Table of Contents

[**Title: Garage shop database design** 2](#_Toc67068283)

[**Author(s)** 2](#_Toc67068284)

[**Database Design Assignment Part A** 3](#_Toc67068285)

[**A.1.** **Case Study Overview** 3](#_Toc67068286)

[5](#_Toc67068287)

[**A.2.** **Database Functions** 5](#_Toc67068288)

[**A.3.** **Data Requirements** 5](#_Toc67068289)

[**A.4.** **Business Rules** 6](#_Toc67068290)

[**Database Design Assignment Part B** 8](#_Toc67068291)

[**B.1.** **Overview of the case study in at most five sentences (From part A)** 8](#_Toc67068292)

[**B.2.** **Revised Business rules and Assumptions** 8](#_Toc67068293)

[**B.3.** **ERD** 8](#_Toc67068294)

[**B.4.** **Justifications** of the ERD based on the business rules and/or assumptions. 8](#_Toc67068295)

[**Database Design Assignment Part C** 9](#_Toc67068296)

[**C. 1.** **Revised Business rules and assumptions (From part B)** 9](#_Toc67068297)

[**C. 2.** **Revised ERD of Part B** 9](#_Toc67068298)

[**C. 3.** **Relations** 9](#_Toc67068299)

[**C. 4.** **List of Functional Dependencies Related to Each Business Rules.** 9](#_Toc67068300)

[**C. 5.** **Normalization** 9](#_Toc67068301)

[**Database Design Assignment Part D** 10](#_Toc67068302)

[**D.1. Created Database** 10](#_Toc67068303)

# **Title: Garage shop database design – group 12**

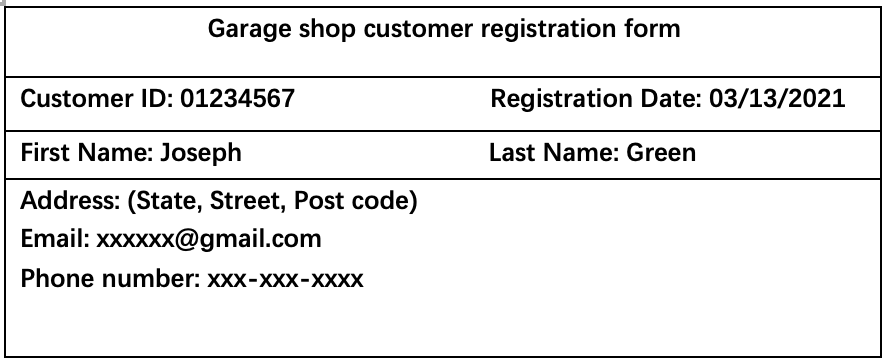
# **Author(s)**

1. **13641283 Jianyin Xu**
2. **13713654 Gongming Shi**
3. **13809987 Yuanchen Shi**

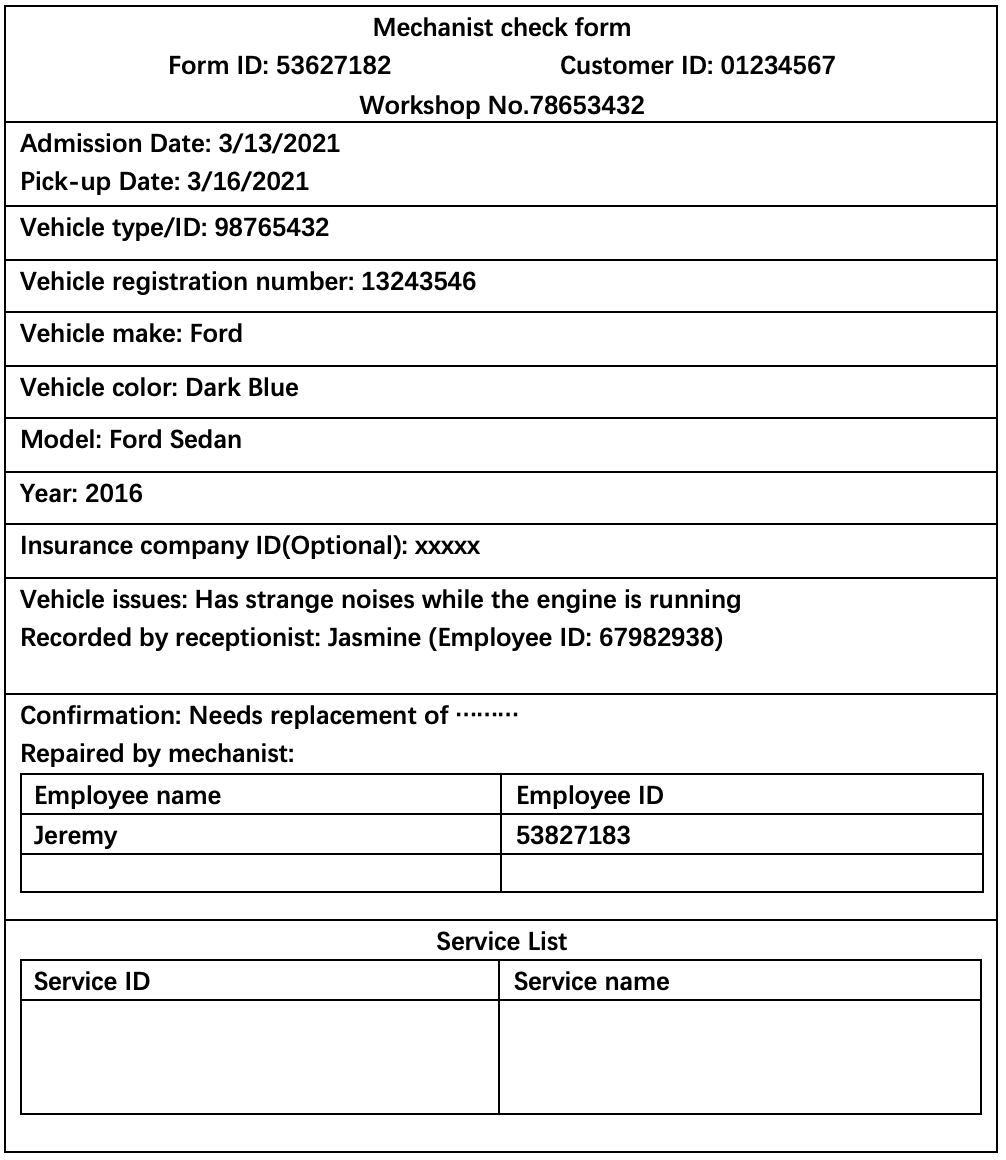
# **Database Design Assignment Part A**

1. **Case Study Overview**

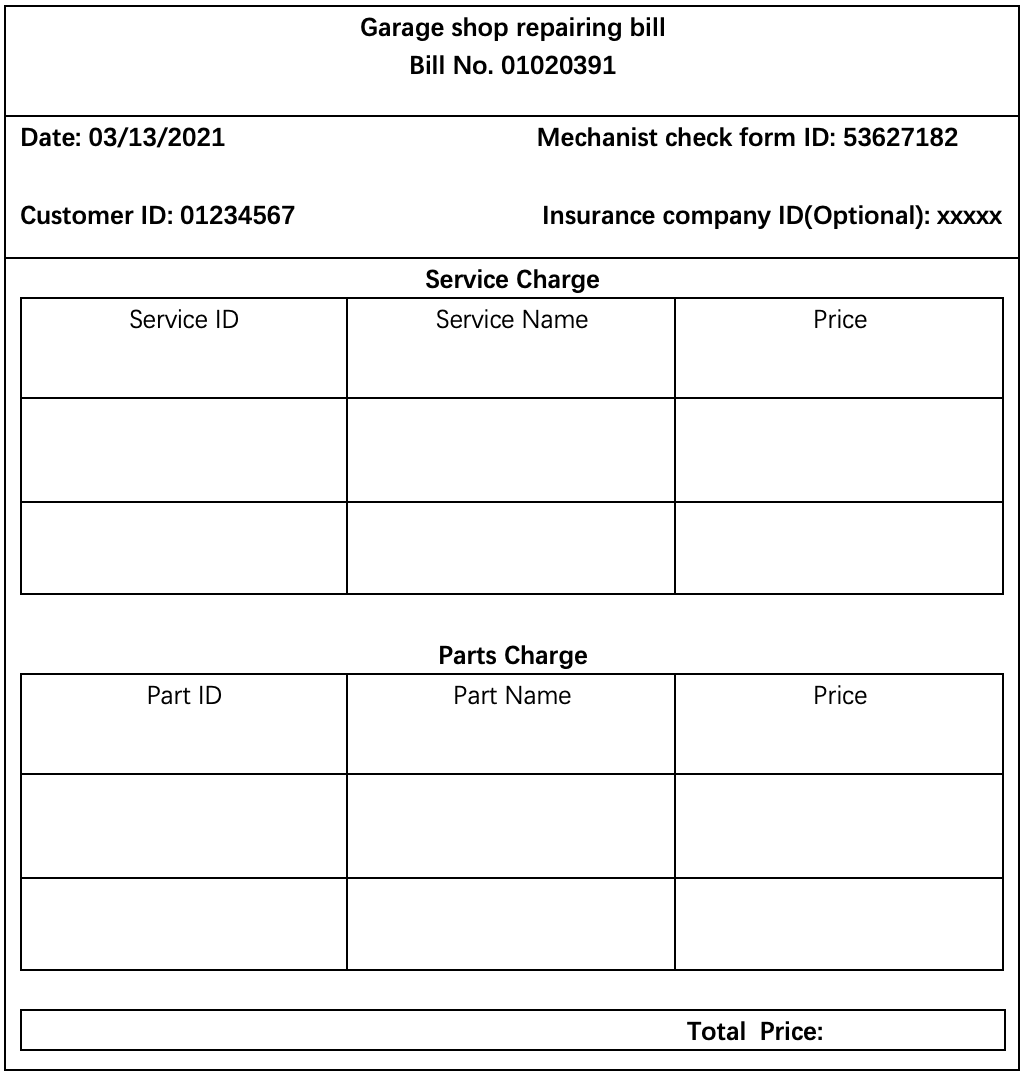
This is a garage shop that provides customers with various repairing services. According to the shop’s policy, customers will need to fill out the registration form if this is their first time coming to this shop. **(See the form below)**



After the customer has provided his name, address, email and phone number, the receptionist will add the customer into the system and the customer will receive a unique customer ID. Then the customer has registered as a member of the garage shop successfully. Carry on, the receptionist will give the customer a mechanist check form and assist the customer to fill out specific information regarding the vehicle, such as the vehicle type, vehicle registration number, vehicle make, vehicle color, model, year and insurance company id (if customer would like to provide one). The system will generate a certain form id for the check form, and it will be assigned to a certain workshop with different workshop number as well. And the receptionist will hand over the form to mechanists. Mechanists will check the vehicle first, then state the issue. One vehicle might be checked and repaired by several mechanists depends on certain situation. Mechanists will let the customer know about the service that is required which includes service id and service name. Finally, mechanists will tell the customer about the date that their vehicle can be ready to pick up. **(See the form below)**



Last step comes to the bill, after the customer has confirmed with the mechanist and made an agreement on the check form, then the garage shop will provide the customer with a bill. Each bill has its own number, so customer can refer back to the bill if there is any issue during the process. Also, it includes the date, customer id, mechanist check form id and insurance company id (if was provided by the customer). The service charge and the parts charge are listed clearly with service id, service name, service price, part id, part name and part price. If the customer has any problem with the bill, he can contact the receptionist and he just need to provide the number of his bill so that the receptionist will be able to assist the customer immediately. **（See the form below)**



1. **Database Functions**

This database of garage shop could provide basic index function for customers and shop. The database will record each customer's identity information such as name, phone number, registration time, insurance company, and the type of service selected, so that it is convenient for the repair shop and customer to check the repair record.

For registration information, each customer is given a digital ID that is unique and matches his or her identity to ensure that no one can steal the information.

The system will also input the ID of each employee, so that the customer only needs to input the ID of the employee who repairs the car for himself to know the solution to the vehicle failure.

The database also records the details of each vehicle and any faults that occur so that employees can find them easily.

The database will also add an ID to each part, so that when settling the price, the garage shop can directly input the ID of the part to query the price of the part.

1. **Data Requirements**

**Customer** refers to customer. It is uniquely identified by customer ID.

Customer attributes: customer\_id, first\_name, last\_name, phone, registration\_date, email, address, insurance, vehicle\_id, registration\_number,

**Vehicle** refers to existing vehicles. It is uniquely identified by vehicle id.

Vehicle:vehicle\_id, make, model, year, colour

**Service type** refers to what services the shop can provide. E.g., logbook service, air-condition repair.

Service type: service\_id, service\_name, price

**Staff refers** to employees working in this garage shop.

Staff: staff\_id, first\_name, last\_name, phone, employed\_date, email, address, department\_id

**Mechanist check form** refers to a specific visit for specific customer contains information about customer name, vehicle information, and reason come to garage shop.

Mechanist check form: form\_id, customer\_id, vehicle\_id, staff\_id(recipetionist), problem\_of\_the\_vehicle

**Workshop** refers to differernt workshop in this garage shop.

Workshop: workshop\_id, workshop\_name, workshop\_manager(staff\_id)

**Insurance company** refers to customer’s insurance.

Insurance company: company\_id, campany\_name, company\_address, company\_phone, company\_representative

**Part** refers to a part, it uniquely identified by part\_id.

Part: part\_id, part\_name, price

**Service schedule** refers to a single schedule.

Service schedule: schedule\_id, form\_id, customer\_id, vehicle\_id, workshop\_id

**Bill** refers to the bill for individual mechanist form.

Bill: service\_id, price, part\_id, price, form\_id, customer\_id, vehicle\_id.

**Skill** refers to services mechanist may provide.

Skill: skill\_id, service\_id

1. **Business Rules**
2. A customer is added to the system when first time come to shop.
3. Customer can provide their insurance company.
4. A customer can have multiple vehicles.
5. A customer is uniquely identified by their customer ID.
6. A customer will have different service forms for each visit.
7. Customer needs to provide personal detail and vehicle’s information (make, mode, registration number).
8. A receptionist is responsible for greeting the customer and help the customer to fill out the service form.
9. One mechanist might be able to provide one or more type of service.
10. Customer’s vehicle will be serviced by one or more mechanist at each visit.
11. The receptionist will arrange certain mechanist to check the vehicle.
12. Vehicle needs to be allocated to a workshop.
13. Vehicle might need replacement for certain parts.
14. Each part is assigned to a unique id and acquired with detailed information to match different vehicles.

# **Database Design Assignment Part B**

1. **Overview of the case study in at most five sentences (From part A)**

For this garage shop database system, customer needs to provide their personal information so that the receptionist can assist him to join the membership. With the help of different department employees, customer’s vehicle can be allocated to certain mechanists. After a detailed check, the system will generate a form ID for the vehicle. Within this form, issues of the vehicle and which type of service that it is required are listed clearly. At the end, the shop will provide the customer an official bill, if the customer has any problem, he can contact the receptionist and get helped quickly.

1. **Revised Business rules and Assumptions**
2. Customer will be added to the system at their first arrival.
3. One customer can have multiple vehicles.
4. Customers are uniquely identified by their customer ID.
5. Customers will have different service forms for each visit.
6. Customers need to provide their personal detail and vehicle’s information (make, mode, registration number).
7. A receptionist is responsible for greeting customers and help them to fill out the service form.
8. One mechanist might be able to provide one or more type of service.
9. Customer’s vehicle will be serviced by one or more mechanist at each visit.
10. The receptionist will arrange certain mechanist to check the vehicle.
11. Vehicle might need replacement for certain parts.
12. Each part is assigned to a unique id and acquired with detailed information to match different vehicles.
13. **ERD**

图示

描述已自动生成

1. **Justifications of the ERD based on the business rules and/or assumptions.**

**Business rules related to entity Customers:**

BR1: Every customer needs to register in the system.

BR2: Customers need to provide the detail information (e.g., name, registration date, phone number, Email, street, etc.).

BR3: The system will generate a unique ID for each customer.

**The PK of CUSTOMER is:** cusID

**Business rules related to entity Vehicle:**

BR4: According to different models, colors and the year of the vehicle, there will be different ID numbers.

**The PK of Vehicle is:** VehicleID

**Business rules related to entity Part:**

BR5: Each part is recorded separately in the database.

BR6: The information in the database contains the name and price of the part.

**The PK of Part is:** PartID

**Business rules related to the relationship between Part and Vehicle and its cardinalities:**

BR7: One applicative part might be applied to several vehicles. (cardinality 1 to many)

BR8: Each vehicle might need several applicative parts. (cardinality 0 to many)

**Note:** based on BR7 & BR8 there is **many to many relationship** between Vehicle and Part that is converted to the associative entity (Applicative\_Part).

**The PK of Applicative\_Part is:** PartID, VehicleID

**The FK of Applicative\_Part is:** PartID(Part), VehicleID(Vehicle)

**Business rules related to entity Employee:**

BR9: Information about each employee needs to be recorded in the system.

BR10: The system will generate a unique ID for each employee.

BR11: Employees need to provide the personal information (name, DOB, phone number, Email, date employed, address, annual salary, etc.).

BR12: Employees are assigned to a certain type of job (e.g., mechanist or receptionist).

**The PK of Employee is:** empID

**Assumption**: 1. There are many types of job such as accountant, mechanist and receptionist. In this ERD we are only going to list mechanist and receptionist. 2. Mechanists cannot be receptionists, and similarly, receptionists cannot be mechanists.

**The PK of Mechanist is**: MechanistID(empID)

**The PK of Receptionist is:** ReceptionistID(empID)

**Business rules related to entity Service:**

BR13: Each service is assigned individually to the system.

BR14: Each service has different names and various prices.

**The PK of Service is:** serviceID

**Business rules related to the relationship between Mechanist and Service and its cardinalities:**

BR15: Each mechanist can provide at least one or more services. (cardinalities 1 to many)

BR16: Services can be applied by at least one or mechanists. (cardinalities 1 to many)

**Note:** based on BR15 & BR16 there is **many to many relationship** between Mechanist and Service that is converted to the associative entity (Mechanist\_Service).

**The PK of Mechanist\_Service is:** serviceID, MechanistID

**The FK of Mechanist\_Service is:** serviceID(Service), MechanistID(Mehcanist)

**Business rules related to entity MechanistServiceForm:**

BR17: The form can be identified under customer’s ID.

BR18: The form will record the admission date of the vehicle, vehicle registration number, pick-up date and vehicle ID.

BR19: A receptionist is responsible for greeting customers and help them to fill out the service form.

**The PK of MechanistServiceForm is:** cusID, formID

**The FK of MechanistServiceForm is**: ReceptionistID(Receptionist), VehicleID(Vehicle)

**Business rules related to relationship between Service and MechanistServiceForm and its cardinalities:**

BR20: Each form can have zero or several services. (cardinalities 0 to many)

BR21: Each service can be applied to zero or several vehicles. (cardinalities 0 to many)

**Note:** based on BR20 & BR21 there is **many to many relationship** between MechanistServiceForm and Service that is converted to the associative entity (Service\_Provided).

**The PK of Service\_Provided is:** serviceID, cusID, formID

**The FK of Service\_Provided is:** serviceID(Service), cusID(MechanistServiceForm), formID(MechanistServiceForm)

**Business rules related to relationship between Mechanist and MechanistServiceForm and its cardinalities:**

BR22: Each vehicle might be checked by at least one or more mechanists. (cardinalities 1 to many)

BR23: Each mechanist might check zero or more vehicles. (cardinalities 0 to many)

BR24: Each mechanist will leave a description about the issues.

**Note:** based on BR22 & BR23 there is **many to many relationship** between MechanistServiceForm and Mechanist that is converted to the associative entity (MechanistCheck).

**The PK of MechanistCheck is:** empID, cusID, formID

**The FK of MechanistCheck is:** empID(Employee), cusID(MechanistServiceForm), formID(MechanistServiceForm)

**Business rules related to relationship between Part and MechanistServiceForm and its cardinalities:**

BR25: Each vehicle might need to replace zero or more parts. (cardinalities 0 to many)

BR26: Each part might be provided to zero or many vehicles (cardinalities 0 to many)

BR27: The system will record the quantity of a part replaced.

**Note:** based on BR25 & BR26 there is **many to many relationship** between MechanistServiceForm and Part that is converted to the associative entity (Part\_Replaced).

**The PK of Part\_Replaced is:** PartID, cusID, formID

**The FK of Part\_Replaced is:** PartID(Part), cusID(MechanistServiceForm), formID(MechanistServiceForm)

**Business rules related to entity Bill:**

BR28: Each service form will have an individual bill.

BR29: Each bill will record the date of the bill.

**The PK of Bill is**: BillNo

**The FK of Bill is**: cusID(MechanistServiceForm), formID(MechanistServiceForm)

# **Database Design Assignment Part C**

## **Revised Business rules and assumptions (From part B)**

BR1: Every customer needs to register in the system.

BR2: Customers need to provide the detail information (e.g., name, registration date, phone number, Email, street, etc.).

BR3: The system will generate a unique ID for each customer.

BR4: According to different models, colors and the year of the vehicle, there will be different ID numbers.

BR5: Each part is recorded separately in the database.

BR6: The information in the database contains the name and price of the part.

BR7: One applicative part might be applied to several vehicles. (cardinality 1 to many)

BR8: Each vehicle might need several applicative parts. (cardinality 0 to many)

BR9: Information about each employee needs to be recorded in the system.

BR10: The system will generate a unique ID for each employee.

BR11: Employees need to provide the personal information (name, DOB, phone number, Email, date employed, address, annual salary, etc.).

BR12: Each employee may have a manager.

BR13: Each employee can manage zero or many employees.

BR14: Employees are assigned to a certain type of job (e.g., mechanist or receptionist).

**Assumption**: 1. There are more subtypes than just accountant, mechanist and receptionist etc. In this ERD we are only going to list mechanist and receptionist. 2. Mechanists cannot be receptionists, and similarly, receptionists cannot be mechanists.

BR15: Each service is assigned individually to the system.

BR16: Each service has different names and various prices.

BR17: Each mechanist can provide at least one or more services. (cardinalities 1 to many)

BR18: Services can be applied by at least one or mechanists. (cardinalities 1 to many)

BR19: The booking depends on each customer, each booking will have a booing ID.

BR20: the admission date of the booking, vehicle registration number, pick-up date and vehicle ID will be recorded at booking.

BR21: A receptionist is responsible for greeting customers and help them to get the booking information.

BR22: Each booking can have zero or several services. (cardinalities 0 to many)

BR23: Each service can be applied to zero or several vehicles. (cardinalities 0 to many)

BR24: Each vehicle might be checked by at least one or more mechanists. (cardinalities 1 to many)

BR25: Each mechanist might check zero or more vehicles. (cardinalities 0 to many)

BR26: Each mechanist will leave a description about the issues.

BR27: Each vehicle might need to replace zero or more parts. (cardinalities 0 to many)

BR28: Each part might be provided to zero or many vehicles (cardinalities 0 to many)

BR29: The system will record the quantity of a part replaced.

BR30: A bill is issued to every Booking.

BR31: Each bill will record the date of the bill.

## **Revised ERD of Part B**

图示, 示意图

描述已自动生成

## **Relations**

Part (PartID, name, price)

Vehicle (VehicleID, make, model, year, colour)

Applicative\_Part (PartID\*, VehicleID\*)

FK (PartID) References Part

FK (vehicleID) references Vehicle

Booking (BookingID, cusID\*, AdmissionDate, Vehicle\_Reg\_No, Pick-up\_Date, ReceptionistID\*, VehicleID\*)

FK (ReceptionistID) references Employee

FK (VehicleID) references Vehicle

FK (cusID) references Customer

Part\_Replaced ((cusID, BookingID)\*, PartID\*, quantity)

FK (PartID) references Part

FK (cusID, BookingID) references Booking

Service (serviceID, name, price)

Service\_Provided (service\_ID\*, (cusID, BookingID)\*)

FK (cusID, BookingID) references Booking

FK (ServiceID) references Service

Single relation for Employee:

Employee (empID, emp\_F\_name, emp\_L\_name, phone, empDate, Street, City, State, Zip\_Code, annual\_salary, empType, managerID\*)

FK (managerID) references Employee

Clarification: the subtypes related to Employee do not have any attribute. Therefore, we used single relation for Employee.

MechanistCheck (MechanistID\*, (cusID, BookingID)\* , description)

FK (cusID, BookingID) references Booking

FK (MechanistID) references Employee

Mechanist\_Service (serviceID\*, empID\*)

FK (Service\_ID) references Service

FK (MechanistID) references Employee

Customer (cusID, cus\_F\_name, cus\_L\_name, phone, registrationDate, Email. Street, City, State, Zip\_Code)

Bill (BillNo, BillDate, (cusID, BookingID)\*)

FK (cusID, BookingID) references Booking

## **List of Functional Dependencies Related to Each Business Rules.**

**BR(s):**

BR1: Every customer needs to register in the system.

BR2: Customers need to provide the detail information (e.g., name, registration date, phone number, Email, street, etc.).

BR3: The system will generate a unique ID for each customer.

**FD1 is determined based on BR1, BR2 and BR3:**

**FD1: Customer\_ID → Customer\_F\_name, Customer\_L\_name, Phone, Registration\_Date, Email, Street, City, State, Zip\_Code**

**BR(s):**

BR4: According to different models, colors and the year of the vehicle, there will be different ID numbers.

**FD2 is determined based on BR4:**

**FD2: Vehicle\_ID → Make, Model, Year, Colour**

**BR(s):**

BR5: Each part is recorded separately in the database.

BR6: The information in the database contains the name and price of the part.

**FD3 is determined based on BR5 and BR6:**

**FD3: Part\_ID → Name, Price**

**BR(s):**

BR9: Information about each employee needs to be recorded in the system.

BR10: The system will generate a unique ID for each employee.

BR11: Employees need to provide the personal information (name, DOB, phone number, Email, date employed, address, annual salary, etc.).

BR12: Each employee may have a manager.

BR13: Each employee can manage zero or many employees.

BR14: Employees are assigned to a certain type of job (e.g., mechanist or receptionist).

**FD4 is determined based on BR9, BR10, BR11, BR12 and BR14:**

**FD4: Employee\_ID → Employee\_F\_name, Employee\_L\_name, phone, employee\_Date, Street, City, State, Zip\_Code, Annual\_salary, Manager\_ID, Employee\_Type**

**FD5 is determined based on BR 13:**

**FD5: Manager\_ID → Employee\_ID**

**BR(s):**

BR15: Each service is assigned individually to the system.

BR16: Each service has different names and various prices.

**FD6 is determined based on BR15 and BR 16:**

**FD6: Service\_ID → Name, Price**

**BR(s):**

BR19: The booking depends on each customer, each booking will have a booing ID.

BR20: The admission date of the booking, vehicle registration number, pick-up date and vehicle ID will be recorded at booking.

BR21: A receptionist is responsible for greeting customers and help them to get the booking information.

**FD7 is determined based on BR19, BR20 and BR21:**

**FD7: CustomerID, BookingID → AdmissionDate, Vehicle\_Registration\_Number, Pick-up\_Date, Receptionist\_ID, Vehicle\_ID, ReceptionistID**

**BR(s):**

BR24: Each vehicle might be checked by at least one or more mechanists. (cardinalities 1 to many)

BR25: Each mechanist might check zero or more vehicles. (cardinalities 0 to many)

BR26: Each mechanist will leave a description about the issues.

**FD8 is determined based on BR24, BR25 and BR26:**

**FD8: MechanistID, CustomerID, BookingID → Description**

**BR(s):**

BR27: Each vehicle (booking) might need to replace zero or more parts. (cardinalities 0 to many)

BR28: Each part might be provided to zero or many vehicles (bookings) (cardinalities 0 to many)

BR29: The system will record the quantity of a part replaced.

**FD9 is determined based on BR27, BR28 and BR29:**

**FD9 : PartID, BookingID, CustomerID → Quantity**

**BR(s):**

BR30: A bill is issued to every Booking.

BR31: Each bill will record the date of the bill.

**FD10 is determined based on BR30 and BR31:**

**FD10: BillNO → BillDate, CustomerID, BookingID**

## **Normalization**

**Relation**: Part (PartID, name, price)

**FD**: Part → PartID, name, price

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and no partial dependencies on the key PartID (single attribute key).

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation**: Vehicle (VehicleID, make, model, year, colour)

**FD**: VehicleID → make, model, year, colour

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and no partial dependencies on the key VehicleID (single attribute key).

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation**: Applicative\_Part (PartID\*, VehicleID\*)

FK (PartID) references Part

FK (VehicleID) references Vehicle

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and because it has no non-key attribute there is no partial dependencies.

3NF – it is in 2NF and because it has no non-key attribute there is no interdependencies between non-key attributes.

**Relation**: Booking (BookingID, cusID\*, AdmissionDate, Vehicle\_Reg\_No, Pick-up\_Date, ReceptionistID\*, VehicleID\*)

FK (ReceptionistID) references Employee

FK (VehicleID) references Vehicle

**FD:** CustomerID, BookingID → AdmissionDate, Vehicle\_Registration\_Number, Pick-up\_Date, Receptionist\_ID, Vehicle\_ID.

VehicleID → make, model, year, colour

Customer\_ID → Customer\_F\_name, Customer\_L\_name, Phone, Registration\_Date, Email, Street, City, State, Zip\_Code

Employee\_ID → Employee\_F\_name, Employee\_L\_name, phone, employee\_Date, Street, City, State, Zip\_Code, Annual\_salary, Manager\_ID, Employee\_Type, managerID

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and all its non-key attributes are dependent on all parts of the key (composite Key). It means there is no partial functional dependency.

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation:** Part\_Replaced ((cusID, BookingID)\*, PartID\*, quantity)

FK (cusID, BookingID) reference Booking

FK (PartID) reference Part

**FD:** PartID, BookingID, CustomerID → Quantity

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and all of its non-key attributes are dependent on all parts of the key (composite Key). It means there is no partial functional dependency.

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation**: Service (serviceID, name, price)

**FD:** Service\_ID → Name, Price

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and no partial dependencies on the key ServiceID (single attribute key).

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation:** Employee (empID, emp\_F\_name, emp\_L\_name, phone, empDate, Street, City, State, Zip\_Code, annual\_salary, empType, managerID\*)

FK (managerID) references Employee

**FD:**Employee\_ID → Employee\_F\_name, Employee\_L\_name, phone, employee\_Date, Street, City, State, Zip\_Code, Annual\_salary, Manager\_ID, Employee\_Type

Manager\_ID → Employee\_ID

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and no partial dependencies on the key empID(single attribute key).

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation**: MechanistCheck (MechanistID\*, (cusID, BookingID)\* , description,)

FK (MechanistID) references Employee

FK (cusID, BookingID) references Booking

**FD:** MechanistID, CustomerID, BookingID → Description

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and all of its non-key attributes are dependent on all parts of the key (composite Key). It means there is no partial functional dependency.

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation:**Mechanist\_Service (serviceID\*, empID\*)

FK (serviceID) references Service

FK (empID) references Employee

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and because it has no non-key attribute there is no partial dependencies.

3NF – it is in 2NF and because it has no non-key attribute there is no interdependencies between non-key attributes.

**Relation:** Customer (cusID, cus\_F\_name, cus\_L\_name, phone, registrationDate, Email. Street, City, State, Zip\_Code)

**FD:** Customer\_ID→ Customer\_F\_name, Customer\_L\_name, Phone, Registration\_Date, Email, Street, City, State, Zip\_Code

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and no partial dependencies on the key cusID(single attribute key).

3NF – it is in 2NF and no interdependencies between non-key attributes.

**Relation:**Bill (BillNo, BillDate, (cusID, BookingID)\*)

FK (cusID, BookingID) references Booking

**FD:** BillNO → BillDate, CustomerID, BookingID

1NF – attributes are atomic and there is not derived attribute.

2NF – It is in 1NF and all of its non-key attributes are dependent on all parts of the key (composite Key). It means there is no partial functional dependency.

3NF – it is in 2NF and no interdependencies between non-key attributes.

# **Database Design Assignment Part D**

**Notes:**

* Please submit Part D as a separate **.txt or .sql file** on both Canvas and Ed (in Part D module).
* **DO NOT** submit a Microsoft word document. The only accepted file extension is txt, or sql.
* Please review the instruction and watch the uploaded video on Ed (in Part D module).
* Please see the provided example file “dbpizza\_Revised.txt”, or in the assignment sample:

## **D.1. Created Database**

1. You should download the dbName.sql file as follows and **submit** it in .**txt** or .**sql** format on both Canvas and Ed.

