

***DBMS PROJECT***

***Amusement Park Management System***

BY :

Sai Surya Pranav Biraka.

ABOUT:

**Amusement Park Management System (APMS)** is a centralized software platform designed to automate and streamline operations in theme parks. It integrates key functions such as ticketing, ride management, customer data tracking, point-of-sale (POS) systems, and workforce coordination into a unified digital ecosystem. By replacing manual processes with real-time data management, the system ensures efficient handling of visitor flow, safety compliance, revenue tracking, and maintenance scheduling. The APMS acts as the backbone of park operations, enabling administrators to monitor and optimize every aspect of the business from a single interface.

Amusement parks require an APMS to enhance efficiency, safety, and profitability. Manual management of tickets, ride queues, food sales, and employee schedules is error-prone and inefficient, especially in large parks with thousands of daily visitors. The system eliminates bottlenecks by automating ticketing, ensuring compliance with safety regulations through maintenance logs. Additionally, it boosts revenue by tracking spending patterns, enabling dynamic pricing, and facilitating targeted promotions. Without an APMS, parks risk operational chaos, lost revenue opportunities, and compromised guest satisfaction.

The APMS operates through interconnected modules that communicate in real time:

1. **Ride Management**: Sensors monitor ride capacity and safety inspections, flagging maintenance needs.
2. **POS & Retail Systems**: Cashless payments track food, merchandise, and photo sales per visitor.
3. **Analytics Dashboard**: Aggregates data on attendance, revenue peaks, and ride popularity for strategic decisions.
4. **Employee Management**: Assigns staff to rides, food outlets, or maintenance tasks based on real-time demand

The APMS enhances parks in **Improved Guest Experience, Operational Efficiency**, **Safety &Compliance**,and**Data-Driven Growth.**

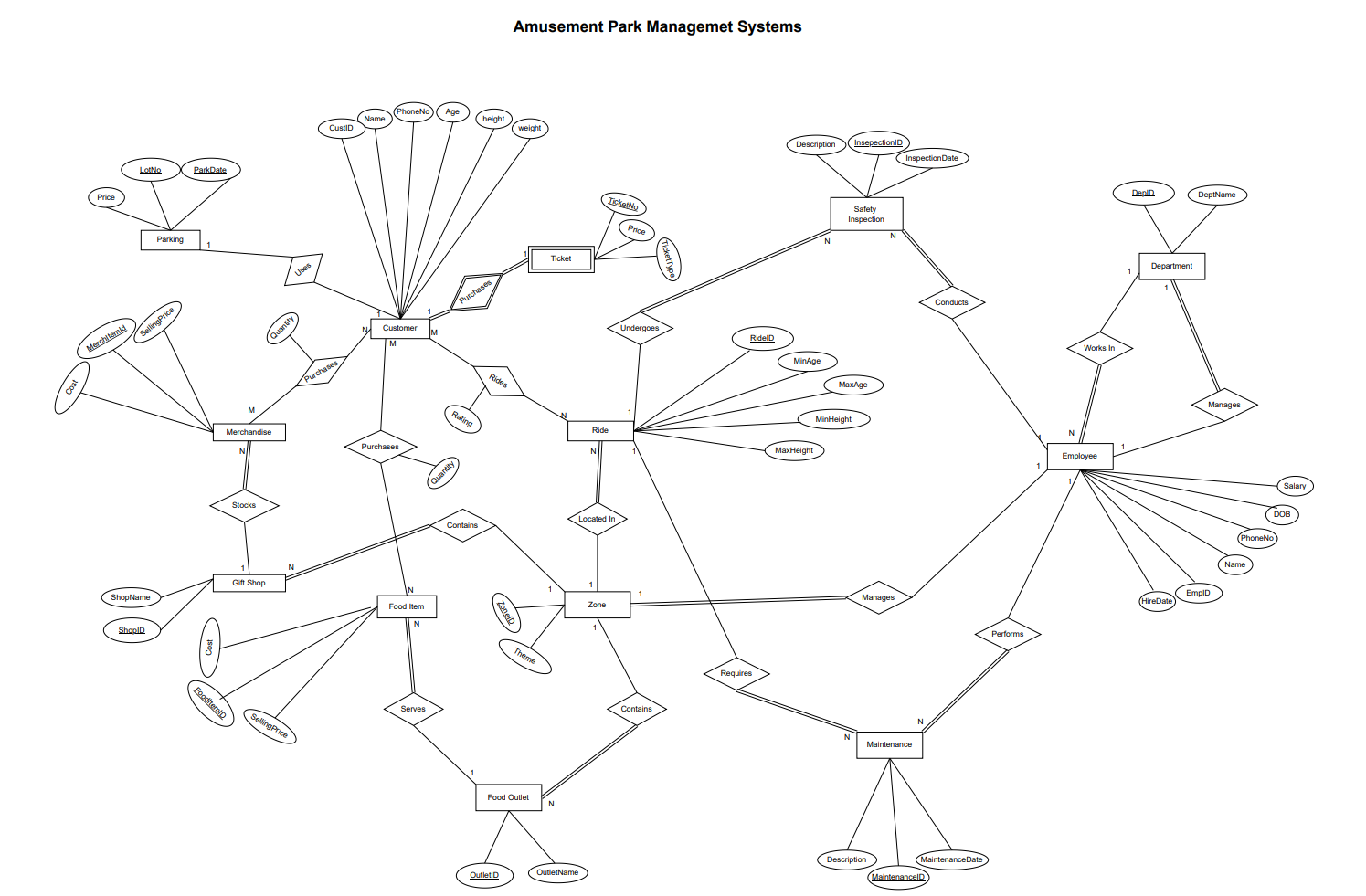
By integrating these functions, the system ensures smoother operations, higher visitor satisfaction, and increased profitability.

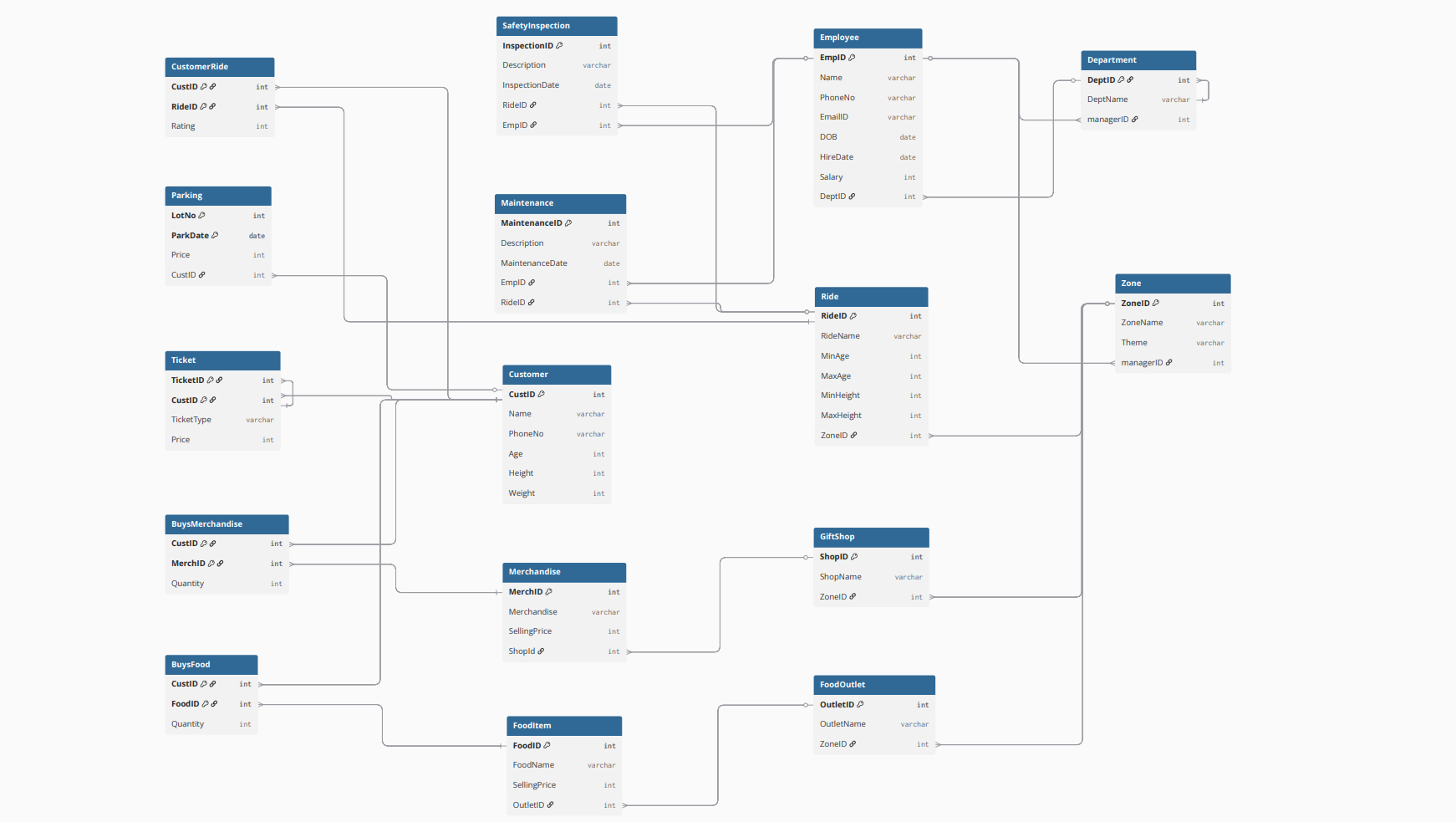
ENTITIES USED:

1. **Customer** - Represents park visitors who use services
2. **Ticket** - Represents admission passes purchased by customers
3. **Parking** - Represents vehicle parking used by customers
4. **Zone** - Represents themed areas containing attractions
5. **Ride** - Represents park attractions available to customers
6. **SafetyInspection** - Represents safety checks performed on rides
7. **Department** - Represents staff organizational units
8. **Employee** - Represents park workers maintaining operations
9. **FoodOutlet** - Represents dining locations in zones
10. **FoodItem** - Represents menu items sold at outlets
11. **GiftShop** - Represents retail stores in zones
12. **Merchandise** - Represents products sold in shops
13. **Maintenance** - Represents ride repair records

**RELATIONSHIPS:**

1. One Customer can purchase One Tickets (One-to-One)
2. One Customer can use One Parking slots (One-to-One)
3. One Zone can contain multiple Rides (One-to-Many)
4. One Ride requires multiple SafetyInspections (One-to-Many)
5. One Department employs multiple Employees (One-to-Many)
6. One Customer can ride multiple Rides (Many-to-Many via CustomerRide)
7. One Zone can have multiple FoodOutlets (One-to-Many)
8. One FoodOutlet offers multiple FoodItems (One-to-Many)
9. One Customer can buy multiple FoodItems (Many-to-Many via BuysFood)
10. One Zone can have multiple GiftShops (One-to-Many)
11. One GiftShop stocks multiple Merchandise items (One-to-Many)
12. One Customer can buy multiple Merchandise items (Many-to-Many via BuysMerchandise)
13. One Ride needs multiple Maintenance records (One-to-Many)
14. One Employee can perform multiple Maintenance tasks (One-to-Many)
15. One Employee can conduct multiple SafetyInspections (One-to-Many)





TABLES:

1)Customer:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| CUST\_ID | NUMERIC | Not null, Primary key |
| NAME | VARCHAR | Not null |
| PHONE\_NO | VARCHAR |  |
| AGE | NUMERIC |  |
| HEIGHT | NUMERIC |  |
| WEIGHT | NUMERIC |  |

2)Ticket:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| TICKET\_ID | NUMERIC | Not null, Primary key |
| CUST\_ID | NUMERIC | Not null, Foreign Key (Customer) |
| TICKET\_TYPE | VARCHAR | Not null |
| PRICE | NUMERIC | Not null |

3)Parking:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| LOT\_NO | NUMERIC | Not null, Primary key |
| PARK\_DATE | DATE | Not null |
| PRICE | NUMERIC | Not null |
| CUST\_ID | NUMERIC | Not null, Foreign Key (Customer) |

4)Zone:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| ZONE\_ID | NUMERIC | Not null, Primary key |
| ZONE\_NAME | VARCHAR | Not null |
| THEME | VARCHAR |  |
| MANAGER\_ID | NUMERIC | Foreign Key (Employee) |

5)Ride:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| RIDE\_ID | NUMERIC | Not null, Primary key |
| RIDE\_NAME | VARCHAR | Not null |
| MIN\_AGE | NUMERIC |  |
| MAX\_AGE | NUMERIC |  |
| MIN\_HEIGHT | NUMERIC |  |
| MAX\_HEIGHT | NUMERIC |  |
| ZONE\_ID | NUMERIC | Not null, Foreign Key (Zone) |

6)Safety\_Insepection:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| INSPECTION\_ID | NUMERIC | Not null, Primary key |
| DESCRIPTION | VARCHAR | Not null |
| INSPECTION\_DATE | DATE | Not null |
| RIDE\_ID | NUMERIC | Not null, Foreign Key (Ride) |
| EMP\_ID | NUMERIC | Not null, Foreign Key (Employee) |

7)Department:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| DEPT\_ID | NUMERIC | Not null, Primary key |
| DEPT\_NAME | VARCHAR | Not null |
| MANAGER\_ID | NUMERIC | Foreign Key (Employee) |

**8) EMPLOYEE:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| EMP\_ID | NUMERIC | Not null, Primary key |
| NAME | VARCHAR | Not null |
| PHONE\_NO | VARCHAR |  |
| EMAIL\_ID | VARCHAR |  |
| DOB | DATE |  |
| HIRE\_DATE | DATE |  |
| SALARY | NUMERIC |  |
| DEPT\_ID | NUMERIC | Not null, Foreign Key (Department) |

**9) CUSTOMER\_RIDE:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| CUST\_ID | NUMERIC | Not null, Foreign Key (Customer) |
| RIDE\_ID | NUMERIC | Not null, Foreign Key (Ride) |
| RATING | NUMERIC |  |
| PRIMARY KEY (CUST\_ID, RIDE\_ID) |  |  |

|  |
| --- |
|  |

**10) FOOD\_OUTLET:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| OUTLET\_ID | NUMERIC | Not null, Primary key |
| OUTLET\_NAME | VARCHAR | Not null |
| ZONE\_ID | NUMERIC | Not null, Foreign Key (Zone) |

**11) FOOD\_ITEM:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| FOOD\_ID | NUMERIC | Not null, Primary key |
| FOOD\_NAME | VARCHAR | Not null |
| SELLING\_PRICE | NUMERIC | Not null |
| OUTLET\_ID | NUMERIC | Not null, Foreign Key (FoodOutlet) |

**12) BUYS\_FOOD:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| CUST\_ID | NUMERIC | Not null, Foreign Key (Customer) |
| FOOD\_ID | NUMERIC | Not null, Foreign Key (FoodItem) |
| QUANTITY | NUMERIC | Not null |
| PRIMARY KEY (CUST\_ID, FOOD\_ID) |  |  |

**13) GIFT\_SHOP:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| SHOP\_ID | NUMERIC | Not null, Primary key |
| SHOP\_NAME | VARCHAR | Not null |
| ZONE\_ID | NUMERIC | Not null, Foreign Key (Zone) |

**14) MERCHANDISE:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| MERCH\_ID | NUMERIC | Not null, Primary key |
| MERCHANDISE\_NAME | VARCHAR | Not null |
| SELLING\_PRICE | NUMERIC | Not null |
| SHOP\_ID | NUMERIC | Not null, Foreign Key (GiftShop) |

**15) BUYS\_MERCHANDISE:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| CUST\_ID | NUMERIC | Not null, Foreign Key (Customer) |
| MERCH\_ID | NUMERIC | Not null, Foreign Key (Merchandise) |
| QUANTITY | NUMERIC | Not null |
| PRIMARY KEY (CUST\_ID, MERCH\_ID) |  |  |

**16) MAINTENANCE:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Constraints** |
| MAINTENANCE\_ID | NUMERIC | Not null, Primary key |
| DESCRIPTION | VARCHAR | Not null |
| MAINTENANCE\_DATE | DATE | Not null |
| EMP\_ID | NUMERIC | Not null, Foreign Key (Employee) |
| RIDE\_ID | NUMERIC | Not null, Foreign Key (Ride) |

NORMALIZAITON:

1) CUSTOMER

Cust\_ID → (Name, Phone\_No, Age, Height, Weight)

**Primary Key** = Cust\_ID  
**Prime Attributes** = Cust\_ID  
**Non-Prime Attributes** = Name, Phone\_No, Age, Height, Weight

* No multivalued/composite attributes → **1NF**
* No partial dependencies (single-column PK) → **2NF**
* No transitive dependencies → **3NF**
* All dependencies come from candidate key → **BCNF**

2) TICKET

Ticket\_ID → (Cust\_ID, Ticket\_Type, Price)

**Primary Key** = Ticket\_ID  
**Prime Attributes** = Ticket\_ID  
**Non-Prime Attributes** = Cust\_ID, Ticket\_Type, Price

* No multivalued/composite attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

3) PARKING

Lot\_No → (Park\_Date, Price, Cust\_ID)

**Primary Key** = Lot\_No  
**Prime Attributes** = Lot\_No  
**Non-Prime Attributes** = Park\_Date, Price, Cust\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

4) ZONE

Zone\_ID → (Zone\_Name, Theme, Manager\_ID)

Manager\_ID->Zone\_ID

**Primary Key** = Zone\_ID  
**Prime Attributes** = Zone\_ID,Manager\_ID  
**Non-Prime Attributes** = Zone\_Name, Theme

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

5) RIDE

Ride\_ID → (Ride\_Name, Min\_Age, Max\_Age, Min\_Height, Max\_Height, Zone\_ID)

**Primary Key** = Ride\_ID  
**Prime Attributes** = Ride\_ID  
**Non-Prime Attributes** = Ride\_Name, Min\_Age, Max\_Age, Min\_Height, Max\_Height, Zone\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

6) SAFETY\_INSPECTION

Inspection\_ID → (Description, Inspection\_Date, Ride\_ID, Emp\_ID)

**Primary Key** = Inspection\_ID  
**Prime Attributes** = Inspection\_ID  
**Non-Prime Attributes** = Description, Inspection\_Date, Ride\_ID, Emp\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

7) DEPARTMENT

Dept\_ID → (Dept\_Name, Manager\_ID)

Manager\_ID->Dept\_ID

**Primary Key** = Dept\_ID  
**Prime Attributes** = Dept\_ID,Manager\_ID  
**Non-Prime Attributes** = Dept\_Name

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

8) EMPLOYEE

Emp\_ID → (Name, Phone\_No, Email\_ID, DOB, Hire\_Date, Salary, Dept\_ID)

**Primary Key** = Emp\_ID  
**Prime Attributes** = Emp\_ID  
**Non-Prime Attributes** = Name, Phone\_No, Email\_ID, DOB, Hire\_Date, Salary, Dept\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

9) CUSTOMER\_RIDE

(Cust\_ID, Ride\_ID) → Rating

**Composite Primary Key** = (Cust\_ID, Ride\_ID)  
**Prime Attributes** = Cust\_ID, Ride\_ID  
**Non-Prime Attributes** = Rating

* No multivalued attributes → **1NF**
* No partial dependencies (all non-prime depend on full PK) → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

10) FOOD\_OUTLET

Outlet\_ID → (Outlet\_Name, Zone\_ID)

**Primary Key** = Outlet\_ID  
**Prime Attributes** = Outlet\_ID  
**Non-Prime Attributes** = Outlet\_Name, Zone\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

11) FOOD\_ITEM

Food\_ID → (Food\_Name, Selling\_Price, Outlet\_ID)

**Primary Key** = Food\_ID  
**Prime Attributes** = Food\_ID  
**Non-Prime Attributes** = Food\_Name, Selling\_Price, Outlet\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

12) BUYS\_FOOD

(Cust\_ID, Food\_ID) → Quantity

**Composite Primary Key** = (Cust\_ID, Food\_ID)  
**Prime Attributes** = Cust\_ID, Food\_ID  
**Non-Prime Attributes** = Quantity

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

13) GIFT\_SHOP

Shop\_ID → (Shop\_Name, Zone\_ID)

**Primary Key** = Shop\_ID  
**Prime Attributes** = Shop\_ID  
**Non-Prime Attributes** = Shop\_Name, Zone\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

14) MERCHANDISE

Merch\_ID → (Merchandise\_Name, Selling\_Price, Shop\_ID)

**Primary Key** = Merch\_ID  
**Prime Attributes** = Merch\_ID  
**Non-Prime Attributes** = Merchandise\_Name, Selling\_Price, Shop\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

15) BUYS\_MERCHANDISE

(Cust\_ID, Merch\_ID) → Quantity

**Composite Primary Key** = (Cust\_ID, Merch\_ID)  
**Prime Attributes** = Cust\_ID, Merch\_ID  
**Non-Prime Attributes** = Quantity

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

16) MAINTENANCE

Maintenance\_ID → (Description, Maintenance\_Date, Emp\_ID, Ride\_ID)

**Primary Key** = Maintenance\_ID  
**Prime Attributes** = Maintenance\_ID  
**Non-Prime Attributes** = Description, Maintenance\_Date, Emp\_ID, Ride\_ID

* No multivalued attributes → **1NF**
* No partial dependencies → **2NF**
* No transitive dependencies → **3NF**
* All dependencies from candidate key → **BCNF**

CREATING TABLES:

1)DEPARTMENT:

CREATE TABLE Department (

DeptID INT PRIMARY KEY,

DeptNameVARCHAR(50) NOT NULL,

managerID INT

);

INSERTING VALUES:

INSERT INTO Department VALUES (1, 'Operations', NULL);

INSERT INTO Department VALUES (2, 'Maintenance', NULL);

INSERT INTO Department VALUES (3, 'Food Services', NULL);

INSERT INTO Department VALUES (4, 'Retail', NULL);

INSERT INTO Department VALUES (5, 'Safety', NULL);

ADD FOREIGN KEY TO DEPARTMENT TABLE AFTER CREATING EMPLOYEE TABLE:

ALTER TABLE Department ADD CONSTRAINT fk\_dept\_manager

FOREIGN KEY (managerID) REFERENCES Employee(EmpID) DEFERRABLE INITIALLY DEFERRED;

UPDATE Department SET managerID = 101 WHERE DeptID = 1;

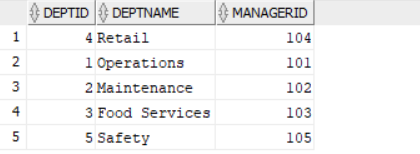
UPDATE Department SET managerID = 102 WHERE DeptID = 2;

UPDATE Department SET managerID = 103 WHERE DeptID = 3;

UPDATE Department SET managerID = 104 WHERE DeptID = 4;

UPDATE Department SET managerID = 105 WHERE DeptID = 5;

OUTPUT:



2)EMPLOYEE:

CREATE TABLE Employee (

EmpID INT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

PhoneNoVARCHAR(20),

EmailIDVARCHAR(100),

DOB DATE,

HireDate DATE,

Salary INT,

DeptID INT,

CONSTRAINT fk\_emp\_dept FOREIGN KEY (DeptID) REFERENCES Department(DeptID)

);

INSERTING VALUES:

INSERT INTO Employee VALUES (101, 'John Smith', '555-0101', 'john.smith@park.com', TO\_DATE('1980-05-15', 'YYYY-MM-DD'), TO\_DATE('2010-06-20', 'YYYY-MM-DD'), 65000, 1);

INSERT INTO Employee VALUES (102, 'Sarah Johnson', '555-0102', 'sarah.j@park.com', TO\_DATE('1985-08-22', 'YYYY-MM-DD'), TO\_DATE('2012-03-15', 'YYYY-MM-DD'), 70000, 2);

INSERT INTO Employee VALUES (103, 'Mike Brown', '555-0103', 'mike.b@park.com', TO\_DATE('1978-11-10', 'YYYY-MM-DD'), TO\_DATE('2015-07-10', 'YYYY-MM-DD'), 60000, 3);

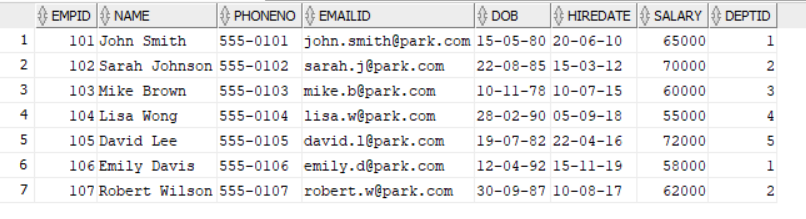
INSERT INTO Employee VALUES (104, 'Lisa Wong', '555-0104', 'lisa.w@park.com', TO\_DATE('1990-02-28', 'YYYY-MM-DD'), TO\_DATE('2018-09-05', 'YYYY-MM-DD'), 55000, 4);

INSERT INTO Employee VALUES (105, 'David Lee', '555-0105', 'david.l@park.com', TO\_DATE('1982-07-19', 'YYYY-MM-DD'), TO\_DATE('2016-04-22', 'YYYY-MM-DD'), 72000, 5);

INSERT INTO Employee VALUES (106, 'Emily Davis', '555-0106', 'emily.d@park.com', TO\_DATE('1992-04-12', 'YYYY-MM-DD'), TO\_DATE('2019-11-15', 'YYYY-MM-DD'), 58000, 1);

INSERT INTO Employee VALUES (107, 'Robert Wilson', '555-0107', 'robert.w@park.com', TO\_DATE('1987-09-30', 'YYYY-MM-DD'), TO\_DATE('2017-08-10', 'YYYY-MM-DD'), 62000, 2);

OUTPUT:



3)CUSTOMER:

CREATE TABLE Customer (

CustID INT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

PhoneNoVARCHAR(20),

Age INT,

Height INT,

Weight INT

);

INSERTING VALUES:

INSERT INTO Customer VALUES (1001, 'Alice Johnson', '555-0201', 25, 165, 60);

INSERT INTO Customer VALUES (1002, 'Bob Williams', '555-0202', 32, 180, 75);

INSERT INTO Customer VALUES (1003, 'Carol Brown', '555-0203', 12, 140, 40);

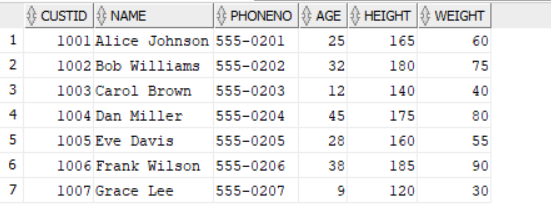
INSERT INTO Customer VALUES (1004, 'Dan Miller', '555-0204', 45, 175, 80);

INSERT INTO Customer VALUES (1005, 'Eve Davis', '555-0205', 28, 160, 55);

INSERT INTO Customer VALUES (1006, 'Frank Wilson', '555-0206', 38, 185, 90);

INSERT INTO Customer VALUES (1007, 'Grace Lee', '555-0207', 9, 120, 30);

OUTPUT:



4)TICKET:

CREATE TABLE Ticket (

TicketID INT PRIMARY KEY,

CustID INT NOT NULL,

TicketTypeVARCHAR(50),

Price INT,

CONSTRAINT fk\_ticket\_customer FOREIGN KEY (CustID) REFERENCES Customer(CustID)

);

INSERTING VALUES:

INSERT INTO Ticket VALUES (5001, 1001, 'Adult', 75);

INSERT INTO Ticket VALUES (5002, 1002, 'Adult', 75);

INSERT INTO Ticket VALUES (5003, 1003, 'Child', 50);

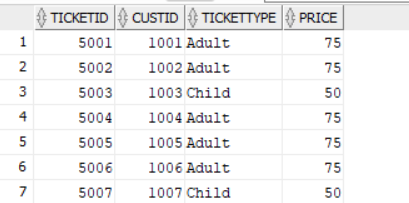
INSERT INTO Ticket VALUES (5004, 1004, 'Adult', 75);

INSERT INTO Ticket VALUES (5005, 1005, 'Adult', 75);

INSERT INTO Ticket VALUES (5006, 1006, 'Adult', 75);

INSERT INTO Ticket VALUES (5007, 1007, 'Child', 50);

OUTPUT:



5)PARKING:

CREATE TABLE Parking (

LotNo INT PRIMARY KEY,

ParkDate DATE,

Price INT,

CustID INT,

CONSTRAINT fk\_parking\_customer FOREIGN KEY (CustID) REFERENCES Customer(CustID)

);

INSERTING VALUES:

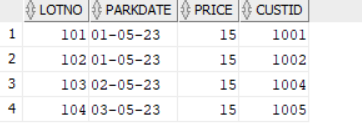
INSERT INTO Parking VALUES (101, TO\_DATE('2023-05-01', 'YYYY-MM-DD'), 15, 1001);

INSERT INTO Parking VALUES (102, TO\_DATE('2023-05-01', 'YYYY-MM-DD'), 15, 1002);

INSERT INTO Parking VALUES (103, TO\_DATE('2023-05-02', 'YYYY-MM-DD'), 15, 1004);

INSERT INTO Parking VALUES (104, TO\_DATE('2023-05-03', 'YYYY-MM-DD'), 15, 1005);

OUTPUT:



6)ZONE:

CREATE TABLE Zone (

ZoneID INT PRIMARY KEY,

ZoneNameVARCHAR(50) NOT NULL,

Theme VARCHAR(50),

managerID INT,

CONSTRAINT fk\_zone\_manager FOREIGN KEY (managerID) REFERENCES Employee(EmpID)

);

INSERTING VALUES:

INSERT INTO Zone VALUES (1, 'Adventure Land', 'Jungle', 101);

INSERT INTO Zone VALUES (2, 'Fantasy World', 'Fairytale', 106);

INSERT INTO Zone VALUES (3, 'Future City', 'Sci-Fi', 101);

INSERT INTO Zone VALUES (4, 'Wild West', 'Western', 106);

INSERT INTO Zone VALUES (5, 'Water World', 'Aquatic', 101);

OUTPUT:



7)RIDE:

CREATE TABLE Ride (

RideID INT PRIMARY KEY,

RideNameVARCHAR(100) NOT NULL,

MinAge INT,

MaxAge INT,

MinHeight INT,

MaxHeight INT,

ZoneID INT,

CONSTRAINT fk\_ride\_zone FOREIGN KEY (ZoneID) REFERENCES Zone(ZoneID)

);

INSERTING VALUES:

INSERT INTO Ride VALUES (2001, 'Jungle Cruise', 8, 70, 120, 200, 1);

INSERT INTO Ride VALUES (2002, 'Dragon Coaster', 10, 65, 140, 200, 2);

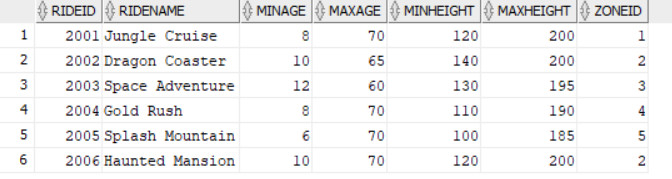
INSERT INTO Ride VALUES (2003, 'Space Adventure', 12, 60, 130, 195, 3);

INSERT INTO Ride VALUES (2004, 'Gold Rush', 8, 70, 110, 190, 4);

INSERT INTO Ride VALUES (2005, 'Splash Mountain', 6, 70, 100, 185, 5);

INSERT INTO Ride VALUES (2006, 'Haunted Mansion', 10, 70, 120, 200, 2);

OUTPUT:



8)SAFETYINSPECTION:

CREATE TABLE SafetyInspection (

InspectionID INT PRIMARY KEY,

Description VARCHAR(200),

InspectionDate DATE,

RideID INT NOT NULL,

EmpID INT NOT NULL,

CONSTRAINT fk\_inspection\_ride FOREIGN KEY (RideID) REFERENCES Ride(RideID),

CONSTRAINT fk\_inspection\_employee FOREIGN KEY (EmpID) REFERENCES Employee(EmpID)

);

INSERTING VALUES:

INSERT INTO SafetyInspection VALUES (3001, 'Routine monthly inspection', TO\_DATE('2023-04-15', 'YYYY-MM-DD'), 2001, 105);

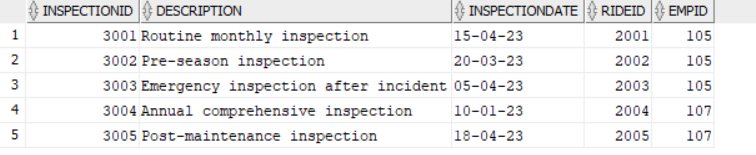
INSERT INTO SafetyInspection VALUES (3002, 'Pre-season inspection', TO\_DATE('2023-03-20', 'YYYY-MM-DD'), 2002, 105);

INSERT INTO SafetyInspection VALUES (3003, 'Emergency inspection after incident', TO\_DATE('2023-04-05', 'YYYY-MM-DD'), 2003, 105);

INSERT INTO SafetyInspection VALUES (3004, 'Annual comprehensive inspection', TO\_DATE('2023-01-10', 'YYYY-MM-DD'), 2004, 107);

INSERT INTO SafetyInspection VALUES (3005, 'Post-maintenance inspection', TO\_DATE('2023-04-18', 'YYYY-MM-DD'), 2005, 107);

OUTPUT:



9)CUSTOMERRIDE:

CREATE TABLE CustomerRide (

CustID INT,

RideID INT,

Rating INT CHECK (Rating BETWEEN 1 AND 5),

PRIMARY KEY (CustID, RideID),

CONSTRAINT fk\_customerride\_customer FOREIGN KEY (CustID) REFERENCES Customer(CustID),

CONSTRAINT fk\_customerride\_ride FOREIGN KEY (RideID) REFERENCES Ride(RideID)

);

INSERTING VALUES:

INSERT INTO CustomerRide VALUES (1001, 2001, 5);

INSERT INTO CustomerRide VALUES (1001, 2002, 4);

INSERT INTO CustomerRide VALUES (1002, 2001, 3);

INSERT INTO CustomerRide VALUES (1002, 2003, 5);

INSERT INTO CustomerRide VALUES (1003, 2005, 5);

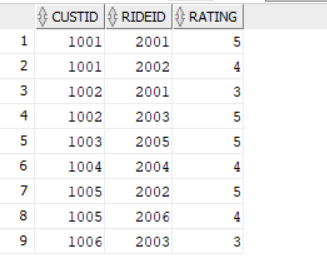
INSERT INTO CustomerRide VALUES (1004, 2004, 4);

INSERT INTO CustomerRide VALUES (1005, 2002, 5);

INSERT INTO CustomerRide VALUES (1005, 2006, 4);

INSERT INTO CustomerRide VALUES (1006, 2003, 3);

OUTPUT:



10)FOODOUTLET:

CREATE TABLE FoodOutlet (

OutletID INT PRIMARY KEY,

OutletNameVARCHAR(100) NOT NULL,

ZoneID INT,

CONSTRAINT fk\_foodoutlet\_zone FOREIGN KEY (ZoneID) REFERENCES Zone(ZoneID)

);

INSERTING VALUES:

INSERT INTO FoodOutlet VALUES (4001, 'Burger Barn', 1);

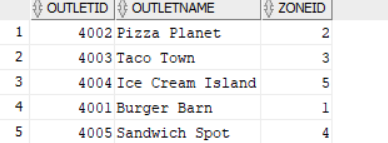
INSERT INTO FoodOutlet VALUES (4002, 'Pizza Planet', 2);

INSERT INTO FoodOutlet VALUES (4003, 'Taco Town', 3);

INSERT INTO FoodOutlet VALUES (4004, 'Ice Cream Island', 5);

INSERT INTO FoodOutlet VALUES (4005, 'Sandwich Spot', 4);

OUTPUT:



11)FOODITEM:

CREATE TABLE FoodItem (

FoodID INT PRIMARY KEY,

FoodNameVARCHAR(100) NOT NULL,

SellingPrice INT,

OutletID INT,

CONSTRAINT fk\_fooditem\_outlet FOREIGN KEY (OutletID) REFERENCES FoodOutlet(OutletID)

);

INSERTING VALUES:

INSERT INTO FoodItem VALUES (6001, 'Cheeseburger', 8, 4001);

INSERT INTO FoodItem VALUES (6002, 'Pepperoni Pizza', 12, 4002);

INSERT INTO FoodItem VALUES (6003, 'Chicken Taco', 5, 4003);

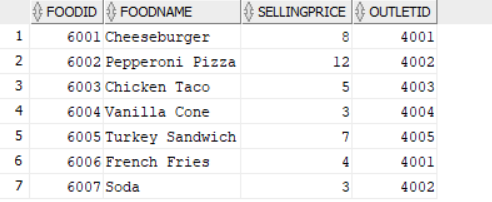
INSERT INTO FoodItem VALUES (6004, 'Vanilla Cone', 3, 4004);

INSERT INTO FoodItem VALUES (6005, 'Turkey Sandwich', 7, 4005);

INSERT INTO FoodItem VALUES (6006, 'French Fries', 4, 4001);

INSERT INTO FoodItem VALUES (6007, 'Soda', 3, 4002);

OUTPUT:



12)BUYSFOOD:

CREATE TABLE BuysFood (

CustID INT,

FoodID INT,

Quantity INT,

PRIMARY KEY (CustID, FoodID),

CONSTRAINT fk\_buysfood\_customer FOREIGN KEY (CustID) REFERENCES Customer(CustID),

CONSTRAINT fk\_buysfood\_fooditem FOREIGN KEY (FoodID) REFERENCES FoodItem(FoodID)

);

INSERTING VALUES:

INSERT INTO BuysFood VALUES (1001, 6001, 2);

INSERT INTO BuysFood VALUES (1001, 6006, 1);

INSERT INTO BuysFood VALUES (1002, 6002, 1);

INSERT INTO BuysFood VALUES (1002, 6007, 2);

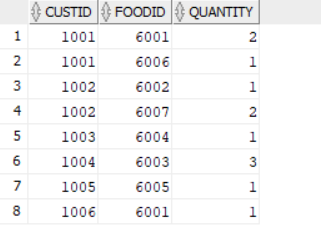
INSERT INTO BuysFood VALUES (1003, 6004, 1);

INSERT INTO BuysFood VALUES (1004, 6003, 3);

INSERT INTO BuysFood VALUES (1005, 6005, 1);

INSERT INTO BuysFood VALUES (1006, 6001, 1);

OUTPUT:



13)GIFTSHOP:

CREATE TABLE GiftShop (

ShopID INT PRIMARY KEY,

ShopNameVARCHAR(100) NOT NULL,

ZoneID INT,

CONSTRAINT fk\_giftshop\_zone FOREIGN KEY (ZoneID) REFERENCES Zone(ZoneID)

);

INSERTING VALUES:

INSERT INTO GiftShop VALUES (7001, 'Adventure Treasures', 1);

INSERT INTO GiftShop VALUES (7002, 'Fantasy Gifts', 2);

INSERT INTO GiftShop VALUES (7003, 'Future Souvenirs', 3);

INSERT INTO GiftShop VALUES (7004, 'Western Wear', 4);

INSERT INTO GiftShop VALUES (7005, 'Beach Shop', 5);

OUTPUT:



14)MERCHANDISE:

CREATE TABLE Merchandise (

MerchID INT PRIMARY KEY,

Merchandise VARCHAR(100) NOT NULL,

SellingPrice INT,

ShopID INT,

CONSTRAINT fk\_merchandise\_shop FOREIGN KEY (ShopID) REFERENCES GiftShop(ShopID)

);

INSERTING VALUES:

INSERT INTO Merchandise VALUES (8001, 'Theme Park T-Shirt', 20, 7001);

INSERT INTO Merchandise VALUES (8002, 'Stuffed Dragon', 15, 7002);

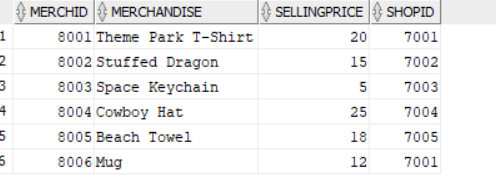
INSERT INTO Merchandise VALUES (8003, 'Space Keychain', 5, 7003);

INSERT INTO Merchandise VALUES (8004, 'Cowboy Hat', 25, 7004);

INSERT INTO Merchandise VALUES (8005, 'Beach Towel', 18, 7005);

INSERT INTO Merchandise VALUES (8006, 'Mug', 12, 7001);

OUTPUT:



15)BUYSMERCHANDISE:

CREATE TABLE BuysMerchandise (

CustID INT,

MerchID INT,

Quantity INT,

PRIMARY KEY (CustID, MerchID),

CONSTRAINT fk\_buysmerch\_customer FOREIGN KEY (CustID) REFERENCES Customer(CustID),

CONSTRAINT fk\_buysmerch\_merch FOREIGN KEY (MerchID) REFERENCES Merchandise(MerchID)

);

INSERTING VALUES:

INSERT INTO BuysMerchandise VALUES (1001, 8001, 1);

INSERT INTO BuysMerchandise VALUES (1002, 8003, 2);

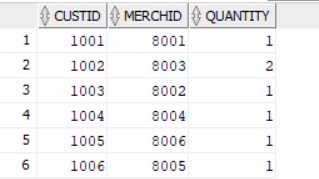
INSERT INTO BuysMerchandise VALUES (1003, 8002, 1);

INSERT INTO BuysMerchandise VALUES (1004, 8004, 1);

INSERT INTO BuysMerchandise VALUES (1005, 8006, 1);

INSERT INTO BuysMerchandise VALUES (1006, 8005, 1);

OUPUT:



16)MAINTENANCE:

CREATE TABLE Maintenance (

MaintenanceID INT PRIMARY KEY,

Description VARCHAR(200),

MaintenanceDate DATE,

EmpID INT,

RideID INT,

CONSTRAINT fk\_maintenance\_employee FOREIGN KEY (EmpID) REFERENCES Employee(EmpID),

CONSTRAINT fk\_maintenance\_ride FOREIGN KEY (RideID) REFERENCES Ride(RideID)

);

INSERTING VALUES:

INSERT INTO Maintenance VALUES (9001, 'Replaced brake system', TO\_DATE('2023-04-10', 'YYYY-MM-DD'), 102, 2001);

INSERT INTO Maintenance VALUES (9002, 'Repainted ride structure', TO\_DATE('2023-03-25', 'YYYY-MM-DD'), 107, 2002);

INSERT INTO Maintenance VALUES (9003, 'Fixed audio system', TO\_DATE('2023-04-05', 'YYYY-MM-DD'), 102, 2003);

INSERT INTO Maintenance VALUES (9004, 'Replaced seat belts', TO\_DATE('2023-04-15', 'YYYY-MM-DD'), 107, 2004);

INSERT INTO Maintenance VALUES (9005, 'Water pump repair', TO\_DATE('2023-04-18', 'YYYY-MM-DD'), 102, 2005);

OUTPUT:

