**CM3010 Databases and Advanced Data Techniques Midterm**

**Stage 1: Find and critique a dataset**

I had chosen the cinema tickets dataset as the field for my analysis, there were many other datasets on this topic but I chose this specific dataset because the data file is well structured and it has all the appropriate data which is needed for my analysis. There are a variety of datasets to pick from, including datasets with diverse features, varying scopes, and varying sizes in this. In addition, this dataset contains eight months’ worth of sales history for various theatres in the year 2018, together with comprehensive screening data and encoded anonymized locations. This dataset was found on an open website called Kaggle.

The cinema industry is growing day by day and sales related to this are also increasing incredibly. Many people love watching movies in cinema’s irrespective of their language, genre, actors, etc. The film business is not exempt from benefiting from predictive modelling. Sale forecasts may assist theatres in cost reduction and improved ROI, just like other industries like retail, banking, and restaurants. To attract more people the movie should have good actors, good screenplay and direction. As long as individuals who watch it maintain a balance between what they believe in and what the film may be trying to shove down their throats, the cinema may continue to be a blessing for us. The purpose of watching movies should be for pure amusement and enlightenment.

From my analysis I would like to know the answers for some of the important questions that I am keen to know from this dataset. One of my main question is to find the month with highest ticket sales, as I have observed a relational connection between these columns. The next question I would like to ask is how is the ticket sales on each day in a month.

The last question I want to know is the average occupancy in each month.

***Questions:***

1. Which month had the highest ticket sales?
2. How are the ticket sales on each day in a month?
3. What is the average occupancy per each month?

From the data which available for me, to find the show timings affecting the cinemas I will compare the show time and the sale columns, by doing this we can get a clear vision and result for my question. Similarly, I will follow the same task for the other questions too. From the data which is there in the dataset I can do this easily because the dataset has all the needed values for doing this analysis.

**Terms of Use:**

I have found this clean dataset containing all the required information from the open website called Kaggle. This dataset has 10 columns which are film code, cinema code, total sales, tickets sold, show time, ticket price, month, quarter, date, day, capacity of the sales. This dataset is accessible anytime by anyone from anywhere.

This is a licensed dataset which means the people who wants to use this has full rights to refer the data from this dataset. It was licensed by “**CC BY-NC-SA 4.0**”. This dataset was uploaded 2 years ago by the author named “**Möbius**“. He is a data scientist at a healthcare centre in Melbourne, Victoria, Australia. He has 31 as his current rank amongst all of the other 77,500 authors on this website. Most of his datasets are being well used by students, and people.

In terms of use I think licensing a data is a very good thing to do because by doing this it means you are keeping your dataset safe from people stealing it or using it without the author’s permission which is considered to be an illegal action.

**Assessing the Dataset:**

***Quality:***

The website Kaggle, which offers a variety of open-source datasets, is where the data originated from. The data that Kaggle offers is openly accessible and free. This website has approximately 50,000 free datasets which can be used by students, researchers and general people for their projects and presentations. This website has all kinds of datasets related to sales, computer science, education, population, etc. Irrespective of the dataset selected it has all the required codes, discussions, csv files, context and description of the dataset. Even general public can use this platform to upload their data.

***Level of Detail:***

For the level of details, I completely agree that this dataset provides all the important data and information for the analysis. We can clearly see the sales and profit of the sales of the cinema tickets in the dataset as it is evidently shown based on all of the aspects. This let me in recognising the most effective factors amongst the columns and get a clear vision of my analysis. I selected this dataset, which prompted me to analyse it in order to discover the answers to the issues raised above.

***Documentation:***

The attributes provided in the dataset file were easily understandable for the analysing the data. Name of the two or three heading for the columns were not mentioned clearly which I deleted it from the file which were not useful for the analysis. The dataset's content was clear and easy to interpret, therefore no additional study was necessary.

***Interrelation:***

Each attribute in my dataset file was very well interrelated. No additional dataset was needed to be integrated because the dataset has characteristics with rich data. We can do the analysis easily as these attributes are interrelated and in addition as the attributes are interrelated it is easy for me to find the answer to the above asked questions.

***Use:***

The dataset is now utilized to do exploratory data analysis to comprehend cinema ticket sales.

Through machine learning, the dataset can be used to predict future sales or profits. For making the analysis clearer I eliminated the unwanted data and columns which will help in making the dataset more understandable.

***Discoverability:***

Substantially for my chosen field of “Sales” it is easy to get a open information from datasets or websites, due to the variety of sources offering datasets related to the field, there was a diversity of scope of data. A variety of other datasets were also available, such as ticket sales by different movie genres but I think analysing the overall ticket sales is better than analysing only on particular section.

**Stage 2: Model your data**

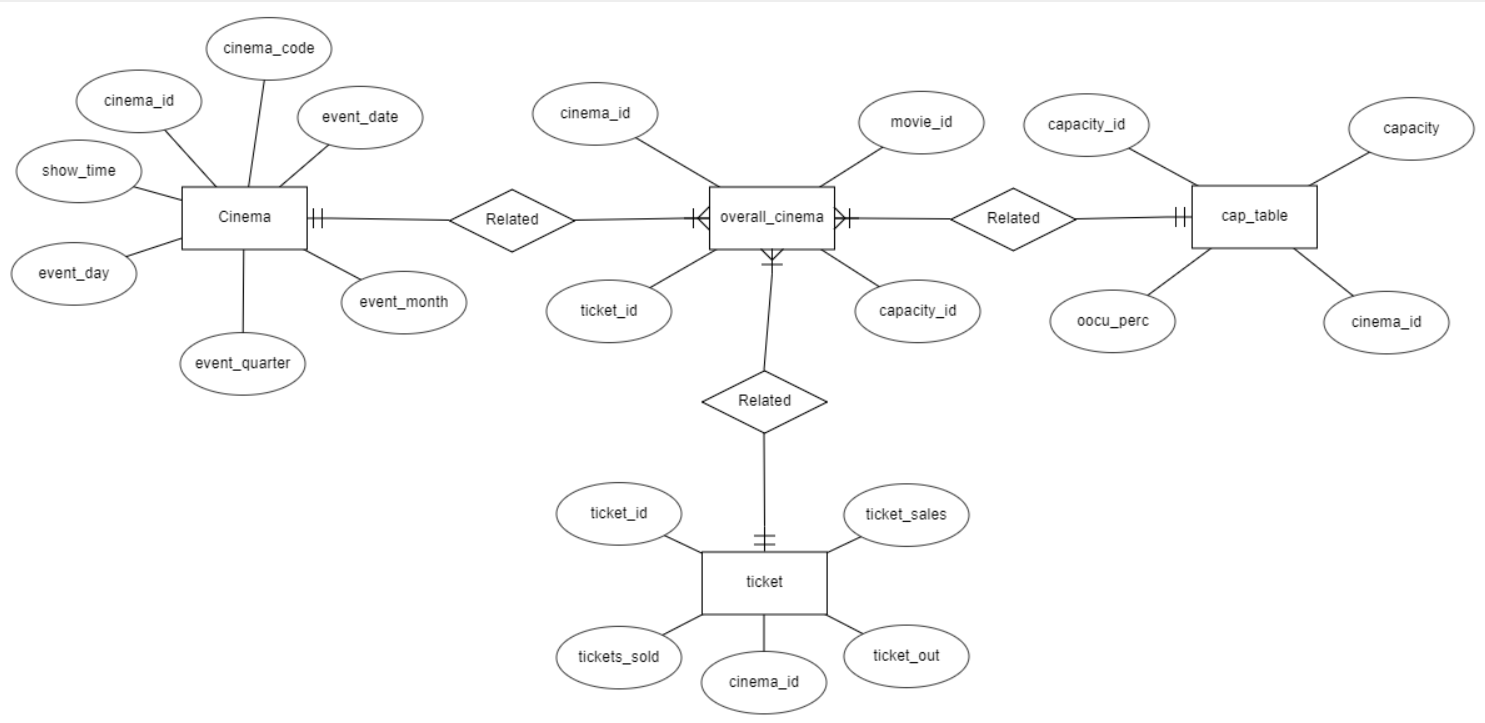
The ER model of the data is depicted in Figure 1 below, where the overall\_cinema table has linked foreign keys that provide a one-to-many relationship with the other tables. The usage of foreign keys to link the tables to one another is further explained in Fig. 2.****

Fig 1. Relational Schema

According to Figure 2, foreign keys are used to link the tables. The database has four tables, so there are four primary keys. There is at least one primary key and one foreign key in each table.

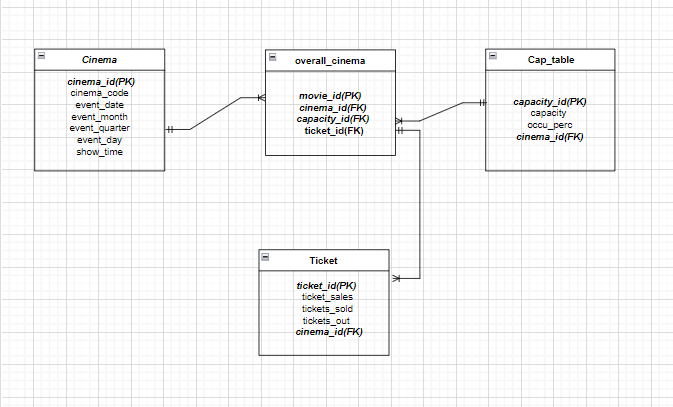


Fig 2. Entity/Relationship Diagram

All the attributes in the original dataset are included in table 1 below, however I'm just utilizing the information on the necessary attributes. I made sure there were no properties with missing data, removed them, and then built the table.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| Cinema\_code | Each cinema theatre has a unique code this describes that code. |
| Tickets\_out | This tells how many tickets have been cancelled. |
| Total\_sales | This shows us the total sales for each screening. |
| Tickets\_sold | This describes how many cinema tickets are sold totally. |
| Show\_time | This tells about the daily screening period |
| Occu\_perc | This shows the cinema occupancy rate based on available capacity. |
| Ticket\_price | This tells about the ticket price of the movie. |
| Capacity | This tells about how many people can fit in the cinema. |
| Month | This tells about the month of event. |
| Date | This tells about the date of event. |
| Quarter | This tells about the quarter of the event. |
| Day | This tells about the day of the event. |

Table 1 Original Dataset

**Normalisation:**

A fact table is created by standardizing the tables in fourth normalized form (4NF). No matter what, in order to create a useful table for doing predictive analytics, the tables would need to be connected together (denormalized).

**Stage 3: Creating the Database**

I produced a new CSV file named tickets.csv after first converting the old CSV file in Jupyter Notebook to 4NF. Later, I added two more folders called data and scripts beneath the midterm/tickets folder that I had previously made. I maintained the new csv file in the data folder and produced 4 files called loading-data.sql, incorporate-data.sql, configure-database.sql, and ingest-data.sql in the script folder. The database is created using these 4 files.

*For the loading-data.sql:*

In the tickets\_data database, a new table called "raw\_data" is being created. This table builds a table with all the properties from the newly created CSV file "tickets.csv." The information from the csv file is used to populate this table.

*Raw\_data:*

CREATE TABLE raw\_data(

   cinema\_code int,

   total\_sales int,

   tickets\_sold int,

   tickets\_out int,

   show\_time int,

   occu\_perc float,

   ticket\_price float,

   ticket\_use int,

   capacity float,

   event\_date timestamp,

   event\_month int,

   event\_quarter int,

   event\_day int,

   row\_no int

);

source /home/coder/project/midterm/tickets/scripts/loading-data.sql; should be run to load the data.

*For the incorporate-data.sql:*

In this file I have created 4 tables in the database namely “cinema”, “cap\_table”, “ticket”, and “cinema\_ticket”. I have filled them with the necessary attributes but at this moment when these are created the tables will be empty. I will later query the data from the raw\_data table. The commands to create the above names tables are below respectively.

*Cinema:*

CREATE TABLE cinema(

    cinema\_id int PRIMARY KEY AUTO\_INCREMENT,

    cinema\_code int,

    event\_date timestamp,

    event\_month int,

    event\_quarter int,

    event\_day int,

    show\_time int,

    row\_no int

);

*Cap\_table:*

CREATE TABLE cap\_table(

    capacity\_id int PRIMARY KEY AUTO\_INCREMENT,

    capacity float,

    occu\_perc float,

    cinema\_id int,

    row\_no int,

    FOREIGN KEY(cinema\_id) REFERENCES cinema(cinema\_id) ON DELETE CASCADE

*Tickets:*

CREATE TABLE ticket(

    ticket\_id int PRIMARY KEY AUTO\_INCREMENT,

    ticket\_sales int,

    tickets\_sold int,

    tickets\_out int,

    cinema\_id int,

    row\_no int,

    FOREIGN KEY(cinema\_id) REFERENCES cinema(cinema\_id) ON DELETE CASCADE

*Cinema\_tickets:*

CREATE TABLE cinema\_tickets(

    movie\_id int PRIMARY KEY AUTO\_INCREMENT,

    cinema\_id int,

    capacity\_id int,

    ticket\_id int,

    row\_no int,

    FOREIGN KEY(cinema\_id) REFERENCES cinema(cinema\_id) ON DELETE CASCADE,

    FOREIGN KEY(capacity\_id) REFERENCES cap\_table(capacity\_id) ON DELETE CASCADE,

    FOREIGN KEY(ticket\_id) REFERENCES ticket(ticket\_id) ON DELETE CASCADE

);

*For the configure-database.sql:*

Here, a database with the name “tickets\_data” is created.

CREATE DATABASE tickets\_data;

To provide access to the database "tickets\_data" we then create a user called "brio".

We should additionally specify "spaniel" as the password for this.

CREATE USER 'brio'@'%' IDENTIFIED WITH mysql\_native\_password BY 'spaniel';

GRANT ALL ON tickets\_data.\* TO 'brio'@'%'

*For the ingest-data.sql:*

At last, I filled up the properties in the tables I made in "incorporate-data.sql" to create the "raw\_data" table in this file. To do this, a new table called "dnorm\_data" is created and connected to the "raw\_data" table. After that, we will empty each table of its contents and then re-fill it with information from the "dnorm table."

*dnorm\_data:*

CREATE TABLE dnorm\_data AS(

    SELECT  cinema\_code,

            total\_sales,

            tickets\_sold,

            tickets\_out,

            show\_time,

            occu\_perc,

            ticket\_price,

            ticket\_use,

            capacity,

            event\_date,

            event\_month,

            event\_quarter,

            event\_day,

            row\_no

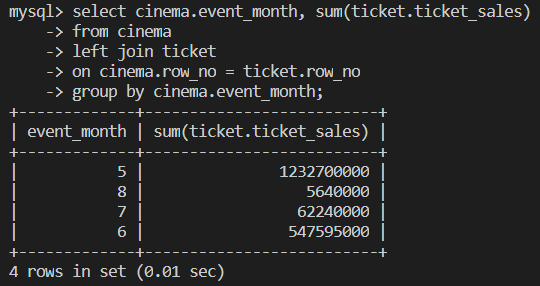
    FROM raw\_data);

*SQL commands that respond to the Stage 1 and inquiries.*

After the database is created, finally I got the answers to my questions asked above by running the SQL steps and generating the data.

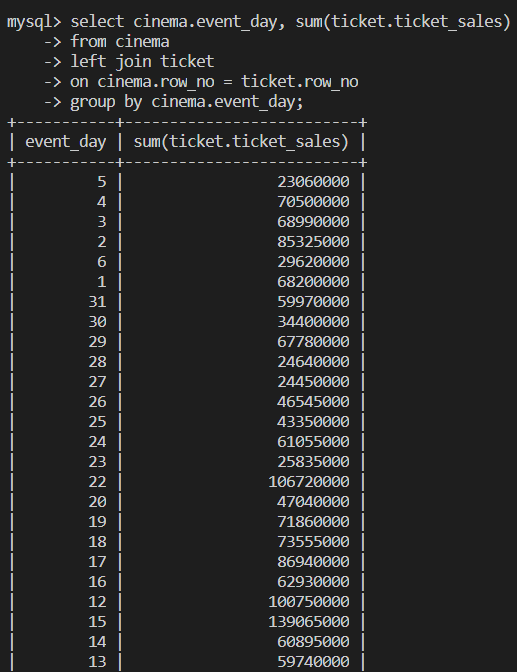
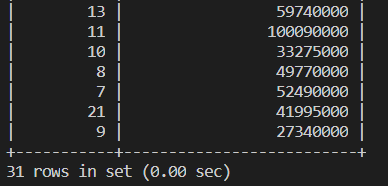
**Question 1:** Which month had the highest ticket sales?

**Answer:** From the below image we can say that the 5th month has the highest ticket sales.



**Question 2:** How are the ticket sales on each day in a month?

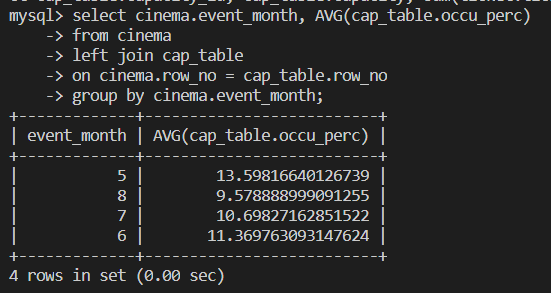
**Answer:** From the below images we can say that the on each day the sales of the cinema tickets are very good because on every day the sales are above 15000000 only.

**Question 3:** What is the average occupancy per each month?

**Answer:** From the below images the avg occupancy per each month are as follows:

* 5th month: 13.59816640126739
* 8th month: 9.578888999091255
* 7th month: 10.69827162851522
* 6th month: 11.369763093147624



**Stage 4: Creating a Simple Web application**

To show the tables we earlier established in the database, we will develop a web application. Each table in turn may be seen by the user via the web application. For instance, using the program, we can view all information pertaining to the cinema's capacity on the cap\_table page. The user may view the movie theatres capacity at this point.

I made the main home page and five distinct web pages, one for each table. I have built a single file named app.js to serve as the JavaScript for linking the sites and querying the data into the pages. The app.js code is seen in the figures below.

*app.js*

const express = require('express');

const bodyParser = require('body-parser');

const mysql = require('mysql');

const mustacheExpress = require('mustache-express');

const app = express();

const port = 3000;

app.engine('html', mustacheExpress());

app.set('view engine', 'html');

app.set('views', './templates');

app.use(bodyParser.urlencoded({ extended: true }));

var dbcon = mysql.createConnection({

    host: 'localhost',

    user: 'brio',

    password: 'spaniel',

    database: 'tickets\_data'

})

function templateRenderer(template, res) {

    return function (error, results, fields) {

        if (error)

            throw error;

        res.render(template, { data: results });

    }

}

app.get('/', function (req, res) {

    res.render('index.html')

})

app.get('/cinema', function (req, res) {

    dbcon.query("SELECT \* FROM cinema LIMIT 25;", templateRenderer('cinema', res));

})

app.get('/cap\_table', function (req, res) {

    dbcon.query("SELECT \* FROM cap\_table LIMIT 25;", templateRenderer('cap\_table', res));

})

app.get('/ticket', function (req, res) {

    dbcon.query("SELECT \* FROM ticket LIMIT 25;", templateRenderer('ticket', res));

})

app.get('/overall\_cinema', function (req, res) {

    dbcon.query("SELECT \* FROM overall\_cinema LIMIT 25;", templateRenderer('overall\_cinema', res));

})

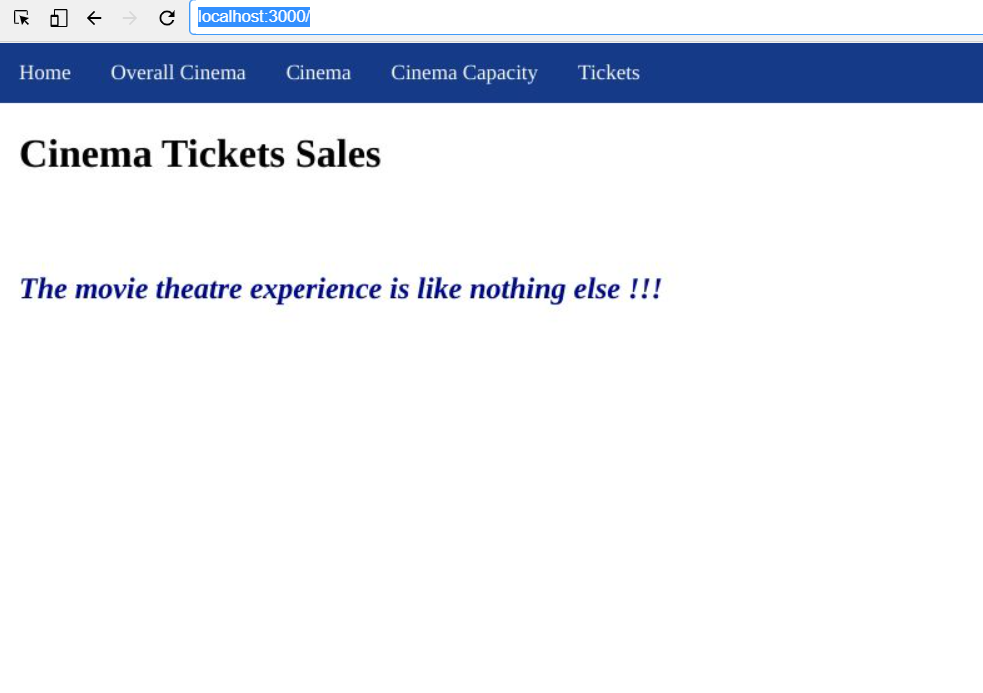
app.listen(port, function () {

    console.log('The app is listening at http://localhost:' + port + '.');

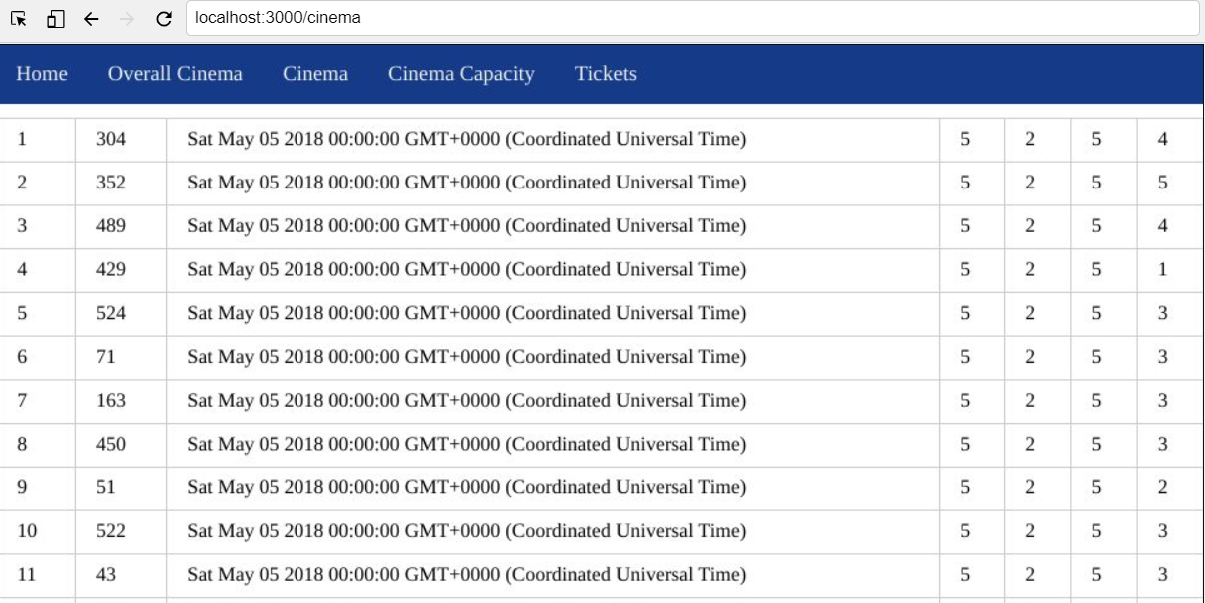
})

Basic HTML is used to generate the cinema, cap\_table(Cinema Cpacity), ticket, and overall\_cinema pages, which are then connected to the database in the app.js file. The pages of the online application resemble the images below.

*Home page:*



*Cinema Page:*

**

*Cinema Capacity Page:*

**

*Overall Cinema Page:*



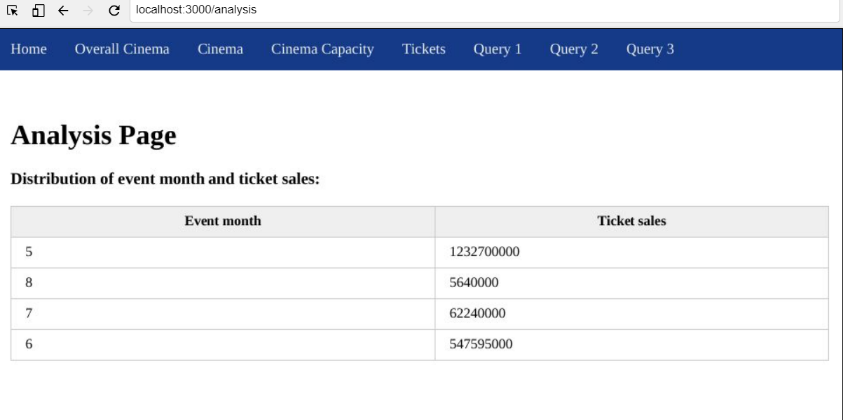
*Tickets Page:*

**

The aforementioned pages make it very obvious that the application's user can find all the data needed to respond to the inquiry from stage 1 and even more.

**You can refer to the images below for the queries SQL.**

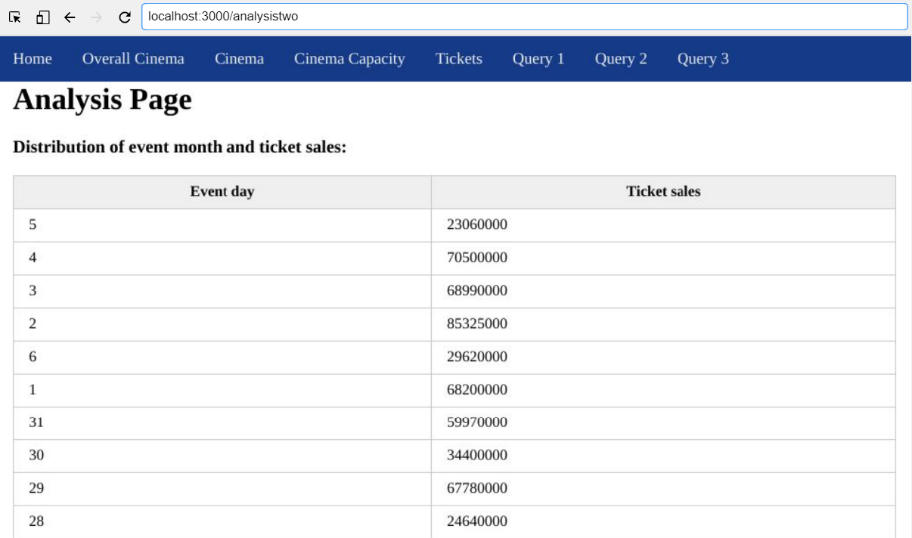
*Query 1:*



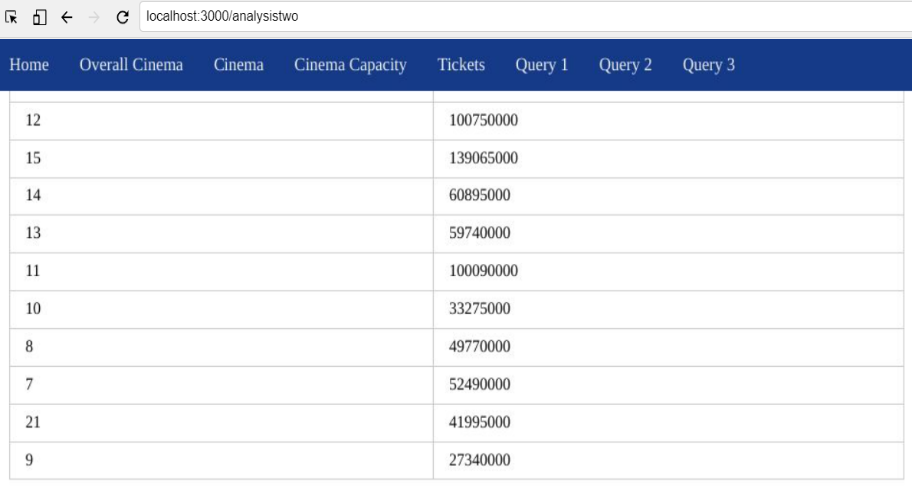
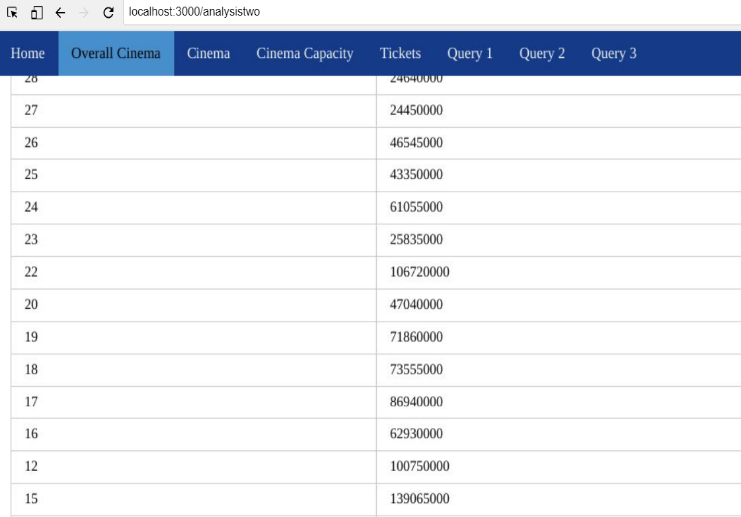
Img.1

*Query 2:*

From Img 2 you can see that the query 2 table is not working due to an error when I run it in the first place, because some inappropriate special characters are added to the http link in the last. If I remove that special character then the query 2 table is working fine as you can see below in the Img. 2.1, 2.2, 2.3

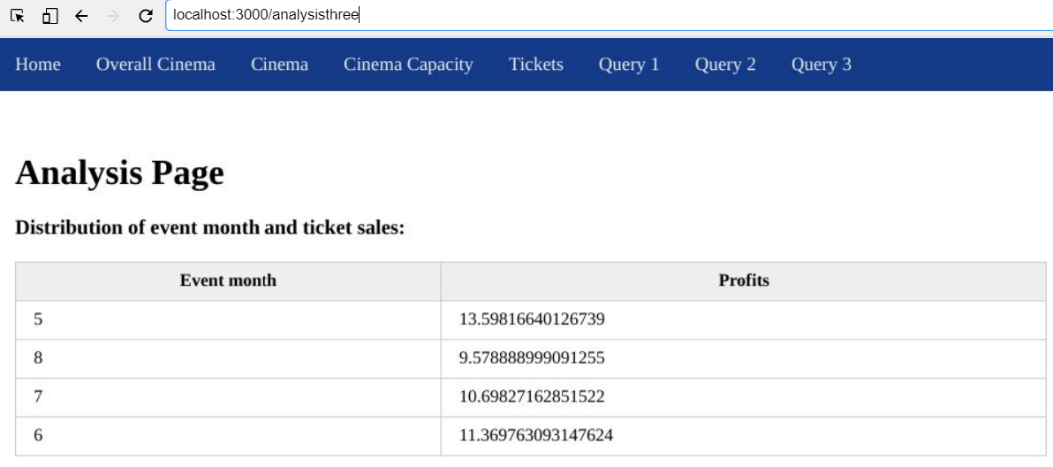


Img.2 Img. 2.1



Img. 2.2 Img. 2.3

*Query 3:*



Img. 3

**References:**

<https://www.kaggle.com/datasets/arashnic/cinema-ticket> - link for the dataset website from which I took the dataset and the csv file for my project.

**Shareable Lab Link**: https://hub.labs.coursera.org:443/connect/sharedgcvfokaz?forceRefresh=false&path=%2F%3Ffolder%3D%2Fhome%2Fcoder%2Fproject