AC51049

Database Systems Development Coursework 1

Team No. 4

Company Name: *RePlast Creations*
Company Slogan: "*Where Sustainability Meets Style and comfort*"

Submitted By:

Ana Das (2622436)

Debdatta Bhattacharyya (2615762)

Anil Gupta (2617934)

Mohini Singh (2622916)

Akshaya Shiva Ganga (2615385)

Trang Linh To (2659519)

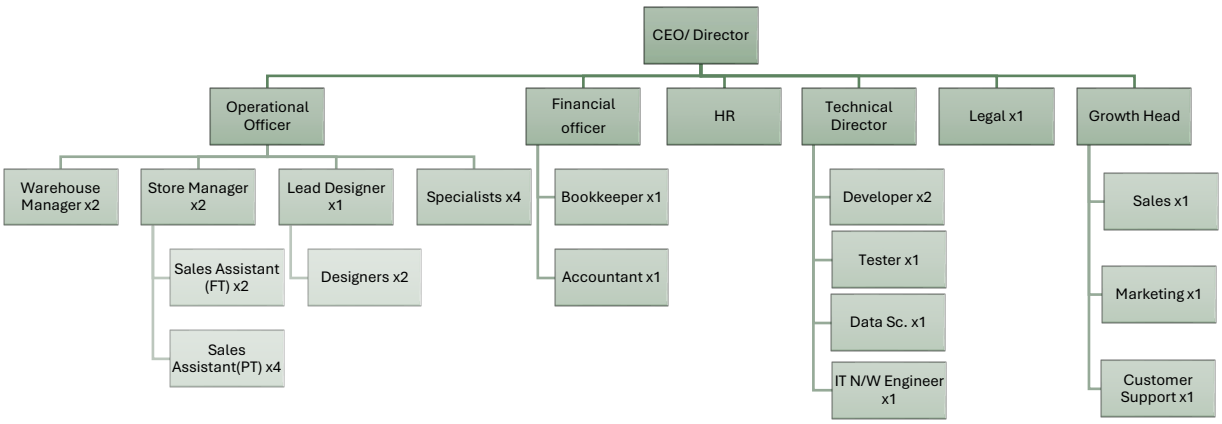
☑ We confirm that the team members have read and understood the University policy on Academic Misconduct.

Company Overview:

Replust Creations specializes in curating processed plastic waste and transforming these raw materials into sustainable items including ***small items like shelving units, and bedside tables, as well as medium-sized furniture such as desks and dining tables and outdoor furniture***. Currently, it is only ***sold at the 2 physical stores to retail customers and through in-person sales to the corporate clients*** and the company is looking to expand into more areas of England, Ireland, and Europe.

After the company CEO, Amada, founded Replast Creations, all records were maintained on paper. However, as the business is growing and with their expansion plan, this has become increasingly difficult to manage. Her goal is to implement a centralized database that consolidates all business-related information for their future endeavour of online presence. The CEO aims to utilise the system to oversee the whole business including office staff managing payroll and invoices, and branch staff managing sales and inventory—all through a single interface.

Headquartered ***at Edinburgh***, the company's Edinburgh Studio is the ***central manufacturing facility*** which ***receives parts and raw materials purchased from several different sustainable suppliers across the country***. The finished products are divided equally between ***two Warehouses or hubs; Edinburgh Hub and Birmingham Hub***. Each hub has one dedicated Warehouse Manager who is responsible for stock-keeping and updating the inventory details every day. ***Each hub extends to a showroom,*** where the products are showcased and customers visit for enquiry or shopping. Employees' shifts, attendance and holidays are tracked by a Third-Party App. Below is the Hierarchy of the company.



Four Types of users and their Roles:

1. Database Administrator (DBA): They manage user permissions, security, and data backups, and optimize performance for entities like Orders, Warranty Claim and Inventory. The DBA is responsible for maintaining the overall structure and ensuring that, all components are optimized for performance, data security, and integrity.

2. Database Designer/Architecture: The designer focuses on the logical structure of the database, ensuring the relationships and data flow between entities are efficient and meet business requirements**.** They focus on entity relationships, normalization, and scalability, ensuring data integrity in Product, Customer, and Supplier entities.

3. End Users: End users interact with the database through applications. Employees update records, Warehouse Managers update Inventory, Store Managers access reports and Customers place orders, all interacting with data through applications.

4. Application Programmers: Developers create and maintain the software applications that connect to the database, providing the interface through which end users interact with the data. Develop interfaces, implement business logic, and integrate APIs, facilitating operations like Order processing, Payment handling, and Warranty management.

Database Design Key Functional Requirements:

Store Management:

- Every facility centre at each location will be uniquely identified by an alphanumerical StoreID (EDI001, BRI005 etc,.).
- Each facility will have different opening and closing times depending on their FacilityType.

Employee Management:

- The company will assign a unique EmployeeID to each new employee which will be used to provide access and create email IDs.
- Each Employee will have a Line Manager assigned to them. Each Manager may or may not have multiple members working under.
- Each employee will be assigned to only one Store/ Facility Type. Hence, the company will assign a StoreID for the location.
- EmployementStatus is tracked to have information about current and previous employees. Employee has a sub-table that keeps records of their personal details.
- As the salary is provided in cash, the company decided not to have their bank details. However; this can be implemented in future along with the extension plan.

Product Management:

- Track detailed information on each product, including specifications, materials used, pricing, and inventory levels. Inventory levels should include stocks at different branches.
- Categorize products into relevant groups for easier management (e.g., Chairs, Tables, Benches).
- Each product will have a different warranty period.

Inventory and Material Tracking

- Maintain an accurate record of all raw materials, including their source, current stock, and replenishment needs.

Supplier Management

- Store and manage detailed information about suppliers, including contact details.
- The company decided not to track purchase order details for materials as it will still be done manually and will be managed by the Warehouse Manager. Currently, the company has a very limited supplier and each delivers one MaterialType. This data will eventually be stored in future and the data will be updated in the database.

Order and Sales Management

- Handle customer orders from placement to fulfilment, including order details, delivery dates, and payment status.
- Support multiple customer types (individuals, business Orgs) and pricing models.
- The transactions are processed through third-party payment gateway, therefore, no confidential payment details(Card no. CVV, Expiry Date etc,.) of the customer will be stored.
- The shipment booking and delivery is done via Third-party logistic providers, however; company keeps the basic details about the shipment as they plan to have their own transport system soon.

Warranty System:

- Each Product has a WarrentyPeriod (ex. 30 days, 60 days, 24 Months etc.)
- Company only storing the OrderDate and WarrentyPeriod ad and then manually check if the product is under warranty. We are also storing the WarrentyClaimStatus.
- Customers can not submit warranty claims via the e-commerce site instead an email will have to be sent out and communications will be done via emails only.

Customer Relationship Management (CRM)

- Store customer details including contact information and order history.
- Customers can create an account using their email address and set a unique password and again use them to log to the portal to browse. Each email ID will be assigned to one CustomerID. Hence, the system will not allow to register the same email ID again. CustomerID will strictly be numerical.
- No queries or complaints will be recorded or tracked. It will either be emailed to or on call.

- Future scope: Implementation of features to identify and prioritize high-value customers for loyalty programs. This service could be used by logging in to customer account.

Security and Access Control

- Ensure that sensitive customer and business data is protected using appropriate encryption and security measures.
- Implement role-based access control to restrict data visibility and modifications based on user roles.

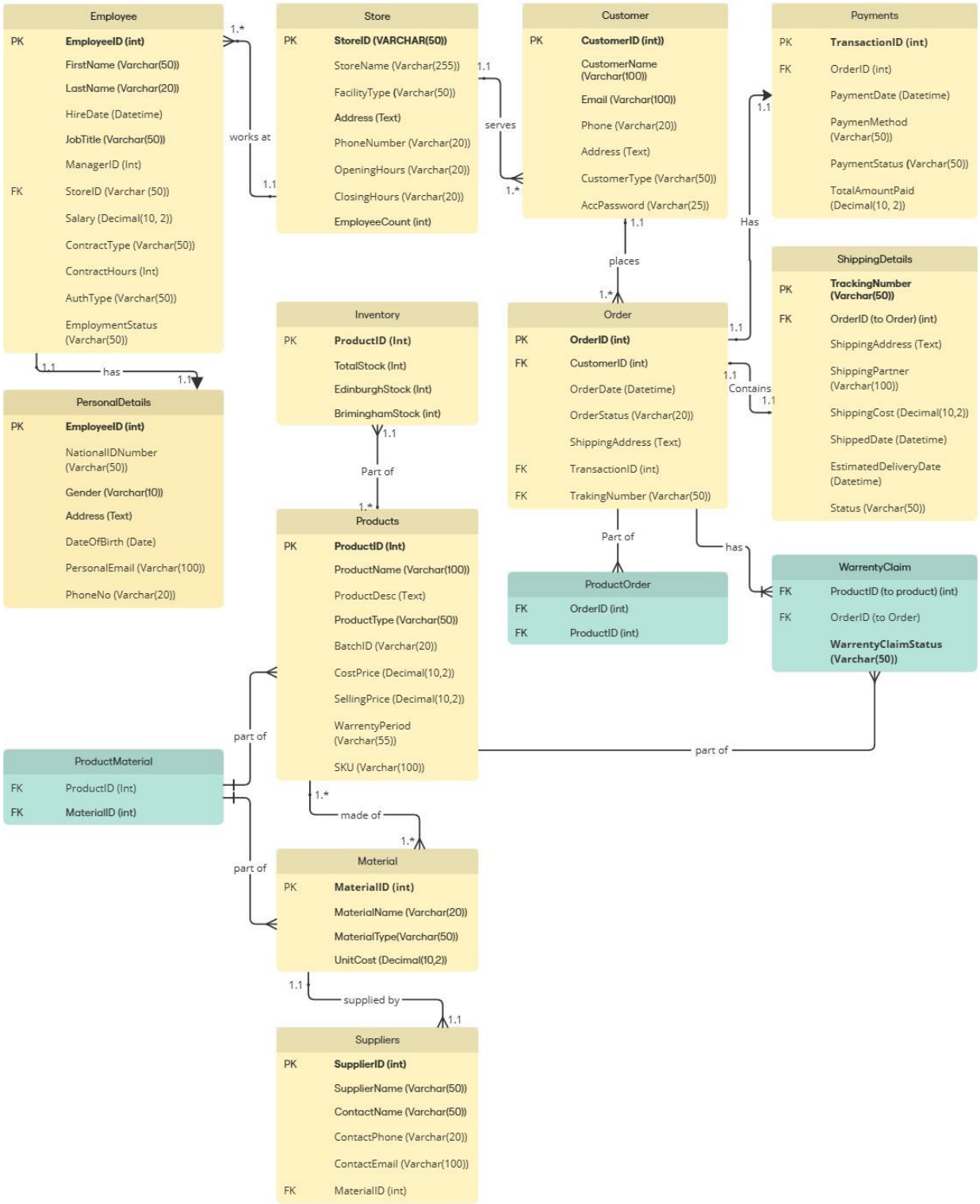
Usability and User Experience

- Design the database in a way that ensures ease of use for non-technical users, with a user-friendly interface.
- Provide search and filter functionalities for quick access to product, order, or customer data.

Scalability and Performance

- The database must be scalable to handle increased data volume as the business expands.

ER Diagram:


Employee
PK EmployeeID (int)
FirstName (Varchar(50))
LastName (Varchar(20))
HireDate (Datetime)
JobTitle (Varchar(50))
ManagerID (Int)
FK StoreID (Varchar (50))
Salary (Decimal(10, 2))
ContractType (Varchar(50))
ContractHours (Int)
AuthType (Varchar(50))
EmploymentStatus (Varchar(50))
Store
PK StoreID (VARCHAR(50))
StoreName (Varchar(255))
FacilityType (Varchar(50))
Address (Text)
PhoneNumber (Varchar(20))
OpeningHours (Varchar(20))
ClosingHours (Varchar(20))
EmployeeCount (int)
Customer
PK CustomerID (int))
CustomerName (Varchar(100))
Email (Varchar(100))
Phone (Varchar(20))
Address (Text)
CustomerType (Varchar(50))
AccPassword (Varchar(25))
Payments
PK TransactionID (int)
FK OrderID (int)
PaymentDate (Datetime)
PaymenMethod (Varchar(50))
PaymentStatus (Varchar(50))
TotalAmountPaid (Decimal(10, 2))
works at
serves
Has
places
has
PersonalDetails
PK EmployeeID (int)
NationalIDNumber (Varchar(50))
Gender (Varchar(10))
Address (Text)
DateOfBirth (Date)
PersonalEmail (Varchar(100))
PhoneNo (Varchar(20))
Inventory
PK ProductID (Int)
TotalStock (Int)
EdinburghStock (Int)
BrimminghamStock (int)
Order
PK OrderID (int)
FK CustomerID (int)
OrderDate (Datetime)
OrderStatus (Varchar(20))
ShippingAddress (Text)
FK TransactionID (int)
FK TrakingNumber (Varchar(50))
ShippingDetails
PK TrackingNumber (Varchar(50))
FK OrderID (to Order) (int)
ShippingAddress (Text)
ShippingPartner (Varchar(100))
ShippingCost (Decimal(10,2))
ShippedDate (Datetime)
EstimatedDeliveryDate (Datetime)
Status (Varchar(50))
Contains
Part of
Products
PK ProductID (Int)
ProductName (Varchar(100))
ProductDesc (Text)
ProductType (Varchar(50))
BatchID (Varchar(20))
CostPrice (Decimal(10,2))
SellingPrice (Decimal(10,2))
WarrentyPeriod (Varchar(55))
SKU (Varchar(100))
ProductOrder
FK OrderID (int)
FK ProductID (int)
WarrentyClaim
FK ProductID (to product) (int)
FK OrderID (to Order)
WarrentyClaimStatus (Varchar(50))
part of
ProductMaterial
FK ProductID (Int)
FK MaterialID (int)
made of
Material
PK MaterialID (int)
MaterialName (Varchar(20))
MaterialType(Varchar(50))
UnitCost (Decimal(10,2))
supplied by
Suppliers
PK SupplierID (int)
SupplierName (Varchar(50))
ContactName (Varchar(50))
ContactPhone (Varchar(20))
ContactEmail (Varchar(100))
FK MaterialID (int)

SQL Statements:

```
CREATE TABLE Employee (
    EmployeeID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(20) NOT NULL,
    HireDate DATE,
    JobTitle VARCHAR(50),
    ManagerID INT,
    StoreID VARCHAR(50),
    Salary DECIMAL(10, 2),
    ContractType VARCHAR(50),
    ContractHours INT,
    AuthType VARCHAR(50),
    EmploymentStatus VARCHAR(50) DEFAULT 'Active',
    FOREIGN KEY (StoreID) REFERENCES Store(StoreID)
  );
CREATE TABLE PersonalDetails (
    EmployeeID INT NOT NULL PRIMARY KEY,
    NationalIDNumber VARCHAR(50) UNIQUE,
    Gender VARCHAR(10),
    Address TEXT,
    DateOfBirth DATE,
    PersonalEmail VARCHAR(100),
    PhoneNumber VARCHAR(20),
    );
CREATE TABLE Store (
    StoreID VARCHAR(50) PRIMARY KEY,
    StoreName VARCHAR(255) NOT NULL,
    FacilityType VARCHAR(50),
    Address TEXT,
    PhoneNumber VARCHAR(20),
    OpeningHours VARCHAR(20),
```

```
    ClosingHours VARCHAR(20),
    EmployeeCount INT,
);
CREATE TABLE Customer (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
    CustomerName VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Phone VARCHAR(20),
    Address TEXT,
    CustomerType VARCHAR(50) DEFAULT 'Individual',
    AccPassword VARCHAR(25),
);
CREATE TABLE 'Order' (
    OrderID INT AUTO_INCREMENT PRIMARY KEY,
    CustomerID INT NOT NULL,
    OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    OrderStatus VARCHAR(20) DEFAULT 'Pending',
    ShippingAddress TEXT,
    TransactionID INT,
    TrackingNumber VARCHAR(50),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
    FOREIGN KEY (TransactionID) REFERENCES Payments(TransactionID)
    FOREIGN KEY (TrackingNumber) REFERENCES ShippingDetails(TrackingNumber)
);
CREATE TABLE Payments (
    TransactionID INT PRIMARY KEY,
    OrderID INT NOT NULL,
    PaymentDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    PaymentMethod VARCHAR(50) DEFAULT 'Credit Card',
    PaymentStatus VARCHAR(50) DEFAULT 'Pending',
    TotalAmountPaid DECIMAL(10, 2) NOT NULL,
```

```
    FOREIGN KEY (OrderID) REFERENCES 'Order'(OrderID)
);
CREATE TABLE ShippingDetails (
    TrackingNumber VARCHAR(50) PRIMARY KEY UNIQUE,
    OrderID INT NOT NULL,
    ShippingAddress TEXT NOT NULL,
    ShippingPartner VARCHAR(100),
    ShippingCost DECIMAL(10, 2) NOT NULL,
    ShippedDate DATETIME,
    EstimatedDeliveryDate DATETIME,
    Status VARCHAR(50) DEFAULT 'Pending',
    FOREIGN KEY (OrderID) REFERENCES `Order`(OrderID)
);
CREATE TABLE ProductOrder (
    OrderID INT NOT NULL,
    ProductID INT NOT NULL,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
CREATE TABLE WarrantyClaim (
    ProductID INT NOT NULL,
    OrderID INT NOT NULL,
    WarrentyClaimStatus VARCHAR(50) DEFAULT 'Pending',  -- Default status when a warranty is
created
   FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
   FOREIGN KEY (OrderID) REFERENCES 'Order'(OrderID)
);
CREATE TABLE Products (
    ProductID INT AUTO_INCREMENT PRIMARY KEY,
    ProductName VARCHAR(100) NOT NULL,
    ProductDesc TEXT,
    ProductType VARCHAR(50),
    BatchID VARCHAR(20),
```

```
    CostPrice DECIMAL(10, 2) NOT NULL,
    SellingPrice DECIMAL(10, 2) NOT NULL,
    WarrentyPeriod VARCHAR(55),
    SKU VARCHAR(100) UNIQUE,
   );
CREATE TABLE Inventory (
    ProductID INT PRIMARY KEY,
    TotalStock INT DEFAULT 0,
     EdinburghStock INT DEFAULT 0,
     BriminghamStock INT DEFAULT 0,
  );
CREATE TABLE Material (
    MaterialID INT PRIMARY KEY AUTO_INCREMENT,
    MaterialName VARCHAR(20) NOT NULL,
    MaterialType VARCHAR(50),
    UnitCost DECIMAL(10, 2),
  );
CREATE TABLE Suppliers (
    SupplierID INT AUTO_INCREMENT PRIMARY KEY,
    SupplierName VARCHAR(50) NOT NULL,
    ContactName VARCHAR(50),
    ContactPhone VARCHAR(20),
    ContactEmail VARCHAR(100),
    MaterialID INT NOT NULL,
    FOREIGN KEY (MaterialID) REFERENCES MaterialType(MaterialID)
);
CREATE TABLE ProductMaterial (
  ProductID INT NOT NULL,
  MaterialID INT NOT NULL,
  FOREIGN KEY (MaterialID) REFERENCES Material(MaterialID),
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```