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (18, 18);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (19, 19);

INSERT INTO managecoupons (admin\_adminid, coupon\_couponid) VALUES (20, 20);



INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (1, 1);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (2, 2);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (3, 3);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (4, 4);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (5, 5);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (6, 6);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (7, 7);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (8, 8);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (9, 9);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (10, 10);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (11, 11);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (12, 12);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (13, 13);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (14, 14);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (15, 15);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (16, 16);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (17, 17);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (18, 18);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (19, 19);

INSERT INTO manages (admin\_adminid, employee\_eid) VALUES (20, 20);



INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (1, 1);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (2, 2);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (3, 3);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (4, 4);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (5, 5);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (6, 6);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (7, 7);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (8, 8);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (9, 9);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (10, 10);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (11, 11);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (12, 12);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (13, 13);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (14, 14);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (15, 15);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (16, 16);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (17, 17);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (18, 18);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (19, 19);

INSERT INTO manageshow (show\_showid, employee\_eid) VALUES (20, 20);



INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (1, 1);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (2, 2);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (3, 3);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (4, 4);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (5, 5);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (6, 6);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (7, 7);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (8, 8);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (9, 9);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (10, 10);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (11, 11);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (12, 12);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (13, 13);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (14, 14);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (15, 15);

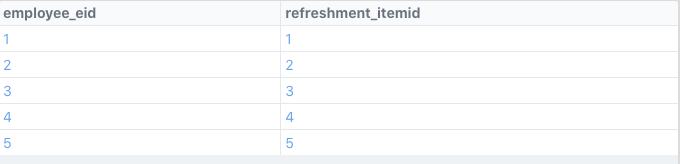
INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (16, 16);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (17, 17);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (18, 18);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (19, 19);

INSERT INTO managesv1 (employee\_eid, refreshment\_itemid) VALUES (20, 20);



INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (1, 'The Great Gatsby', 'Drama', 'Positive', 'Baz Luhrmann', '2013-05-10');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (2, 'Inception', 'Sci-Fi', 'Highly acclaimed', 'Christopher Nolan', '2010-07-16');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (3, 'The Shawshank Redemption', 'Drama', 'Highly acclaimed', 'Frank Darabont', '1994-09-23');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (4, 'The Dark Knight', 'Action', 'Blockbuster', 'Christopher Nolan', '2008-07-18');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (5, 'Pulp Fiction', 'Crime', 'Cult classic', 'Quentin Tarantino', '1994-10-14');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (6, 'Movie F', 'Action', 'Good', 'Director F', '2023-05-01');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (7, 'Movie G', 'Drama', 'Excellent', 'Director G', '2023-05-02');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (8, 'Movie H', 'Comedy', 'Average', 'Director H', '2023-05-03');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (9, 'Movie I', 'Thriller', 'Good', 'Director I', '2023-05-04');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (10, 'Movie J', 'Action', 'Excellent', 'Director J', '2023-05-05');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (11, 'Movie K', 'Drama', 'Average', 'Director K', '2023-05-06');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (12, 'Movie L', 'Comedy', 'Good', 'Director L', '2023-05-07');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (13, 'Movie M', 'Thriller', 'Excellent', 'Director M', '2023-05-08');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (14, 'Movie N', 'Action', 'Average', 'Director N', '2023-05-09');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (15, 'Movie O', 'Drama', 'Good', 'Director O', '2023-05-10');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (16, 'Movie P', 'Comedy', 'Excellent', 'Director P', '2023-05-11');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (17, 'Movie Q', 'Thriller', 'Average', 'Director Q', '2023-05-12');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (18, 'Movie R', 'Action', 'Good', 'Director R', '2023-05-13');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (19, 'Movie S', 'Drama', 'Excellent', 'Director S', '2023-05-14');

INSERT INTO movie (movieid, moviename, genre, reviews, director, releasedate) VALUES (20, 'Movie T', 'Comedy', 'Average', 'Director T', '2023-05-15');



INSERT INTO notification (nid, "date", description) VALUES (1, '2023-05-15', 'Upcoming event: Movie Marathon');

INSERT INTO notification (nid, "date", description) VALUES (2, '2023-05-16', 'Schedule change: Movie showtime');

INSERT INTO notification (nid, "date", description) VALUES (3, '2023-05-17', 'Special discount on tickets!');

INSERT INTO notification (nid, "date", description) VALUES (4, '2023-05-18', 'Movie premiere tonight!');

INSERT INTO notification (nid, "date", description) VALUES (5, '2023-05-19', 'Upcoming movie marathon event');

INSERT INTO notification (nid, "date", description) VALUES (6, '2023-05-20', 'New movie release!');

INSERT INTO notification (nid, "date", description) VALUES (7, '2023-05-21', 'Special screening event');

INSERT INTO notification (nid, "date", description) VALUES (8, '2023-05-22', 'Limited-time offer: Buy one, get one free');

INSERT INTO notification (nid, "date", description) VALUES (9, '2023-05-23', 'Upcoming movie premiere');

INSERT INTO notification (nid, "date", description) VALUES (10, '2023-05-24', 'Discounted tickets for students');

INSERT INTO notification (nid, "date", description) VALUES (11, '2023-05-25', 'Exclusive VIP event');

INSERT INTO notification (nid, "date", description) VALUES (12, '2023-05-26', 'Newly added movie showtime');

INSERT INTO notification (nid, "date", description) VALUES (13, '2023-05-27', 'Special offer: Free popcorn with ticket purchase');

INSERT INTO notification (nid, "date", description) VALUES (14, '2023-05-28', 'Upcoming movie marathon');

INSERT INTO notification (nid, "date", description) VALUES (15, '2023-05-29', 'Limited seating available for popular movie');

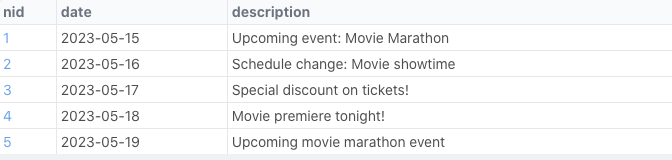
INSERT INTO notification (nid, "date", description) VALUES (16, '2023-05-30', 'Exclusive advanced screening');

INSERT INTO notification (nid, "date", description) VALUES (17, '2023-05-31', 'Newly released movie now showing');

INSERT INTO notification (nid, "date", description) VALUES (18, '2023-06-01', 'Special event: Q&A with movie director');

INSERT INTO notification (nid, "date", description) VALUES (19, '2023-06-02', 'Upcoming blockbuster movie');

INSERT INTO notification (nid, "date", description) VALUES (20, '2023-06-03', 'Limited-time discount on refreshments');



INSERT INTO operation (employee\_eid, notification\_nid) VALUES (1, 1);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (2, 2);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (3, 3);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (4, 4);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (5, 5);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (6, 6);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (7, 7);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (8, 8);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (9, 9);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (10, 10);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (11, 11);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (12, 12);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (13, 13);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (14, 14);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (15, 15);

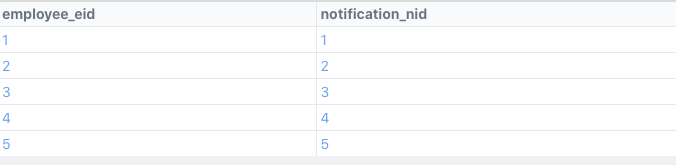
INSERT INTO operation (employee\_eid, notification\_nid) VALUES (16, 16);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (17, 17);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (18, 18);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (19, 19);

INSERT INTO operation (employee\_eid, notification\_nid) VALUES (20, 20);



INSERT INTO refreshment (itemid, itemname, description) VALUES (1, 'Popcorn', 'Classic movie snack');

INSERT INTO refreshment (itemid, itemname, description) VALUES (2, 'Soda', 'Refreshing beverage');

INSERT INTO refreshment (itemid, itemname, description) VALUES (3, 'Nachos', 'Crunchy snack');

INSERT INTO refreshment (itemid, itemname, description) VALUES (4, 'Hot Dog', 'Classic cinema food');

INSERT INTO refreshment (itemid, itemname, description) VALUES (5, 'Ice Cream', 'Sweet treat');

INSERT INTO refreshment (itemid, itemname, description) VALUES (6, 'Candy', 'Assorted sweet treats');

INSERT INTO refreshment (itemid, itemname, description) VALUES (7, 'Popcorn Combo', 'Popcorn with a drink');

INSERT INTO refreshment (itemid, itemname, description) VALUES (8, 'Pizza', 'Delicious pizza slices');

INSERT INTO refreshment (itemid, itemname, description) VALUES (9, 'Mozzarella Sticks', 'Cheesy appetizers');

INSERT INTO refreshment (itemid, itemname, description) VALUES (10, 'Pretzel', 'Salty snack');

INSERT INTO refreshment (itemid, itemname, description) VALUES (11, 'Coffee', 'Hot beverage');

INSERT INTO refreshment (itemid, itemname, description) VALUES (12, 'Smoothie', 'Refreshing fruit drink');

INSERT INTO refreshment (itemid, itemname, description) VALUES (13, 'Cheeseburger', 'Classic fast food');

INSERT INTO refreshment (itemid, itemname, description) VALUES (14, 'Fries', 'Crispy side dish');

INSERT INTO refreshment (itemid, itemname, description) VALUES (15, 'Brownie', 'Rich chocolate dessert');

INSERT INTO refreshment (itemid, itemname, description) VALUES (16, 'Sundae', 'Ice cream with toppings');

INSERT INTO refreshment (itemid, itemname, description) VALUES (17, 'Salad', 'Healthy option');

INSERT INTO refreshment (itemid, itemname, description) VALUES (18, 'Chicken Wings', 'Spicy appetizers');

INSERT INTO refreshment (itemid, itemname, description) VALUES (19, 'Hot Chocolate', 'Warm and comforting drink');

INSERT INTO refreshment (itemid, itemname, description) VALUES (20, 'Soft Pretzel', 'Freshly baked pretzel');



INSERT INTO show (showid, timing, "date") VALUES (1, '14:00', '2023-05-20');

INSERT INTO show (showid, timing, "date") VALUES (2, '19:30', '2023-05-21');

INSERT INTO show (showid, timing, "date") VALUES (3, '16:30', '2023-05-22');

INSERT INTO show (showid, timing, "date") VALUES (4, '21:00', '2023-05-23');

INSERT INTO show (showid, timing, "date") VALUES (5, '13:00', '2023-05-24');

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (6, '2023-06-04 10:00:00', 6);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (7, '2023-06-04 12:30:00', 7);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (8, '2023-06-04 15:00:00', 8);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (9, '2023-06-04 17:30:00', 9);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (10, '2023-06-04 20:00:00', 10);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (11, '2023-06-05 10:00:00', 11);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (12, '2023-06-05 12:30:00', 12);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (13, '2023-06-05 15:00:00', 13);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (14, '2023-06-05 17:30:00', 14);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (15, '2023-06-05 20:00:00', 15);

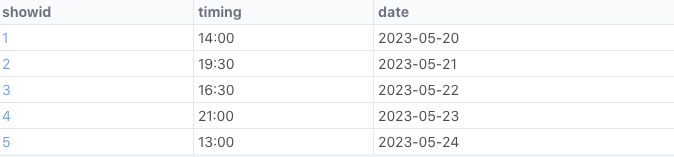
INSERT INTO show (showid, showtime, screen\_screenid) VALUES (16, '2023-06-06 10:00:00', 16);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (17, '2023-06-06 12:30:00', 17);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (18, '2023-06-06 15:00:00', 18);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (19, '2023-06-06 17:30:00', 19);

INSERT INTO show (showid, showtime, screen\_screenid) VALUES (20, '2023-06-06 20:00:00', 20);



INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (1, 15, 1, 1, 1);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (2, 20, 2, 2, 2);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (3, 18, 3, 2, 3);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (4, 22, 4, 1, 4);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (5, 16, 5, 3, 5);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (6, 12, 6, 4, 6);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (7, 18, 7, 5, 7);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (8, 14, 8, 3, 8);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (9, 20, 9, 2, 9);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (10, 16, 10, 1, 10);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (11, 22, 11, 6, 11);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (12, 17, 12, 7, 12);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (13, 19, 13, 8, 13);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (14, 15, 14, 9, 14);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (15, 21, 15, 10, 15);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (16, 13, 16, 11, 16);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (17, 23, 17, 12, 17);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (18, 16, 18, 13, 18);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (19, 24, 19, 14, 19);

INSERT INTO ticket (ticketid, price, customer\_customerid, coupon\_couponid, show\_showid) VALUES (20, 18, 20, 15, 20);



Part-V: Now in this part, you are going to first create a meaningful query in English and then in SQL for each of the

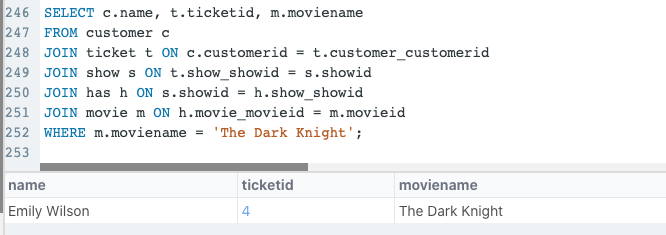
following situations:

* Think of a need that will require joining at least 3 tables.
* Think of a need that will require the use of join with grouping and aggregation.
* Think of a need that involves the use of a sub-query.
* Think of a need that involves the need of using any of the set operators.

Situation 1: Need requiring joining at least 3 tables.

English Query: Show the details of customers who have purchased tickets for a specific movie.

SQL Query:



Situation 2: Need requiring join with grouping and aggregation.

English Query: Calculate the total revenue generated by each movie.

SQL Query:

SELECT m.moviename, SUM(t.price) AS total\_revenue

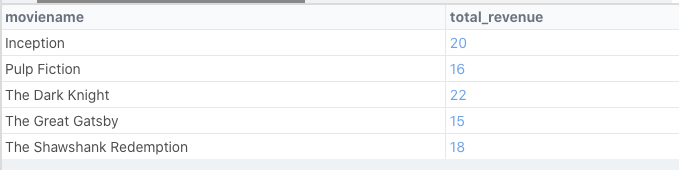
FROM movie m

JOIN has h ON m.movieid = h.movie\_movieid

JOIN show s ON h.show\_showid = s.showid

JOIN ticket t ON s.showid = t.show\_showid

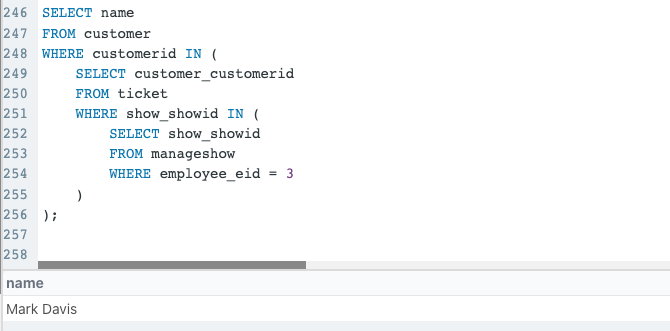
GROUP BY m.moviename;



Situation 3: Need involving the use of a sub-query.

English Query: Find the customer names who have purchased tickets for shows managed by a specific employee.

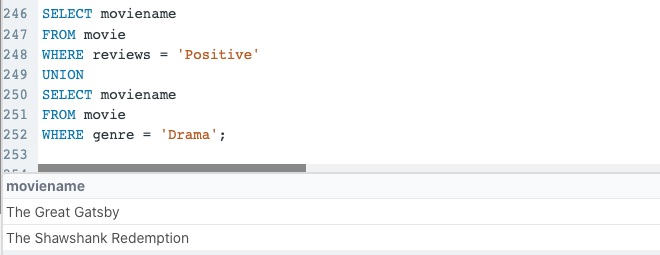
SQL Query:



Situation 4: Need involving the use of set operators.

English Query: Retrieve the list of movies that either have positive reviews or are classified as a drama.

SQL Query:



MongoDB

JSon Files

Admin:

[

{

"adminid": 1,

"name": "John Doe",

"password": "admin123"

},

{

"adminid": 2,

"name": "Jane Smith",

"password": "password456"

},

{

"adminid": 3,

"name": "Sarah Adams",

"password": "admin987"

},

{

"adminid": 4,

"name": "David Wilson",

"password": "passadmin123"

},

{

"adminid": 5,

"name": "Karen Thompson",

"password": "adminpass456"

},

{

"adminid": 6,

"name": "Michael Smith",

"password": "admin13"

},

{

"adminid": 7,

"name": "Jessica Johnson",

"password": "password436"

},

{

"adminid": 8,

"name": "Andrew Wilson",

"password": "admin654"

},

{

"adminid": 9,

"name": "Emma Thompson",

"password": "passadmin907"

},

{

"adminid": 10,

"name": "Christopher Davis",

"password": "adminpass107"

}

]

Customer:

[

{

"customerid": 1,

"name": "Alice Johnson",

"username": "alice23",

"password": "pass789"

},

{

"customerid": 2,

"name": "Bob Thompson",

"username": "bob88",

"password": "secureuwd"

},

{

"customerid": 3,

"name": "Mark Davis",

"username": "mark85",

"password": "customer763"

},

{

"customerid": 4,

"name": "Emily Wilson",

"username": "emily28",

"password": "passcustomer463"

},

{

"customerid": 5,

"name": "Samuel Roberts",

"username": "sam98",

"password": "customerpass456"

},

{

"customerid": 6,

"name": "Sophie Wilson",

"username": "sophie34",

"password": "pass73"

},

{

"customerid": 7,

"name": "Henry Roberts",

"username": "henry56",

"password": "securepwd"

},

{

"customerid": 8,

"name": "Grace Davis",

"username": "grace72",

"password": "customer23"

},

{

"customerid": 9,

"name": "Sebastian Thompson",

"username": "seb89",

"password": "passcustomer46"

},

{

"customerid": 10,

"name": "Lily Adams",

"username": "lily99",

"password": "customerpass79"

},

{

"customerid": 11,

"name": "Jackson Wilson",

"username": "jackson77",

"password": "pass12"

},

{

"customerid": 12,

"name": "Victoria Davis",

"username": "victoria44",

"password": "securewd"

},

{

"customerid": 13,

"name": "Ryan Thompson",

"username": "ryan78",

"password": "customer23"

},

{

"customerid": 14,

"name": "Zoe Adams",

"username": "zoe22",

"password": "passcustomer456"

},

{

"customerid": 15,

"name": "Christopher Johnson",

"username": "chris33",

"password": "customerpass89"

},

{

"customerid": 16,

"name": "Nora Wilson",

"username": "nora12",

"password": "pass123"

},

{

"customerid": 17,

"name": "Elijah Davis",

"username": "elijah90",

"password": "securepwd"

},

{

"customerid": 18,

"name": "Avery Thompson",

"username": "avery46",

"password": "customer123"

},

{

"customerid": 19,

"name": "Hannah Adams",

"username": "hannah65",

"password": "passcustomer456"

},

{

"customerid": 20,

"name": "Benjamin Johnson",

"username": "ben88",

"password": "customerpass789"

}

]

Employee:

[

{

"eid": 1,

"name": "Michael Brown",

"password": "emp367"

},

{

"eid": 2,

"name": "Emily Davis",

"password": "emp789"

},

{

"eid": 3,

"name": "Jennifer Johnson",

"password": "emp987"

},

{

"eid": 4,

"name": "Robert Smith",

"password": "employeepass321"

},

{

"eid": 5,

"name": "Laura Davis",

"password": "emp456"

},

{

"eid": 6,

"name": "Alexander Wilson",

"password": "emp654"

},

{

"eid": 7,

"name": "Charlotte Davis",

"password": "emp978"

},

{

"eid": 8,

"name": "Gabriel Johnson",

"password": "emp123"

},

{

"eid": 9,

"name": "Scarlett Roberts",

"password": "employeepass987"

},

{

"eid": 10,

"name": "Henry Davis",

"password": "emp785"

},

{

"eid": 11,

"name": "Zoey Johnson",

"password": "emp799"

},

{

"eid": 12,

"name": "Carter Adams",

"password": "emp947"

},

{

"eid": 13,

"name": "Mila Thompson",

"password": "employeepass237"

},

{

"eid": 14,

"name": "Liam Davis",

"password": "emp496"

},

{

"eid": 15,

"name": "Harper Roberts",

"password": "emp729"

},

{

"eid": 16,

"name": "Lucas Johnson",

"password": "emp907"

},

{

"eid": 17,

"name": "Aria Adams",

"password": "employeepass923"

},

{

"eid": 18,

"name": "Jackson Thompson",

"password": "emp102"

},

{

"eid": 19,

"name": "Evelyn Davis",

"password": "emp105"

},

{

"eid": 20,

"name": "Caleb Roberts",

"password": "emp621"

}

]

Feedback:

[

{

"feedbackid": 1,

"description": "Great service!",

"customer\_customerid": 1

},

{

"feedbackid": 2,

"description": "Could be better.",

"customer\_customerid": 2

},

{

"feedbackid": 3,

"description": "The seats were uncomfortable.",

"customer\_customerid": 3

},

{

"feedbackid": 4,

"description": "Excellent customer service!",

"customer\_customerid": 4

},

{

"feedbackid": 5,

"description": "The sound quality needs improvement.",

"customer\_customerid": 5

},

{

"feedbackid": 6,

"description": "Loved the movie!",

"customer\_customerid": 6

},

{

"feedbackid": 7,

"description": "Average experience.",

"customer\_customerid": 7

},

{

"feedbackid": 8,

"description": "Great service!",

"customer\_customerid": 8

},

{

"feedbackid": 9,

"description": "The food was delicious.",

"customer\_customerid": 9

},

{

"feedbackid": 10,

"description": "The staff was friendly.",

"customer\_customerid": 10

},

{

"feedbackid": 11,

"description": "Movie was disappointing.",

"customer\_customerid": 11

},

{

"feedbackid": 12,

"description": "Could be better.",

"customer\_customerid": 12

},

{

"feedbackid": 13,

"description": "Service was slow.",

"customer\_customerid": 13

},

{

"feedbackid": 14,

"description": "Loved the ambiance.",

"customer\_customerid": 14

},

{

"feedbackid": 15,

"description": "Movie was thrilling!",

"customer\_customerid": 15

},

{

"feedbackid": 16,

"description": "Could improve cleanliness.",

"customer\_customerid": 16

},

{

"feedbackid": 17,

"description": "Theater seats were uncomfortable.",

"customer\_customerid": 17

},

{

"feedbackid": 18,

"description": "Good value for money.",

"customer\_customerid": 18

},

{

"feedbackid": 19,

"description": "Not worth the price.",

"customer\_customerid": 19

},

{

"feedbackid": 20,

"description": "Highly recommended!",

"customer\_customerid": 20

}

]

Movie:

[

{

"movieid": 1,

"moviename": "The Great Gatsby",

"genre": "Drama",

"reviews": "Positive",

"director": "Baz Luhrmann",

"releasedate": "2013-05-10"

},

{

"movieid": 2,

"moviename": "Inception",

"genre": "Sci-Fi",

"reviews": "Highly acclaimed",

"director": "Christopher Nolan",

"releasedate": "2010-07-16"

},

{

"movieid": 3,

"moviename": "The Shawshank Redemption",

"genre": "Drama",

"reviews": "Highly acclaimed",

"director": "Frank Darabont",

"releasedate": "1994-09-23"

},

{

"movieid": 4,

"moviename": "The Dark Knight",

"genre": "Action",

"reviews": "Blockbuster",

"director": "Christopher Nolan",

"releasedate": "2008-07-18"

},

{

"movieid": 5,

"moviename": "Pulp Fiction",

"genre": "Crime",

"reviews": "Cult classic",

"director": "Quentin Tarantino",

"releasedate": "1994-10-14"

},

{

"movieid": 6,

"moviename": "Movie F",

"genre": "Action",

"reviews": "Good",

"director": "Director F",

"releasedate": "2023-05-01"

},

{

"movieid": 7,

"moviename": "Movie G",

"genre": "Drama",

"reviews": "Excellent",

"director": "Director G",

"releasedate": "2023-05-02"

},

{

"movieid": 8,

"moviename": "Movie H",

"genre": "Comedy",

"reviews": "Average",

"director": "Director H",

"releasedate": "2023-05-03"

},

{

"movieid": 9,

"moviename": "Movie I",

"genre": "Thriller",

"reviews": "Good",

"director": "Director I",

"releasedate": "2023-05-04"

},

{

"movieid": 10,

"moviename": "Movie J",

"genre": "Action",

"reviews": "Excellent",

"director": "Director J",

"releasedate": "2023-05-05"

},

{

"movieid": 11,

"moviename": "Movie K",

"genre": "Drama",

"reviews": "Average",

"director": "Director K",

"releasedate": "2023-05-06"

},

{

"movieid": 12,

"moviename": "Movie L",

"genre": "Comedy",

"reviews": "Good",

"director": "Director L",

"releasedate": "2023-05-07"

},

{

"movieid": 13,

"moviename": "Movie M",

"genre": "Thriller",

"reviews": "Excellent",

"director": "Director M",

"releasedate": "2023-05-08"

},

{

"movieid": 14,

"moviename": "Movie N",

"genre": "Action",

"reviews": "Average",

"director": "Director N",

"releasedate": "2023-05-09"

},

{

"movieid": 15,

"moviename": "Movie O",

"genre": "Drama",

"reviews": "Good",

"director": "Director O",

"releasedate": "2023-05-10"

},

{

"movieid": 16,

"moviename": "Movie P",

"genre": "Comedy",

"reviews": "Excellent",

"director": "Director P",

"releasedate": "2023-05-11"

},

{

"movieid": 17,

"moviename": "Movie Q",

"genre": "Thriller",

"reviews": "Average",

"director": "Director Q",

"releasedate": "2023-05-12"

},

{

"movieid": 18,

"moviename": "Movie R",

"genre": "Action",

"reviews": "Good",

"director": "Director R",

"releasedate": "2023-05-13"

},

{

"movieid": 19,

"moviename": "Movie S",

"genre": "Drama",

"reviews": "Excellent",

"director": "Director S",

"releasedate": "2023-05-14"

},

{

"movieid": 20,

"moviename": "Movie T",

"genre": "Comedy",

"reviews": "Average",

"director": "Director T",

"releasedate": "2023-05-15"

}

]

Notification:

[

{

"nid": 1,

"date": "2023-05-15",

"description": "Upcoming event: Movie Marathon"

},

{

"nid": 2,

"date": "2023-05-16",

"description": "Schedule change: Movie timing"

},

{

"nid": 3,

"date": "2023-05-17",

"description": "Special discount on tickets!"

},

{

"nid": 4,

"date": "2023-05-18",

"description": "Movie premiere tonight!"

},

{

"nid": 5,

"date": "2023-05-19",

"description": "Upcoming movie marathon event"

},

{

"nid": 6,

"date": "2023-05-20",

"description": "New movie release!"

},

{

"nid": 7,

"date": "2023-05-21",

"description": "Special screening event"

},

{

"nid": 8,

"date": "2023-05-22",

"description": "Limited-time offer: Buy one, get one free"

},

{

"nid": 9,

"date": "2023-05-23",

"description": "Upcoming movie premiere"

},

{

"nid": 10,

"date": "2023-05-24",

"description": "Discounted tickets for students"

},

{

"nid": 11,

"date": "2023-05-25",

"description": "Exclusive VIP event"

},

{

"nid": 12,

"date": "2023-05-26",

"description": "Special offer: Combo deals"

},

{

"nid": 13,

"date": "2023-05-27",

"description": "Movie premiere this weekend!"

},

{

"nid": 14,

"date": "2023-05-28",

"description": "Special screening: Director's cut"

},

{

"nid": 15,

"date": "2023-05-29",

"description": "Upcoming movie marathon event"

},

{

"nid": 16,

"date": "2023-05-30",

"description": "New movie release!"

},

{

"nid": 17,

"date": "2023-05-31",

"description": "Exclusive Q&A session with the director"

},

{

"nid": 18,

"date": "2023-06-01",

"description": "Special discount: Family packages"

},

{

"nid": 19,

"date": "2023-06-02",

"description": "Upcoming movie premiere"

},

{

"nid": 20,

"date": "2023-06-03",

"description": "Limited-time offer: Free popcorn"

}

]

Show:

[

{

"showid": 1,

"timing": "14:00",

"date": "2023-05-20"

},

{

"showid": 2,

"timing": "19:30",

"date": "2023-05-21"

},

{

"showid": 3,

"timing": "16:30",

"date": "2023-05-22"

},

{

"showid": 4,

"timing": "21:00",

"date": "2023-05-23"

},

{

"showid": 5,

"timing": "13:00",

"date": "2023-05-24"

},

{

"showid": 6,

"timing": "14:00",

"date": "2023-06-04"

},

{

"showid": 7,

"timing": "10:00",

"date": "2023-04-12"

},

{

"showid": 8,

"timing": "11:00",

"date": "2023-07-26"

},

{

"showid": 9,

"timing": "06:00",

"date": "2023-09-31"

},

{

"showid": 10,

"timing": "02:00",

"date": "2023-04-24"

}

]

Ticket:

[

{

"ticketid": 1,

"price": 15,

"customer\_customerid": 1,

"coupon\_couponid": 1,

"show\_showid": 1

},

{

"ticketid": 2,

"price": 20,

"customer\_customerid": 2,

"coupon\_couponid": 2,

"show\_showid": 2

},

{

"ticketid": 3,

"price": 18,

"customer\_customerid": 3,

"coupon\_couponid": 2,

"show\_showid": 3

},

{

"ticketid": 4,

"price": 22,

"customer\_customerid": 4,

"coupon\_couponid": 1,

"show\_showid": 4

},

{

"ticketid": 5,

"price": 16,

"customer\_customerid": 5,

"coupon\_couponid": 3,

"show\_showid": 5

},

{

"ticketid": 6,

"price": 12,

"customer\_customerid": 6,

"coupon\_couponid": 4,

"show\_showid": 6

},

{

"ticketid": 7,

"price": 18,

"customer\_customerid": 7,

"coupon\_couponid": 5,

"show\_showid": 7

},

{

"ticketid": 8,

"price": 14,

"customer\_customerid": 8,

"coupon\_couponid": 3,

"show\_showid": 8

},

{

"ticketid": 9,

"price": 20,

"customer\_customerid": 9,

"coupon\_couponid": 2,

"show\_showid": 9

},

{

"ticketid": 10,

"price": 16,

"customer\_customerid": 10,

"coupon\_couponid": 1,

"show\_showid": 10

},

{

"ticketid": 11,

"price": 22,

"customer\_customerid": 11,

"coupon\_couponid": 6,

"show\_showid": 11

},

{

"ticketid": 12,

"price": 17,

"customer\_customerid": 12,

"coupon\_couponid": 7,

"show\_showid": 12

},

{

"ticketid": 13,

"price": 19,

"customer\_customerid": 13,

"coupon\_couponid": 8,

"show\_showid": 13

},

{

"ticketid": 14,

"price": 15,

"customer\_customerid": 14,

"coupon\_couponid": 9,

"show\_showid": 14

},

{

"ticketid": 15,

"price": 21,

"customer\_customerid": 15,

"coupon\_couponid": 10,

"show\_showid": 15

},

{

"ticketid": 16,

"price": 13,

"customer\_customerid": 16,

"coupon\_couponid": 11,

"show\_showid": 16

},

{

"ticketid": 17,

"price": 23,

"customer\_customerid": 17,

"coupon\_couponid": 12,

"show\_showid": 17

},

{

"ticketid": 18,

"price": 16,

"customer\_customerid": 18,

"coupon\_couponid": 13,

"show\_showid": 18

},

{

"ticketid": 19,

"price": 24,

"customer\_customerid": 19,

"coupon\_couponid": 14,

"show\_showid": 19

},

{

"ticketid": 20,

"price": 18,

"customer\_customerid": 20,

"coupon\_couponid": 15,

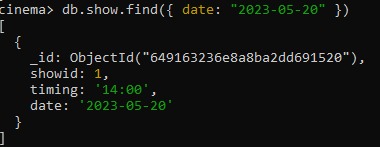
"show\_showid": 20

}

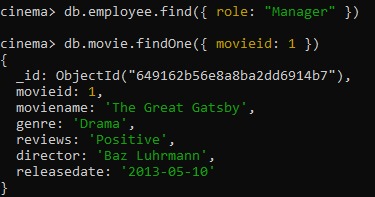
]

MongoDB Queries:

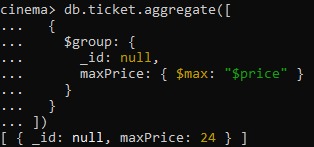
1. Retrieve the shows scheduled for a specific date



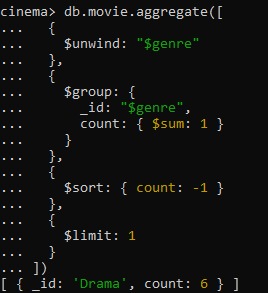
1. Retrieve the details of a specific movie by its ID



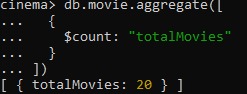
1. Find the maximum price among all tickets.



1. Determine the most frequent movie genre.



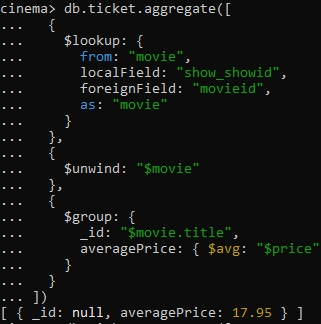
1. Count the total number of movies in the database.



1. Get the minimum and maximum ticket prices per movie.



1. Calculate the average ticket price per movie.



1. Retrieve all tickets purchased by a specific customer.

db.Ticket.find({ customer\_customerid: 1 })

1. Retrieve all notifications sent to a specific employee.

db.Notification.find({ employee\_eid: 1 })

1. Find the number of shows scheduled per month.

db.Show.aggregate([

{

$group: {

\_id: { $month: "$date" },

showCount: { $sum: 1 }

}

},

{

$sort: { \_id: 1 }

}

])

Front End

Code (Python)

**from** **tkinter** **import**\*

**from** **tkinter** **import** ttk

**from** **PIL** **import** Image,ImageTk

**from** **tkinter** **import** messagebox

**import** **mysql.connector**

**def** **main**():

win=Tk()

app=Welcome\_Window(win)

win.mainloop()

**class** **Welcome\_Window**:

**def** **\_init\_**(self,root):

self.root=root

self.root.title("CINEMA MANAGEMENT SYSTEM --> 'Welcome' ")

self.root.geometry("1366x768+0+0")

self.bg = ImageTk.PhotoImage(file="C:\Ali\DB\Cinema main.jpg")

lbl\_bg= Label(self.root,image=self.bg)

lbl\_bg.place(x=**0**,y=**0**,relwidth=**1**,relheight=**1**)

frame = Frame(self.root,bg="black")

frame.place(x=**320**,y=**570**,width=**700**,height=**100**)

#img1 = Image.open("C:\Ali\DB\CInema 1.jpg")

#self.photoimg1=ImageTk.PhotoImage(img1)

#lblimg1=Label(image=self.photoimg1, borderwidth=0)

#lblimg1.place(x=390,y=450,width=700,height=200)

get\_str = Label(frame,text="CINEMA MANAGEMENT SYSTEM",font = ("times new roman",**20**,"bold"),fg="white",bg="black")

get\_str.place(x=**150**,y=**0**)

adminbtn = Button(frame,text="ADMIN",font=("times new roman",**20**,"bold"), command= self.adminlogin\_window,bd=**3**,relief =RIDGE,fg = "white", bg = "black",activeforeground="white",activebackground="white")

adminbtn.place(x=**70**,y=**50**,width=**130**,height=**40**)

employeebtn = Button(frame,text="EMPLOYEE",font=("times new roman",**20**,"bold"),command= self.employeelogin\_window,bd=**3**,relief =RIDGE,fg = "white", bg = "black",activeforeground="white",activebackground="white")

employeebtn.place(x=**290**,y=**50**,width=**170**,height=**40**)

customerbtn = Button(frame,text="CUSTOMER",font=("times new roman",**20**,"bold"),command= self.customerlogin\_window,bd=**3**,relief =RIDGE,fg = "white", bg = "black",activeforeground="white",activebackground="white")

customerbtn.place(x=**510**,y=**50**,width=**170**,height=**40**)

**def** **adminlogin\_window** (self):

self.new\_window=Toplevel(self.root)

self.app=Admin\_Login\_Window(self.new\_window)

**def** **employeelogin\_window** (self):

self.new\_window=Toplevel(self.root)

self.app=Employee\_Login\_Window(self.new\_window)

**def** **customerlogin\_window** (self):

self.new\_window=Toplevel(self.root)

self.app=Customer\_Login\_Window(self.new\_window)

**class** **Admin\_Login\_Window**:

**def** **\_init\_**(self,root):

self.root=root

self.root.title("Admin")

self.root.geometry("1366x768+0+0")

self.bg = ImageTk.PhotoImage(file="C:\Ali\DB\cinepax-jpg.webp")

lbl\_bg= Label(self.root,image=self.bg)

lbl\_bg.place(x=**50**,y=**50**,width=**600**,height=**600**)

#img1 = Image.open("C:\Ali\Machine Learning\cinepax-jpg.webp")

#self.photoimg1=ImageTk.PhotoImage(img1)

#lblimg1=Label(image=self.photoimg1, borderwidth=0)

#lblimg1.place(x=50,y=50,width=600,height=600)

frame = Frame(self.root,bg="#65350F")

frame.place(x=**750**,y=**150**,width=**400**,height=**410**)

get\_str = Label(frame,text="GET STARTED",font = ("times new roman",**20**,"bold"),fg="black",bg="#65350F")

get\_str.place(x=**100**,y=**30**)

#label

username=lbl=Label (frame, text="Username", font=("times new roman", **15**, "bold"), fg="black", bg="#65350F")

username.place(x=**50**, y=**120**)

self.txtuser=ttk. Entry(frame, font=("times new roman", **15**, "bold"))

self.txtuser.place(x=**50**, y=**145**,width=**270**)

password=lbl=Label (frame, text="Password", font=("times new roman", **15**, "bold"), fg="black", bg="#65350F")

password.place(x=**50**, y=**200**)

self.txtpass=ttk. Entry(frame, font=("times new roman", **15**, "bold"))

self.txtpass.place(x=**50**, y=**225**,width=**270**)

checkbtn=Checkbutton(frame, text="I Agree The Terms & Conditions", font=("times new roman", **15**, "bold"), onvalue=**1**,offvalue=**0**, bg = "#65350F",activeforeground="white",activebackground="#65350F")

checkbtn.place(x=**50**, y=**270**)

loginbtn = Button(frame,text="login",font=("Calibri",**15**),command=self.login,bd=**3**,relief =RIDGE,fg = "white", bg = "black",activeforeground="white",activebackground="black")

loginbtn.place(x=**150**,y=**310**,width=**90**,height=**30**)

**def** **login**(self):

**if** self.txtuser.get()=="" **or** self.txtpass.get()=="":

messagebox.showerror("Error","all field required")

**elif** self.txtuser.get()=="kapu" **and** self.txtpass.get()=="ashu":

messagebox.showinfo("Success")

**else**:

conn=mysql.connector.connect(host="localhost", user="sys", password="aliatta123", database="mydata", port = **1521**)

my\_cursor=conn.cursor()

my\_cursor.execute("select from Admin where name=%s and password=%s",(

self.txtpass.get(),

self.txtuser.get()

))

row=my\_cursor.fetchone()

# print (row)

**if** row==**None**:

messagebox.showerror("Error", "Inavalid Username & password")

**else**:

open\_main=messagebox.askyesno ("YesNo", "Access only admin")

**if** open\_main>**0**:

self.new\_window=Toplevel(self.root)

self.app=Welcome\_Window(self.new\_window)

**else**:

**if** **not** open\_main:

**return**

conn.commit()

conn.close()

**class** **Employee\_Login\_Window**:

**def** **\_init\_**(self,root):

self.root=root

self.root.title("Employee")

self.root.geometry("1366x768+0+0")

self.bg = ImageTk.PhotoImage(file="C:\Ali\DB\cinema 2.jpg")

lbl\_bg= Label(self.root,image=self.bg)

lbl\_bg.place(x=**0**,y=**0**,relwidth=**1**,relheight=**1**)

#img1 = Image.open("C:\Ali\Machine Learning\cinepax-jpg.webp")

#self.photoimg1=ImageTk.PhotoImage(img1)

#lblimg1=Label(image=self.photoimg1, borderwidth=0)

#lblimg1.place(x=50,y=50,width=600,height=600)

frame = Frame(self.root,bg="#550015")

frame.place(x=**450**,y=**150**,width=**400**,height=**410**)

get\_str = Label(frame,text="GET STARTED",font = ("times new roman",**20**,"bold"),fg="black",bg="#550015")

get\_str.place(x=**100**,y=**30**)

#label

username=lbl=Label (frame, text="Username", font=("times new roman", **15**, "bold"), fg="black", bg="#550015")

username.place(x=**50**, y=**120**)

self.txtuser=ttk. Entry(frame, font=("times new roman", **15**, "bold"))

self.txtuser.place(x=**50**, y=**145**,width=**270**)

password=lbl=Label (frame, text="Password", font=("times new roman", **15**, "bold"), fg="black", bg="#550015")

password.place(x=**50**, y=**200**)

self.txtpass=ttk. Entry(frame, font=("times new roman", **15**, "bold"))

self.txtpass.place(x=**50**, y=**225**,width=**270**)

checkbtn=Checkbutton(frame, text="I Agree The Terms & Conditions", font=("times new roman", **15**, "bold"), onvalue=**1**,offvalue=**0**, bg = "#550015",activeforeground="white",activebackground="#65350F")

checkbtn.place(x=**50**, y=**270**)

loginbtn = Button(frame,text="login",font=("Calibri",**15**),command=self.Elogin,bd=**3**,relief =RIDGE,fg = "white", bg = "black",activeforeground="white",activebackground="black")

loginbtn.place(x=**150**,y=**310**,width=**90**,height=**30**)

**def** **Elogin**(self):

**if** self.txtuser.get()=="" **or** self.txtpass.get()=="":

messagebox.showerror("Error","all field required")

**elif** self.txtuser.get()=="kapu" **and** self.txtpass.get()=="ashu":

messagebox.showinfo("Success")

**else**:

conn=mysql.connector.connect(host="localhost", user="sys", password="aliatta123", database="oracle")

my\_cursor=conn.cursor()

my\_cursor.execute("select from register where email=%s and password=%s",(

self.txtpass.get(),

self.txtuser.get()

))

row=my\_cursor.fetchone()

# print (row)

**if** row==**None**:

messagebox.showerror("Error", "Inavalid Username & password")

**else**:

open\_main=messagebox.askyesno ("YesNo", "Access only admin")

**if** open\_main>**0**:

self.new\_window=Toplevel(self.root)

self.app=Welcome\_Window(self.new\_window)

**else**:

**if** **not** open\_main:

**return**

conn.commit()

conn.close()

**class** **Customer\_Login\_Window**:

**def** **\_init\_**(self,root):

self.root=root

self.root.title("Customer")

self.root.geometry("1366x768+0+0")

self.bg = ImageTk.PhotoImage(file="C:\Ali\DB\Felix-mooneeram-evlkOfkQ5rE-unsplash.jpg")

lbl\_bg= Label(self.root,image=self.bg)

lbl\_bg.place(x=**0**,y=**0**,relwidth=**1**,relheight=**1**)

#img1 = Image.open("C:\Ali\DB\cinepax-jpg.webp")

#self.photoimg1=ImageTk.PhotoImage(img1)

#lblimg1=Label(image=self.photoimg1, borderwidth=0)

#lblimg1.place(x=50,y=50,width=600,height=600)

frame = Frame(self.root,bg="#550015")

frame.place(x=**450**,y=**150**,width=**400**,height=**410**)

get\_str = Label(frame,text="GET STARTED",font = ("times new roman",**20**,"bold"),fg="black",bg="#550015")

get\_str.place(x=**100**,y=**30**)

#label

username=lbl=Label (frame, text="Username", font=("times new roman", **15**, "bold"), fg="black", bg="#550015")

username.place(x=**50**, y=**120**)

self.txtuser=ttk. Entry(frame, font=("times new roman", **15**, "bold"))

self.txtuser.place(x=**50**, y=**145**,width=**270**)

password=lbl=Label (frame, text="Password", font=("times new roman", **15**, "bold"), fg="black", bg="#550015")

password.place(x=**50**, y=**200**)

self.txtpass=ttk. Entry(frame, font=("times new roman", **15**, "bold"))

self.txtpass.place(x=**50**, y=**225**,width=**270**)

checkbtn=Checkbutton(frame, text="I Agree The Terms & Conditions", font=("times new roman", **15**, "bold"), onvalue=**1**,offvalue=**0**, bg = "#550015",activeforeground="white",activebackground="#65350F")

checkbtn.place(x=**50**, y=**270**)

loginbtn = Button(frame,text="login",font=("Calibri",**15**),command=self.Clogin,bd=**3**,relief =RIDGE,fg = "white", bg = "black",activeforeground="white",activebackground="black")

loginbtn.place(x=**150**,y=**310**,width=**90**,height=**30**)

**def** **Clogin**(self):

**if** self.txtuser.get()=="" **or** self.txtpass.get()=="":

messagebox.showerror("Error","all field required")

**elif** self.txtuser.get()=="kapu" **and** self.txtpass.get()=="ashu":

messagebox.showinfo("Success")

**else**:

conn=mysql.connector.connect(host="localhost", user="sys", password="aliatta123", database="oracle")

my\_cursor=conn.cursor()

my\_cursor.execute("select from register where email=%s and password=%s",(

self.txtpass.get(),

self.txtuser.get()

))

row=my\_cursor.fetchone()

# print (row)

**if** row==**None**:

messagebox.showerror("Error", "Inavalid Username & password")

**else**:

open\_main=messagebox.askyesno ("YesNo", "Access only admin")

**if** open\_main>**0**:

self.new\_window=Toplevel(self.root)

self.app=Welcome\_Window(self.new\_window)

**else**:

**if** **not** open\_main:

**return**

conn.commit()

conn.close()

**if** \_name\_ == "\_main\_":

main()

Images of Front End

