

Assignment 2: Database Design

Kieran Cosson

Scenario Description

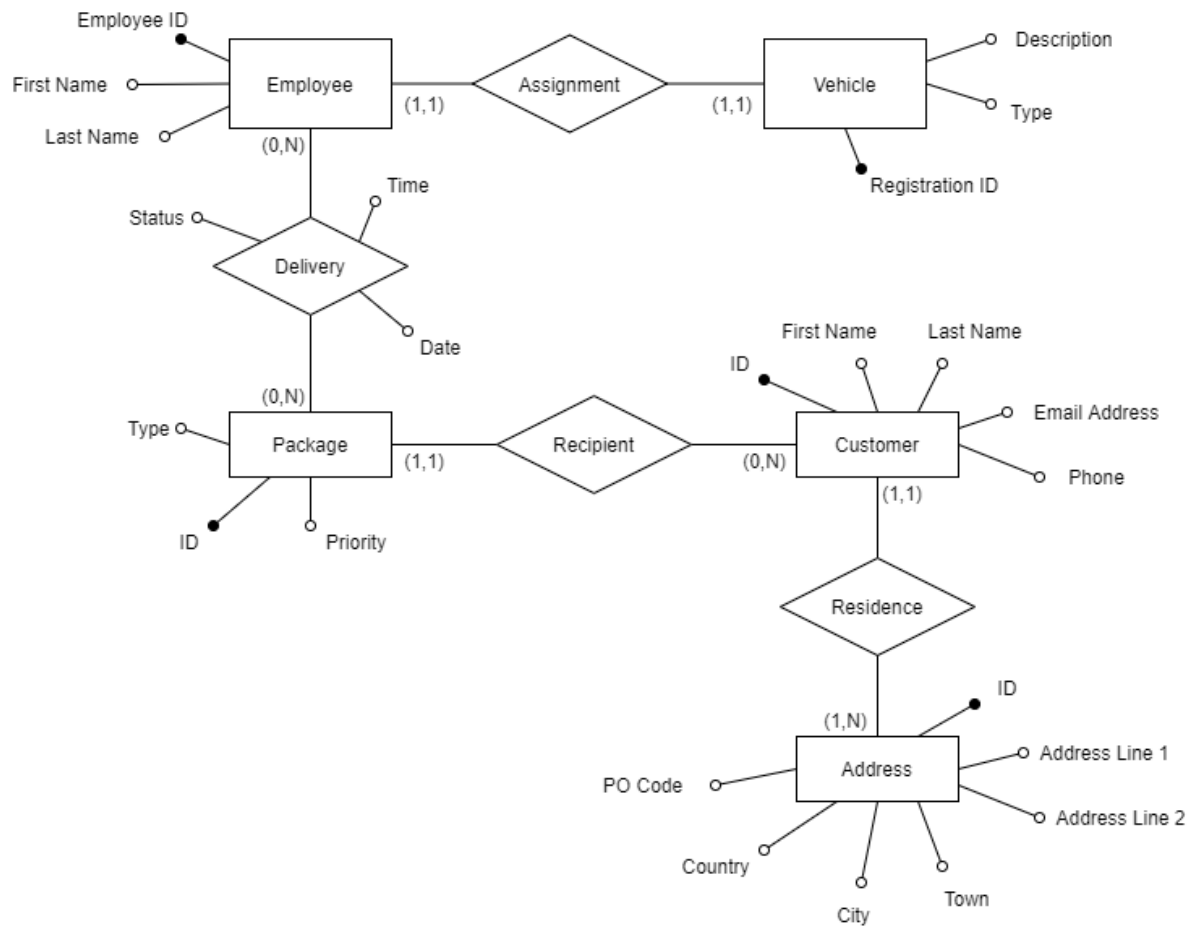
A delivery service needs to keep track of packages to be delivered. For the employees, this means knowing their delivery schedule. For customers, this means knowing the status of their delivery.

These packages have a type, priority, and recipient. A recipient customer will have an address, and some contact details will be kept on them in case they need to be contacted. The deliveries are carried out by employees (drivers), who have an assigned company vehicle. The vehicle is identified by its registration number, but should have a text description also for the sake of the user. The delivery has a datetime, and associated status - which may be pending, in progress, failed, or successful. The date and time will only be filled in when that information becomes known.

Assumptions and Constraints:

- Each person may be the recipient of many packages. Each package has a single recipient. (A one-to-many relationship). A person does not necessarily have to have a current order to be present in the database.
- Each person has a single (delivery) address. This is done for simplicity in the assignment, though in practice the database might be extended to support multiple delivery addresses. The address is not optional.
- Each package can be assigned to a delivery driver. The delivery date may be in the future, if the delivery person has yet to make the delivery, or it may be in the past. If the delivery has yet to occur, it might not yet have any employee tasked to it yet. If a delivery fails, another attempt may be made, potentially on a different date with a different driver. Hence, this is a many-to-many relationship.
- An employee may have a company vehicle assigned to them. They will need one at some point to carry out the deliveries, however, the vehicle assigned may change, and there may not be enough vehicles for all employees at the same time. (Especially if some are working part time) . This is a one-to-one relationship.
- One final note about IDs - all entities have some sort of numeric ID. The vehicle's registration number is the only number tied to the physical world, though the employee ID may be used in other company information systems. There are no assumptions being made regarding the uniqueness of employee, customer, address, or package attributes, other than the ID assigned to them by the company.

E-R Diagram



Relational Model

Tables:

Address (ID, Line 1, Line 2, Town, City, Country, PO Code)

Customer (ID, First Name, Last Name, Email, Phone, Address)

Package (ID, Name, Type, Priority, Customer)

Delivery (ID, Employee, Package, Date, Time, Status)

Employee (Employee Number, First Name, Last Name, Assigned Vehicle)

Vehicle (Registration Number, Type)

Additional Notes:

- The one-to-many relationship "Residence" is represented by the "Address" foreign key reference in the **Customer** table, referencing the **Address** table.
- The one-to-many relationship "Recipient" is represented by the "Customer" foreign key reference in the **Package** table, referencing the **Customer** table.
- The many-to-many relationship "Delivery" is represented by the table **Delivery**, which has "Package" and "Employee" foreign key references referencing the **Package** and **Employee** tables, respectively.
- The one-to-one relationship "Assignment" is represented by the "Assigned Vehicle" foreign key reference in the **Employee** table, referencing the **Vehicle** table.

Two Frequent Queries

1) Find the current status of all orders for a given customer ID

This seems likely to be useful for a (possibly web?) client facing app, where a client could login and check the current status of their past/pending orders.

This will involve the Customer, Package, and Delivery tables.

2) Find packages assigned for delivery by an employee, along with the address/datetime

Within the company, there might be a timetable application that would allow employees to check their schedule over the coming days.

This will involve all tables except for the Vehicles table.

I'm assuming sorting and filtering (perhaps by date) would be handled by the app.

Queries Expressed in SQL

1) Find the current status of all orders for a given customer ID

(This query is parametric, and takes the input variable `Customer ID`)

```
SELECT Package.ID, Package.Name, Delivery.Date, Delivery.Status FROM Delivery
JOIN Package ON Delivery.Package = Package.ID
JOIN Customer ON Package.Customer = Customer.ID
WHERE Customer.ID = `Customer ID`;
```

2) Check vehicle assigned to an employee ID

(This query is parametric, and also takes the input variable `Employee ID`)

```
SELECT Vehicle.*, FROM Vehicle
JOIN Employee ON `Employee.Assigned Vehicle` = `Vehicle.Registration Number`
WHERE Employee.ID = `Employee ID`
```