“Movie Ticket Booking System”

Project submitted to the

Dhirubhai Ambani Institute of Information and Communication Technology,

Gandhinagar

In Partial Fulfilment of the Requirement For The Course

Master of Science

Information Technology

Autumn 2022 – 2023

Submitted By

Team Member 01 : Darsh Doshi (202212061)

Team Member 02 : Viplove Jain (202212014)

(MSc(IT) 1st Sem)

INDEX

Acknowledgement

We Would Like To Take This Opportunity To Express Our Gratitude To **Prof. Minal Bhise** Our Course Instructor, **Parth Modi** Our Teaching Assistant For Their Support And Guidance In Completing Our Project On The Topic Movie Ticket Booking System.

And Also We Would Like To Thank Lab Assistant And The Lab Management Team Of DA-IICT For Providing Computer Laboratory Infrastructure.

SRS (System Requirements Specifications)

- **Description of case study** -

What is a movie ticket booking management system ?

It is a system which is created to solve the problem of almost every movie enthusiast , the problem of standing in queues to watch their favorite movie. This system eliminates queues and allows the audience to book a movie ticket virtually without compromising their comfort.

This type of system is created to book online tickets and complete management revolving around movie tickets bookings.

The users in general are required to create an ID before booking tickets and need to have a few things like a digital device (mobile, tablet, laptop etc) , a stable internet connection, online payment tool like upi , credit card , debit card , net banking , digital wallets etc.

Within the website users need to input movie name and location and using this information the system provides all the details regarding the request , after this user can proceed towards booking the tickets if users wish to do so.

Problem statement -

The whole process before this kind of system was developed used to be offline and lacked convenience , the main objective of this kind of system is to increase convenience for all the stakeholders involved be it movie theaters or customers and this offline to online shift was made to provide users with a good experience .

System perspective -

Online movie ticket booking management system is a replacement for manual , traditional and old existing systems where a lot of work is human intensive and needs a lot of physical effort.

Different stakeholders of this system are -

- **Movie theaters -**

System collaborates with them and the reach of these cinema halls increases and the workload at these halls are also reduced as digital tickets are easy to manage .

- **Customers -**

They enjoy the benefit of saving time and money as they don't need to stand in queues. They can choose seats beforehand , attractives offers are also present while booking online.

- **Management -**

They are managing all the processes in the backend. Any error or any miscommunications are handled by them , they form the backbone of the system and integrate everything .

- **Admin -**

The Admin department is the main developer of the system working for overall working of the system.

Basic features of this kind of system are -

- **Register-**

  To use the system, the user needs to sign up with some basic details like mobile number , email address, city , state etc.

- **Login-**

  After registration, the user can login with the set password and access the services.

- **To check availability of tickets -**

  Users can check according to the movie and theater how many total seats are available at any particular moment.

- **To check prices of movie tickets-**

  Prices are indicated according to the seats selected for a particular theater.

- **To compare different movie theaters on a single screen -**

  Users can check out every theater on the website .

- **To check upcoming movies-**

  Users can check the list of movies that will be screening in the near future along with its date of release and some other details like genre , language, duration of movie to decide if he/she wants to book this show.

- **To book ticket -**

  The user can book tickets according to his/her needs and get a ticket id which displays basic information about the movie like screen number , seat number etc and can be used for future processes.

- **To cancel tickets -**

  Users can cancel the tickets online but refund is according to terms and conditions of the cinema.

- **Online payment -**

  Users can pay online with resources like debit card, credit card , upi, net banking , digital wallets etc.

- **Online meals booking in advance -**

  Users can add additional meals or particular food items along with the quantity if required during the movie.

- **Reschedule tickets -**

  Some cinemas will allow users to reschedule the tickets.

Pros of online movie ticketing -

- Can be easily done with phone and internet access
- No need to stand in long queues to book tickets.
- The seats can be chosen in advance according to one's need which is in favor of the audience.
- Attractive offers are available while booking online.
- Tickets can be canceled online.

Cons of online ticketing -

- Requires a stable internet connection
- Online booking extra charges are needed to be paid by users.

Goals of the system -

- To provide a remote service to users that means users can book tickets from anywhere and at any time.
- To maintain proper records and statistics of the bookings.
- To increase profit.
- To promote movies over the internet.

❖ **Requirement Collection**

Input and outputs -

- **Background reading -**

1. **Inputs -**

- Understanding topic
- Understanding problem statement
- Existing system analysis

2. **Outputs -**

About movie ticket booking management system

It is a system which is created to solve the problem of almost every movie enthusiast , the problem of standing in queues to watch their favorite movie. This system eliminates queues and allows the audience to book a movie ticket virtually without compromising their comfort.

Basic features of this kind of system are -

- To check availability of tickets
- To check prices of movie tickets
- To compare different movie theaters on a single screen
- To book ticket
- To cancel ticket
- Online payment
- Online meals booking in advance
- Reschedule tickets

Merits of using such system -

- Can be easily done with phone and internet access
- No need to stand in long queues to book tickets.
- The seats can be chosen in advance according to one's need which is in favor of the audience.
- Attractive offers are available while booking online.
- Tickets can be canceled online.

Demerits of online ticketing -

- Requires a stable internet connection

Existing system study (BookMyShow) -

About BookMyShow-

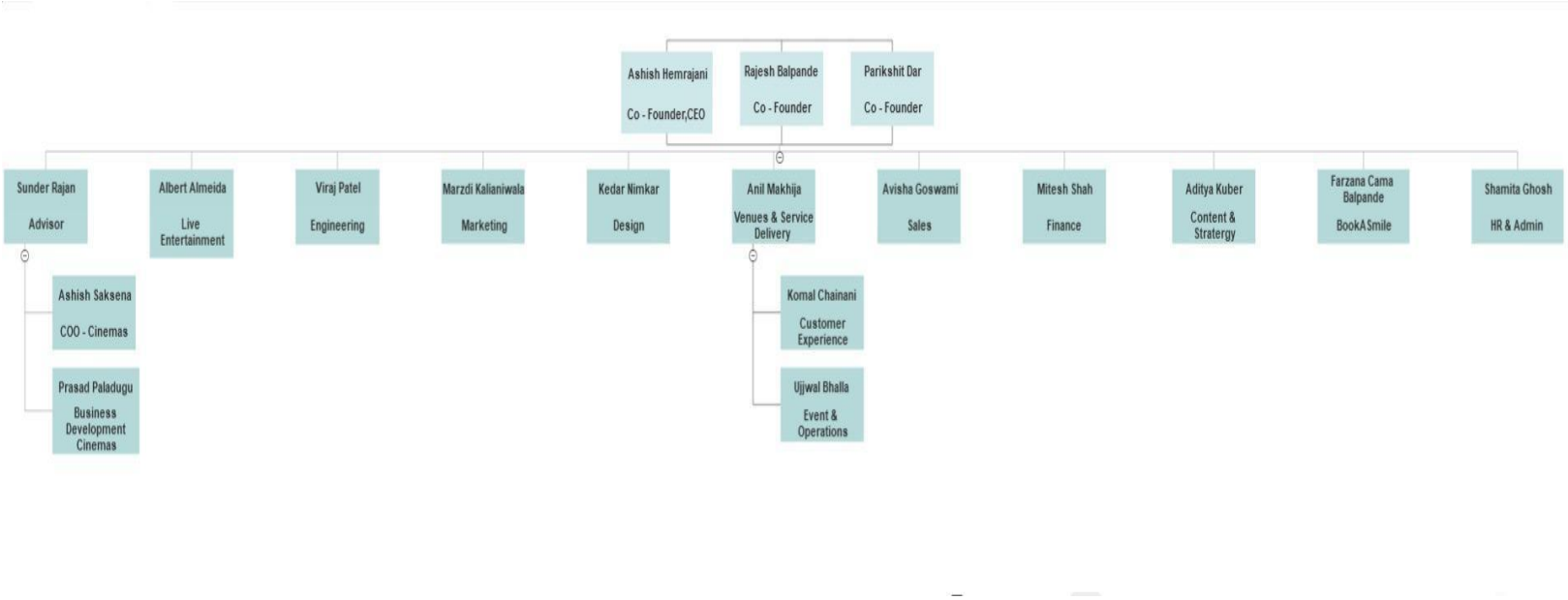
It is an online movie ticketing portal available in form of website and application , and can be used to avail services like movie ticket booking, tickets for different happenings like - sports match tickets, live shows tickets, and online virtual show tickets.

Financial report of bookmyshow-

Revenue of the company is consistently growing.

Organizational structure-

Founded by - Ashish Hemrajani , Rajesh Balpande , Parikshit Dar
CEO- Ashish Hemrajani



- **Interview -**

1. **Inputs -**

- Request for interview through email
- Preparing questions for interview

- **Interview plan -**

DBMS: Interview Plan
Project Name : movie ticket booking management system

Participants:
Interviewee - Employee BookMyshow(SDE)
Interviewer - Darsh doshi(202212061) - Student DA-IICT
Viplove jain(202212014) - Student DA-IICT

Date : 02 / 09 / 2022 Time : 7:30 PM
Duration : 15 minutes Place: virtual(zoom meeting)

Purpose of Interview :

- Understanding basic structure of application
- Day to day problems while working on this type of product.

Documents to be brought to the interview : -None-

- Questions prepared for interview -

1. What does it feel like to work on this kind of a product?
2. What are the challenges you face?
3. Which different types of user categories does **BookMyShow** Have?
4. Can you explain briefly what are the business constraints?
5. Any future challenges you see for your company?

- Conducting interview

2. Outputs -

- **Interview summary-**

Interview started with a brief introduction of all members involved. After the introduction we discussed the questions that we prepared before the interview session. Our interviewee opened about working at a bookmyshow and talked about the basic operations at the company.
Talking about the challenges sir said that it's a dynamically evolving industry and requires frequent updations from time to time.
Users of bookmyshow are distributed in two categories, one is the customers who are using their system for online movie ticketing etc , and one are the theaters which get registered on the system to sell their tickets.
Major Future challenge is growth of OTT platforms as a chunk of audience is also diverting towards ott platforms.

- Collected information relevant to real life project
- Data helpful for observation phase

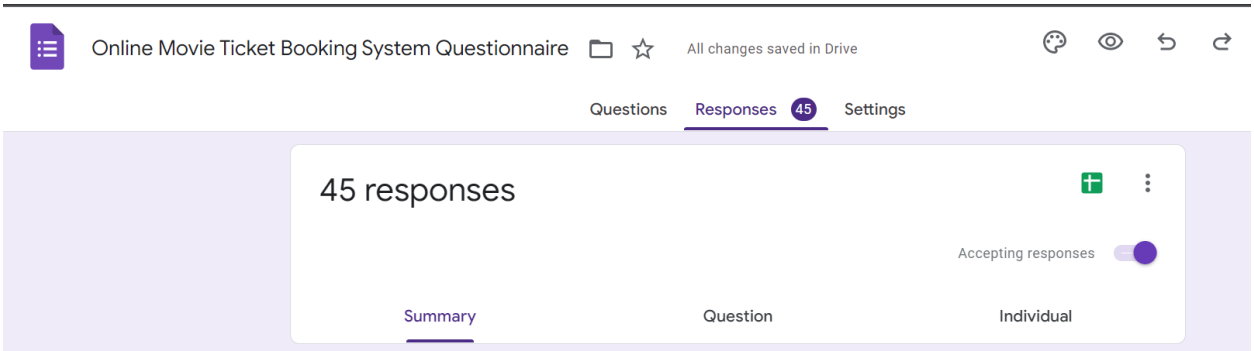
- **Questionnaire-**

1. **Inputs -**

- Preparation of questions
- Creation of google forms
- Circulation of google forms

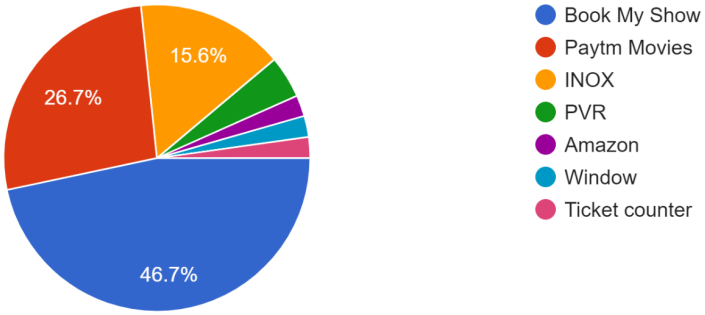
2. **Outputs -**

- Collected data from customers perspective
- Data helpful for observation phase



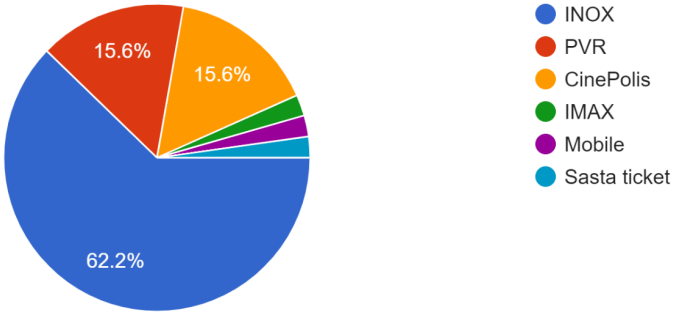
Q1- Which platform do you prefer to book movie tickets?

45 responses



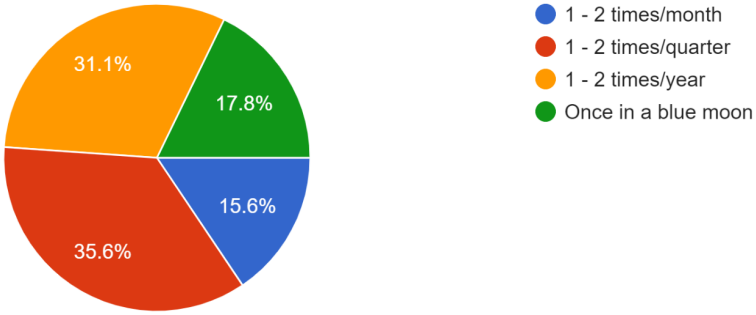
Q2- Which movie theatre do you prefer?

45 responses



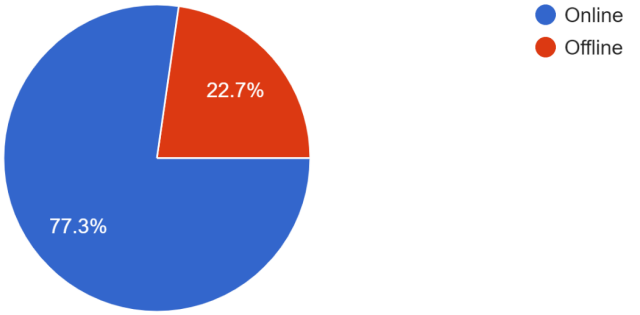
Q3- How often do you go for a movie?

45 responses



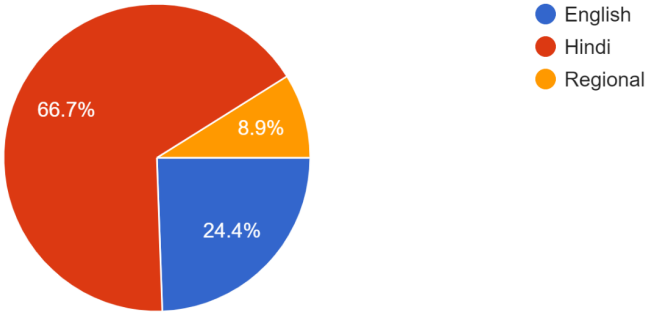
Q4- How do you book tickets?

44 responses



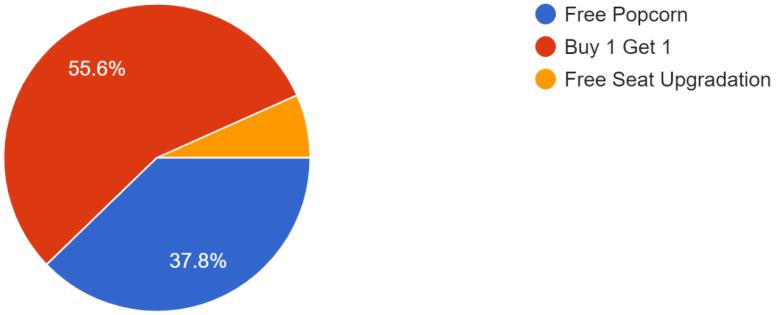
Q5- Which language movie do you generally prefer?

45 responses



Q6- Which free service would you prefer if new movie ticket booking system is launched?

45 responses



- **Observation -**

1. **Input -**

- Background reading
- Existing system study
- Interview
- Questionnaire

2. **Output -**

From existing system -

- Bookmyshow is currently the market leader for booking online movie tickets in india.
- Company's revenue is growing consistently but still it's not profitable and depends on fundings from investors.
- Company has multiple sources of income apart from movie ticketing like sports match tickets and stand up comedy show tickets etc.

From interview -

- These kind of system requires a lot of updation as they are dynamically growing industry
- Ott platforms are a barrier in capturing more customers

From questionnaire -

- Maximum number of people prefer bookmyshow (about 47%)to book online movie tickets followed by paytm movies, inox app etc.
- Maximum number of people prefer inox movie theater (about 62%) followed by PVR and cinepolis etc.
- In general people go for one - two movies in a quarter, or a year.
- More than 75% of people prefer online booking over offline.
- ⅔ people prefer a hindi movie over an english or a regional language movie.
- If a new app is launched and a free service is provided more than half (about 55%) people will prefer a free seat over a free popcorn or a seat upgrade.

❖ **Fact Finding Chart** -

Objective	Technique	Subject	Time
To find basic structure of system	Background reading	Existing system study	2 hours
To understand different components of system	Background reading	Existing system study/ Online study at various portals	2 hours
To understand core operations of system	Interview	Employee working at this kind of system	30 mins
To understand what records should be kept	Interview	Employee working at this kind of system	30 mins
To get users input about system	Questionnaire	Multiple users	2 hours
To follow up development of system understanding	Observation	From existing system study and questionnaire	2 hours

❖ **Requirements -**

Non-Functional Requirements :

- **Performance** : System should be able handle traffic of multiple users at any point of time and also any of the web browsers.
- **Security** : System should be secure enough to protect data of stakeholders involved
- **Maintainability** : System should be easy to maintain.
- **Browser Compatibility** : The project being web based requires compatibility with all browsers available.

Functional Requirements :

- New users can see the movie details but cannot book it until they are registered in the system.
- Registered users can change the password after logging into the system.
- See his/her current reservations on different movies along with the details.
- Able to choose the seats which are available for a certain class like silver, gold, platinum.
- Able to order snacks/meals.
- Give details about the credit card for the payment like account number, CVV number.
- The system should automatically show the fare for the corresponding movies and amount of money that is needed to be paid for selected seats.

To build a complete system with a following features -

- Client-Side
- Admin Panel
- Theater Panel
- Customer registration
- Upcoming movies, trailers, now showings
- Make bookings
- Payments
- Booking history
- Movie arrangements
- List upcoming movie details
- Show management
- Arrange movie screens and show timings
- Start and Stop running shows

Tools required to built this system is -

Back - End Tool : Postgresql

Why Postgre?
- Security
- Portability
- Maintainability
- Efficiency

❖ **User Categories -**

- Admin
- Management
- Theater partner
- Client (customer)
- Payment gateway partner

Features :

- Client-Side
- Admin Panel
- Theater Panel
- Customer registration
- Upcoming movies, trailers, now showings
- Make bookings
- Payments
- Booking history
- Movie arrangements
- List upcoming movie details
- Show management
- Arrange movie screens and show timings
- Start and Stop running shows

❖ **Privileges -**

Admin Privileges :

- Client side
- Admin panel
- Customer registration
- Upcoming movies, trailers, now showings
- Make bookings
- Payments
- Booking history
- Movie arrangements
- List upcoming movie details
- Show management
- Arrange movie screens and show timings
- Start and Stop running shows

Management Privileges :

- Customer registration
- Upcoming movies, trailers, now showings
- Make bookings
- Payments
- Booking history
- Movie arrangements
- List upcoming movie details
- Show management
- Arrange movie screens and show timings
- Start and Stop running shows

Theater Partner Privileges :

- Upcoming movies, trailers, now showings
- Own Theater Booking history
- Movie arrangements
- List upcoming movie details
- Show management
- Arrange movie screens and show timings
- Start and Stop running shows

Customer Privileges :

- Client side
- Customer registration
- Upcoming movies, trailers, now showings
- Make bookings
- Payments
- Own Booking history
- See upcoming movie details

❖ **Assumptions -**

- Admin is created in the system already.
- Roles and tasks are predefined.
- There is a limit of booking a movie. If the hall is houseful then the user cannot book the movie at that time.
- In general it has been assumed that the user has complete knowledge of the system that means the user is not a naive user. Any data entered by him/her will be valid.
- users have Internet accessibility.

❖ **Business Constraints -**

- User interface is only available in English, Option To change language is not available.
- Ott platforms hamper the revenue of booking systems
- Stable Internet access is required.
- Data might get corrupted in case of system crash or power failure.
- Users need to carry A digital ticket with him/her.

References -

www.prezi.com
www.bookmyshow.com
www.wikipedia.com

Design

❖ **Noun Analysis** -

Initial List Of Nouns			
Movie	Details	System	Payment
Ticket	Request	System	Users
Management	Problem	Features	Resource
System	Statement	System	Debit-Card
System	System	Registration	Credit-Card
Problem	Convenience	System	Upi
Enthusiast	Stakeholder	User	Net-Banking
Problem	Movie	Details	Digital-Wallets
Favorite	Theaters	Password	Meal
Movie	Customers	Services	Users
System	Users	Tickets	Meal
Queues	System	Users	Movie
Audience	Movie	Movie	Tickets
Movie	Ticket	Theater	Cinemas
Ticket	Management	Seats	Users
Comfort	System	Prices	Tickets
Type	Replacement	Movie	Movie
System	Manual	Tickets	Ticket
Tickets	System	Prices	Phone
Management	Human	Seats	Internet
Movie	Effort	Theater	Access
Ticket	Stakeholders	Movie	Queues
User	System	Theater	Tickets
Tickets	Movie	Screen	Seats
Users	Favor	Ticket	Goals

Theater	Audience	Users	System
Website	Offers	Tickets	Remote
Movies	Tickets	Tickets	Service
Users	Ticket	Total Seats	Screen No
Time	Duration	Language	Genre
Mobile No	Ticket No	Transaction Id	Food Item
Department	Show	Books	Pays
Halls	Money	Errors	Developer
Digital Tickets	Queue	Backbone	Internet
Customers	Seats	System	Theaters
Users	Offers	Admin	System
Users	Management	Admin	Cinema Halls
State	Backend	Password	Screening
Quantity	Date	Credit Card	Name
Displays	User	Debit Card	Location
Users	City	Net Banking	Information
Tickets	Age	Digital Wallets	Movies
Records	Laptop	Website	Digital Device
Statistics	Internet	Users	Mobile
Email	Payment	Movie	Tablet
Terms	Tool	Cinema	Tickets
Conditions	Upi	List	Refund
Movies	Profit	Internet	Charges

Accepted Nouns Table		
Candidate Entity Set	**Candidate Attribute**	**Candidate Relationship Set**
Movie	Type	Books
Ticket	Seat No	Pays
Admin	Price	Displays
Customer	Total Seats	Screening
Theaters	Screen No	
Payment	Name	
Meals	Location	
Seats	Date	
Screen	City	
Show	State	
	Time	
	Duration	
	Language	
	Genre	
	Age	
	Email	
	Password	
	Mobile No	
	Ticket No	
	Transaction Id	
	Food Item	
	Quantity	

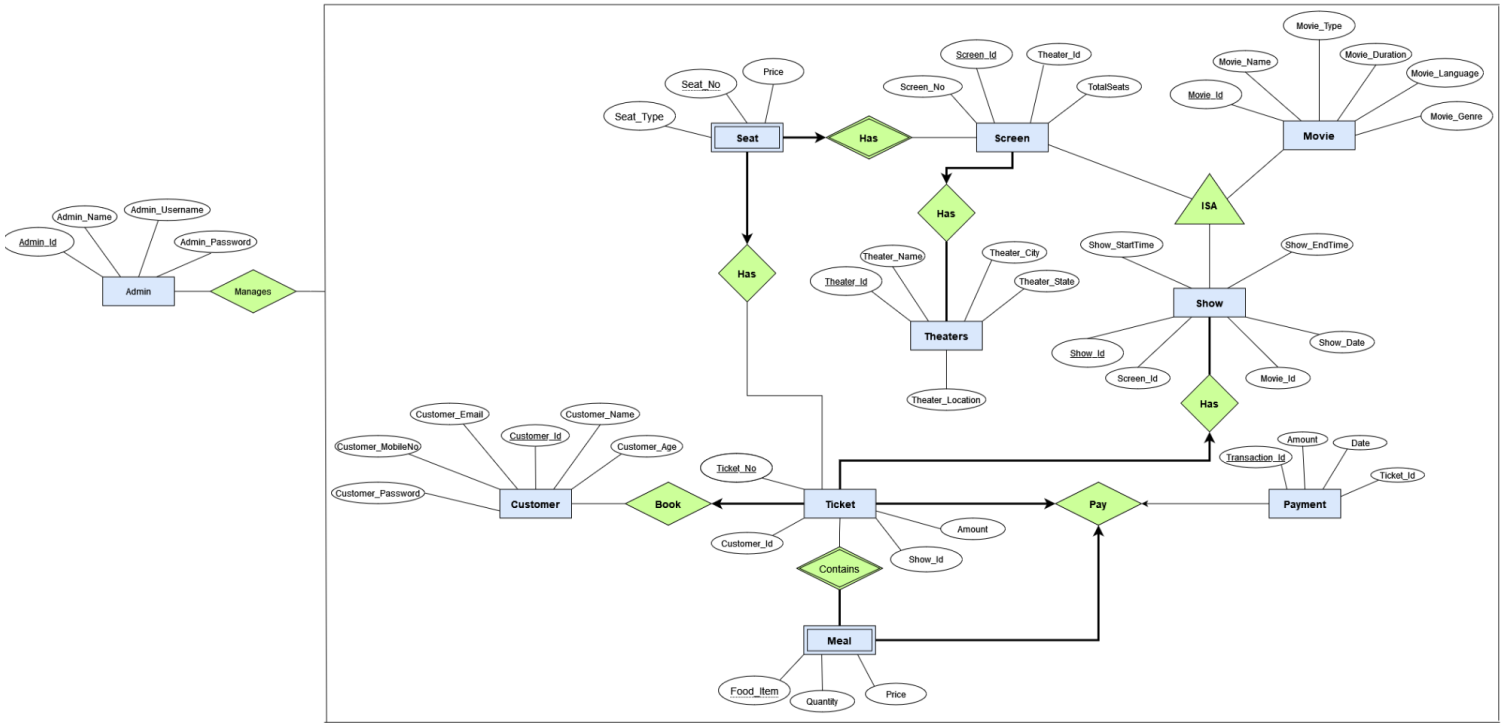
Rejected Nouns Table	
Nouns	**Reasons**
System	Irrelevant
Problem	Irrelevant
Enthusiast	Vague
Queue	Irrelevant
Audience	Duplicate
Comfort	Irrelevant
Statement	Irrelevant
Convenience	Irrelevant
Stakeholder	Irrelevant
Users	Duplicate
Replacement	Irrelevant
Manual	Vague
Human	Vague
Effort	Vague
Features	Irrelevant
Password	Irrelevant
Services	Irrelevant
Resource	Irrelevant
Credit Card	Irrelevant
Debit Card	Irrelevant
Net Banking	Irrelevant
Upi	Irrelevant
Digital Wallets	Irrelevant

Cinemas	Duplicate
Phone	Vague
Internet	Irrelevant
Queue	Vague
Digital Device	Irrelevant
Mobile	Irrelevant
Tablet	Irrelevant
Laptop	Irrelevant
Tool	Irrelevant
Website	Irrelevant
Information	Vague
Cinema Halls	Irrelevant
Halls	Irrelevant
Digital Tickets	Irrelevant
Money	Vague
Offers	Irrelevant
Backend	Vague
Errors	Irrelevant
Backbone	Irrelevant
Management	Irrelevant
Developer	Irrelevant
List	Vague
Refund	Irrelevant
Terms	Vague
Conditions	Vague

Favor	Vague
Charges	Irrelevant
Goals	Irrelevant
Remote	Irrelevant
Service	Irrelevant
Records	Irrelevant
Statistics	Vague
Profit	Vague
Request	Irrelevant
Registration	Irrelevant
Access	Vague
Favorite	Irrelevant
Details	Irrelevant
Department	Irrelevant

Strong Entity	**Weak Entity**	
Entities	**Entities**	**Identifier**
Theater	Seats	Seat No
Movie	Meal	Meal Name
Show		
Screen		
Customer		
Payment		
Ticket		
Admin		

E - R Diagram


Admin
Admin_Id
Admin_Name
Admin_Username
Admin_Password
Manages
Seat
Seat_No
Price
Seat_Type
Has
Screen
Screen_Id
Theater_Id
Screen_No
TotalSeats
Movie
Movie_Type
Movie_Name
Movie_Duration
Movie_Language
Movie_Id
Movie_Genre
ISA
Has
Has
Theaters
Theater_Name
Theater_City
Theater_Id
Theater_State
Theater_Location
Show
Show_StartTime
Show_EndTime
Show_Id
Show_Date
Screen_Id
Movie_Id
Has
Customer
Customer_Email
Customer_Name
Customer_Id
Customer_Age
Customer_MobileNo
Customer_Password
Book
Ticket
Ticket_No
Customer_Id
Amount
Show_Id
Contains
Meal
Food_Item
Quantity
Price
Pay
Payment
Amount
Date
Transaction_Id
Ticket_Id

E - R To Relational Mapping

Table 1 : Customer

Field Name	Data Type	Description	Constraints
Customer_Id	int	ID Of Customer	PRIMARY KEY
Customer_Name	varchar(100)	Name of The Customer	NOT NULL
Customer_Age	int(3)	Age of Customer	CHECK Customer_Age > 0
Customer_MobileNo	bigint	Mobile Number of Customer	NOT NULL
Customer_Email	varchar	Email Of Customer	UNIQUE KEY, NOT NULL
Customer_Password	varchar(MAX)	Password of Customer	NOT NULL

Tabel 2 : Theaters

Field Name	Data Type	Description	Constraints
Theater_Id	int	ID of Theater	PRIMARY KEY
Theater_Name	varchar(200)	Name of Theater	NOT NULL
Theater_Location	varchar(MAX)	Location of Theater	NOT NULL
Theater_City	varchar(100)	City of Theater	NOT NULL
Theater_State	varchar(100)	State of Theater	NOT NULL

Table 3 : Admin

Field Name	Data Type	Description	Constraints
Admin_Id	int	ID for Customer	PRIMARY KEY
Admin_Name	varchar(100)	Name of Admin	NOT NULL
Admin_Username	varchar(100)	Username of Admin	NOT NULL, UNIQUE
Admin_Password	varchar(MAX)	Password of Admin	NOT NULL

Table 4 :

Field Name	Data Type	Description	Constraints
Transaction_Id	bigint	Id of transaction	Primary Key
Amount	int(10)	Amount of booking	NOT NULL, CHECK Amount > 0
Date	date	Date of transaction	NOT NULL
Ticket_No	varchar(10)	ID of ticket	FOREIGN KEY

Table 5 : Screen

Field Name	Data Type	Description	Constraints
Screen_Id	int	Id of screen	PRIMARY KEY
Screen_No	int	Screen number	NOT NULL
TotalSeats	int(5)	Total number of seats	NOT NULL, CHECK TotalSeats > 0
Theater_Id	int	Id of theater	FOREIGN KEY

Table 6 : Ticket

Field Name	Data Type	Description	Constraints
Ticket_No	varchar(10)	Unique Number Of Ticket	PRIMARY KEY
Customer_Id	int	Id Of Customer	FOREIGN KEY
Seat_Id	int	Id Of Seat	FOREIGN KEY
Show_Id	int	Id Of Show	FOREIGN KEY
Amount	int(10)	Amount Of Ticket	NOT NULL, CHECK Amount > 0

Table 7 : Seat

Field Name	Data Type	Description	Constraints
Seat_Id	int	Unique Id of Seat	PRIMARY KEY
Seat_No	varchar(3)	Seat Number	NOT NULL
Seat_Type	varchar(50)	Type of Seat	NOT NULL
Price	bigint	Price of Seat	NOT NULL, CHECK Price > 0
Screen_Id	int	Id of Screen	FOREIGN KEY

Table 8 : Meal

Field Name	Data Type	Description	Constraints
Food_Item	varchar(MAX)	Name of Food Item	PRIMARY KEY
Quantity	int(3)	Quantity of Food Item	NOT NULL, CHECK Quantity > 0
Price	Double precision	Price of Food Item	NOT NULL, CHECK Price > 0
Ticket_No	varchar(10)	Id of Ticket	FOREIGN KEY

Table 9 : Movie

Field Name	Data Type	Description	Constraints
Movie_Id	int	Unique Id of Movie	PRIMARY KEY
Movie_Name	varchar(MAX)	Name of the Movie	NOT NULL
Movie_Type	varchar(100)	Type of the Movie	NOT NULL
Movie_Duration	int	Duration of the Movie	NOT NULL, CHECK DURATION>0
Movie_Language	varchar(100)	Language of the Movie	NOT NULL, DEFAULT = ‘HINDI’
Movie_Genre	varchar(100)	Genre of the Movie	NOT NULL

Table 10 : Show

Field Name	Data Type	Description	Constraints
Show_Id	int	Unique Id of Show	PRIMARY KEY
Screen_Id	int	Id of Screen	FOREIGN KEY
Movie_Id	int	Id of Movie	FOREIGN KEY
Show_Date	date	Date of the Show	NOT NULL
Show_StartTime	time	Show Start Time	NOT NULL
Show_EndTime	time	Show End Time	NOT NULL

DDL Statements

Schema :

```
CREATE SCHEMA MovieTicket_DB;
SET SEARCH_PATH TO MovieTicket_DB;
```

Table 1 : Customer

```
CREATE TABLE IF NOT EXISTS movieticket_db."Customer"
(
Customer_Id int,
Customer_Name varchar(100) NOT NULL,
Customer_Age int CHECK (Customer_Age > 0),
Customer_MobileNo bigint NOT NULL,
Customer_Email varchar NOT NULL UNIQUE,
Customer_Password varchar NOT NULL,
CONSTRAINT Customer_Pkey PRIMARY KEY (Customer_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;
ALTER TABLE IF EXISTS movieticket_db."Customer"
OWNER to postgres;
```

Table 2 : Theater

```
CREATE TABLE IF NOT EXISTS movieticket_db."Theater"
(
Theater_Id int,
Theater_Name varchar NOT NULL,
Theater_Location varchar NOT NULL,
Theater_City varchar NOT NULL,
Theater_State varchar NOT NULL,
CONSTRAINT Theater_Pkey PRIMARY KEY (Theater_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Theater"
OWNER to postgres;
```

Table 3 : Admin

```
CREATE TABLE IF NOT EXISTS movieticket_db."Admin"
(
Admin_Id int,
Admin_Name varchar(100) NOT NULL,
Admin_Username varchar(100) NOT NULL UNIQUE,
Admin_Password varchar NOT NULL,
CONSTRAINT Admin_Pkey PRIMARY KEY (Admin_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Admin"
OWNER to postgres;
```

Table 4 : Payment

```
CREATE TABLE IF NOT EXISTS movieticket_db."Payment"
(
Transaction_Id bigint,
Amount int NOT NULL CHECK (Amount>0),
Date date NOT NULL,
Ticket_No varchar(10),
CONSTRAINT Payment_Pkey PRIMARY KEY (Transaction_Id),
CONSTRAINT Ticket_Fkey FOREIGN KEY (Ticket_No)
REFERENCES movieticket_db."Ticket" (Ticket_No)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Payment"
OWNER to postgres;
```

Table 5 : Screen

```
CREATE TABLE IF NOT EXISTS movieticket_db."Screen"
(
Screen_Id int,
Screen_No int NOT NULL,
TotalSeats int NOT NULL CHECK (TotalSeats>0),
Theater_Id int,
CONSTRAINT Screen_Pkey PRIMARY KEY (Screen_Id),
CONSTRAINT Theater_Fkey FOREIGN KEY (Theater_Id)
REFERENCES movieticket_db."Theater"(Theater_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Screen"
OWNER to postgres;
```

Table 6 : Ticket

```
CREATE TABLE IF NOT EXISTS movieticket_db."Ticket"
(
Ticket_No varchar(10),
Customer_Id int,
Seat_Id int,
Show_Id int,
Amount bigint NOT NULL CHECK (Amount>0),
CONSTRAINT Ticket_Pkey PRIMARY KEY (Ticket_No),
CONSTRAINT Customer_Fkey FOREIGN KEY (Customer_Id)
REFERENCES movieticket_db."Customer" (Customer_Id),
CONSTRAINT Seat_Fkey FOREIGN KEY (Seat_Id)
REFERENCES movieticket_db."Seat" (Seat_Id),
CONSTRAINT Show_Fkey FOREIGN KEY (Show_Id)
REFERENCES movieticket_db."Show" (Show_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Ticket"
OWNER to postgres;
```

Table 7 : Seat

```
CREATE TABLE IF NOT EXISTS movieticket_db."Seat"
(
Seat_Id int,
Seat_No varchar(3) NOT NULL,
Seat_Type varchar(50) NOT NULL,
Price int NOT NULL CHECK (Price>0),
Screen_Id int,
CONSTRAINT Seat_Pkey PRIMARY KEY (Seat_Id),
CONSTRAINT Screen_Fkey FOREIGN KEY (Screen_Id)
REFERENCES movieticket_db."Screen" (Screen_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Seat"
OWNER to postgres;
```

Table 8 : Meal

```
CREATE TABLE IF NOT EXISTS movieticket_db."Meal"
(
Food_Item varchar,
Quantity int NOT NULL CHECK (Quantity>0),
Price double precision NOT NULL CHECK (Price>0),
Ticket_No varchar(10),
CONSTRAINT Meal_Pkey PRIMARY KEY (Food_Item,Ticket_No),
CONSTRAINT Ticket_Fkey FOREIGN KEY (Ticket_No)
REFERENCES movieticket_db."Ticket" (Ticket_No)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Meal"
OWNER to postgres;
```

Table 9 : Movie

```
CREATE TABLE IF NOT EXISTS movieticket_db."Movie"
(
Movie_Id int,
Movie_Name varchar NOT NULL,
Movie_Type varchar(100) NOT NULL,
Movie_Duration int NOT NULL CHECK (Movie_Duration>0),
Movie_Language varchar(100) NOT NULL DEFAULT 'HINDI',
Movie_Genre varchar(100) NOT NULL,
CONSTRAINT Movie_Pkey PRIMARY KEY (Movie_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Movie"
OWNER to postgres;
```

Table 10 : Show

```
CREATE TABLE IF NOT EXISTS movieticket_db."Show"
(
Show_Id int,
Screen_Id int,
Movie_Id int,
Show_Date date NOT NULL,
Show_StartTime time without time zone NOT NULL,
Show_EndTime time without time zone NOT NULL,
CONSTRAINT Show_Pkey PRIMARY KEY (Show_Id),
CONSTRAINT Screen_Fkey FOREIGN KEY (Screen_Id)
REFERENCES movieticket_db."Screen" (Screen_Id),
CONSTRAINT Movie_Fkey FOREIGN KEY (Movie_Id)
REFERENCES movieticket_db."Movie" (Movie_Id)
)
WITH
(
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS movieticket_db."Show"
OWNER to postgres;
```

Detail Of Populating Data In Tables

Process :

- Generate a CSV file using generatedata.com
- Copy data from csv to database table using following statement

  COPY
  movieticket_db.movieticket_db."Customer"(customer_id,customer_name,customer_age,
  customer_mobileno,customer_email,customer_password)
  FROM 'D:/Movie DB/Customer.csv' DELIMITER ',' CSV HEADER;

- Repeat this process for every relation

Snapshot of populated tables :

1. Admin

	admin_id [PK] integer	admin_name character varying (100)	admin_username character varying (100)	admin_password character varying
1	101	darsh	ceodarsh	abcd123
2	102	viplove	cfoviplove	abc1234
3	103	prayag	cmoprayag	qwe1234
4	104	asish	ctoasish	qwe12
5	105	karan	mdkaran	zz123

2. Customer

	customer_id [PK] integer	customer_name character varying (100)	customer_age integer	customer_mobileno bigint	customer_email character varying	customer_password character varying
1	1	Odessa Benson	28	758563172	dolor@icloud.couk	SFV55GBK5DP
2	2	Leandra Bennett	8	908572754	mollis.phasellus@google.com	BYH41DKR7QM
3	3	Naomi Banks	79	534645184	nibh.phasellus@outlook.org	VTY56LHO0KM
4	4	Xandra Mathis	26	646010076	et.magnis.dis@aol.com	VTY35HIW5BT
5	5	Rahim Good	62	824124931	nec.mauris.blandit@aol.ca	MNL96WCH8GW
6	6	Chanda Crane	53	748743778	vitae.semper.egestas@protonmail.n...	QLJ76FKR4KK
7	7	Kelly Hernandez	40	436584332	sit@outlook.com	YNP86ONL4KW
8	8	Clinton Alvarado	23	589876774	duis.volutpat.nunc@hotmail.com	OXJ52HBT1LU
9	9	Daquan Franklin	78	402447951	praesent.luctus.curabitur@hotmail.ca	UJM38QON4JS
10	10	Flavia Strong	70	748616820	dignissim@yahoo.com	ZCN67GWK8RE
11	11	Wayne Duran	13	351827478	non.nisi@icloud.com	CAH77CQH5NJ
12	12	Richard Hunt	53	234598452	neque.nullam@protonmail.couk	TRE36GHX4OJ
13	13	Karly Buckley	55	272863233	lacus.aliquam@yahoo.net	HRF61COI6VU
14	14	Elton Dodson	60	351096363	quisque.ac.libero@aol.org	IVL72ONC9BN
15	15	Tanek Knowles	19	927261331	etiam@protonmail.edu	MJA70SHJ4AM

Total rows: 100 of 100 Query complete 00:00:00.330 Ln 1, Col 1

3. Meal

	food_item [PK] character varying	quantity integer	price double precision	ticket_no [PK] character varying (10)
1	butter popcorn	5	985.6	A106
2	butter popcorn	4	30.4	A116
3	butter popcorn	4	1387.2	A122
4	butter popcorn	5	288.8	A125
5	butter popcorn	1	1232	A138
6	butter popcorn	5	1387.2	A170
7	butter popcorn	4	63.2	A176
8	butter popcorn	3	93.6	A184
9	caramel popcorn	1	1522.4	A105
10	caramel popcorn	1	432	A107
11	caramel popcorn	2	792.8	A164
12	caramel popcorn	4	418.4	A180
13	caramel popcorn	2	1053.6	A182
14	caramel popcorn	1	722.4	A189
15	cheese nachos	2	636	A128

Total rows: 100 of 100 Query complete 00:00:00.136 Ln 1, Col 1

4. Movie

	movie_id [PK] integer	movie_name character varying	movie_type character varying (100)	movie_duration integer	movie_language character varying (100)	movie_genre character varying (100)
1	1001	Stage Fright (Deliria)	2D	112	English	Horror
2	1002	Thumbelina	3D	144	English	Animation
3	1003	Moonlight and Cactus	3D	197	Armenian	Comedy
4	1004	Me and Orson Welles	2D	199	Bosnian	Drama
5	1005	Fist of Fury (Chinese Connection, The...	3D	158	Yiddish	Action
6	1006	Berlin 36	3D	164	Polish	Drama
7	1007	Age of the Earth, The (A Idade da Terra)	2D	108	Aymara	No Genre
8	1008	Sumo Do, Sumo Don't (Shiko funjatta)	3D	191	Afrikaans	Comedy
9	1009	Humanit??, L'	3D	96	French	Crime
10	1010	The Land Before Time IX: Journey to t...	2D	153	English	Adventure
11	1011	Oh Happy Day	3D	182	English	Comedy
12	1012	Eighth Day, The (Den ??ttonde dagen)	2D	145	Hindi	Children
13	1013	Cherry Falls	3D	93	Lithuanian	Comedy
14	1014	Better Than Chocolate	2D	119	Hindi	Comedy

Total rows: 100 of 100 Query complete 00:00:00.139 Ln 1, Col 1

5. Payment

	transaction_id [PK] bigint	amount integer	date date	ticket_no character varying (10)
1	107490	500	2022-08-07	A137
2	110129	500	2021-12-25	A138
3	128580	650	2022-03-15	A163
4	142618	500	2022-02-15	A118
5	144529	650	2022-02-10	A146
6	156041	500	2022-04-04	A148
7	171464	500	2022-05-21	A105
8	172974	500	2022-05-17	A196
9	173533	650	2022-08-20	A136
10	198365	500	2022-06-18	A131
11	210931	650	2022-10-17	A112
12	236199	650	2021-11-26	A133
13	242342	650	2022-08-04	A120
14	248150	650	2022-05-25	A111
15	257747	500	2022-10-18	A125

Total rows: 100 of 100 Query complete 00:00:00.153 Ln 1, Col 1

6. Screen

	screen_id [PK] integer	screen_no integer	totalseats integer	theater_id integer
1	501	1	391	1
2	502	2	262	1
3	503	3	320	1
4	504	4	382	1
5	505	1	446	2
6	506	2	395	2
7	507	3	205	2
8	508	4	284	2
9	509	5	186	2
10	510	6	438	2
11	511	1	239	3
12	512	2	174	3
13	513	3	407	3
14	514	1	288	4
15	515	2	316	4

Total rows: 167 of 167 Query complete 00:00:00.135 Ln 1, Col 1

7. Seat

	seat_id [PK] integer	seat_no character varying (3)	seat_type character varying (50)	price integer	screen_id integer
1	1	A1	Recliner	650	501
2	2	A2	Recliner	650	501
3	3	A3	Recliner	650	501
4	4	A4	Recliner	650	501
5	5	A5	Recliner	650	501
6	6	A6	Recliner	650	501
7	7	A7	Recliner	650	501
8	8	A8	Recliner	650	501
9	9	A9	Recliner	650	501
10	10	A10	Recliner	650	501
11	11	A11	Recliner	650	501
12	12	A12	Recliner	650	501
13	13	A13	Recliner	650	501
14	14	A14	Recliner	650	501
15	15	A15	Recliner	650	501

Total rows: 1000 of 4129 | Query complete 00:00:00.713 | Ln 1, Col 1

8. Show

	show_id [PK] integer	screen_id integer	movie_id integer	show_date date	show_starttime time without time zone	show_endtime time without time zone
1	1	501	1001	2022-03-26	13:28:00	22:45:00
2	2	502	1002	2022-03-14	15:22:00	17:03:00
3	3	503	1003	2022-04-26	11:24:00	11:23:00
4	4	504	1004	2022-08-22	10:10:00	17:37:00
5	5	505	1005	2022-08-30	09:00:00	16:08:00
6	6	506	1006	2022-08-19	09:13:00	13:51:00
7	7	507	1007	2022-01-09	18:26:00	14:34:00
8	8	508	1008	2022-10-16	12:08:00	11:45:00
9	9	509	1009	2022-07-30	19:07:00	12:00:00
10	10	510	1010	2022-02-01	14:55:00	18:19:00
11	11	511	1011	2022-04-22	16:23:00	14:28:00
12	12	512	1012	2022-07-31	16:53:00	19:20:00
13	13	513	1013	2022-09-22	19:57:00	21:44:00
14	14	514	1014	2022-03-28	09:46:00	14:02:00
15	15	515	1015	2022-08-07	19:37:00	19:55:00

Total rows: 167 of 167 | Query complete 00:00:00.116 | Ln 1, Col 1

9. Theater

	theater_id [PK] integer	theater_name character varying	theater_location character varying	theater_city character varying	theater_state character varying
1	1	Shelby Molina	7736 Fringilla Av.	Eluru	Uttar Pradesh
2	2	Chancellor Serrano	P.O. Box 692, 8266 Aliquam Street	Thrissur	Tripura
3	3	Isabella Workman	Ap #806-3847 Adipiscing Street	Tirupati	Meghalaya
4	4	Kadeem Garrison	Ap #573-2191 Feugiat Avenue	Eluru	Odisha
5	5	Sarah Pate	619-4008 Eget, Rd.	Ratlam	Tripura
6	6	Cleo Blankenship	Ap #263-5641 Magna St.	Saharanpur	Uttar Pradesh
7	7	Hilary Small	P.O. Box 758, 4020 Orci, Rd.	Nellore	Jammu and Kash…
8	8	Ivory Downs	641-7801 Natoque Street	Varanasi	Himachal Pradesh
9	9	Oscar Holland	Ap #251-9026 Dictum. Road	Bharatpur	Telangana
10	10	Lucas Drake	Ap #100-7483 Duis Street	Loni	Odisha
11	11	Steel Dyer	9759 Faucibus Av.	Gandhidham	Tripura
12	12	September Whitehead	Ap #929-9753 Pharetra Ave	Indore	Maharashtra
13	13	Laith Moss	Ap #647-5710 A, Rd.	Guna	Goa
14	14	Sylvia Ryan	3831 Ullamcorper Rd.	Secunderabad	Haryana
15	15	Duncan Estes	942-7547 Orci. Avenue	Ranchi	Haryana

Total rows: 80 of 80 Query complete 00:00:00.660 Ln 1, Col 1

10. Ticket

	ticket_no [PK] character varying (10)	customer_id integer	seat_id integer	show_id integer	amount bigint
1	A101	1	1022	4	650
2	A102	2	2128	6	500
3	A103	3	2812	9	500
4	A104	4	2651	8	500
5	A105	5	2865	9	500
6	A106	6	2284	7	650
7	A107	7	374	1	350
8	A108	8	152	1	500
9	A109	9	17	1	650
10	A110	10	2328	7	500
11	A111	11	2947	10	650
12	A112	12	1901	6	650
13	A113	13	973	3	500
14	A114	14	3852	13	500
15	A115	15	2490	8	650

Total rows: 100 of 100 Query complete 00:00:00.158 Ln 1, Col 1

Implementation

❖ English Queries

Simple Queries

1. Show Customer Details
2. Show Payment Details Whose Amount Is Greater Than 500
3. Get Unique List Of Theaters In Jamnagar
4. Show Movie Details Of Drama Genre
5. Display Today's 14 March 2022
6. Show Admin Details
7. Write A Query To Display Unique Food Item Using The Alias Name As Delicacies
8. Display Show Where Time Is Between 3 PM - 11 PM
9. Count Total Customer
10. Sort Admin By Name
11. Count The Number Of Shows On 31 July 2022 Which Were Screened On Screen Id 512
12. Display List Of Theaters Excluding The Ones Present In Maharashtra And Madhya Pradesh.
13. Find Out The Average Meal Price.
14. Print The Count Of Different Types Of Seats.
15. Display The Name And Email Of A Customer Who Has 'y' In Their Names.
16. Display List Of Movies Which Have Genre Comedy And Type Not 2D.
17. Select Details Of Top 2 Highest Price Tickets.
18. Display The Total Amount Received In The January Month Of 2022.
19. Write A Query To Fetch The Details Of The Last Ticket Sold.
20. Print The Details Of Movies Where Genre Is Comedy Or Adventure
21. Display Maximum,Minimum And Average Of Transaction Amount.
22. Count The Number Of Movies Genre Wise.
23. Write A Query To Fetch Only Odd Ids From Customer Table
24. Display Meal With Average Price

Complex Queries

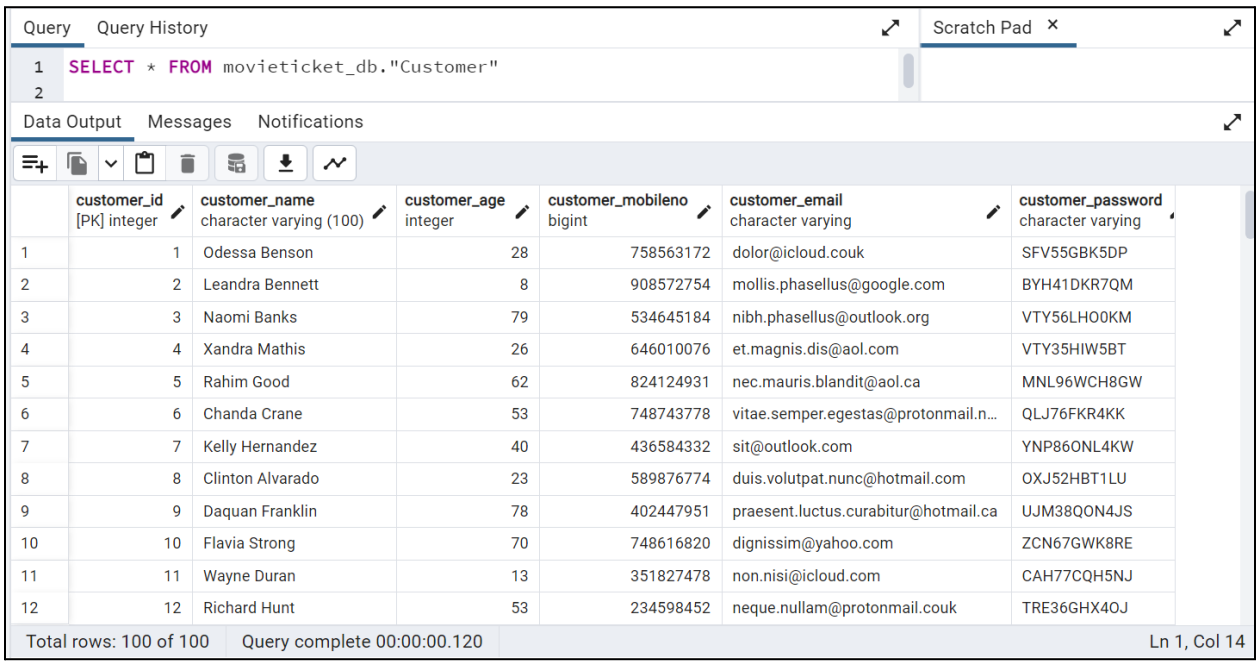
25. Find The Screen With The Maximum Number Of Seats.
26. Count The Number Of Seats In Theater Id 2.
27. Create And Display A View Of All The Transactions Whose Date Is 17 November 2021.
28. Display The List Of Theater Which Have More Than 5 Screens.
29. List The Movies Which Have Only One Show.
30. Display The Count Of Seats Alongside Seat Type, Price And Sort Them Price Wise In Descending Order.
31. Create A View For Admin Details Then Insert And Display Some New Admin To The Recently Created View.
32. Create View Of Ticket Whose Amount Is Less Than 500.
33. Create View Of All Seat Details.
34. Create View Of Movie Where Duration Is Less Than 120 Minutes, Language Is English, And Movie Name Starts With ‘S’.
35. Display Transaction Ids And Their Amount Whose Amount Is Greater Than Average Amount Spent By User.
36. Display The Movie Names Alongside Customer Name Which Are Booked By Customer Id 1
37. Display The Details Of The 2nd Highest Amount Paid For A Meal.
38. Display Theater Name With Movie Name Where Movie "Thumbelina" Is Running.
39. Display Show Details For The Movies Having Duration Less Than 120 Mins.
40. Display The Shows In Theater "Oscar Holland"

❖ **SQL Statements**

1. Show Customer Details

SQL - SELECT * FROM movieticket_db."Customer"

Output -



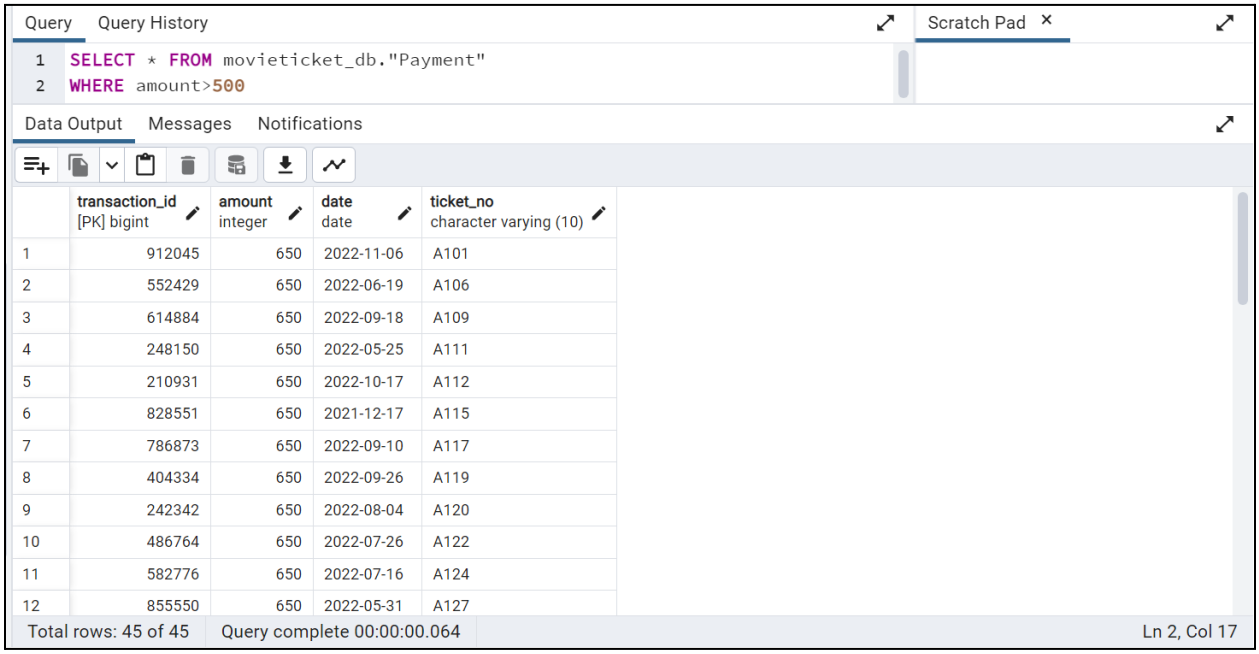
	customer_id [PK] integer	customer_name character varying (100)	customer_age integer	customer_mobileno bigint	customer_email character varying	customer_password character varying
1	1	Odessa Benson	28	758563172	dolor@icloud.couk	SFV55GBK5DP
2	2	Leandra Bennett	8	908572754	mollis.phasellus@google.com	BYH41DKR7QM
3	3	Naomi Banks	79	534645184	nibh.phasellus@outlook.org	VTY56LHO0KM
4	4	Xandra Mathis	26	646010076	et.magnis.dis@aol.com	VTY35HIW5BT
5	5	Rahim Good	62	824124931	nec.mauris.blandit@aol.ca	MNL96WCH8GW
6	6	Chanda Crane	53	748743778	vitae.semper.egestas@protonmail.n...	QLJ76FKR4KK
7	7	Kelly Hernandez	40	436584332	sit@outlook.com	YNP86ONL4KW
8	8	Clinton Alvarado	23	589876774	duis.volutpat.nunc@hotmail.com	OXJ52HBT1LU
9	9	Daquan Franklin	78	402447951	praesent.luctus.curabitur@hotmail.ca	UJM38QON4JS
10	10	Flavia Strong	70	748616820	dignissim@yahoo.com	ZCN67GWK8RE
11	11	Wayne Duran	13	351827478	non.nisi@icloud.com	CAH77CQH5NJ
12	12	Richard Hunt	53	234598452	neque.nullam@protonmail.couk	TRE36GHX4OJ

Total Tuples - 100

2. Show Payment Details Whose Amount Is Greater Than 500

SQL - SELECT * FROM movieticket_db."Payment"
WHERE amount>500

Output -



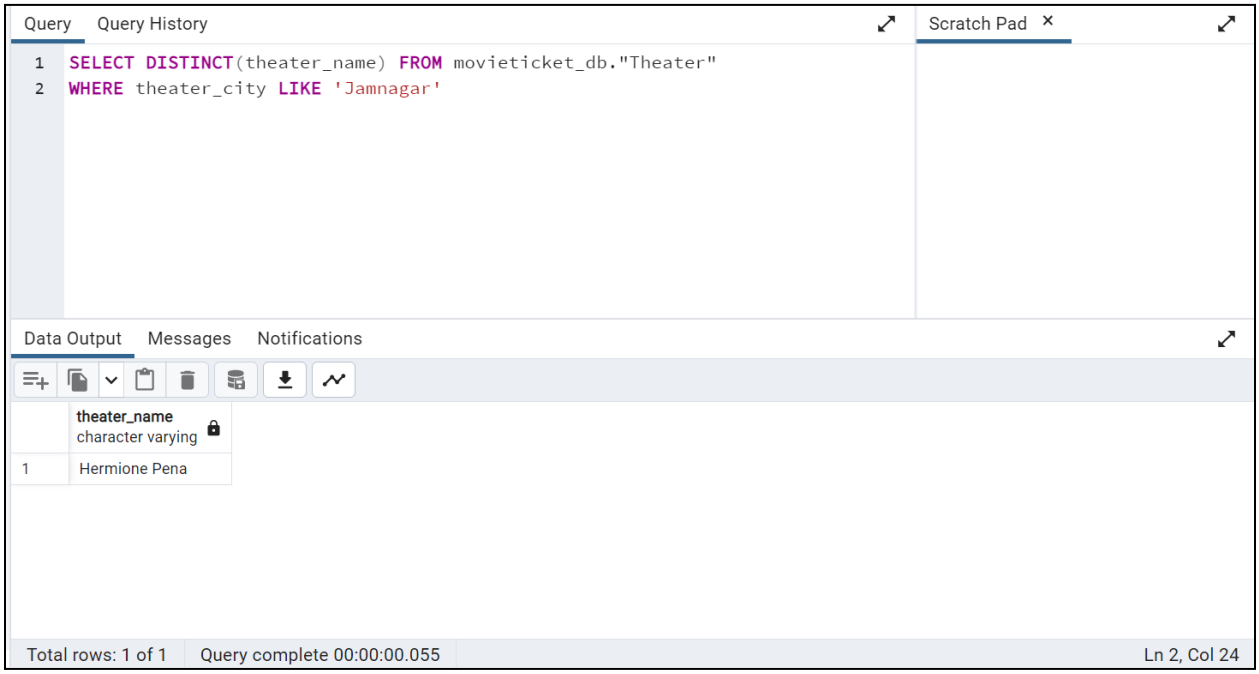
	transaction_id [PK] bigint	amount integer	date date	ticket_no character varying (10)
1	912045	650	2022-11-06	A101
2	552429	650	2022-06-19	A106
3	614884	650	2022-09-18	A109
4	248150	650	2022-05-25	A111
5	210931	650	2022-10-17	A112
6	828551	650	2021-12-17	A115
7	786873	650	2022-09-10	A117
8	404334	650	2022-09-26	A119
9	242342	650	2022-08-04	A120
10	486764	650	2022-07-26	A122
11	582776	650	2022-07-16	A124
12	855550	650	2022-05-31	A127

Total Tuples - 45

3. Get Unique List Of Theaters In Jamnagar.

SQL - SELECT DISTINCT(theater_name) FROM movieticket_db."Theater"
WHERE theater_city LIKE 'Jamnagar'

Output -

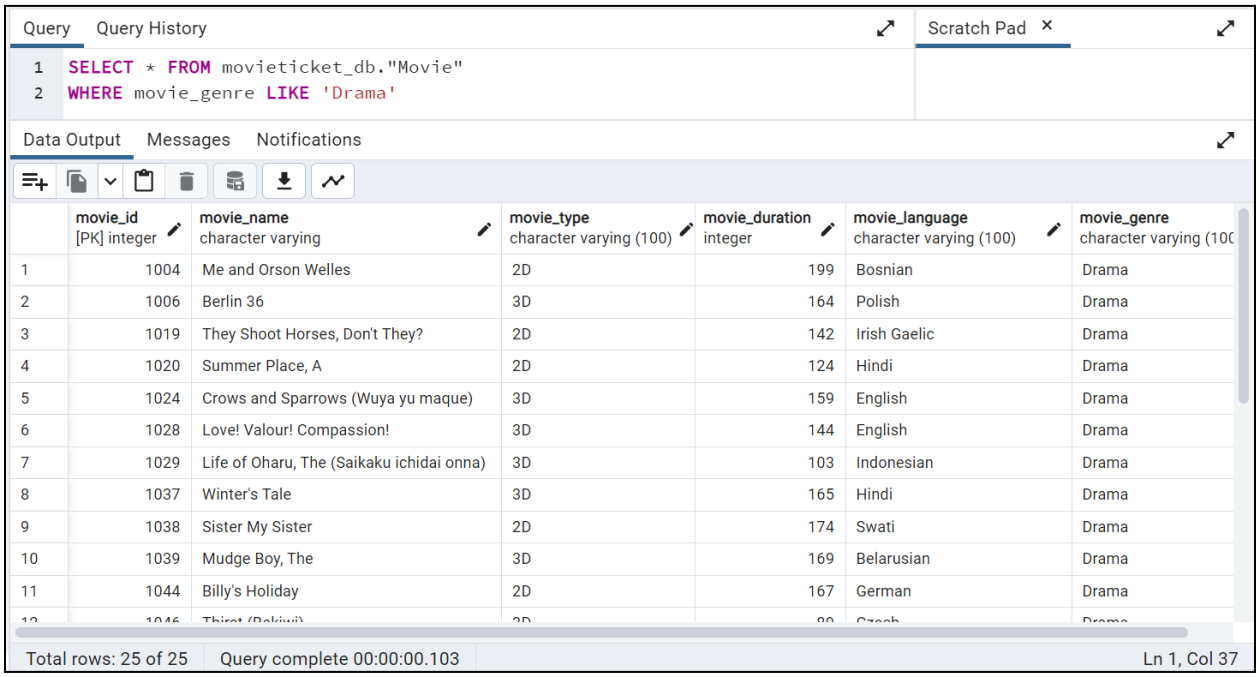


Total Tuples - 1

4. Show Movie Details Of Drama Genre

SQL - SELECT * FROM movieticket_db."Movie"
WHERE movie_genre LIKE 'Drama'

Output -

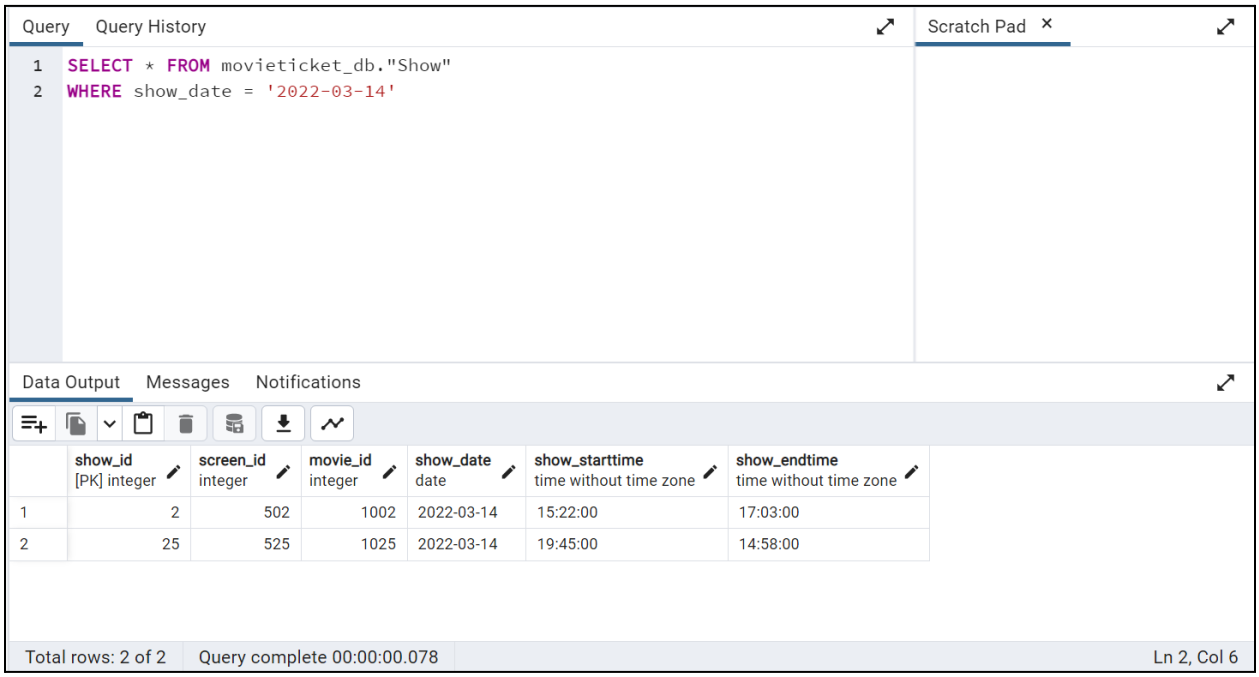


Total Tuples - 25

5. Display Show Details Of 14 March 2022

SQL - SELECT * FROM movieticket_db."Show"
WHERE show_date = '2022-03-14'

Output -

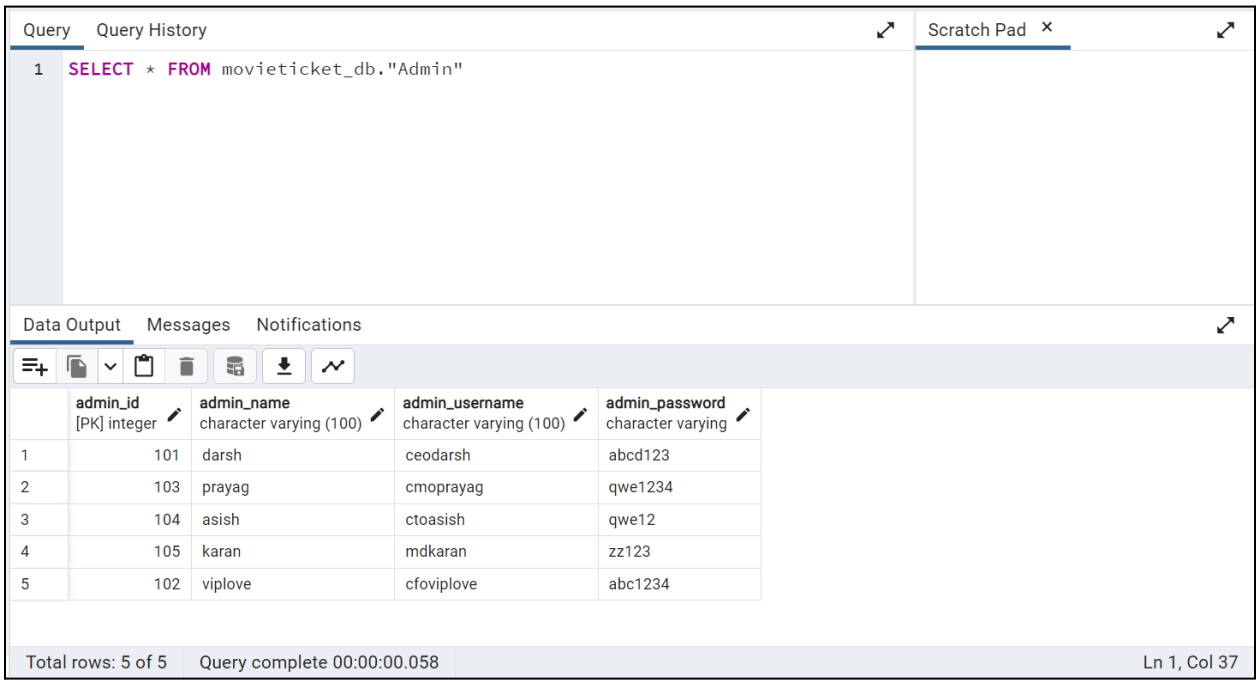


Total Tuples - 2

6. Show Admin Details

SQL - SELECT * FROM movieticket_db."Admin"

Output -

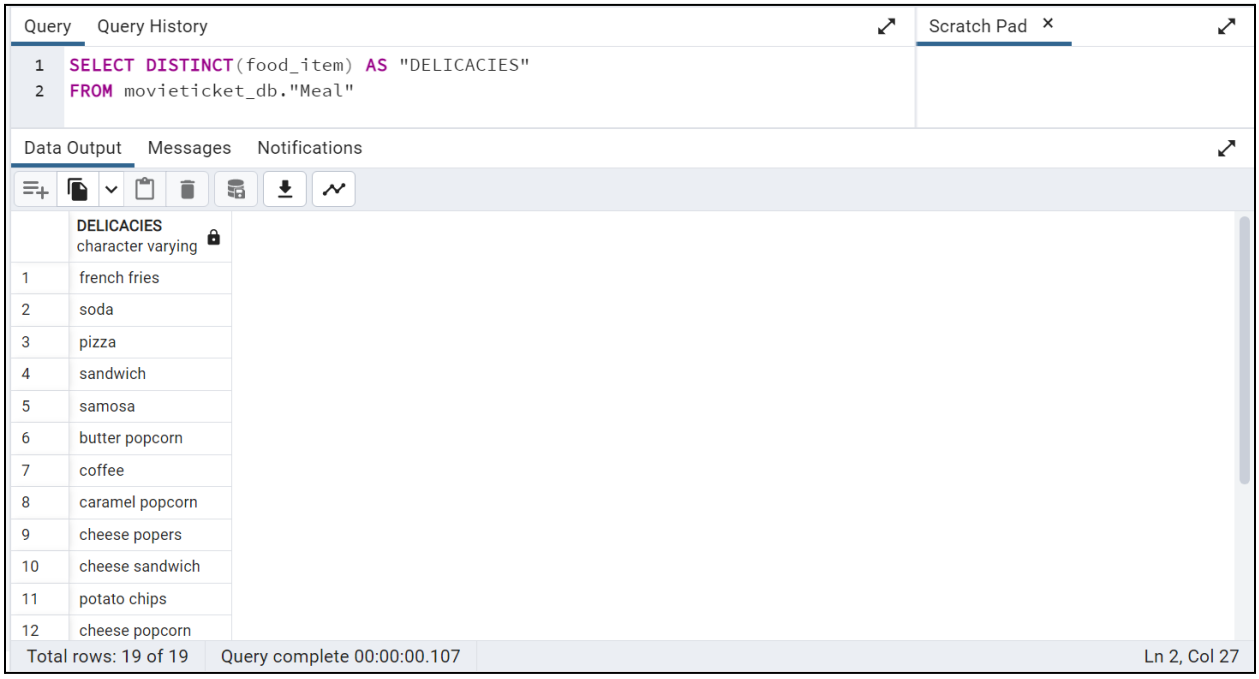


Total Tuples - 5

7. Write A Query To Display Unique Food Item Using The Alias Name As Delicacies

SQL - SELECT DISTINCT(food_item) AS "DELICACIES"
FROM movieticket_db."Meal"

Output -



Total Tuples - 19

8. Display Show Where Time Is Between 3 PM - 11 PM

SQL - SELECT * FROM movieticket_db."Show"
WHERE (show_starttime BETWEEN '15:00:00' AND '23:00:00')
AND (show_endtime BETWEEN '15:00:00' AND '23:00:00')

Output -

Query | Query History | Scratch Pad

```
1  SELECT * FROM movieticket_db."Show"
2  WHERE (show_starttime BETWEEN '15:00:00' AND '23:00:00')
3  AND (show_endtime BETWEEN '15:00:00' AND '23:00:00')
```

Data Output | Messages | Notifications

	show_id [PK] integer	screen_id integer	movie_id integer	show_date date	show_starttime time without time zone	show_endtime time without time zone
1	2	502	1002	2022-03-14	15:22:00	17:03:00
2	12	512	1012	2022-07-31	16:53:00	19:20:00
3	13	513	1013	2022-09-22	19:57:00	21:44:00
4	15	515	1015	2022-08-07	19:37:00	19:55:00
5	17	517	1017	2022-06-18	15:44:00	22:34:00
6	18	518	1018	2022-01-18	18:07:00	18:39:00
7	19	519	1019	2022-04-12	16:06:00	19:17:00
8	23	523	1023	2022-09-26	20:05:00	21:04:00
9	24	524	1024	2022-03-20	20:06:00	19:14:00
10	31	531	1031	2022-07-29	17:30:00	18:46:00

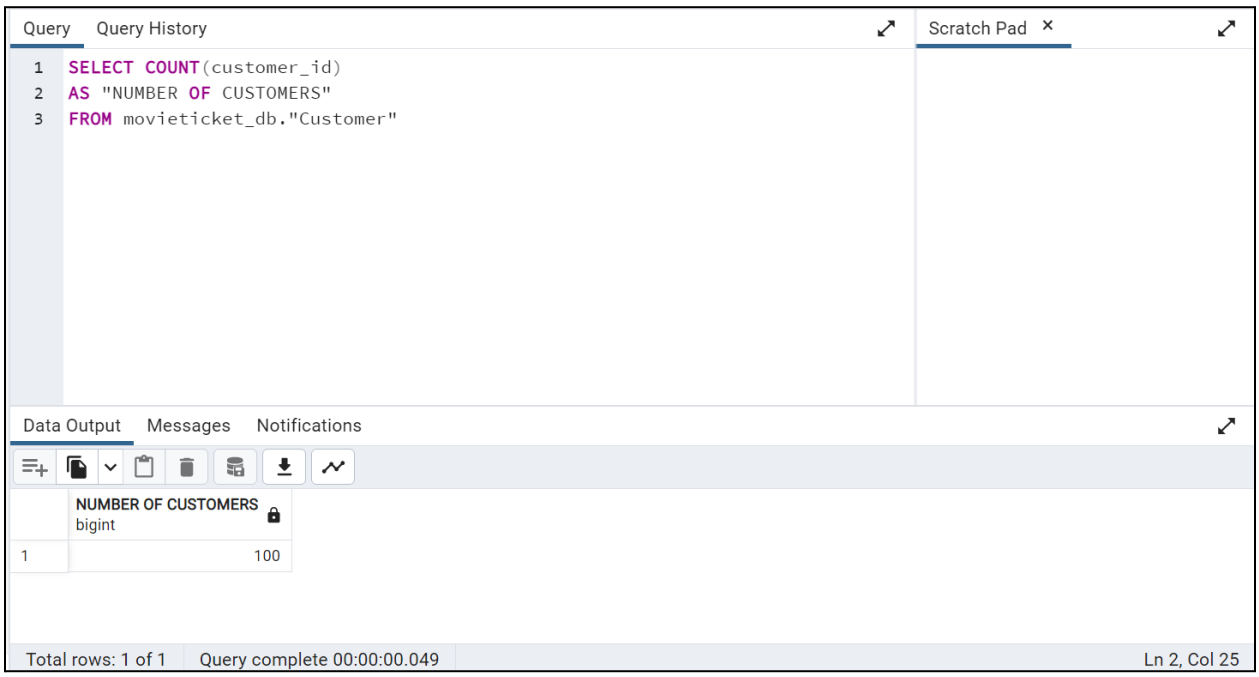
Total rows: 51 of 51 | Query complete 00:00:00.061 | Ln 2, Col 8

Total Tuples - 51

9. Count Total Customer

SQL - SELECT COUNT(customer_id)
AS "NUMBER OF CUSTOMERS"
FROM movieticket_db."Customer"

Output -

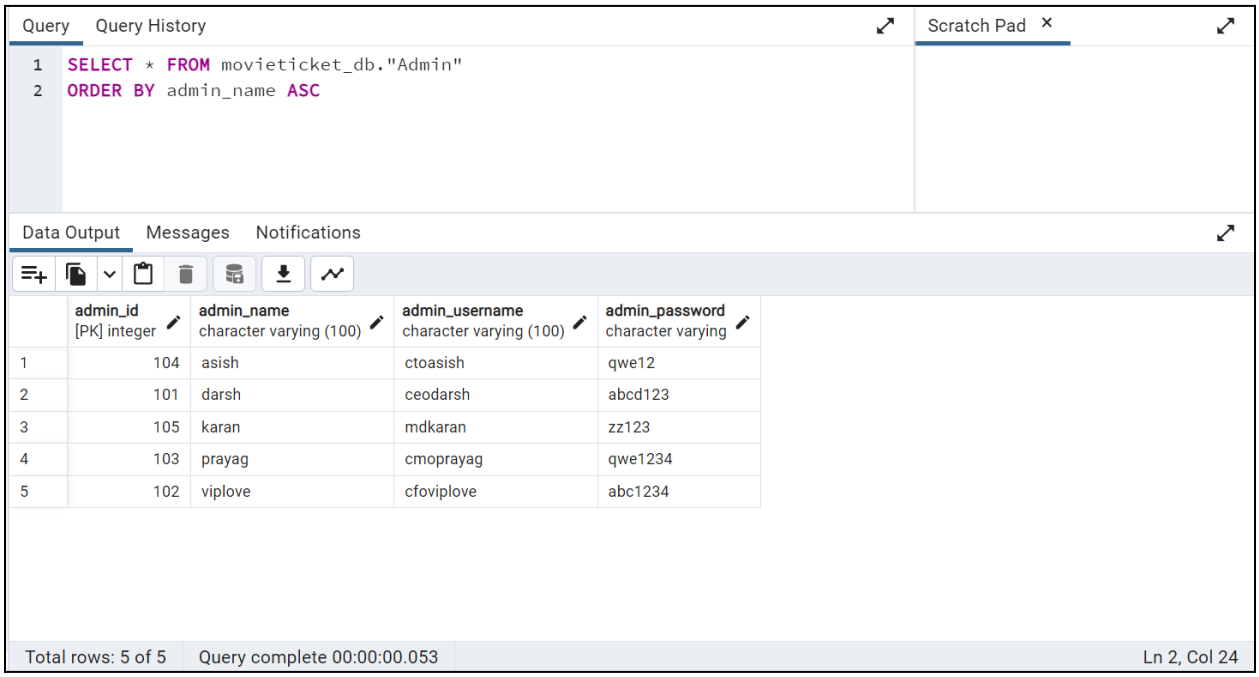


Total Tuples - 1

10. Sort Admin By Name

SQL - SELECT * FROM movieticket_db."Admin"
ORDER BY admin_name ASC

Output -



Total Tuples - 5

11. Count The Number Of Shows On 31 July 2022 Which Were Screened On Screen Id 512

SQL - SELECT COUNT(show_id)
FROM movieticket_db."Show"
WHERE show_date = '31-7-2022'AND screen_id=512

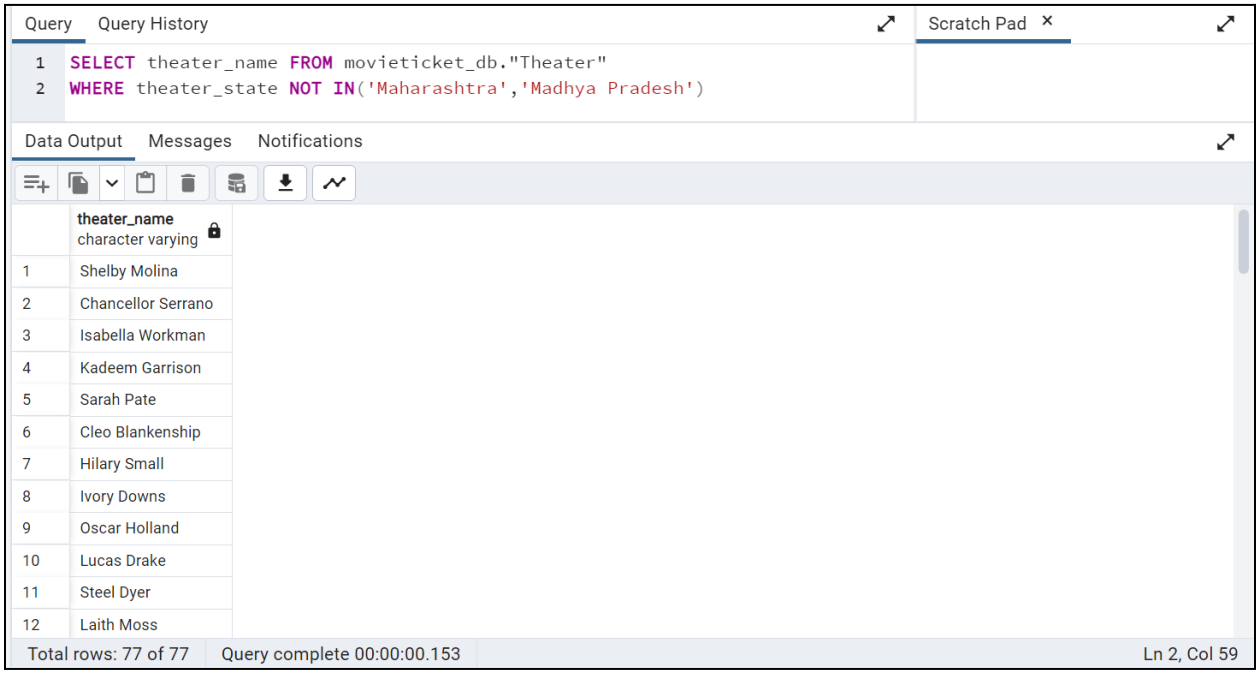
Output -

Total Tuples - 1

12. Display List Of Theaters Excluding The Ones Present In Maharashtra And Madhya Pradesh.

SQL - SELECT theater_name FROM movieticket_db."Theater"
WHERE theater_state NOT IN('Maharashtra','Madhya Pradesh')

Output -

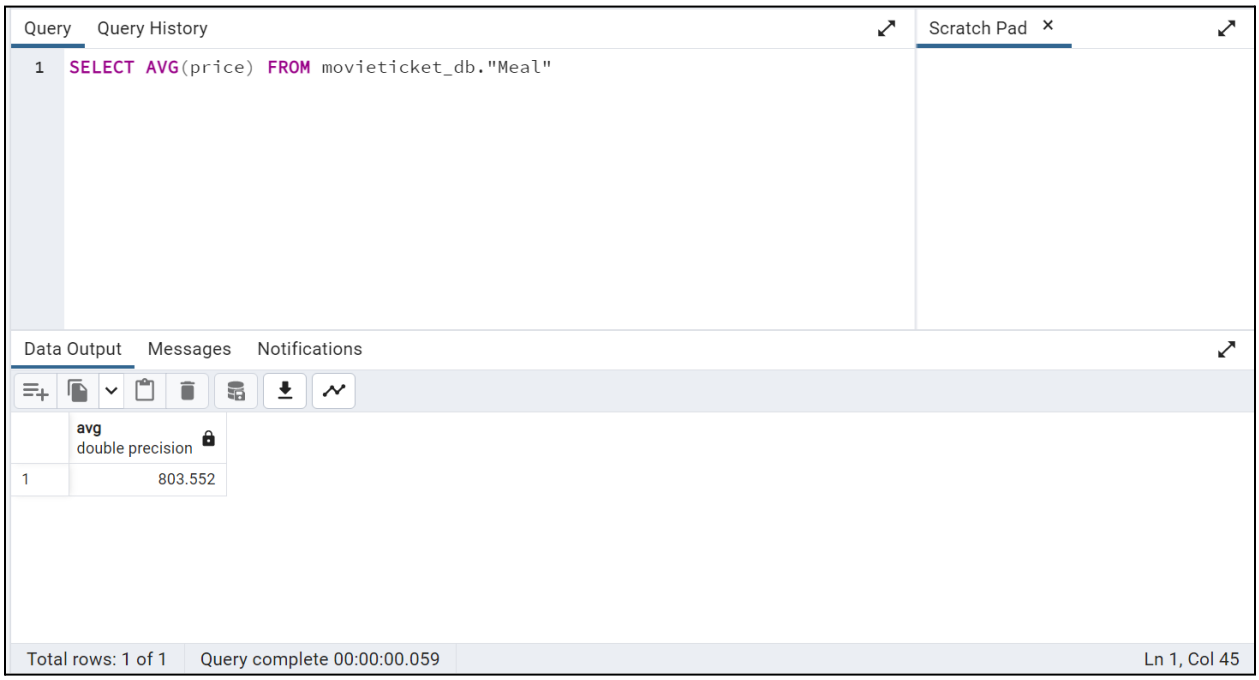


Total Tuples - 77

13. Find Out The Average Meal Price.

SQL - SELECT AVG(price) FROM movieticket_db."Meal"

Output -



Total Tuples - 1

14. Print The Count Of Different Types Of Seats.

SQL - SELECT COUNT(DISTINCT(seat_type))
FROM movieticket_db."Seat"

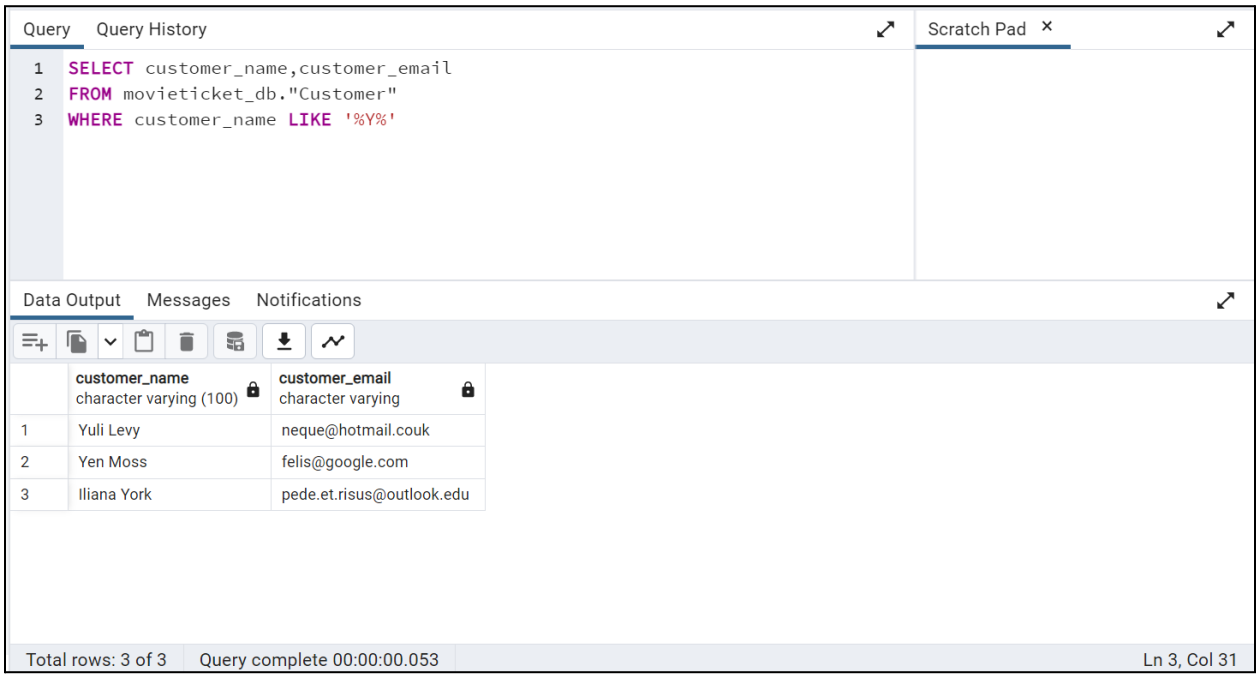
Output -

Total Tuples - 1

15. Display The Name And Email Of A Customer Who Has ‘y’ In Their Names.

SQL - SELECT customer_name,customer_email
FROM movieticket_db."Customer"
WHERE customer_name LIKE '%Y%'

Output -



Total Tuples - 3

16. Display List Of Movies Which Have Genre Comedy And Type Not 2D.

SQL - SELECT * FROM movieticket_db."Movie"
WHERE movie_genre LIKE 'Comedy'
AND NOT movie_type = '2D'

Output -

Query | Query History | Scratch Pad

```
1 SELECT * FROM movieticket_db."Movie"
2 WHERE movie_genre LIKE 'Comedy'
3 AND NOT movie_type = '2D'
```

Data Output | Messages | Notifications

	movie_id [PK] integer	movie_name character varying	movie_type character varying (100)	movie_duration integer	movie_language character varying (100)	movie_genre character varying (100)
1	1003	Moonlight and Cactus	3D	197	Armenian	Comedy
2	1008	Sumo Do, Sumo Don't (Shiko funj...	3D	191	Afrikaans	Comedy
3	1011	Oh Happy Day	3D	182	English	Comedy
4	1013	Cherry Falls	3D	93	Lithuanian	Comedy
5	1016	Four Eyed Monsters	3D	96	English	Comedy
6	1032	Why Not Me? (Pourquoi pas moi ?)	3D	106	Dutch	Comedy
7	1041	Rum Diary, The	3D	94	Portuguese	Comedy
8	1045	Louis C.K.: Shameless	3D	64	Swedish	Comedy
9	1049	Combien Tu M'aimes? (How Muc...	3D	90	Armenian	Comedy
10	1050	Peculiarities of the National Hunt ...	3D	119	English	Comedy
11	1069	Papa	3D	190	French	Comedy

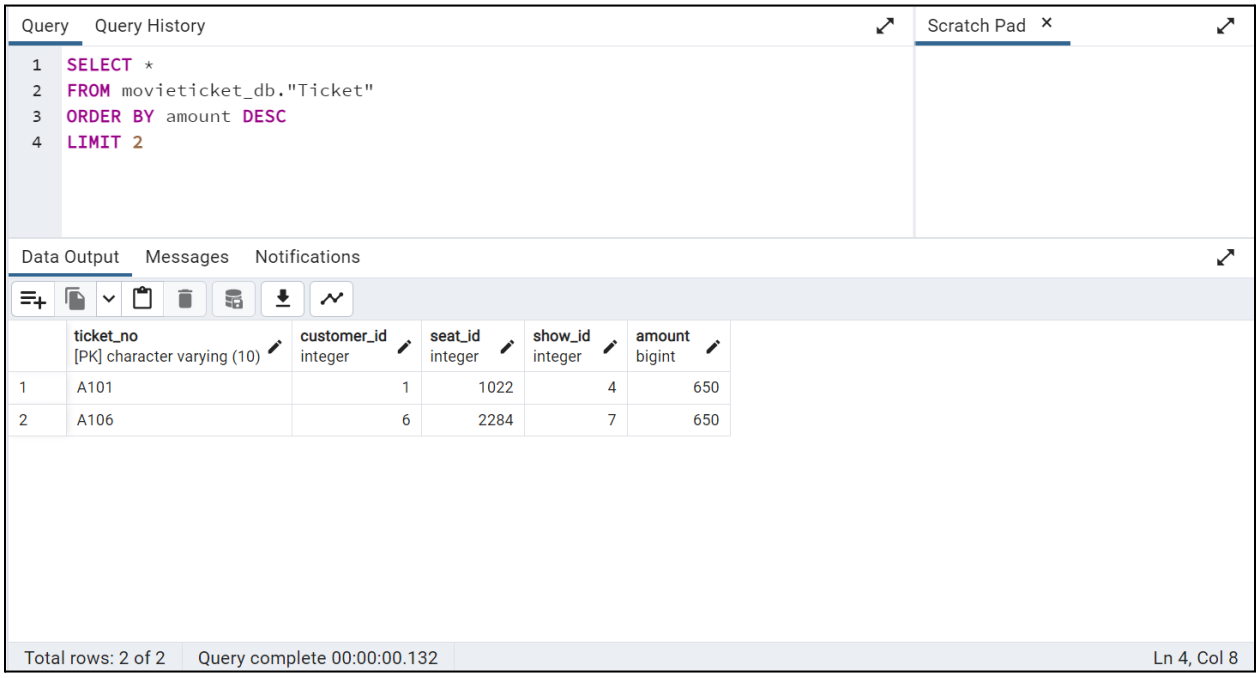
Total rows: 16 of 16 | Query complete 00:00:00.137 | Ln 3, Col 26

Total Tuples - 16

17. Select Details Of Top 2 Highest Price Tickets.

SQL - SELECT *
FROM movieticket_db."Ticket"
ORDER BY amount DESC
LIMIT 2

Output -

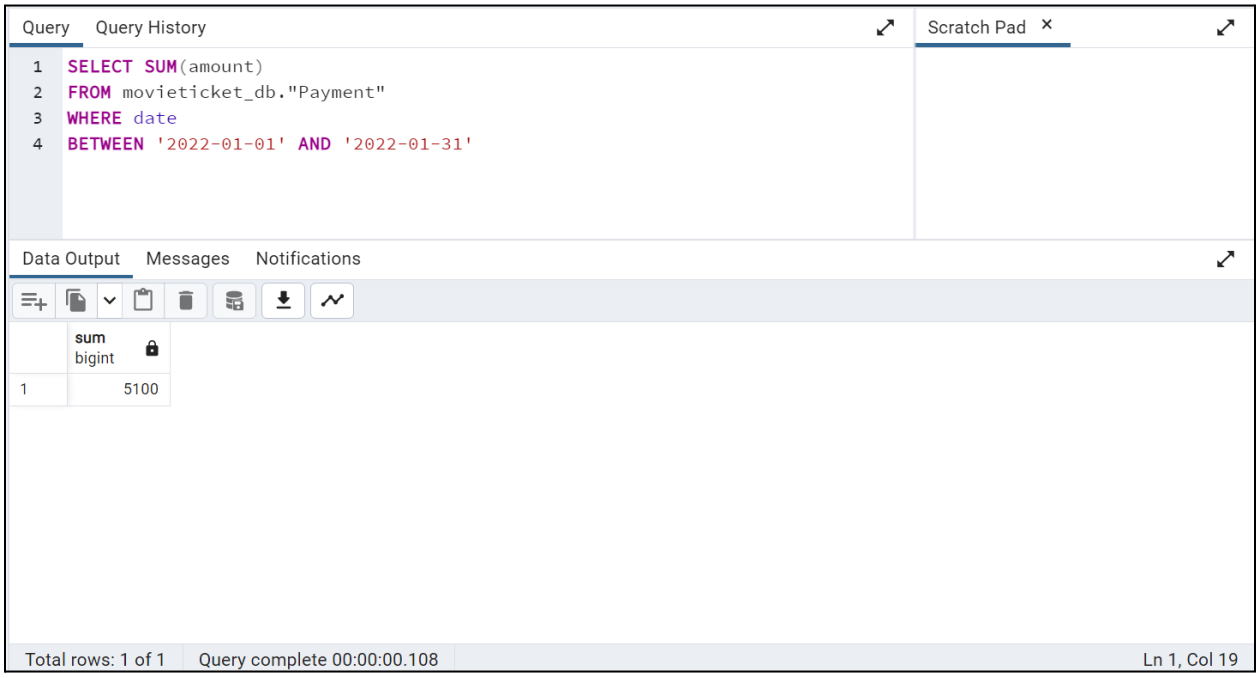


Total Tuples - 2

18. Display The Total Amount Received In The January Month Of 2022.

SQL - SELECT SUM(amount)
FROM movieticket_db."Payment"
WHERE date
BETWEEN '2022-01-01' AND '2022-01-31'

Output -

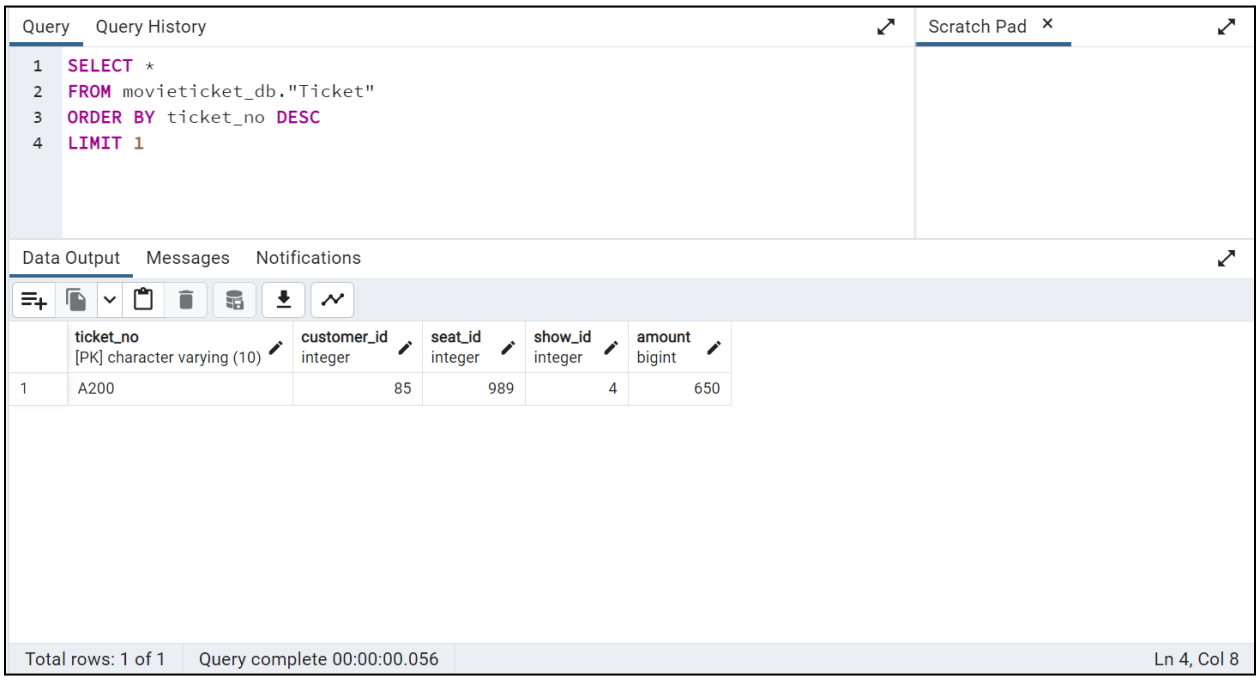


Total Tuples - 1

19. Write A Query To Fetch The Details Of The Last Ticket Sold.

SQL - SELECT *
FROM movieticket_db."Ticket"
ORDER BY ticket_no DESC
LIMIT 1

Output -

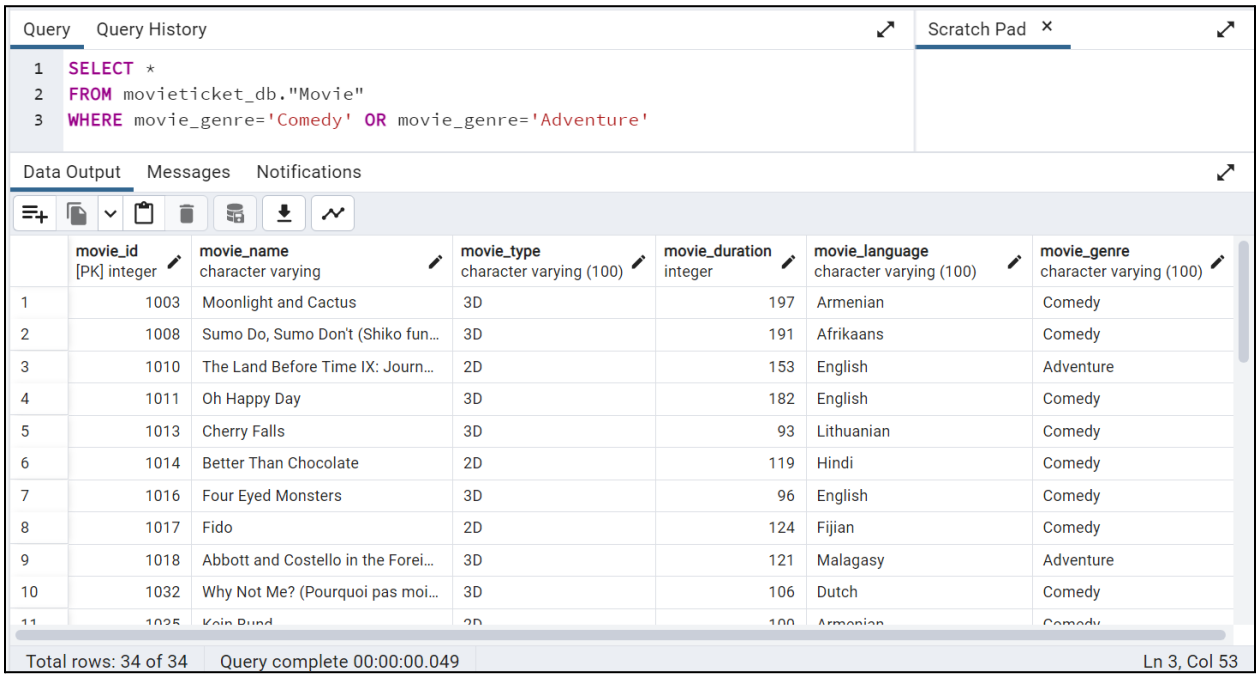


Total Tuples - 1

20. Print The Details Of Movies Where Genre Is Comedy Or Adventure

SQL - SELECT *
FROM movieticket_db."Movie"
WHERE movie_genre='Comedy' OR movie_genre='Adventure'

Output -



	movie_id [PK] integer	movie_name character varying	movie_type character varying (100)	movie_duration integer	movie_language character varying (100)	movie_genre character varying (100)
1	1003	Moonlight and Cactus	3D	197	Armenian	Comedy
2	1008	Sumo Do, Sumo Don't (Shiko fun...	3D	191	Afrikaans	Comedy
3	1010	The Land Before Time IX: Journ...	2D	153	English	Adventure
4	1011	Oh Happy Day	3D	182	English	Comedy
5	1013	Cherry Falls	3D	93	Lithuanian	Comedy
6	1014	Better Than Chocolate	2D	119	Hindi	Comedy
7	1016	Four Eyed Monsters	3D	96	English	Comedy
8	1017	Fido	2D	124	Fijian	Comedy
9	1018	Abbott and Costello in the Forei...	3D	121	Malagasy	Adventure
10	1032	Why Not Me? (Pourquoi pas moi...	3D	106	Dutch	Comedy
11	1035	Koin Rund	2D	100	Armenian	Comedy

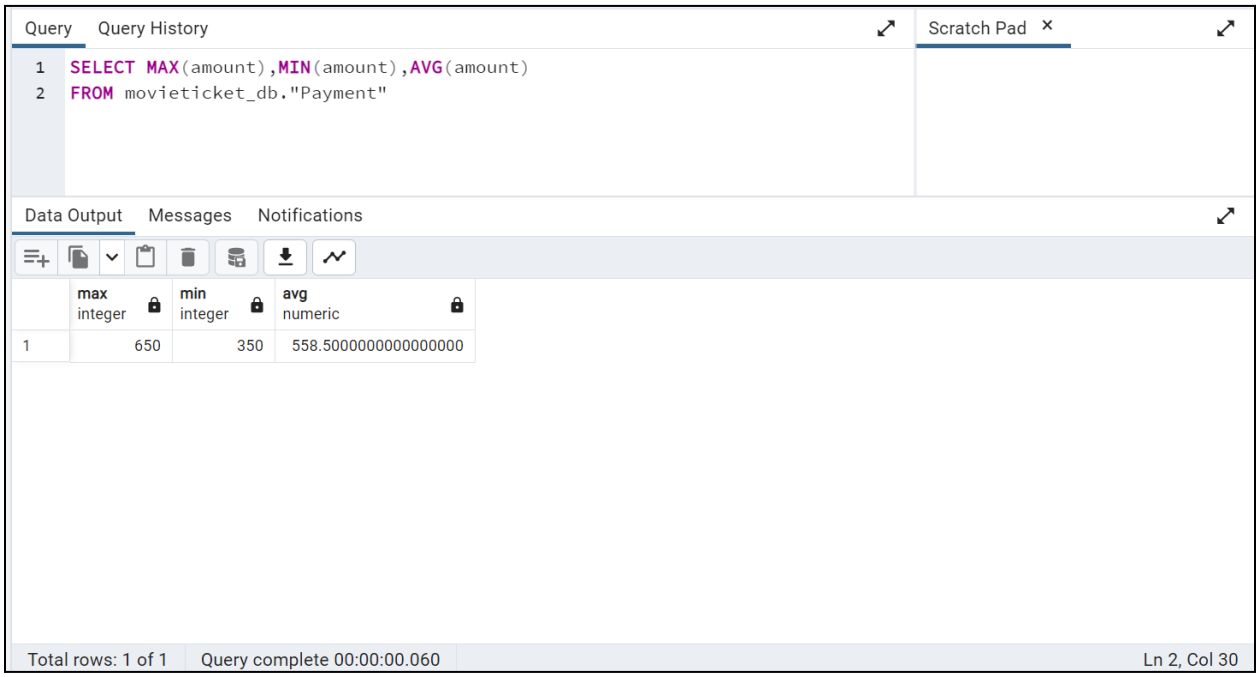
Total rows: 34 of 34 Query complete 00:00:00.049 Ln 3, Col 53

Total Tuples - 34

21. Display Maximum , Minimum And Average Of Transaction Amount.

SQL - SELECT MAX(amount),MIN(amount),AVG(amount)
FROM movieticket_db."Payment"

Output -

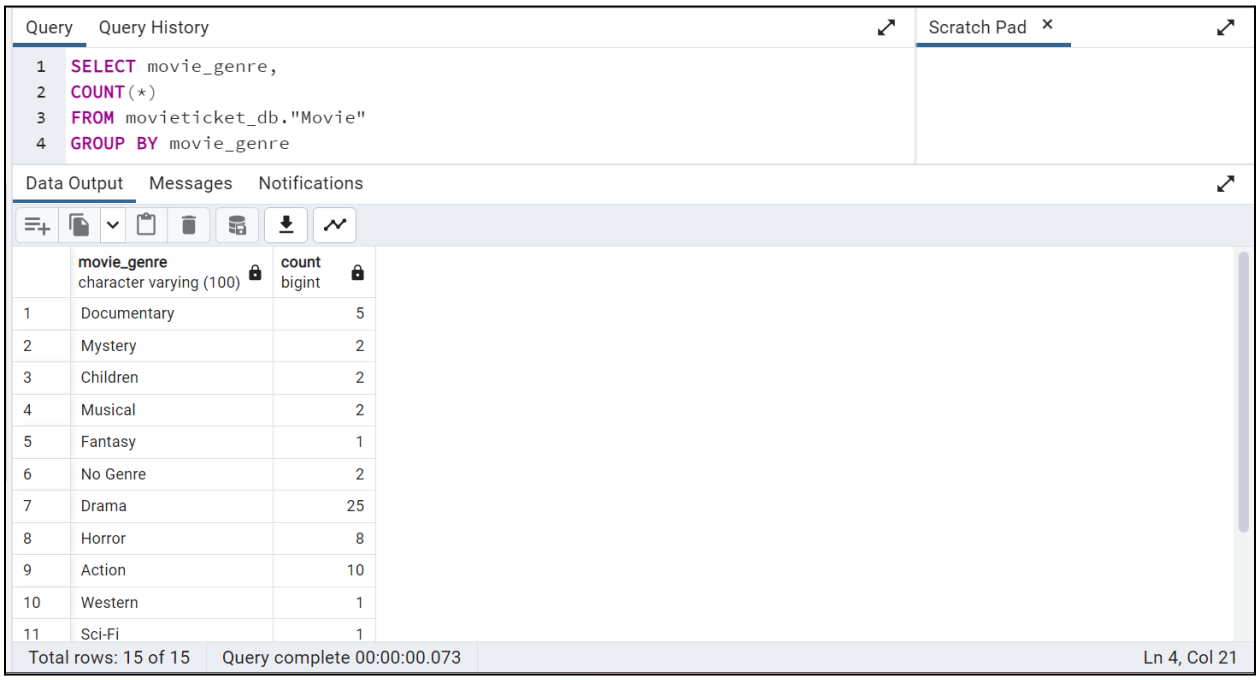


Total Tuples - 1

22. Count The Number Of Movies Genre Wise.

SQL - SELECT movie_genre,
COUNT(*)
FROM movieticket_db."Movie"
GROUP BY movie_genre

Output -



Total Tuples - 15

23. Write A Query To Fetch Only Odd Ids From Customer Table

SQL - SELECT * FROM movieticket_db."Customer"
WHERE MOD(customer_id,2)=1

Output -

Query | Query History | Scratch Pad

```
SELECT * FROM movieticket_db."Customer"
WHERE MOD(customer_id,2)=1
```

Data Output | Messages | Notifications

	customer_id [PK] integer	customer_name character varying (100)	customer_age integer	customer_mobileno bigint	customer_email character varying	customer_password character varying
1	1	Odessa Benson	28	758563172	dolor@icloud.couk	SFV55GBK5DP
2	3	Naomi Banks	79	534645184	nibh.phasellus@outlook.org	VTY56LHO0KM
3	5	Rahim Good	62	824124931	nec.mauris.blandit@aol.ca	MNL96WCH8GW
4	7	Kelly Hernandez	40	436584332	sit@outlook.com	YNP86ONL4KW
5	9	Daquan Franklin	78	402447951	praesent.luctus.curabitur@hotmail.ca	UJM38QON4JS
6	11	Wayne Duran	13	351827478	non.nisi@icloud.com	CAH77CQH5NJ
7	13	Karly Buckley	55	272863233	lacus.aliquam@yahoo.net	HRF61COI6VU
8	15	Tanek Knowles	19	927261331	etiam@protonmail.edu	MJA70SHJ4AM
9	17	Sophia Allison	7	446477791	ultrices@icloud.ca	PWW56BGS6XT
10	19	Hadley Wolfe	60	822987651	ut.erat@outlook.org	IWC21KJV6RF
11	21	Serina Walsh	22	855922063	ridiculus@google.ca	LTF02FPR3GI
12	23	Cassidy Avila	61	422217549	pellentesque@icloud.net	ZUY94UFR4CV
13	25	Wing Arnold	7	216523446	posuere.enim@protonmail.net	NUB22PRR7MV
14	27	Christian Stewart	20	221417536	tellus.nunc@hotmail.couk	GIX75FJM4OU

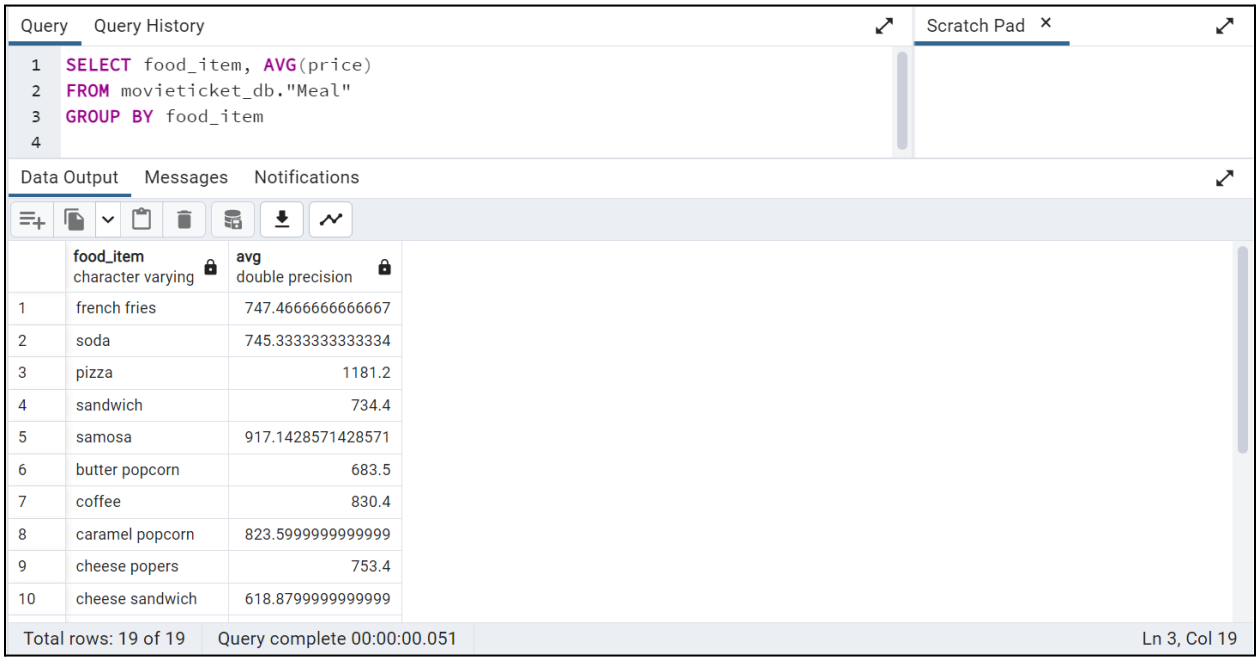
Total rows: 50 of 50 | Query complete 00:00:00.286 | Ln 2, Col 26

Total Tuples - 50

24. Display Meal With Average Price

SQL - SELECT food_item, AVG(price)
FROM movieticket_db."Meal"
GROUP BY food_item

Output -

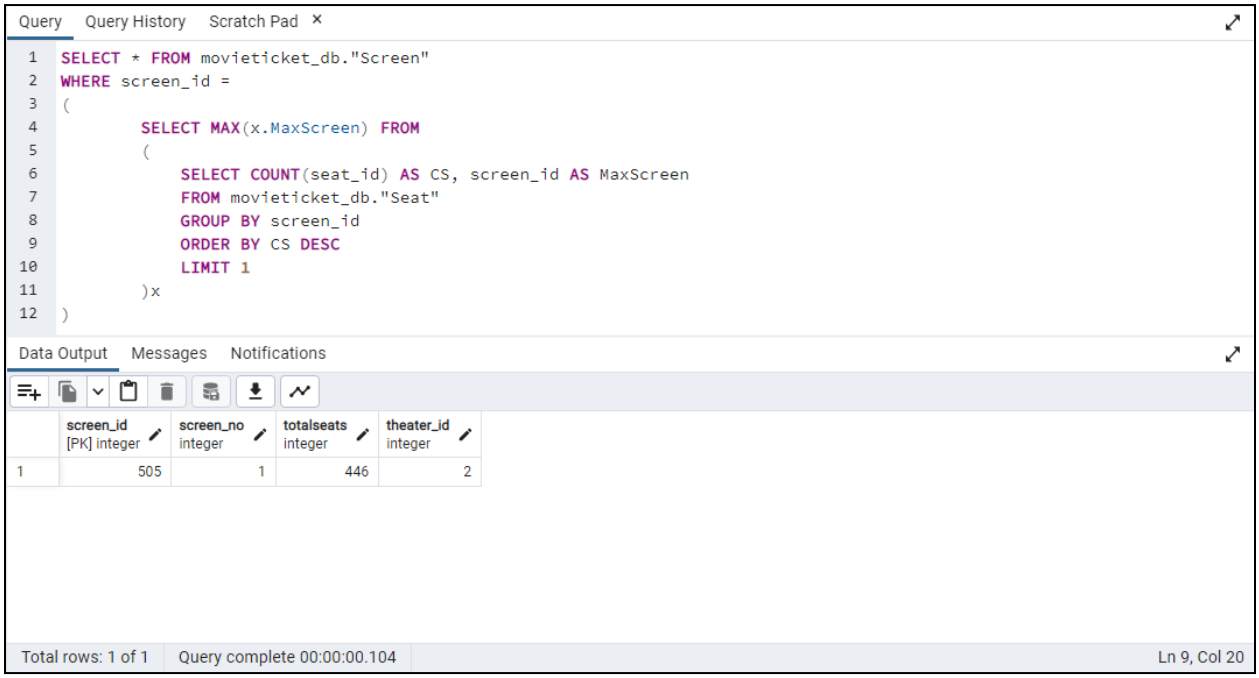


Total Tuples - 19

25. Find The Screen With The Maximum Number Of Seats.

SQL - SELECT * FROM movieticket_db."Screen"
WHERE screen_id =
(
SELECT MAX(x.MaxScreen) FROM
(
SELECT COUNT(seat_id) AS CS, screen_id AS MaxScreen
FROM movieticket_db."Seat"
GROUP BY screen_id
ORDER BY CS DESC
LIMIT 1
)x
)

Output -

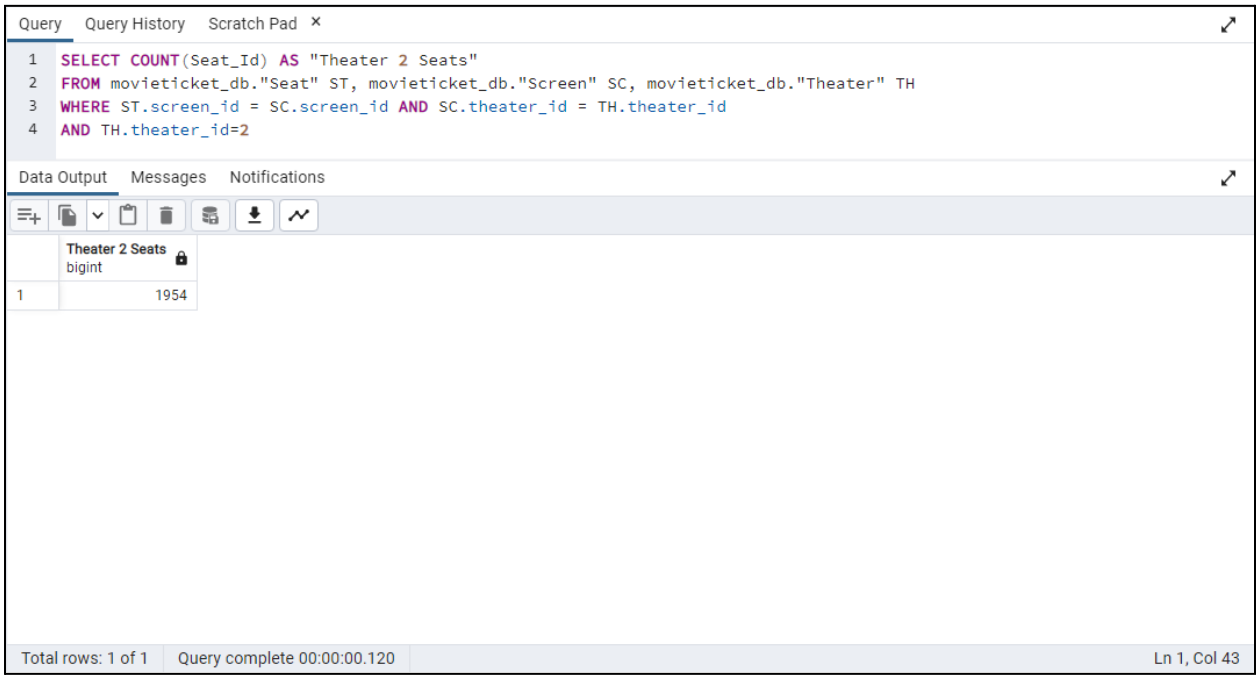


Total Tuples - 1

26. Count The Number Of Seats In Theater Id 2

SQL - SELECT COUNT(Seat_Id) AS "Theater 2 Seats"
FROM movieticket_db."Seat" ST, movieticket_db."Screen" SC,
movieticket_db."Theater" TH
WHERE ST.screen_id = SC.screen_id AND SC.theater_id = TH.theater_id
AND TH.theater_id=2

Output -



Total Tuples - 1

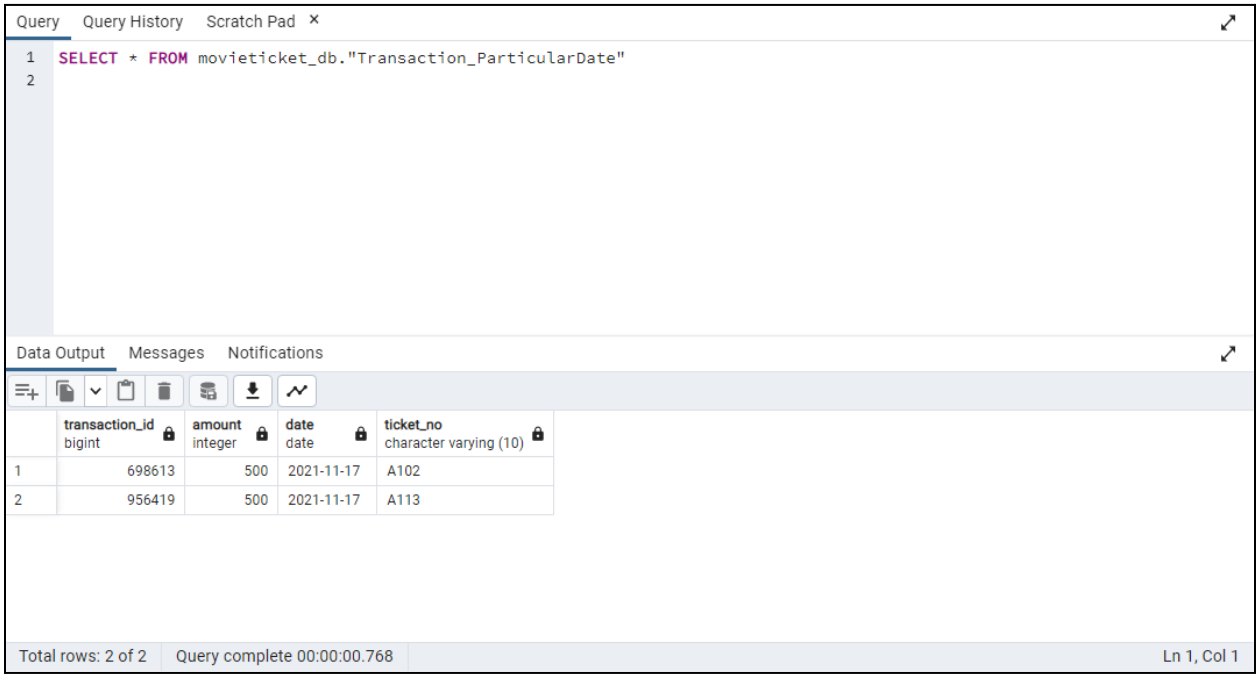
27. Create And Display A View Of All The Transactions Whose Date Is 17 November 2021

SQL - CREATE VIEW movieticket_db."Transaction_ParticularDate" AS
SELECT * FROM movieticket_db."Payment"
WHERE date = '17/11/2021'

Output -

SQL - SELECT * FROM movieticket_db."Transaction_ParticularDate"

Output -

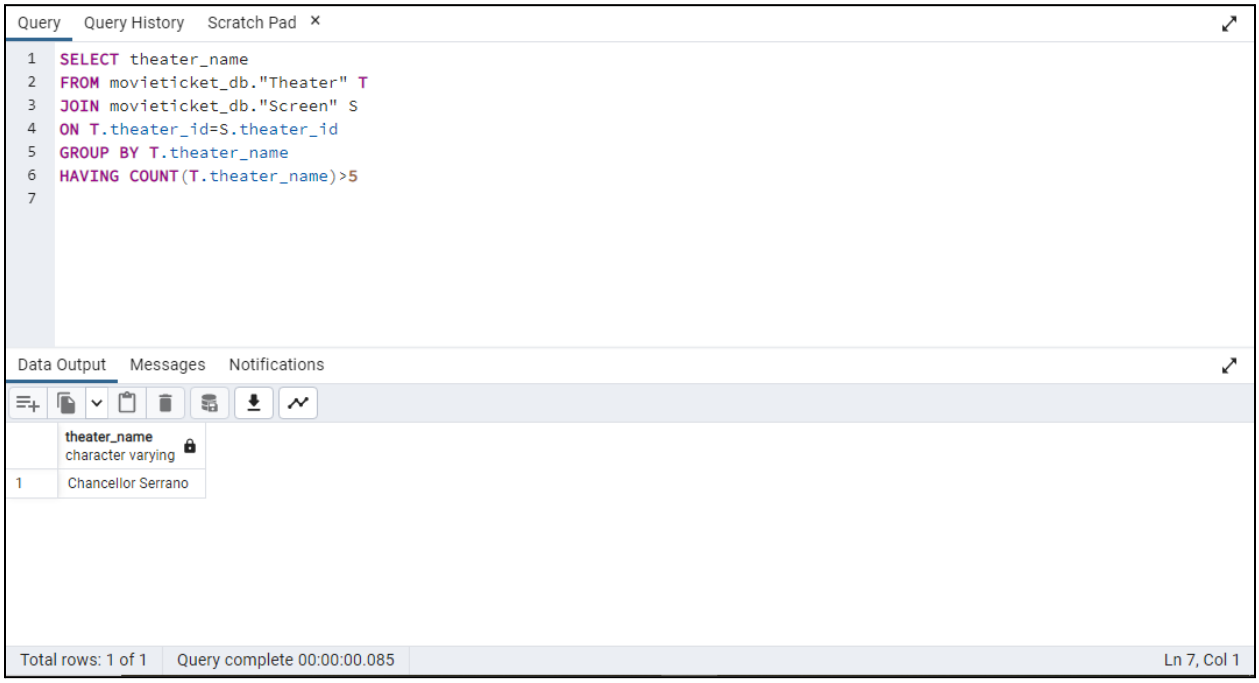


Total Tuple - 2

28. Display The List Of Theater Which Have More Than 5 Screens

SQL - SELECT theater_name
FROM movieticket_db."Theater" T
JOIN movieticket_db."Screen" S
ON T.theater_id=S.theater_id
GROUP BY T.theater_name
HAVING COUNT(T.theater_name)>5

Output -

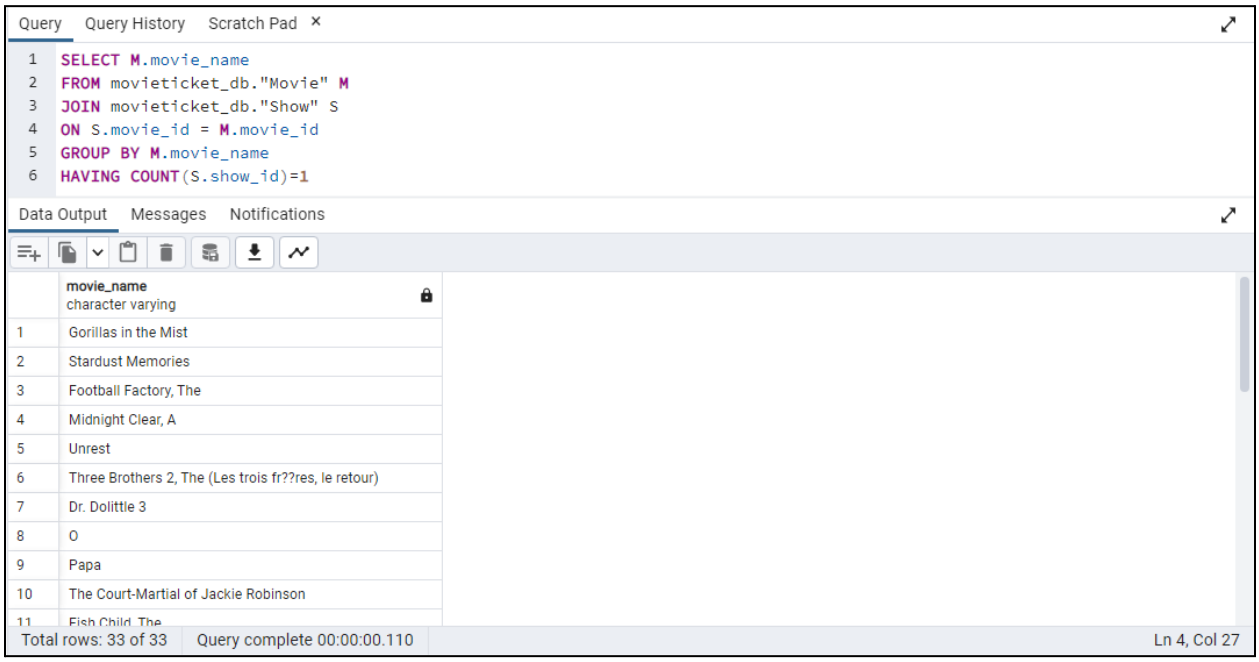


Total Tuple - 1

29. List The Movies Which Have Only One Show.

SQL - SELECT M.movie_name
FROM movieticket_db."Movie" M
JOIN movieticket_db."Show" S
ON S.movie_id = M.movie_id
GROUP BY M.movie_name
HAVING COUNT(S.show_id)=1

Output -

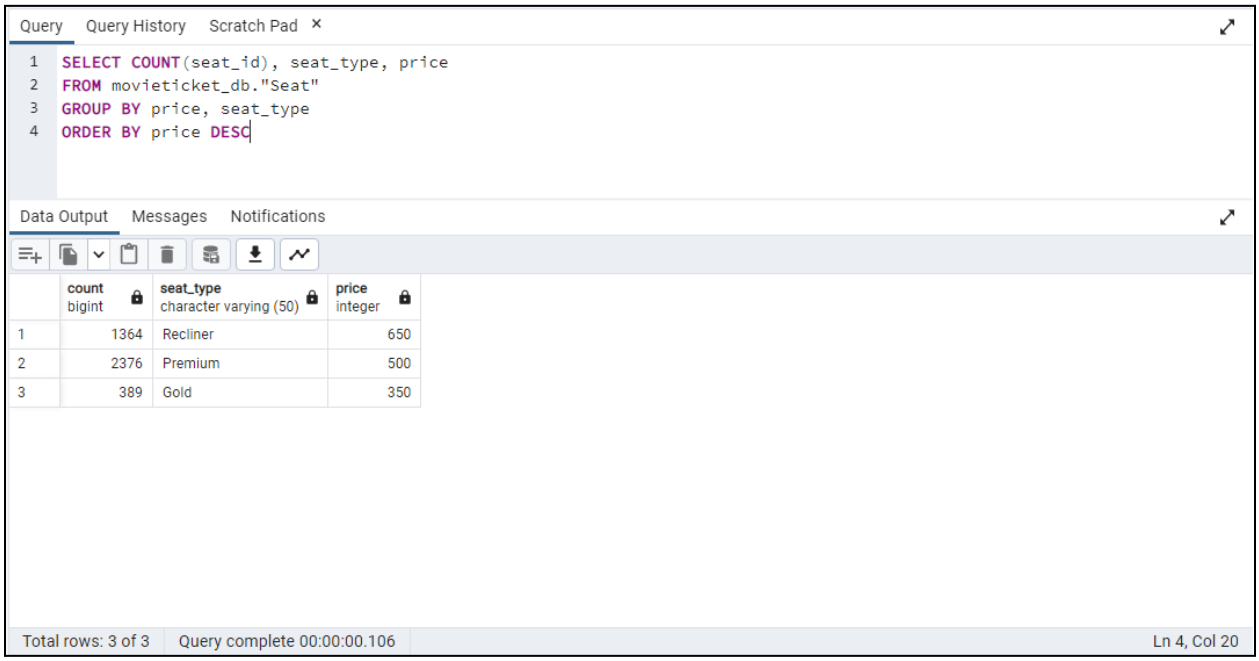


Total Tuples - 33

30. Display The Count Of Seats Alongside Seat Type, Price And Sort Them Price Wise In Descending Order.

SQL - SELECT COUNT(seat_id), seat_type, price
FROM movieticket_db."Seat"
GROUP BY price, seat_type
ORDER BY price DESC

Output -



Total Tuples - 3

31. Create A View For Admin Details Then Insert And Display Some New Admin To The Recently Created View

SQL - CREATE VIEW movieticket_db."Admin_Details" AS
SELECT * FROM movieticket_db."Admin"

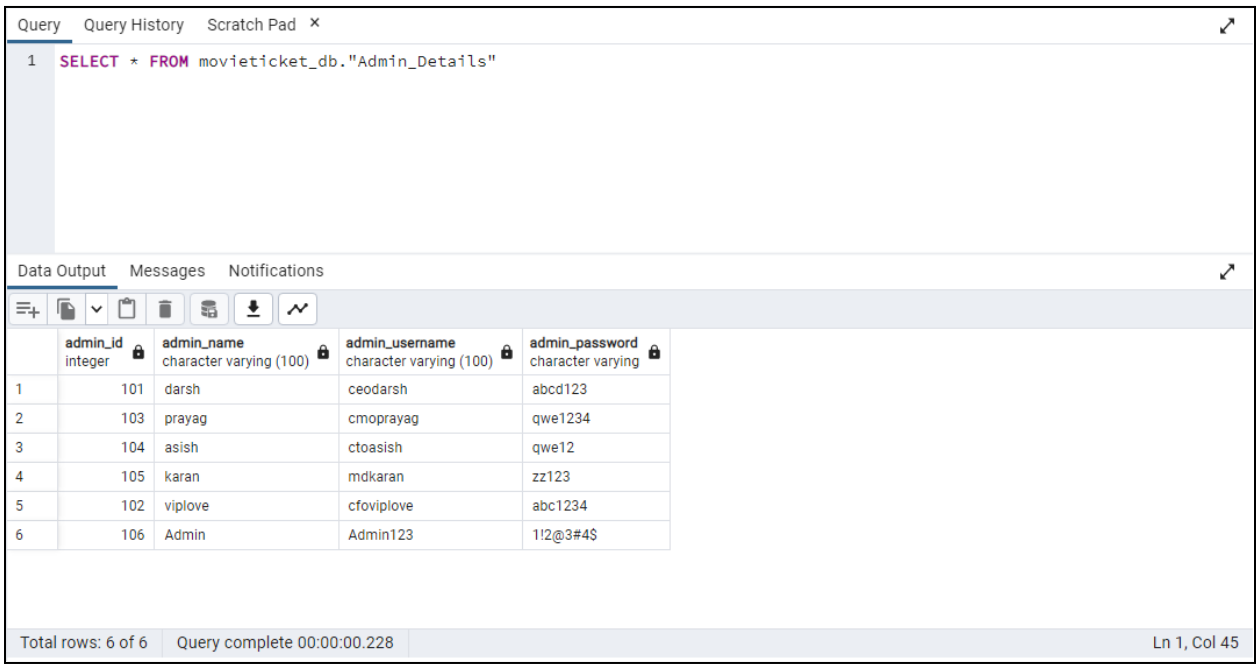
Output -

SQL - INSERT INTO movieticket_db."Admin_Details"
VALUES
(
(SELECT MAX(admin_id)+1 FROM movieticket_db."Admin_Details"),
'Admin',
'Admin123',
'1!2@3#4$'
)

Output -

SQL - SELECT * FROM movieticket_db."Admin_Details"

Output -



Total Tuples - 6

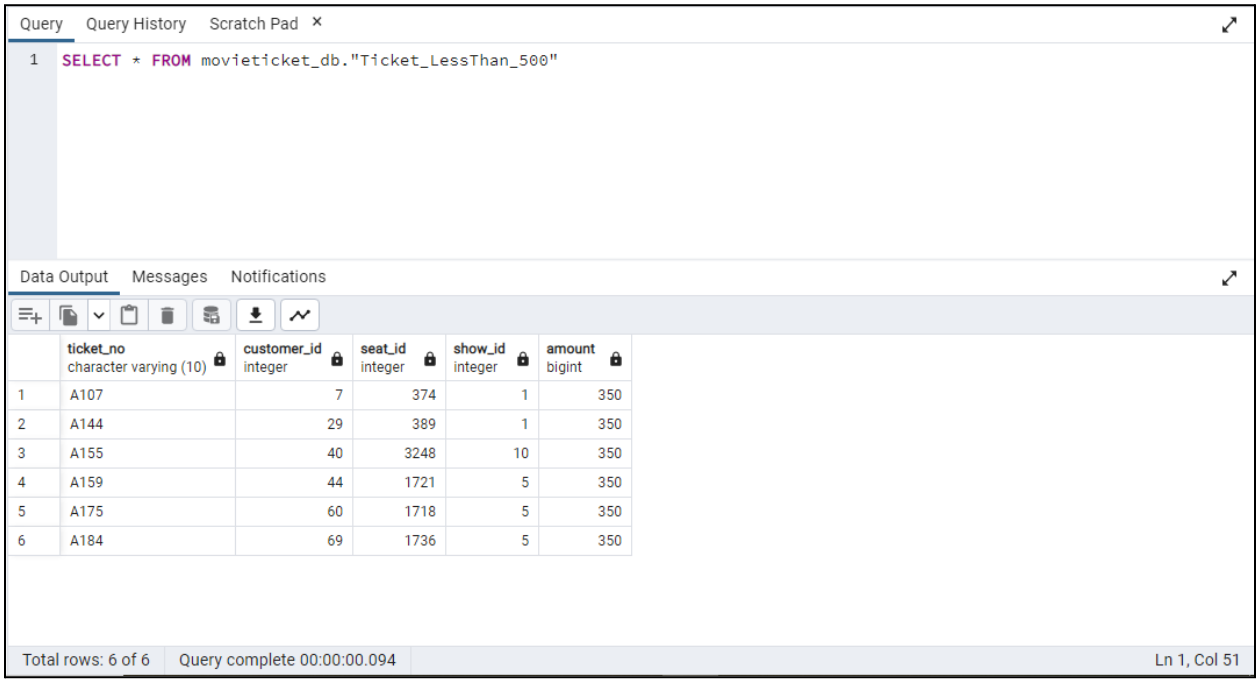
32. Create View Of Ticket Whose Amount Is Less Than 500

SQL - CREATE VIEW movieticket_db."Ticket_LessThan_500" AS
SELECT * FROM movieticket_db."Ticket"
WHERE amount<500
ORDER BY ticket_no ASC

Output -

SQL - SELECT * FROM movieticket_db."Ticket_LessThan_500"

Output -

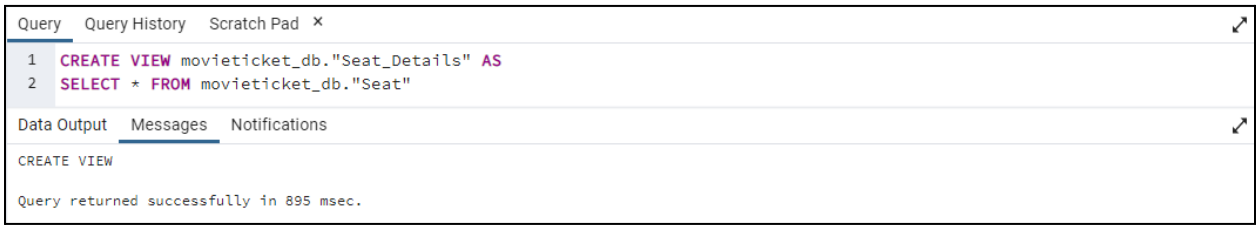


Total Tuples - 6

33. Create View Of All Seat Details

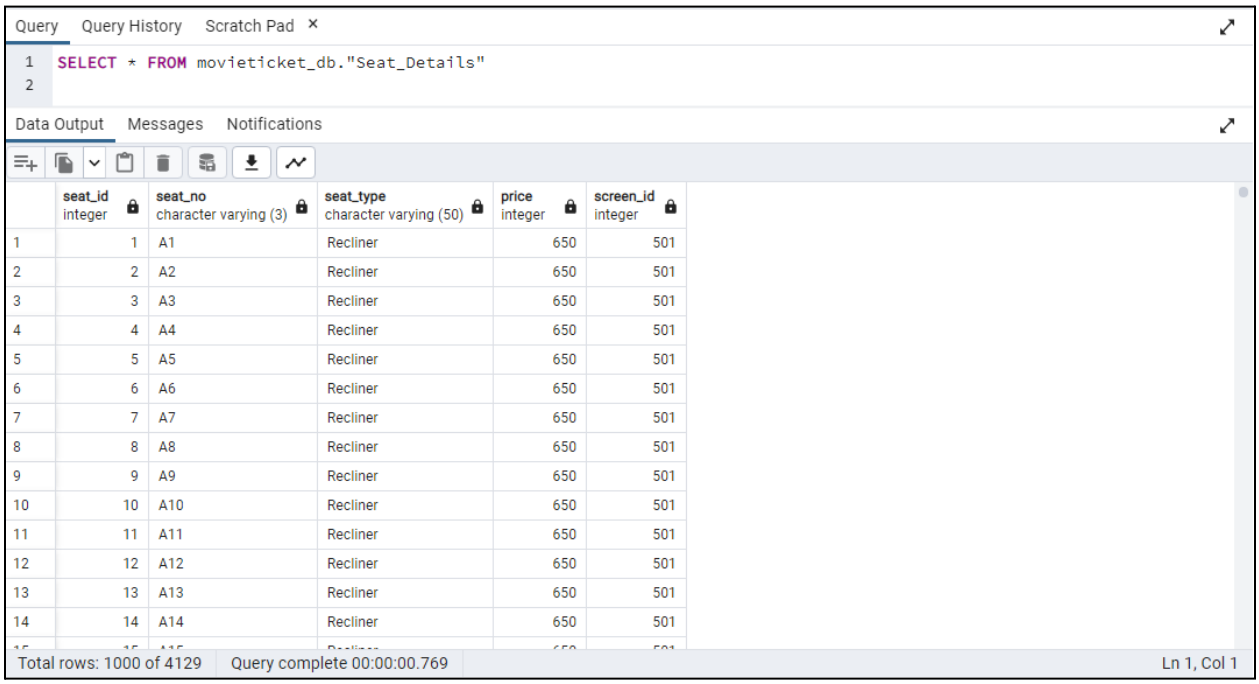
SQL - CREATE VIEW movieticket_db."Seat_Details" AS
SELECT * FROM movieticket_db."Seat"

Output -



SQL - SELECT * FROM movieticket_db."Seat_Details"

Output -



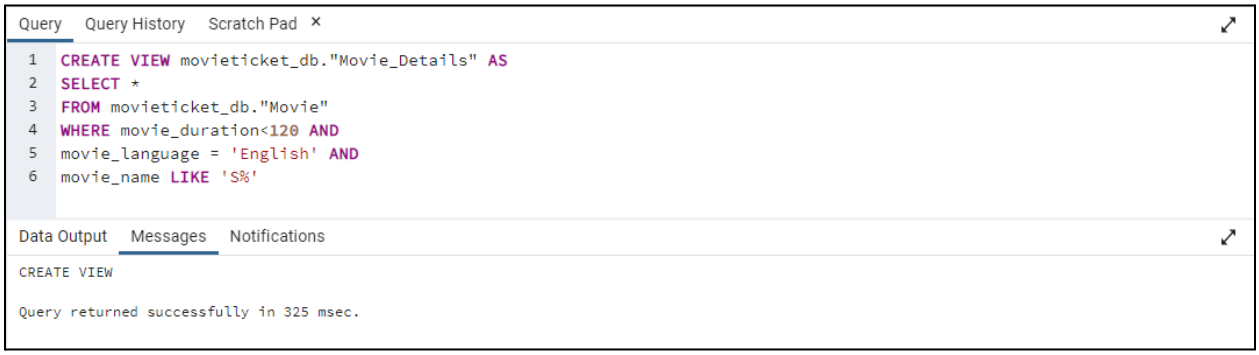
	seat_id integer	seat_no character varying (3)	seat_type character varying (50)	price integer	screen_id integer
1	1	A1	Recliner	650	501
2	2	A2	Recliner	650	501
3	3	A3	Recliner	650	501
4	4	A4	Recliner	650	501
5	5	A5	Recliner	650	501
6	6	A6	Recliner	650	501
7	7	A7	Recliner	650	501
8	8	A8	Recliner	650	501
9	9	A9	Recliner	650	501
10	10	A10	Recliner	650	501
11	11	A11	Recliner	650	501
12	12	A12	Recliner	650	501
13	13	A13	Recliner	650	501
14	14	A14	Recliner	650	501

Total Tuples - 4129

34. Create View Of Movie Where Duration Is Less Than 120 Minutes, Language Is English, And Movie Name Starts With ‘S’

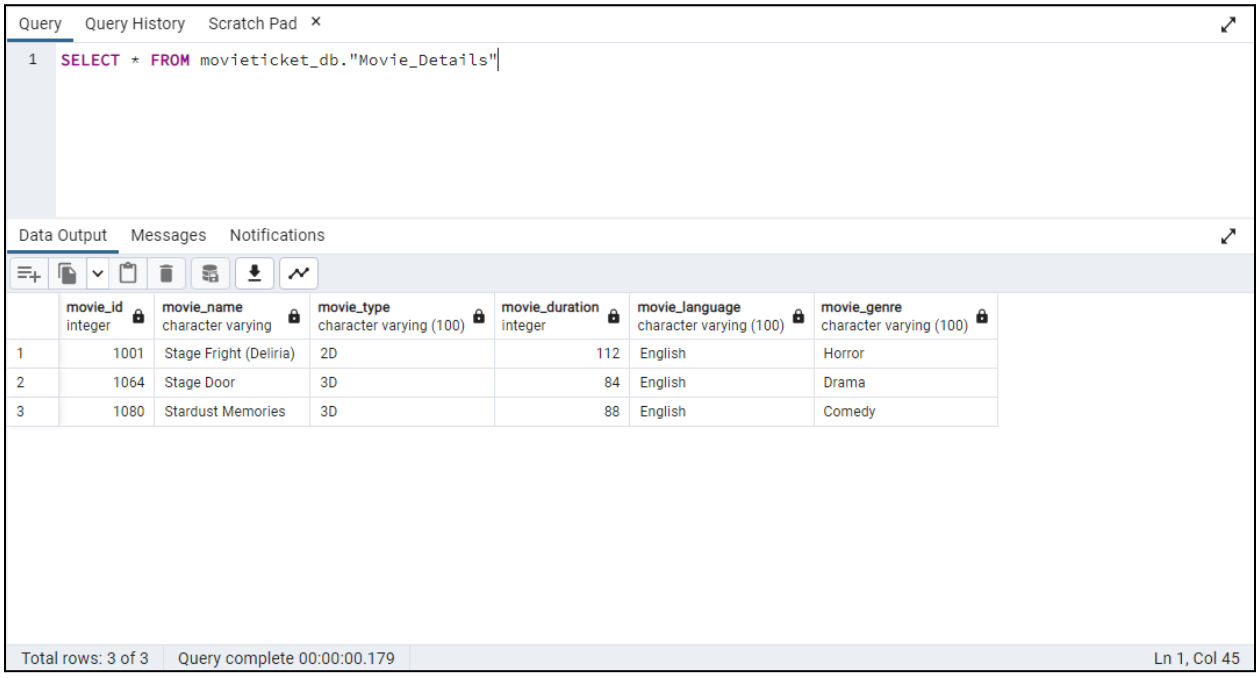
SQL - CREATE VIEW movieticket_db."Movie_Details" AS
SELECT *
FROM movieticket_db."Movie"
WHERE movie_duration<120 AND
movie_language = 'English' AND
movie_name LIKE 'S%'

Output -



SQL - SELECT * FROM movieticket_db."Movie_Details"

Output -

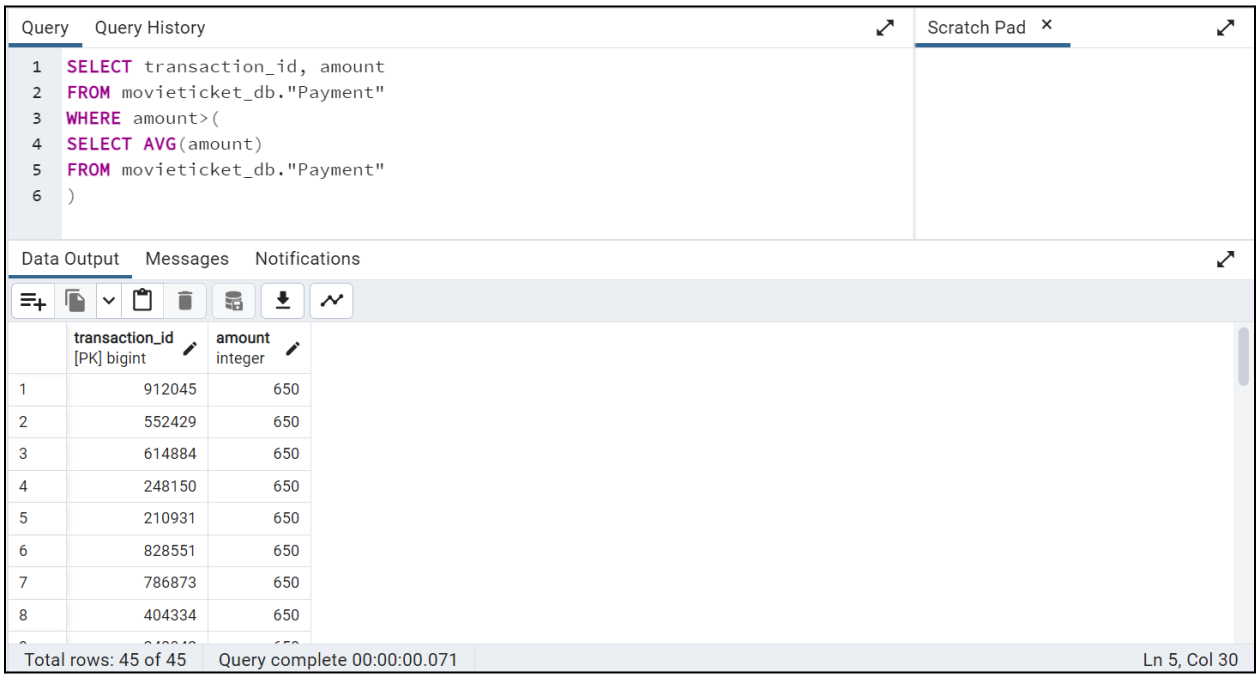


Total Tuples - 3

35. Display Transaction Ids And Their Amount Whose Amount Is Greater Than Average Amount Spent By User

SQL - SELECT transaction_id, amount
FROM movieticket_db."Payment"
WHERE amount>(
SELECT AVG(amount)
FROM movieticket_db."Payment"
)

Output -



Total Tuples - 45

36. Display The Movie Names Alongside Customer Name Which Are Booked By Customer Id 1

SQL - SELECT C.customer_name, M.movie_name
FROM movieticket_db."Ticket" T,
movieticket_db."Show" S,
movieticket_db."Movie" M,
movieticket_db."Customer" C
WHERE T.show_id = S.show_id AND
S.movie_id = M.movie_id AND
C.customer_id = T.customer_id AND
T.customer_id = 1

Output -

```
1  SELECT C.customer_name, M.movie_name
2  FROM movieticket_db."Ticket" T,
3  movieticket_db."Show" S,
4  movieticket_db."Movie" M,
5  movieticket_db."Customer" C
6  WHERE T.show_id = S.show_id AND
7  S.movie_id = M.movie_id AND
8  C.customer_id = T.customer_id AND
9  T.customer_id = 1
```

Data Output | Messages | Notifications

	movie_name character varying	customer_name character varying (100)
1	Stage Fright (Deliria)	Odessa Benson
2	Me and Orson Welles	Odessa Benson
3	Age of the Earth, The (A Idade da Terra)	Odessa Benson
4	Cherry Falls	Odessa Benson

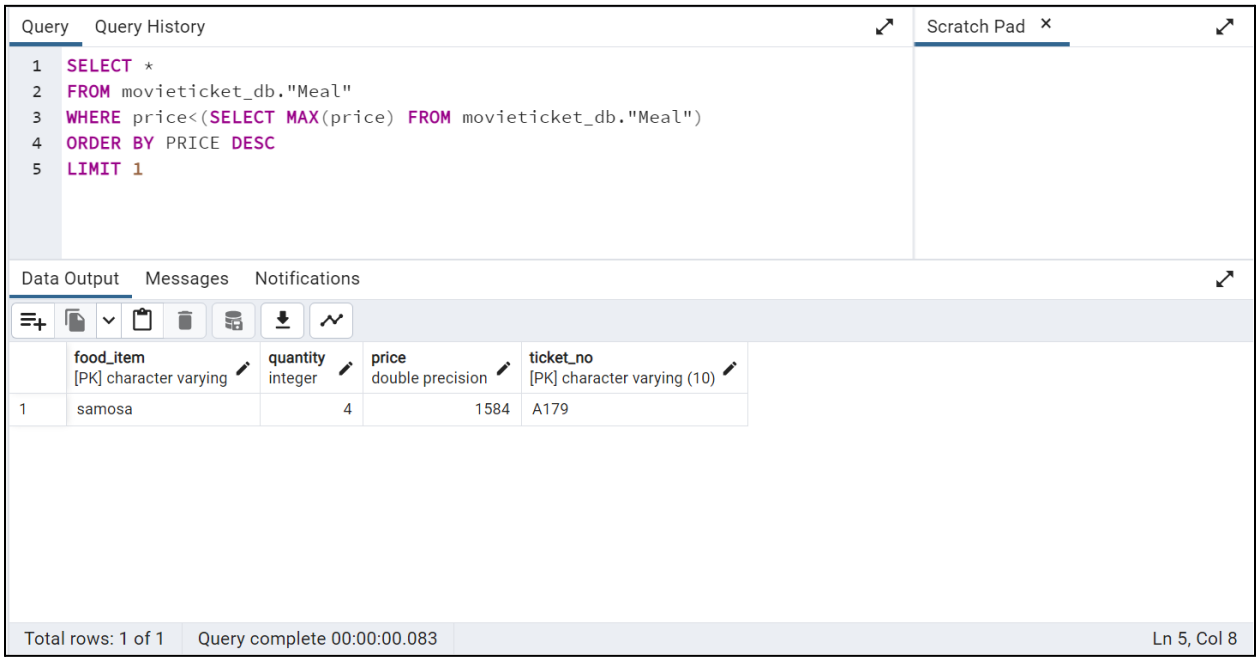
Total rows: 4 of 4 | Query complete 00:00:00.083 | Ln 1, Col 25

Total Tuples - 4

37. Display The Details Of The 2nd Highest Amount Paid For A Meal.

SQL - SELECT *
FROM movieticket_db."Meal"
WHERE price<(SELECT MAX(price) FROM movieticket_db."Meal")
ORDER BY PRICE DESC
LIMIT 1

Output -

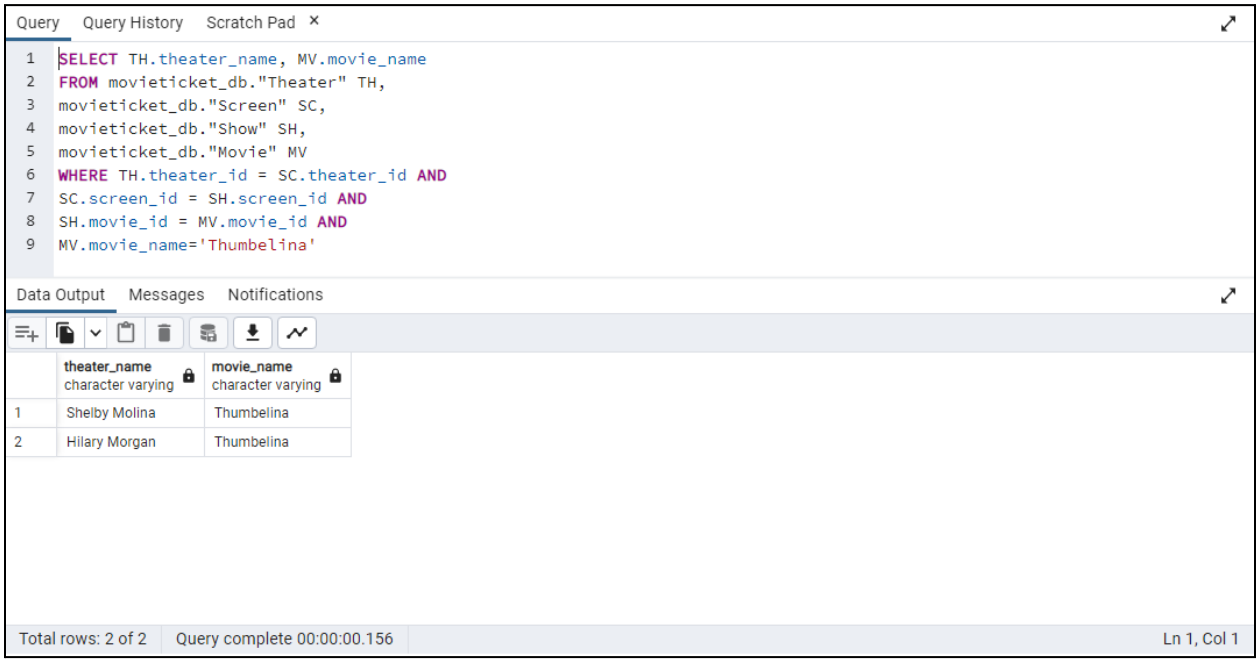


Total Tuples - 1

38. Display Theater Name With Movie Name Where Movie "Thumbelina" Is Running.

SQL - SELECT TH.theater_name, MV.movie_name
FROM movieticket_db."Theater" TH,
movieticket_db."Screen" SC,
movieticket_db."Show" SH,
movieticket_db."Movie" MV
WHERE TH.theater_id = SC.theater_id AND
SC.screen_id = SH.screen_id AND
SH.movie_id = MV.movie_id AND
MV.movie_name='Thumbelina'

Output -

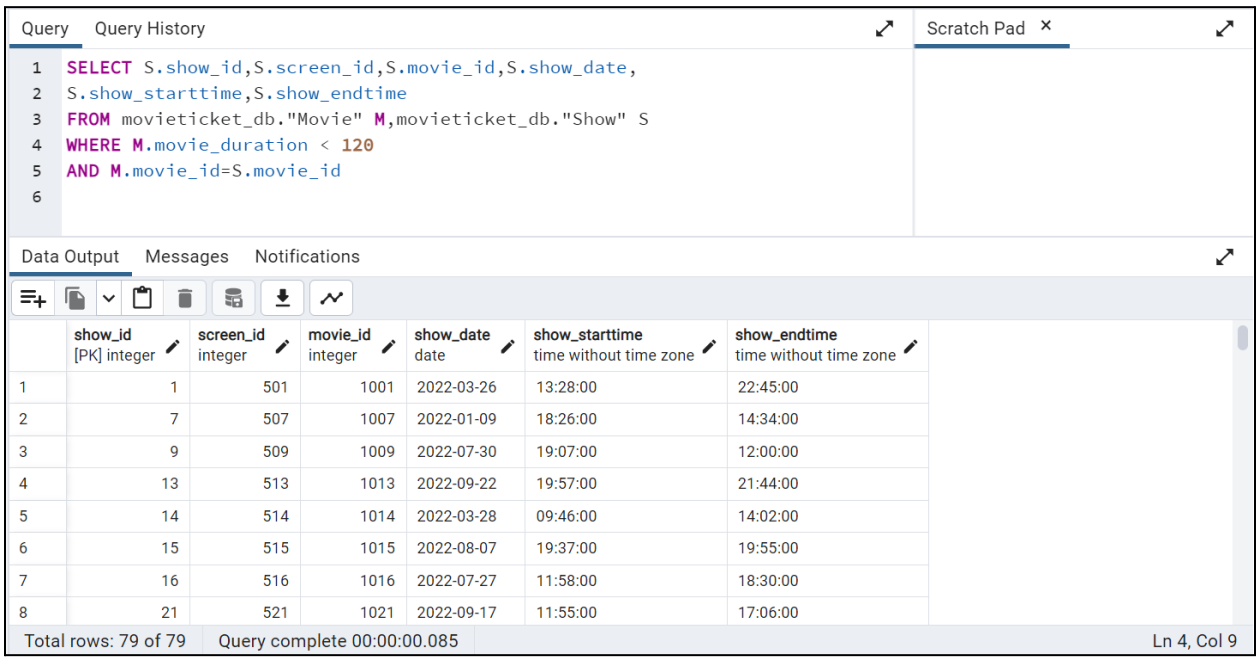


Total Tuples - 2

39. Display Show Details For The Movies Having Duration Less Than 120 Mins.

SQL - SELECT S.show_id,S.screen_id,S.movie_id,S.show_date,
S.show_starttime,S.show_endtime
FROM movieticket_db."Movie" M,movieticket_db."Show" S
WHERE M.movie_duration < 120
AND M.movie_id=S.movie_id

Output -



Query | Query History | Scratch Pad

```
1  SELECT S.show_id,S.screen_id,S.movie_id,S.show_date,
2  S.show_starttime,S.show_endtime
3  FROM movieticket_db."Movie" M,movieticket_db."Show" S
4  WHERE M.movie_duration < 120
5  AND M.movie_id=S.movie_id
6
```

Data Output | Messages | Notifications

	show_id [PK] integer	screen_id integer	movie_id integer	show_date date	show_starttime time without time zone	show_endtime time without time zone
1	1	501	1001	2022-03-26	13:28:00	22:45:00
2	7	507	1007	2022-01-09	18:26:00	14:34:00
3	9	509	1009	2022-07-30	19:07:00	12:00:00
4	13	513	1013	2022-09-22	19:57:00	21:44:00
5	14	514	1014	2022-03-28	09:46:00	14:02:00
6	15	515	1015	2022-08-07	19:37:00	19:55:00
7	16	516	1016	2022-07-27	11:58:00	18:30:00
8	21	521	1021	2022-09-17	11:55:00	17:06:00

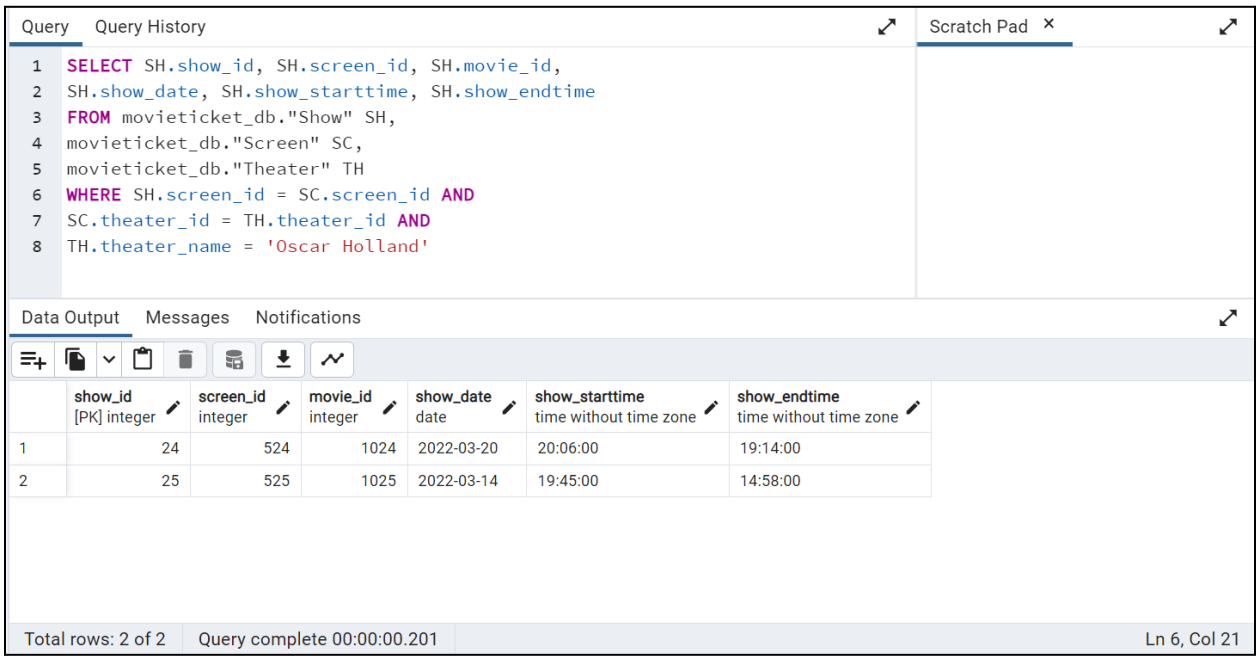
Total rows: 79 of 79 | Query complete 00:00:00.085 | Ln 4, Col 9

Total Tuples - 79

40. Display The Shows In Theater "Oscar Holland"

SQL - SELECT SH.show_id, SH.screen_id, SH.movie_id,
SH.show_date, SH.show_starttime, SH.show_endtime
FROM movieticket_db."Show" SH,
movieticket_db."Screen" SC,
movieticket_db."Theater" TH
WHERE SH.screen_id = SC.screen_id AND
SC.theater_id = TH.theater_id AND
TH.theater_name = 'Oscar Holland'

Output -



Total Tuples - 2