DAMG 6210

P2: Database Design and Initial ERD

Team 6:
Shaoshuai Xu
Shuxin Li
Yarong Wang
Ruoxin Wang
Yifan Jiang

Feb 21, 2022

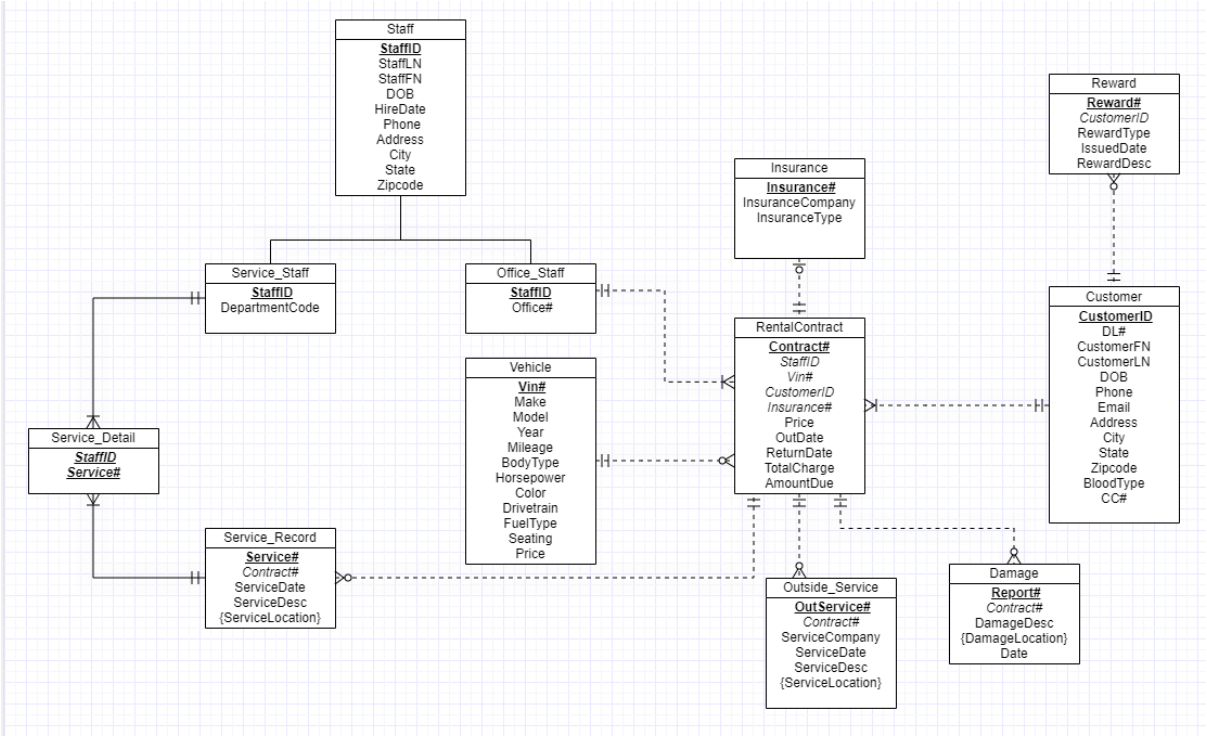
Database Topic:

Rental Car System

Database Purpose:

The purpose of the database is to select the suitable vehicles accurately according to customer requirements, track the vehicle rental records and vehicle maintenance status, maintain customer information, vehicle information and employee information. It will be used by vehicle rental corporations only.

ERD:



Business Problems Addressed:

Allow office staff to track the status for maintenance of vehicles.
Allow administrative and office staff to generate rental contracts.
Allow service staff to anticipate repair materials for damaged cars.
Provide vehicle's information to improve the quality of rental choices for customers.
Provide promotions to specific market segments(e.g., to reward the customer with a good rental history).

Business Rules:

Each customer can only rent one vehicle at one time.
Each vehicle can only be rented by one customer at one time.
Each rental contract has to be signed by one office staff.
Each Office staff can be signed to many rental contracts.

Each rental contract can have none or no more than one insurance.
Each rental contract can have none or many damage reports.
Each rental contract can have none or many service reports by non-company owned auto repair shops.
Each rental contract can have none or many service reports by the company owned auto repair shops.
Each service record of the company owned auto repair shops can be signed by many service staff.
Each service staff can be signed to many service records.
Each customer can have none or many rewards.
Each customer can not have a new rental contract if his/her last contract is not fully paid.

Design Decisions:

Entity Name	Why Entity Included	How Entity is Related to Other Entities
Staff	The company will need to store the information of their staff. Using it to maintain staff data and status.	Staff entity has two sub tables, Office_Staff and Service_Staff.
Office_Staff	This entity is a subtype of Staff entity. Store special attributes only for office staff.	The sub table of Staff. Also related to the Rental Contract entity as one of the necessary elements of the contract.
Service_Staff	This entity is a subtype of Staff entity. Store special attributes only for service staff.	The sub table of Staff. Also related to the bridge table Service_Detail between Service_Staff and Service_Record many to many relationships.
Vehicle	Store all vehicle data such as Make, Model and Year. Using it to maintain vehicle data and track the vehicle status. Using it to select the suitable vehicles accurately according to customer requirements.	Vehicle entity is related to the RentalContract entity as one of the necessary elements of the contract. Making sure it is able to view the vehicle information, update the vehicle data and track the vehicle status by using the Vin# from the RentalContract entity.

Customer	The purpose of this table is to collect information about customers, Store all Customer data such as First Name, Last Name, date of birth and Phone number. Using it to maintain customer data.	Customer entity is related to the Rental Contract entity as one of the necessary elements of the contract.
RentalContract	The bridge table between Vehicle, Office staff and Customer. Using it to avoid many to many relationships. Store Customer ID, Office staff ID, Vehicle Vin# and other rental contract details. Using it to view and track the contract data and status such as damage report, rental period, and amount due.	RentalContract entity is related to Vehicle, Office_Staff, and Customer entity. It is the bridge table between Vehicle, Office_Staff, Customer, many to many relationships. Contract # is the primary key of the entity and also include vehicle Vin#, Office_StaffID, CustomerID as foregin key from each entity.
Insurance	Using it to maintain the insurance information of each car for renting such as insurance company and insurance type.	Each car for renting can have one insurance at most. Thus the relationship between the RentalContract entity and the Insurance entity can be considered as a one to one relationship. The attribute Insurance# is the primary key of the Insurance table and is also the foreign key of the RentalContract table. Each corresponding insurance information can be found through the attribute Insurance# in the RentalContract table.
Damage	It is necessary to maintain and update conditions of vehicles for renting.	The Damage entity is associated with the RentalContract entity. Each damage report describes conditions of only one vehicle. Therefore the relationship between the Damage table and the RentalContract table belongs to a one to one relationship. The attribute Report# is the primary key and the attribute Contract# is the foreign key

		of the Damage entity separately. Relevant details about contracts can be obtained by joining the RentalContract table and the Damage table on their common attribute Contract#.
Outside_Service	Customers do not have to enjoy the services provided by vehicle rental companies. Some customers may have their cars repaired in other vehicle repair companies.Using it to maintain and update the details about outside services chosen by customers.	The Outside_Service entity is related to the RentalContract entity. Each Outside_Service represents that it provides one customer with its service. Therefore the relationship between the Outside_Service entity and the RentalContract entity can be regarded as a one to one relationship. The attribute Contract# serves as the foreign key of the Outside_Service table. Customers' information can be tracked by Contract#.
Service_Record	Using it to seek for the details about customers through the attribute Contract#. It can also be used to identify the information about services such as service date and service location.	The Service_Record entity is associated with Service_Detail entity and Contract entity. The attribute Contract# acts as the foreign key of the Service_Record entity. Through Contract#, corresponding customer information can be abstracted. The relationship between the RentalContract entity and the Service_Record entity is a one to one relationship. Service# is not only the primary key but also the foreign key of the Service_Record table, Corresponding service staff can also be identified through Service#. The relationship between the Service_Record entity and the Service_Detail entity is a one to many relationship.

Service_Detail	The relationship entity which displays the relationship between Service_Staff and Service_Record. Using it to track the services completed by the service staff.	The Service_Detail entity is related to the Service_Staff entity and the Service_Record entity. Each service staff can engage in many services. Each service can also be completed by many service staff. Thus the relationship between the Service_Staff entity and the Service_Record entity is a many to many relationship. The attribute StaffID and the attribute Service# in the Service_Detail table also play the role as the foreign key of the Service_Staff table and the Service_Record table separately.
Reward	Using it to maintain and update each reward each customer received.Using it can acquire relevant information about rewards the customers got such as reward type and issued date.	The Reward entity is related to the Customer entity. Each customer can have more than one reward. As a result, the relationship between the Customer entity and the Reward entity is a one to many relationship. The attribute CustomerID in the Reward table is the foreign key of the Reward table. Customers who got rewards can be tracked through CustomerID in the Reward table and the Customer table.