1. First set of Tables

The following is a list of the tables obtained from the ERD-to-Tables process described in the notes. Primary keys are underlined. You could have chosen to complete this part of the deliverable in one of two ways: (1) Merge tables with key constraints (as discussed in lecture) or (2) Create a table for each entity and relationship in the diagram.

Branch (location, city)

- Represents: entity set *Branch*.
- Constraints: none

VehicleType(vtname, features, wrate, drate, hrate, wirate, dirate, hirate, krate))

- Represents: entity set VehicleType
- Constraints: none

If you chose to complete this deliverable by using option 2 (i.e., a table for each entity and relationship), you would have a table for:

- The VKeeps relationship
- The V_is_of relationship
- The Vehicle relation below would not have foreign keys

Vehicle (vid, vlicense, make, model, year, color, odometer, status, vtname, location, city)

- Represents: entity set Vehicle and relationship sets Vkeeps and V_is_of
- Foreign Keys:
  - (location, city) references Branch
  - (vtname) references VehicleType
- Other constraints:
  - status can take the values (for_ rent, for_sale)
    this will be used to define the ForSale and ForRent subsets.

- In this case the tables for the entity sets ForSale and ForRent will be defined as two VIEWS:

ForSale (vid, vlicense, make, model, year, color, odometer, status, vtname, location, city)
Includes all vehicles whose status is "for_sale")

ForRent (vid, vlicense, make, model, year, color, odometer, status, vtname, location, city)
Includes all vehicles whose status is "for_rent")

Alternatively: we could replace table Vehicle with the following two tables:

ForSale (vid, vlicense, make, model, year, color, odometer, status, vtname, location, city)

- Represents the entity set ForSale
- status is 'available' or 'not_available'

ForRent (vid, vlicense, make, model, year, color, odometer, status, vtname, location, city)

- Represents the entity set ForRent
- status is 'available' or 'not_available'

NOTE. In the given diagram, the Vehicle and the Equipment Entity sets have the same attribute "id". This can be confusing. In our tables, we have changed the names to "vid" and "eid" respectively. This is not required.

EquipType(etname, drate, hrate)

- Represents: entity set EquipType
- Constraints: none

If you chose to complete this deliverable by using option 2 (i.e., a table for each entity and relationship), you would have a table for:

- The Ekeeps relationship
- The E_is_of relationship
- The Equipment relation below would not have foreign keys

Equipment (eid, etname, status, location, city)

- Represents: entity set Equipment and relationship sets Ekeeps and E_is_of
- Foreign Keys:
  - (location, city) references Branch
  - (etname) references EquipType
- Other constraints:
  - status can take the values (available, rented, not_available)

Is_for(etname, vtname) (or we can give it a better name like *EforV*(etname, vtname))

- Represents: relationships set *Is_for*
- Foreign Keys:
  - (etname) references EquipType
  - (vtname) references VehicleType
- Other constraints: none

Customer (cellphone, name, address, dlicense)

- Represents the entity set Customer
- Additional constraints: none

ClubMember (cellphone, points, fees)

- Represents: entity set ClubMember
- Foreign Keys:
  - (cellphone) references Customer
- Additional constraints: none

If you chose to complete this deliverable by using option 2 (i.e., a table for each entity and relationship), you would have a table for:

- The Is_for_V relationship
- The Makes relationship
- The Made_for relationship
- The Reservation relation below would not have foreign keys

Reservation (confNo, vtname, cellphone, fromDate, fromTime, toDate, toTime)

- Represents: the entity set Reservation and the relationship sets Is_forV, Makes, Made_for
- Foreign Keys:
  - (vtname) references VehicleType
  - (cellphone) references Customer
  - (fromDate, fromTime, toDate, toTime) references TimePeriod
- Additional constraints: none

Reserve_Includes (confNo, etname)

- Represents: the relationship set Is_forE
- Foreign Keys:
  - (confNo) references Reservation
  - (etname) references EquipType
- Additional constraints: none

If you chose to complete this deliverable by using option 2 (i.e., a table for each entity and relationship), you would have a table for:

- The RecordRent relationship
- V_is_of relationship
- The Rent relation below would not have foreign keys

Rent(rid, vid, cellphone, fromDate, fromTime, toDate, toTime, odometer, cardName, cardNo, ExpDate, confNo)

- Represents: the entity set Rent and the relationship sets RecordRent and the relationship sets IsRented, Signs, Rented_for, and Rent-Reserve
- Foreign Keys:
  - (vid) references ForRent
  - (cellphone) references Customer
  - (fromDate, fromTime, toDate, toTime) references TimePeriod
  - (confNo) references Reservation ; it can be null
- Additional constraints: none

Rent_Includes (rid, eid)

- Represents: the relationship set Includes
- Foreign Keys:
  - (rid) references Rent
  - (eid) references Equipment
- Additional constraints: none

Return(rid, date, time, odometer, fulltank, value)

- Represents: the weak entity set Return and its defining relationship set Return-Rent
- Foreign Keys:
  - (rid) references Rent
- Additional constraints: none

[You can also combine this Return relation with the Rent relation.]

TimePeriod(fromDate, fromTime, toDate, toTime)

- Represents: the entity set TimePeriod
- Additional constraints: none

[You do not need to have this relation. This information could have also been captured as attributes for Rental.]

2. Functional Dependencies

Instead of renaming the attributes that appear in most than one table by putting the table name in front, we list the functional dependencies for each table separately.

Branch:

- none

VehicleType:

- vtname --> features, wrqte, drate, hrate, wirate, dirate, hirate, krate (key)

Vehicle:

- vid --> vlicense, make, model, year, color, odometer, status, vtname, location, city (key)

[Someone familiar with cars may also the following FD:

make, model, year --> vtname
(i.e. the make, model and year of a vehicle determines its category)

We did not deduct or give any marks for this FD.]

EquipType:

- etname -->drate, hrate (key)

Equipment:

- eid --> etname, status, location, city (key)

Is_for:

- none

Customer

- cellphone --> name, address, dlicense (key)

ClubMemeber:

- cellphone --> points, fees (key)

Reservation:

- confNo --> vtname, cellphone, fromDate, fromTime, toDate, toTime (key)

Reserve_Includes:

- none

Rent

- rid --> vid, cellphone, fromDate, fromTime, toDate, toTime, odometer, cardName, cardNo, ExpDate, confNo (key)
- cardNo --> ExpDate

Rent_Includes:

- none

Return

- rid --> date, time, odometer, fulltank, value (key)

TimePeriod:

- none

3. Additional Constraints

a) IC1: The plate license (i.e., vlicense) of a vehicle uniquely identifies it.

   vlicencse --> vid (and by transitivity, all other attributes in the relation)

b) IC2: The odometer value at the time of return must not be smaller than the odometer value at the beginning of the rental.

   Not in the ER model.

c) IC3: Each customer's driver's license (dlicense) is owned by a single customer. In order to make a reservation or to rent a car, a customer has to have a driver's license.

   dlicense --> cellphone (and by transitivity, all other attributes in the relation)

d) IC4: The credit card number determines what type of card it is (e.g., Visa, Mastercard, etc.).

   Note: If you interpreted cardName as the name of the cardholder, this FD would still apply.

   cardNo → cardName

4. Tables after Normalization

All tables except for *Rent* are in BCNF

- In all the functional dependencies marked by "(key)" the determinants are the superkeys of the tables.
- In *Vehicle*, the additional dependency indicates that vlicencse is another superkey for this table; the table is still in BCNF.
- in *Customer*, the additional dependency indicates that dlicencse is another superkey for this table; the table is still in BCNF.

Taking into account the cardNo -> cardName , we decompose the Rent table in the following tables:

- *Rent1*(rid, vid, phone, fromDate, fromTime, toDate, toTime, odometer, cardNo, ExpDate, confNo)
- *Card*(cardNo, cardName)

If we also take into account the dependency cardNo --> ExpDate, then Rent1 should be further decomposed into:

- *Rent2*(rid, vid, phone, fromDate, fromTime, toDate, toTime, odometer, cardNo, ExpDate, confNo)
- *CardExpire*(cardNo, ExpDate)

  [Ideally, you would merge tables to Card(cardNo, CardName, ExpDate). We did not give or deduct any marks regardless of whether or not this merge was performed.]