

# **Introduction**

Based on case study, we examine the development of a Maintenance Management System for a small computer repair shop. The system aims to efficiently track and manage repair jobs, including details about the computers being repaired, the items used, labor costs, and the repairmen involved. The shop's operations are streamlined by accurately documenting customer interactions, deposits, and payments, ensuring a smooth workflow from the moment a computer is brought in for repair to its eventual return to the customer. This relational database structure provides a robust framework to support the shop's goal of delivering high-quality repair services while maintaining meticulous records.

## **Statement of Current System Problem**

The current system at the small computer repair shop lacks an integrated approach to managing and tracking repair jobs, resulting in inefficiencies and potential errors. Key challenges include:

1. **Manual Tracking:** The shop relies on manual methods to record repair details, labor costs, and items used, leading to inaccuracies and difficulty in accessing information.
2. **Customer Management:** There is no centralized system for managing customer information, deposits, and payment records, making it hard to provide seamless customer service and follow-up.
3. **Inventory Management:** The tracking of items used in repairs is inefficient, causing issues with inventory levels and reordering processes, which can delay repair jobs.
4. **Labor and Cost Allocation:** Labor costs and repair job details are not systematically recorded, making it challenging to accurately calculate total repair costs and track the work done by repairmen.
5. **Communication and Coordination:** The lack of a unified system hampers effective communication and coordination among repairmen, leading to potential delays and misunderstandings.

## **Objectives of the project**

## **General Objective**

To develop a comprehensive Maintenance Management System for a small computer repair shop that streamlines and automates the tracking and management of repair jobs, inventory, labor costs, and customer interactions, thereby improving operational efficiency, accuracy, and customer satisfaction.

## **Specific Objectives**

### **1. Automate Repair Job Tracking**

- Develop a system to automatically record and track details of each repair job, including dates received, started, completed, and returned, as well as the repair details, labor details, and costs.

### **2. Centralize Customer Management**

- Create a centralized database for managing customer information, including contact details, deposit records, and payment histories, to ensure seamless customer service and follow-up.

### **3. Optimize Inventory Management**

- Implement an inventory management system to track items used in repairs, monitor stock levels, and automate reordering processes when stock levels fall below a predefined threshold.

### **4. Accurate Labor Cost Allocation**

- Design a system to accurately record labor costs and allocate them to specific repair jobs, ensuring precise calculation of total repair costs and enabling effective tracking of repairmen's work.

### **5. Enhance Communication and Coordination**

- Facilitate effective communication and coordination among repairmen by providing a unified platform where repair job details, labor allocations, and inventory usage are easily accessible.

### **6. Improve Customer Payment and Receipt Process**

- Streamline the process for customer payments, ensuring that customers receive accurate receipts and can easily uplift their repaired computers upon presenting payment receipts.

## **7. Generate Comprehensive Reports**

- Develop reporting capabilities to generate comprehensive reports on repair jobs, labor costs, inventory levels, and customer interactions, providing valuable insights for decision-making and business improvement.

## **8. Ensure Data Security and Integrity**

- Implement robust security measures to protect customer data, repair job details, and inventory records, ensuring data integrity and compliance with relevant regulations.

## **Scope of the Project**

This project aims to design and implement a Maintenance Management System for a small computer repair shop. The system will automate repair job tracking, customer management, inventory control, labor cost allocation, and payment processing. It will provide centralized data storage, facilitate efficient communication among staff, and generate comprehensive reports. The system will ensure data security and integrity, and offer user training and support to enhance overall operational efficiency and customer satisfaction. Advanced analytics and multi-location integration are not within the project's scope.

## **Functional Requirements**

### **1. Repair Job Management**

- Allow users to create, update, and delete repair job records.
- Record and track dates associated with repair jobs (received, started, completed, returned).
- Store detailed information about repair jobs, including repair and labor details, costs, and additional comments.

### **2. Customer Management**

- Create, update, and manage customer records with contact details and interaction history.
- Link each customer to their respective computer and repair job records.
- Manage customer deposits and payment records.

### **3. Labor Cost Allocation**

- Record labor details and costs for each repair job.
- Allocate labor costs accurately to specific repair jobs.
- Track repairmen's work and assignment

#### **4. Reporting and Analytics**

- Generate comprehensive reports on repair jobs, labor costs, inventory levels, and customer interactions.
- Provide insights and data visualization to support decision-making and business improvement.

#### **5. User Access and Security**

- Implement user authentication and access control to ensure data security.
- Protect customer data and repair job details with robust security measures.
- Ensure data integrity and compliance with relevant regulations.

### **Database design**

#### **Entity Types with Their Attributes:**

##### **1. RepairJobs**

- **JobNum:** A unique identifier for each repair job.
- **DateReceived:** The date when the computer was brought in for repair.
- **DateToReturn:** The expected date for the customer to return and pick up the computer.
- **DateReturned:** The actual date when the computer was collected by the customer.
- **DateStarted:** The date when the repair work started.
- **DateEnded:** The date when the repair work was completed.
- **RepairDetails:** Detailed description of the repairs performed.
- **LaborDetails:** Detailed description of the labor involved in the repair job.
- **LaborCost:** The cost associated with the labor for the repair job.
- **TotalCost:** The total cost of the repair job, including labor and items used.
- **PaidInFull:** A flag indicating whether the repair job has been paid in full.

- **AdditionalComments:** Any additional comments or notes about the repair job.

## 2. Computers

- **ComputerId:** A unique identifier for each computer.
- **SerialNum:** The serial number of the computer.
- **Make:** The manufacturer of the computer.
- **Model:** The model of the computer.
- **ComputerDescription:** A description of the computer.

## 3. Items

- **ItemId:** A unique identifier for each item.
- **PartNum:** The part number of the item.
- **ShortName:** A short name or abbreviation for the item.
- **ItemDescription:** A detailed description of the item.
- **Cost:** The cost of the item.
- **NumInStock:** The number of items currently in stock.
- **ReorderLow:** The stock level at which the item should be reordered.

## 4. Repairmen

- **RepairmenId:** A unique identifier for each repairman.
- **LastName:** The last name of the repairman.
- **FirstName:** The first name of the repairman.
- **MI:** The middle initial of the repairman.
- **Email:** The email address of the repairman.
- **Mobile:** The mobile phone number of the repairman.
- **HTel:** The home telephone number of the repairman.
- **Extension:** The extension number for contacting the repairman at work.

## 5. Vehicles

- **VehicleId:** A unique identifier for each vehicle.

- **VIN:** The Vehicle Identification Number.
- **RegistrationNo:** The registration number of the vehicle.
- **Year:** The year the vehicle was manufactured.
- **Make:** The manufacturer of the vehicle.
- **Model:** The model of the vehicle.
- **Color:** The color of the vehicle.

## **6.Customer**

- **CustomerId:** A unique identifier for each customer.
- **FirstName:** The first name of the customer.
- **LastName:** The last name of the customer.
- **Email:** The email address of the customer.
- **Phone:** The phone number of the customer.
- **Address:** The physical address of the customer.
- **DepositAmount:** The amount of money deposited by the customer when bringing in the computer for repair.
- **DateOfDeposit:** The date when the deposit was made by the customer.

# **Entity Relationships**

## **1.Customers and Computers**

- **Relationship:** One-to-One
- **Description:** Each customer can own one computer, and each computer is owned by one customer.

## **2.Customers and RepairJobs**

- **Relationship:** One-to-Many
- **Description:** Each customer can request multiple repair jobs, but each repair job is requested by only one customer.

## **3.Computers and RepairJobs**

- **Relationship:** One-to-One
- **Description:** Each computer can have one repair job at a time, and each repair job is associated with one computer.

#### **4. RepairJobs and Vehicles**

- **Relationship:** Many-to-One
- **Description:** Each repair job may involve one vehicle, and a vehicle can be used for multiple repair jobs.

#### **5. RepairJobs and Items (via RepairJobItems)**

- **Relationship:** Many-to-Many
- **Description:** Each repair job can use multiple items, and each item can be used in multiple repair jobs.

#### **6. RepairJobs and Repairmen (via RepairJobRepairmen)**

- **Relationship:** Many-to-Many
- **Description:** Each repair job can involve multiple repairmen, and each repairman can work on multiple repair jobs.

#### **7. Repairmen and Vehicles**

- **Relationship:** Many-to-One
- **Description:** Each repairman can be assigned one vehicle, and a vehicle can be assigned to multiple repairmen.

#### **Relationship phrase**

Customers **HAS** Computers

Customers **REQUEST** RepairJobs

Computers **UNDERGOES** RepairJobs

RepairJobs **INVOLVE** Vehicles

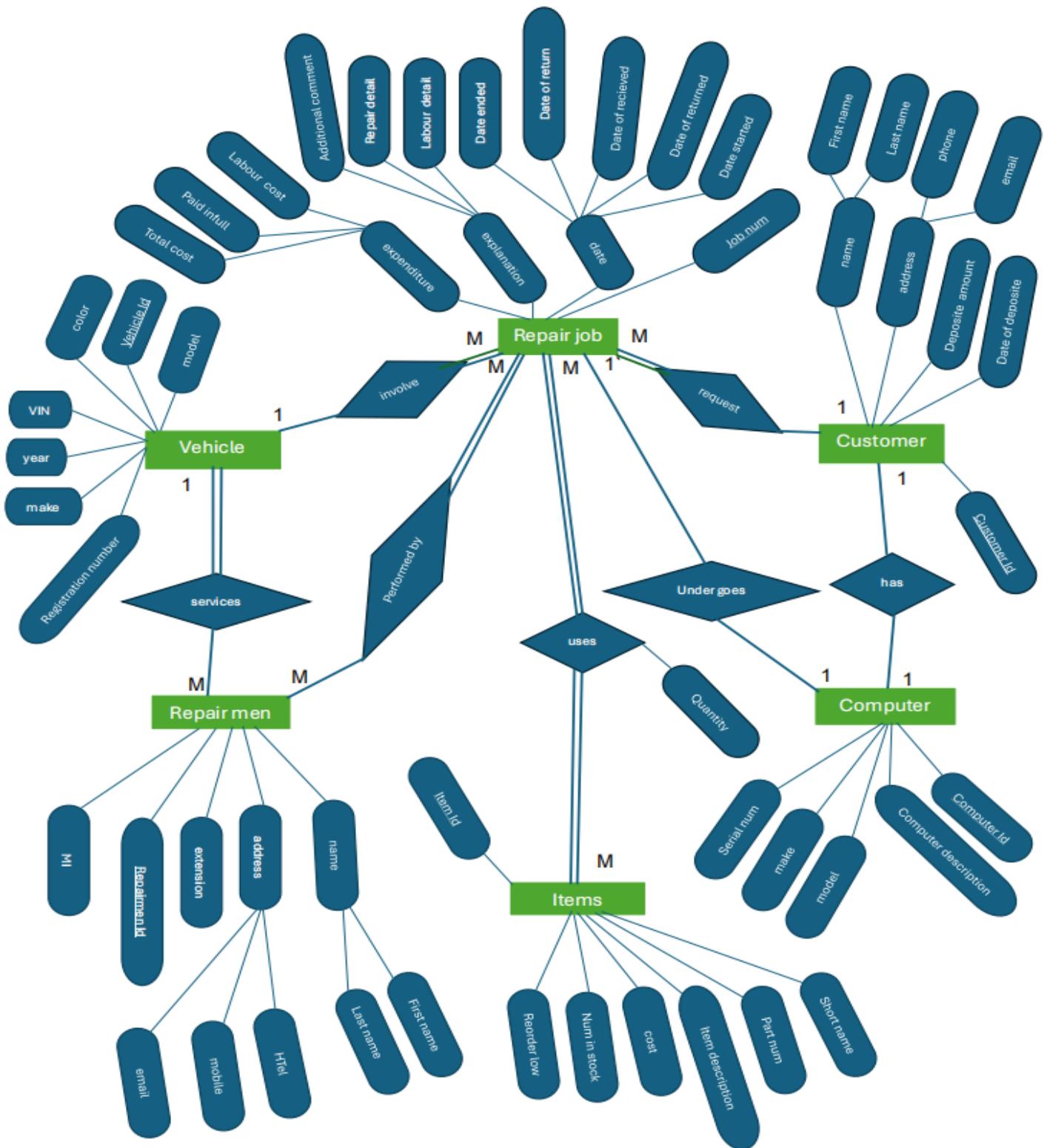
RepairJobs **USES** Items

RepairJobs **PERFORMED BY** Repairmen

Repairmen **SERVICES** Vehicles

#### **E-R diagram**





## normalization form

### First Normal Form (1NF) for the Maintenance Management System

To bring the database into First Normal Form (1NF), we ensure that:

1. All attributes contain atomic (indivisible) values.
2. Each column contains values of a single type.
3. Each table has a unique identifier (Primary Key).

#### For customer

CustomerID	First Name	Last name	email	phone	address	Deposit amount	Date of deposit
1	Behayi lu	Addis	baddis@example.com	0923121234	Addis	50.00	2025-01-01
2	Zelalem	Aguagna	z.m.smith@example.com	09555567	Shewa	75.00	2025-01-02
3	Aster	Girma	aster.girma@example.com	095559012	Gojam	100.00	2025-01-03

- ❖ Already all entities in 1NF since all attributes are atomic.

### Second Normal Form (2NF) for the Maintenance Management System

To achieve Second Normal Form (2NF), we ensure that:

1. The database is already in First Normal Form (1NF).
2. There are no partial dependencies, meaning all non-key attributes are fully dependent on the primary key.

## For repairjobItem

The Quantity field depends on the combination of JobNum and ItemID, so the table is already in 2NF.

RepairJobItemID	JobNum	ItemID	Quantity
1	1	1	1
2	2	2	1
3	3	3	1

- ❖ Already all entities in **2NF**.

## Third Normal Form (3NF) for the Maintenance Management System

To achieve **Third Normal Form (3NF)**, we ensure that:

1. The database is already in **Second Normal Form (2NF)**.
2. **There are no transitive dependencies**, meaning **no non-key attribute depends on another non-key attribute**.
3. Every non-key attribute must depend **only on the primary key**.

## Repairmen Table

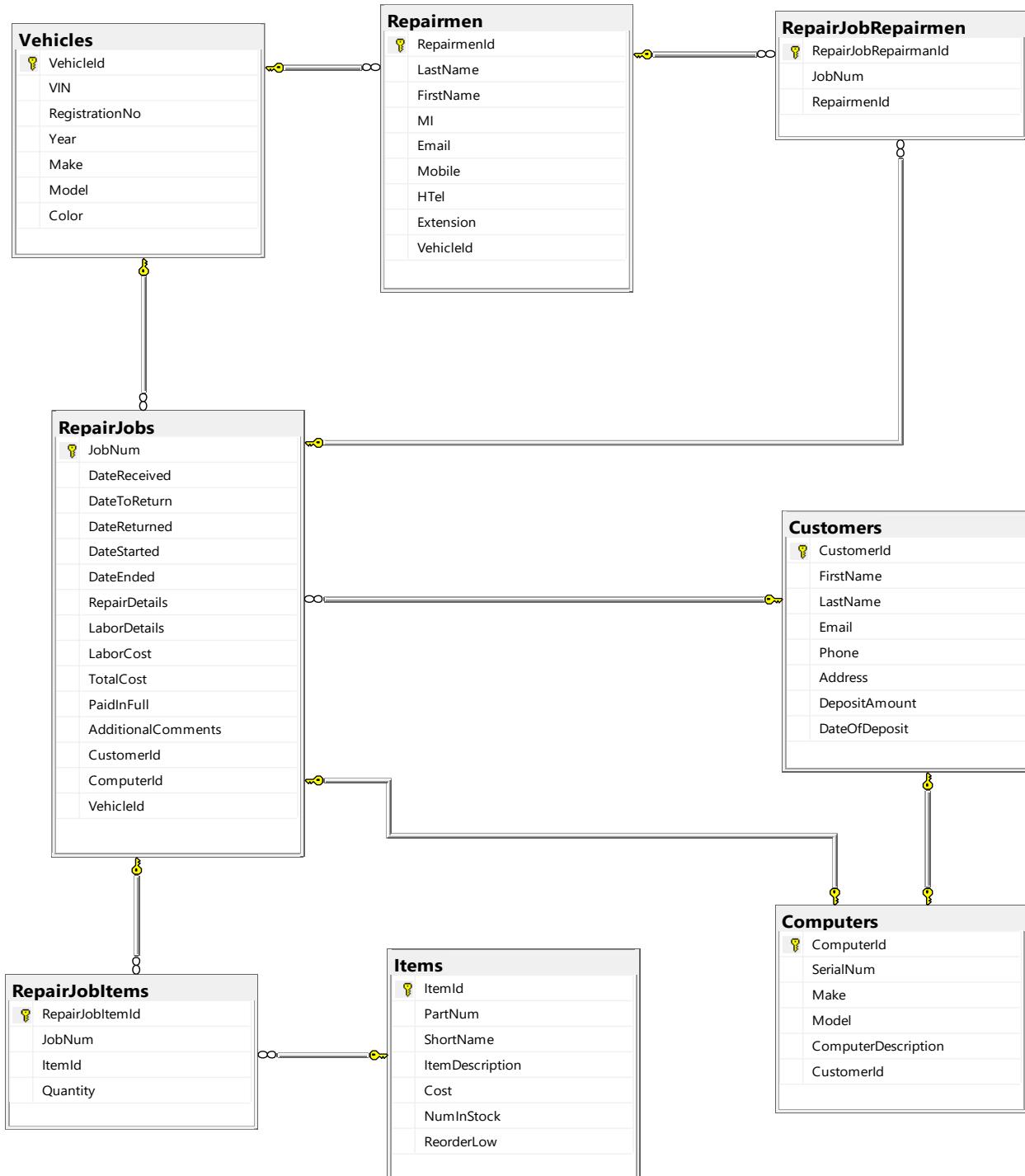
**No transitive dependencies.** Already in **3NF**

All other are also completely in 3NF

RepairmenID	firstname	lastname	M	email	mobile	height	extension	vehicleID
1	Million	Brhane	M	milli.bthen@example.com	095553456	555-7890	101	1

2	Hiwet	Desalegn	L	hiwe.white@example.com	055565 43	55 5- 32 10	102	2
3	Bereket	Teshome	K	brhe.black@example.com	555- 9876	55 5- 65 49	103	3

# Physical database design



## Implementation and testing

### SQL script for creating the database

```
1  CREATE DATABASE MAINTENANCE_MANAGEMENT_SYSTEM;
2 USE MAINTENANCE_MANAGEMENT_SYSTEM;
3
4 CREATE TABLE Customers (
5     CustomerId INT IDENTITY PRIMARY KEY,
6     FirstName VARCHAR(50) NOT NULL,
7     LastName VARCHAR(50) NOT NULL,
8     Email VARCHAR(100) UNIQUE,
9     Phone INT,
10    Address VARCHAR(200),
11    DepositAmount DECIMAL(10, 2),
12    DateOfDeposit DATE
13 );
14
15 CREATE TABLE Computers (
16     ComputerId INT IDENTITY PRIMARY KEY,
17     SerialNum VARCHAR(100) UNIQUE NOT NULL,
18     Make VARCHAR(50) NOT NULL,
19     Model VARCHAR(50) NOT NULL,
20     ComputerDescription TEXT,
21     CustomerId INT UNIQUE NOT NULL,
22     FOREIGN KEY (CustomerId) REFERENCES Customers(CustomerId)
23         ON DELETE CASCADE
24     ON UPDATE CASCADE
25
26 );
27
28 CREATE TABLE Vehicles (
29     VehicleId INT IDENTITY PRIMARY KEY,
30     VIN VARCHAR(100) UNIQUE NOT NULL,
31     RegistrationNo VARCHAR(100) UNIQUE NOT NULL,
32     Year INT NOT NULL,
33     Make VARCHAR(50) NOT NULL,
34     Model VARCHAR(50) NOT NULL,
35     Color VARCHAR(20)
36 );
37
38 CREATE TABLE Repairmen (
39     RepairmenId INT IDENTITY PRIMARY KEY,
40     LastName VARCHAR(50) NOT NULL,
41     FirstName VARCHAR(50) NOT NULL,
42     MI CHAR(1),
43     Email VARCHAR(100) UNIQUE,
44     Mobile VARCHAR(20),
45     HTel VARCHAR(20),
```

```

46     Extension VARCHAR(10),
47     VehicleId INT,
48     FOREIGN KEY (VehicleId) REFERENCES Vehicles(VehicleId)
49         ON DELETE CASCADE
50         ON UPDATE CASCADE
51
52 );
53
54 CREATE TABLE RepairJobs (
55     JobNum INT IDENTITY PRIMARY KEY,
56     DateReceived DATE NOT NULL,
57     DateToReturn DATE NOT NULL,
58     DateReturned DATE,
59     DateStarted DATE,
60     DateEnded DATE,
61     RepairDetails TEXT,
62     LaborDetails TEXT,
63     LaborCost DECIMAL(10, 2) NOT NULL,
64     TotalCost DECIMAL(10, 2) NOT NULL,
65     PaidInFull VARCHAR(50),
66     AdditionalComments TEXT,
67     CustomerId INT NOT NULL,
68     ComputerId INT UNIQUE NOT NULL,
69     VehicleId INT,
70     FOREIGN KEY (CustomerId) REFERENCES Customers(CustomerId),
71     FOREIGN KEY (ComputerId) REFERENCES Computers(ComputerId),
72     FOREIGN KEY (VehicleId) REFERENCES Vehicles(VehicleId)
73 );
74
75 CREATE TABLE Items (
76     ItemId INT IDENTITY PRIMARY KEY,
77     PartNum VARCHAR(100) UNIQUE NOT NULL,
78     ShortName VARCHAR(50),
79     ItemDescription TEXT,
80     Cost DECIMAL(10, 2) NOT NULL,
81     NumInStock INT,
82     ReorderLow INT
83 );
84
85 CREATE TABLE RepairJobItems (
86     RepairJobItemId INT PRIMARY KEY,
87     JobNum INT NOT NULL,
88     ItemId INT NOT NULL,
89     Quantity INT NOT NULL,
90     FOREIGN KEY (JobNum) REFERENCES RepairJobs(JobNum),
91     FOREIGN KEY (ItemId) REFERENCES Items(ItemId)
92         ON DELETE CASCADE
93         ON UPDATE CASCADE
94
95 );
96
97 CREATE TABLE RepairJobRepairmen (

```

```

98     RepairJobRepairmanId INT PRIMARY KEY,
99     JobNum INT NOT NULL,
100    RepairmenId INT NOT NULL,
101    FOREIGN KEY (JobNum) REFERENCES RepairJobs (JobNum),
102    FOREIGN KEY (RepairmenId) REFERENCES Repairmen (RepairmenId)
103      ON DELETE CASCADE
104      ON UPDATE CASCADE
105
106 );

```

## Inserted value

```

1 INSERT INTO Customers (FirstName, LastName, Email, Phone, Address,
2 DepositAmount, DateOfDeposit)
3 VALUES
4 ( 'behayilu', 'addis', 'baddis@example.com', '0923121234', 'addis',
5 50.00, '2025-01-01'),
6 ( 'zelalem', 'agugna', 'z.m.smith@example.com', '09555567', 'shewa',
7 75.00, '2025-01-02'),
8 ( 'Aster', 'girma', 'aster.girma@example.com', '095559012', 'gojam',
9 100.00, '2025-01-03');
10
11 INSERT INTO Computers ( SerialNum, Make, Model, ComputerDescription,
12 CustomerId)
13 VALUES
14 ( 'SN12345', 'Dell', 'XPS 13', '13-inch laptop', 1),
15 ( 'SN67890', 'Apple', 'MacBook Pro', '15-inch laptop', 2),
16 ( 'SN54321', 'HP', 'Envy', '17-inch laptop', 3);
17 INSERT INTO Vehicles (VIN, RegistrationNo, Year, Make, Model, Color)
18 VALUES
19 ('1HGCM82633A123456', 'ABC123', 2022, 'Toyota', 'Corolla', 'Blue'),
20 ('1HGCM82633A654321', 'DEF456', 2023, 'Ford', 'Fiesta', 'Red'),
21 ('1HGCM82633A789012', 'GHI789', 2021, 'Honda', 'Civic', 'Black');
22
23 INSERT INTO Repairmen ( FirstName, LastName, MI, Email, Mobile, HTel,
24 Extension, VehicleId)
25 VALUES
26 ('million', 'brhane', 'M', 'milli.bthen@example.com', '095553456', '555-
27 7890', '101', 1),
28 ( 'hiwet', 'desalegn', 'L', 'hiwe.white@example.com', '05556543', '555-
29 3210', '102', 2),
30 ( 'bereket', 'teshome', 'K', 'brhe.black@example.com', '555-9876', '555-
31 6549', '103', 3);
32
33
34 INSERT INTO RepairJobs (DateReceived, DateToReturn, DateReturned,
35 DateStarted, DateEnded, RepairDetails, LaborDetails, LaborCost,
36 TotalCost, PaidInFull, AdditionalComments, CustomerId, ComputerId,
37 VehicleId)
38 VALUES

```

```
39 ( '2025-01-01', '2025-01-10', '2025-01-10', '2025-01-02', '2025-01-09',
40 'Replaced hard drive', '2 hours', 200.00, 250.00, 345, 'None', 1, 1, 1),
41 ('2025-01-03', '2025-01-12', '2025-01-12', '2025-01-04', '2025-01-11',
42 'Fixed screen', '3 hours', 300.00, 375.00, 7885, 'None', 2, 2, 2),
43 ('2025-01-05', '2025-01-15', '2025-01-15', '2025-01-06', '2025-01-14',
44 'Replaced battery', '1 hour', 100.00, 150.00, 6575, 'None', 3, 3, 3);
45
    INSERT INTO Items ( PartNum, ShortName, ItemDescription, Cost,
    NumInStock, ReorderLow)
VALUES
( 'P001', 'HDD', 'Hard Drive', 50.00, 20, 5),
( 'P002', 'SCN', 'Screen', 75.00, 15, 3),
( 'P003', 'BAT', 'Battery', 25.00, 30, 10);
INSERT INTO RepairJobItems (RepairJobItemId, JobNum, ItemId, Quantity)
VALUES
(1, 1, 1, 1),
(2, 2, 2, 1),
(3, 3, 3, 1);
INSERT INTO RepairJobRepairmen (RepairJobRepairmanId, JobNum,
RepairmenId)
VALUES
(1, 1, 1),
(2, 2, 2),
(3, 3, 3);
```

SQLQuery1.sql - D...P-0GBVIVD\WY (64)\* X

```
select * from computers;
select * from customers;
```

•

75 % <

Results Messages

	ComputerId	SerialNum	Make	Model	ComputerDescription	CustomerId
1	1	SN12345	Dell	XPS 13	13-inch laptop	1
2	2	SN67890	Apple	MacBook Pro	15-inch laptop	2
3	3	SN54321	HP	Envy	17-inch laptop	3

	CustomerId	FirstName	LastName	Email	Phone	Address	DepositAmount	DateOfDeposit
1	1	behayilu	addis	baddis@example.com	923121234	addis	50.00	2025-01-01
2	2	zelalem	agugna	z.m.smith@example.com	9555567	shewa	75.00	2025-01-02
3	3	Aster	girma	aster.girma@example.com	95559012	gojam	100.00	2025-01-03