**Project Report**

**For**

**Amusement Park Database Management System**

**Project Members :-**

* Rathod Anooksha – 17BCP005
* Lalcheta Vatsal – 17BCP031
* Ponkiya Pranay – 17BCP041
* Ravani Bhavya – 17BCP046
* **Contents**
  + 1. [**Experiment 1:**](#Experiment_1)

Difference between standard file management system and DBMS regarding Amusement Park Database Management System.

* + 1. [**Experiment 2:**](#Experiment_2)

Creating Relation Model for the Amusement Park Database Management System

* [Schema](#Experiment_2_Schema)
* [Super Keys, Candidate Keys, Primary Keys and Foreign Keys of Schema](#Experiment_2_Keys)
* [Relational Model](#Experiment_2_Relational_Model)
  + 1. [**Experiment 3:**](#Experiment_3)

Relational Algebra queries for the Amusement Park Database System. Clearly write the definition as well as relational algebra queries for each.

* Develop at least N queries using each of following operators
  1. Selection
  2. Projection
  3. Cartesian product
  4. Union
  5. Set difference
  6. Natural join
  7. Composition of any two from (1-6) operators
  8. Composition of any three of above (1-6) operators
     1. [**Experiment 4:**](#Experiment_4)

Development of E-R model for the Amusement Park Database System.

* + 1. [**Experiment 5:**](#Experiment_5)

Installation of XYZ relational database management system.

* **Experiment – 1**

**FILE SYSTEM vs DBMS**

**In Amusement Park Project**

1. **DATA REDUNDANCY**

Data redundancy means the repetition of data, it means that some data fields are repeated.

**REGARDING AMUSEMENT PARK PROJECT**

There are certain data such as T\_ID (Theme ID), R\_ID (Ride ID) which are used in different data records within the data relations. This will lead to data redundancy. If an incorrect entry is made in any one record then this will in turn lead to data inconsistency.

1. **DATA INCONSISTENCY:**

Data Inconsistency means duplication of data and for those duplicated fields the associated field values are different.

**REGARDING AMUSEMENT PARK PROJECT**

When file system will be used it will require us to take certain data from the visitors again and again at different spots, this will annoy the visitors. Moreover, if any one entry is filled wrong in any one data record that is also present in another data record of the same data relations, then it will lead to data inconsistency.

1. **DATA ISOLATION**

In computer file-based system, data is isolated in separate files. It is difficult to update and to access particular information from data files.

**REGARDING AMUSEMENT PARK PROJECT**

If every entry, that the customer has made in the amusement park are scattered in the different files in file-system, there will be trouble for the customer in checking out and it takes too much time to get the whole data from different file and then process on it.

1. **DIFFICULTY IN ACCESSING DATA**

In classical file organization the data is stored in the files. Whenever data is to be retrieved as per the requirements then a new application program has to be written.

**REGARDING AMUSEMENT PARK PROJECT**

If the data is stored in the classical file-system, we have to make many other programs too in order to access the data.

For instance, when the customer is checking out we have to retrieve the data of the activities of the customer so we have to make the program that can fetch the details of the customer and likewise for the many other things.

1. **INTEGRITY PROBLEMS**

A collection of data is integrated if it meets certain consistency constraints. A programmer always puts these constraints in the programs by adding some codes. In File Processing System, poor data integrity often arises and it becomes very difficult to add new constraints at that time.

**REGARDING AMUSEMENT PARK PROJECT**

If constraints for number of customers per day is not set than a number of problems may arise such as over-crowding, unavailability of ride for long periods, unavailability of efficient services at counters and many more.

This can happen if no integrity constraints are applied in ticket selling systems.

1. **ATOMICITY OF UPDATES**

Atomicity is required to save the data values; it means that information is completely entered or cancelled at all. Any system may fail at any time and at that time it is desired that data should be in a consistent state.

**REGARDING AMUSEMENT PARK PROJECT**

While any online transaction carried out such as buying tickets or making payment at checkout, it may happen that the transaction gets jammed in the midway with money debited but not prompted about the payment been done in the server of the corresponding counter.

1. **SECURITY PROBLEM**

Security problems in the file system can arise in file system when all the users using the data in file system have access to all the data.

**REGARDING AMUSEMENT PARK PROJECT**

In amusement parks there are various of departments having different responsibilities and these departments should be restricted to use only their part of the database and should not be allowed to access other departments data. For example, if the food section has the access to the data of the bookings department or the finance department then there are possibilities that some problems might occur and also the customers should only be allowed to access their own booking details rather than the details of all the visitors of the park. These problems can be easily faced and resolved by the Database Management System.

1. **CONCURRENT ACCESS BY MULTIPLE USERS**

Concurrent access by multiple users means that multiple users can simultaneously access and modify the data.

**REGARDING AMUSEMENT PARK PROJECT**

In amusement park there are many processes which occurs simultaneously in the whole database and even in the same department there are many users who can access or modify the data at the same time. For example, multiple booking at the same time, multiple orders of the food items at food court, multiple online orders of merchandise, multiple check-outs etc. for all these processes controlled concurrent access is needed to be used because uncontrolled concurrent data can lead to data inconsistency. It is difficult in file system data management to handle such kind of database and for sack of better performance and faster response it is better to use the Database Management System.

* **Experiment – 2**

**Relation Model**

**For Amusement Park Project**

* **Schema**

Customers (C\_ID, C\_Name, DOB, Check\_In\_Date, Check\_Out\_Date, Contact\_Number)

Themes (T\_ID, T\_Name, Budget, Party\_Budget)

Rides (R\_ID, R\_Name, Restriction, R\_Price)

Rides\_Ticket (C\_ID, R\_ID, R\_Date, R\_Time)

Includes (T\_ID, R\_ID)

Maintenance (R\_ID, E\_ID, Start\_Date, End\_Date, Status)

Employees (E\_ID, E\_Name, E\_Mail, Contact\_No, DOB, Salary, Designation)

Works (E\_ID, T\_ID)

Shows (S\_ID, S\_Name, S\_Price, Capacity)

Show\_Tickets (S\_ID, C\_ID, Seat\_No, S\_Time, S\_Date)

Incorporates (T\_ID, S\_ID)

Food\_Item (F\_ID, F\_Name, F\_Price)

Food\_Billing (F\_ID, C\_ID, Food\_Quantity, F\_Date, F\_Time)

Shops (Shop\_ID, Shop\_Name)

Shop\_Items (I\_ID, Item\_Name, I\_Price)

Sells (Shop\_ID, I\_ID)

Sold (C\_ID, I\_ID, Quantity, Sold\_Date, Sold\_Time)

Parties (P\_ID, P\_Time, P\_Duration, Party\_Date)

Party\_Billing (P\_ID, T\_ID)

Gives (C\_ID, P\_ID)

Hotel\_Room\_Types (RT\_ID, Type\_Name, Charge)

Stays (C\_ID, RT\_ID, R\_No)

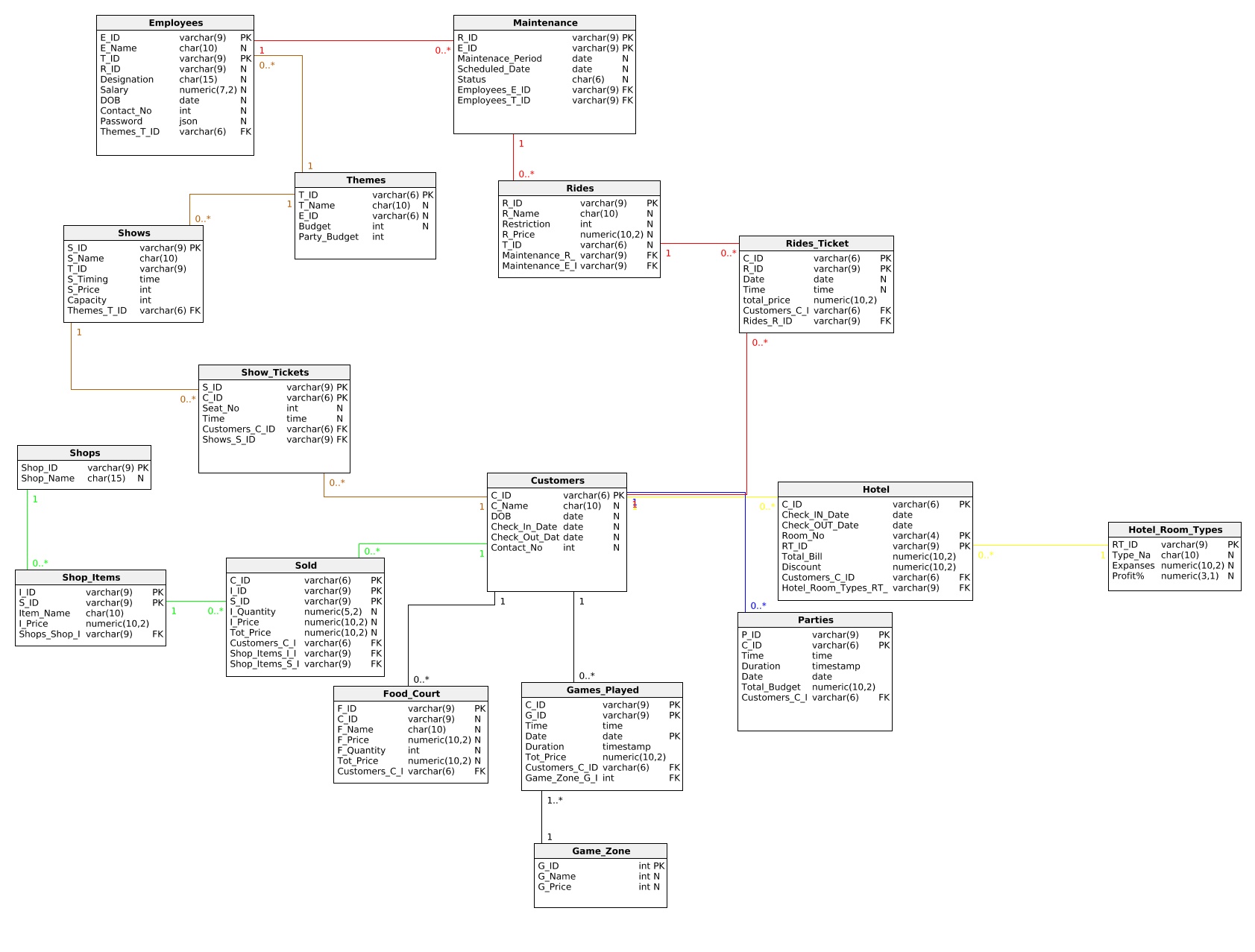
Game\_Zone (G\_ID, G\_Name, G\_Price)

Games\_Played (C\_ID, G\_ID, G\_Time, G\_Date)

**Table of Super Keys, Candidate Keys, Primary Keys and Foreign Keys**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Relation** | **Super key** | **Candidate key** | **Primary key** | **Foreign key** |
| Themes | {T\_ID}  {T\_Name}  {T\_ID,T\_Name}  {T\_ID,Budget}  {T\_ID,Party\_Budget}  {T\_Name,Budget}  {T\_Name,Party\_Budget}  {T\_ID,T\_Name,Budget}  {T\_ID,T\_Name,Party\_Budget}  {T\_Name,Budget,Party\_Budget}  {T\_ID,T\_Name,Budget,Party\_Budget} | {T\_ID}  {T\_Name} | {T\_ID} |  |
| Rides | {R\_ID}  {R\_Name}  {R\_ID, R\_Name}  {R\_ID, Restriction}  {R\_ID, R\_price}  {R\_Name, Restriction}  {R\_Name, R\_price}  {R\_ID,R\_Name,Restriction}  {R\_ID,R\_Name,R\_Price}  {R\_ID,Restriction,R\_Price}  {R\_Name,Restriction,R\_Price}  { R\_ID, R\_Name,Restriction,R\_Price} | {R\_ID}  {R\_Name} | {R\_ID} |  |
| Rides\_Ticket | {C\_ID}  {R\_ID}  {C\_ID,R\_ID}  {C\_ID,R\_Date}  {C\_ID,R\_Time}  {R\_ID,R\_Date}  {R\_ID,R\_Time}  {C\_ID,R\_ID,R\_Date}  { C\_ID,R\_ID,R\_Time}  {R\_ID,R\_Date,R\_Time}  {C\_ID,R\_ID,R\_Date,R\_Time} | {C\_ID}  {R\_ID} | {C\_ID} | {R\_ID}  {C\_ID} |
| Employee | {E\_ID}  {Contact No}  {E\_ID,E\_Name}  {E\_ID, E\_Mail}  {E\_ID,Designation}  {E\_ID,Salary}  {E\_ID, DOB}  {E\_ID,Contact No}  {Contact No,E\_Name}  {Contact No, E\_Mail}  { Contact No,Designtion}  { Contact No,Salary}  { Contact No,DOB}  {E\_ID,E\_Name,E\_Mail}  {E\_ID,E\_Name,Designation}  {E\_ID,E\_Name,Salary}  {E\_ID,E\_Name,DOB}  {E\_ID,E\_Name,Contact No}  {E\_ID,E\_Mail,Designation}  {E\_ID,E\_Mail,Salary}  {E\_ID,E\_Mail,DOB}  {E\_ID,E\_Mail,Contact No.}  {E\_ID,Designation,Salary}  {E\_ID,Designation,DOB}  {E\_ID,Designation,Contact No}  {E\_ID,Salary,DOB}  {E\_ID,Salary,Contact No}  {E\_ID,DOB,Contact No}  {Contact No,E\_Name,E\_Mail}  {Contact No,E\_Name,Designation}  {Contact No,E\_Name,Salary}  {Contact No,E\_Name,DOB}  {Contact No,E\_Mail,Designation}  {Contact No,E\_Mail,Salary}  {Contact No,E\_Mail,DOB}  {Contact No,Designation,Salary}  {Contact No,Designation,DOB}  {Contact No,Salary,DOB}  { E\_ID,E\_Name,E\_Mail,Designation}  { E\_ID,E\_Name,E\_Mail,Salary}  { E\_ID,E\_Name,E\_Mail,DOB}  { E\_ID,E\_Name,E\_Mail,Contact No}  {E\_ID,E\_Name,Designation,Salary}  {E\_ID,E\_Name,Designation,DOB}  {E\_ID,E\_Name,Designation,Contact\_No}  {E\_ID,E\_Name,Salary,DOB}  {E\_ID,E\_Name,Salary,Contact No}  {E\_ID,E\_Name,DOB,Contact\_No}  {E\_ID,E\_Mail,Designation,Salary}  {E\_ID, E\_Mail,Designation,DOB}  {E\_ID, E\_Mail,Designation,Contact No}  {E\_ID, E\_Mail,Salary,DOB}  {E\_ID, E\_Mail,Salary,Contact No}  {E\_ID, E\_Mail,DOB,Contact No}  {E\_ID,Designation,Salary,DOB}  {E\_ID,Designation,Salary,Contact No}  {E\_ID,Designation,DOB,Contact No}  {E\_ID,Salary,DOB,Contact No}  { Contact No.,E\_Name,E\_Mail,Designation}  { Contact No.,E\_Name, E\_Mail,Salary}  { Contact No.,E\_Name, E\_Mail,DOB}  {Contact No.,E\_Name,Designation,Salary}  {Contact No.,E\_Name,Designation,DOB}  { Contact No.,E\_Name,Salary,DOB}  {Contact No, E\_Mail,Designation,Salary}  {Contact No., E\_Mail,Designation,DOB}  { Contact No., E\_Mail,Salary,DOB}  {Contact No.,Designation,Salary,DOB}  {E\_ID,E\_Name, E\_Mail,Designation,Salary}  {E\_ID,E\_Name, E\_Mail,Designation,DOB}  {E\_ID,E\_Name, E\_Mail,Designation,Contact No.}  { E\_ID,E\_Name, E\_Mail,Salary,DOB}  { E\_ID,E\_Name, E\_Mail,Salary,Contact No.}  { E\_ID,E\_Name, E\_Mail,Salary,Password}  { E\_ID,E\_Name, E\_Mail,DOB,Contact No.}  { E\_ID,E\_Name,Designation,Salary,DOB}  { E\_ID,E\_Name,Designation,Salary,Contact No.}  { E\_ID,E\_Name,Designation,DOB,Contact No.}  { E\_ID,E\_Name,Salary,DOB,Contact No.}  {E\_ID, E\_Mail,Designation,Salary,DOB}  {E\_ID, E\_Mail,Designation,Salary,Contact No.}  {E\_ID,R\_ID,Salary,DOB,Contact No.}  {Contact No.,E\_Name, E\_Mail,Designation,Salary}  {Contact No.,E\_Name, E\_Mail,Designation,DOB}  { Contact No.,E\_Name, E\_Mail,Salary,DOB}  { Contact No.,E\_Name,Designation,Salary,DOB}  {Contact No., E\_Mail,Designation,Salary,DOB}  {E\_ID,E\_Name, E\_Mail,Designation,Salary,DOB}  {E\_ID,E\_Name, E\_Mail,Designation,Salary,Contact No.}  { E\_ID,E\_Name, E\_Mail,Salary,DOB,Contact No.}  {E\_ID,E\_Name,Designation,Salary,DOB,Contact No.}  {Contact No.,E\_Name, E\_Mail,Designation,Salary,DOB}  {E\_ID,E\_Name,E\_Mail,Designation,Salary,Contact\_No,DOB} | {E\_ID}  {Contact No.} | {E\_ID} |  |
| Customers | {C\_ID}  {Contact\_Number}  {C\_ID, C\_Name}  {C\_ID, DOB}  {C\_ID, Check\_In\_Date}  {C\_ID, Check\_Out\_Date}  {C\_ID, Contact\_Number}  {Contact\_Number, C\_Name}  {Contact\_Number, DOB}  {Contact\_Number, Check\_In\_Date}  {Contact\_Number, Check\_Out\_Date}  {C\_ID, C\_Name, DOB}  {C\_ID, C\_Name, Check\_In\_Date}  {C\_ID, C\_Name, Check\_Out\_Date}  {C\_ID, C\_Name, Contact\_Number}  {C\_ID, DOB, Check\_In\_Date}  {C\_ID, DOB, Check\_ Out\_Date}  {C\_ID, DOB, Contact\_Number}  {C\_ID, Check\_In\_Date, Check\_Out\_Date}  {C\_ID, Check\_In\_Date, Contact\_Number}  {C\_ID, Check\_Out\_Date, Contact\_Number}  {Contact\_Number, C\_Name, DOB}  {Contact\_Number, C\_Name, Check\_In\_Date}  {Contact\_Number, C\_Name, Check\_Out\_Date}  {Contact\_Number, DOB, Check\_In\_Date}  {Contact\_Number, DOB, Check\_Out\_Date}  {Contact\_Number, Check\_In\_Date, Check\_Out\_Date}  {C\_ID, C\_Name, DOB, Check\_In\_Date}  {C\_ID, C\_Name, DOB, Check\_Out\_Date}  {C\_ID, C\_Name, DOB, Contact\_Number}  {C\_ID, C\_Name, Check\_In\_Date, Check\_Out\_date}  {C\_ID, C\_Name, Check\_In\_Date, Contact\_Number}  {C\_ID, C\_Name, Check\_Out\_Date, Contact\_Number}  {C\_ID, DOB, Check\_In\_Date, Check\_Out\_Date}  {C\_ID, DOB, Check\_In\_Date, Contact\_Number}  {C\_ID, DOB, Check\_Out\_Date, Contact\_Number}  {C\_ID, Check\_In\_Date, Check\_Out\_date, Contact\_Number}  {Contact\_Number, C\_Name, DOB, Check\_In\_Date}  {Contact\_Number, C\_Name, DOB, Check\_Out\_Date}  {Contact\_Number, DOB, Check\_In\_Date, Check\_Out\_date}  {C\_ID, C\_Name, DOB, Check\_In\_Date, Check\_Out\_Date}  {C\_ID, C\_Name, DOB, Check\_In\_Date, Contact\_Number}  {C\_ID, C\_Name, DOB, Check\_Out\_Date, Contact\_Number}  {C\_ID, C\_Name, Check\_In\_Date, Check\_Out\_Date, Contact\_Number}  {C\_ID, C\_Name, Check\_In\_Date, Check\_Out\_Date, Contact\_Number}  {C\_ID, DOB, Check\_In\_Date, Check\_Out\_Date, Contact\_Number}  {C\_ID, C\_Name, DOB, Check\_In\_Date, Check\_Out\_Date,Contact\_Number} | {C\_ID}  {Contact\_No} | {C\_ID} |  |
| Shows | {S\_ID}  {S\_ID, S\_Name}  {S\_ID, T\_ID}  {S\_ID, S\_Timing}  {S\_ID, S\_Price}  {S\_ID, Capacity}  {S\_ID, S\_Name, T\_ID}  {S\_ID, S\_Name, S\_Timing}  {S\_ID, S\_Name, S\_Price}  {S\_ID, S\_Name, Capacity}  {S\_ID, T\_ID, S\_Timing}  {S\_ID, T\_ID, S\_Price}  {S\_ID, T\_ID, Capacity}  {S\_ID, S\_Timing, S\_Price}  {S\_ID, S\_Timing, Capacity}  {S\_ID, S\_Price, Capacity}  {S\_Name, S\_Timing, T\_ID}  {S\_Name, S\_Timing, S\_Price}  {S\_Name, S\_Timing, Capacity}  {S\_ID, S\_Name, T\_ID, S\_Timing}  {S\_ID, S\_Name, T\_ID, S\_Price}  {S\_ID, S\_Name, T\_ID, Capacity}  {S\_ID, S\_Name, S\_Timing, S\_Price}  {S\_ID, S\_Name, S\_Timing, Capacity}  {S\_ID, S\_Name, S\_Price, Capacity}  {S\_ID, T\_ID, S\_Timing, S\_Price}  {S\_ID, T\_ID, S\_Timing, Capacity}  {S\_ID, S\_Timing, S\_Price, Capacity}  {S\_Name, S\_Timing, T\_ID, S\_Price}  {S\_Name, S\_Timing, T\_ID, Capacity}  {S\_Name, S\_Timing, S\_Price, Capacity}  {S\_ID, S\_Name, T\_ID, S\_Timing, S\_Price}  {S\_ID, S\_Name, T\_ID, S\_Timing, Capacity}  {S\_ID, S\_Name, T\_ID, S\_Price, Capacity}  {S\_ID, S\_Name, S\_Timing, S\_Price, Capacity}  {S\_ID, T\_ID, S\_Timing, S\_Price, Capacity}  {S\_Name, T\_ID, S\_Timing, S\_Price, Capacity}  {S\_ID, S\_Name, T\_ID, S\_Timing, S\_Price, Capacity} | {S\_ID}  {S\_Name, S\_Timing} | {S\_ID} | {T\_ID} |
| Show\_Tickets | {S\_ID}  {S\_ID,C\_ID}  {S\_ID,Seat\_No}  {S\_ID,Time}  {S\_ID,C\_ID,Seat\_No}  {S\_ID,C\_ID,Time}  {S\_ID,Seat\_No,Time}  {S\_ID,C\_ID,Seat\_No,Time} | {S\_ID} | {S\_ID} | {C\_ID}  {S\_ID} |
| Food\_Court | {F\_ID}  {F\_ID, C\_ID}  {F\_ID, F\_Name}  {F\_ID,F\_Price}  {F\_ID,F\_Quantity}  {F\_ID,Tot\_Price}  {F\_ID, C\_ID,F\_Name}  {F\_ID, C\_ID,F\_Price}  {F\_ID,C\_ID,F\_Quantity}  {F\_ID, C\_ID,Tot\_Price}  {F\_ID, C\_ID, F\_Name,F\_Price}  {F\_ID, C\_ID, F\_Name,F\_Quantity}  {F\_ID, C\_ID, F\_Name,Tot\_Price}  { F\_ID, C\_ID,F\_Price,F\_Quantity}  { F\_ID, C\_ID,F\_Price,Tot\_Price}  {F\_ID, C\_ID,F\_Quantity,Tot\_Price}  {F\_ID,F\_Name,F\_Price,F\_Quantity}  {F\_ID,F\_Name,F\_Price,Tot\_Price}  { F\_ID,F\_Name,F\_Quantity,Tot\_Price}  {F\_ID,F\_Price,F\_Quantity,Tot\_Price}  {F\_ID,C\_ID,F\_Name,F\_Price,F\_Quantity}  {F\_ID,C\_ID,F\_Name,F\_Price,Tot\_Price}  {F\_ID,C\_ID,F\_Name,F\_Quantity,Tot\_Price}  {F\_ID,C\_ID,F\_Price, F\_Quantity,Tot\_Price}  {F\_ID,F\_Name,F\_Price,F\_Quantity,Tot\_Price}  {F\_ID,C\_ID,F\_Name,F\_Price,F\_Quantity,Tot\_Price} | {F\_ID} | {F\_ID} | {C\_ID} |
| Shops | {Shop\_ID}  {Shop\_ID, Shop\_Name} | {Shop\_ID} | {Shop\_ID} |  |
| Shop\_Items | {I\_ID}  {I\_ID, I\_Name}  {I\_ID, I\_Price}  {I\_ID, I\_Name, I\_Price} | {I\_ID} | {I\_ID} |  |
| Sold | {C\_ID, I\_ID, Shop\_ID}  {C\_ID, I\_ID, Shop\_ID, I\_Quantity}  {C\_ID, I\_ID, Shop\_ID, I\_Price}  {C\_ID, I\_ID, Shop\_ID, Tot\_Price}  {C\_ID, I\_ID, Shop\_ID, I\_Quantity, I\_Price}  {C\_ID, I\_ID, Shop\_ID, I\_Quantity, Tot\_Price}  {C\_ID, I\_ID, Shop\_ID, I\_Price, Tot\_Price}  {C\_ID, I\_ID, Shop\_ID, I\_Quantity, I\_Price, Tot\_Price} | {C\_ID, I\_ID, Shop\_ID} | {C\_ID, I\_ID, Shop\_ID} | {C\_ID}  {I\_ID}  {Shop\_ID} |
| Maintenance | {R\_ID, Scheduled\_Date}  {R\_ID,Status}  {R\_ID, Maintenance\_Period,T\_Manager}  {R\_ID, Maintenance\_Period,Scheduled\_Date}  {R\_ID, Maintenance\_Period,Status}  {R\_ID, T\_Manager, Scheduled\_Date}  {R\_ID, T\_Manager,Status}  {R\_ID,Maintenance\_Period,T\_Manager,Scheduled\_Date}  { R\_ID,Maintenance\_Period,T\_Manager,Status}  {R\_ID,Maintanace\_Period,T\_Manager,Scheduled\_Date,Status} | {R\_ID, Scheduled\_Date} | {R\_ID, Scheduled\_Date} | {T\_MAnager} |
| Parties | {P\_ID}  {P\_ID, C\_ID}  {P\_ID, T\_ID}  {P\_ID, Time}  {P\_ID,Duration}  {P\_ID,Date}  {P\_ID,Total\_Budget}  {P\_ID,C\_ID,Time}  {P\_ID,C\_ID,Duration}  {P\_ID,C\_ID,Date}  {P\_ID,C\_ID,Total\_Budget}  {P\_ID,Time,Duration}  {P\_ID,Time,Date}  {P\_ID,Time,Total\_Budget}  {P\_ID,Duration,Date}  {P\_ID,Duration,Total\_Budget}  {P\_ID, Date, Total\_Budget}  {P\_ID,C\_ID,Time,Duration}  {P\_ID,C\_ID,Time,Date}  {P\_ID,C\_ID,Time,Total\_Budget}  {P\_ID,C\_ID,Duration,Date}  {P\_ID,C\_ID,Duration, Total\_Budget}  {P\_ID,C\_ID,Date,Total\_Budget}  {P\_ID,Time,Duration,Date}  {P\_ID,Time,Duration,Total\_Budget}  {P\_ID,Time,Date,Total\_Budget}  {P\_ID,Duration,Date,Total\_Budget}  {P\_ID,C\_ID,Time,Duration,Date}  {P\_ID,C\_ID,Time,Duration,Total\_Budget}  { P\_ID,C\_ID,Time,Duration,Date,Total\_Budget} | {P\_ID} | {P\_ID} | {T\_ID}  {C\_ID} |
| Hotel | {C\_ID}  {C\_ID, Check\_IN\_Date}  {C\_ID, Check\_OUT\_Date}  {C\_ID, Room\_No}  {C\_ID, RT\_ID}  {C\_ID, Total\_Bill}  {C\_ID, Discount}  {C\_ID, Check\_IN\_Date,Check\_OUT\_Date}  {C\_ID, Check\_IN\_Date,Room\_No}  {C\_ID, Check\_IN\_Date,RT\_ID}  {C\_ID, Check\_IN\_Date,Total\_Bill}  {C\_ID, Check\_IN\_Date,Discount}  {C\_ID,Check\_OUT\_Date,Room\_No}  {C\_ID,Check\_OUT\_Date,RT\_ID}  {C\_ID,Check\_OUT\_Date,Total\_Bill}  {C\_ID,Check\_OUT\_Date,Discount}  {C\_ID,Room\_No,RT\_ID}  {C\_ID,Room\_No,Total\_Bill}  {C\_ID,Room\_No,Discount}  {C\_ID,RT\_ID,Total\_Bill}  {C\_ID,RT\_ID,Discount}  {C\_ID,Total\_Bill,Discount}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Room\_No}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,RT\_ID}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Total\_Bill}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Discount}  {C\_ID,Check\_IN\_Date,Room\_No,RT\_ID}  {C\_ID,Check\_IN\_Date,Room\_No,Total\_Bill}  {C\_ID,Check\_IN\_Date,Room\_No,Discount}  {C\_ID,Check\_IN\_Date,RT\_ID,Total\_Bill}  {C\_ID,Check\_IN\_Date,RT\_ID,Discount}  {C\_ID,Check\_IN\_Date,Total\_Bill,Discount}  {C\_ID,Check\_OUT\_Date,Room\_No,RT\_ID}  {C\_ID,Check\_OUT\_Date,Room\_No,Total\_Bill}  {C\_ID,Check\_OUT\_Date,Room\_No,Discount}  {C\_ID,Check\_OUT\_Date,RT\_ID,Total\_Bill}  {C\_ID,Check\_OUT\_Date,RT\_ID,Discount}  {C\_ID,Check\_OUT\_Date,Total\_Bill,Discount}  {C\_ID,Room\_No,RT\_ID,Total\_Bill}  {C\_ID,Room\_No,RT\_ID,Discount}  {C\_ID,Room\_No,Total\_Bill,Discount}  {C\_ID,RT\_ID,Total\_Bill,Discount}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Room\_No,RT\_ID}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Room\_No,Total\_Bill}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Room\_No,Discount}  { C\_ID,Check\_IN\_Date,Check\_OUT\_Date,RT\_ID,Total\_Bill}  { C\_ID,Check\_IN\_Date,Check\_OUT\_Date,RT\_ID,Discount}  { C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Total\_Bill,Discount}  { C\_ID,Check\_IN\_Date,Room\_No,RT\_ID,Total\_Bill}  { C\_ID,Check\_IN\_Date,Room\_No,RT\_ID,Discount} { C\_ID,Check\_IN\_Date,Room\_No,Total\_Bill,Discount}  { C\_ID,Check\_IN\_Date, RT\_ID, Total\_Bill, Discount}  {C\_ID,Check\_OUT\_Date,Room\_No,RT\_ID,Total\_Bill}  {C\_ID,Check\_OUT\_Date,Room\_No,RT\_ID,Discount}  {C\_ID,Check\_OUT\_Date,Room\_No,Total\_Bill,Discount}  {C\_ID,Check\_OUT\_Date,RT\_ID,Totat\_Bill,Discount}  {C\_ID,Room\_No,RT\_ID,Total\_Bill,Discount}  {C\_ID, Check\_IN\_Date, Check\_OUT\_Date,Room\_No,RT\_ID,Total\_Bill}  {C\_ID, Check\_IN\_Date, Check\_OUT\_Date,Room\_No,RT\_ID,Discount}  {C\_ID, Check\_IN\_Date, Check\_OUT\_Date,Room\_No,Total\_Bill,Disount}  {C\_ID, Check\_IN\_Date, Check\_OUT\_Date,RT\_ID,Total\_Bill,Discount}  {C\_ID, Check\_IN\_Date,Room\_No,RT\_ID,Total\_Bill,Discount}  {C\_ID, Check\_OUT\_Date, Room\_No,RT\_ID,Total\_Bill,Discount}  {C\_ID,Check\_IN\_Date,Check\_OUT\_Date,Room\_No,RT\_ID,Total\_Bill,Discount} | {C\_ID} | {C\_ID} | {C\_ID}  {RT\_ID} |
| Room\_Types | {RT\_ID}  {Type\_Name}  {RT\_ID, Type\_Name}  {RT\_ID, Expanses}  {RT\_ID, Profit%}  {Type\_Name, Expanses}  {Type\_Name, Profit%}  {RT\_ID, Type\_Name, Expanses}  {RT\_ID, Type\_Name, Profit%}  {RT\_ID,Expenses,Profit%}  {Type\_Name, Expanses, Profit%}  {RT\_ID, Type\_Name, Expanses, Profit%} | {RT\_ID}  {Type\_Name} | {RT\_ID} |  |
| Game\_Zone | {G\_ID}  {G\_Name}  {G\_ID,G\_Name}  {G\_ID,G\_Price}  {G\_Name,G\_Price}  {G\_ID,G\_Name,G\_Price} | {G\_ID}  {G\_Name} | {G\_ID} |  |
| Games\_Played | {C\_ID, G\_ID, Time, Date}  {C\_ID, G\_ID, Time, Date, Duration}  {C\_ID, G\_ID, Time, Date, Tot\_Price}  {C\_ID, G\_ID, Time, Date, Duration, Tot\_Price} | {C\_ID, G\_ID, Time, Date} | {C\_ID, G\_ID, Time, Date} | {C\_ID}  {G\_ID} |

* **Relational – Model**

****

* **Experiment – 3**

**ALGEBRAIC QUERIES**

1. Show Customer name and contact No. of customers who bought Item “XYZ”

ΠC\_name, Contact\_No((σItem\_Name=”XYZ”(Shop\_Items ⨝ Sold)) ⨝ Customers)

1. Show Theme name of rides whose maintenance was done on date “XYZ”.

ΠTheme\_Name(σScheduled\_Date=”XYZ”(Themes ⨝ Maintenance))

1. Show Customer name and Contact no of all customers who played the game ”XYZ” on date “AB:CD:EF” in the game zone.

ΠC\_Name, Contact\_No((σG\_Name=”XYZ”,Date=”AD:CD:EF”(Game\_Zone ⨝ Games\_Playes)) ⨝ Customers)

1. Show the Room No, Room Type and total hotel bill of customer having customer ID ”XYZ”.

ΠRoom\_No, Room\_Type, Total\_Bill((σC\_ID=”XYZ”(Hotel ⨝ Customers)) ⨝ Hotel\_Room\_Types)

1. Show name of customers who threw a party having budget grater than “100000$” and show the theme of the party too.

ΠC\_Name, Theme\_Name(σbudget>100000(Parties ⨝ Customer)) ⨝ Themes)

1. Show the list of customers who bought item with ID”ABCDEF” and also display the name of shop which sold it.

ΠC\_Name, Shop\_Name((((σI\_ID=”ABCDEF”(Shop\_Items ⨝ Sold)) ⨝ Shops) ⨝ Customers)

1. Name of rides and their theme name which are closed on date(AB:CD:EF)

ΠR\_Name, T\_Name(((σScheduled\_Date = “AB:CD:EF”(Maintenance ⨝ Rides) ⨝ Themes)

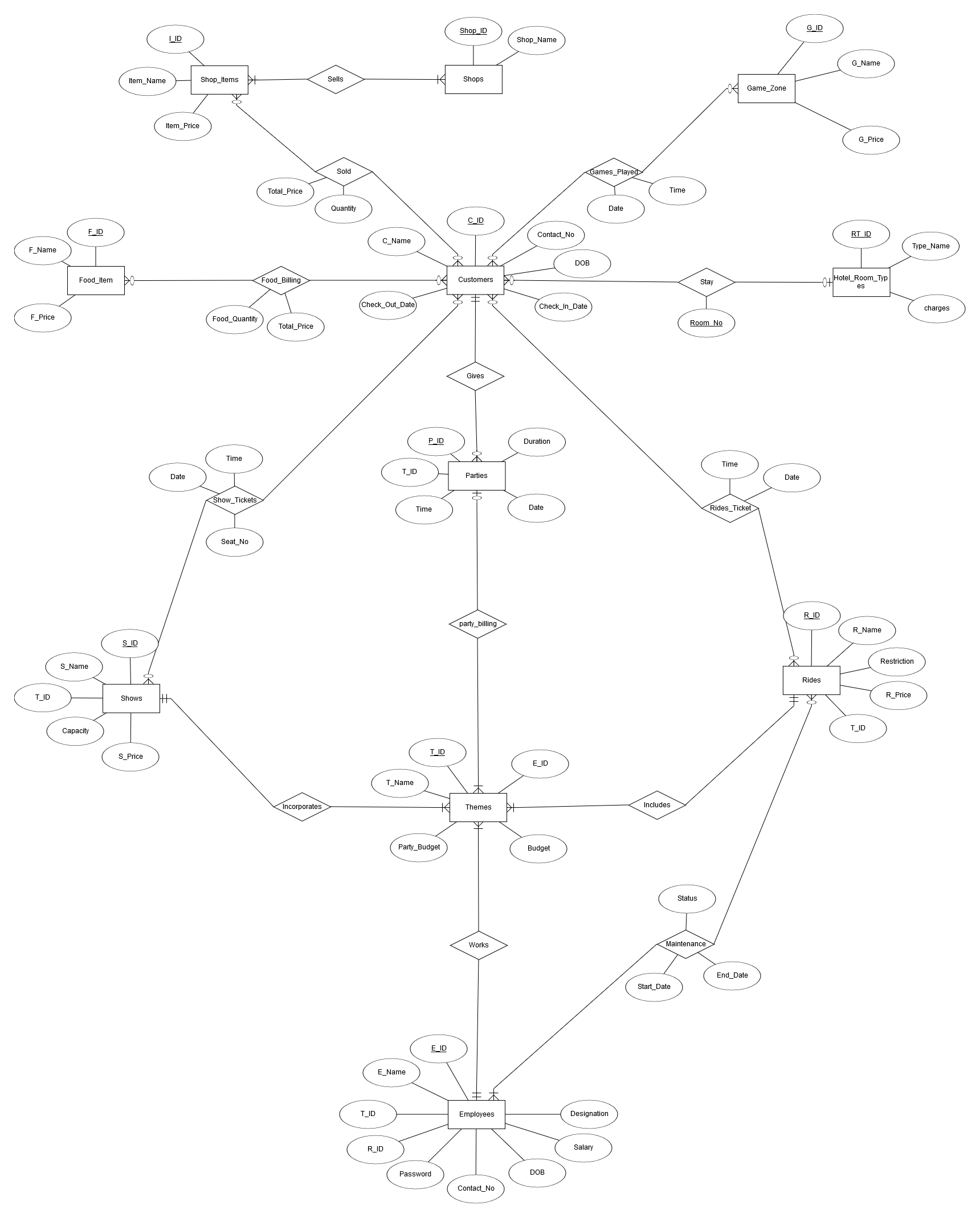
1. Show customer Name and Contact No of customers who bought item “XYZ” and played game “ABC” on date “IJ:KL:MN”

ΠC\_name, Contact\_No((σItem\_Name=”XYZ”(Shop\_Items ⨝ Sold)) ⨝ Customers)

∪

ΠC\_Name, Contact\_No((σG\_Name=”ABC”,Date=”IJ:KL:MN”(Game\_Zone ⨝ Games\_Playes)) ⨝ Customers)

* **Experiment – 4**

**ER-MODEL**

* **Experiment – 5**

**Installation of Amusement Park Database relational database management system**

* **DDL Statements**

create table themes(

T\_ID int primary key,

T\_Name varchar(20),

Budget int,

Party\_Budget int

);

create table customers(

C\_ID int primary key,

C\_Name varchar(20),

DOB date,

Check\_In\_Date date,

Check\_Out\_Date date,

Contact\_No int

);

create table Employees(

E\_ID int primary key,

E\_Name varchar(20),

E\_Mail varchar(30),

Contact\_No int,

DOB date,

Salary int,

Designation varchar(20)

);

create table Rides(

R\_ID int primary key,

R\_Name varchar(20),

Restriction int,

R\_Price int

);

create table includes(

T\_ID int,

R\_ID int primary key,

foreign key (T\_ID) references themes(T\_ID),

foreign key (R\_ID) references rides(R\_ID)

);

create table works(

E\_ID int primary key,

T\_ID int,

foreign key (E\_ID) references employees(E\_ID),

foreign key (T\_ID) references themes(T\_ID)

);

create table Shows(

S\_ID int primary key,

S\_Name varchar(20),

capacity int,

S\_Price int

);

create table maintenance(

R\_ID int,

Start\_Date date,

End\_Date date,

E\_ID int,

Status varchar(10),

foreign key (R\_ID) references rides(R\_ID),

foreign key (E\_ID) references employees(E\_ID),

primary key (R\_ID,Start\_Date)

);

create table parties(

P\_ID int primary key,

P\_Time timestamp,

Party\_Date date,

P\_Duration timestamp

);

create table Food\_Item(

F\_ID int primary key,

F\_Name varchar(20),

F\_Price int

);

create table Shops(

Shop\_ID int primary key,

Shop\_Name varchar(30)

);

create table shop\_items(

I\_ID int primary key,

Item\_Name varchar(20),

Item\_Price int

);

create table Game\_Zone(

G\_ID int primary key,

G\_Name varchar(20),

G\_Price int

);

create table Hotel\_Room\_Type(

RT\_ID int primary key,

Type\_Name varchar(20),

Expanses int ,

Profit\_pcent int

);

create table incorporates(

T\_ID int,

S\_ID int primary key,

foreign key (T\_ID) references themes(T\_ID),

foreign key (S\_ID) references shows(S\_ID)

);

create table Show\_Tickets(

C\_ID int,

S\_ID int,

Seat\_No int,

S\_Time timestamp,

primary key(C\_ID,S\_ID,S\_Time),

foreign key(C\_ID) references customers(C\_ID),

foreign key(S\_ID) references shows(S\_ID)

);

create table rides\_ticket(

R\_ID int,

C\_ID int,

R\_Date date,

R\_Time timestamp,

primary key(R\_ID,C\_ID,R\_Time),

foreign key(R\_ID) references rides(R\_ID),

foreign key(C\_ID) references customers(C\_ID)

);

create table Food\_Billing(

C\_ID int,

F\_ID int,

Food\_Quantity int,

Total\_Price int,

F\_Time TIMESTAMP,

primary key(C\_ID,F\_ID,F\_Time),

foreign key(C\_ID) references customers(C\_ID),

foreign key(F\_ID) references food\_item(F\_ID)

);

create table stay(

C\_ID int,

RT\_ID int,

Room\_No int,

Total\_Bill int,

Discount int,

S\_Date date,

primary key(C\_ID,S\_Date),

foreign key(C\_ID) references customers(C\_ID),

foreign key(RT\_ID) references hotel\_room\_type(RT\_ID)

);

create table sold(

C\_ID int,

I\_ID int,

Total\_Price int,

Quantity int,

S\_Time timestamp,

primary key(C\_ID,I\_ID,S\_Time),

foreign key (C\_ID) references customers(C\_ID),

foreign key (I\_ID) references shop\_items(I\_ID)

);

create table sells(

Shop\_ID int,

I\_ID int,

primary key(Shop\_ID,I\_ID),

foreign key(Shop\_ID) references shops(Shop\_ID),

foreign key(I\_ID) references shop\_items(I\_ID)

);

create table games\_played(

C\_ID int,

G\_ID int,

G\_Date date,

G\_Time timestamp,

Total\_Price int,

primary key(C\_ID,G\_Time),

foreign key (C\_ID) references customers (C\_ID),

foreign key (G\_ID) references game\_zone (G\_ID)

);