

DAMG 6210

Massachusetts Courier Service Management System

Presented by: Group 12

Group Members:

- Tejas Choudhari
- Shreeyash Lahane
- Anushree Janardhan
- Hemant Jomraj



deli





Introduction



It is currently very vital for people to ship or receive commodities such as imported furniture, electrical equipment, gifts, commercial goods, and so on. But, there lies a major inconvenience of constantly being in touch with the courier office for updates about the shipments.

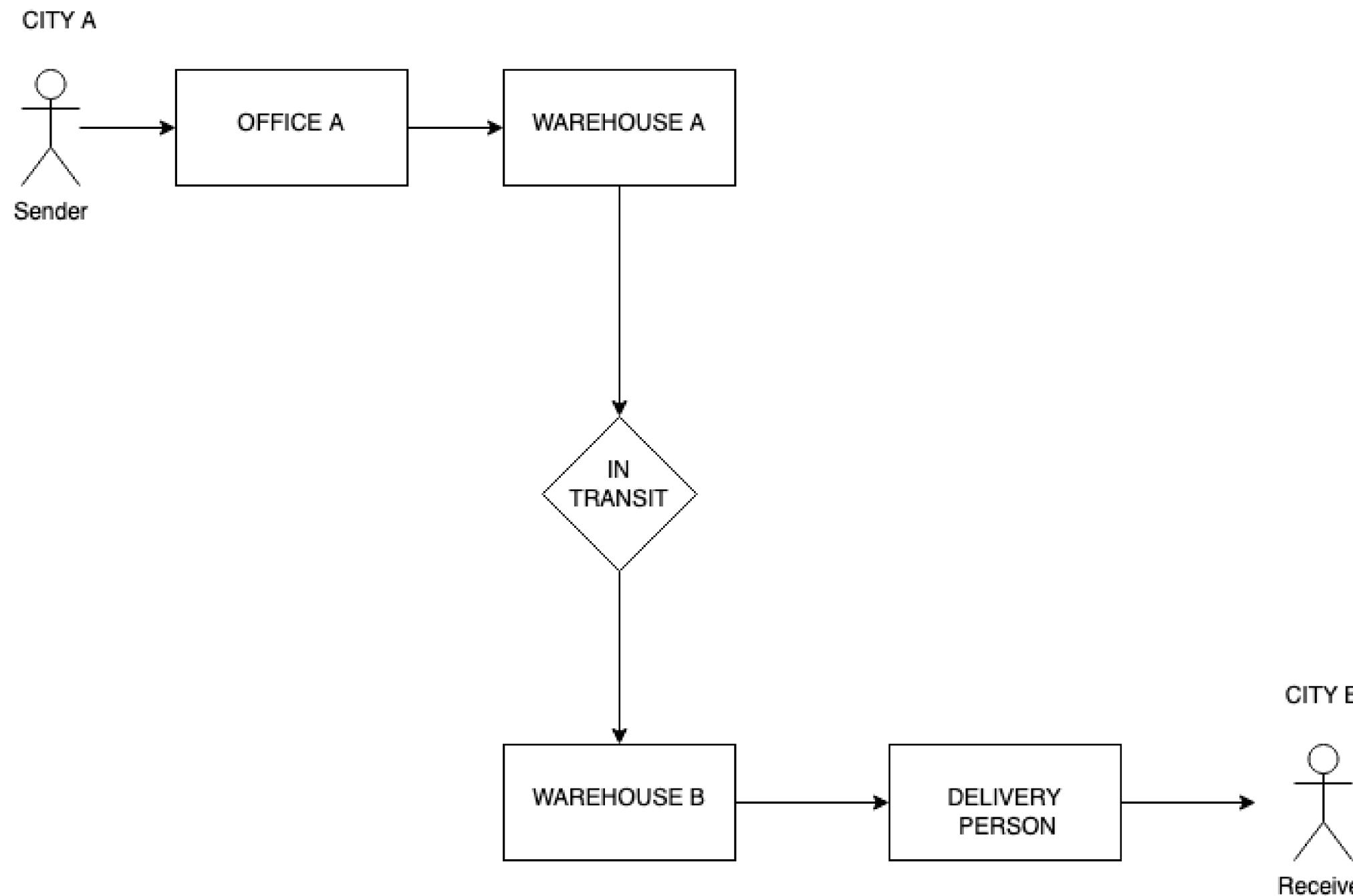
The Massachusetts Courier Service Management System is a system that enables customers to order courier services to send goods from a source location to a specified destination.

OBJECTIVES

- To track activities such as booking, delivery, and status checks and provide timely notification updates to customers
- To provide effective time management, which ensures that remote labor operates well while accommodating various locations
- To offer priority-based and location-specific services, with varied payment options

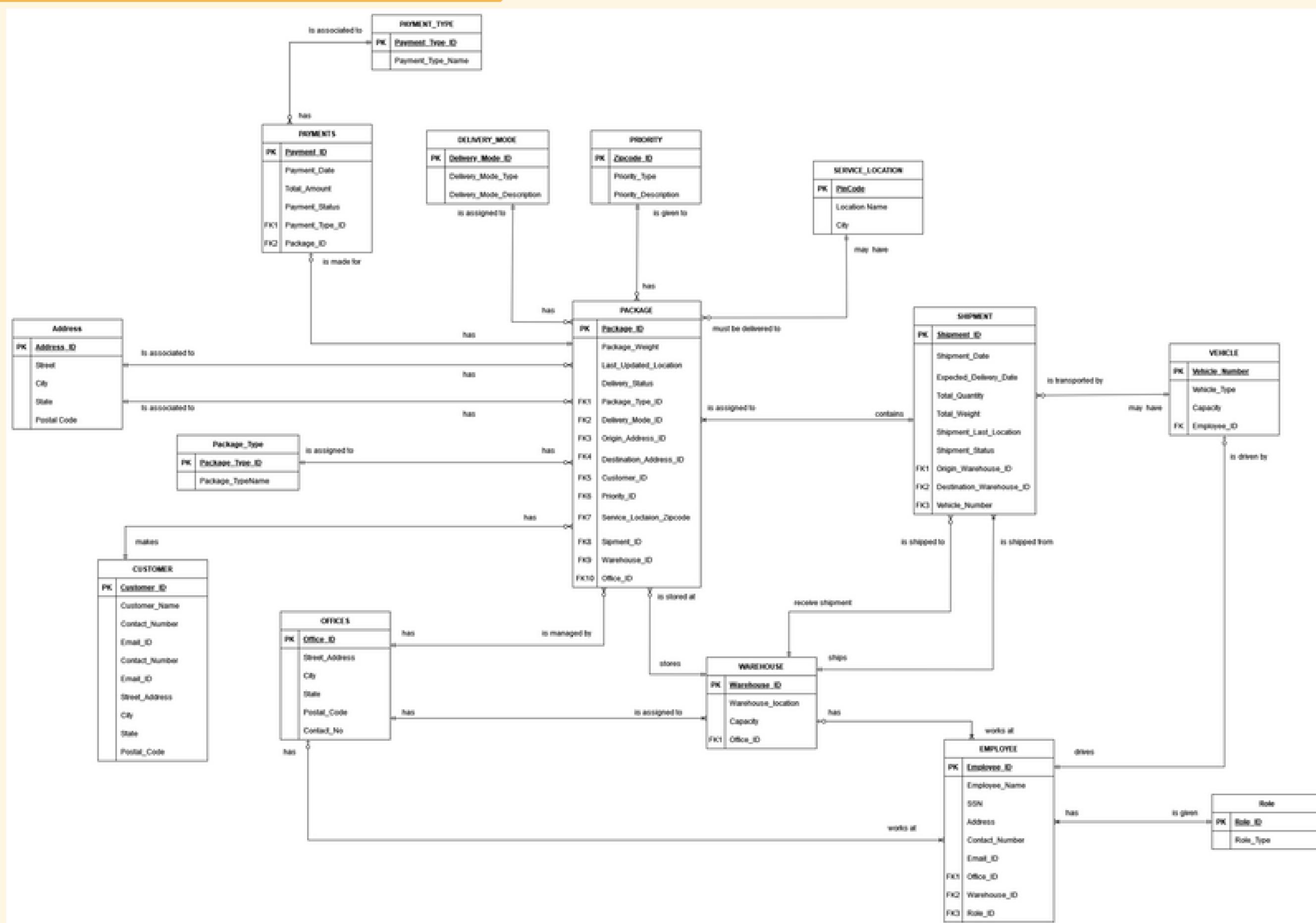
Business Logic

Entities



- Customers
- Package
- Address
- Payments
- Shipment
- Offices
- Warehouse
- Employee
- Role
- Vehicle
- Package_Type
- Delivery_Mode
- Priority_Type
- Service_Location
- Package
- Priority
- Delivery_Mode
- Service_Location

Final ERD



Database Implementation

Views

```
CREATE TABLE Package(
    Package_ID INT IDENTITY(1000,1) NOT NULL PRIMARY KEY,
    Package_Weight NUMERIC(3,1) NOT NULL,
    Last_Updated_Location VARCHAR(20) NOT NULL,
    Delivery_Status VARCHAR(15) NOT NULL,
    Package_Type_ID INT NOT NULL,
    Delivery_Mode_ID INT NOT NULL,
    Origin_Address_ID INT NOT NULL,
    Destination_Address_ID INT NOT NULL,
    Customer_ID INT NOT NULL,
    Priority_ID INT NOT NULL,
    Service_Location_Zipcode VARCHAR(5) NOT NULL,
    Shipment_ID INT,
    Warehouse_ID INT,
    Office_ID INT NOT NULL,
    CONSTRAINT pkg_weight_ck CHECK(Package_Weight < 100),
    CONSTRAINT delivery_status_ck CHECK(Delivery_Status IN('Order Placed','Dispatched','In-Transit','Delivered','Cancelled','Delayed')),
    CONSTRAINT pkg_pti_fk FOREIGN KEY(Package_Type_ID) REFERENCES Package_Type(Package_Type_ID),
    CONSTRAINT dm_dmi_fk FOREIGN KEY(Delivery_Mode_ID) REFERENCES Delivery_Mode(Delivery_Mode_ID),
    CONSTRAINT add_org_fk FOREIGN KEY(Origin_Address_ID) REFERENCES Address(Address_Id),
    CONSTRAINT add_dest_fk FOREIGN KEY(Destination_Address_ID) REFERENCES Address(Address_Id),
    CONSTRAINT cust_custid_fk FOREIGN KEY(Customer_ID) REFERENCES Customer(Customer_ID),
    CONSTRAINT priority_pid_fk FOREIGN KEY(Priority_ID) REFERENCES Priority(Priority_ID),
    CONSTRAINT service_location_zip FOREIGN KEY(Service_Location_Zipcode) REFERENCES Service_Location(Zipcode),
    CONSTRAINT shipment_sid_fk FOREIGN KEY(Shipment_ID) REFERENCES Shipment(Shipment_ID),
    CONSTRAINT warehouse_pkg_wid_fk FOREIGN KEY(Warehouse_ID) REFERENCES Warehouse(Warehouse_ID),
    CONSTRAINT office_pkg_oid_fk FOREIGN KEY(Office_ID) REFERENCES Offices(Office_ID));
```

vw_highPriorityCustomers

(View to see Information of customers that have 'ONE DAY RUSH' priority packages)

```
CREATE VIEW vw_highPriorityCustomers AS
(
    Select C.Customer_ID, C.Customer_Name,
    C.Contact_Number, C.Email_ID, C.
    [State],C.Postal_Code, C.Street_Address,
    PR.Priority_Type,PR.Priority_Description
    from CUSTOMER C INNER JOIN Package Pack
    ON C.CUSTOMER_ID= Pack.CUSTOMER_ID
    INNER JOIN PRIORITY PR
    ON Pack.Priority_ID = PR.priority_id
    Where PR.Priority_Description = 'One Day
    Rush'
);
```

User Defined Functions

udf_GetPaymentStatus (Returns Payment Status From Package ID)

```
CREATE FUNCTION udf_GetPaymentStatus
(
    @Package_ID int
)
RETURNS VARCHAR(25)
AS
BEGIN
    DECLARE @PaymentStatus varchar(25)

    SELECT @PaymentStatus = Payment_Status
    FROM PAYMENTS pay INNER JOIN Package p ON
    p.Package_ID=pay.Package_ID
    WHERE pay.Package_ID=@Package_ID
    RETURN @PaymentStatus
END
```

Stored Procedures

usp_UpdateShipmentStatus (Updates Shipment Status as 'Delayed' on condition that Current Date exceeds Expected Delivery Date and Package is not yet Delivered)

```
CREATE PROCEDURE usp_UpdateShipmentStatus
AS
BEGIN
    BEGIN TRY

        BEGIN TRAN
        UPDATE Shipment
        SET Shipment_Status = 'Delayed'
        WHERE Expected_Delivery_Date < CAST(GETDATE() as DATE) AND
        Shipment_Status <> 'Delivered'
        COMMIT
        RETURN 1
    END TRY

    BEGIN CATCH
        PRINT 'Error occurred while updating Shipment Status'
        ROLLBACK
        RETURN -99
    END CATCH
END
GO
```

Triggers

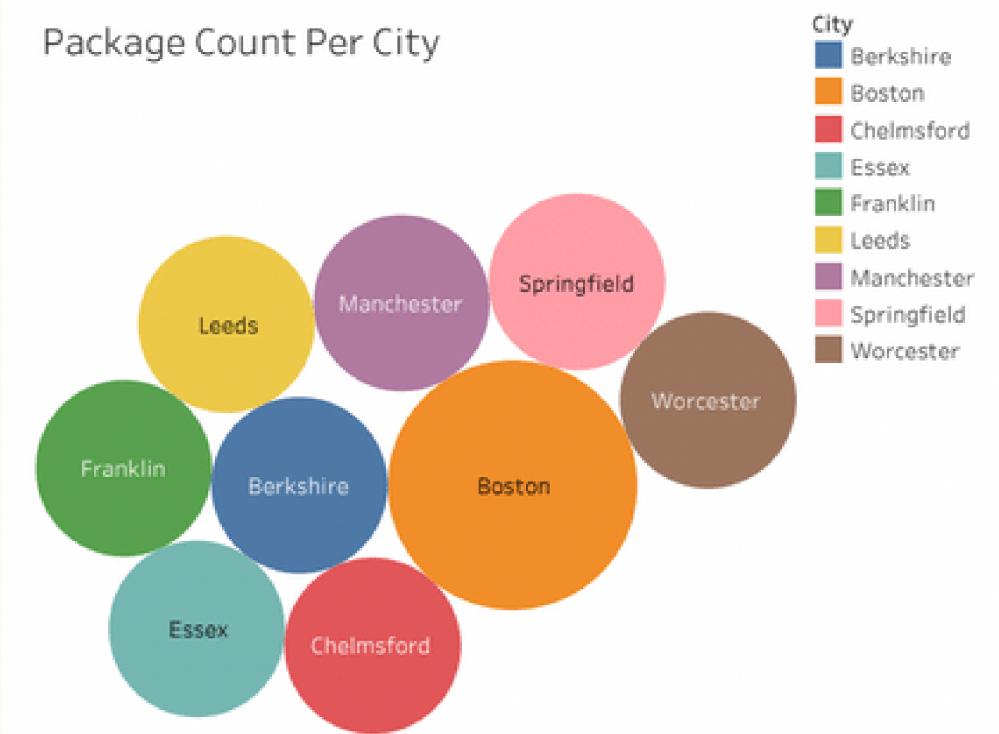
```
CREATE TRIGGER trg_Shipment_Logs
ON Shipment
AFTER INSERT,UPDATE,DELETE
AS BEGIN
SET NOCOUNT ON;

INSERT INTO Shipment_Logs(
    Shipment_ID, Shipment_Date, Expected_Delivery_Date, Total_Quantity, Total_Weight,
    Shipment_Last_Location, Shipment_Status, Origin_Warehouse_ID, Destination_Warehouse_ID,
    Vehicle_Number, updated_at, operation
)
SELECT
    i.Shipment_ID, Shipment_Date, Expected_Delivery_Date, Total_Quantity,
    Total_Weight, i.Shipment_Last_Location, Shipment_Status,
    Origin_Warehouse_ID, Destination_Warehouse_ID,
    Vehicle_Number, GETDATE(), 'INS'
FROM
    inserted i
UNION ALL
SELECT
    d.Shipment_ID, Shipment_Date, Expected_Delivery_Date,
    Total_Quantity, Total_Weight, d.Shipment_Last_Location,
    Shipment_Status, Origin_Warehouse_ID, Destination_Warehouse_ID,
    Vehicle_Number,
    GETDATE(),
    'DEL'
FROM
    deleted d;

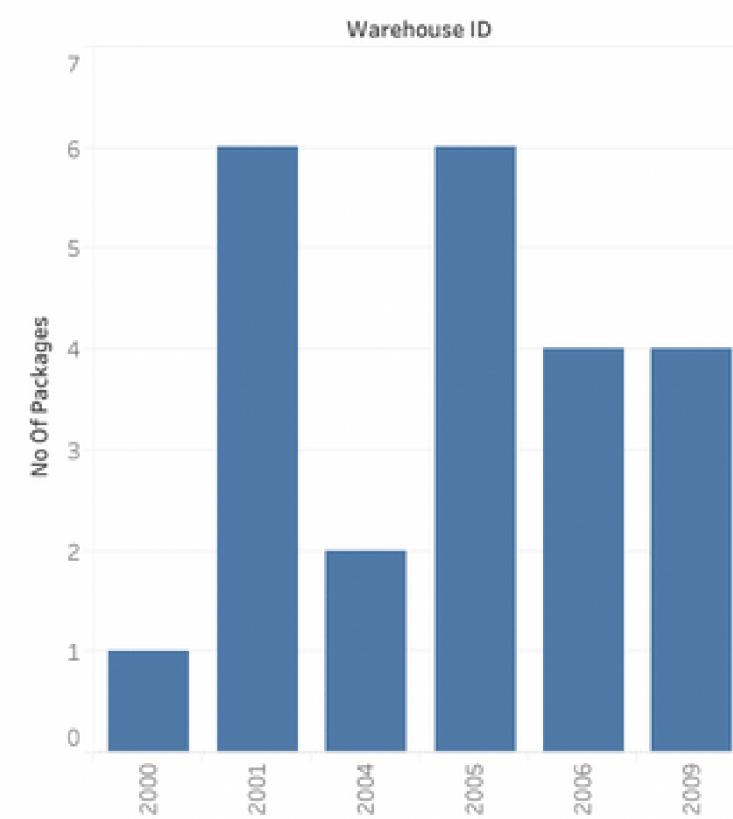
END
GO
```

Data Visualization

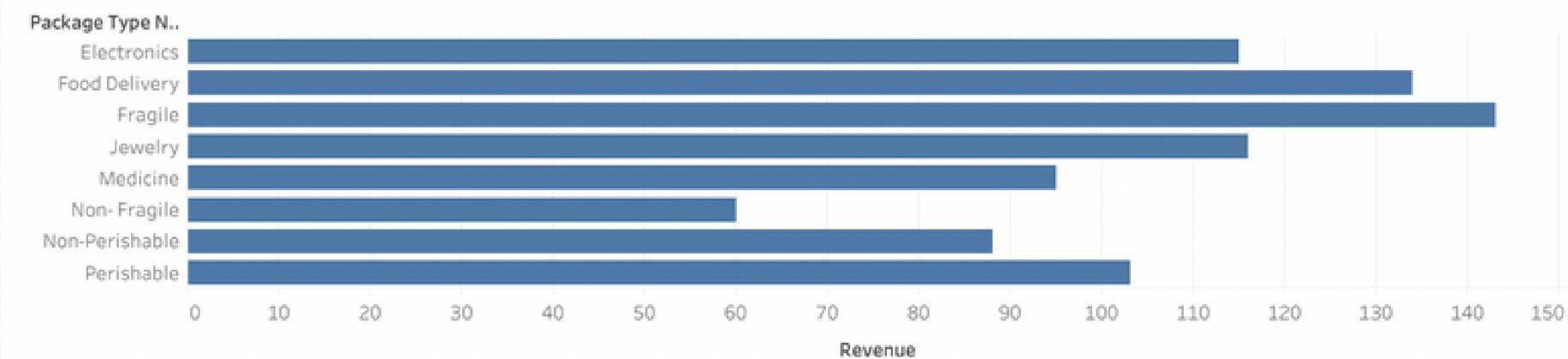
Package Count Per City



Package In Warehouses To Be Shipped



Revenue Per PackageType



Thank you for listening!

