INTEGRITY CONSTRAINTS

Not null data and reasoning

* Branch
  + BID
    - Bid is sequential and most notably the primary key for the branch so it can not be null
  + City
    - The City should know when we decide to open a branch so we should know where it is
  + Province
    - Same as a city for the province
* CarType
  + CarType
    - The primary key for car type it is things like SUV truck sedan
  + Daily Rate/WeeklyRate/MonthlyRate
    - All of the rates are required
* Car
  + Vin
    - Each car has a unique vin that we can use as an identifier
  + Make/Model
    - We should know the make and model of any car we order
  + Year
    - We should know the year of any cars we order(also for the searching year can be important to someone)
  + noofseats
    - We should know the seat count of any car we order(also for searching noofseats can be very important for larger family’s)
  + Colour
    - Should be known when ordered
  + Branch\_id
    - An FK referencing the brach that currently possesses it
    - The branch ordering the car should know their bench id and put it in
  + Cartype
    - An FK referencing cartype
    - The car type should be known when ordered
* Customers
  + Customer ID
    - The sequential primary key for the customer, every customer should be assigned a unique number and as it is a primary key cannot be null
  + First/Last Name
    - The customer should know their first/last name when they create an account
  + Dateofbirth
    - The customer should know when they were born when they create an account
  + Membership Status
    - The customer will be assigned a membership status when they create an account of 0(false)
* Rental
  + TID
    - The sequential primary key for each different transaction should have its own individual identifier
  + PickupDate/ReturnDate
    - A pickup and return time is necessary when creating a rental, the time when you get the car and the time when you are required to return the car by should be known from the start
  + CustomerID
    - FK referencing customer, every rental should have a customer
  + VIN
    - FK referencing Car
    - Every rental needs a car
  + PickupBID
    - FK referencing Branch
    - The branch where the car gets picked up
    - If ReturnBID is Null it is returned to the same branch
  + Total\_rentRevenue
    - The income calculated that we will make on the transaction

Null Data and reasoning

* Branch
  + Most of the Branches information has been left as Nullable as if this is a new location for a franchise all the franchise would have to do is buy the rights to open a branch and choose a general location (city and province) from there they need to determine more details but it is not required to do so immediately
* CarType
  + Everything is required no null values
* Car
  + While most things should be known when buying a new car the odometer number and insurance arent required
    - Odometernumber: we leave this out as it won't necessarily be 0 depending on where we pick it up from and how it reaches that location it will be higher
    - Insurance: while insurance will be required to rent out the vehicle it is not required to buy one so we do not include it
* Customer
  + Customer home address fields are not required if they are in the process of moving they should still be able to rent a car(like a truck)
  + While it seems insane that one would not have a phone in the modern-day and age it is possible
  + As for insurance and driving license these are left empty for customer convenience. While both will be required to rent a vehicle they may not be required to make an account with us
* Rental
  + Pickup Time and return Time are left null as those don't need to be decided when you place a rental reservation
  + As referenced in the not null section the return bid is able to be left null as if it is null we can have it that it auto returns to the pickup bid

Other constraints

* Cannot rent the same car out at the same time (unimplemented)
* Cannot rent out a uninsured car (unimplemented)
* Cannot allow an unlicenced or uninsured customer to rent a car (unimplemented)
* Return date must be after pickup date
* Vehicle must be picked up from home branch
  + pickupBID must = vehicle BID
* Need a birthday >= 18
* Rental cannot be in the past? (arguable)
* Cartype dailyrate<weeklyrate<monthlyrate (unimplemented)
* Basic input validation: ints like year and seats cannot be negative, name cannot be empty string, color must not be castable as decimal, etc (mostly unimplemented)