A Report on

THEMEPARK DATABASE

Submitted By

Kruttika Abhyankar J001

Jay Ajmera J004

Under The Guidance Of

**Name of the Faculty Mentor**

**Prof. Rajni Aron**

Logo, company name

Description automatically generated

**Mukesh Patel School of Technology & Management Engineering**

**Department of Data Science**

**Vile Parle (w), Mumbai- 400056**

|  |
| --- |
| **Table of Contents** |
| **Project Description** |
| **ER Diagram** |
| **Relational Schema Diagram** |
| **Implementation** |
| **References and Bibliography** |

**Introduction**

The themepark database is a comprehensive database that provides insights into the various aspects of a theme park. It contains tables for rides, restaurants, shops, employees, customers, tickets, and transactions. By analyzing the data in these tables using SQL queries, we can gain valuable insights into the operations of the theme park, such as which rides are the most popular, how much revenue is generated by each restaurant and shop, and which employees are the most productive. The themepark database is a powerful tool for understanding and improving the performance of a theme park.

**Overview**

Our team has designed and implemented a database to store data related to the rides, restaurants, shops, employees, customers, tickets, and transactions of a themepark.

We have identified the key entities and their relationships and created tables with appropriate attributes and constraints. We have also ensured data integrity and security by implementing primary and foreign keys, and data validation rules.

We have developed a comprehensive set of SQL queries that can be used by the park management to gain insights into various aspects of the park's operations, such as customer behavior, ride popularity, employee performance, and revenue generation.

Our team has worked collaboratively to deliver a well-designed and functioning database that meets the needs of the themepark. We believe that this database will enable the management to make data-driven decisions, improve operational efficiency and enhance the visitor experience.

**Motivation:**

After visiting various theme parks, we decided to choose this database as we felt that by organising the data into various sectors by using SQL, we could make the work of managing the transactions and relationships easier.

We saw that the management tasks at theme parks were poorly implemented as people did not have the resources and technology to do so. Creating a database of all the major divisions in the park will help in making the tasks simpler as well as answer important questions about how to optimize the management process.

Some examples of theme parks are Universal Studios, Disneyland and Imagica.

**Objective:**

This database is designed to store information about a theme park. It contains several tables such as Ride, Restaurant, Shop, Employee, Ticket, Customer, and Transaction database to analyse the relationships between these entities.

The Ride table stores information about the rides in the park, including their unique ID, name, type, duration, and location. The Restaurant table contains information about the different restaurants in the park, including their unique ID, name, type, and location. The Shop table stores information about the different shops in the park, including their unique ID, name, products, and location.

The Employee table stores information about the park employees, including their unique ID, name, salary, and the ID of the ride, restaurant, or shop they are assigned to. The Ticket table stores information about the tickets purchased by customers, including their unique ID, the ID of the customer who purchased it, the ID of the ride associated with the ticket, and the purchase date.

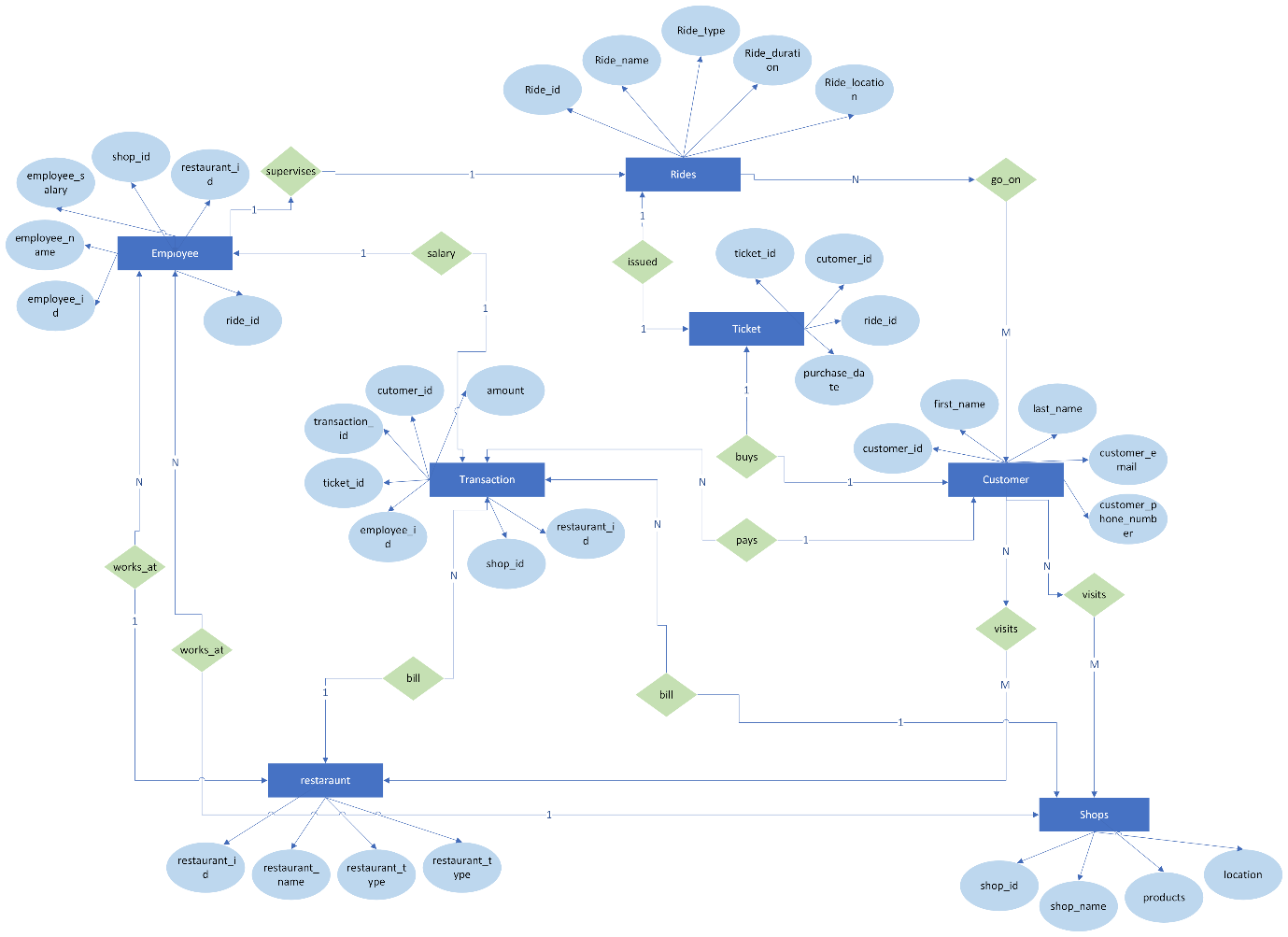
The Customer table stores information about the park customers, including their unique ID, first name, last name, email address, and phone number. The Transaction table stores information about the transactions that occur in the park, including their unique ID, the ID of the customer who made the transaction, the amount of the transaction, the IDs of the restaurant, shop, or employee associated with the transaction, and the ID of the ticket associated with the transaction.

Overall, this database provides a comprehensive system for tracking various aspects of a theme park, including the rides, restaurants, shops, employees, customers, and transactions that occur within it.

The THEMEPARK database consists of six tables: Ride, Restaurant, Shop, Employee, Ticket, and Customer.

**Entity Relationship Diagram**

The THEMEPARK database consists of six entities: Ride, Restaurant, Shop, Employee, Ticket, and Customer.



**1.Ride** table

**Attributes:** ride\_id, ride\_name, ride\_type, ride\_duration, and ride\_location

**Primary key:** ride\_id

**2.Restaurant** table

**Attributes:** restaurant\_id, restaurant\_name, restaurant\_type and location.

**Primary key:** restaurant\_id

**3. Shop** table

**Attributes:** shop\_id, shop\_name, products, and location.

**Primary key:** shop\_id

**4.Employee** table

**Attributes:** employee\_id, employee\_name, employee\_salary, shop\_id, restaurant\_id, and ride\_id.

**Primary key:** employee\_id,

**5. Ticket** table

**Attributes:** attributes ticket\_id, customer\_id, ride\_id, and purchase\_date.

**Primary key:** ticket\_id

**6. Customer** table

**Attributes:** customer\_id, first\_name, last\_name, customer\_email, and customer\_phone\_number.

**Primary key:** customer\_id

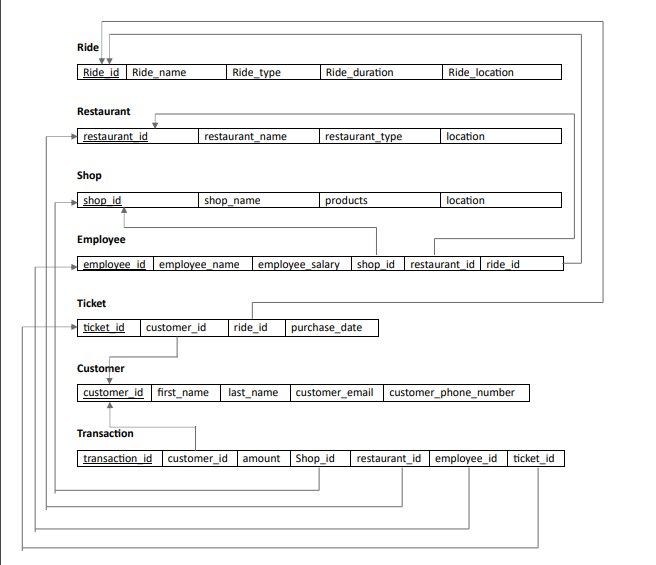
**7. Transaction** table

**Attributes:** transaction\_id, customer\_id, amount, restaurant\_id, shop\_id, employee\_id, and ticket\_id.

**Primary key:** transaction\_id

**Assumptions**

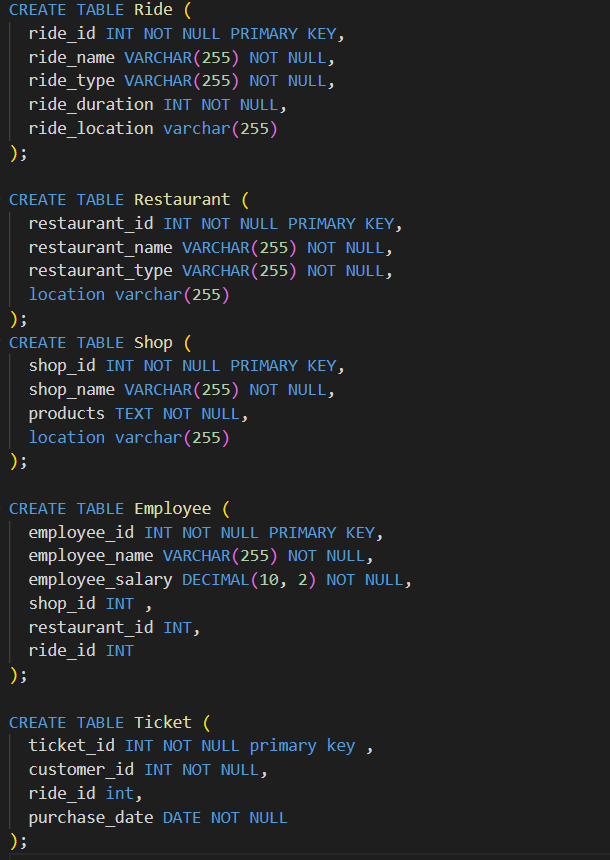
* multiple employees can work in a restaurant.
* multiple employees can work in a shop.
* only one employee can supervise a ride.
* no employee can work in multiple restaurants, shops, rides.
* all transactions are registered in the transactions table.
* one ticket can be used by one customer only.
* only one ride can be accessed with one ticket.

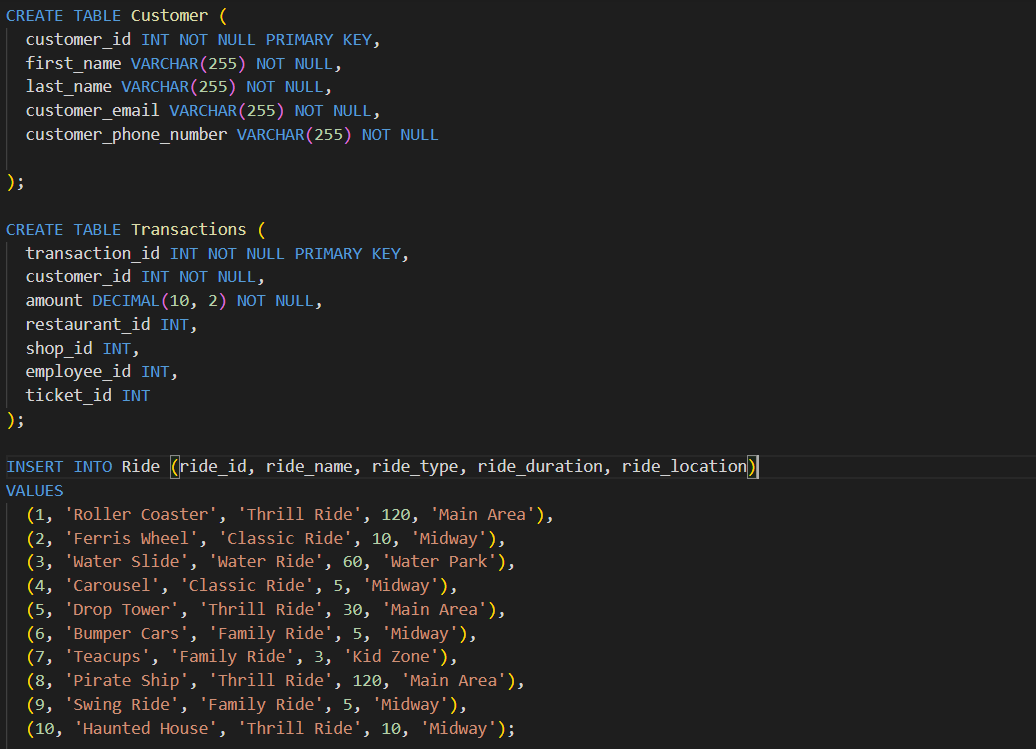
**Schema Diagram**:

**Foreign Keys**

* shop\_emp: A foreign key in the Employee table that references the shop\_id attribute in the Shop table.
* rest\_emp: A foreign key in the Employee table that references the restaurant\_id attribute in the Restaurant table.
* ride\_emp: A foreign key in the Employee table that references the ride\_id attribute in the Ride table.
* cust\_tick: A foreign key in the Ticket table that references the customer\_id attribute in the Customer table.
* ride\_tick: A foreign key in the Ticket table that references the ride\_id attribute in the Ride table.
* cust\_trans: A foreign key in the Transaction table that references the customer\_id attribute in the Customer table.
* fk\_transaction\_ride: A foreign key in the Transaction table that references the ride\_id attribute in the Ride table.
* fk\_transaction\_restaurant: A foreign key in the Transaction table that references the restaurant\_id attribute in the Restaurant table.
* fk\_transaction\_shop: A foreign key in the Transaction table that references the shop\_id attribute in the Shop table.
* fk\_transaction\_employee: A foreign key in the Transaction table that references the employee\_id attribute in the Employee table.
* fk\_transaction\_ticket: A foreign key in the Transaction table that references the ticket\_id attribute in the Ticket table.

**Creating Database**





**Implementation**

Retrieve Contact info of Customers





Graphical user interface, text, application

Description automatically generated

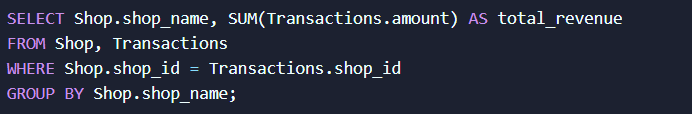
Get the average ride duration



Graphical user interface, text, application

Description automatically generated

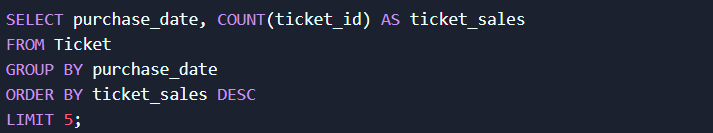
Find the total revenue generated by each shop



Table

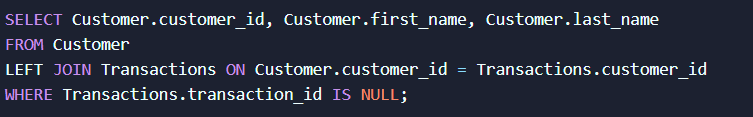
Description automatically generated

Find the top 5 busiest days (i.e., days with the highest number of tickets sold).

Graphical user interface, text, table

Description automatically generated

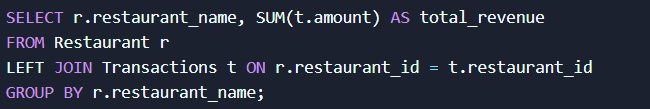
Customers which dont have any transactions



Graphical user interface, table

Description automatically generated

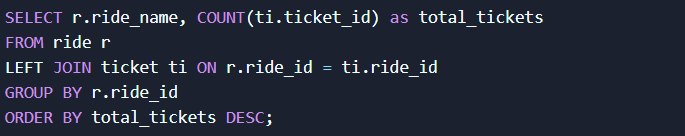
Find the total revenue generated by each restaurant.



Table

Description automatically generated

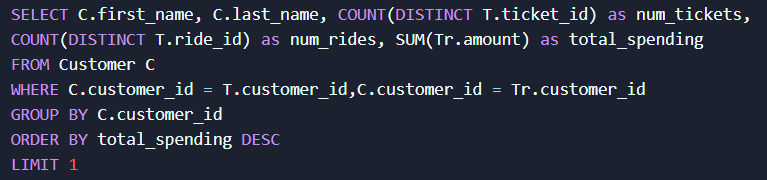
Find the total number of tickets sold for each ride:



Graphical user interface, table

Description automatically generated

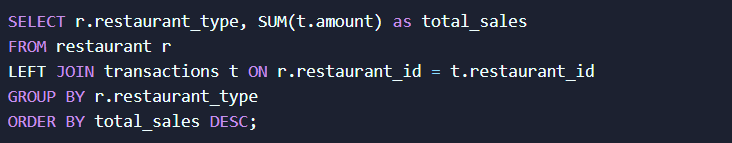
Find the top customers by total spending, along with the total number of tickets and the total number of rides they have been on.



Graphical user interface, text, application

Description automatically generated

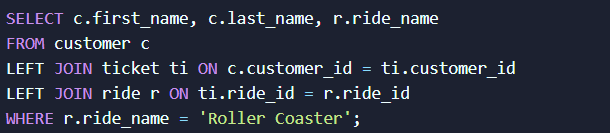
Find the total sales for each restaurant type:



Table

Description automatically generated

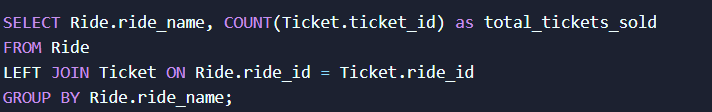
Get the list of customers who have visited a specific ride:



Table

Description automatically generated

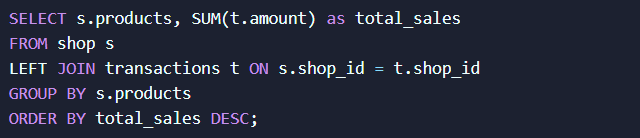
How many tickets sold of each ride



Table

Description automatically generated

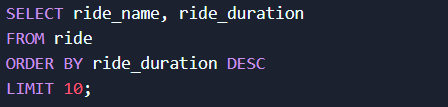
Find the total sales for each product category in the shops:



Text, table

Description automatically generated

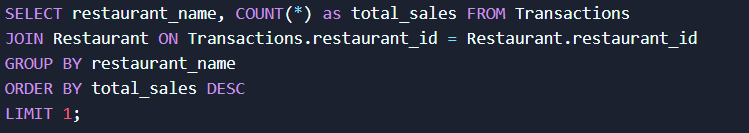
Get the list of rides with the longest duration:



Graphical user interface, table

Description automatically generated

Get the most popular restaurant:



Graphical user interface, text, application

Description automatically generated

**CONCLUSION**

After analyzing the data of the themepark using various queries, several insights can be gained. Some of these include:

1. Understanding the most popular rides, restaurants, and shops, which can help in making decisions about resource allocation and marketing strategies.
2. Identifying the busiest days and months of the themepark, which can help in staffing decisions and operational planning.
3. Analyzing employee performance and identifying areas where training may be necessary to improve efficiency and customer satisfaction.
4. Identifying customer trends and preferences, which can help in tailoring marketing strategies and creating personalized experiences.
5. Monitoring revenue and profitability by ride, restaurant, and shop, which can help in making strategic decisions about pricing and product offerings.

Overall, the insights gained through analyzing the themepark database using SQL queries can help in making informed decisions to improve operations, increase customer satisfaction, and drive revenue growth.