Arda Aşık 23653
Barış Batuhan Topal 24071
İbrahim Buğra Demir 23744

27.03.2018

# CS306 – PROJECT STEP 2

After correcting our ER diagram regarding the feedback of step 1, in this assignment all the relations will be investigated and checked one by one, if they satisfy 3NF or BSNF:

**1st Relation:**

```
CREATE TABLE car (
    cid CHAR(7),
    photo VARBINARY(8000),
    color VARCHAR(20),
    geartype CHAR,
    price DOUBLE(10, 2),
    availability CHAR,
    model VARCHAR(20),
    PRIMARY KEY(cid)
);
```

In the first relation we examine, cid is unique for every user and for each distinct cid there is a unique combination of all the attributes excluding availability. About availability, nothing in these attributes build a functional dependence:

- cid → photo color geartype price model

To be able to have a specific price on the vehicle users will rent, the model geartype and the color of the car should be known. Considering all these attributes, a specific price can be set on the car:

- model geartype color → price

Every cars' appearance is determined by its model and color. Regarding to this, model and the color of the car has a functional dependency on the photo.

- model color → photo

With respect to those 3 different functional dependencies, if we analyse the closure of the combination cid model geartype and color (which is: cid photo color geartype price model) it contains all the attributes except availability. While availability has no dependency with the others, we can consider it separately and for the remaining it can be said that cid model geartype and color together form a super key. However, if the second and third FDs are evaluated, the left-hand sides of the dependencies are not this super key. Moreover, their right sides are also not a subset of the super key. That's why, these dependencies are violating both 3NF and BCNF formation. To be able to reduce redundancy we can separate these attributes into 3 different relations. The first relation can be R1 = {cid, color, model, geartype, availability}, the second one can be R2 = {model, geartype, color, price} and lastly the third one is R3 = {model, color, photo}. After that separation the super keys of every relation are:

- R1: cid
- R2: model geartype color
- R3: model color

For every single relation, there is one FD and the left sides of every FDs are super keys. That indicates, these 3 relations are in BCNF.

- Dependency preserving decomposition check:

If we investigate the second and third relations, still the dependencies below hold:

- model geartype color → price
- model color → photo holds for these tables.

For R1, in this case the functional dependency is cid → color geartype model. But if we combine all these 3 relations, with the help of other two dependencies we can reach to the initial form of the first dependency: cid → photo color geartype price model.

To sum up, all the FDs are preserved if this type of decomposition takes place.

- Lossless join check:

If we take the intersections of all the tables:

| | | |
|---|---|---|
| - R1 with R2: | model geartype color | → key of R2 |
| - R2 with R3: | model color | → key of R3 |
| - R2 with R3: | model color | → key of R3 |
| - R1 with R2 with R3: | model color | → key of R3 |

While all the common attributes are key for the relations, this decomposition is lossless.

**<u>2nd Relation:</u>**

```
CREATE TABLE hirer (
        hid INTEGER(11),
        haccount VARCHAR(22),
        hname VARCHAR(20)
        PRIMARY KEY(hid)
);
```

In that relation, every hid (which is the Personal ID) is unique for every user. For any single hid there is a unique bank account of hirer (haccount) and username (hname) combination.

However, if we speak about haccount or hname, same bank accounts can be used for different users or different users can have same names. Therefore, the only dependency we have is:

- hid → haccount hname

In that case, if the closure of hid is written, then all the attributes hid, haccount and hname are included. That's why, hid is the candidate key (also for this relation a super key). Since we have only one relation and the left side of it is a super key, we can say that this relation is in BCNF type.

**<u>3rd Relation:</u>**

```
CREATE TABLE contract (
    hid INTEGER(11),
    tid INTEGER(11),
    contid INTEGER(10),
    begindate DATE,
    enddate DATE,
    PRIMARY KEY (contid, hid, tid),
    FOREIGN KEY(hid) REFERENCES hirer ON DELETE CASCADE,
    FOREIGN KEY(tid) REFERENCES tenant ON DELETE CASCADE
);
```

Contid is a specific integer number that is assigned uniquely to each contract. The combination of it with the ID values of hirer and the tenant (hid and tid), we can specify the begin and end dates of the rental process. Thus, contid hid and tid together build a FD to the enddate and begindate:

- contid hid tid → begindate enddate

However, for every specific contid, the IDs of both tenant and hirer can be uniquely determined:

- contid → hid tid

Considering these 2 relationships, contid tid and hid can be called together as a super key. Because if the closure of them is written (which is: contid, hid, tid, begindate, enddate), it includes all the attributes in that relation. So, for the first dependency, the left-hand side includes the super key and that's why there is no violation of BCNF. Controversially, the second FD does not contain that super key on the left-hand side. But, if we look at the right-hand side of that dependency, hid and tid are a part of the super key. That indicates, this relation has no violation of 3NF.

The relation is in 3NF form.

Arda Aşık 23653
Barış Batuhan Topal 24071
İbrahim Buğra Demir 23744

27.03.2018

**4th Relation:**

```
CREATE TABLE tenant(
   age INTEGER,
   tid CHAR(11),
   taccount VARCHAR(22),
   tname VARCHAR(20),
   PRIMARY KEY(tid)
);
```

For the relation that defines the user who wants a rental car, tid value (which is the licence number of that person) is unique and has the power to identify all the other attributes age, name of the tenant (tname) and tenant's bank account (taccount):

- tid → age taccount tname

In that relation there is only one functional dependency and if we examine the closure of tid, it contains all the attributes age, tid, tname, taccount. Because of that tid can be identified as a candidate key. So, the left-hand side of the FD only contains a candidate key (which is only the minimal form of the super key) and that shows us that this relationship is in BCNF.

**For the remaining relations;**

All the attributes are taken from other relations as foreign key and they are defined in their parents as primary keys. So, they all are unique and have no dependency on each other. Because of that, we can directly say that they are in BCNF form. You can find the list of those relations in the following pages:

**5th Relation:**

```
CREATE TABLE owns(
   cid CHAR(7),
   hid INTEGER(11),
   PRIMARY KEY(cid, hid),
   FOREIGN KEY(cid) REFERENCES car(cid) ON DELETE CASCADE,
   FOREIGN KEY(hid) REFERENCES hirer(hid) ON DELETE CASCADE
);
```

Arda Aşık 23653
Barış Batuhan Topal 24071
İbrahim Buğra Demir 23744

27.03.2018

**6th Relation:**

```
CREATE TABLE agree(
   tid INTEGER(11),
   hid INTEGER(11),
   contid INTEGER(10),
   PRIMARY KEY(tid, hid, contid),
FOREIGN KEY(tid) REFERENCES tenant(tid) ON DELETE CASCADE,
   FOREIGN KEY(hid) REFERENCES hirer(hid) ON DELETE CASCADE,
   FOREIGN KEY(contid) REFERENCES contract(contid) ON DELETE CASCADE
);
```

**7th Relation:**

```
CREATE TABLE charges [
         tid INTEGER(11),
         hid INTEGER(11),
         PRIMARY KEY(hid, tid),
         FOREIGN KEY(hid) REFERENCES hirer(hid) ON DELETE CASCADE,
         FOREIGN KEY(tid) REFERENCES tenant(tid) ON DELETE CASCADE
);
```

**8th Relation:**

```
CREATE TABLE hires(
   cid VARCHAR(7),
   tid INTEGER(11),
   PRIMARY KEY(cid, tid),
   FOREIGN KEY(cid) REFERENCES car(cid) ON DELETE CASCADE,
   FOREIGN KEY(tid) REFERENCES tenant(tid) ON DELETE CASCADE
);
```

Arda Aşık 23653
Barış Batuhan Topal 24071
İbrahim Buğra Demir 23744

27.03.2018

# Our Updated ER Diagram without BCNF checks: