

DA-IICT, Gandhinagar

Autumn 2022

IT615 - Database Management Systems (DBMS)

Lab Report

Movie Ticket Booking System

Group ID - 17

Mentor :- Parth Modi

Himang Patel (202212005)

Laxit Shah(202212042)

**Acknowledgement**

We would like to express our special thanks to Our Prof. Minal Bhise as well as our project guide Parth Modi who gave us the golden opportunity to do this challenging and wonderful project on Movie Ticket Booking System, which helped us learning a whole new skill set with doing a lot of research and acquiring new skill sets and we are really thankful for that.

Table of Contents

[1 SRS 1](#_Toc120559804)

[1.1 Introduction 1](#_Toc120559805)

[1.2 Goals of this system 1](#_Toc120559806)

[1.3 Benefits of using this system: 1](#_Toc120559807)

[2. Problem Statement 2](#_Toc120559808)

[2.1 Defects of the old system: 2](#_Toc120559809)

[2.2 Basic Flow Of Online Movie Ticket Booking System 3](#_Toc120559810)

[2.3 Background Reading 3](#_Toc120559811)

[2.4 Requirements Collection Documentation 4](#_Toc120559812)

[2.4.1 Questionnaire 4](#_Toc120559813)

[2.4.2 Interview 8](#_Toc120559814)

[2.5 Requirement in the database 9](#_Toc120559815)

[2.5.1 Functional Requirement 9](#_Toc120559816)

[2.5.2 Non-Functional Requirement 9](#_Toc120559817)

[2.6 Business Constraints 10](#_Toc120559818)

[2.7 Assumptions 12](#_Toc120559819)

[2.8 User Categories 12](#_Toc120559820)

[2.9 Privileges 13](#_Toc120559821)

[3 ER Diagram 15](#_Toc120559822)

[3.1 Noun Analysis 15](#_Toc120559823)

[3.2 Accepted Nouns: 16](#_Toc120559824)

[3.3 Rejected Nouns 17](#_Toc120559825)

[3.4 Entity Relation Diagram 19](#_Toc120559826)

[3.5 ER-Diagram Explanation 19](#_Toc120559827)

[4. ER to relational mapping. 20](#_Toc120559828)

[5. List of relations with attributes and constraints 24](#_Toc120559829)

[6. ER to DDL 25](#_Toc120559830)

[7. Populating the data in tables 28](#_Toc120559831)

[Bibliography 57](#_Toc120559832)

**Movie Ticket Booking**

## 1 SRS

* 1. **Introduction**

The aim of the project is to develop an online ticket reservation system. The ticket reservation system is an online database that can be used easily . This application automates the booking of tickets and inquires about ticket availability. This database has a regular updates on the available movies and tickets availability details.

* 1. **Goals of this system**
* Show available movies in the near by cinemas.
* Record customer details
* Record the available tickets and the number of tickets bought.
* Show available beverages at the cinemas
  1. **Benefits of using this system:**
* Constant Service: It does not require pauses or breaks. It works continuously, giving access to information to every customer at any time of the day.
* User Friendly:This makes the user experience more enjoyable.It provides best services and to meet all their needs.
* Flexible :The online system allows customers to access services from the comfort of their homes. As long as users have access to a phone or computer, they can easily access available services.
* Less User Management: Making the service more accessible to customers means fewer people come over the counter to buy. It also reduces business costs as it does not require many people at the counter. Employees will be able to divert their energies elsewhere.
* Business Analytics: Businesses can easily measure their
* performance through analytics. It is easy to analyze digitally collected data. Customers can also provide feedback on services that the company can take into consideration.

# 2. Problem Statement

Movie ticket reservation system is a process by which consumers book tickets online. Online movie ticket booking system is the process of ordering tickets from merchants that are sold online. A theater has Screens that run shows for different movies. Each show has a

particular movie, start time, duration and is played on screen in the theater. Each screen has an arrangement of seats that can be booked by users.The system will reserve the tickets .Any user with an internet connection can access it.

Cinemas are considered an integral part of the city and contribute to defining local identity and geography.Watching movies is one of the most popular cultural activities outside the home, as it affects some social, economic and cultural phenomena of modern society. It also helps to preserve collective memory with friends and family, as it constitutes the basic social and cultural practices associated with a particular place, which serves as a common reference or landmark for many individuals. Tickets can be collected at the counter and watching a movie with family and friends in the theater is the best form of entertainment after a busy day. But after hours of waiting in long queues to book tickets, all the excitement evaporated. We all know that manual systems are very old and time consuming. Also, people don't like handmade or traditional things anymore. They prefer things online. So this is also a big concern.

## 2.1 Defects of the old system:

* Slow processing
* Too long
* Uncertain
* Not very smooth
* No way to keep records

There are still fundamental problems with the traditional system that many people follow. There is still no proper solution to this problem.

Through this system, we are demonstrating an online movie ticket booking system. Theater Management System, an easy-to-understand and easy-to-use online ticket booking software that offers customers the simplicity of a fast, to the point system. Online Movie Ticket Booking was developed to overcome the common problems of manual systems. This program is supported to eliminate and in some cases reduce the problems faced by old system.In addition, the system is designed to meet the specific needs of movie theaters in order to conduct business flexibly and efficiently.The system provides complete information on which films are showing on all screens, along with details on times and available seats.Ticket reservations are made by online payment partners .Our booking system also allows you to cancel pre-booked tickets.Users can book tickets 24\*7 with this system.

To order tickets, the user must first log into the system .An easy and alternative way to buy movie tickets. Once the customer has entered information related to the movie, the system may include options to order tickets. Currently, cinema staff do not need to do anything to book tickets, as the process will be done by customers online. Online movie ticket booking benefits distributors and users. Users will be able to buy tickets anywhere in the world. Users of the system will be able to view new movies, schedules and movie locations.

## 2.2 Basic Flow Of Online Movie Ticket Booking System

* The system will help users to get all the details of movies available in their localities.
* On selecting a movie, the user will get the show details. In showing details, they can be able to see the available time and date for the movie.
* Users can select cinemas, halls, and seats according to their requirements. On selecting the cinema, the user will see the seat number. After the user selects an available seat, the system temporarily blocks the seat for 10 minutes. The database management system interacts with the main database and locks the seat to the user. The ticket will be temporarily reserved for the current user and all other users using a different online ticket service provider, and the seat will be unavailable for the next 10 minutes. Free up space for other pools if the user does not request entry within this period.
* Customers can purchase food and beverages from the stalls section.
* Employees of the cinema center are able to manage movies, shows, cinema, halls, and seats.
* Admin will manage the entire system.

## 2.3 Background Reading

BookMyShow

BookMyShow, a Mumbai-based company, is India's largest online platform that allows users to book tickets for various events including movies and sports. In its inception, it was known as the parent company Bigtree Entertainment Pvt. Launched in 1999 as a theater software vendor, it transformed its model into the platform, providing cloud-based services for events, movies, sports and entertainment. BookMyShow is India's largest online ticketing platform with a high percentage of brands, covering around 80% of online entertainment ticketing in India.

The wide geographical distribution of BookMyShows is its greatest advantage. It operates in more than 650 cities, has more than 5,000 screens, serves more than 30 million customers annually, sells more than 15 million tickets per month and manages 2 billion page views per month. In fact, almost half of Bollywood's box office collections are collected through BookMyShow, so they have a good understanding of the business. BookMyShow has also expanded its business to other countries eg. Indonesia, Sri Lanka, UAE, Caribbean, USA, UK and 1000+ companies also working with Ola, HP, Club Mahindra, Mastercard etc. to meet your leisure needs as entertainment partners.

BookMyShow offers a lot of comfort and convenience to customers as it offers 24/7 availability and is compatible with both Android and iOS application platforms. The interface of BookMyShow is very user-friendly, attractive and unique ie. different from its competitors. Apart from Hindi and English, it is also available in five regional languages. With BookMyShow, there is no need to visit multiple theaters to check ticket availability. Guests do not need to go to the room in advance to make a reservation. BookMyShow allows advance planning and booking. Customers do not have to wait in long queues, which is very expensive to queue, especially for the elderly.

BookMyShow works with other online ticketing platforms like Grapon.in and mobile wallets like Mobikwick, Paypal and Amazon Pay. Flexibility in payment options through UPI, mobile wallet, online banking and more. Which makes it a more popular ticketing platform. This makes price comparisons and understanding show times easier. Where earlier shows only started at 12:00pm, 9:00am, 3:00pm, 7:00pm and 9:00pm, now shows start at odd times like 2:10pm, 5:20pm, etc. until then. This makes the stage manager's job easier. This gives customers different options. Now they can easily choose their favorite seat.

How BookMyShow books ticket

When users use a mobile app or website to access BookMyShow platform, you can see the available and occupied seats for a movie currently playing in that theater. The question now is how do these BookMyShow communicate with theaters to get available seat information and present it to users? Obviously, the app needs to work with theater servers to get seat assignments and serve them to users.

So for that Each online ticket service provider will be assigned a certain number of time slots, which will then be made available to users. In this strategy, some seats are already reserved for these online ticket service providers , so there is no need to constantly update the seat information for all rooms. Another way is the app can work with theaters and other aggregators to continuously update seat availability information. The ticket will then be presented to the user. Usually BookMyShows apis are connected with the theater servers. So whenever a ticket is booked it will be updated in the theater server.Now there can be two or more users trying to book the same tickets. So for this problem, In Theater side server, BookMyShow is expected to block seats and temporarily assign seats to users until they book. If the reservation is not made within 10 minutes, the seat will be lost.

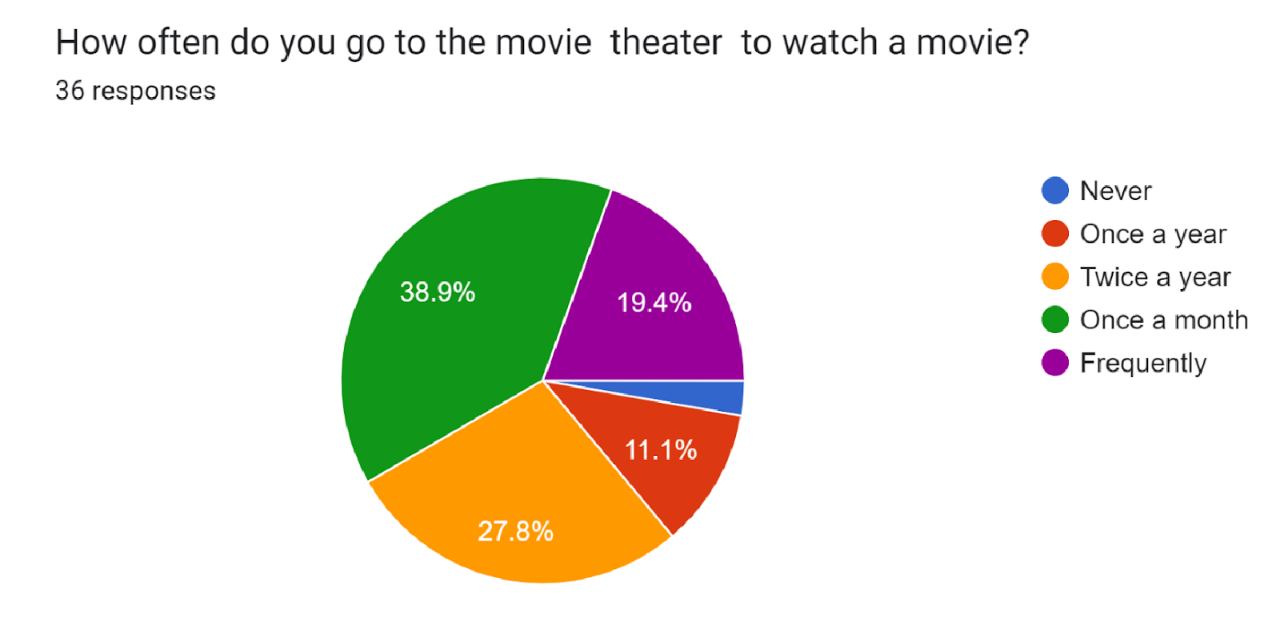
## 2.4 Requirements Collection Documentation

### 2.4.1 Questionnaire

Mode of Questionnaire: Online google form responses

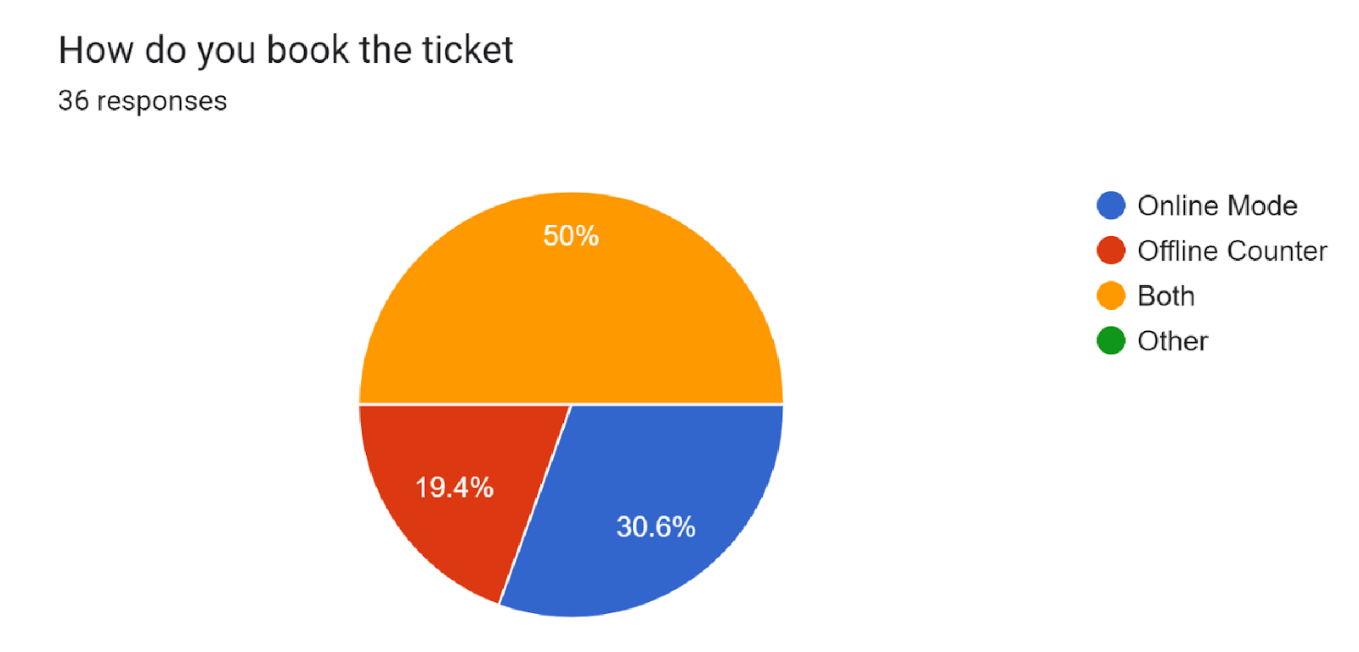
Number of Responses: 36 Responses

Agenda: To find the flaws in the online ticket booking system

**Data Analysis**

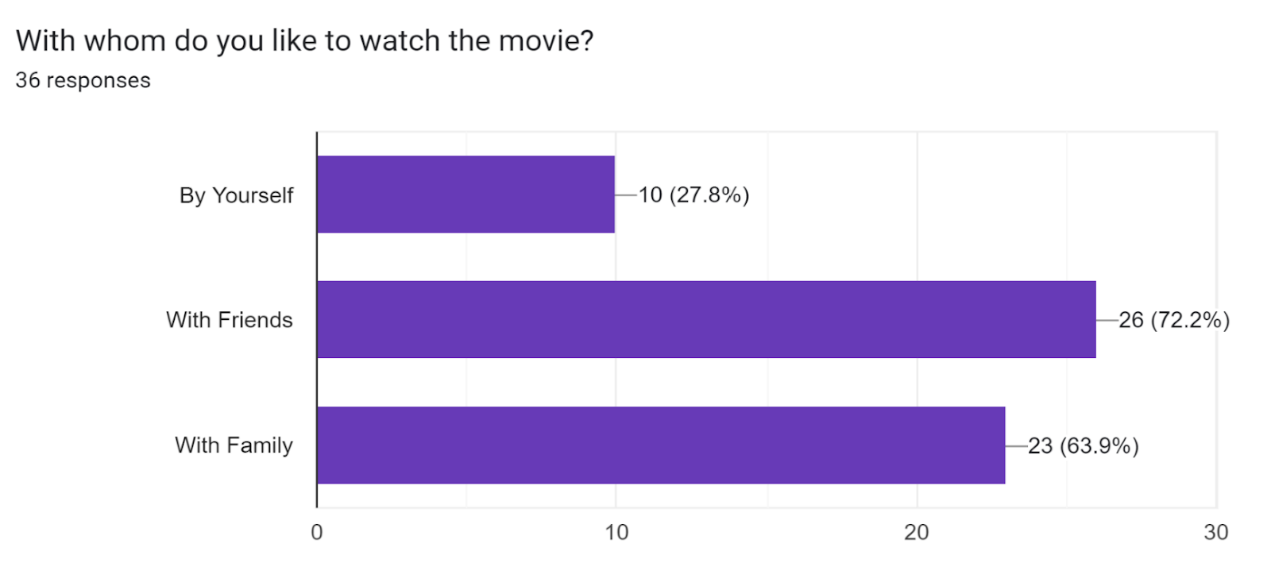
**Interpretation:**

From the analysis, it has been found that out of 36 respondents, the majority of 66.7% of the respondents goes to watch the movie once or twice a year.

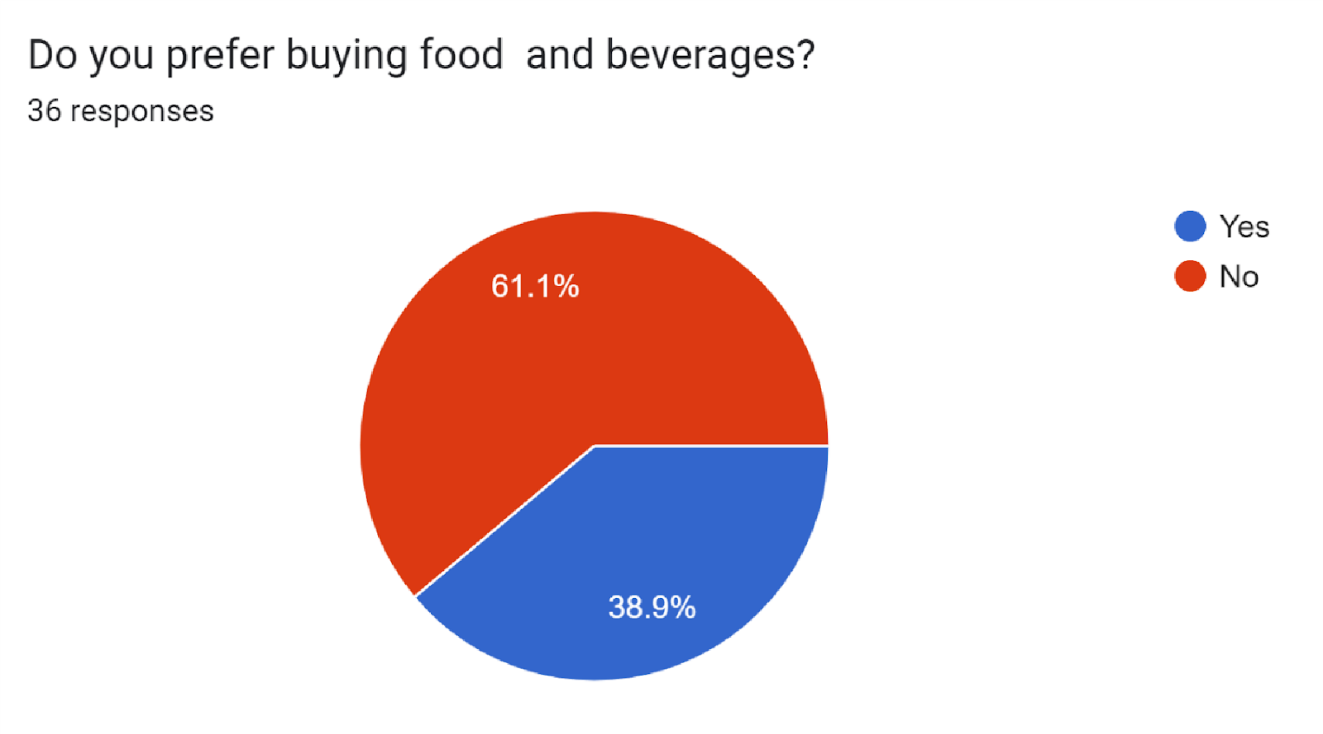


**Interpretation:**

From the analysis, it has been found that out of 36 respondents, 50% of the respondents book tickets both ways i.e Online Mode and offline Counter.

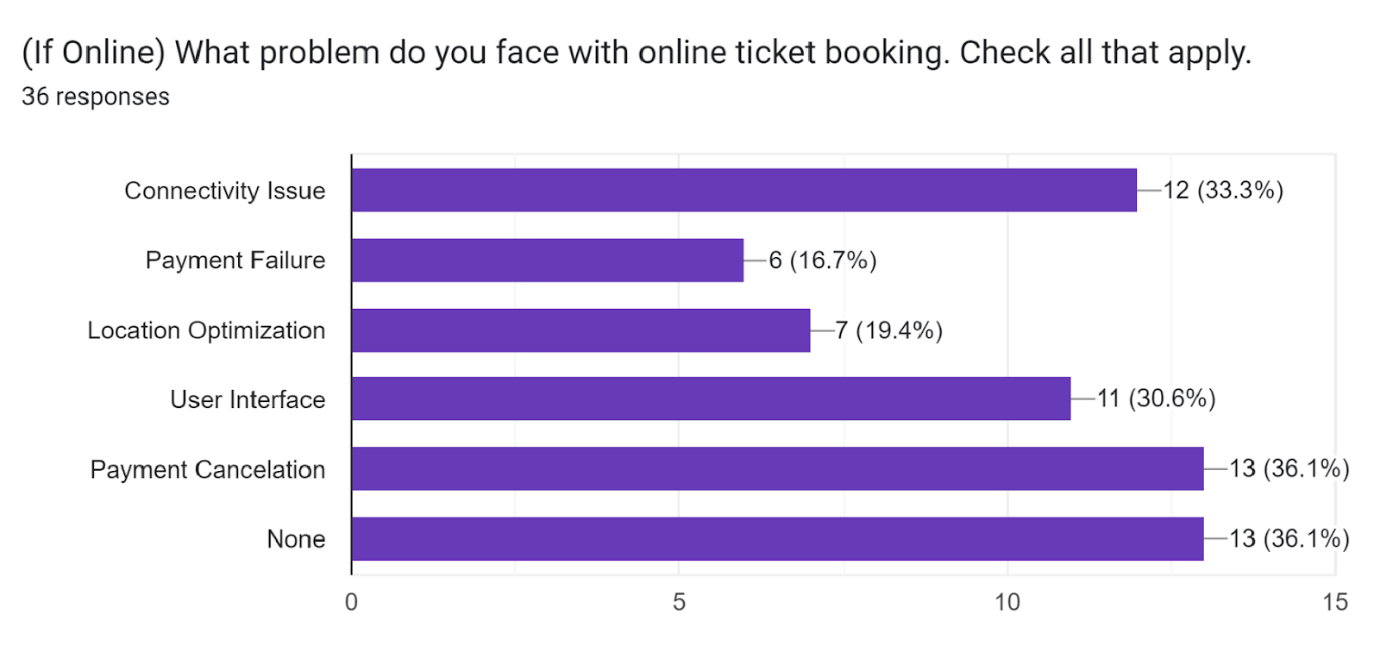


**Interpretation:** From the analysis, it has been found that out of 36 respondents, the majority of the respondents prefer watching movies with their friends and family members.



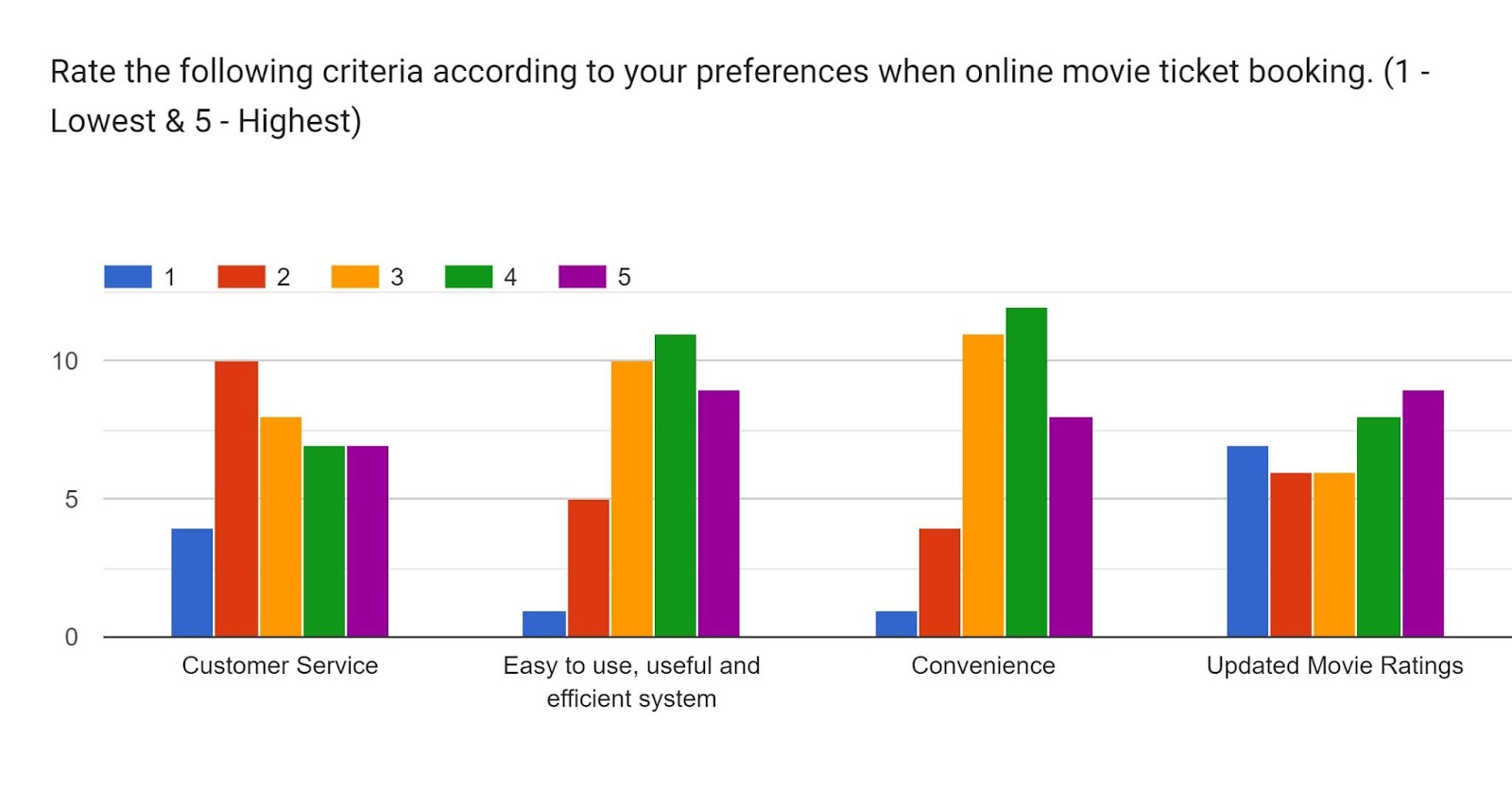
**Interpretation:**

From the analysis, it has been found that out of 36 respondents, 61.1% of the respondents prefer not to buy food and beverages from the theater.



**Interpretation:**

From the analysis, it has been found that out of 36 respondents, the majority of the people face connectivity issues and payment cancellation issues as a major problem while online ticket booking.



**Interpretation:**

From the analysis, it has been found that out of 36 respondents, the majority of the respondents did not find the online booking system as an easy to use and efficient system. It was also found that the customer likes the updated movie ratings.

**2.4.2 Interview**

**Movie Ticket Booking System: Interview Plan**

**Participants:** Laxit Shah

Himang  Patel

Jaani Patel(Manager, City Pulse,Gandhinagar)

**Date:** 29/8/2022    **Time:** 11:00 AM

**Duration:** 15 minutes **Place:** City Pulse,Kudasan,Gandhinagar

**Purpose of Interview:**

To discuss how the online movie ticket system works and what are the responses of people towards it.

**Agenda:**

* Ask if the theater uses an online reservation system?
* Discuss about the existence system?
* What type of customer data do they collect?
* Who are their existing partners?
* Problems faced by them to use such a system?
* Problems faced by customers to use such a system?

**Findings:**

Movie Theater uses their own system and they use the api of BookMyShow and Paytm and this website charges some nominal

fees for using their services. From the interview we found out that the payment gateways are not working sometimes and customers face some problems. Sometimes there are  server issues. Another problem we found out was Refund Policy ,sometimes when the customer books an online ticket it does not register in the offline theater system due to connectivity issues so sometimes the user doesn’t get confirmation about their ticket. Sometimes the customer doesn’t get a booking id even though they paid the ticket.

## 2.5 Requirement in the database

### 2.5.1 Functional Requirement

* Login Page - Customers should be able to login to the portal.
* Account details - Users can see their account details.
* Setting - to change the user details.
* Display movies - display all the movies available in their area.
* Display timing of the movie.
* Filter - filter based on location, cinema type, language and much more.
* Seat details - system should allow users to select the desired seat they want.
* Ticket details - Get detailed information of ticket booking.
* Automatically calculate the ticket cost.
* Payment details - Get all the transaction history of the user.
* Send a copy booking details to the user.
* Show all payment history.

### 2.5.2 Non-Functional Requirement

* Reliability - It determines the probability that a system or its components will function without failure for a given period of time under predetermined conditions.
* Maintenance - A well maintained system helps the time required for a solution or its component to be fixed, changed to increase performance or other qualities, or adapted to a changing environment.
* User Friendly - System should be easy to operate by the users.
* Safety Requirements- Seat selection and payment for movie reservation systems must be secure enough to prevent data being hacked.Also the data should be backed up in case of any data loss.

**Observation**

From the readings and the research we found out that there are still some flaws in the existing system. Customers face issues like refunds for canceled tickets and vouchers, payment gateways, protection of user data, non-response to complaints and failed transactions. Therefore, the refund policy should be more user-friendly and fair. Sometimes, when a user makes an online booking it does not get updated in the movie theater system. Also, the majority of the users of online booking platforms are youngsters. People of 30+ ages find it difficult to operate the online booking system. They prefer an offline booking system rather than an online booking system. So the GUI must be simple to use for all types of age groups. The main purpose of the system is to provide users with comprehensive information about movies, users can order both tickets and snacks, and the ability to refund money gives more flexibility to the system.

## 2.6 Business Constraints

* Reliable internet access is required to confirm and add reservations made over the  phone.
* Managing cancellations for online reservations has always been a challenge.
* The Main Challenge of Online Booking System is Handling Customer Complaints. Customers may have bad experience with the system.We need to have the option on the booking system to handle such issues raised.
* Data should not be corrupted incase of system crash or any kind of failure.

**Fact Finding Chart**

|  |  |  |
| --- | --- | --- |
| Objective | Technique | Subjects |
| To get the background of BookMyShow company  and its blueprint | Background  Reading | Company's official website |
| To understand how offline cinemas get the Movie details. | Interview | Manager of City pulse  Theater |
| To identify the main business operation | Interview | Manager of City pulse  Theater |
| To understand the main connection between offline theater servers and Online booking platform | Interview | Manager of City pulse  Theater |
| To know the common issues faced by the users while online ticket booking. | Questionnaire | Users |
|  |  |  |
| To Understand the business model of BookMyShow. | Background  Reading | Research Paper |
| To understand the flaws in existing system | Background  Reading,  Observation | Research Paper |
| To know the role of staff and admin in the theater | Interview | Manager of City pulse  Theater |
| To know what  data are stored in the database of the the theater | Interview | Manager of City pulse  Theater |

**2.7 Assumptions**

* The application is based on internet connected users, where they can  search  movie information.
* While using the program, it is assumed that there is enough  disk space to save the file.
* The full operation of the online movie ticket booking depends on the availability of an Internet connection.
* It is also assumed that the user will make payment while booking the ticket.So the payment gateway is a must in this system.
* The user is assumed to be fully familiar with the system, which means the user is not a naive user. All data entered is correct.
* There are limits on booking tickets. If the theater is full, the user cannot book the movie at that  time.

**2.8 User Categories**

**1>Customer**

* Customers register their details on the platform.
* Customers can check movie on the platforms which are their on the platforms also they can see upcoming movies on the platform
* Customers can book tickets ,they can select multiple seats and also they can buy snacks and beverages.

**2>Employee**

* Employees of the particular cinema hall must have to register themselves.
* Employees of a particular cinema hall manages hows,movies,seats,halls,cinema,stalls and menus.
* Employees of a particular cinema will be able to update the movie which is shown by them .They also will be able to manage show timing and date.They can add,update and delete the movie record of each cinema.

**3>Admin**

* Admin will Manage all the Records.
* Admin shall control the whole platform.
* Admin will be able to create ,update and delete the record.
* Admin has access to the Customer record who are registered on the platform.

**2.9 Privileges**

**1>Customer**

* Based on the location the customer will be able to view a list of movies running near him/her.
* Customers will be able to select the movie and cinema of his/her choice.
* Customers will be able to book a seat if the seat is empty and can select multiple seats .
* A Booking confirmation will be sent via an email or text message to verify that tickets have been booked.

**2>Employee**

* Employees of a particular cinema hall will be able to get all the info about Shows Information.
* They will also be able to get info about the customer who has booked the tickets in their cinema including the seats booked.
* They will be able to modify the records of movies,halls,seats,stalls and menus.

**3>Admin**

* Admin will manage all the records.
* Admin will be able to get all the information about all the platform records.
* Admin shall be able to create,update and delete all the records.

**Conclusion**

This system offers an online booking system that allows customers to save time when booking tickets.By using this system, the company can provide reservation services and information to their customers without the limitation of office hours or manpower.It allows customers to book a ticket from anywhere with an internet connection, it is also designed for companies to manage their internal business processes; Reducing human errors and overcoming difficulties and problems that arose in previous systems.Through Online Booking System records are maintained and database is updated with time to time.The system is designed with a modern system architecture to meet changing requirements.

**3 ER Diagram**

## 3.1 Noun Analysis

|  |  |  |
| --- | --- | --- |
| **Nouns** |  |  |
| consumers | movie | language |
| ticket | theater | reservation |
| movie | cancel\_ticket | user |
| merchants | seats | movie |
| show | user | partner |
| movie | seat | payment |
| start time | Database management system | cinema |
| duration | seat | venue |
| screen | user | customer |
| users | current user | times |
| city | service provider | theaters |
| admin | login | user |
| counter | seat | state |
| book tickets | period | payment |
| cinema hall | user | date |
| theater | seat | time |
| guest | system | register |
| portal | seat | server |
| language | minutes | payment |
| sites | seat | payment |
| software | user | order |
| cinemas | ticket | receipt |
| films | user | theater server |
| cinemas | payment | log\_out |
| software | beverages | business |
| users | option | genre |
| data | payment | movie\_format |
| database | receipt | rating |
| system | gateway | role |
| ticket | application |  |

## 3.2 Accepted Nouns:

|  |  |  |
| --- | --- | --- |
| **Candidate Entity**  **Set** | **Candidate Attribute** | **Candicade relationship** |
| Halls | Hall\_ID, Cinema\_ID, Screen\_ID | Shows details of cinema hall |

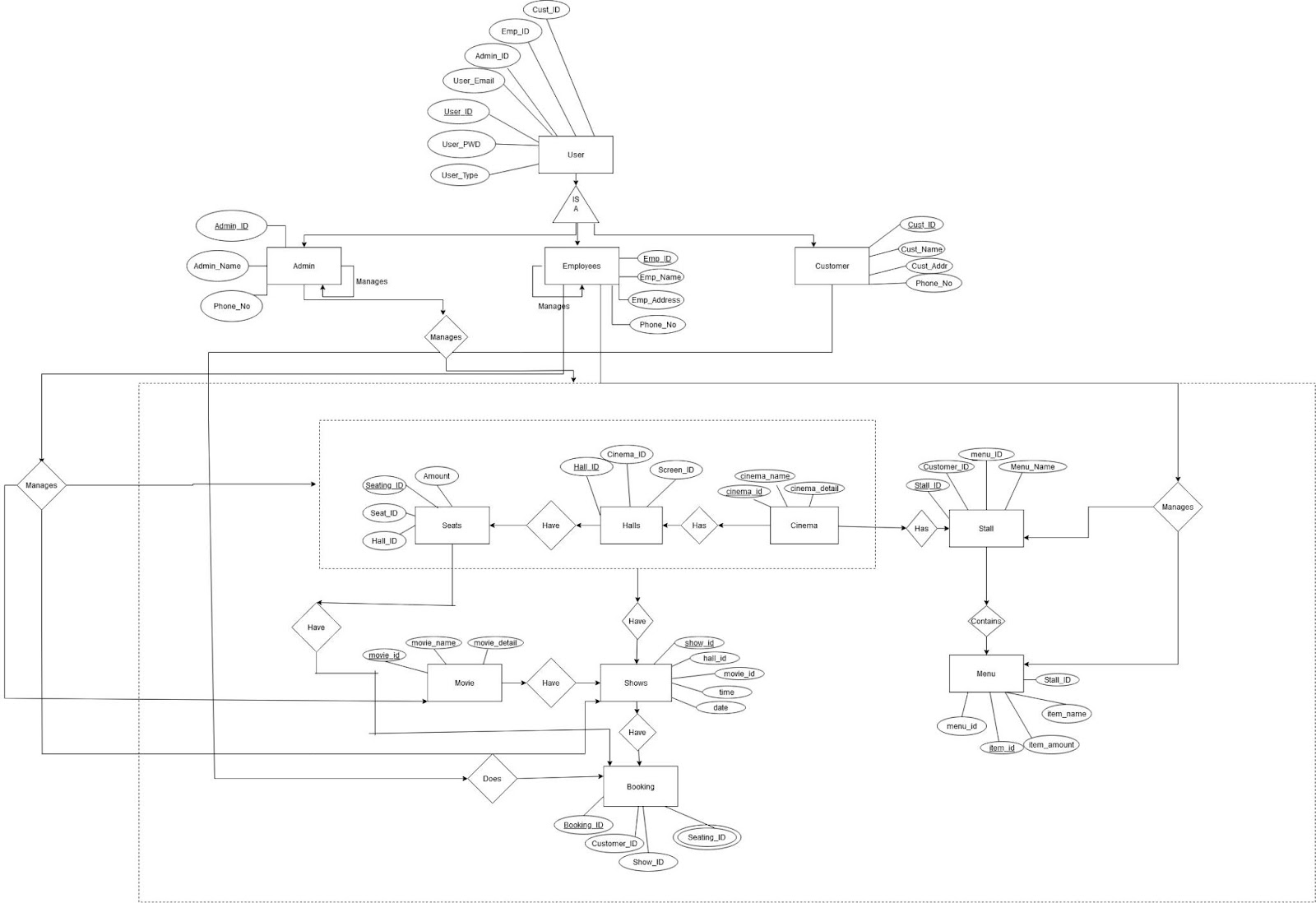
|  |  |  |
| --- | --- | --- |
| Stall | Stall\_ID, User\_ID, Menu\_ID,  Menu\_Name | View and Order From Stalls |
| Menu | Menu\_ID, Item\_ID, Item\_Amount,  Item\_Name | View and Order Beverages |
| Shows | Show\_ID, Hall\_ID, Movie\_ID, Time | Display available shows on a particular day |
| Movie | Movie\_Name, Movie\_ID,  Movie\_Detail | Display available Movie and its detail |
| Booking | Booking\_ID, User\_ID, Show\_ID,  Seating\_ID | Book tickets and Beverages |
| User | User\_Name, User\_ID, User\_PWD,  User\_Address, User\_Phone\_No. ,  User\_Email |  |
| Cinema | Cinema\_ID, Cinema\_Name,  Cinema\_Detail | View details of available  Cinemas |
| Seats | Seating\_ID, Amount, Seat\_ID,  Hall\_ID | View available seats |
| Admin | Admin\_ID, Admin\_Name, Admin\_Phone\_No. | Manages Entire System |
| Customer | Customer\_ID, Customer\_Name,  Customer\_Address,  Customer\_Phone\_No. | Booking ticket from the system |
| Employee | Emp\_ID,Emp\_Name,  Emp\_Phone\_No,Emp\_Address | Manages the cinema |

## 3.3 Rejected Nouns

|  |  |
| --- | --- |
|  | **Reason** |
| merchants | Irrelevant |
| software | Irrelevant |

|  |  |
| --- | --- |
| seat | Duplicate |
| data | Irrelevant |
| Database management system | Irrelevant |
| current user | Irrelevant |
| service provider | Irrelevant |
| period | Irrelevant |
| user | Duplicate |
| system | Irrelevant |
| minutes | Irrelevant |
| ticket | Duplicate |
| payment | Duplicate |
| option | Duplicate |
| receipt | Duplicate |
| application | Irrelevant |
| language | Duplicate |
| partner | Irrelevant |
| cinema | Duplicate |
| login | Duplicate |
| state | Irrelevant |
| time | Duplicate |
| server | Irrelevant |
| order | Irrelevant |
| Staff | Irrelevant |

**3.4 Entity Relation Diagram**



**Note:If the image is not visible properly, please refer to this links**

1>[Link1](https://drive.google.com/file/d/1Akffiglacuz-UCIi9TNCWkmRqbxXacHf/view?usp=sharing)

**3.5 ER-Diagram Explanation**

* Entity User has three children - employee, customer, and admin.
* Admin manages the entire system.
* The employee manages their cinema center which includes movies, shows, halls, seats, cinema stall, and menu.
* Customers can book tickets for a movie.
* The movie have shows. Seats, halls, and Cinema are managed under shows.
* Every cinema has one or more halls and each hall have seats.
* Cinema has stalls and stalls containing a menu that displays all the foods and beverages available at that cinema.

**4. ER to relational mapping.**

|  |  |  |  |
| --- | --- | --- | --- |
| User | | | |
| **Field\_Name** | **Data\_Type** | **Description** | **Constrain** |
| Customer\_Id | int(50) | fetch customer id | Foreign Key |
| Emp\_id | int(50) | fetch employee id | Foreign Key |
| Admin\_Id | int(50) | fetch admin id | Foreign Key |
| User\_Email | varchar(50) | stores user email | NOT NULL |
| User\_Id | int(50) | stores user id | primary key |
| User\_Passwrod | varchar(50) | stores user password | NOT NULL |
| User\_Type | varchar(50) | stores user type | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| Admin | | | |
| **Field\_Name** | **Data\_Type** | **Description** | **Constrain** |
| Admin\_ID | int(50) | stores admin Id | Primary Key |
| Admin\_Name | varchar(50) | stores admin Name | NOT NULL |
| Phone\_No | int(10) | stores admin phone no. | Unique key |

|  |  |  |  |
| --- | --- | --- | --- |
| Employee | | | |
| **Field\_Name** | **Data\_Type** | **Description** | **Constrain** |
| Employee\_ID | int(50) | stores employee id | Primary Kaey |
| Employee\_Name | varchar(50) | stores employee name | NOT NULL |
| Employee\_Address | varchar(100) | stores employee address | NOT NULL |
| Phone\_No | int(10) | stores employee pone no. | Unique Key |

|  |  |  |  |
| --- | --- | --- | --- |
| Customer | | | |
| **Field\_Name** | **Data\_Type** | **Description** | **Constrain** |
| Customer\_ID | int(50) | stores customer id | Primary Key |
| Customer\_Name | varchar(50) | stores customer name | NOT NULL |
| Customer\_Address | varchar(100) | stores customer address | NOT NULL |
| Phone\_No | int(10) | stores customer phone no | Unique Key |

|  |  |  |  |
| --- | --- | --- | --- |
| Booking | | | |
| **Field\_Name** | **Data\_Type** | **Description** | **Constrain** |
| Booking\_Id | int(50) | stores booking id | Primary Key |
| Customer\_Id | int(50) | fetch customer id | Foreign Key |
| Show\_Id | int(50) | fetch show id | Foreign Key |
| Seating\_Id | int(50) | fetch seating id | Foreign Key |

|  |  |  |  |
| --- | --- | --- | --- |
| Seats | | | |
| **Field Name** | **Data Type** | **Description** | **Constraint** |
| Seating\_ID | int(50) | stores seating id | Primary Key |
| Seat\_ID | int(50) | fetch the seat id | Unique Key |
| Hall\_ID | int(50) | fetch the hall id | Foreign Key |
| Amount | int(50) | stores amount | Not NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| Halls | | | |
| **Field Name** | **Data Type** | **Description** | **Constraint** |
| Hall\_ID | int(50) | stores hall id | primary key |
| Cinema\_ID | int(50) | fetch the cinema id | foreign key |
| Show\_ID | int(50) | fetch the show id | foreign key |

|  |  |  |  |
| --- | --- | --- | --- |
| Cinema | | | |
| **Field Name** | **Data Type** | **Description** | **Constraint** |
| cinema\_id | int(50) | stores cinema id | primary key |
| cinema\_name | varchar(50) | stores cinema name | not null |
| cinema\_detail | varchar(50) | stores cinema details | not null |

|  |  |  |  |
| --- | --- | --- | --- |
| Stall | | | |
| **Field Name** | **Data Type** | **Description** | **Constraint** |
| stall\_id | int(50) | stores stall id | primary key |
| customer\_id | int(50) | fetch the cinema id | foreign key |
| menu\_id | int(50) | stores menu id | unique key |
| menu\_name | varchar(50) | stores menu name | not null |

|  |  |  |  |
| --- | --- | --- | --- |
| Menu | | | |
| **Field Name** | **Data Type** | **Description** | **Constraint** |
| stall\_id | int(50) | fetch the stall id | foreign key |
| item\_name | varchar(50) | stores item name | stores item name |
| item\_amount | int(50) | stores item amount | not null |
| item\_id | int(50) | stores item id | primary key |

|  |  |  |  |
| --- | --- | --- | --- |
| Movie | | | |
| **Field Name** | **Data Type** | **Description** | **Constraint** |
| movie\_id | int(50) | stores movie id | primary key |
| movie\_name | varchar(50) | stores movie name | not null |
| movie\_detail | varchar(50) | stores movie details | not null |

|  |  |  |  |
| --- | --- | --- | --- |
| Shows | | | |
| **Field Name** | **Data Type** | **Description** | **Constraint** |
| show\_id | int(50) | stores show id | primary key |
| hall\_id | int(50) | fetch the hall id | foreign key |
| movie\_id | int(50) | fetch the movie id | foreign key |
| time | int(50) | stores show time | not null |
| date | date | stores show date | not null |

**5. List of relations with attributes and constraints**

|  |  |
| --- | --- |
| **Entity** | **Attributes** |
| User | User\_Id(PK), Customer\_Id(FK), Emp\_id(FK), Admin\_Id(FK), User\_Email, User\_Passwrod, User\_Type |
| Admin | Admin\_ID(PK), Admin\_Name, Phone\_No |
| Employee | Employee\_ID(PK), Employee\_Name, Employee\_Address, Phone\_No |
| Customer | Customer\_ID(PK), Customer\_Name, Customer\_Address, Phone\_No |
| Booking | Booking\_Id(PK), Customer\_Id(FK), Show\_Id(FK), Seating\_Id(FK) |
| Seats | Seating\_ID(PK), Seat\_ID, Hall\_ID(FK), Amount |
| Halls | Hall\_ID(PK), Cinema\_ID(FK), Show\_ID(FK) |
| Cinema | cinema\_id(PK), cinema\_name, cinema\_detail |
| Stalls | stall\_id(PK), customer\_id(FK), menu\_id, menu\_name |
| Menu | item\_id(PK), stall\_id(FK), item\_name, item\_amount |
| Movies | movie\_id(PK), movie\_name, movie\_detail |
| Shows | show\_id(PK), hall\_id(FK), movie\_id(FK), time, date |

**6. ER to DDL**

Create TABLE Users (

User\_ID int NOT NULL Primary key,

Customer\_ID int,

Admin\_ID int,

Employee\_ID int,

User\_Email varchar(50) NOT NULL,

User\_Password varchar(50) NOT NULL,

User\_Type varchar(50) NOT NULL,

FOREIGN KEY (Customer\_ID)

REFERENCES Customer(Customer\_ID),

FOREIGN KEY (Admin\_ID)

REFERENCES Admin(Admin\_ID),

FOREIGN KEY (Employee\_ID)

REFERENCES Employee(Employee\_ID)

);

======================================================

Create TABLE Admin (

Admin\_ID int NOT NULL Primary KEY,

Admin\_Name varchar(50) Not NULL,

Phone\_No int NOT NULL

);

======================================================

Create TABLE Employee (

Employee\_ID int NOT NULL Primary KEY,

Employee\_Name varchar(50) Not NULL,

Employee\_Address varchar(50) Not NULL,

Phone\_No int NOT NULL

);

======================================================

Create TABLE Customer (

Customer\_ID int NOT NULL Primary KEY,

Customer\_Name varchar(50) Not NULL,

Customer\_Address varchar(50) Not NULL,

Phone\_No int NOT NULL

);

======================================================

Create TABLE Stalls (

Stall\_ID int NOT NULL Primary KEY,

Customer\_ID int ,

Menu\_ID int,

Menu\_Name varchar(50) Not NULL,

FOREIGN KEY (Customer\_ID)

REFERENCES Customer(Customer\_ID)

);

======================================================

Create TABLE Menu (

Item\_ID int NOT NULL Primary KEY,

Item\_Amount int ,

Stall\_ID int,

Item\_Name varchar(50) Not NULL,

FOREIGN KEY (Stall\_ID)

REFERENCES Stalls(Stall\_ID)

);

======================================================

CREATE TABLE Booking(

Booking\_Id int NOT NULL Primary key,

Customer\_Id int,

Show\_Id int,

Seating\_Id int,

FOREIGN KEY(customer\_id)

REFERENCES customer(customer\_id),

FOREIGN KEY(Show\_id)

REFERENCES shows(Show\_id),

FOREIGN KEY(Seating\_id)

REFERENCES Seats(Seating\_id)

);

======================================================

CREATE TABLE Seats(

Seating\_Id int NOT NULL Primary Key,

Seat\_Id int UNIQUE,

Hall\_Id int,

Amount int NOT NULL,

FOREIGN KEY(Hall\_id)

REFERENCES Halls(Hall\_id)

);

CREATE TABLE Halls(

Hall\_Id int NOT NULL Primary Key,

Cinema\_Id int,

Show\_Id int,

FOREIGN KEY(Cinema\_id)

REFERENCES Cinema(Cinema\_id)

);

======================================================

Create TABLE Cinema (

Cinema\_ID int NOT NULL Primary key,

Cinema\_Name varchar(50) NOT NULL,

Cinema\_Details varchar(50) NOT NULL

);

======================================================

Create TABLE Shows (

Show\_ID int NOT NULL Primary key,

Hall\_ID int ,

Movie\_ID int ,

Date\_ date,

Time int,

FOREIGN KEY (Hall\_ID)

REFERENCES Halls(Hall\_ID),

FOREIGN KEY (Movie\_ID)

REFERENCES Movie(Movie\_ID)

);

======================================================

Create TABLE Movie (

Movie\_ID int NOT NULL Primary key,

Movie\_Name varchar(50) NOT NULL,

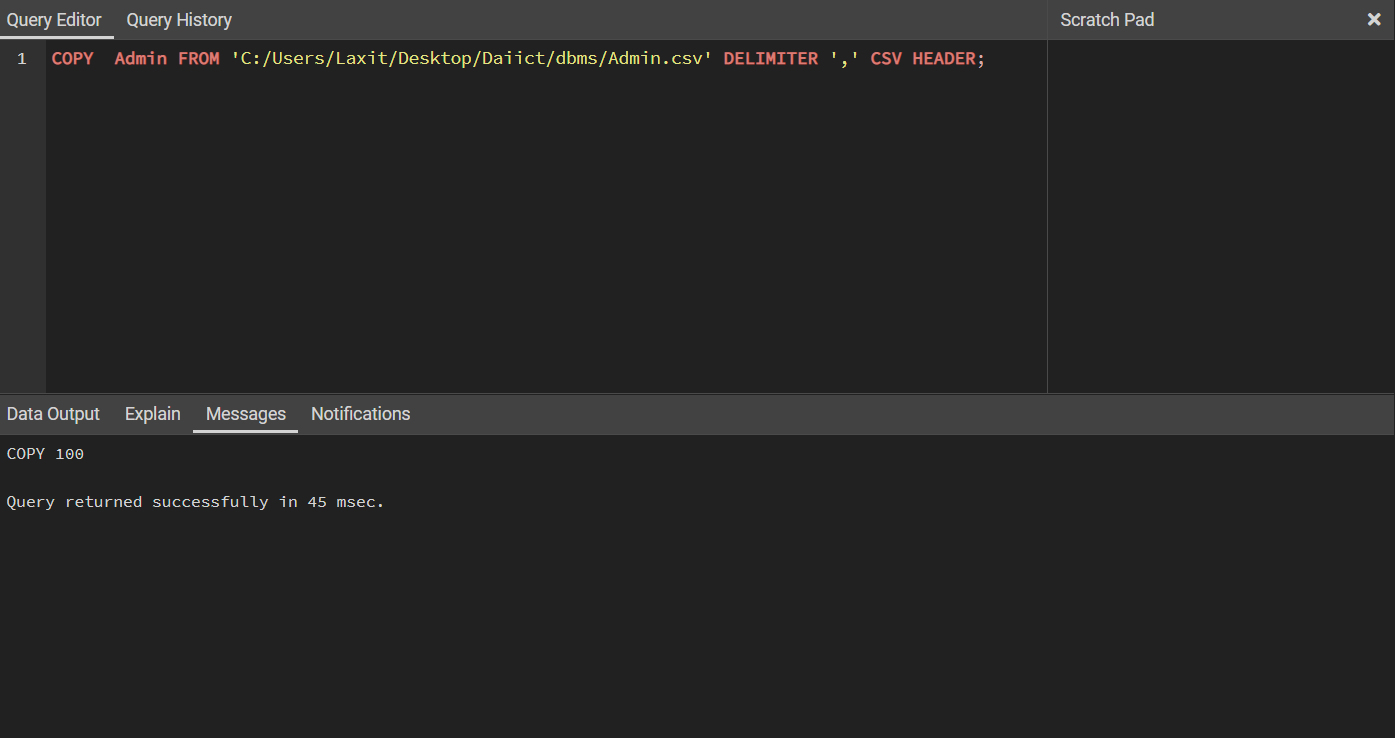
Movie\_Details varchar(50) NOT NULL

);

**7. Populating the data in tables**

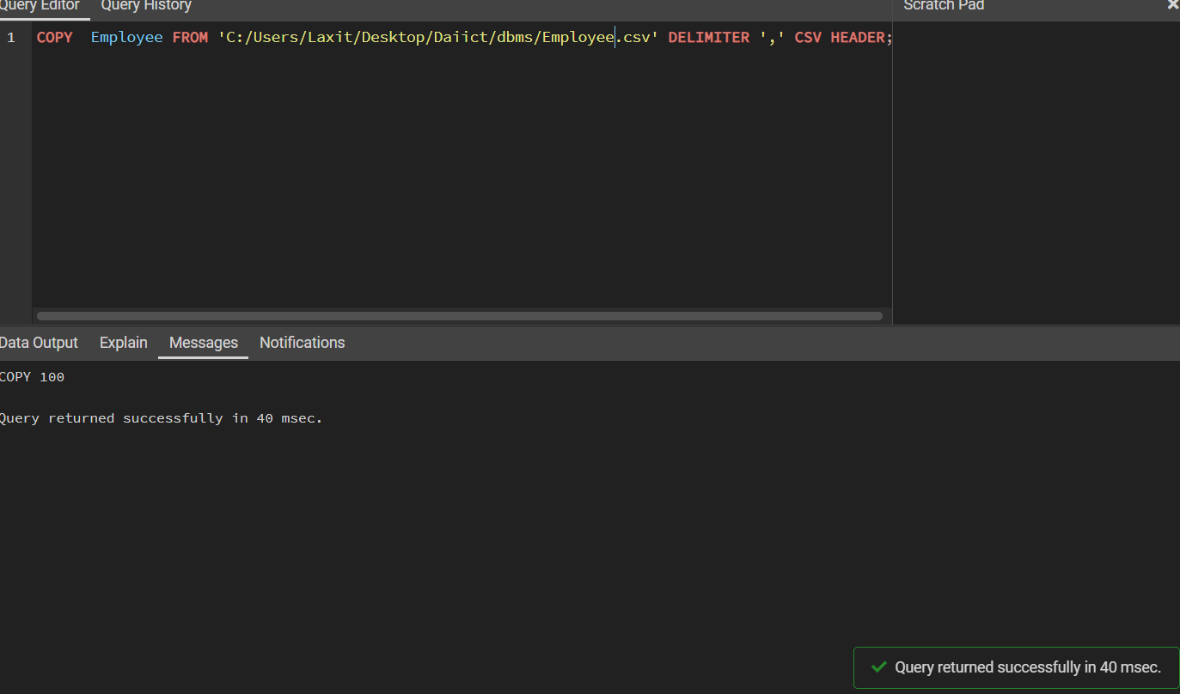
**1>Admin**

COPY  Admin FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/Admin.csv' DELIMITER ',' CSV HEADER;



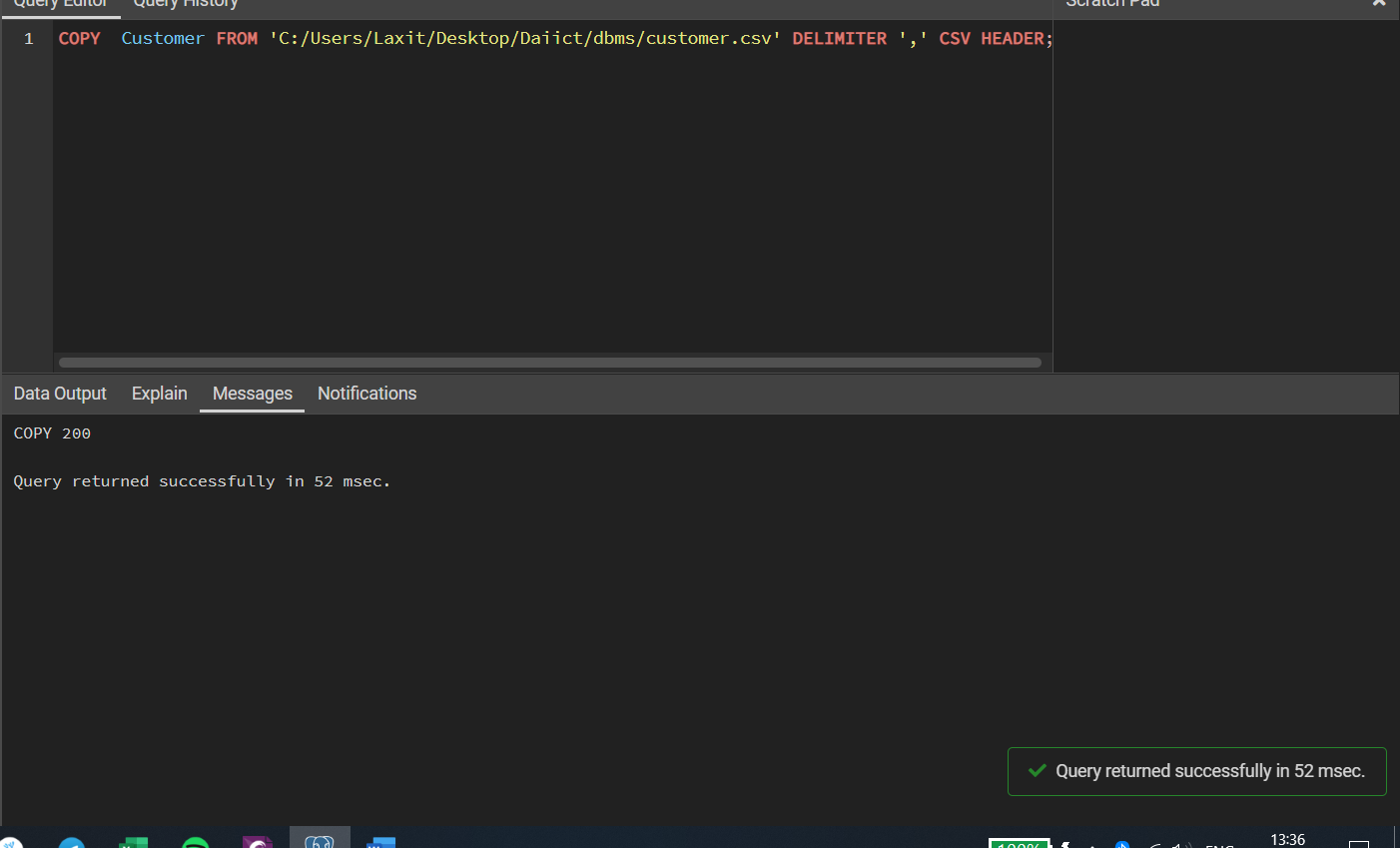
**2>Employee**

COPY  Employee FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/Employee.csv' DELIMITER ',' CSV HEADER;



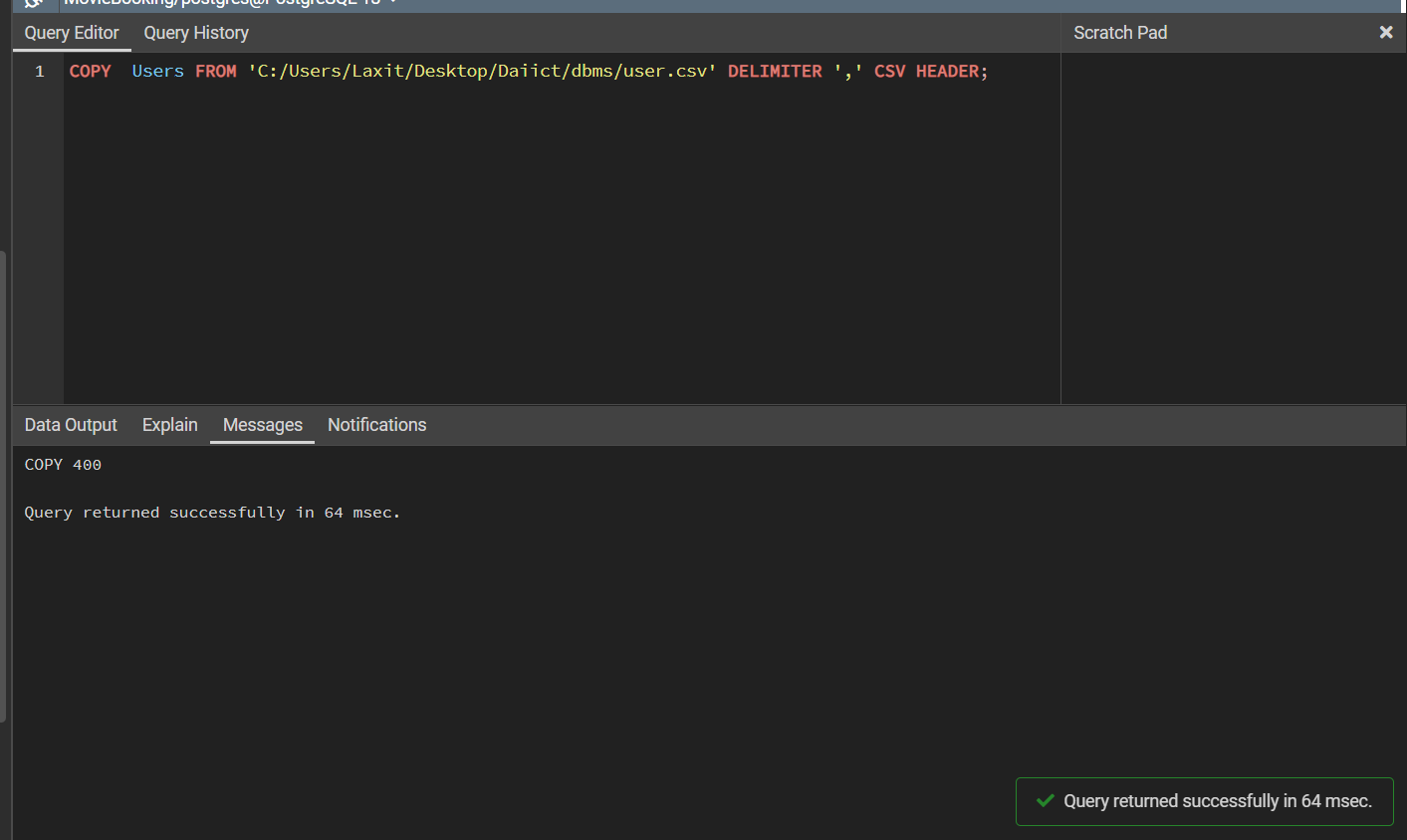
**3>Customer**

COPY  Customer FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/customer.csv' DELIMITER ',' CSV HEADER;



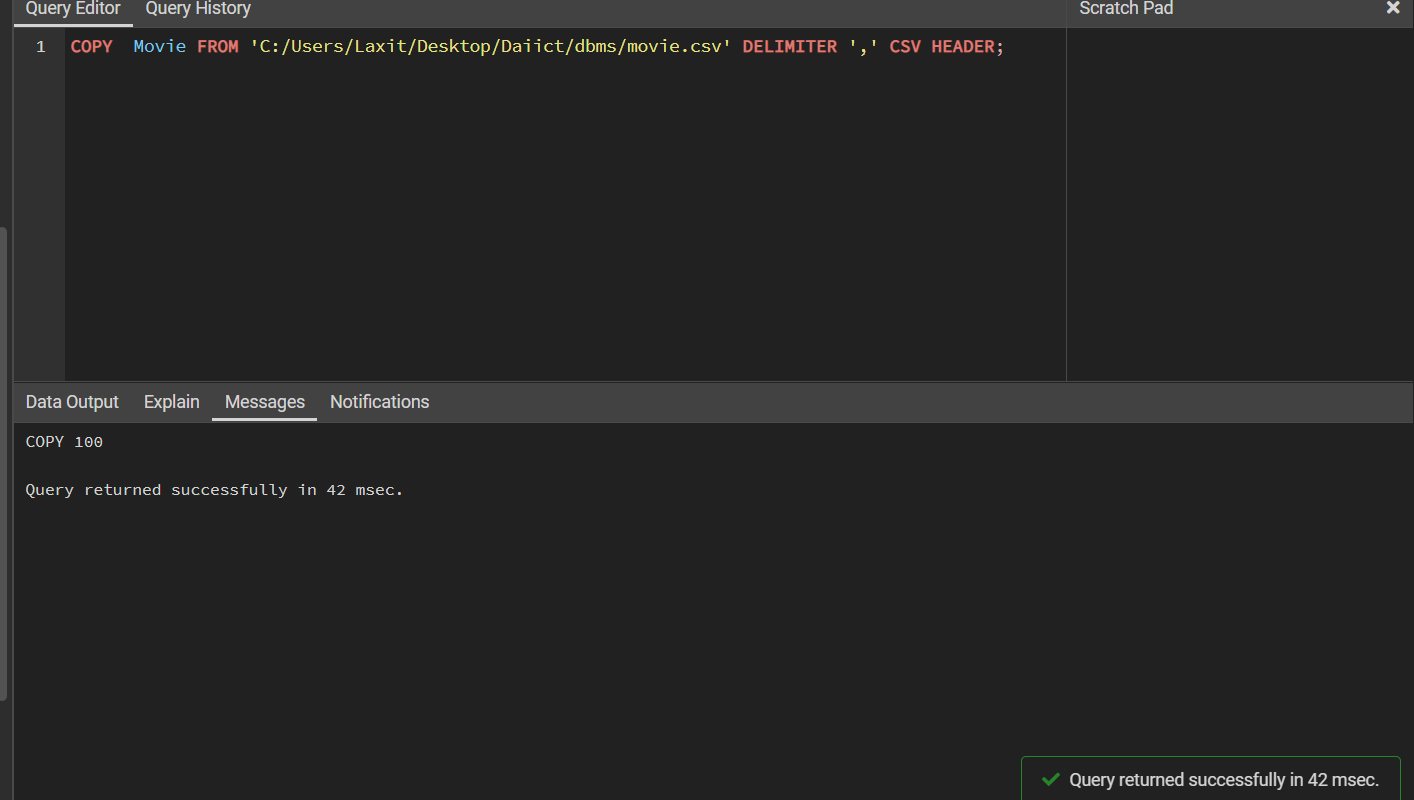
**4>User**

COPY  Users FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/user.csv' DELIMITER ',' CSV HEADER;



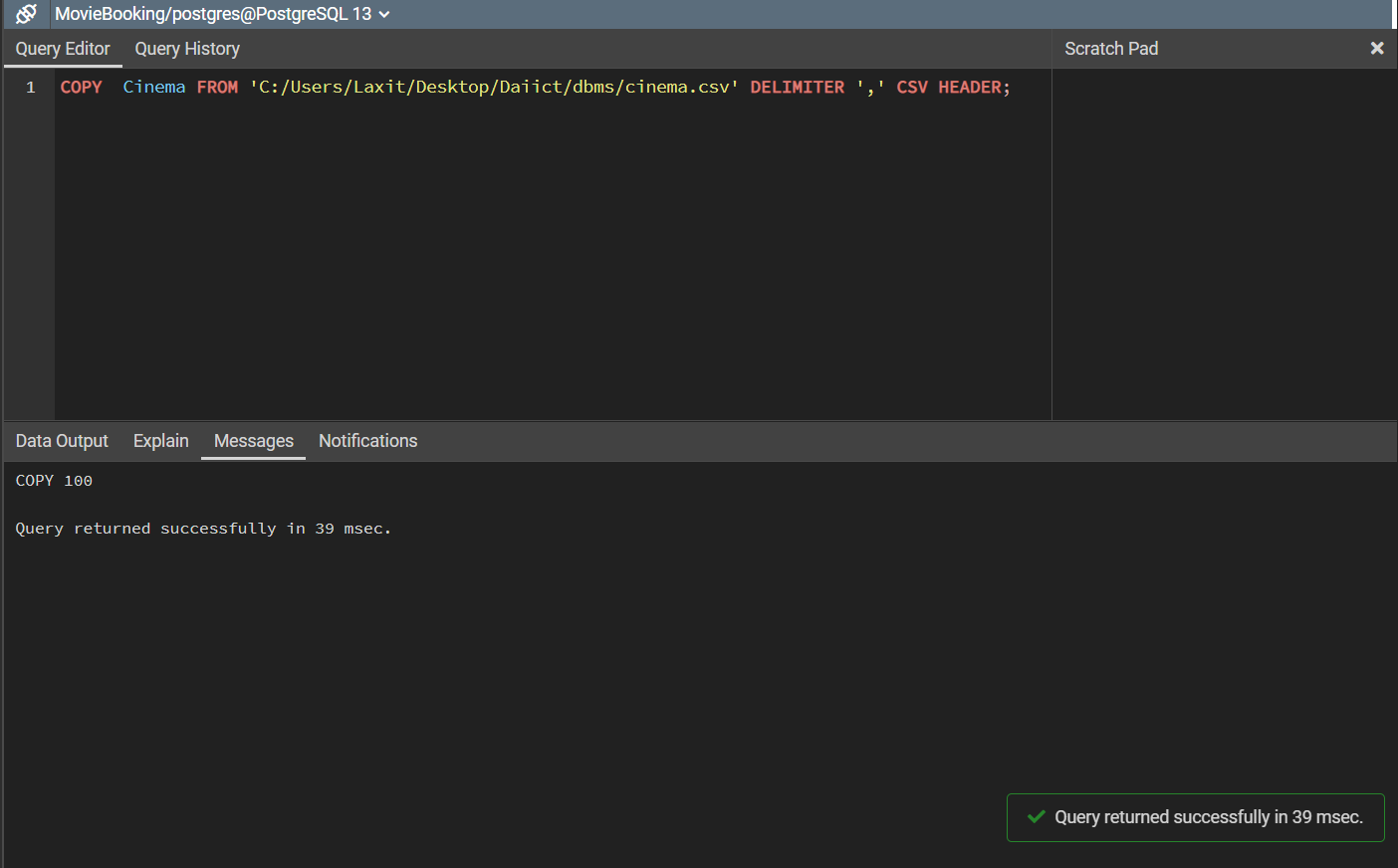
**5>Movie**

COPY  Movie FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/movie.csv' DELIMITER ',' CSV HEADER;



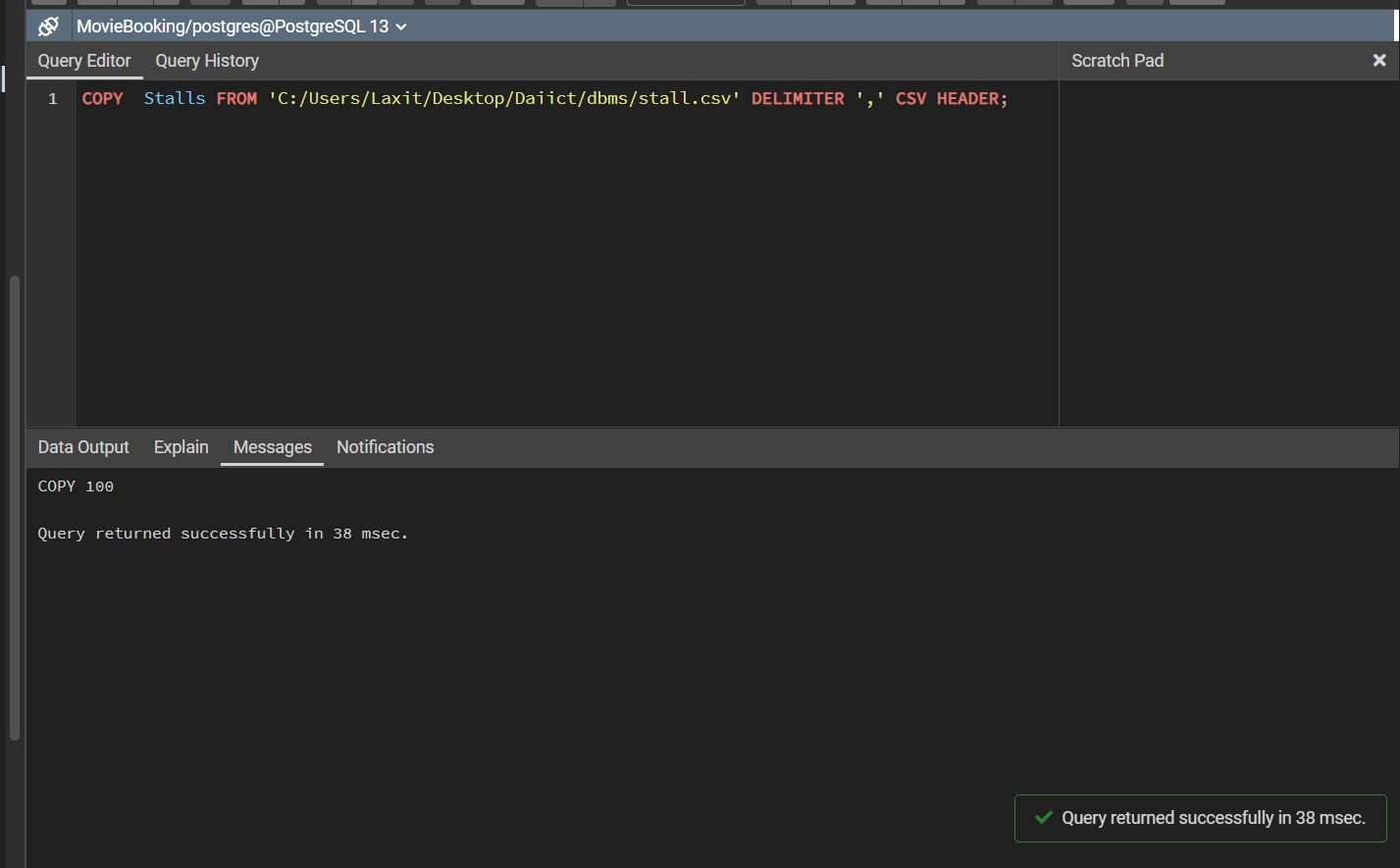
**6>Cinema**

COPY  Cinema FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/cinema.csv' DELIMITER ',' CSV HEADER;



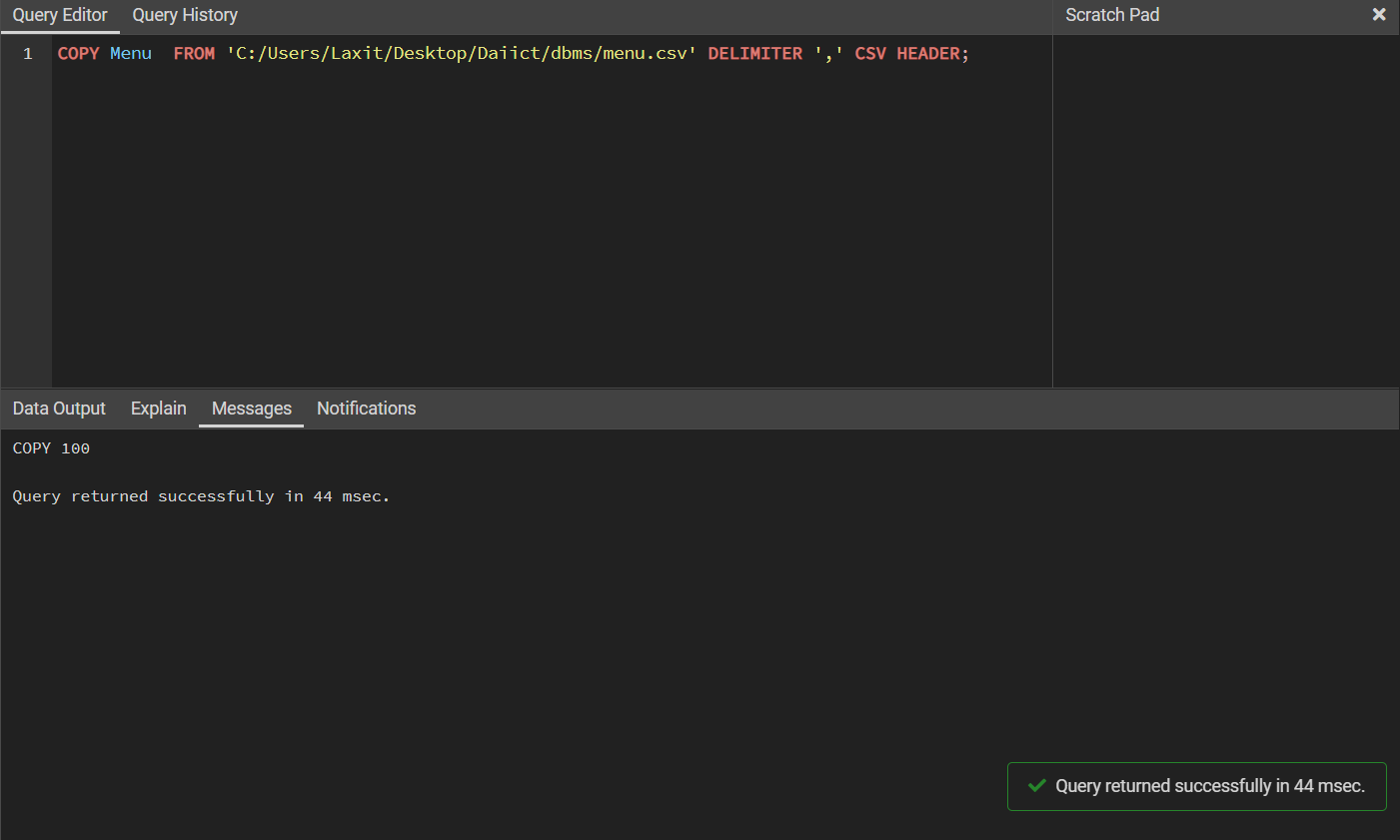
**7>Stalls**

COPY  Stalls FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/stall.csv' DELIMITER ',' CSV HEADER;



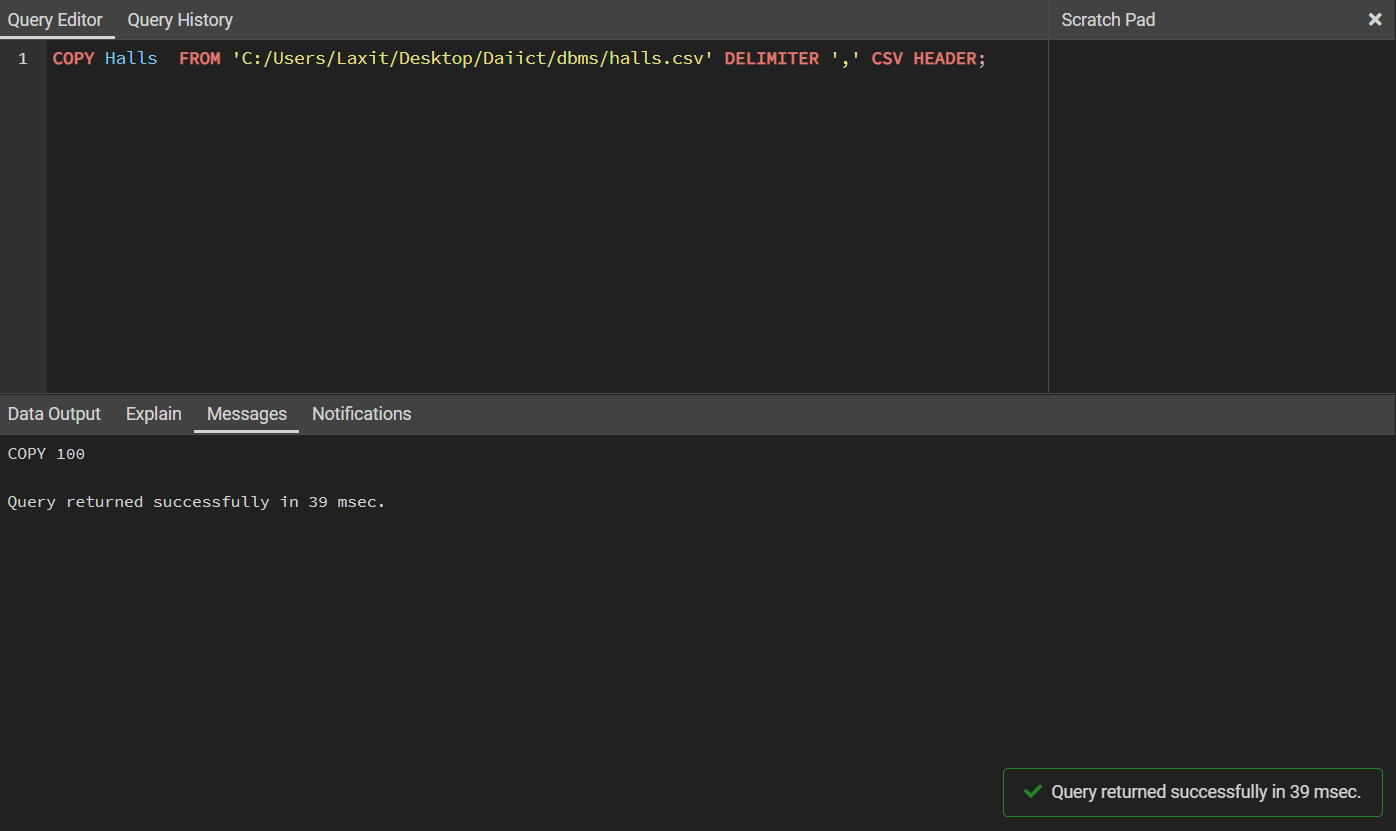
**8>Menu**

COPY Menu  FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/menu.csv' DELIMITER ',' CSV HEADER;



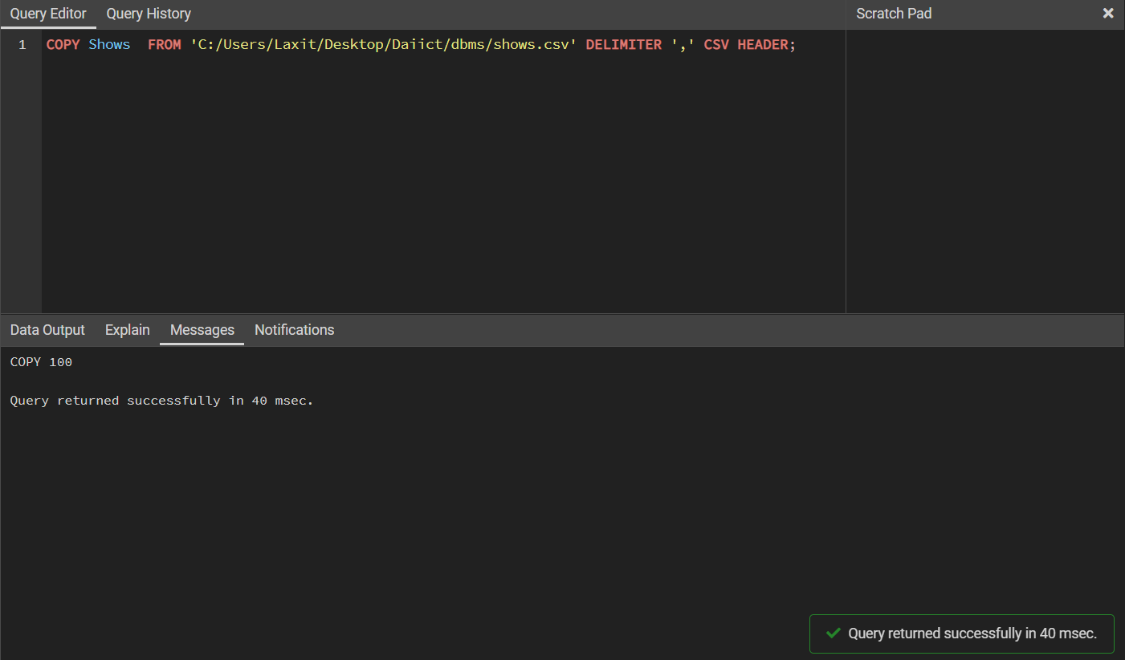
**9>Halls**

COPY Halls  FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/halls.csv' DELIMITER ',' CSV HEADER;



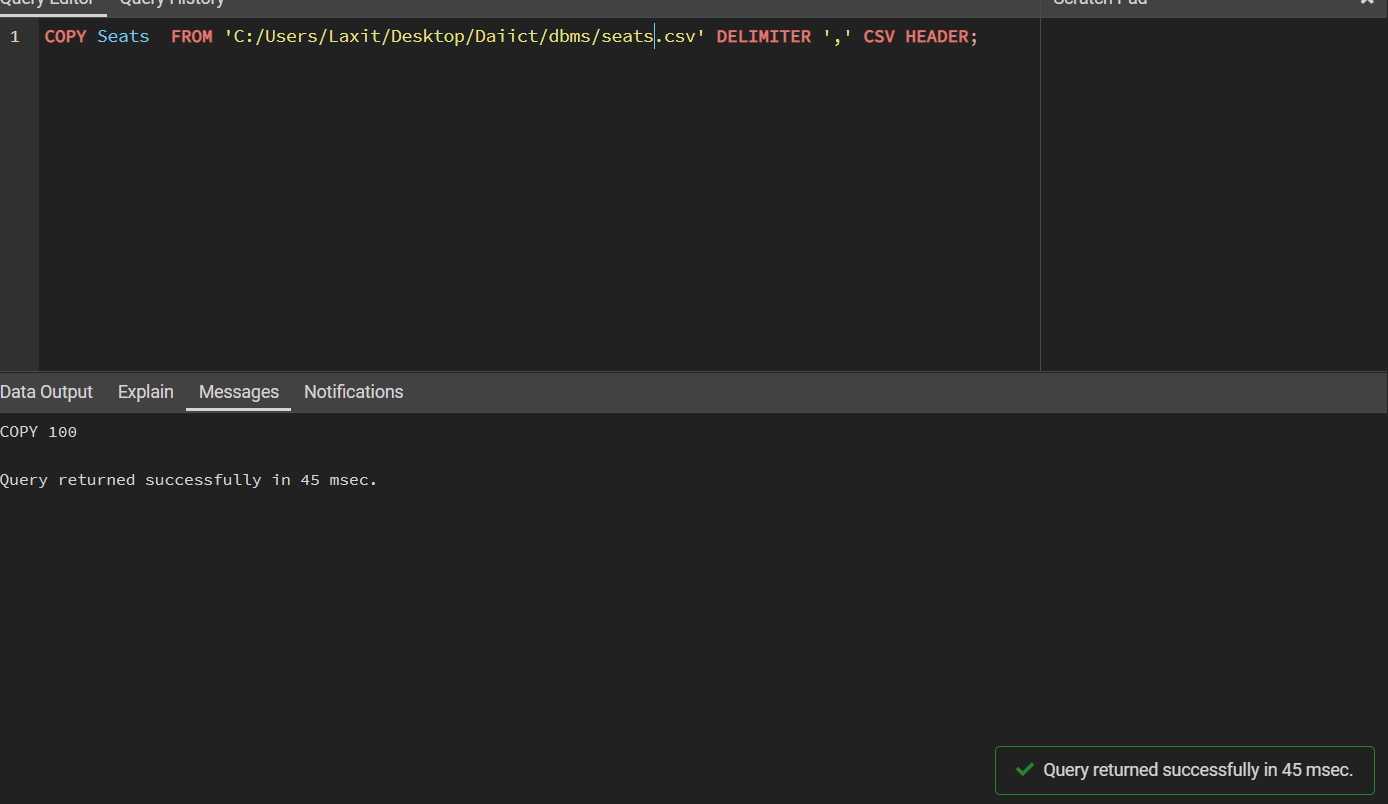
**10>Shows**

COPY Shows  FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/shows.csv' DELIMITER ',' CSV HEADER;



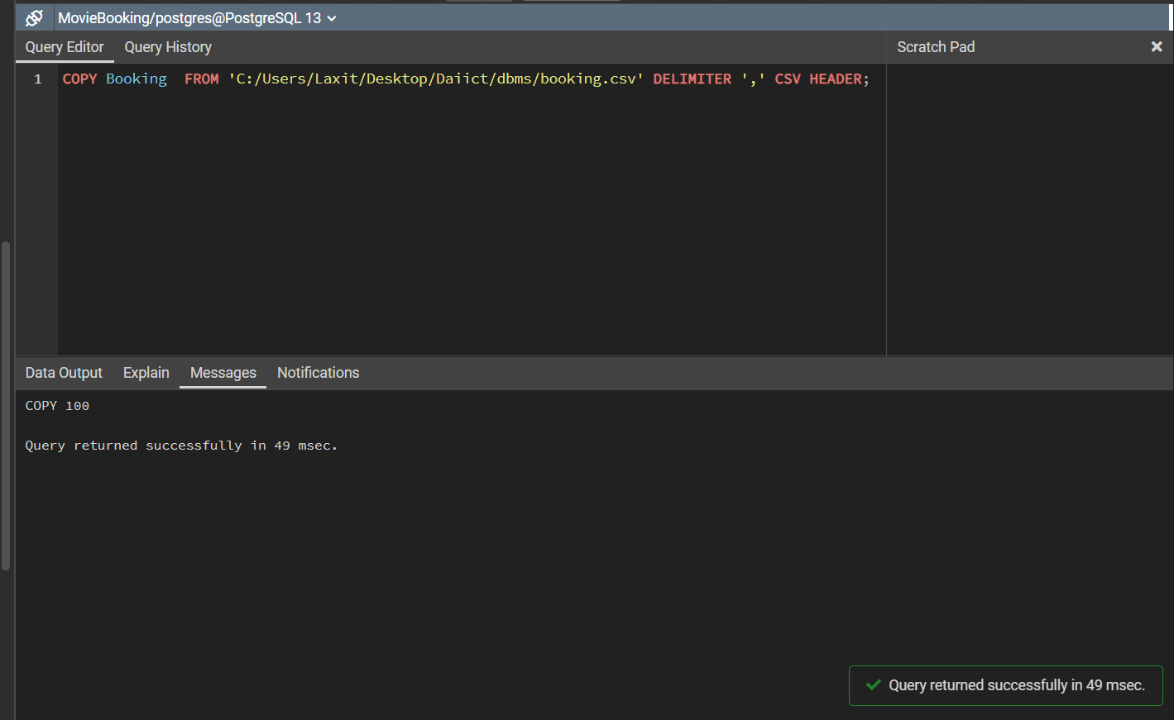
**11>Seats**

COPY Seats  FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/seats.csv' DELIMITER ',' CSV HEADER;



**12>Booking**

COPY Booking  FROM 'C:/Users/Laxit/Desktop/Daiict/dbms/booking.csv' DELIMITER ',' CSV HEADER;

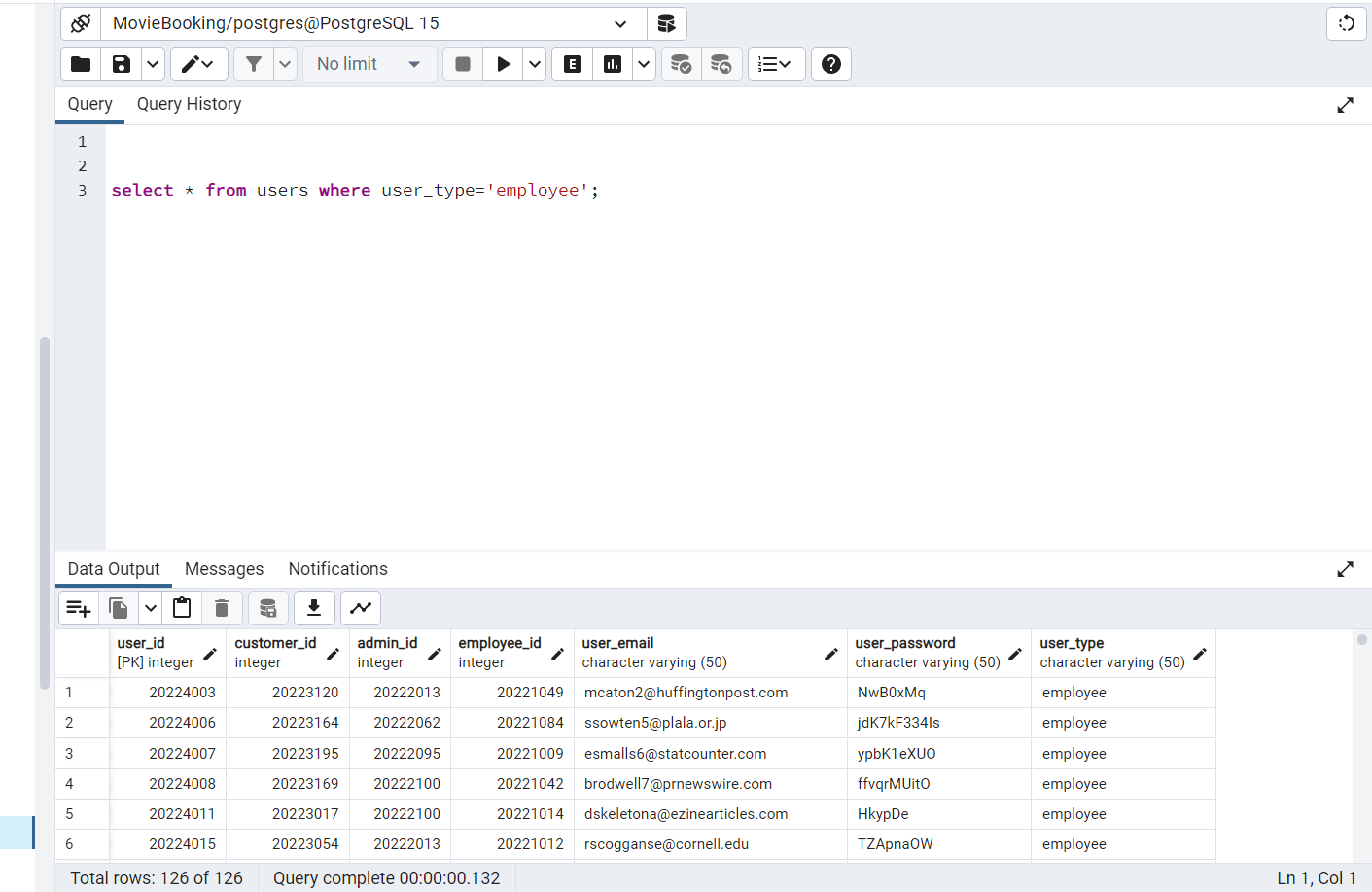


**9. List of All SQL Queries**

**1>show user table whose user\_type is employee**

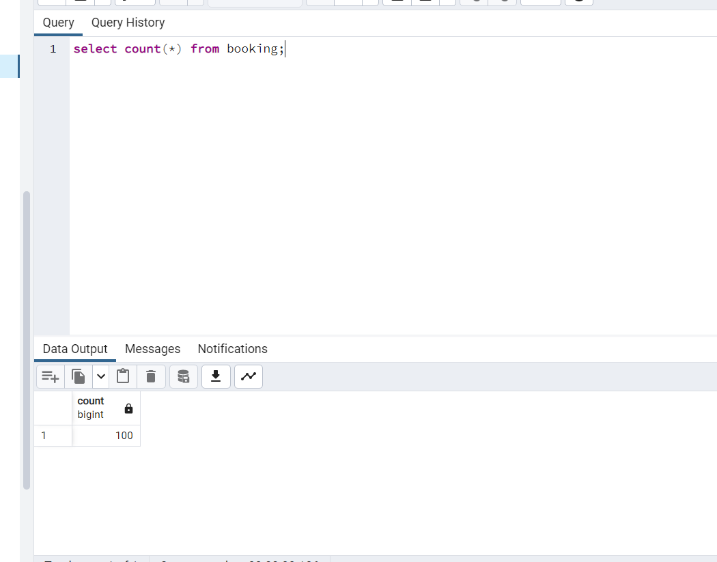
select \* from users where user\_type='employee';

**Output:**

****

**2>Count all The booking Numbers.**

select count(\*) from booking;

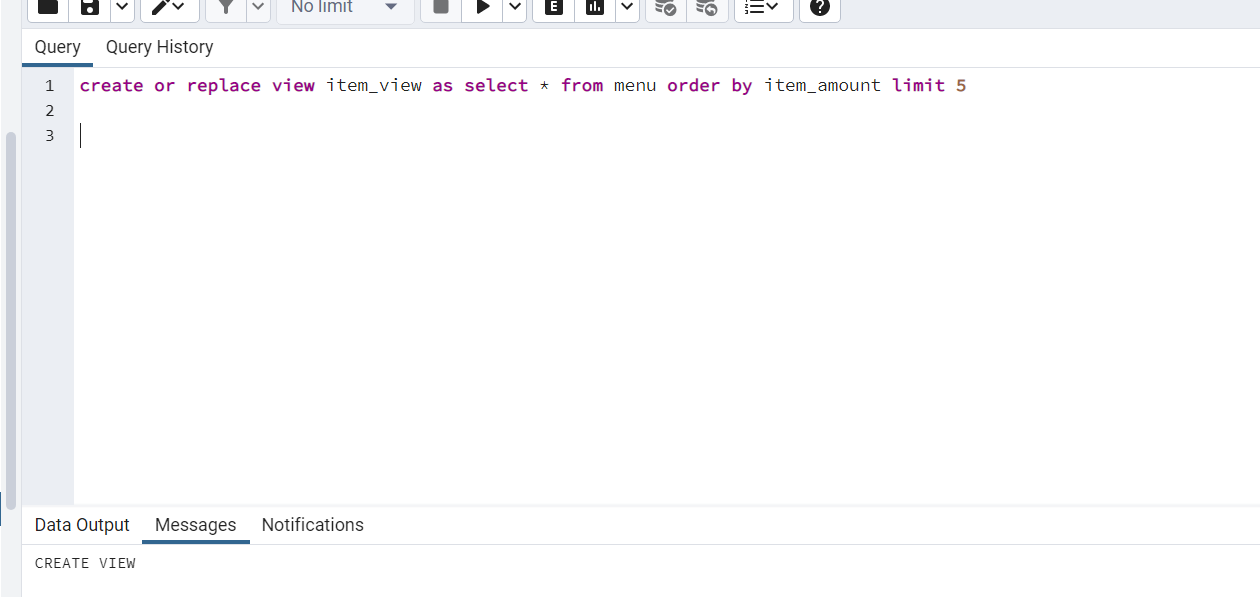
**Output:**

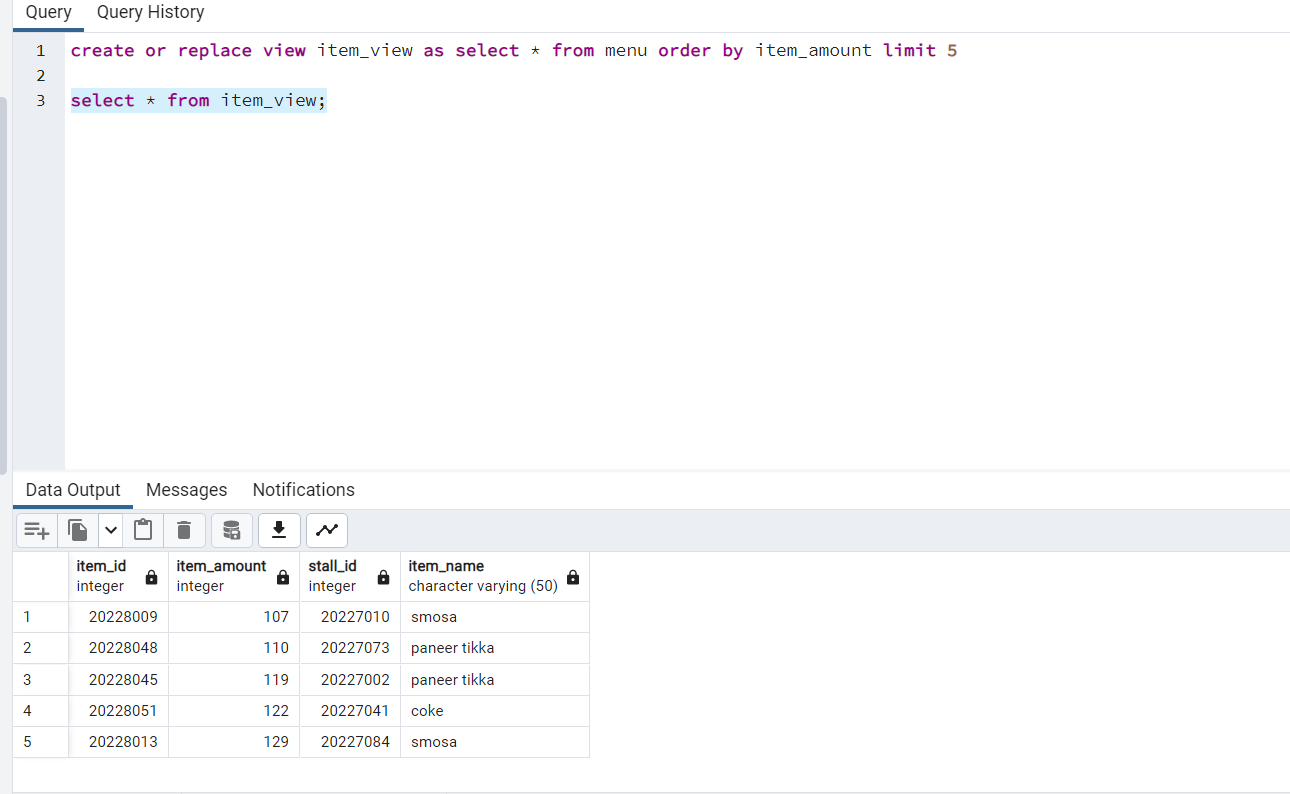
**3>Create view of 5 highest item amount and display it.**

create or replace view item\_view as select \* from menu order by item\_amount limit 5

select \* from item\_view;

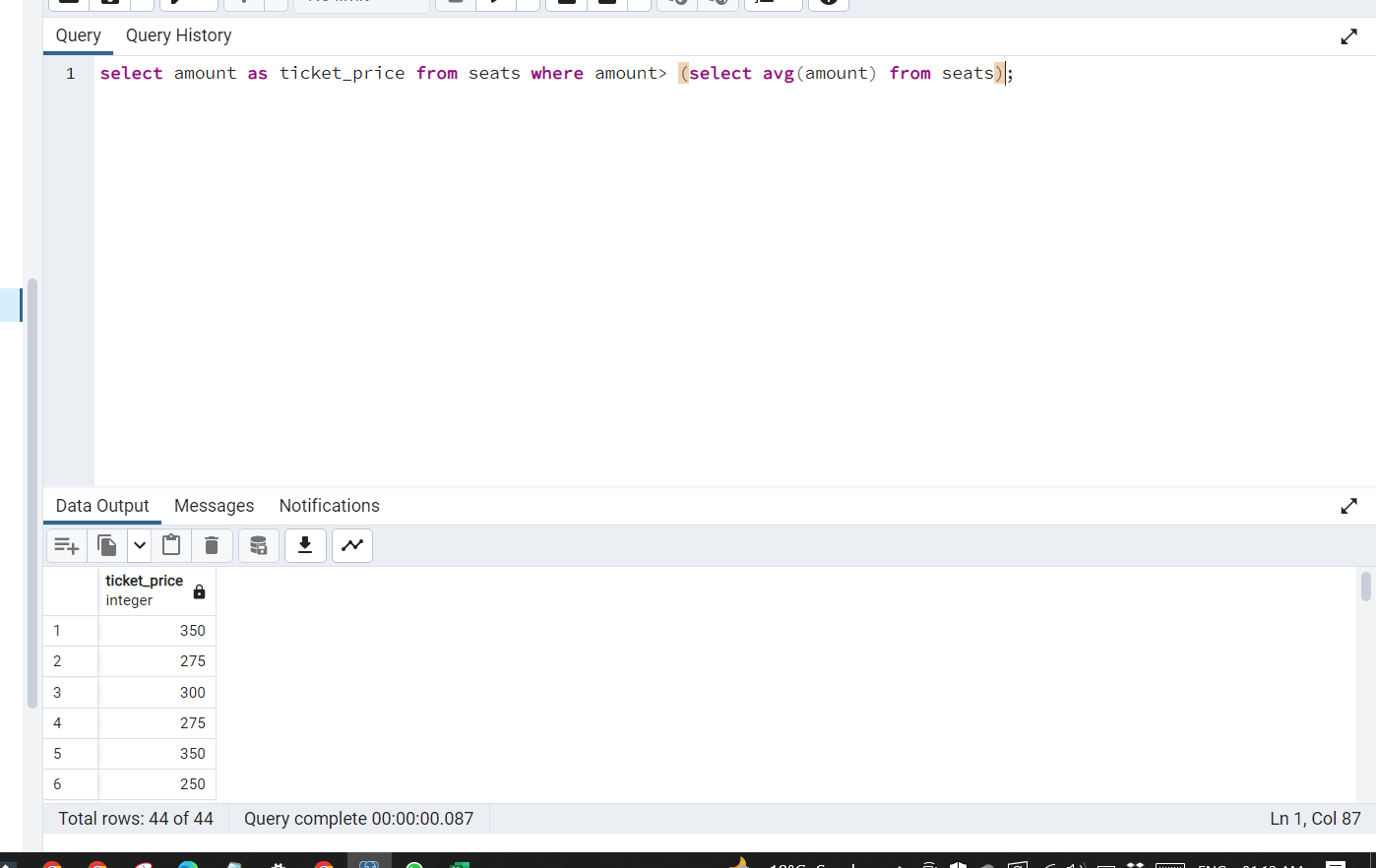
**Output:**

****

****

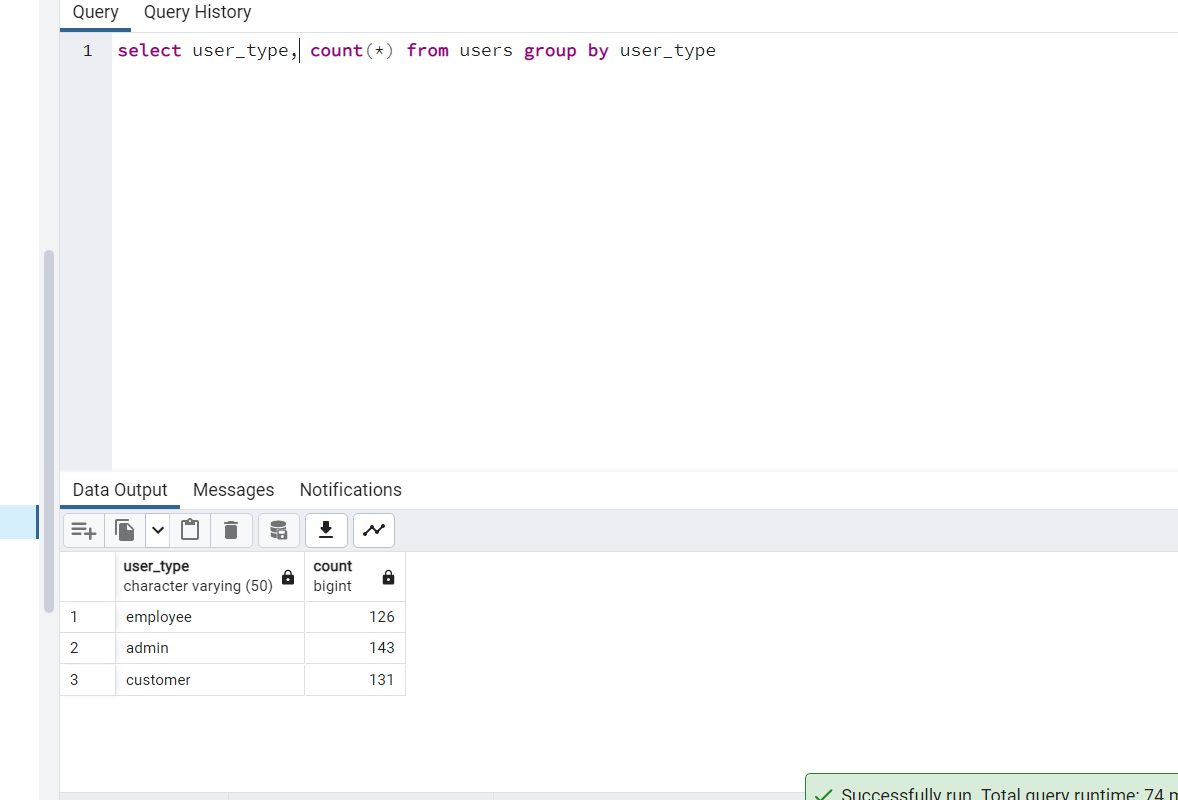
**4>Find ticket\_price which is greater than average ticket\_price**

**ans>**select amount as ticket\_price from seats where amount> (select avg(amount) from seats);

**Output:**

**5>Find count of different types of user\_role**

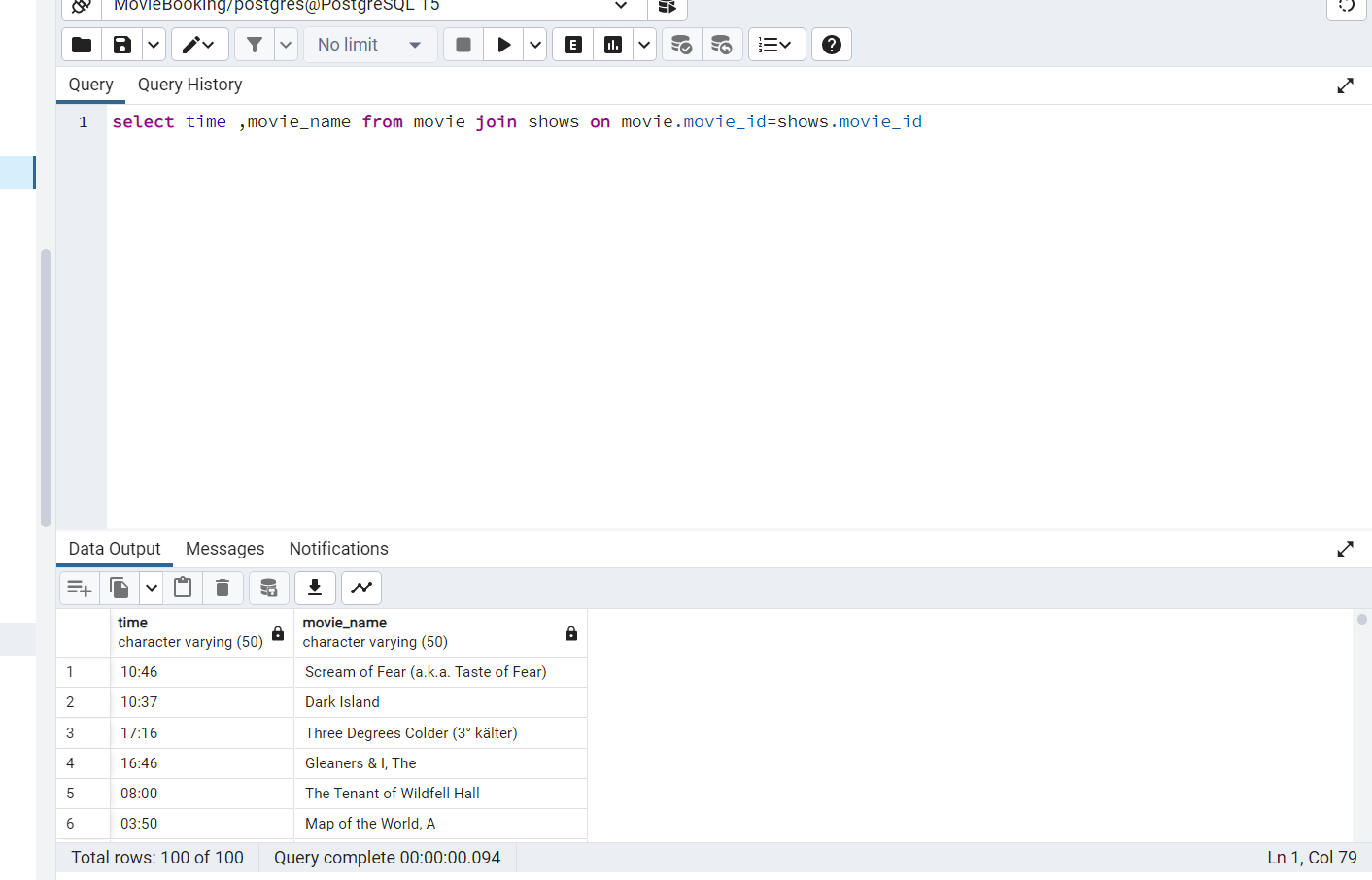
**ans>**select user\_type, count(\*) from users group by user\_type

**Output:**

**6>Display show time and movie name**

select time ,movie\_name from movie join shows on movie.movie\_id=shows.movie\_id

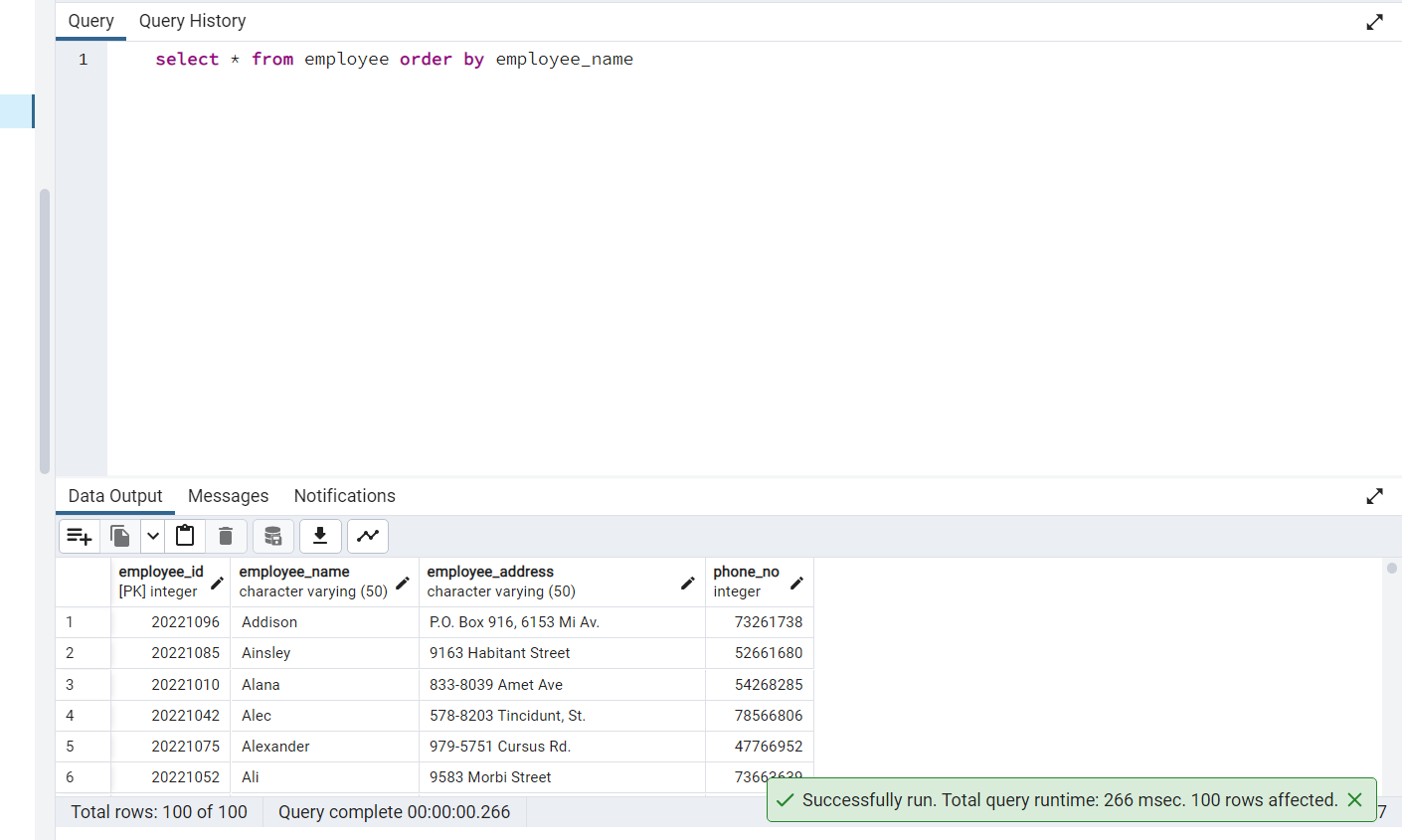
**Output:**

****

**7>Sort Employee Name**

select \* from employee order by employee\_name

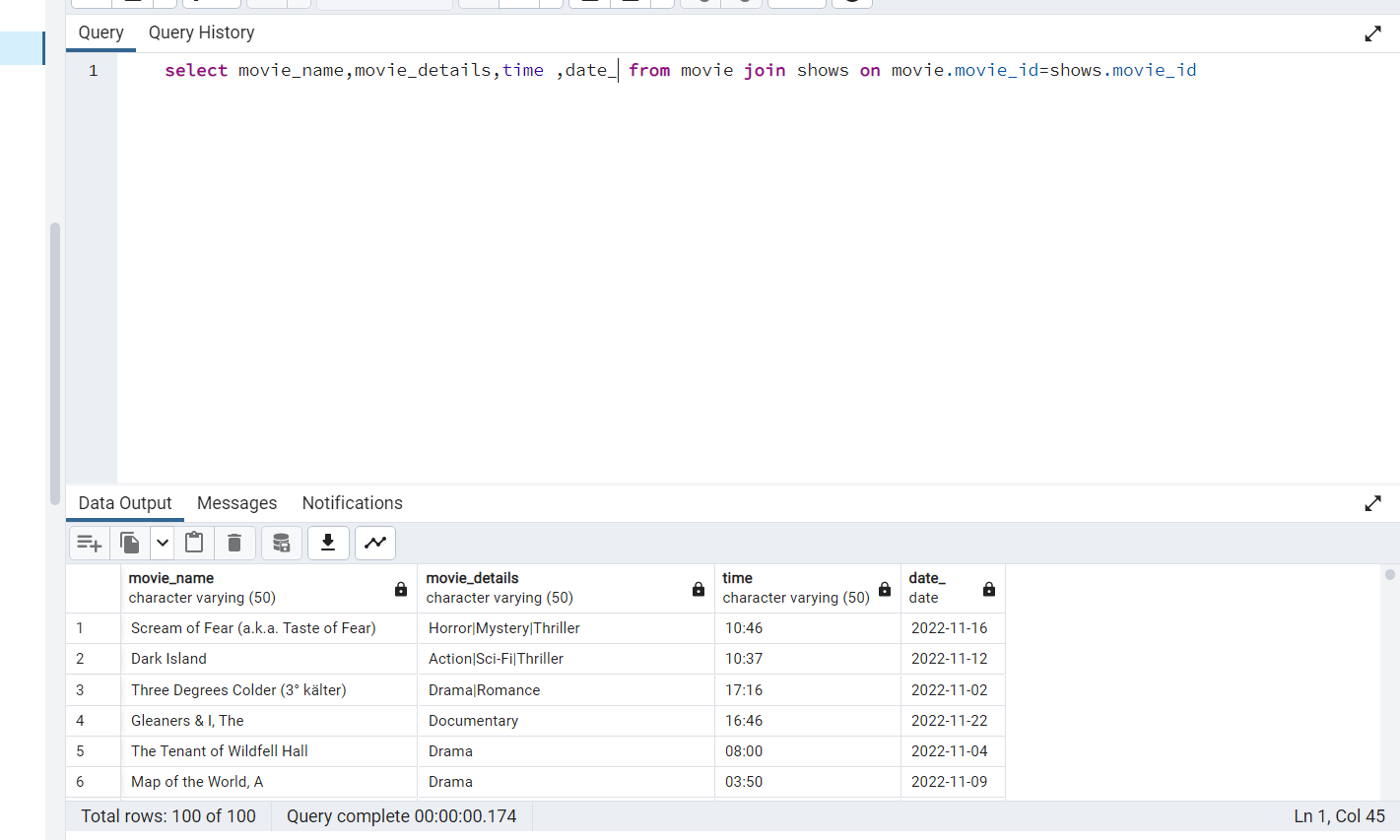
**Output:**

****

**8>Show  Movie Name, Details and time and date .**

**ans>** select movie\_name,movie\_details,time ,date\_ from movie join shows on movie.movie\_id=shows.movie\_id

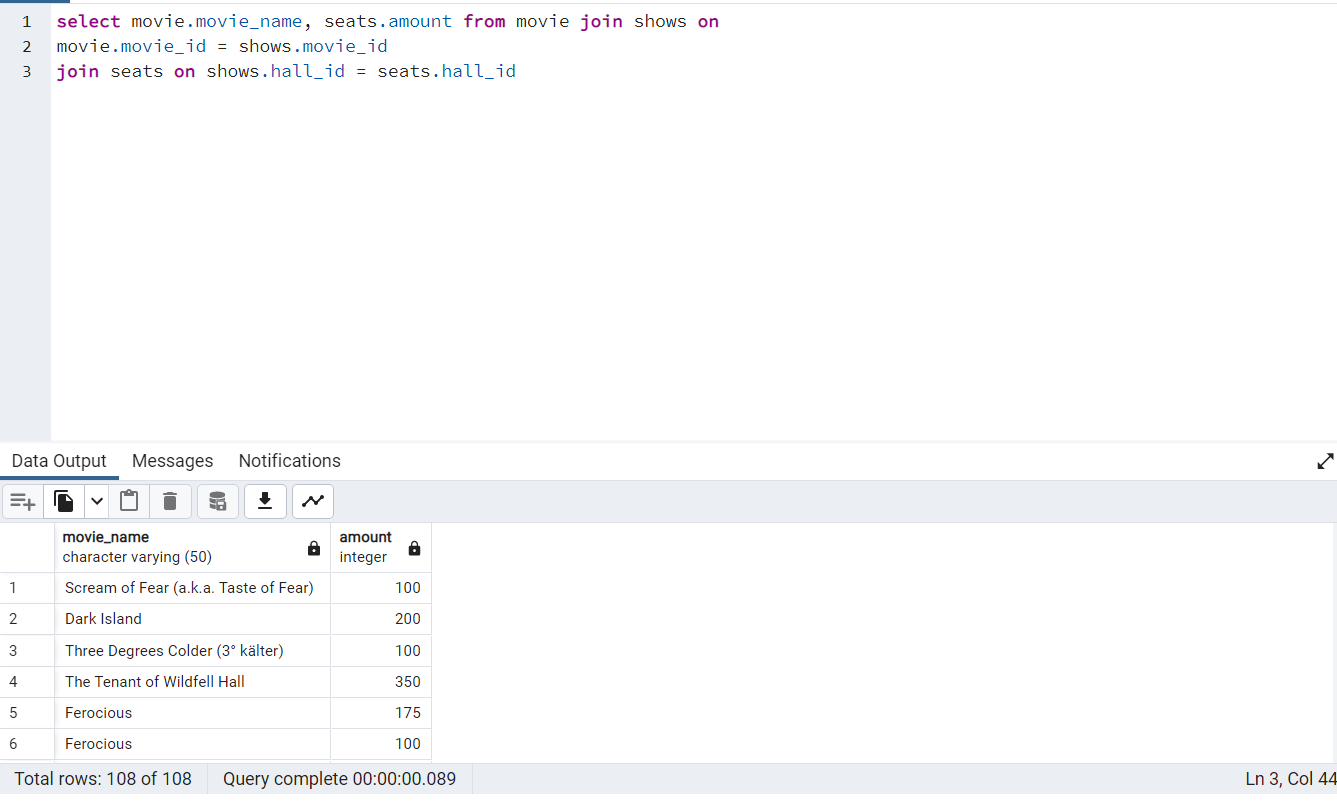
**Output:**



**9>Show Movie Name and amount**

**ans>**select movie.movie\_name, seats.amount from movie join shows on movie.movie\_id = shows.movie\_id  join seats on shows.hall\_id = seats.hall\_id

**Output:**

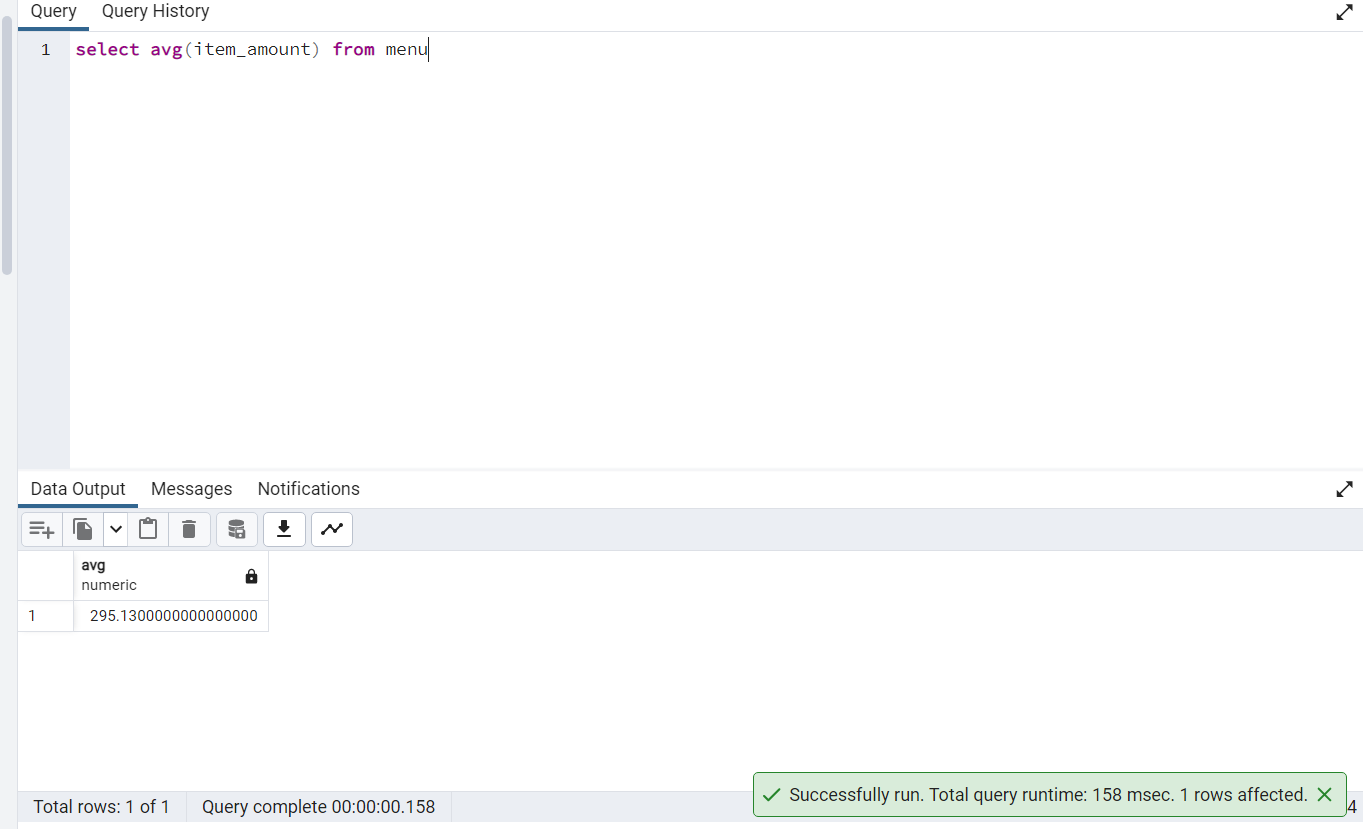


**10>select average item\_amount**

**ans>**

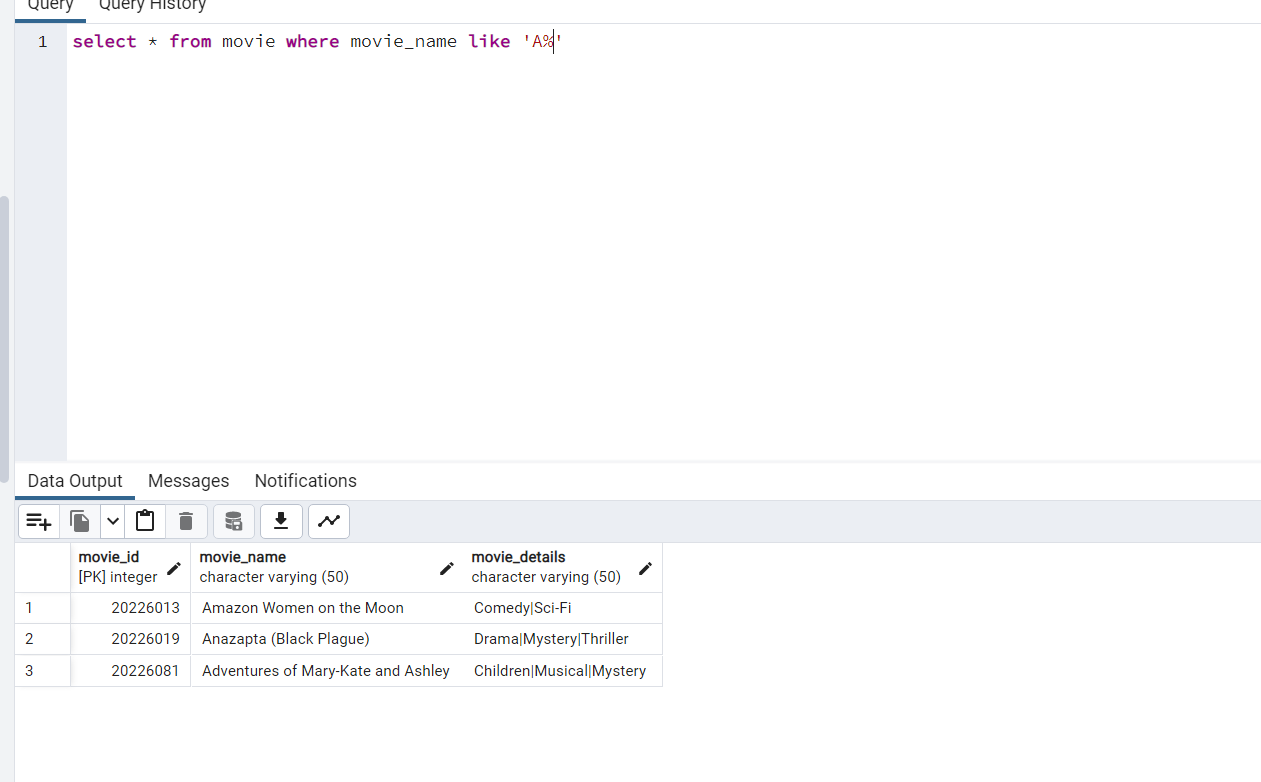
select avg(item\_amount) from menu

**Output:**

****

**11>Show Movie Name whose name start with 'A'**

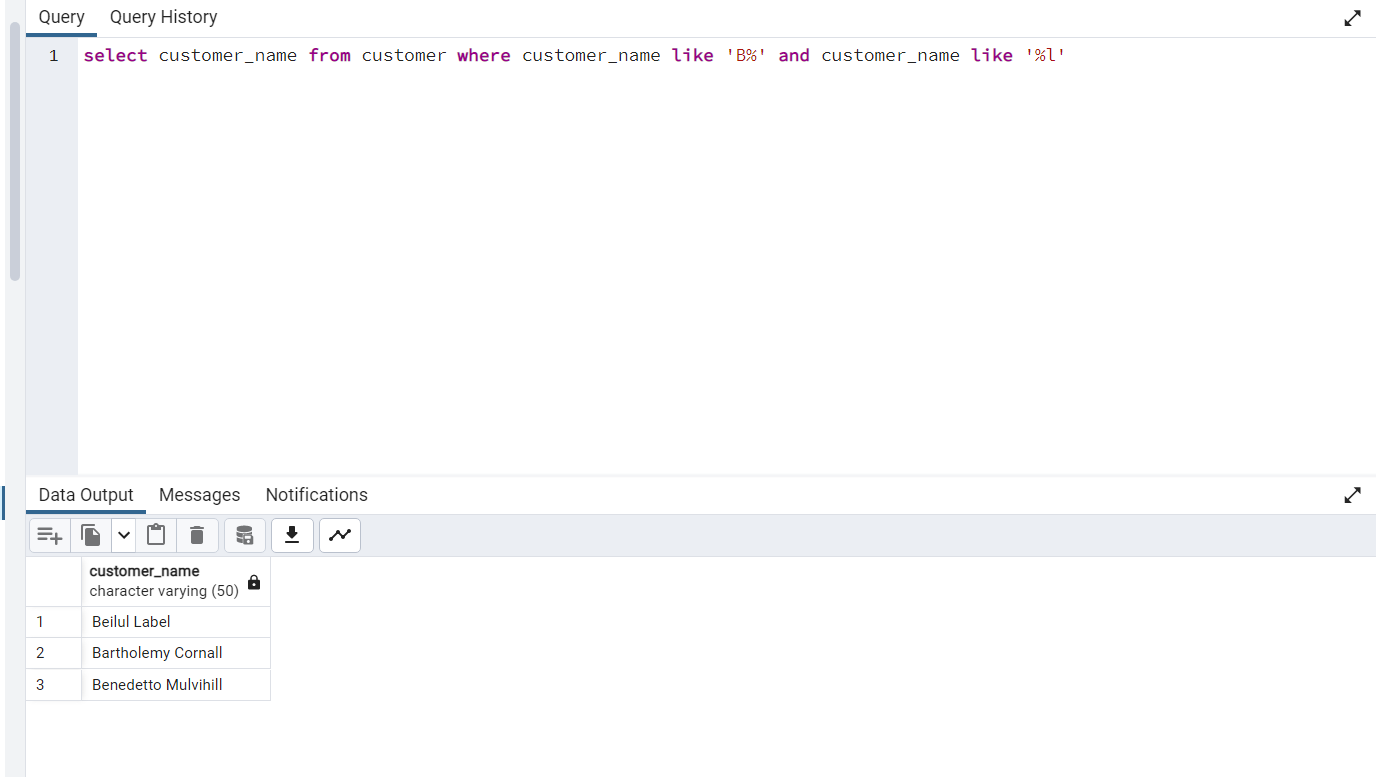
select \* from movie where movie\_name like 'A%'

****

**12>Show Customer Name whose name start with 'B' and End With 'L'**

**ans>**select customer\_name from customer where customer\_name like 'B%' and customer\_name like '%l'

**Output:**

****

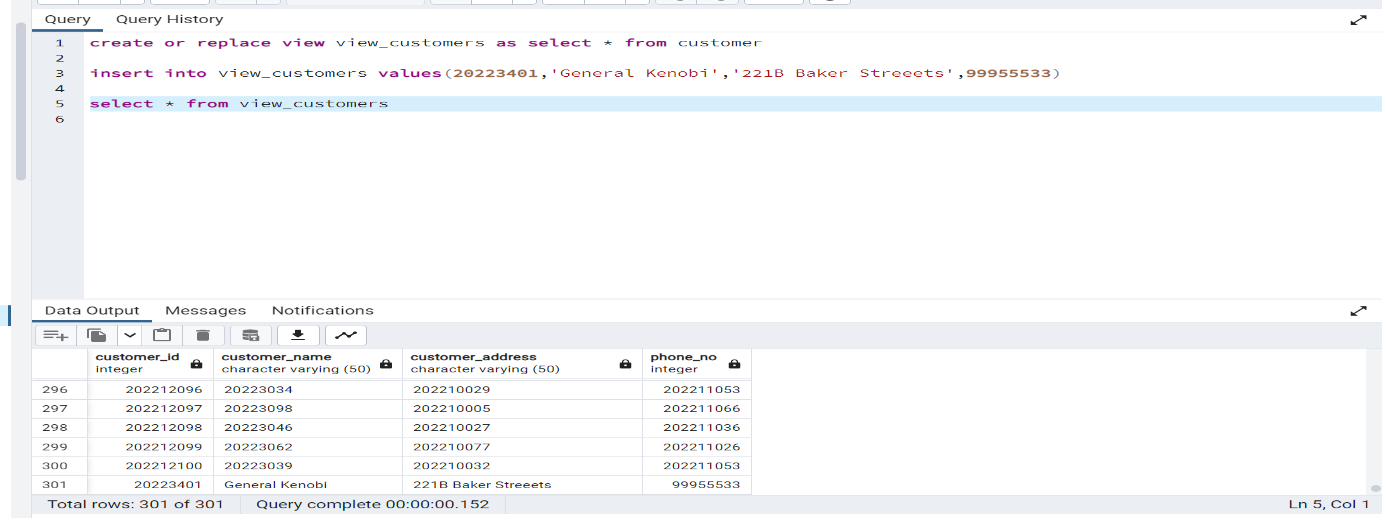
**13>Create a view for customer\_details ,insert and display the view**

create or replace view view\_customers as select \* from customer

insert into view\_customers values(20223401,'General Kenobi','221B Baker Streeets',99955533)

select \* from view\_customers

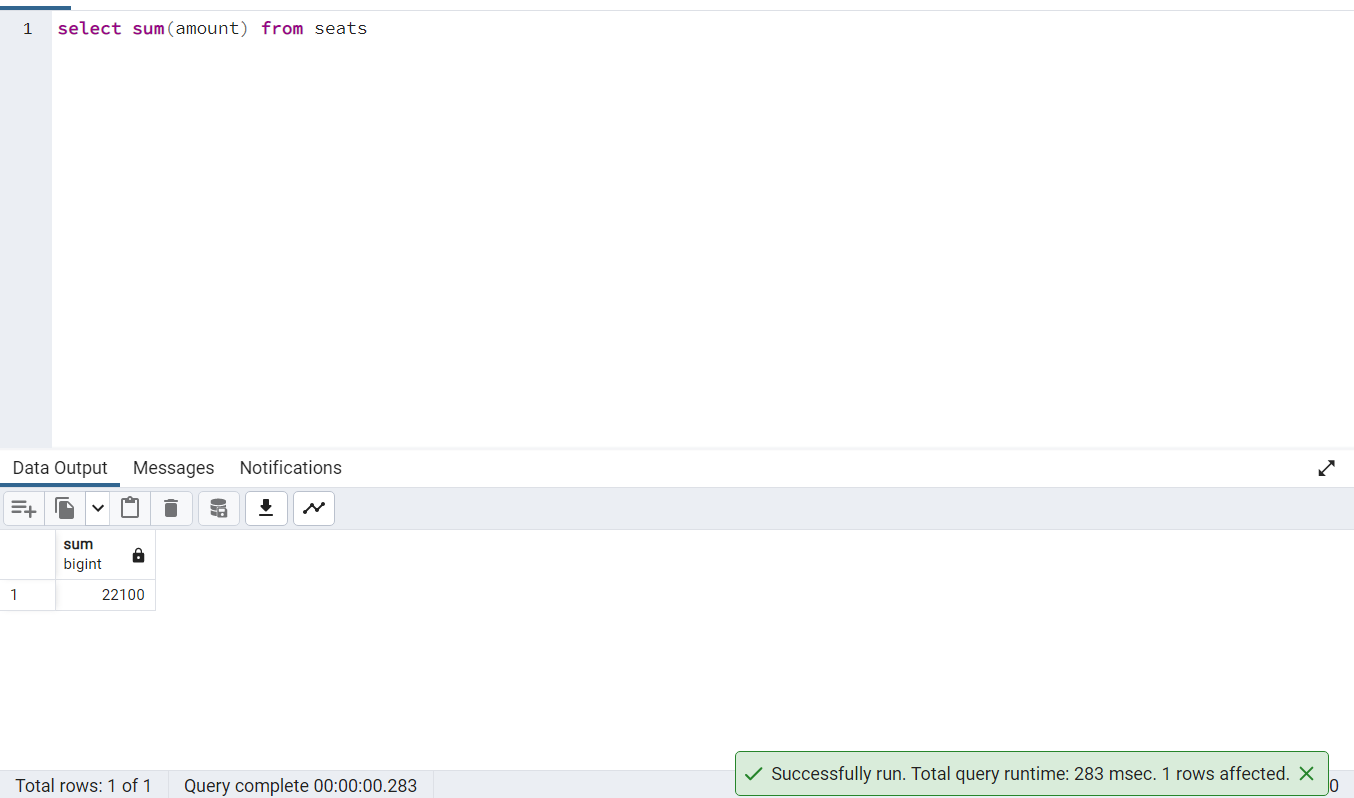
**Output:**

****

**14>Show the sum of total seat amount**

**ans>**select sum(amount) from seats

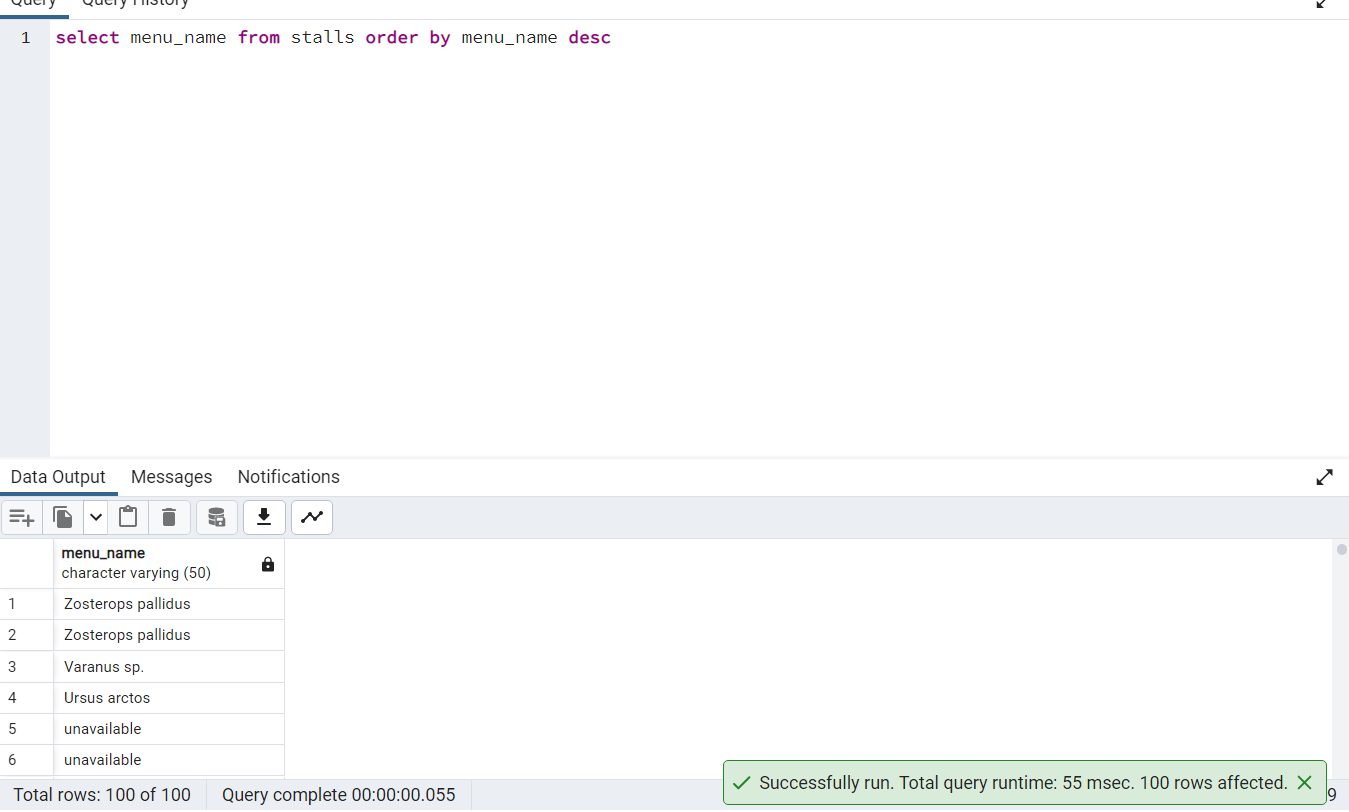
**Output:**

****

**15>Display Menu Name Descending wise.**

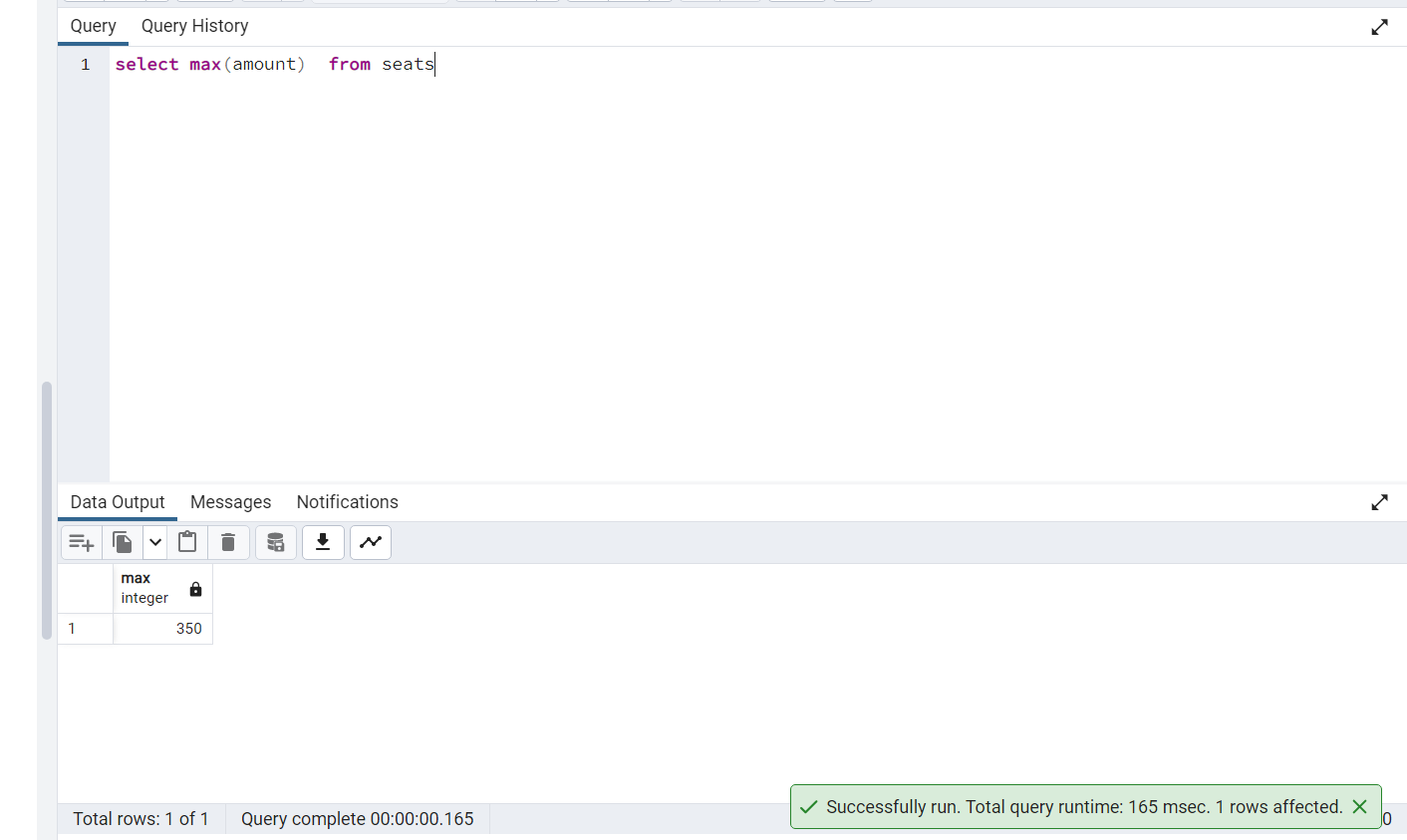
select menu\_name from stalls order by menu\_name desc

**Output:**

****

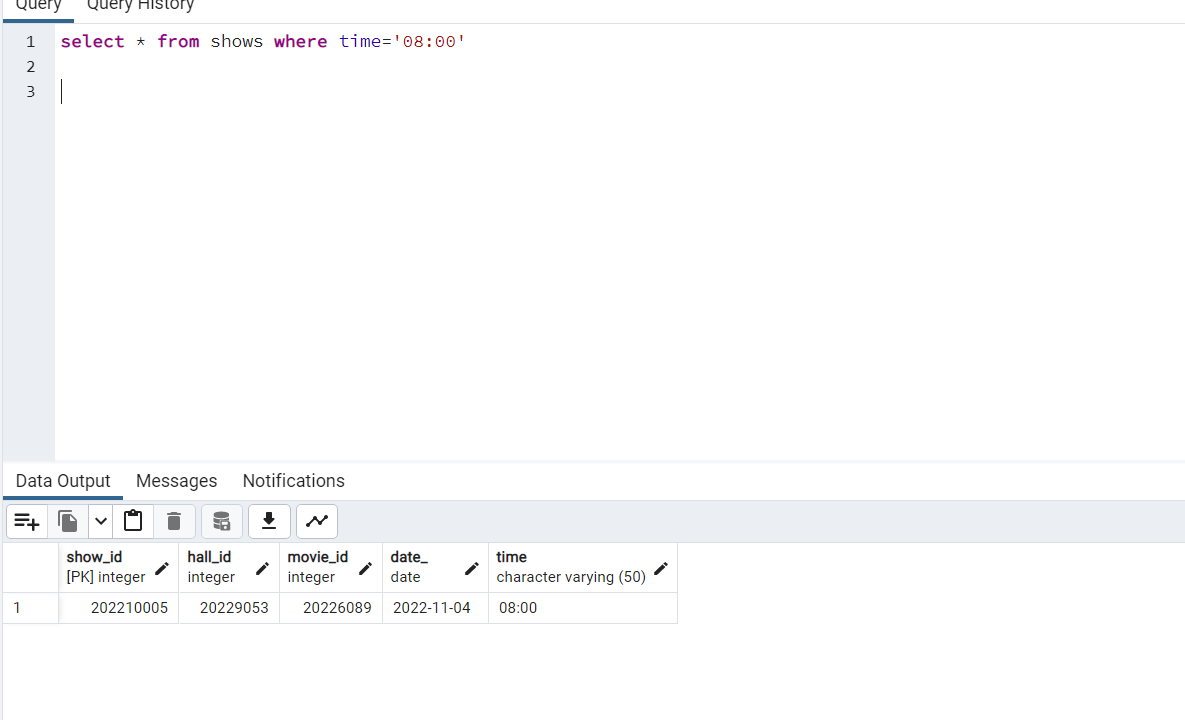
**16>  Display record for the maximum total amount in seats**

select max(amount) from seats

****

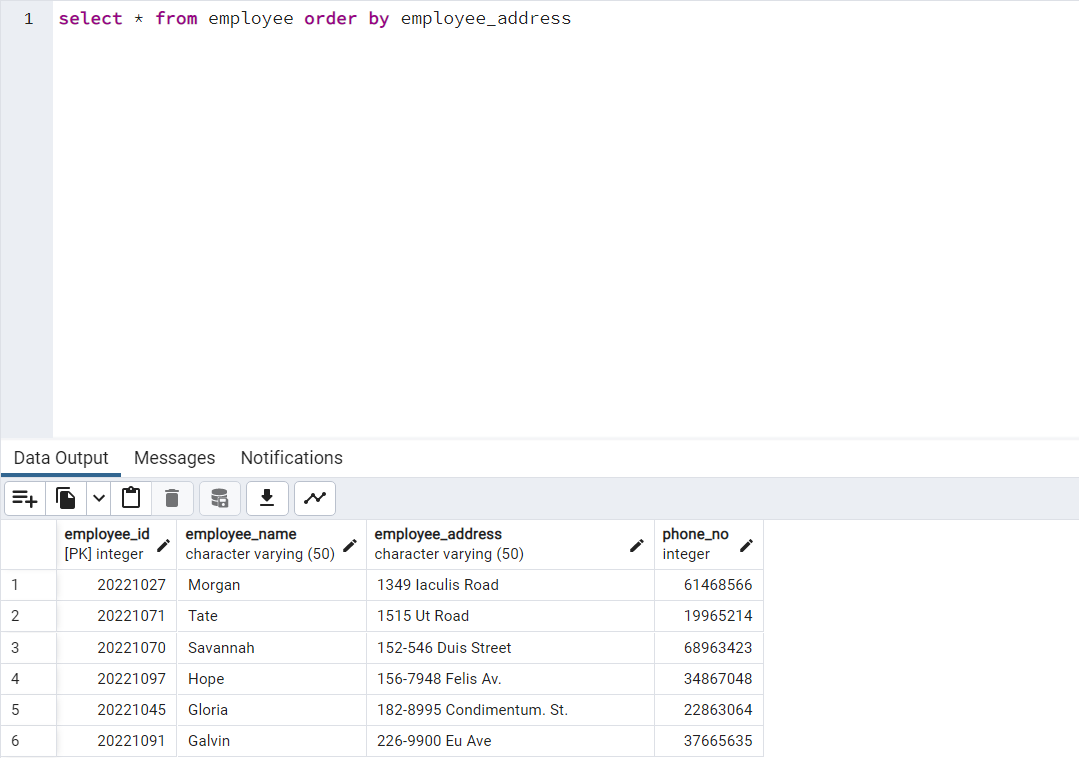
**17> show all the movies available at 08:00**

select \* from shows where time='08:00'

****

**18> Display employee details with sorted Email**

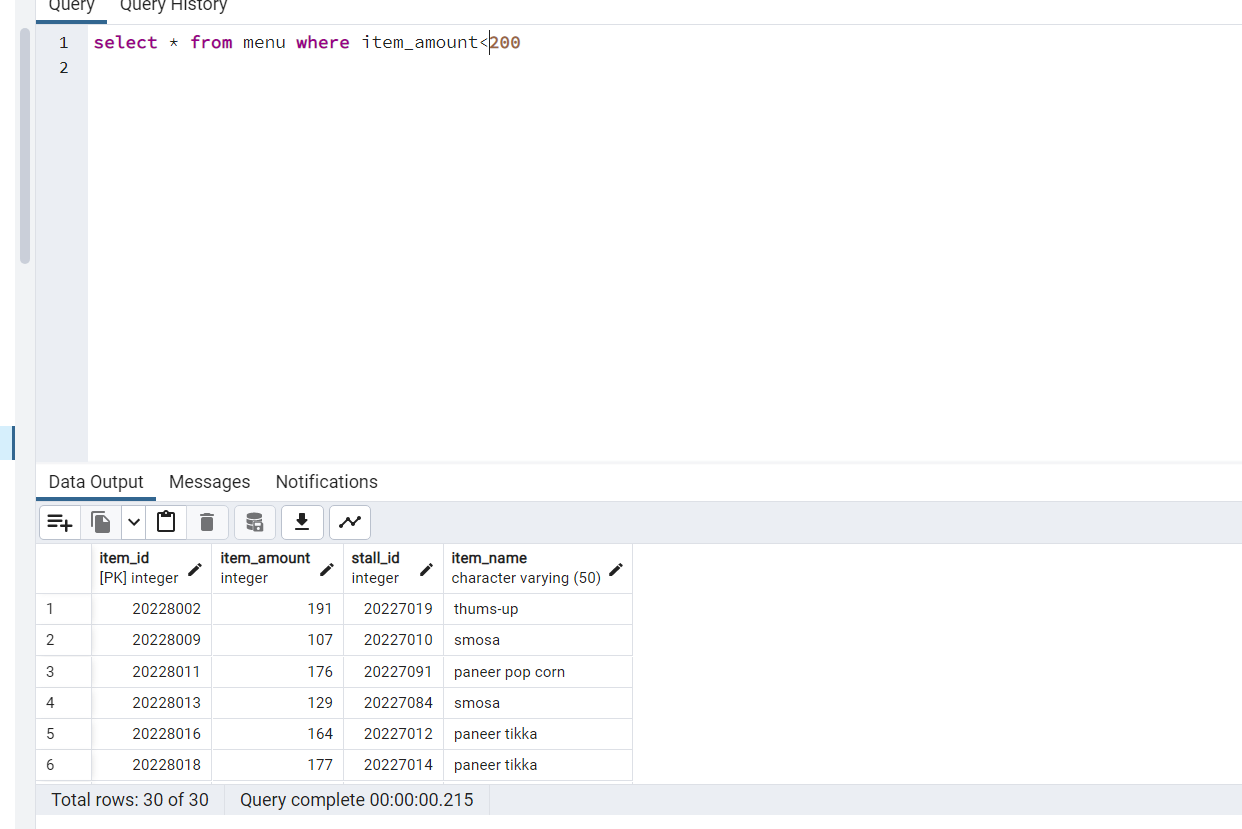
select \* from employee order by employee\_address

****

**19> show all the cinemas where menu price is less then Rs200**

**ans>**select \* from menu where item\_amount<200

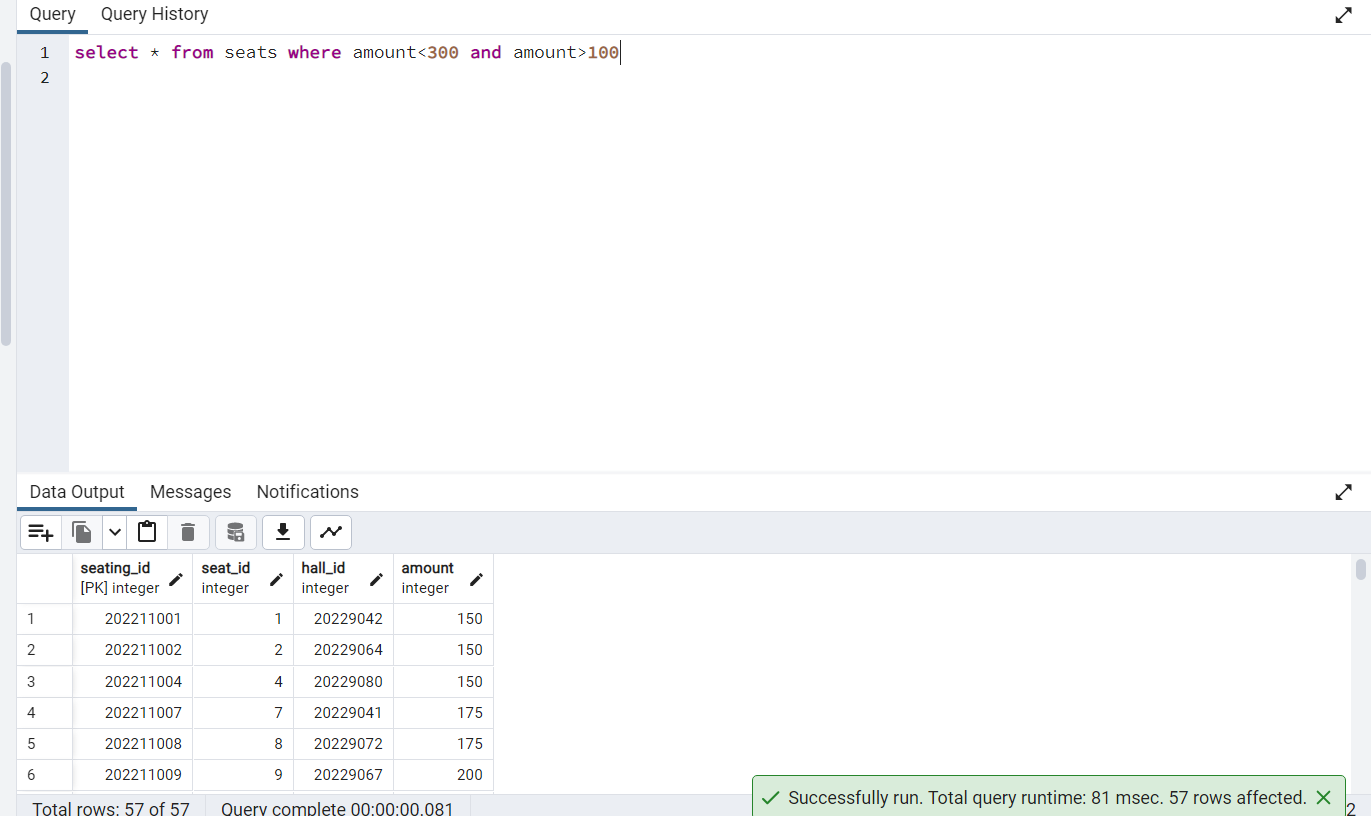
**Output:**

****

**20>  show seats how's amount is less than Rs300 and greater than 100**

select \* from seats where amount<300 and amount>100

**Output:**

****

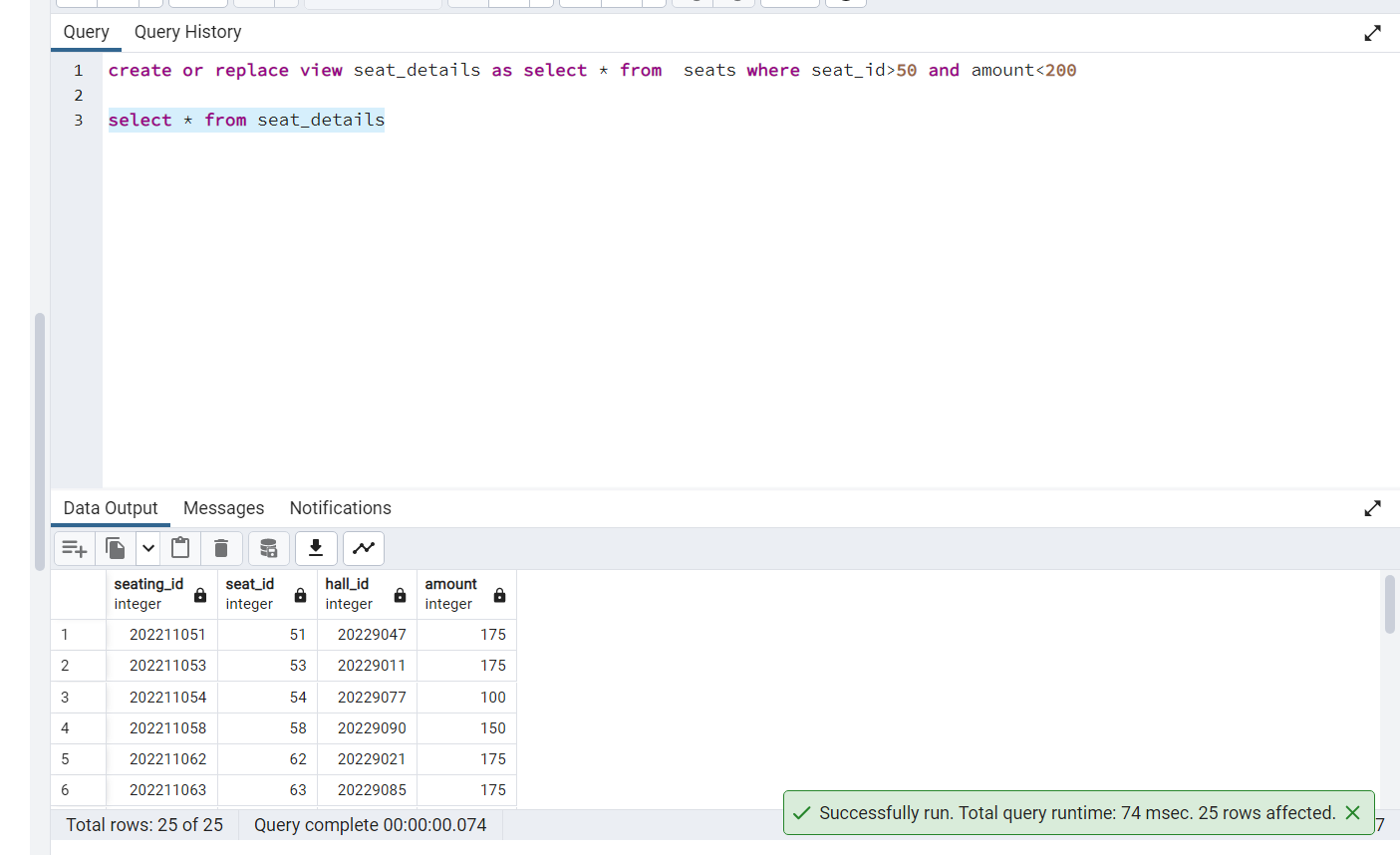
**21> create a view of seats details where seat\_id is greater than 50 and amount less than 200**

**ans>**

create or replace view seat\_details as select \* from  seats where seat\_id>50 and amount<200

select \* from seat\_details

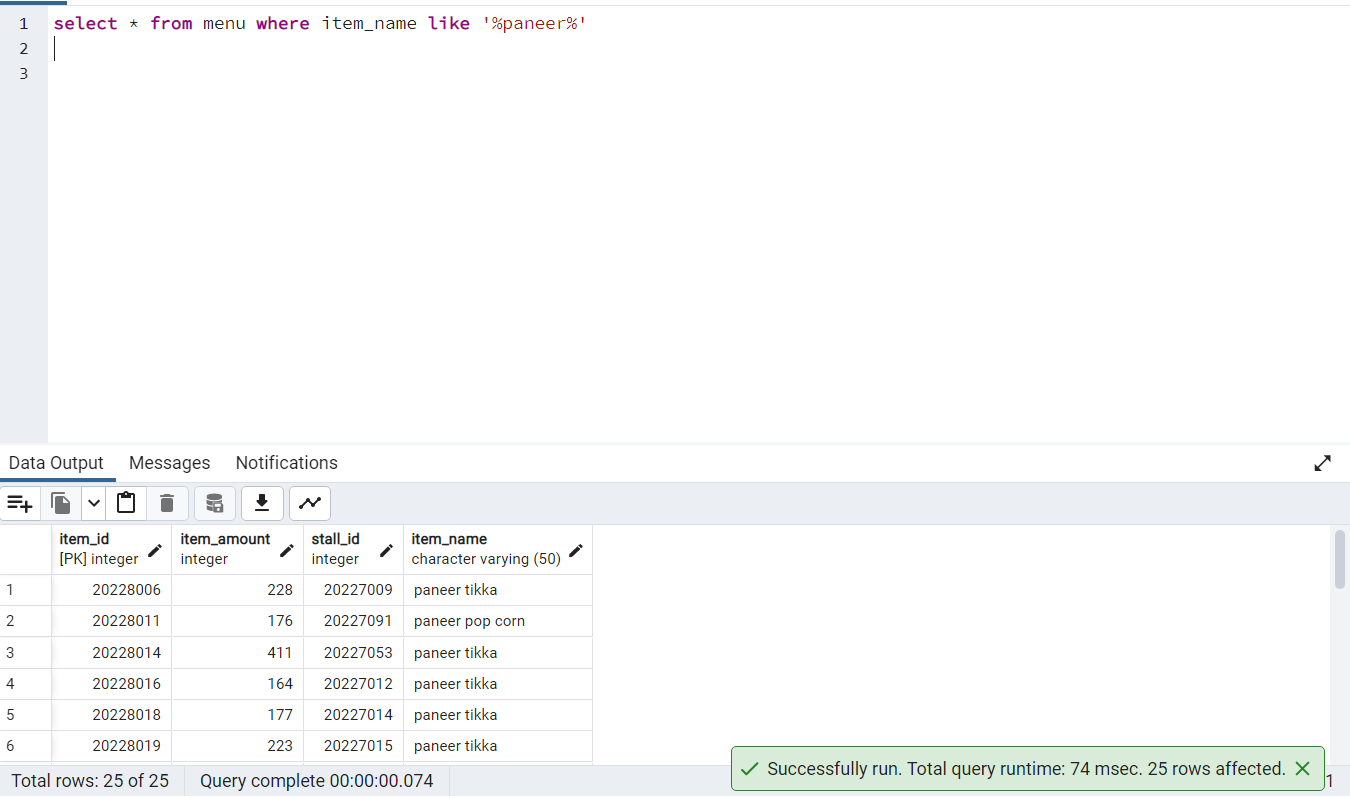
**Output:**

****

**22>  list down the item details which has paneer in it**

**ans>**select \* from menu where item\_name like '%paneer%'

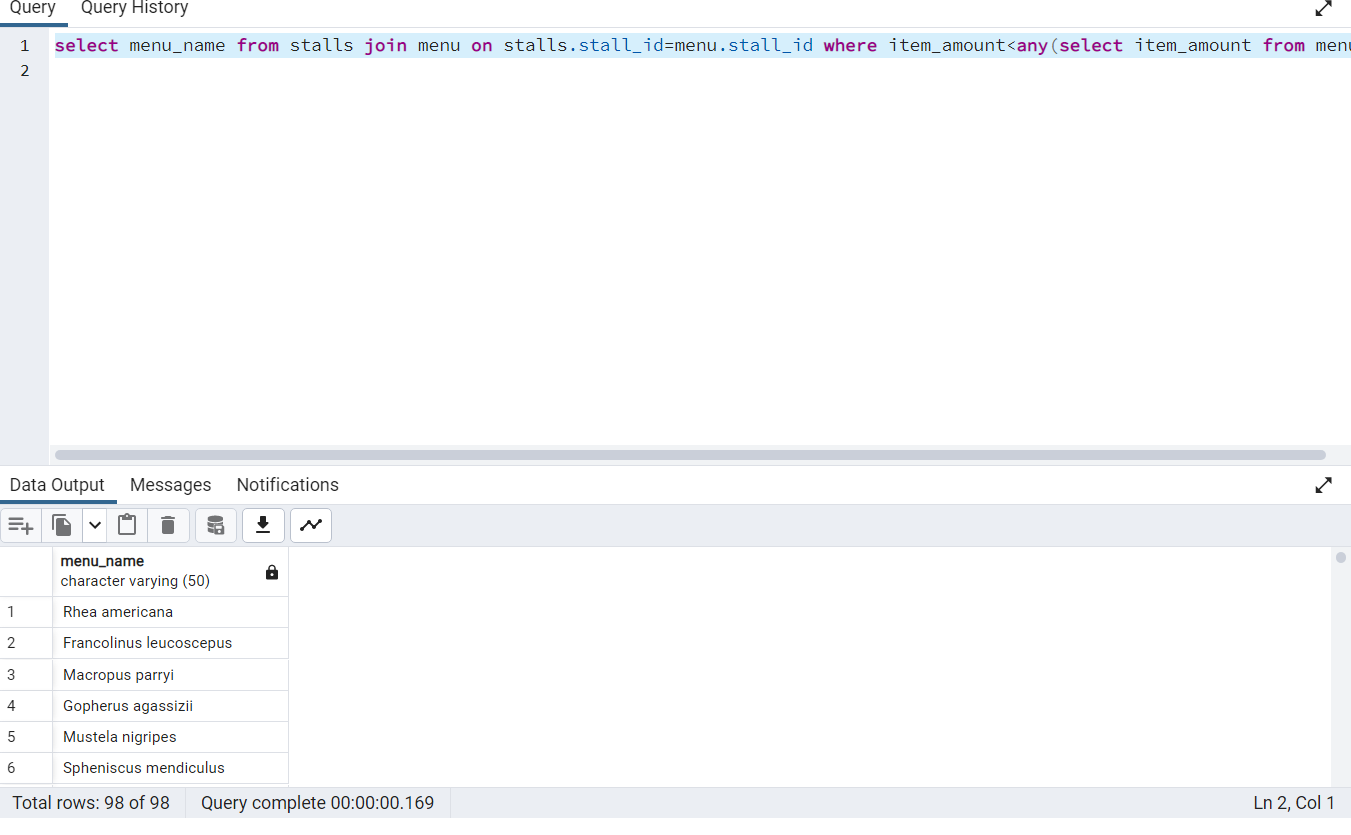
**Output:**

****

**23>  Display the menu\_name whose item\_amount is less than average of item amount**

**ans>** select menu\_name from stalls join menu on stalls.stall\_id=menu.stall\_id where item\_amount<any(select item\_amount from menu)

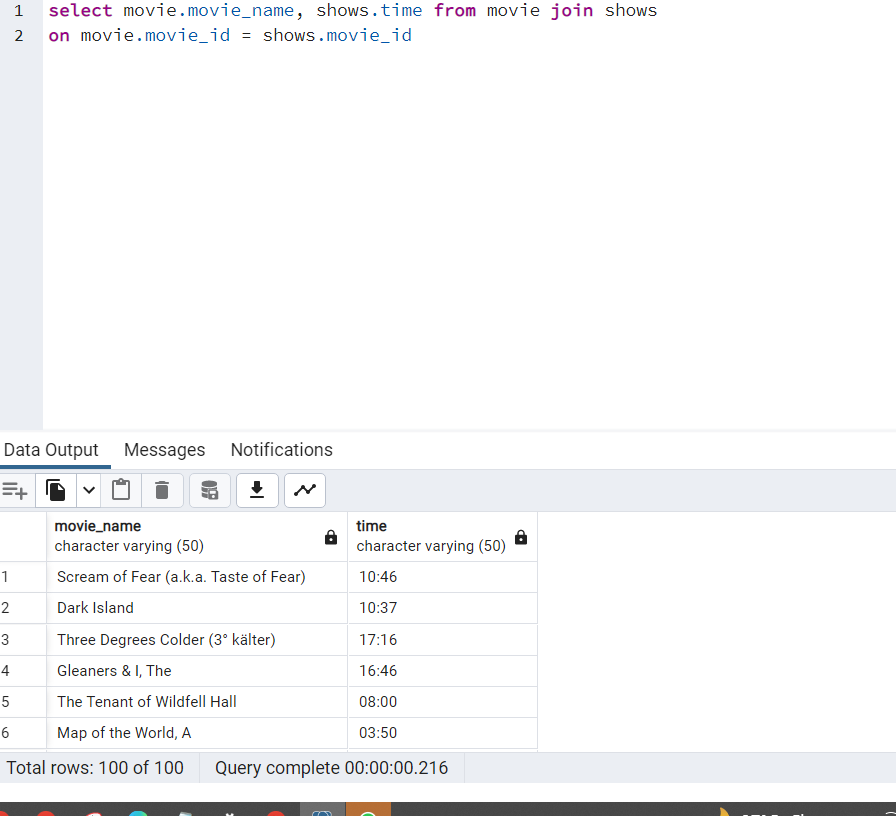
**Output:**



**24> Display all the Movie name and show time**

select movie.movie\_name, shows.time from movie join shows

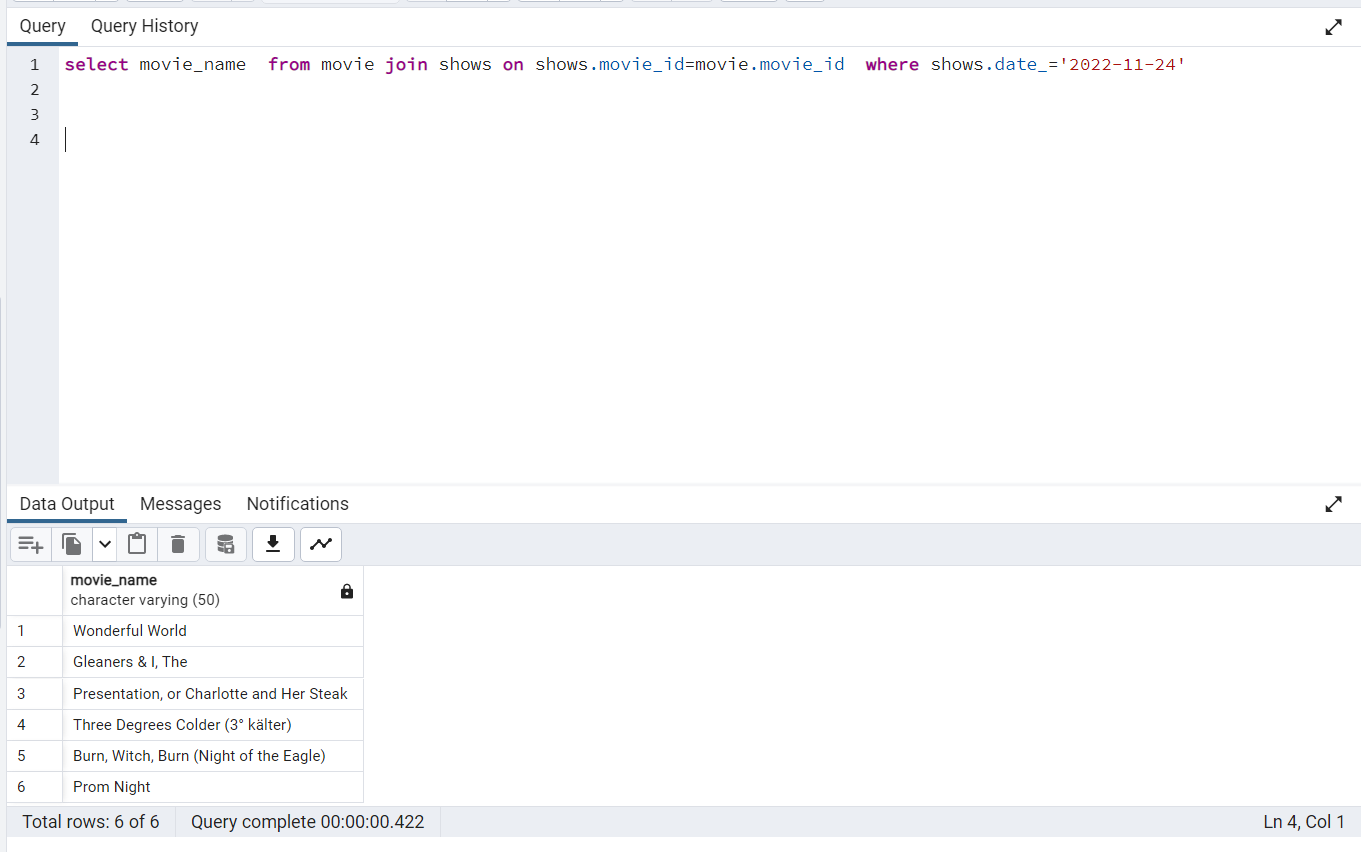
on movie.movie\_id = shows.movie\_id

****

**25>  display all the movies on 27th Nov 2022**

**ans>**

select movie\_name  from movie join shows on shows.movie\_id=movie.movie\_id  where shows.date\_='2022-11-24'

****

**26>  Display the timing of movie name ‘Mad Money’**

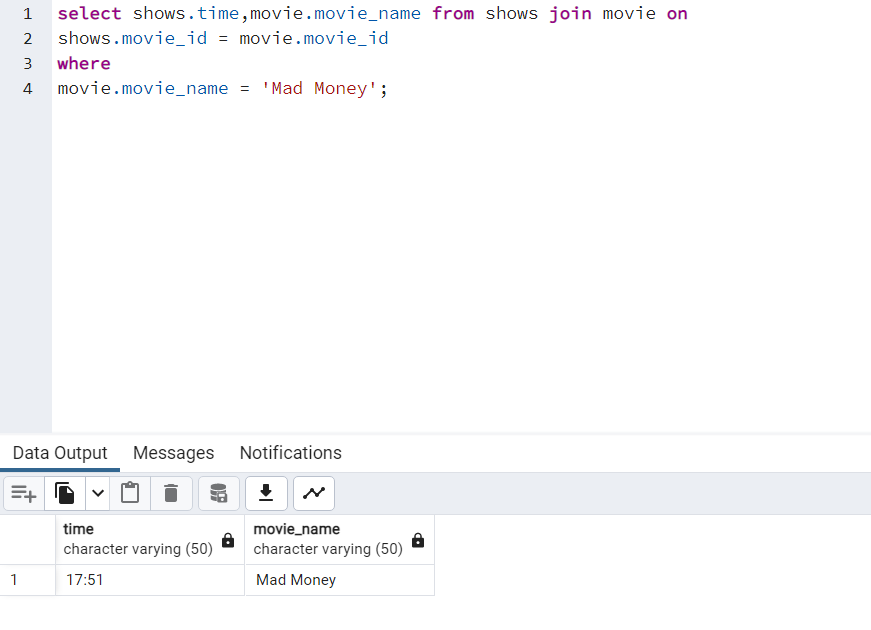
select shows.time,movie.movie\_name from shows join movie on

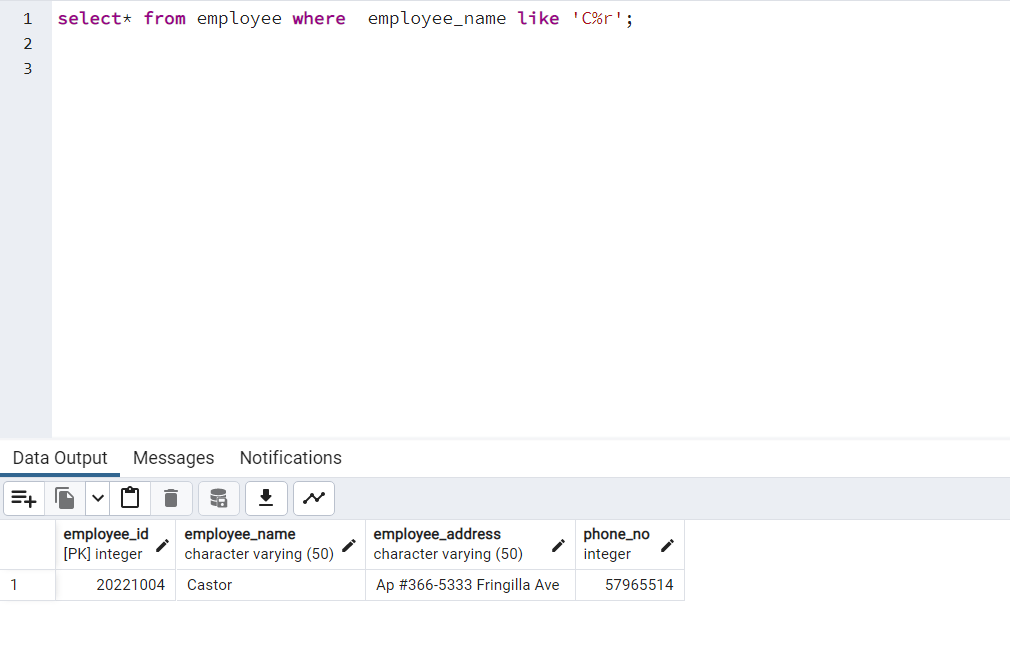
shows.movie\_id = movie.movie\_id

where

movie.movie\_name = 'Mad Money';

**Output:**

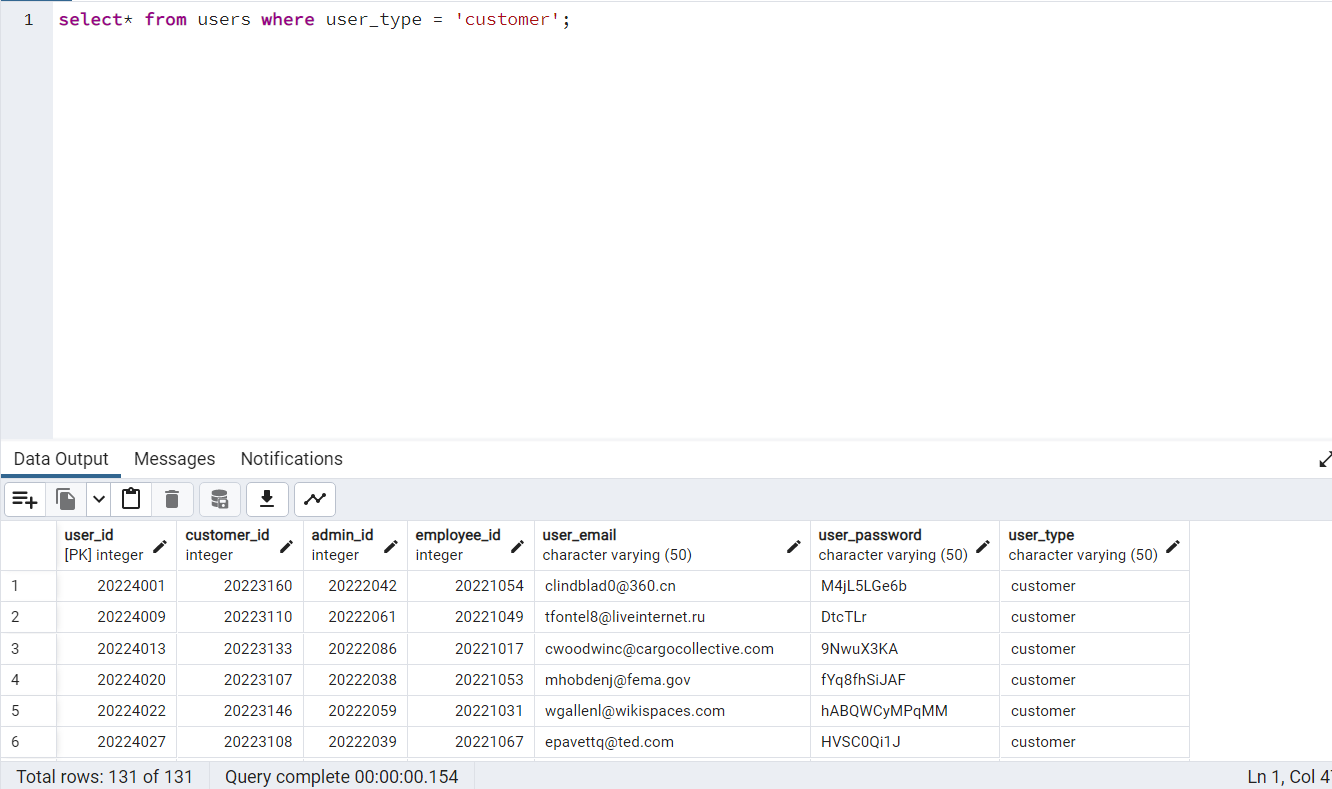
****

**27>  Display employee name whose name starts with ‘C’ and ends with ‘r’ans>**select\* from employee where  employee\_name like 'C%r';****

**28>  show all the users who are customers**

select\* from users where user\_type = 'customer';

**Output:**

****

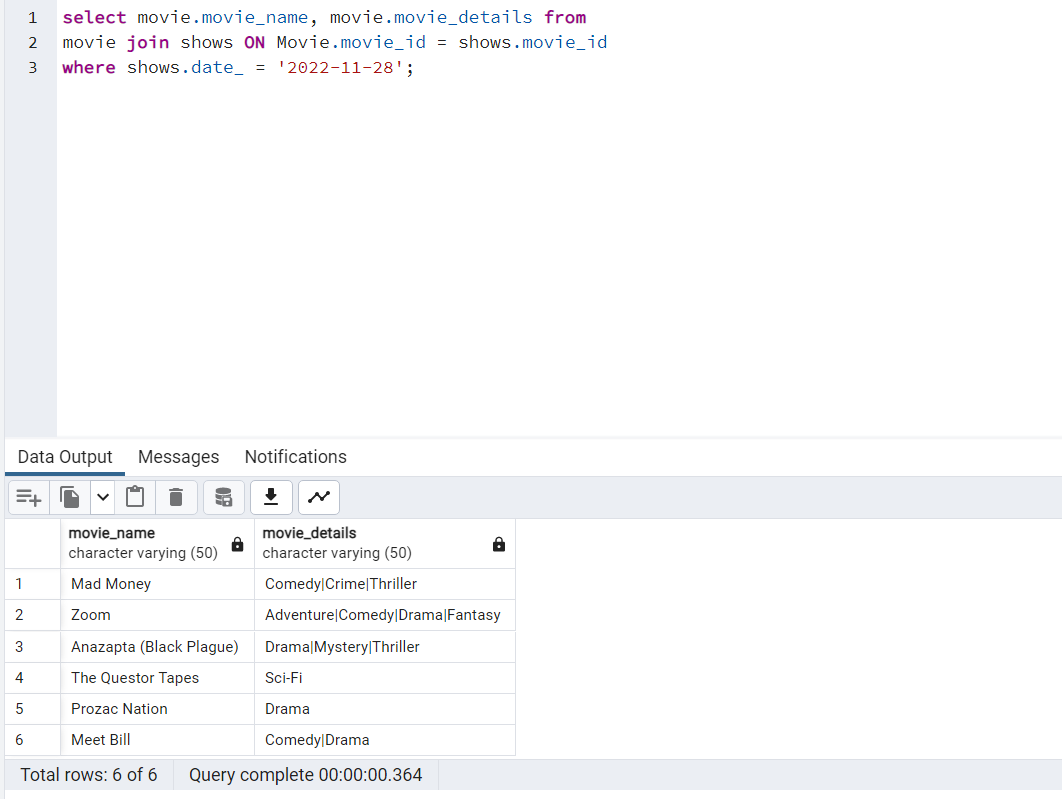
**29>  show all the movie available on 27th Nov 2022**

select movie.movie\_name, movie.movie\_details from

movie join shows ON Movie.movie\_id = shows.movie\_id

where shows.date\_ = '2022-11-28';

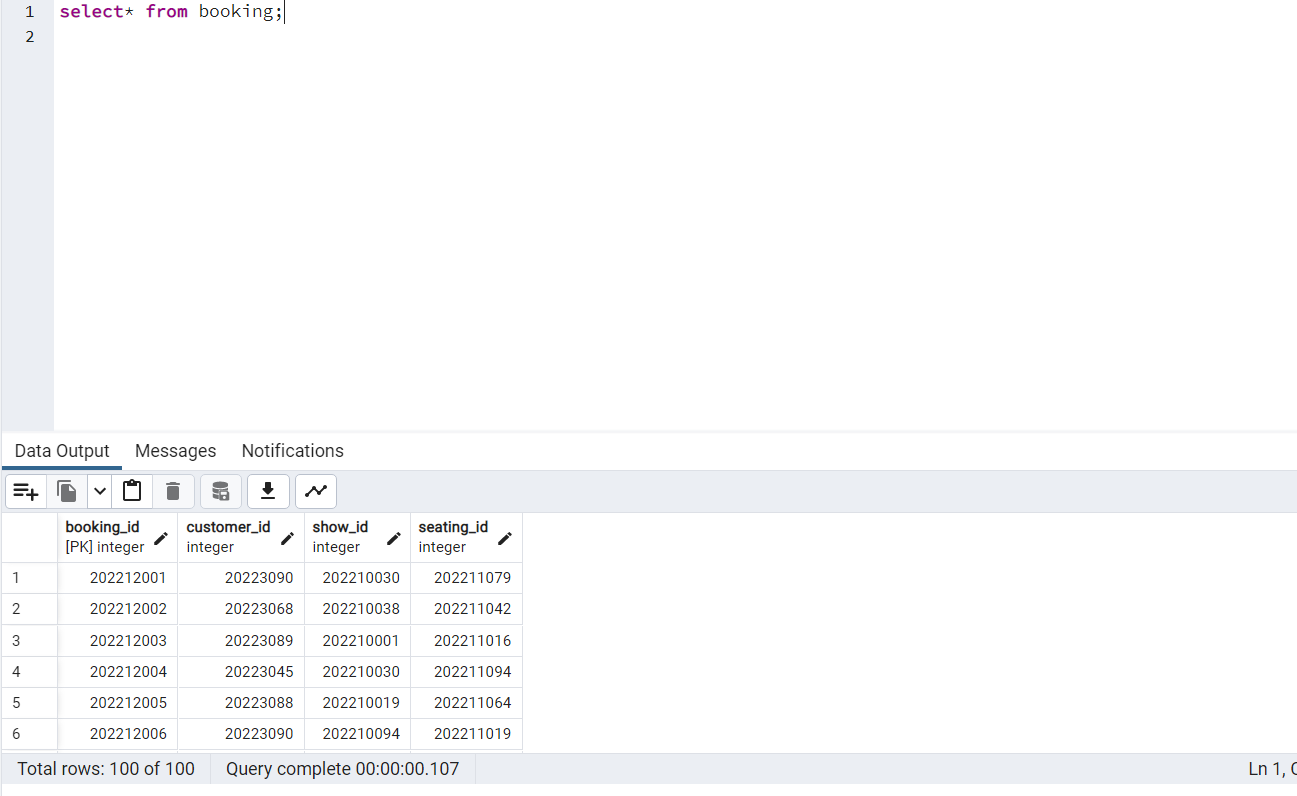
**Output:**

****

**30>  show all the booking details**

Select \* from bookings;

**Output:**

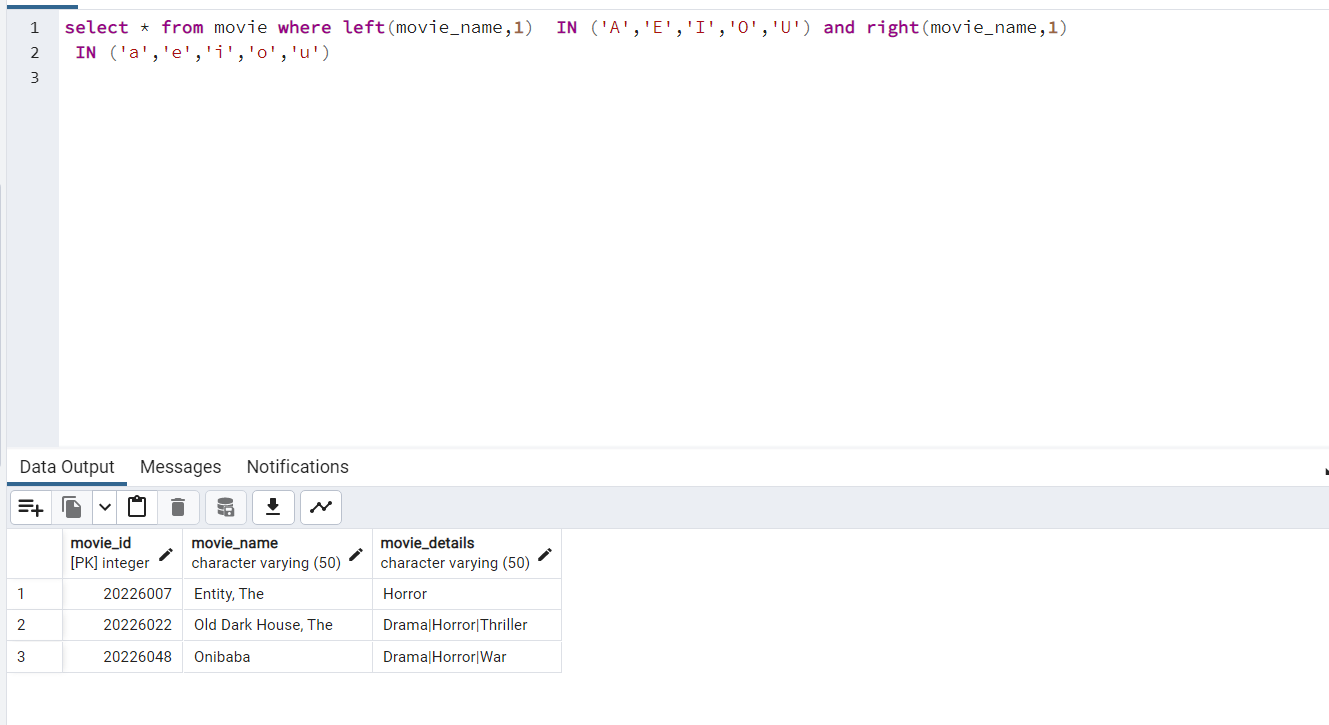


**31>show movie details whose name starts with vowels and ends with vowels**

select \* from movie where left(movie\_name,1)  IN ('A','E','I','O','U') and right(movie\_name,1)

 IN ('a','e','i','o','u')

**Output:**

****

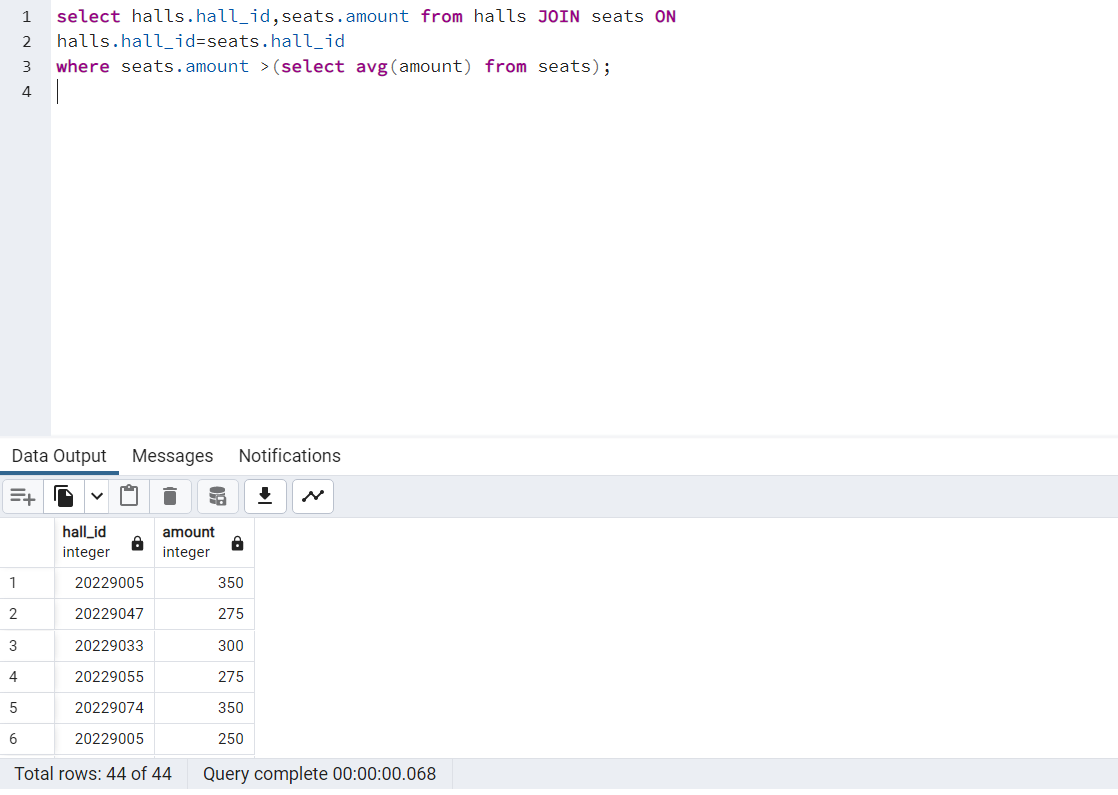
**32> Display halls detail where seats amount is more then average**

select halls.hall\_id,seats.amount from halls JOIN seats ON

halls.hall\_id=seats.hall\_id

where seats.amount >(select avg(amount) from seats);

**Output:**

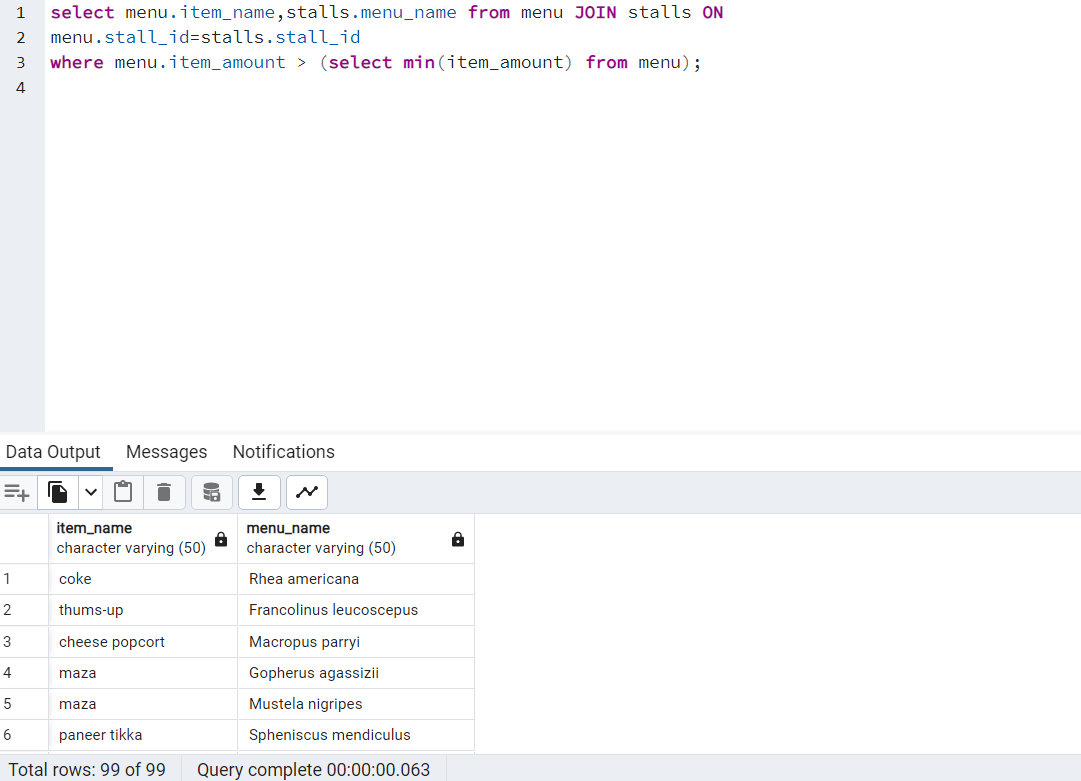


**33>Display menu name, item name where item\_amount is more then the min item price**

select menu.item\_name,stalls.menu\_name from menu JOIN stalls ON

menu.stall\_id=stalls.stall\_id

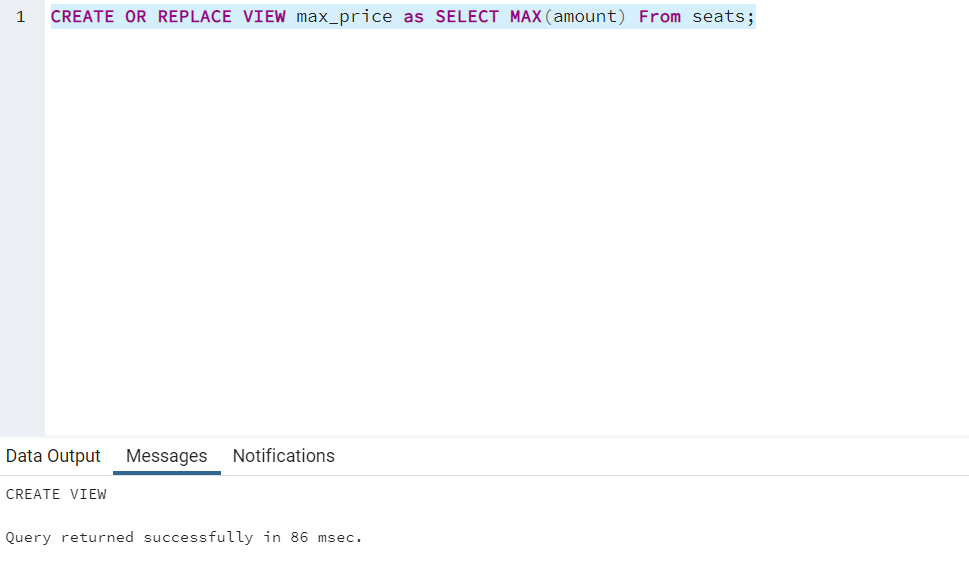
where menu.item\_amount > (select min(item\_amount) from menu);

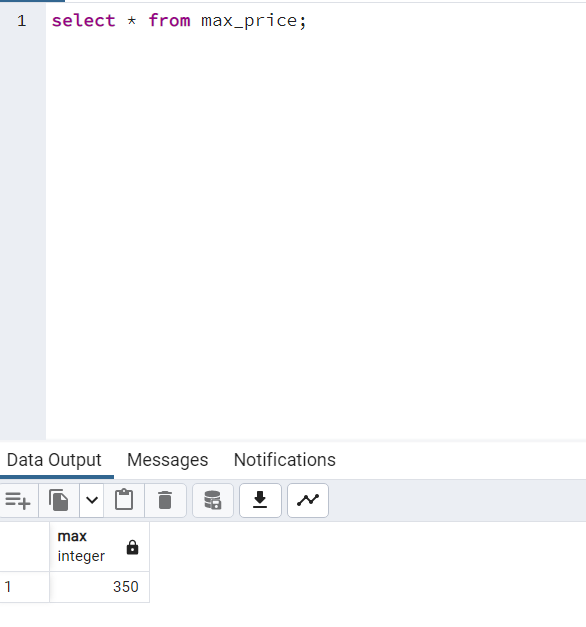


**34>Create a view on maximum price**

CREATE OR REPLACE VIEW max\_price as SELECT MAX(amount) From seats;

select \* from max\_price

****

****

**35> Display all the shows time where seats amount is more than then the minimum price**

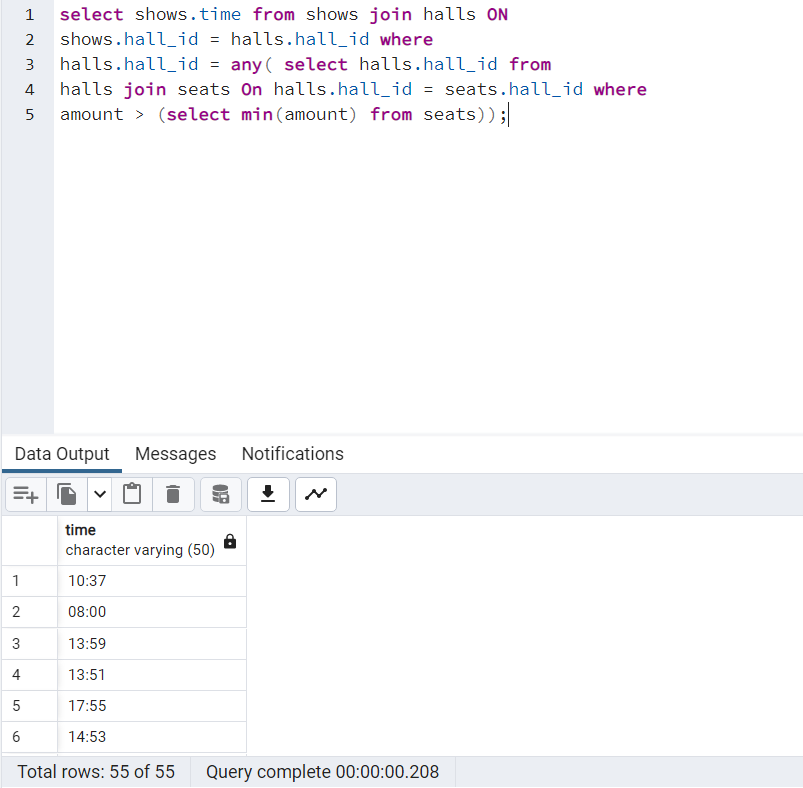
select shows.time from shows join halls ON

shows.hall\_id = halls.hall\_id where

halls.hall\_id = any( select halls.hall\_id from

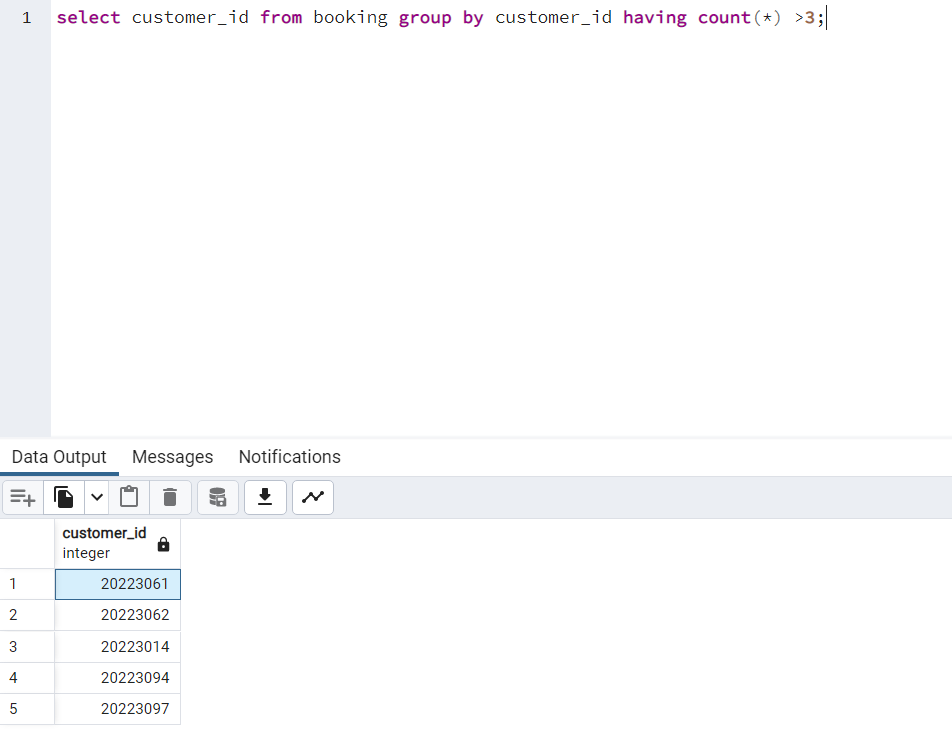
halls join seats On halls.hall\_id = seats.hall\_id where

amount > (select min(amount) from seats));



**36 > Display customer\_id who have booked more then 3 times**

select customer\_id from booking group by customer\_id having count(\*) >3;



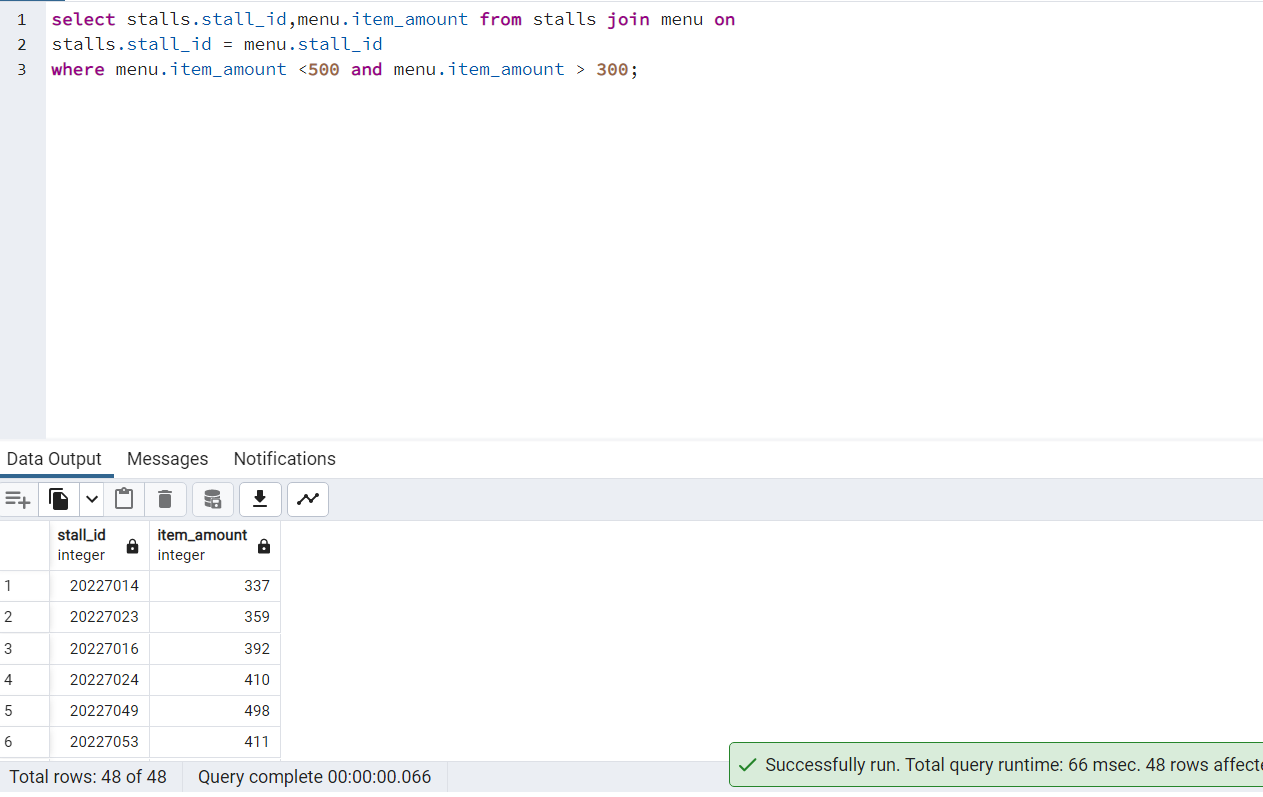
**37> Display stall details and menu item amount where item amount is more than Rs300 and less than Rs500**

select stalls.stall\_id,menu.item\_amount from stalls join menu on

stalls.stall\_id = menu.stall\_id

where menu.item\_amount <500 and menu.item\_amount > 300;

**Output:**

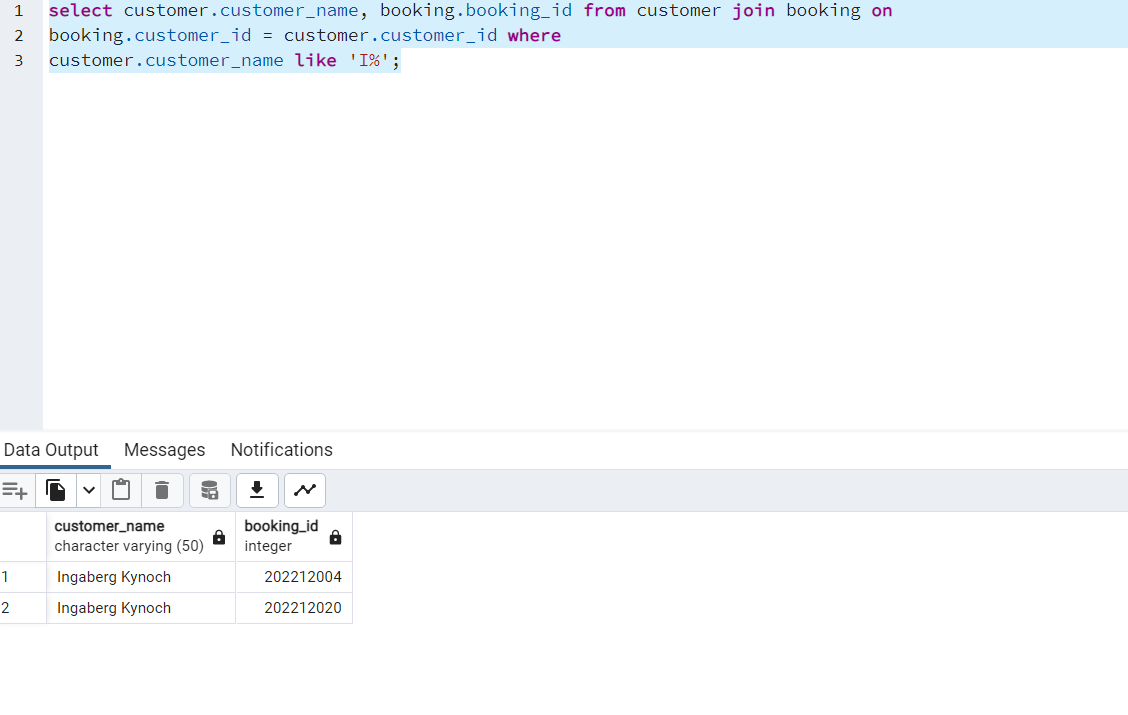
****

**38> DIsplay the booking\_Id details of customer whose name starts with ‘I’.**

select customer.customer\_name, booking.booking\_id from customer join booking on

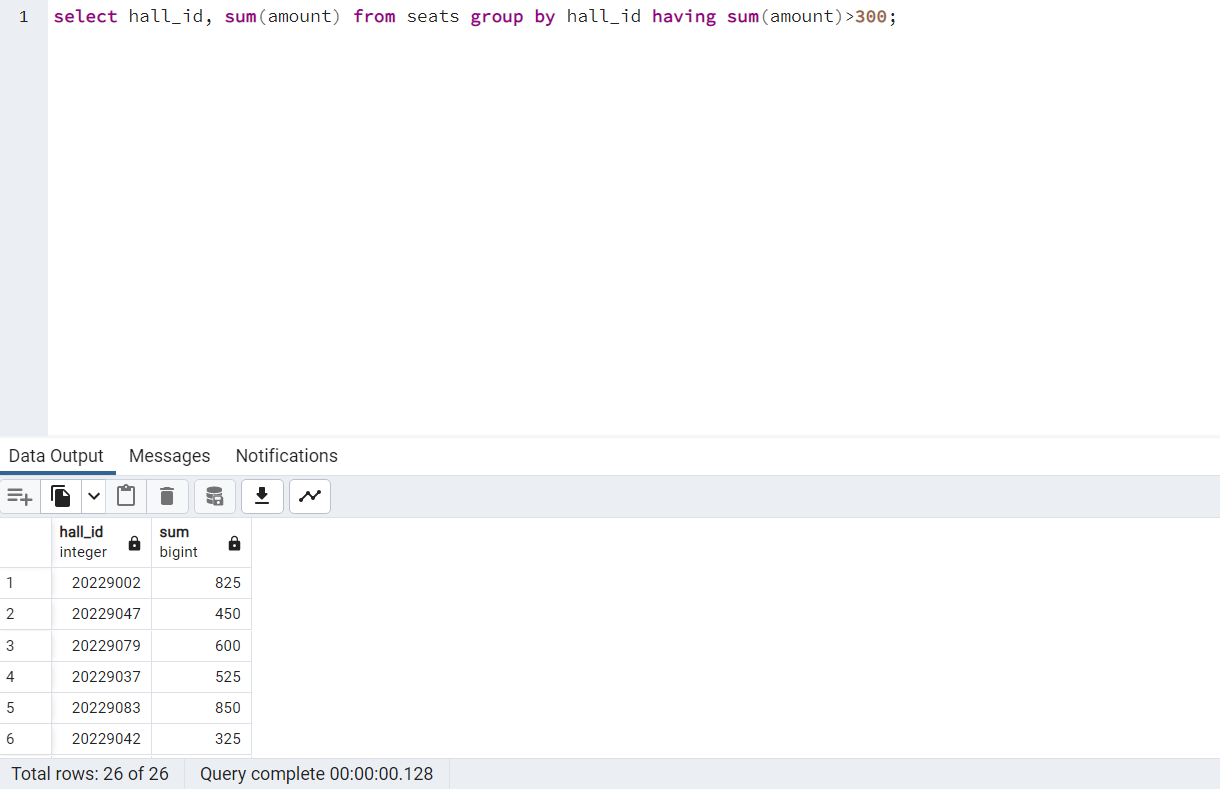
booking.customer\_id = customer.customer\_id where

customer.customer\_name like 'I%';



**39>Find the total sum of amount more then 300 of seat of individual halls**

select hall\_id, sum(amount) from seats group by hall\_id having sum(amount)>300;



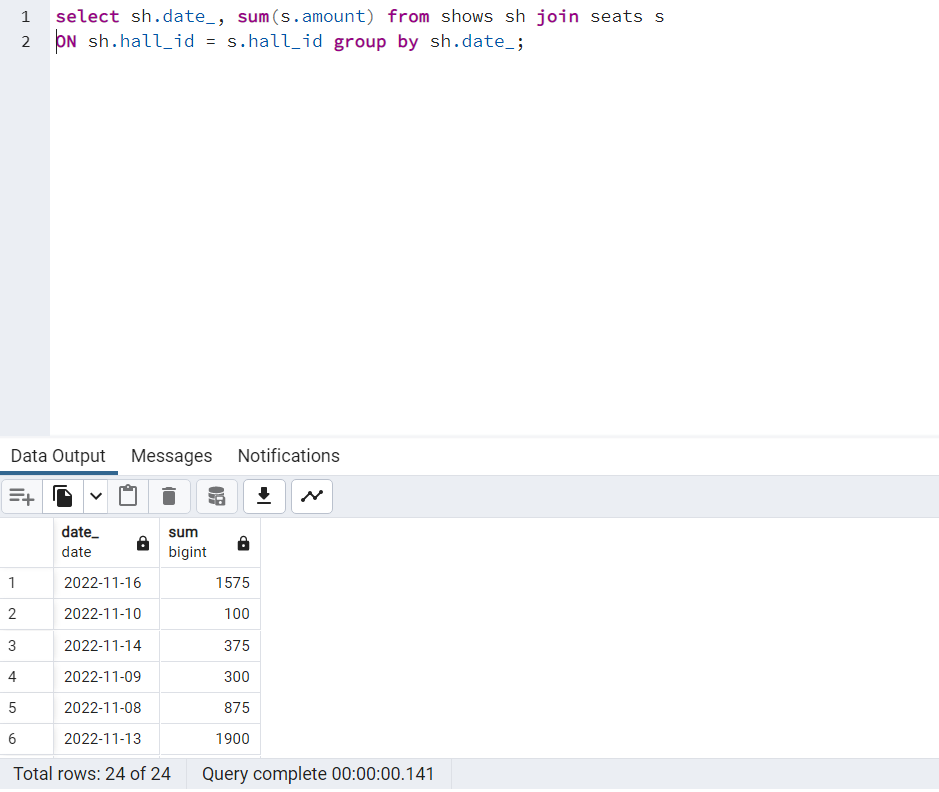
**‘**

**40>Display the sum of amount of seats on individual date**

select sh.date\_, sum(s.amount) from shows sh join seats s

ON sh.hall\_id = s.hall\_id group by sh.date\_;

**Output:**



### Bibliography

* Interview Link:
* [https://drive.google.com/file/d/11iUW8kJ2rWPi-OWG1BZnP3-](https://drive.google.com/file/d/11iUW8kJ2rWPi-OWG1BZnP3-ZPU8jBn-T/view?usp=sharing)

[ZPU8jBn-T/view?usp=sharing](https://drive.google.com/file/d/11iUW8kJ2rWPi-OWG1BZnP3-ZPU8jBn-T/view?usp=sharing)

* [https://medium.com/double-pointer/system-design-interview-how-todesign-bookmyshow-fandango-or-similar-movie-ticketing-](https://medium.com/double-pointer/system-design-interview-how-to-design-bookmyshow-fandango-or-similar-movie-ticketing-7140fb75468e)

[7140fb75468e](https://medium.com/double-pointer/system-design-interview-how-to-design-bookmyshow-fandango-or-similar-movie-ticketing-7140fb75468e)

* [https://medium.com/@narengowda/bookmyshow-system-designe268fefb56f5](https://medium.com/@narengowda/bookmyshow-system-design-e268fefb56f5)
* <https://www.ijtre.com/images/scripts/20160309101.pdf>
* <https://www.youtube.com/watch?v=lBAwJgoO3Ek>
* <https://in.bookmyshow.com/aboutus>