Entities

Customer:
A strong entity with the attributes: CustomerID, Name, PhoneNumber, Email, Address, DriversLicense
Primary key: CustomerID
Composite Attributes: Name(FirstName, MiddleName, LastName), Address(StreetNumber, StreetName, City), PhoneNumber, Email *Although all of these are attributes, we only showed the subdivisions of those attributes that our DBMS will need to refer to. For example, our DBMS will never need to refer to just the domain of an email address or just the area code of a phone number.*

Review:
A strong entity with the attributes: Rating, ReviewDate, ReviewID, Feedback
Primary key: ReviewID
Composite attributes: ReviewDate(Day, Month, Year) *Once again, we do not show the subvisions of the date the review was written because we do not need it for our DBMS*

Reservation:
A strong entity with the attributes: ReservationID, RStatus, PPickupDate, PReturnDate
Primary key: ReservationID
Composite attributes: PPickupDate(Day, Month, Year, PReturnDate(Day, Month, Year) *In this case we kept the details of the dates because we need to know it to cross check when the car was actually picked up and dropped off down to the time, for cases like we have a reservation for that same car right after*

Vehicle:
A strong entity based on the rental object itself with attributes: Vehicle ID, FuelCapacity, DailyRate, VType, Plate, Model, VStatus, Color.
Primary Key: Vehicle ID
Multivalued Attribute: Colour (In the case of a two tone colour car)

Maintenance:
A weak entity whose owner entity is Vehicle. It has the attributes: ServiceDate, MStatus. It is also a super or parent class.
Partial Key: Service Date: uniquely identifies it within the context of a maintenance record but must be combined with the vehicle ID to form its primary key.
Composite attributes: ServiceDate which is broken down into the Day, Month, and Year.

Cleaning:
A specialized entity that is a subclass of maintenance with attributes: CleaningArea, DetailLevel

Repair:
A specialized entity that is a subclass of maintenance with attributes: RepairType, PartsRepaired
Multivalued attributes: PartsTRepaired (if multiple parts are repaired)

Rental:
A strong entity with attributes: RentalID, PickupDate, ReturnDate, RentalCost, FuelCharge, FuelLevel
Primary Key: RentalID
Composite attributes: PickupDate(Day, Month, Year), ReturnDate(Day, Month, Year)

Payment:
A strong entity with attributes: PaymentID, PaymentDate, PaymentMethod, TotalAmount
Primary Key: PaymentID
Composite attribute: PaymentDate(Day, Month, Year)
Derived attribute: TotalAmount (this can be derived from the sum of RentalCost and FuelCharge from the Rental entity)

Relationships

Customer - Review:
Cardinality: 1:N because each customer can WRITE multiple reviews, but each review can only be written by one customer
Participation: Customer has partial participation in this relationships because not every customer WRITES a review. Review has total participation because every review must be WRITTEN by a customer
Relationship Attributes: The relationship WRITE has the attribute WritingMethod, which basically is how the review was written (online, text, etc.)

Review - Vehicle:
Cardinality: N:1 because each review is ABOUT one vehicle. However, a vehicle record can be related to multiple reviews over time by various customers.
Participation: Review to Vehicle is total participation because a review cannot exist unless it is tied to some vehicle rented. However, Vehicle to Review is partial because not every vehicle needs or will have a review on it.

Customer - Reservation:
Cardinality: 1: N because each customer can BOOK multiple reservations, but each reservation can only be BOOKED by one customer

Participation: Customer has partial participation in this relationships because not every customer BOOKS a reservation. Reservation has total participation because every reservation must be BOOKED by a customer

Reservation - Vehicle:
Cardinality: N: 1 because every reservation is FOR one vehicle (no overlaps), but each vehicle can be tied to multiple reservations over time.
Participation: Reservation FOR Vehicle is Total as a vehicle entity cannot exist without a valid Reservation record. Vehicle to Reservation is partial because not all vehicles in the system would be connected to a reservation at a given time.

Vehicle - Maintenance:
Cardinality: 1:N because every vehicle can GOES_UNDER multiple maintenance records over time, not restricted to one. However, each maintenance record is strictly for one vehicle.
Participation: Maintenance being a weak entity, must have total participation to its owner entity. However, vehicle to maintenance is partial participation because some vehicle records can exist without having gone under maintenance. The Maintenance entity has an identifying relationship with the Vehicle entity .

Reservation - Rental:
Cardinality: 1:1 because each reservation LEADS_TO at most one rental, and each rental comes from exactly one reservation.
Participation: Reservation has partial participation in this relationship because not every reservation LEADS_TO to a rental. Rental has total participation in this relationship because every rental must come from one reservation.

Rental - Payment:
Cardinality: 1:1 because each rental record BELONGS_TO exactly one payment record and vice versa.
Participation: Rental has total participation in this relationship because every rental must be paid. Payment has total participation as well because every payment must be linked to a rental. No rental or payment can exist without the other.

Disjointness constraint for the subclasses because a vehicle can only be under cleaning maintenance or repair maintenance at a time, not both