Analyzing Customer Behavior, Expenses, and Supply Management in Amusement Parks

Milestone: Project Report
Group 21

Anagha Veena Sanjeev
Surya Vinay Kumar

857-398-5307
925-404-9121

sanjeev.an@northeastern.edu
vinaykumar.s@northeastern.edu

Percentage of Effort Contributed by Anagha: 50%
Percentage of Effort Contributed by Surya: 50%

Signature of Anagha: 

Signature of Surya: 

Submission Date: 10th December 2023

USE CASE STUDY REPORT

Group No.: 21

Student Names: Anagha Veena Sanjeev & Surya Vinay Kumar

Executive Summary:

The project "Analyzing Customer Behavior, Expenses, and Supply Management in Amusement Parks" aims to understand amusement park dynamics, focusing on customer behavior, expenses, and supply management in a dynamic and competitive environment. Success is linked to meeting visitor expectations, with data analysis providing valuable insights for informed decision-making. The study targets improved customer satisfaction, optimized expenses, and overall operational efficiency.

The relational database comprises ten interconnected tables capturing customer interactions, expenses, and supply chain relationships. EER and UML diagrams guide the modeling process, translating the conceptual model into a relational model with primary and foreign keys. Implementation is done using MySQL, with Studio 3t exploring NoSQL feasibility. The successful database integration with Python enhances analytical capabilities. To further enhance the SQL Database, the next phase involves implementing cost-saving measures and strengthening supplier relationships for optimized inventory management.

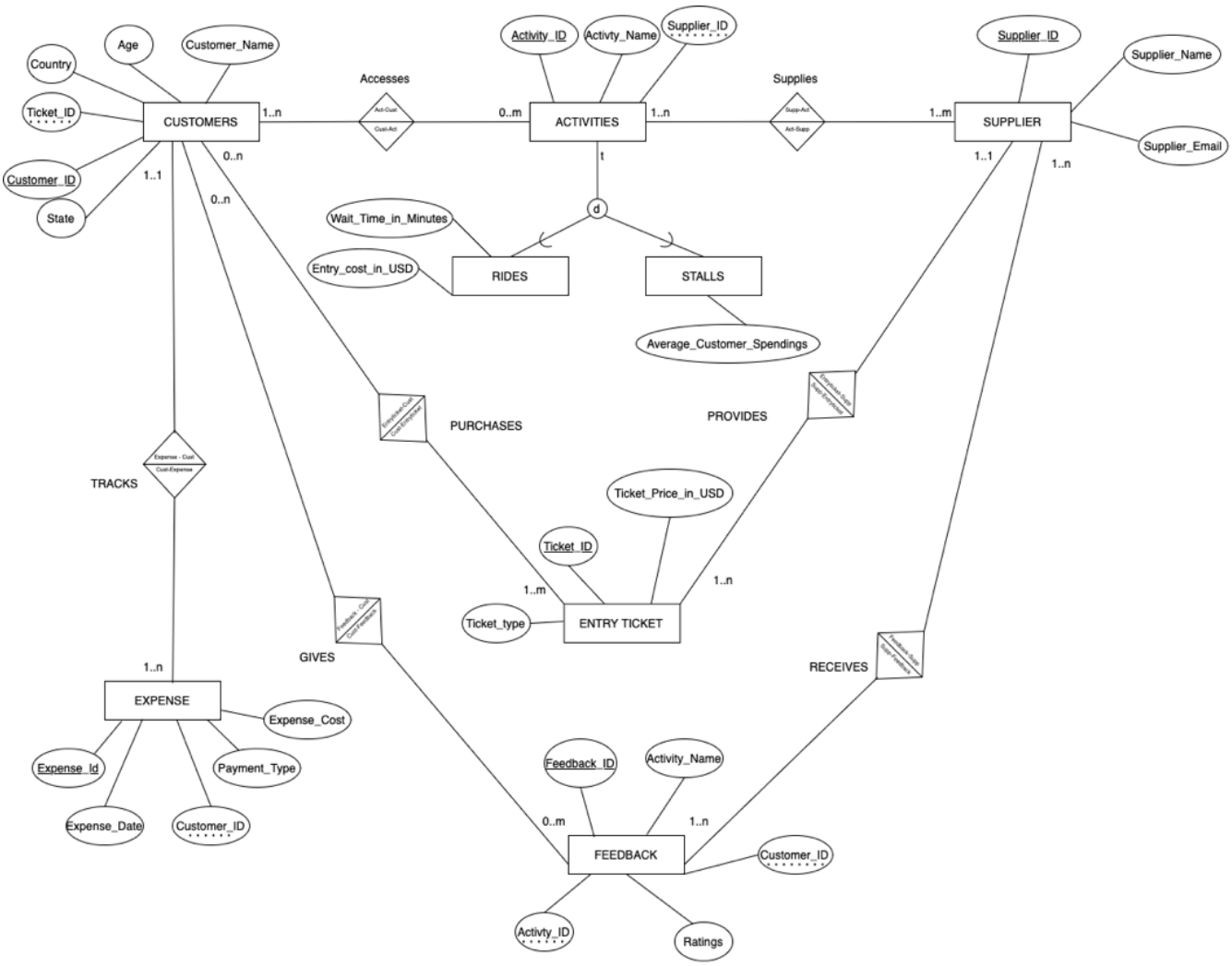
I. Introduction

Analyzing customer behavior, expenses, and supply management in amusement parks reveals insights into the complex dynamics of these entertainment venues. These parks, known for fun and excitement, represent intricate ecosystems where visitor experiences, operational efficiency, and financial sustainability intersect. Delving into customer preferences, expenditures, and supply logistics provides the means to enhance visitor satisfaction, optimize resource allocation, and ensure smooth park functioning. This analysis explores the interplay of visitor behavior, financial management, and supply chain operations, ultimately helping park owners create enjoyable attractions while managing costs effectively.

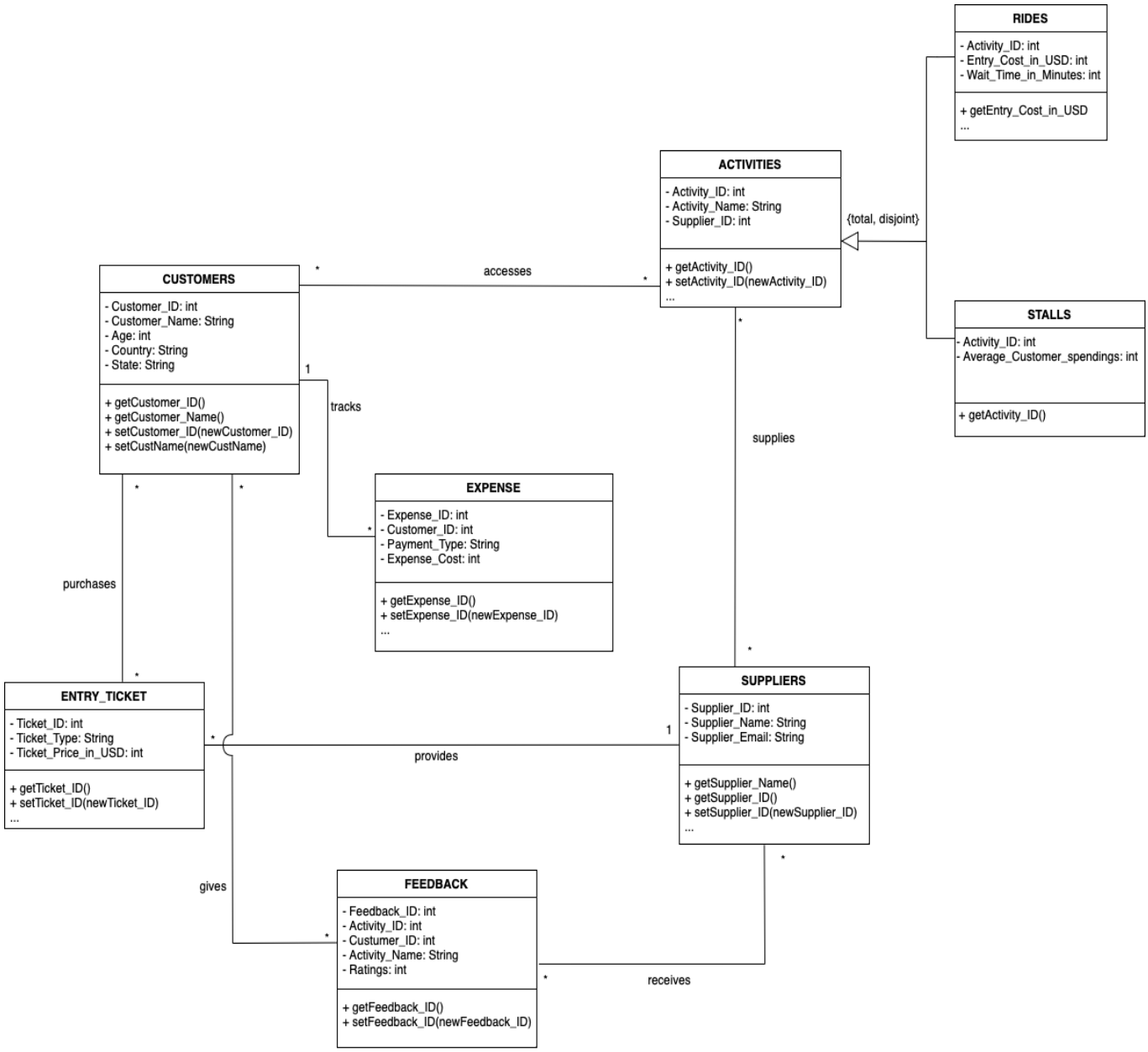
Amusement parks, as intricate ecosystems, rely on the interaction of numerous variables for a unique visitor experience. Despite challenges posed by the diverse data generated, this study aims to conduct a comprehensive analysis. Utilizing a relational database encompassing key entities like customers, activities, expenses, and suppliers, the goal is to uncover patterns that inform strategic decisions, contributing to operational efficiency, customer satisfaction, and the long-term success of amusement park management. The study employs a relational database with interconnected tables, covering diverse aspects of park operations, from customer details to supply chain management. Beyond data analysis, the study emphasizes translating findings into actionable recommendations for amusement park operators. The aim is to improve visitor experiences through data-driven decision-making, shedding light on the intricate dynamics of amusement park management. The study showcases the potential of leveraging data for shaping the future of this exciting industry.

II. Conceptual Data Modeling

1. Enhanced Entity Relationship Diagram (EER)



2. Unified Modeling Language (UML)



III. Mapping Conceptual Model to Relational Model

Primary Key - Underlined

Foreign Key – *Italicized*

CUSTOMERS(Customer_ID, *Ticket_ID*, Age, Customer_Name, State, Country)

FOREIGN KEY Ticket_ID refers to Ticket_ID in ENTRY_TICKET; NULL NOT ALLOWED

ACTIVITIES(Activity_ID, Activity_Name, *Supplier_ID*)

FOREIGN KEY Supplier_ID refers to Supplier_ID in SUPPLIER; NULL NOT ALLOWED

SUPPLIER(Supplier_ID, Supplier_Name, Supplier_Email)

STALLS (Activity_ID,Average_Customer_spendings)

FEEDBACK(Feedback_ID, Activity_Name, Customer_ID, *Activty_ID*,Ratings)

FOREIGN KEY Activity_ID refers to Activity_ID in ACTIVITIES; NULL NOT ALLOWED

ENTRY TICKET(Ticket_ID, Ticket_Type, Ticket_Price_in_USD)

EXPENSE(Expense_ID, Expense_Date, *Customer_ID*, Payment_Type,Expense_cost)

FOREIGN KEY Customer_ID refers to Customer_ID in CUSTOMERS; NULL NOT ALLOWED

RIDES(Entry_Cost_in_USD, Wait_Time_in_Minutes, Activity_ID)

ACCESSES(Customer_ID, Activity_ID)

RECEIVES(Feedback_ID, Supplier_ID)

IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:
The database was created in MySQL and the following queries were performed:

QUERY 1: TOTAL NUMBER OF RIDES IN THE PARK

```
SELECT COUNT(Activity_Name) AS Number_of_Activities
FROM activities;
```

Number_of_Activities
60

QUERY 2: AVERAGE RATINGS FOR ACTIVITIES

```
SELECT Activity_Name, AVG(Ratings) AS Average_Ratings
FROM feedback
GROUP BY 1
ORDER BY 2 DESC;
```

Activity_Name	Average_Ratings
Jack Rabbit (Kennywood)	3.6250
Aero Dips	3.6000
Giant Dipper (Belmont Park)	3.5500
Jack Rabbit (Seabreeze)	3.5333
The Bobs (roller coaster)	3.5294
The Thriller (roller coaster)	3.5294
Lightning (Revere Beach)	3.4706
The Wild One (roller coaster)	3.4500
Restaurant - 4	3.4348
Restaurant - 2	3.3000
Cyclone (Palisades Amusement Park)	3.2500
Switchback Railway (Euclid Beach Park)	3.2500
Crystal Beach Cyclone	3.2500
Green Dragon (Lake Compounce)	3.1818
Coney Island Cyclone	3.1818
The Great Scenic Railway	3.1364

QUERY 3: TOTAL EXPENSE INCURRED BY EACH CUSTOMER

```
SELECT C.Customer_ID, C.Customer_Name,
SUM(E.Expense_Cost) AS Total_Expense
FROM Customers C INNER JOIN Expense E ON
C.Customer_ID = E.Customer_ID
GROUP BY C.Customer_ID, C.Customer_Name
ORDER BY 3 DESC;
```

YEAR	MONTH	TOTAL_EXPENSE_COST
2022	1	4846
2022	2	4547
2022	3	4013
2022	4	4718
2022	5	4979
2022	6	4000
2022	7	4029
2022	8	4056
2022	9	4042
2022	10	5249
2022	11	4520
2022	12	4084

QUERY 4: TOP 10 CUSTOMER WITH HIGHEST EXPENSE COST IN THE PARK

```
SELECT C.Customer_Name, E.total_expense
FROM( SELECT Customer_ID, SUM(Expense_cost) as
total_expense FROM expense GROUP BY 1) E
LEFT OUTER JOIN customers C
ON E.Customer_ID = C.Customer_ID
ORDER BY 2 DESC
LIMIT 10;
```

Customer_Name	total_expense
Rosabel Leads	192
Bill Artis	182
Fairleigh Gilliam	170
Tanhya Boldero	169
Sawyer Verdie	161
Verine Waleworke	160
Nadia Tulloch	157
Grove Nind	156
Lucio Trembley	155
Coretta Grigoire	154

QUERY 5: CUSTOMERS WITH AGE GREATER THAN 50 AND PAYMENT TYPE AS APPLE PAY

```
SELECT Customer_ID,Customer_Name
FROM CUSTOMERS
WHERE Customer_ID IN( SELECT Customer_ID
FROM EXPENSE
WHERE Payment_Type="Apple Pay")
AND Age>50;
```

Customer_ID	Customer_Name
7	Paten Possek
14	Inglebert Folkerd
16	Sissie Mixer
19	Editha Hattiff
20	Anna-diana Braime
28	Julianna Catherine
34	Julina Doorbar
35	Roderick Dennerly
36	Bettine Goodwell
37	Joeann Gianelli
45	Heida Guillem
46	Adriena Van Der…
57	Tandi Batistelli
61	Anthiathia Gaine…
64	Libbey Craghead
66	Dido Lissemore

Query 6: CUSTOMERS WHO ARE BELOW THE AVERAGE SPENDINGS

```
WITH A AS(
SELECT Customer_Name, SUM(E.Expense_cost) as Expense_cost
FROM customers C
LEFT JOIN expense E ON C.Customer_ID = E.Customer_ID
GROUP BY 1)
SELECT Customer_Name FROM A
WHERE Expense_cost < (SELECT AVG(Expense_cost) FROM A);
```

QUERY 6: CUSTOMERS WHO ARE BELOW THE AVERAGE SPENDINGS

```
WITH A AS(
SELECT Customer_Name, SUM(E.Expense_cost) as Expense_cost
FROM customers C
LEFT JOIN expense E
ON C.Customer_ID = E.Customer_ID
GROUP BY 1)
SELECT Customer_Name FROM A
WHERE Expense_cost < (SELECT AVG(Expense_cost)
                      FROM A);
```

Customer_Name
Dorolice Delia
Pip Treweke
Paten Possek
Tibold Currey
Elyn Pumphrey
Lynna Rabbitts
Timmy Pfeiffer
Inglebert Folkerd
Editha Hattiff
Anna-diana Braime
Trenton Eslinger
Aloin Stirtle
Binky Gurner
Flinn Coggell
Shawna Lenard
Joeann Gianelli

QUERY 7: DAY WITH MOST NUMBER OF CUSTOMERS

```
WITH A AS( SELECT Expense_Date, COUNT(Expense_ID) AS COUNT_ID
FROM EXPENSE
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1)
SELECT Expense_Date FROM A
WHERE COUNT_ID >= ALL(
SELECT MAX(COUNT_ID) FROM(
SELECT Count(Customer_ID) AS COUNT_ID
from EXPENSE
group by Expense_Date
order by COUNT_ID desc) AS B
);
```

Expense_Date
2022-10-23

QUERY 8: CUSTOMERS WHO HAVE NOT PROVIDED ANY FEEDBACK

```
SELECT Customer_ID, Customer_Name
FROM Customers C
WHERE NOT EXISTS (
   SELECT 1
   FROM Feedback F
   WHERE C.Customer_ID = F.Customer_ID);
```

Customer...	Customer_Name
279	Martelle Bateson
914	Christie Matzke
581	Lilas Synnot
850	Myles Muttitt
139	Lezlie Kirkbride
890	Eleonore Blainey
178	Julee Parnaby
983	Dorothea Fedynski
618	Martelle Wickling
155	Lethia Kirman
470	Franny Deavall
466	Dominic Arkley
133	Auria Edginton
14	Inglebert Folkerd
211	Carolus Dewerson
143	Emmott McGrory
553	Terese Ballsdon
126	Flss Lisciardelli

QUERY 9: CUSTOMERS WHO HAVE ACCESSED ACTIVITIES BUT HAVE NOT PROVIDED FEEDBACK

```
WITH A AS(
SELECT Customer_ID
FROM ACCESSES
EXCEPT
SELECT Customer_ID
FROM FEEDBACK)
SELECT A.Customer_ID, C.Customer_Name
FROM A
LEFT JOIN CUSTOMERS C
ON A.Customer_ID = C.Customer_ID;
```

Customer...	Customer_Name
279	Martelle Bateson
914	Christie Matzke
581	Lilas Synnot
850	Myles Muttitt
139	Lezlie Kirkbride
890	Eleonore Blainey
178	Julee Parnaby
983	Dorothea Fedynski
618	Martelle Wickling
155	Lethia Kirman
470	Franny Deavall

QUERY 10: TOTAL EXPENSE COST OVER DIFFERENT MONTHS

```
WITH RECURSIVE TOTAL_EXPENSE_COST AS(
SELECT
YEAR(Expense_Date) AS YEAR,
MONTH(Expense_Date) AS MONTH,
SUM(Expense_cost) AS TOTAL_EXPENSE_COST
FROM expense
GROUP BY 1,2
ORDER BY 2)
SELECT * FROM TOTAL_EXPENSE_COST;
```

Customer_ID	Customer_Name
3	Jeni Newnham
6	Nichole Geer
9	Elyn Pumphrey
10	Lynna Rabbitts
11	Leisha Corradino
13	Edan Jacquemot
14	Inglebert Folkerd
17	Peadar Suatt
21	Trenton Eslinger
22	Aloin Stirtle
24	Tara Yegorovnin
27	Binky Gurner
29	Kane Matys
30	Flinn Coggell
35	Roderick Denn…
36	Bettine Goodwell

NoSQL Implementation:

QUERY 1: FIND ALL ACTIVITIES IN THE RIDES COLLECTION

```
db.RIDES.find({})
```

_id	Activity_ID	Entry_Cost_in_USD	Wait_Time_in_Minutes
656bcb58d8ad4d31e632a89b	ACT001	5	13
656bcb58d8ad4d31e632a89c	ACT002	5	20
656bcb58d8ad4d31e632a89d	ACT003	15	5

QUERY 2: FIND CUSTOMERS WHO PROVIDED FEEDBACK WITH RATINGS GREATER THAN 4.

```
db.FEEDBACK.find({
   Ratings: { $gt: 4 }
})
```

_id	Feedback_ID	Activity_ID	Activity_Name	Ratings	Customer_ID
656c0f72d8ad4d31e632a973	2	ACT021	The Wild One (roller coaster)	5	480
656c0f72d8ad4d31e632a975	4	ACT023	Jack Rabbit (Kennywood)	5	200
656c0f72d8ad4d31e632a976	5	ACT024	Jack Rabbit (Seabreeze)	5	520

QUERY 3: CALCULATE AVERAGE RATINGS PER ACTIVITY

```
db.FEEDBACK.mapReduce(
   function () {
      emit(this.Activity_ID, this.Ratings);
   },
   function (key, values) {
      return Array.sum(values) / values.length;
   },
   {
      out: { inline: 1 },
      finalize: function (key, reducedValue) {
         // Rounding the 'Average_Ratings' to 2 decimals
         var roundedValue = Math.round(reducedValue * 100) / 100;
         return { Activity_ID: key, Average_Ratings: roundedValue };
      }
   }
)
```

results	ok
[59 elements]	1.0

Activity_ID	Average_Ratings
ACT047	2.58

V. Database Access via Python

Python accesses the database for analysis, establishing a link to MySQL through mysql.connector, cursor.execute runs queries and fetchall retrieves results. The obtained list is converted to a pandas dataframe, and matplotlib generates graphical plots for analytics.

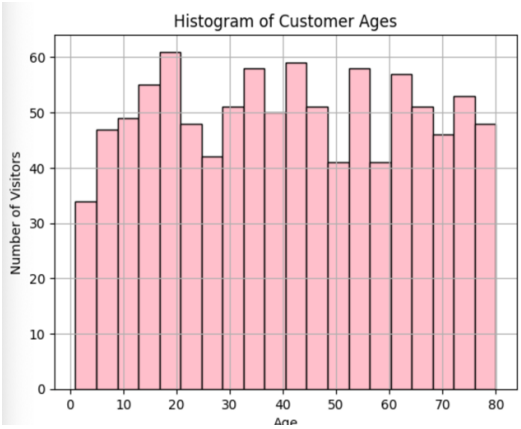


Figure 1: Distribution of customer ages

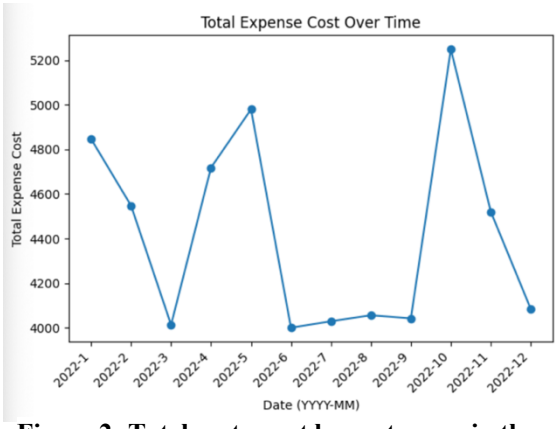


Figure 2: Total cost spent by customers in the year 2022

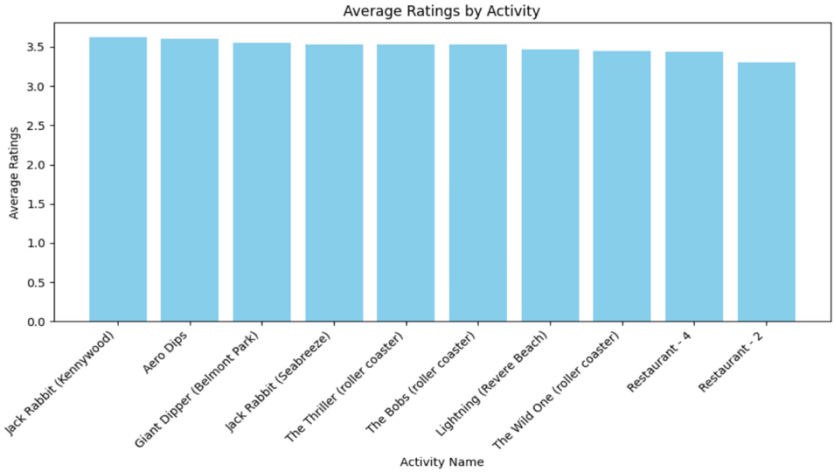


Figure 3: Average rating for top 10 activities

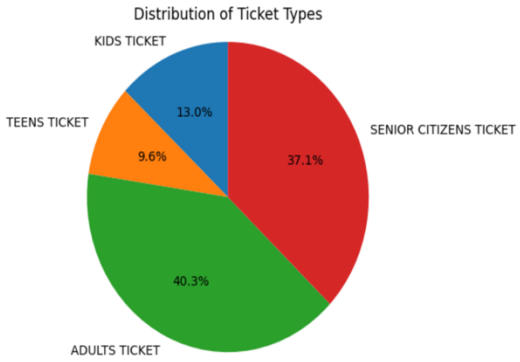


Figure 4: Percentage distribution of types of tickets

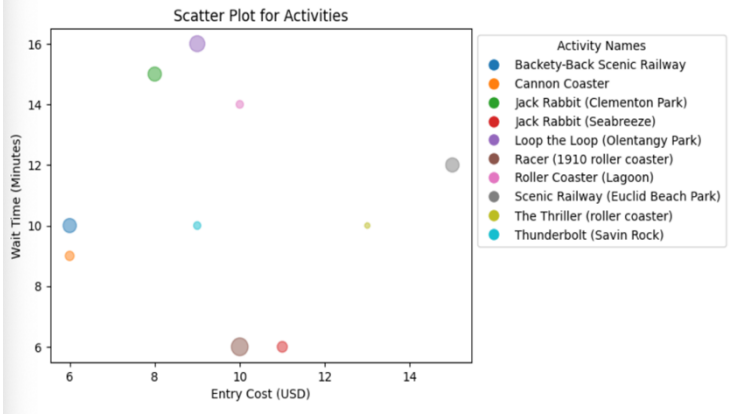


Figure 5: Entry cost vs Wait time for top 10 activities by total customer count

VII. Summary and recommendation

In this project, our focus was on delving into the intricacies of customer behavior, expenses, and supply management within amusement parks through the development and utilization of a relational database. The primary objective was to offer actionable insights that could empower amusement park operators to enhance visitor satisfaction, optimize operational expenses, and refine supply chain processes for improved efficiency.

One of the key advantages of the developed database was its ability to provide a comprehensive understanding of customer behavior. This insight proved invaluable for tailoring services to individual preferences, ultimately contributing to a more satisfying and personalized experience for park visitors. Additionally, the database facilitated the identification of cost-saving opportunities through thorough expense analysis, enabling better financial management. The strengthened link between suppliers and park activities enhanced supply chain efficiency, leading to improved inventory control.

The exploration of NoSQL databases in the project revealed potential challenges, including the complexity of transitioning from a relational model, the need for significant adjustments to accommodate schema-less structures, and limited insights into specific use cases where NoSQL excels. A more thorough examination of these aspects is essential for a comprehensive evaluation of the feasibility and advantages of incorporating NoSQL databases in amusement park management. Careful consideration of trade-offs between consistency and scalability is necessary, given the challenges related to data consistency and transactional integrity in a NoSQL setting. A more thorough examination of the feasibility and advantages of NoSQL databases could provide a more holistic view, potentially uncovering scalability and flexibility benefits.

In conclusion, this project has successfully provided valuable insights into amusement park operations. By implementing the recommended improvements and addressing identified challenges, the database can continue to serve as a powerful tool for amusement park operators, contributing to ongoing operational efficiency and elevated visitor satisfaction in this dynamic and competitive industry.