DBMS Project 1 -Milestone2

Team Name: Data Knights

Members:

Bhargav Deshpande - bbdeshpa@ncsu.edu – 200260617

Mohit Gupta – mgupta6@ncsu.edu - 200262832

Aditya Duneja – aduneja@ncsu.edu - 200254941

Rutvik Kolhe - rkolhe@ncsu.edu – 200258232

Modifications done after MileStone1:

1. **Role-based access management added-**
   Added Attribute "Role" in login table which will store role like Manager, Customer, Mechanic, etc. When User enters username and password and authentication is successful, we will retrieve its Role and will show menu based on this role.

2. **Added logic to capture invoice-**
   The customer invoice would contain the following attributes. The tables from which they would be retrieved are originally present in the database. Logic to display invoice details will be added in Java. The following information contains attribute names and name of the table from which they are obtained:

   Service Center Details (Table: Service Center)
   Name: Center_Name->Service_Center
   Address: Center_Address-> Service_Center
   Phone: Center_Tele#-> Service_Center

   Service/Repair Appointment Details (Table: Appointment)
   ID: appointmentID
   Date In: Date

   Customer Contact Details (Table: Customer)
   Name: Cust_Name
   Email: Cust_email
   Address: Cust_Address
   Phone: Cust_phone

   Mechanic who worked on the car (Table: Employee)
   ID: emp_id

Name: emp_name
Phone: emp_phone

Car Details (Table: Cars)
License number: LicensePlateID
Model: Car_type
Year: datePurchase

Service Details (Table: Service):
Service Type(A/B/C/D/Repair): service_type
Amount to be paid: Total_fees

3. **Attributes modified-**
   Attributes names modified to be more relevant to the project description, made it more consistent. Data types of the attributes modified according to the project description. For Example, earlier we thought LicensePlateID will be numeric, now modified to string as it can be alpha-numeric

4. **Added Check, Not Null and Unique constraints-**
   Added constraints as per application description. For Example, Mechanic can work only 11 hours a day. So, can work maximum 11*15= 165, check constraint added so that hours worked should not be more than 165. Not Null constraints added Password and role can not be Null.

5. **Modified incorrect FDs-**
   Based on the feedback we received for Milestone1, we have corrected functional dependencies. For Example, PartName is can be determined by PartId only. CenterId is not required for it.

6. **Modified E-R diagram-**
   ER Diagram modified accordingly to incorporate these changes

7. **Use of triggers-**
   We are working on addition of trigger. When partId quantity will be less than threshold, new row will be inserted in Order table which will resemble a new order request. This implementation will be done by trigger.

8. **Use of View-**
   We are working on use of views. We are thinking if we can use Views so that we can show specific data for the specific roles. We will create Views based on the roles. This will help doing role - access management.

9. **Acknowledgement statement added**
   Acknowledgment statement added which we mistakenly missed in Milestone1 report.

Details:

Entities:

Login	Customer	Employee	Monthly_paid_emp
Mechanic_emp	Service Center	Cars	OrderPart
Distributor	Distributor	Service	Maintenance
Repair			

Weak Entities:

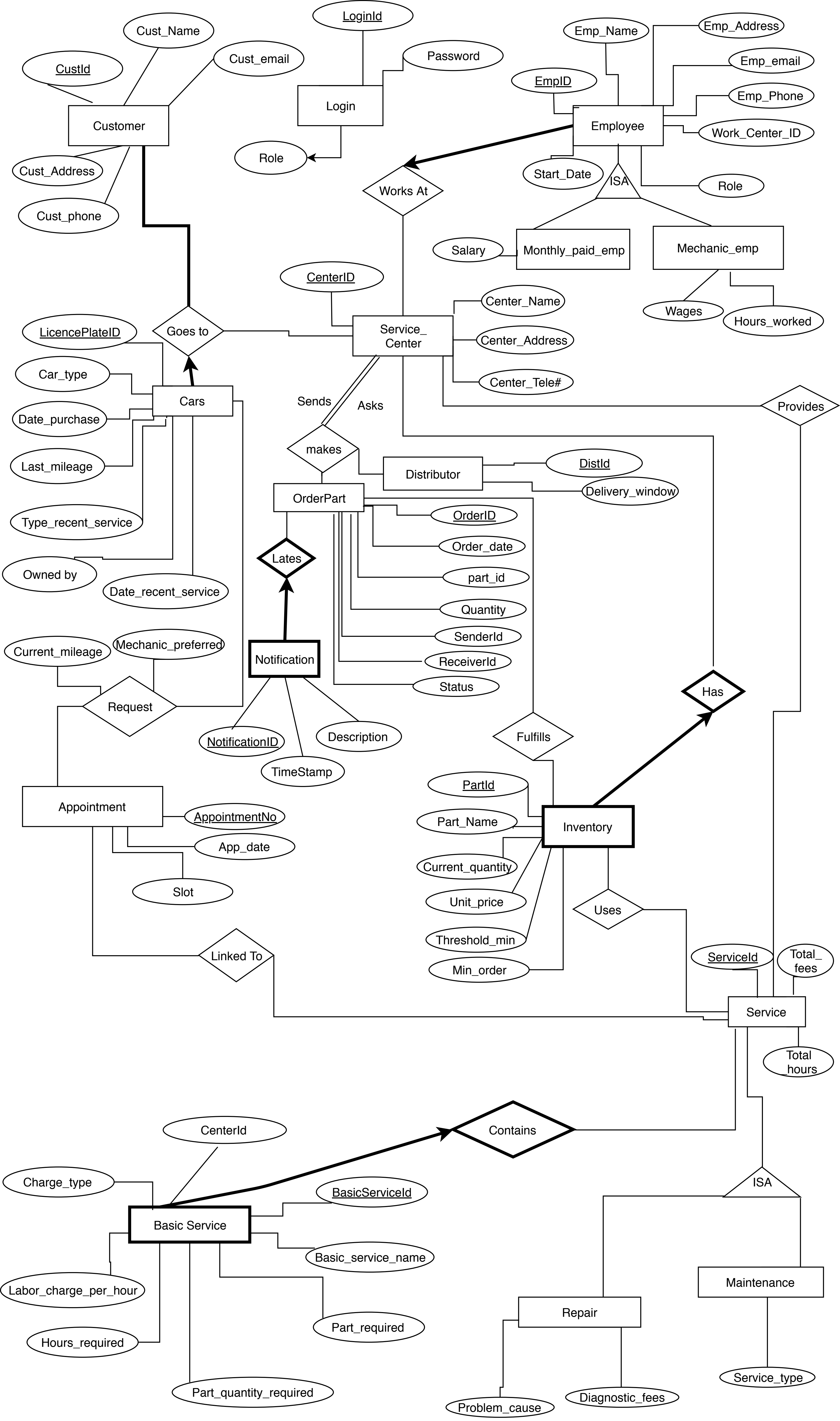
Notification	Inventory	Basic_Service

Relationships:

Binary Relationships: WorksAt, Provides, Lates, Request, Fulfills, Has, Linked To, Uses, Contains
Ternary Relationships: GoesTo, makes

Hierarchies:

- Monthly_paid_emp, Mechanic_emp ISA Employee
- Repair, Maintainance ISA Service



Cust_Name
CustId
Cust_email
Customer
Cust_Address
Cust_phone
LoginId
Password
Login
Role
Emp_Name
Emp_Address
Emp_email
EmpID
Emp_Phone
Employee
Work_Center_ID
Start_Date
ISA
Role
Works At
Salary
Monthly_paid_emp
Mechanic_emp
Wages
Hours_worked
CenterID
Center_Name
Service_ Center
Center_Address
Center_Tele#
LicencePlateID
Goes to
Car_type
Cars
Date_purchase
Last_mileage
Type_recent_service
Owned by
Date_recent_service
Sends
Asks
Provides
makes
Distributor
DistId
Delivery_window
OrderPart
OrderID
Order_date
part_id
Quantity
SenderId
ReceiverId
Status
Lates
Current_mileage
Mechanic_preferred
Notification
Request
NotificationID
Description
TimeStamp
Has
Fulfills
PartId
Part_Name
Inventory
Current_quantity
Unit_price
Threshold_min
Min_order
Uses
Appointment
AppointmentNo
App_date
Slot
Linked To
ServiceId
Total_ fees
Service
Total hours
Contains
CenterId
Charge_type
ISA
BasicServiceId
Basic Service
Basic_service_name
Labor_charge_per_hour
Part_required
Hours_required
Part_quantity_required
Maintenance
Repair
Service_type
Problem_cause
Diagnostic_fees

Relational Model and Description

- **SERVICE CENTER** (CenterID: int, Center_Name: string, Center_Address: String, Center_Tele: String)

  Functional Dependencies:
  CenterID -> Centre_Name, Center_Address, Center_Tele

  Constraints:

  Primary Key: CenterID

  Normal Form: 1 NF

- **INVENTORY**(CenterID: int, PartID: int, Part_Name: string, Current_quantity: int, Unit_price: float, Threshold_Min: int, Min_Order: int)

  Functional Dependencies:
  (CenterID, PartId) -> Part_Name,Current_quantity, Unit_price, Threshold_Min, Min_Order
  PartID -> Part_Name

  Constraints:

  Primary Key: (CenterID, PartID)

  Foreign Key: CenterID

  Normal Form: 1 NF

- **CUSTOMER** (CustID: int, Cust_Name: string, Cust_email: string, Cust_address: string, Cust_Phone : String)

  Functional Dependencies:

  CustID -> Cust_Name, Cust_email, Cust_address, Cust_Phone

  Cust_email -> Cust_Name, Cust_address, Cust_Phone

  Constraints:

  Primary Key: (CustID)

  Candidate Key: Cust_Email

  Not NULL : Cust_Email

  Normal Form: 2 NF

- **EMPLOYEE** (EmpID: int, Emp_Name: string, Emp_email: string, Emp_address: string, Emp_Phone : String, Role: String)

  Functional Dependencies:

  EmpID -> Emp_Name, Emp_email, Emp_address, Emp_Phone, Role

  Empl_email-> Emp_Name, Emp_address, Emp_Phone , Role

  Constraints:

  Primary Key: (EmpID)

  Candidate Key: Emp_Email

  Not NULL : Emp_Email

  Normal Form: 2 NF

- **MONTHLY_PAID_EMP** (EmpID: int, Salary: String)

  Functional Dependencies:

  EmpID -> Salary

  Primary Key: (EmpID)

  Normal Form: BCNF

- **MECHANIC_EMP** (EmpID: int, Wages: float, Hours_Worked: float)

  Functional Dependencies:

  EmpID -> Wages, Hours_Worked

  Primary Key: (EmpID)

  Normal Form: BCNF

- **CARS** (LicensePlateID: int, Car_Type: string, Date_Purchase: date, Last_Mileage: int, Type_Recent_Service : String, OwnedBy: String, Date_Recent_Service: Date)

  Functional Dependencies:

  LicensePlateID -> Car_Type, Date_Purchase, Last_Mileage, Type_Recent_Service, OwnedBY, Date_Recent_Service

  Constraints:

Primary Key: LicensePlateID

Normal Form: BCNF

- **SERVICE** (ServiceID: int, Total_Fees: float, Total_Hours: float)

  Functional Dependencies:

  ServiceID -> Total_Fees, Total_Hours

  Constraints:

  Primary Key: ServiceID

  Normal Form: BCNF

- **REPAIR** (ServiceID: int, Problem_Cause: String, Diagnostic_Fees: float)

  Functional Dependencies:

  ServiceID -> Problem_Cause, Diagnostic_Fees

  Primary Key: (ServiceID)

  Normal Form: BCNF

- **MAINTENANCE** (ServiceID: int, Service_Type: String)

  Functional Dependencies:

  ServiceID -> Service_Type

  Primary Key: (ServiceID)

  Normal Form: BCNF

- **BASIC SERVICE** (ServiceID: int, BasicServiceID:int , Basic_Service_Name: string, Part_Required: int, CenterID: int, Part_Quantity_Required: int, Labour_Charge_Per_Hour: float, Hours_ Required: float)

  Functional Dependencies:

  ServiceID, BasicServiceID -> Basic_Service_Name, Part_Required, Part_Quantity_Required, Labour_Charge_Per_Hour, Hours_ Required

  Basic_Service_ID -> Basic_Service_Name, Part_Required, Part_Quantity_Required, Labour_Charge_Per_Hour, Hours_ Required

  Constraints:

  Primary Key: (ServiceID, Basic_Service_ID)

Foreign Key : ServiceID, PartID, CenterID

Normal Form: 3 NF

- **ORDER PART** (OrderID: int, CenterID:int, Order_date: Date, Part_ID: int , quantiy: string, SenderID : int, ReceiverID: int, Status: String)

  Functional Dependencies:

  OrderID -> Order_date, Part_ID, quantiy, SenderID, ReceiverID, Status

  Constraints:

  Primary Key: (OrderID)

  Foreign Key: PartID, CentreID

  Normal Form: BCNF

- **NOTIFICATION** (OrderId:int, NotificationID: int, TimeStamp:string , description: string)

  Functional Dependencies:

  OrderID, NotificationID -> TimeStamp, description

  Constraints:

  Primary Key: (OrderID, NotificationID)

  Foreign Key : OrderID

  Normal Form: 1 NF

- **DISTRIBUTOR** (DistId:int, Delivery_Window: int)

  Functional Dependencies:

  DistId -> Delivery_Window

  Constraints:

  Primary Key: (DistId)

  Normal Form: BCNF

- **APPOINTMENT** (AppointmentNo:int, date: Date, slot: string)

  Functional Dependencies:

  AppointmentNo -> Date, slot

Constraints:

Primary Key: (AppointmentNo)

Normal Form: BCNF

- **LOGIN** (LoginID:int, password: string)

  Constraints: LoginID -> password

  Primary Key: (LoginID)

  Normal Form: BCNF

- **GOES TO** (LicensePlateID:int, CustID:int, CenterID: int)
  Relationship: Ternary
  Normal Form: BCNF

- **LATES** (NotificationID:int, OrderID:int)
  Relationship: Binary
  Normal Form: BCNF

- **Makes** (CenterID:int, OrderID:int, DistID:int)
  Relationship: Ternary
  Normal Form: BCNF

- **PROVIDES** (CenterID:int, ServiceID:int)
  Relationship:Binary
  Normal Form: BCNF

- **CONTAINS** (BasicSericeID:int, ServiceID:int)
  Relationship:Binary
  Normal Form: 1NF

- **LINKED TO** (Appointmentno:int, ServiceID:int)
  Relationship:Binary
  Normal Form: BCNF

- **USES** (PartID:int, ServiceID:int, CenterID:int)
  Relationship: Binary
  Normal Form: 1NF

- **HAS** (PartID:int, CenterID:int)
  Relationship: Binary

Normal Form: 1 NF

- **REQUEST** (AppointmentNo:int,, LicensePlateID:int, Current_Mileage: float, Mechanic_Preferred : string)
  Relationship: Binary

  Functional Dependencies:

  AppointmentNo -> Current-Mileage, LicencePlateID, MechanicPreferred

  Normal Form: BCNF

- **FULFILLS** (OrderID:int, PartID:int, CenterID:int)
  Relationship: Binary
  Normal Form: BCNF

- **WORKS AT** (EmpID:int, CenterID:int)
  Relationship: Binary
  Normal Form: BCNF

SQL QUERIES

```
CREATE TABLE Service_Center (
   CenterID int NOT NULL PRIMARY KEY,
   Center_Name varchar(255) NOT NULL,
   Center_Address varchar(255),
   Center_Tele varchar(255)
);

CREATE TABLE Inventory (
   CenterID int,
   PartID int ,
   Part_Name varchar(255),
   Current_quantity varchar(255),
   Unit_price float,
   Threshold_Min int,
   Min_Order int,
CONSTRAINT PK_Inventory PRIMARY KEY ( CenterID, PartID),
CONSTRAINT FK_Inventory FOREIGN KEY (CenterID)REFERENCES
Service_Center(CenterID) ON  DELETE CASCADE
);

CREATE TABLE Customer (
   CustID int PRIMARY KEY,
   Cust_Name varchar(255),
   Cust_email varchar(255) NOT NULL UNIQUE,
   Cust_address varchar(255),
   Cust_Phone number(10)
);

CREATE TABLE Employee (
   EmpID int PRIMARY KEY,
   Emp_Name varchar(255),
   Emp_email varchar(255) NOT NULL UNIQUE,
   Emp_address varchar(255),
   Emp_Phone number(10)
);


CREATE TABLE Monthly_Paid_Emp (
   EmpID int,
   Salary varchar(255),
   PRIMARY KEY (EmpID),
   CONSTRAINT FK_Monthly_Paid_Emp FOREIGN KEY (EmpID)
   REFERENCES Employee(EmpID) ON  DELETE CASCADE
);
```

```
CREATE TABLE Service
(
ServiceID int PRIMARY KEY,
Total_Fees float ,
Total_Hours float
);


CREATE TABLE BasicService (
   CenterID int,
   ServiceID int,
   BasicServiceID int,
   Basic_Service_Name varchar(255),
   Part_Required int,
   Part_Quantity_Required int,
   Labour_Charge_Per_Hour float,
   Hours_Required float,
   CONSTRAINT PK_BasicService PRIMARY KEY (ServiceID, BasicServiceID),
   CONSTRAINT FK_BasicService1 FOREIGN KEY (ServiceID)
   REFERENCES Service(ServiceID) ON  DELETE CASCADE,
   CONSTRAINT FK_BasicService2 FOREIGN KEY (Part_Required, CenterID)
   REFERENCES Inventory(PartId, CenterId) ON  DELETE CASCADE
);



CREATE TABLE Mechanic_Emp
(
EmpID int PRIMARY KEY,
Wages float,
Hours_Worked float CHECK(Hours_Worked<=11*15),
CONSTRAINT FK_EMPID FOREIGN KEY (EmpID)
REFERENCES Employee(EmpID) ON  DELETE CASCADE
);


CREATE TABLE Cars
(
LicensePlateID  varchar(255) PRIMARY KEY,
Car_Type varchar(255),
Date_Purchase DATE,
Last_Mileage int,
Type_Recent_Service varchar(255),
OwnedBy varchar(255),
Date_Recent_Service DATE
);



CREATE TABLE Repair
(
ServiceID int PRIMARY KEY,
Problem_Cause varchar(255),
Diagnostic_Fees float,
CONSTRAINT FK_Repair FOREIGN KEY (ServiceID)
REFERENCES Service(ServiceID) ON  DELETE CASCADE
);





CREATE TABLE Maintenance
(
ServiceID int PRIMARY KEY,
Service_Type varchar(255),
CONSTRAINT FK_Maintenance FOREIGN KEY (ServiceID)
REFERENCES Service(ServiceID) ON DELETE CASCADE
);

CREATE TABLE OrderPart(
CenterID int,
OrderID int PRIMARY KEY,
Order_date DATE,
Part_ID  int NOT NULL,
quantity int,
SenderID int NOT NULL,
ReceiverID int NOT NULL,
Status varchar(255),
CHECK(quantity>25),
CONSTRAINT FK_PtID FOREIGN KEY (Part_ID,CenterID)
REFERENCES Inventory(PartId,CenterId) ON DELETE CASCADE
);

CREATE TABLE Notification
(
OrderID int,
NotificationID int ,
Timestamp varchar(255),
description varchar(255),
CONSTRAINT PK_Notification PRIMARY KEY (OrderID, NotificationID),
CONSTRAINT FK_Notification FOREIGN KEY (OrderID)
REFERENCES OrderPart (OrderID) ON DELETE CASCADE
);

CREATE TABLE Distributor
(
DistID int PRIMARY KEY ,
Delivery_Window  int
);

CREATE TABLE Appointment
(
AppointmentNo int PRIMARY KEY,
App_Date DATE,
slot varchar(255)
);

CREATE TABLE Login
(
LoginID int PRIMARY KEY,
Password  varchar(255) NOT NULL,
Role varchar(255) NOT NULL
);




CREATE TABLE  Has
(
PartID int,
CenterID int ,
CONSTRAINT PK_PID PRIMARY KEY (PartID, CenterID),
CONSTRAINT FK_PID FOREIGN KEY (PartID,centerId)
REFERENCES Inventory(PartID,centerId) ON DELETE CASCADE
);



CREATE TABLE GoesTo (
    LicensePlateID varchar(255),
    CustID int,
     CenterID int,
    CONSTRAINT PK_GOESTO PRIMARY KEY (LicensePlateID, CustID,  CenterID),
   CONSTRAINT FK_GOESTO_LicensePlateID FOREIGN KEY (LicensePlateID)
REFERENCES Cars(LicensePlateID)
```

```
   ON DELETE CASCADE,
  CONSTRAINT FK_GOESTO_CustID  FOREIGN KEY (CustID) REFERENCES
CUSTOMER(CustID)
   ON DELETE CASCADE,
   CONSTRAINT FK_GOESTO_CenterID FOREIGN KEY (CenterID) REFERENCES
Service_Center ( CenterID)
   ON DELETE CASCADE
);


CREATE TABLE Lates (
   NotificationID int,
   OrderID int,
   CONSTRAINT PK_Lates PRIMARY KEY (NotificationID , OrderID),
   CONSTRAINT FK_Lates_NotificationID FOREIGN KEY (NotificationID,OrderId)
REFERENCES Notification(NotificationID , OrderID)
   ON DELETE CASCADE
);


CREATE TABLE Makes(
CenterID int,
OrderID int,
DistID int,
CONSTRAINT PK_Makes PRIMARY KEY ( CenterID , OrderID, DistID),
CONSTRAINT FK_Makes_CenterID FOREIGN KEY ( CenterID) REFERENCES
Service_Center( CenterID)
ON DELETE CASCADE,
CONSTRAINT FK_Makes_OrderID  FOREIGN KEY (OrderID) REFERENCES
OrderPart(OrderID)
ON DELETE CASCADE,
CONSTRAINT FK_Makes_DistID FOREIGN KEY (DistID) REFERENCES
Distributor(DistID)
ON DELETE CASCADE
);

CREATE TABLE Contains(
   BasicServiceID int,
   ServiceID int,
   CONSTRAINT PK_Contains_PID PRIMARY KEY (BasicServiceID,ServiceID),
   CONSTRAINT FK_Contains_BasicServiceID FOREIGN KEY (BasicServiceID,ServiceID)
REFERENCES BasicService(BasicServiceID,ServiceID)
   ON DELETE CASCADE
);

CREATE TABLE Provides (
   CenterID int,
   ServiceID int,
   CONSTRAINT PK_Provides PRIMARY KEY ( CenterID , ServiceID),
   CONSTRAINT FK_Provides_CenterID FOREIGN KEY ( CenterID) REFERENCES
Service_Center( CenterID)
   ON DELETE CASCADE,
   CONSTRAINT FK_Provides_ServiceID FOREIGN KEY (ServiceID) REFERENCES
Service(ServiceID)
   ON DELETE CASCADE
);




CREATE TABLE LinkedTo(
   Appointmentno int,
   ServiceID int,
   CONSTRAINT PK_LinedTo PRIMARY KEY (Appointmentno, ServiceID ),
   CONSTRAINT FK_Contains_ServiceID FOREIGN KEY (ServiceID) REFERENCES
Service(ServiceID)
   ON DELETE CASCADE,
   CONSTRAINT FK_Contains_Appointmentno  FOREIGN KEY (Appointmentno)
REFERENCES Appointment(Appointmentno)
   ON DELETE CASCADE
);



CREATE TABLE Uses(
   PartID int,
   ServiceID int,
    CenterID int,
   CONSTRAINT PK_Uses PRIMARY KEY (PartID, ServiceID, CenterID ),
  CONSTRAINT FK_Uses_ServiceID FOREIGN KEY (ServiceID) REFERENCES
Service(ServiceID)
   ON DELETE CASCADE,
   CONSTRAINT FK_Uses_PIDCID FOREIGN KEY (PartID, CenterID) REFERENCES
Inventory(PartID, CenterID)
   ON DELETE CASCADE
);


CREATE TABLE Request(
AppointmentNo int,
LicensePlateID varchar(255),
Current_Mileage float,
Mechanic_Preference varchar(255),
CONSTRAINT PK_Request PRIMARY KEY (Appointmentno,LicensePlateID),
CONSTRAINT FK_Request_Appointmentno FOREIGN KEY (Appointmentno) REFERENCES
Appointment(Appointmentno)
ON DELETE CASCADE,
CONSTRAINT FK_Request_LicensePlateID FOREIGN KEY (LicensePlateID) REFERENCES
Cars(LicensePlateID)
ON DELETE CASCADE
);


CREATE TABLE Fulfills(
OrderID int,
PartID int,
CenterID int,
CONSTRAINT PK_Fulfill PRIMARY KEY (OrderID, PartID, CenterID),
CONSTRAINT FK_Fulfill_OrderID FOREIGN KEY (OrderID) REFERENCES OrderPart
(OrderID)
ON DELETE CASCADE,
CONSTRAINT FK_Fulfill_PIDCID FOREIGN KEY (PartID,CenterID) REFERENCES
Inventory(PartID,CenterID)
ON DELETE CASCADE
);


CREATE TABLE WorksAt(
EmpID int,
CenterID int,
CONSTRAINT PK_WorksAt PRIMARY KEY (EmpID, CenterID),
CONSTRAINT FK_WorksAt_EmpID FOREIGN KEY (EmpID) REFERENCES
Employee(EmpID)
ON DELETE CASCADE,
CONSTRAINT FK_WorksAt_Service_CenterID FOREIGN KEY (CenterID) REFERENCES
Service_Center(CenterID)
ON DELETE CASCADE
);
```

APPLICATION CONSTRAINTS

- Each service center has only 1 manager, 1 receptionist and 5 mechanics
- An employee can work at only 1 service center
- Manager and Receptionist get a monthly salary while Mechanic get an hourly salary
- Mechanic can only work for 11 hrs/day
- Inventory must maintain a minimum quantity threshold for each part
- There is a minimum quantity of parts that must be ordered while placing a new order
- Each distributor has a particular delivery window
- There is a predetermined service schedule based on the number of miles traveled
- Each appointment slot is divided in 30 minutes and there must not be any overlap
- No more than half a day should be allocated to maintenance service
- There is a time gap between offering a slot and the customer confirming it
- Before, scheduling an appointment, the availability of the parts must be checked
- The delivery from the supplier for a particular part should be completed one day prior to customer appointment service date which requires that particular part
- Each part must only be ordered so that if fulfills the exact quantity required
- The application should display menus with respect to the role of the application user- i.e. Manager, Customer

Acknowledgement:

We acknowledge the fact that we have asked all the questions related to the project and there are no ambiguities in the description.

- Bhargav Deshpande
- Mohit Gupta
- Rutvik Kolhe
- Aditya Duneja