**Student ID**: 20232004

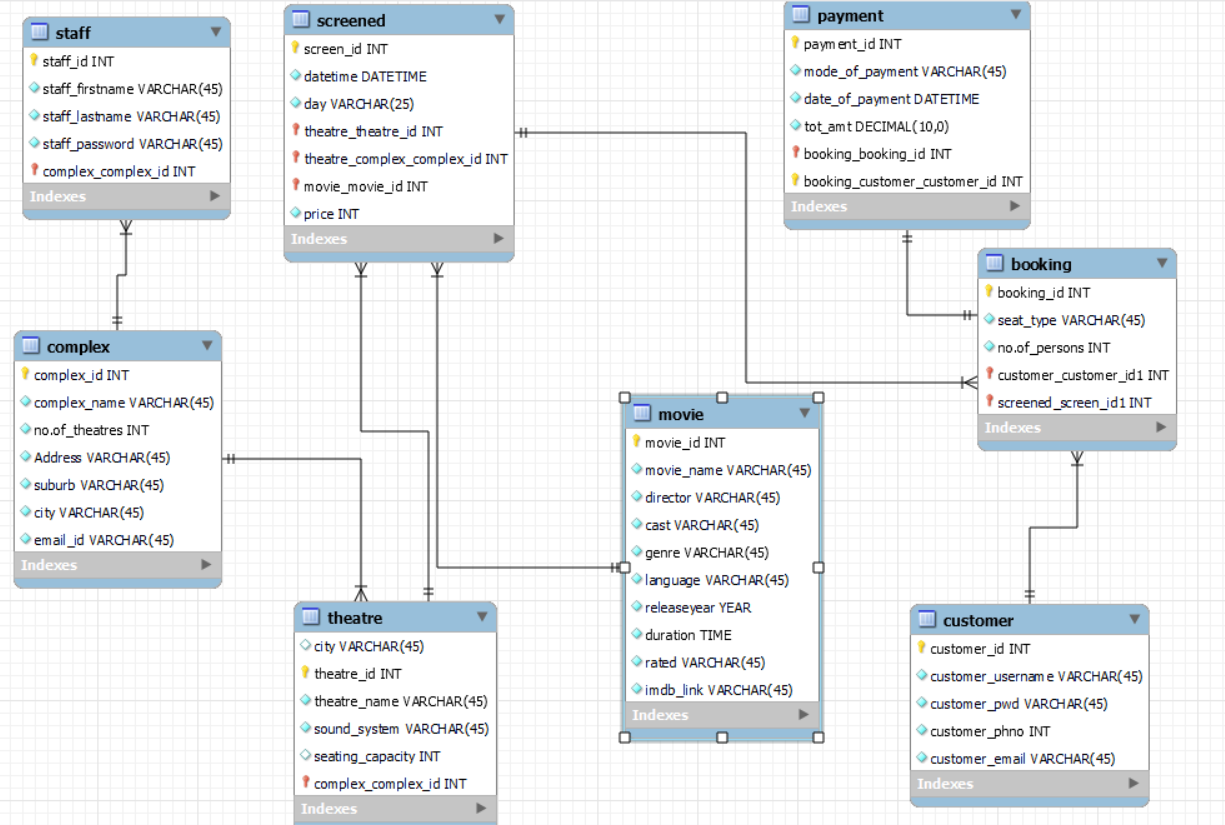
**Name**: Sudharshiya Ganesan

**IT5015 Assessment 2** – Application of Information systems to Business Brief.

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- **PART 1. Design the Database**

**TASK1:**

**1.1 Final Erd**



**1.2 LIST ENTITIES AND DESCRIBE THEM**

The entities used to create ERD (Entity Relationship Diagrams) are:

1. Complex- The entity Complex is the cinema complexes which are in the country under Eyecandy cinemas. Each Cinema Complex has its name, address, Suburb, No. of theatres and other necessary fields. Attributes are Complex\_Id, Complex\_name, No.of\_theatres, Address, Suburb, City,

email\_id

2. Staff- The employee has a separate login to operate the website with a secure password.

Attributes: staff\_Id, staff\_firstname, staff\_lastname, staff\_password, complex\_Id

3. Customer-It contains all the customers’ details with their username and password.

Attributes: customer\_id, customer\_username, customer\_pwd, customer\_phno, customer\_email.

4. Theatre- There are a number of theatres in different complexes. Each theatre has

different name seating capacity and sound system.

Attributes: Theatre\_Id, Theatre\_name, sound\_system, seating\_capacity.

5. Movie- Movie is the entertaining source for the user to relax. It has all the movie information such as moviename, cast, director, duration, rated classification, IMDB link to check the rating.

Attributes: movie Id, movie\_name, director, cast, duration, cast, genre, language, release

year, rated, IMDB link

6. Screened- The movies that are scheduled to be screened in theatres at time, Date and day will be displayed.

Attributes: screen\_Id, datetime, day, price

7. Booking- The booking of the tickets by movie title, time, theatre and price details are available

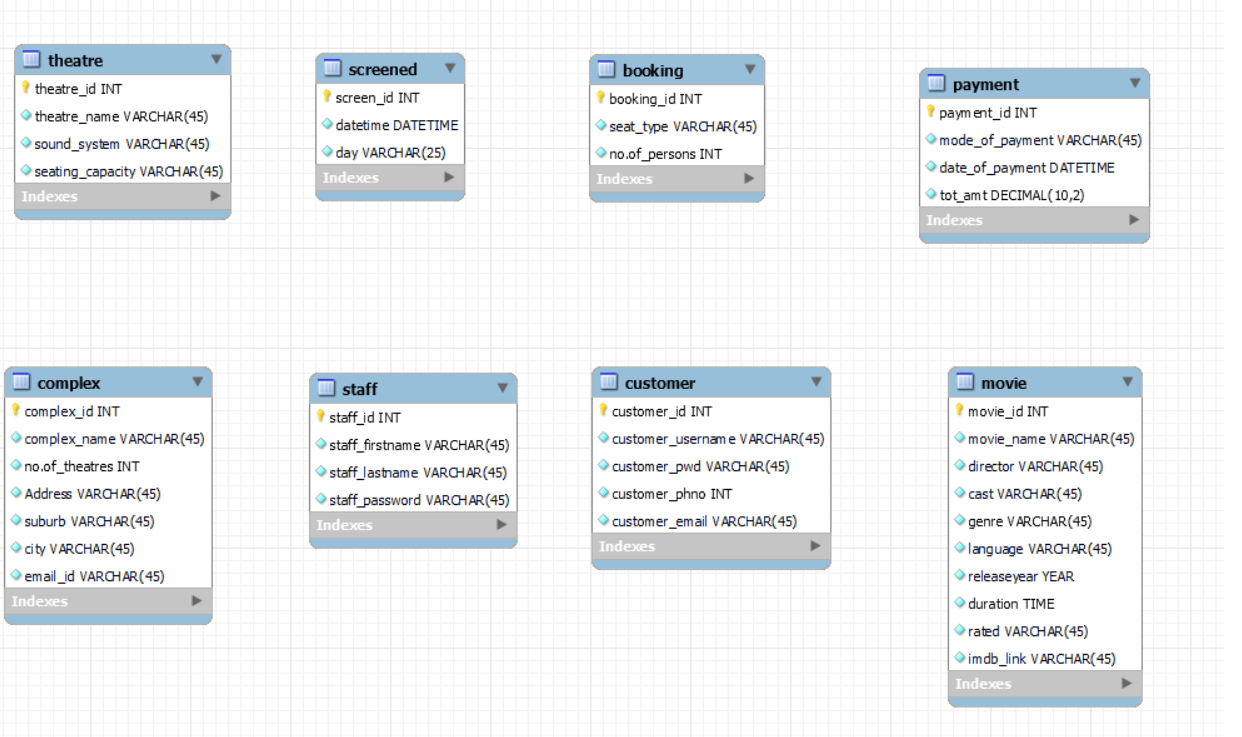
Attributes: booking\_id, no.of\_persons, seat\_type.

8. Payment- The payment by different modes is enabled in this entity.

Attributes: Payment\_ID, booking id, mode\_of\_payment,

date\_of\_payment, tot\_amt.

**1.3 List the attributes of each entity, identifying the datatype and constraints, choose primary key**



I made changes with the datatype of seating capacity in the table theatre.

**1.4 Relationship and referential constraints:**

**1. Relationship between complex and theatre:**

One-to-many relationships between complex and theatre because there are many theatres in each complex. So complex\_id which is the primary key of complex is the foreign key of theatre under referential constraints.

**2. Relationship between theatre and screened:**

One-to-many relationships between theatre and screened because one theatre is used to screen movies at different schedules. So, theatre\_id which is the primary key of theatre is the foreign key of screened table under referential constraints.

* One Theatre can have multiple movie schedules.
* Each movie schedule is associated with exactly one Theatre.

**3. Relationship between movie and screened:**

One-to-many relationships between movie and screened because one movie is used to screened at different schedules. So, movie\_id which is the primary key of movie is the foreign key of screened table under referential constraints.

* One Movie can have multiple schedules.
* Each Movie schedule is associated with exactly one Movie.

**4. Relationship between complex and staff:**

One-to-many relationships between complex and staff because each complex has many staffs. So, complex\_id which is the primary key of table complex is the foreign key of staff table under referential constraints.

**5. Relationship between screened and booking:**

One-to-many relationships between screened and booking because each movie schedule is booked multiple times by different customers. So, screen\_id which is the primary key of screened is the foreign key of booking table under referential constraints.

**6. Relationship between customer and booking:**

One-to-many relationships between customer and booking because one customer can make many bookings in the same complex at different times. So, customer\_id which is the primary key of customer is the foreign key of booking table under referential constraints.

* One Customer can have multiple bookings.
* Each Booking is associated with exactly one Customer.

**7. Relationship between booking and payment:**

One-to-one relationship between booking and payment because each booking leads to one payment. So, booking\_id which is the primary key of booking is the foreign key of payment table under referential constraints.

* Each Booking is associated with exactly one Payment.
* Each Payment is associated with exactly one Booking.

**1.5 Meet Normalization.**

**1. First Normal Form (1NF):**

* Ensures that each column contains atomic (indivisible) values and there are no repeating groups or arrays of data.

Normalization is done by staff name is divided into first name and last name in table ‘STAFF’

**2. Second Normal Form (2NF):**

* Builds on 1NF and eliminates partial dependencies. It ensures that non-prime attributes are fully functionally dependent on the primary key.

Removed city column from theatre since we have complex\_id as foreign key

**3. Third Normal Form (3NF):**

* Builds on 2NF and eliminates transitive dependencies. It ensures that non-prime attributes are not transitively dependent on the primary key.

All the contents of the table satisfy 1NF, 2NF, 3NF and it has the necessary data in the table.

TASK2: DESIGN DATA QUERIES

2.1 QUERIES:

|  |  |
| --- | --- |
| Function | Queries |
| Search for Cinema | searchCinemaByCity  searchTheatreByCityAndCapacity  searchTheatreByCityAndSoundSystem |
| Show Cinema Information | Search theatre\_name, sound\_system by complex\_city  Search theatres by their suburbs |
| Search Movie screening | Search movie by theatre Id, movie name and genre  Search movie by movie\_datetime, price |
| Show Movie Schedule | Search movie by movie name, theatre name, **day**  Search movie by **date**, theatre name, movie name |
| Monitor Movie Screenings | Search screening BY DATE  Search screening movie BY PRICE |

REVIEW YOUR DATABASE:

All the tables and data content are properly reviewed

**TASK3: CREATE DATA CONTENT**

1.Entity name: complex

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **complex\_id** | **complex\_name** | **no.of\_theatres** | **Address** | **suburb** | **city** | **email\_id** |
| 1 | Sparklight | 5 | 25, Larch Road | Devonport | Auckland | [sparklightbook@hotmail.co.nz](mailto:sparklightbook@hotmail.co.nz) |
| 2 | Sunlight | 4 | 29, Pokere road | Crofton Downs | Wellington | [sunlight@hotmail.co.nz](mailto:sunlight@hotmail.co.nz) |
| 3 | Spotlight | 4 | 15, Richmond road | Hillmorton | Christchurch | [spotlight@hotmail.co.nz](mailto:spotlight@hotmail.co.nz) |
| 4 | Sparkcity | 3 | 32, Nelson road | Frankton | Hamilton | [sparkcity@hotmil.co.nz](mailto:sparkcity@hotmil.co.nz) |
| 5 | Sparkpoint | 2 | 48, Fairlie road | Glenorchy | Queenstown | [sparkpoint@hotmail.co.nz](mailto:sparkpoint@hotmail.co.nz) |

2. Entity name: Theatre

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Theatre\_id** | **Theatre\_name** | **Sound\_system** | **Seating\_capacity** | **Complex\_id** |
| 101 | Starlight Showcase | SonicSphere | 250 | 1 |
| 201 | CineSpectra | Imax | 200 | 2 |
| 301 | VelvetVision | 4DX | 150 | 3 |
| 401 | Prism playhouse | Dolby | 250 | 4 |
| 501 | Galaxy Grand | screenx | 200 | 5 |

3. Entity name: movie

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Movie\_id** | **Movie\_name** | **director** | **cast** | **genre** | **language** | **Release\_year** | **duration** | **rated** | **Imdb\_link** |
| 1 | The Good Mother | Miles Joris-Peyrafitte | Hilary Swank | crime | English | 2023 | 01:29:00 | R | <https://www.imdb.com/mother> |
| 2 | The Hill | Jeff Celentano | Jeff Celentano | Drama | English | 2023 | 02:06:00 | PG | <https://www.imdb.com/hill> |
| 3 | Gran Turismo | Neill Blomkamp | David Harbour | Action | English | 2023 | 02:14:00 | PG-13 | <https://www.imdb.com/gran> |
| 4 | The Girl and the Spider | Ramon Zürcher | Ramon Zürcher | Drama | German | 2021 | 01:38:00 | R | <https://www.imdb.com/spider> |
| 5 | Animal | Sandeep Reddy Vanga | Ranbir Kapoor | Action | Hindi | 2023 | 03:22:00 | PG-13 | <https://www.imdb.com/animal> |

4. Entity: Screened

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen\_id** | **datetime** | **price** | **day** | **theatre\_id** | **Complex\_id** | **Movie\_id** |
| 1001 | 2023-12-22 10:34:23 | 20 | Friday | 101 | 1 | 1 |
| 1002 | 2023-12-23 16:30:30 | 20 | Saturday | 201 | 2 | 2 |
| 1003 | 2023-12-24 11:30:30 | 25 | Sunday | 301 | 3 | 4 |
| 1004 | 2023-12-25 20:30:30 | 25 | Monday | 401 | 4 | 3 |
| 1005 | 2023-12-26 12:30:30 | 25 | Tuesday | 501 | 5 | 5 |

5. Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer\_Id** | **Customer\_username** | **Customer\_pwd** | **Customer\_phno** | **Customer\_email** |
| 2023100 | Jerry | abc123 | 2787878 | [jerry@gmail.com](mailto:jerry@gmail.com) |
| 2023101 | shirley | sh2587 | 2789456 | [shirley@gmail.com](mailto:shirley@gmail.com) |
| 2023102 | mathew | math122 | 2564785 | [mathew@hotmail.co.nz](mailto:mathew@hotmail.co.nz) |
| 2023103 | manvir | singh478 | 2874512 | [manvir@gmail.com](mailto:manvir@gmail.com) |
| 2023104 | Hilda | Hil4598 | 2789487 | [hilda@gmail.com](mailto:hilda@gmail.com) |

6. Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Staff\_id** | **Staff\_firstname** | **Staff\_lastname** | **Staff\_password** | **Complex\_id** |
| 1000100 | Anne | green | green123 | 1 |
| 2000100 | Jose | Lee | Lee345 | 2 |
| 3000100 | Justin | White | White564 | 3 |
| 4000100 | Nicole | Christine | Nic58798 | 4 |
| 5000100 | Robert | Western | Robert200 | 5 |

7. Booking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Booking\_id** | **Seat\_type** | **No.of\_persons** | **Customer\_id** | **Screen\_id** |
| 1000111 | recliner | 8 | 2023100 | 1001 |
| 1000112 | special | 5 | 2023101 | 1002 |
| 1000113 | normal | 2 | 2023102 | 1003 |
| 1000114 | deluxe | 2 | 2023103 | 1004 |
| 1000115 | recliner | 2 | 2023104 | 1005 |

8.payment

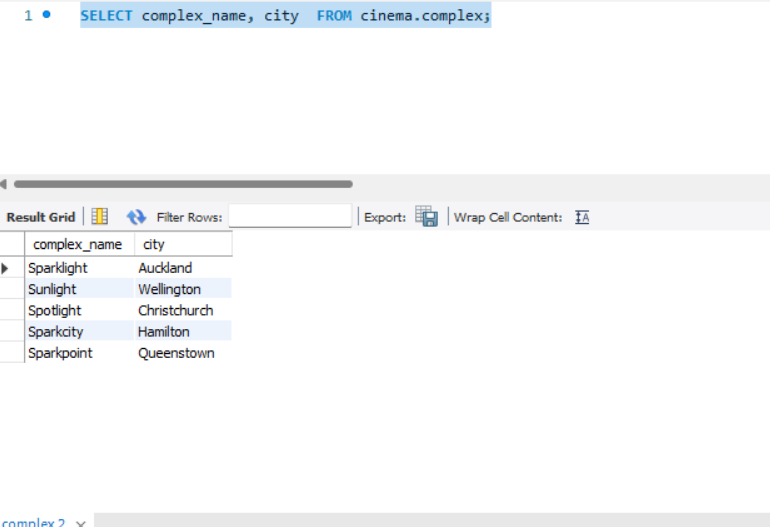
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Payment\_ID** | **Mode\_of\_payment** | **Date\_of\_payment** | **Tot\_amt** | **Booking\_id** | **Customer\_id** |
| 12601 | credit | 2023-12-02 10:34:23 | 200.00 | 1000111 | 2023100 |
| 12602 | debit | 2023-12-03 04:34:23 | 150.00 | 1000112 | 2023101 |
| 12603 | credit | 2023-12-04 14:34:23 | 100.00 | 1000113 | 2023102 |
| 12604 | debit | 2023-12-05 16:34:23 | 50.00 | 1000114 | 2023103 |
| 12605 | debit | 2023-12-02 18:34:23 | 50.00 | 1000114 | 2023104 |

**PART 2: DATABASE IMPLEMENTATION:**

1. FUNCTION- SEARCH FOR CINEMA

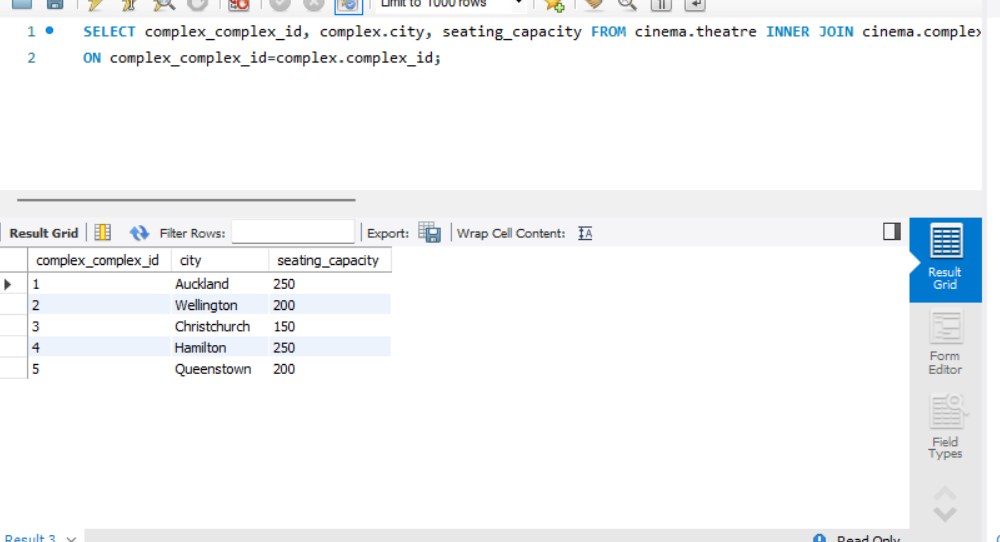
QUERY 1: Search by city

* Testing the search functionalities with valid city names
* Ensure that the search returns relevant cinemas based on the provided city.
* Ensure that the search functionality is user-friendly.
* Document the query requirements, implementation details, and testing procedures for future reference.



**QUERY2: Search by theatre capacity, city and complex\_Id**

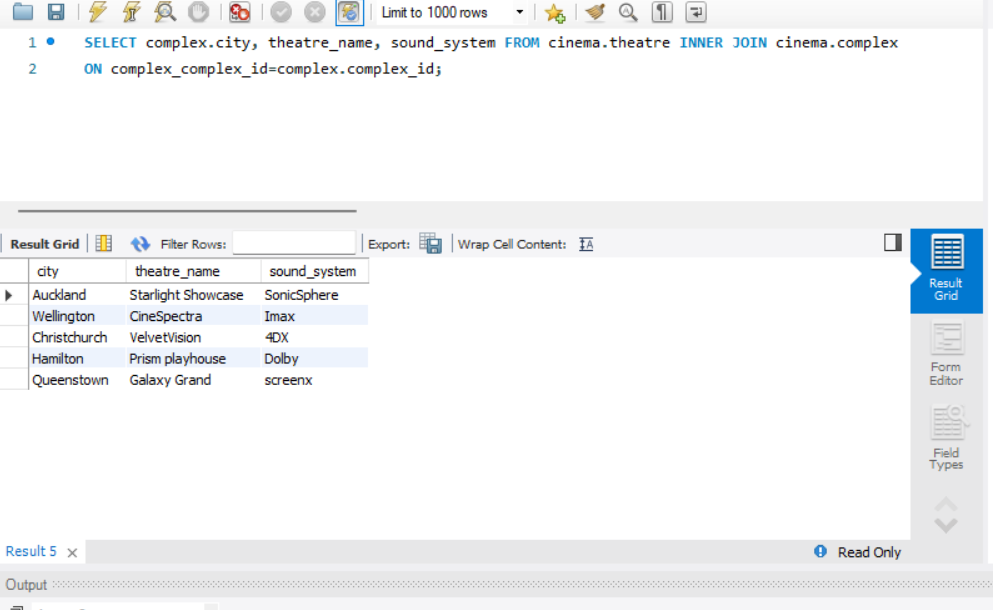
* Verify that the system correctly displays the capacity of theaters in a given city.
* Test with different cities and ensure the capacity information is accurate.
* Check for theaters with varying capacities.
* Manually execute the search for theater capacity in various cities and verify the correctness of the results.
* Check for a clear presentation of data, including theater names, capacities, and other relevant details.



**FUNCTION2: Show Cinema Information.**

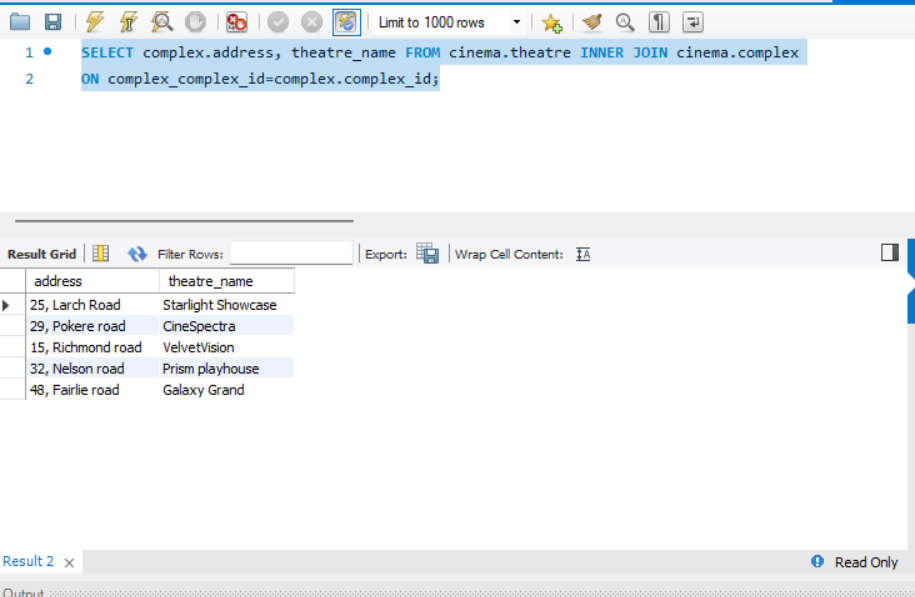
**Query3: Search theatre\_name, sound\_system by complex\_city.**

* Users should be able to search for cinema information based on theater name and sound system.
* The search should be case-insensitive.
* Partial matches for theater names should be considered.
* Test the search functionality with valid theater names and sound system specifications.
* Ensure that the search returns to relevant cinemas based on the criteria provided.



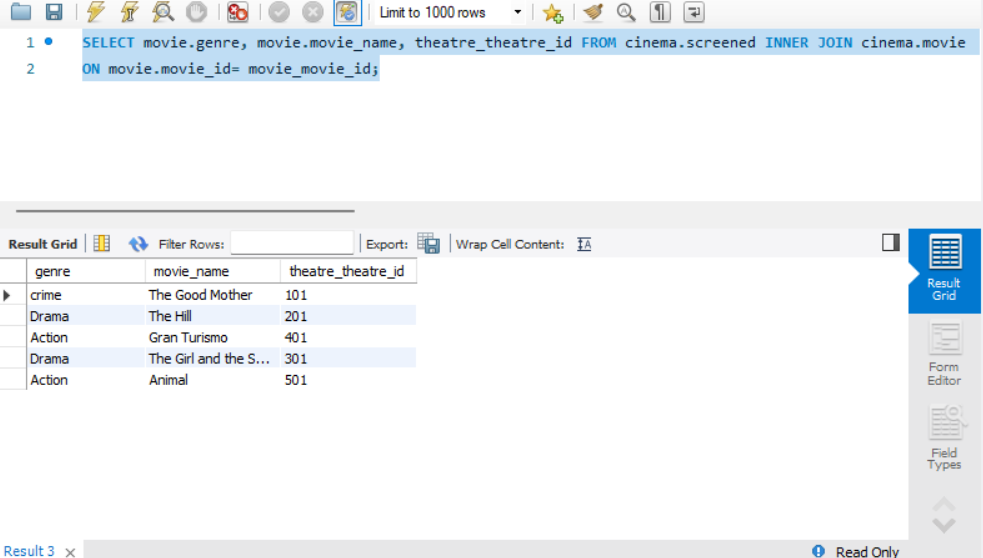
**Query 4: Search theatre name by Address**

* Test with valid theater names and ensure that the search returns the correct theater names along with their addresses.
* Test with invalid theater names or non-existent theaters to ensure the system handles errors gracefully.
* Verify that the information is presented in a user-friendly way.
* Ensure that users can easily understand and navigate the displayed theater names and addresses.

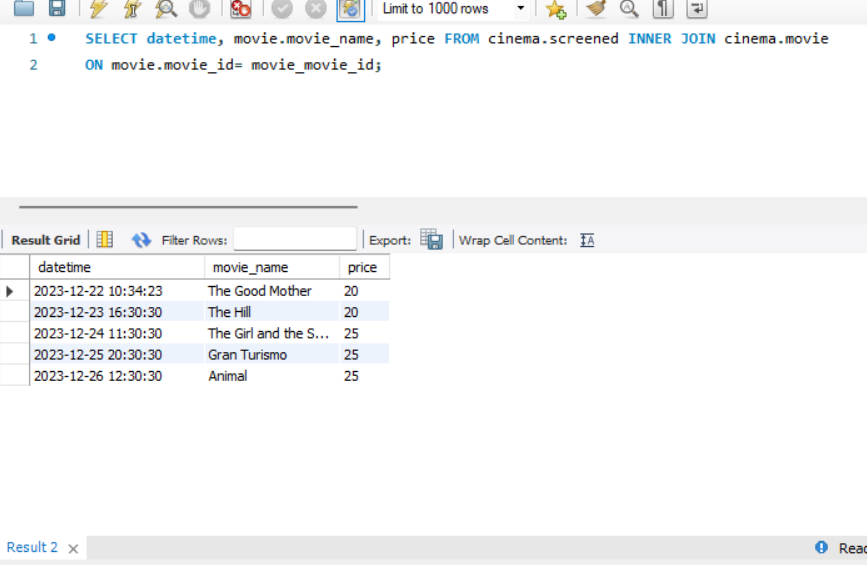


**FUNCTION3: SEARCH MOVIE SCREENING:**

**QUERY 5: SEARCH MOVIE BY theatre Id, movie name and genre**

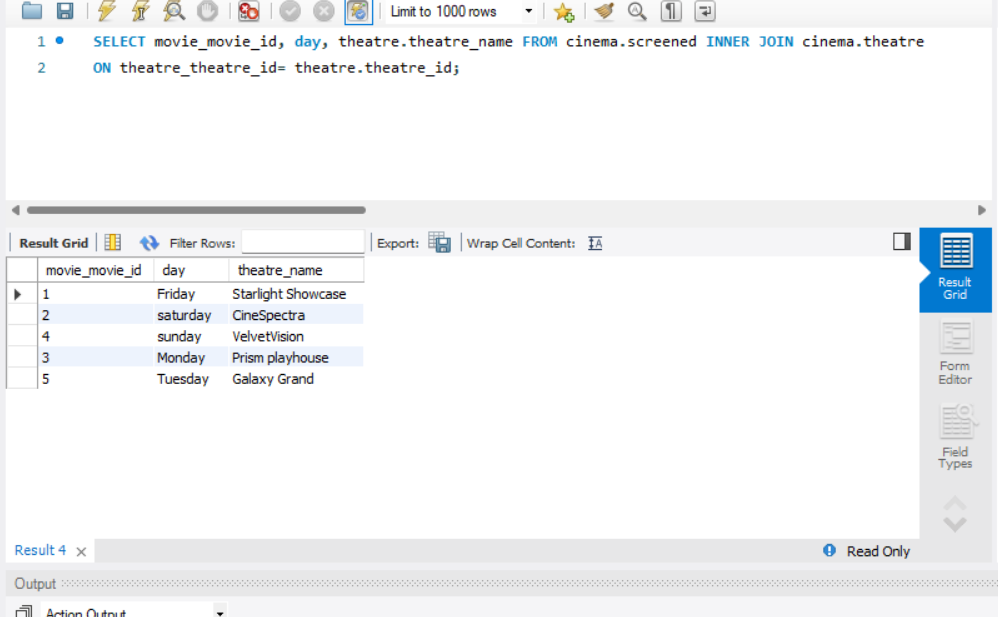


**QUERY 6: SEARCH BY MOVIE NAME, DATETIME, PRICE**

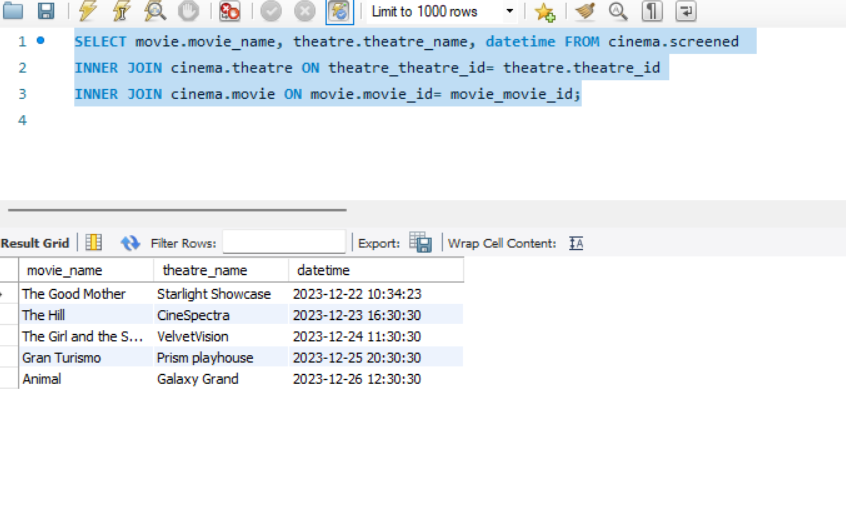


**FUNCTION 4: MOVIE SCHEDULE**

**Query 7: search movie by movie Id, theatre name and day it is scheduled to screen.**

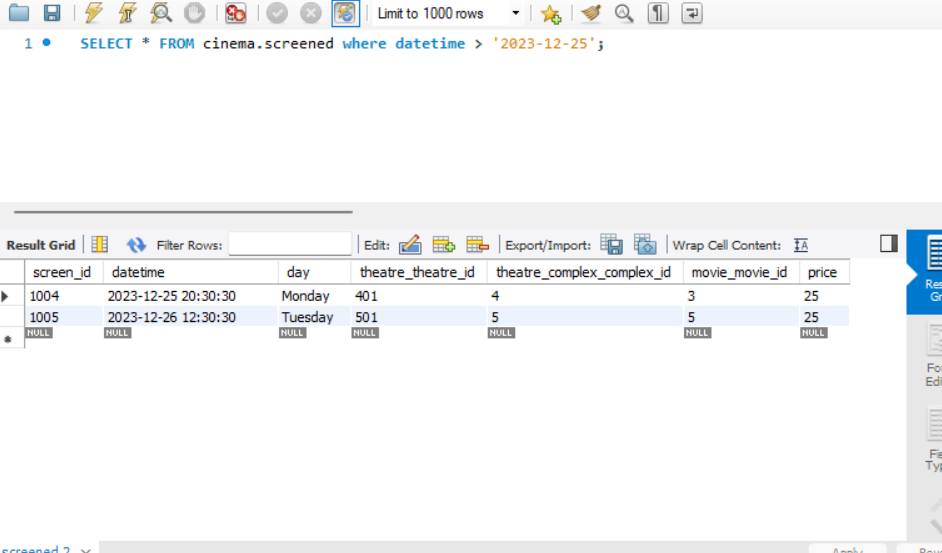


**Query 8: select movie name by theatre name and datetime**



**Function5: Monitor movie screening**

**Query 9: check movie is screening on all dates**



Query10: check the movies can be viewed by shortlisting price.

