Normal Forms Discussion

MainePad Finder

By Jeffrey Fosgate, Yunlong Li, Ashley Pike and Sophia Priola

Highest Normal Form:

4NF

We found the highest normal form our database satisfies is 4NF. All of our entities, `REVIEW`, `PROPERTY`, `ADDRESS`, `NOTIFICATION`, `USER`, `RENTER`, `RENTER_PREEFERENCES`, `RENTER_SETTINGS`, `PREFERENCES`, `LANDLORD` and `MESSAGE`, were examined to ensure that each satisfies the properties of 4NF.

The entity, `MESSAGE`, is shown given the relation schema

`MESSAGE(MSG_ID, SENDER_ID, RECIPIENT_ID, SENT_TIMESTAMP, MESSAGE_TEXT, IS_READ)`

with the functional dependencies

`MSG_ID → SENDER_ID, RECIPIENT_ID, SENT_TIMESTAMP, MESSAGE_TEXT, IS_READ`

`MSG_ID` would be a candidate key in this instance since it can determine every other attribute in the relation. Since α (`MSG_ID`) in this relation is a primary key, this relation satisfies BCNF. This relation also satisfies 4NF because there are no non-trivial multivalued dependencies. For a fixed `MSG_ID` each attribute (`RECIPIENT_ID`, `SENDER_ID`, `SENT_TIMESTAMP`, `MESSAGE_TEXT`, `IS_READ`) has a single value. Therefore, any multi-valued dependency with determinant `MSG_ID` is trivial and no two independent lists under the same owner are stored in this table.

The same can be done for `USER` since it has the primary key of `USER_ID` that can determine the rest of the attributes in the relation, `USERNAME`, `PASSWORD`, `EMAIL`, `PHONE_NUMBER`, `GENDER`, `DESC`, `PICTURE_URL`, `NAME`, `BIRTH_DATE`. As stated previously, `USER_ID` is trivial and there are no two independent lists that are stored in this table. Meaning once again, BCNF and 4NF are satisfied.

As we went into this project, we took a proactive stance with regards to creating entities only with the information that could be determined by an ID as the primary key and not by any other attribute. When we saw tables such as `NOTIFICATION` and `MESSAGE` that could have been further decomposed, we took initiation to create two separate tables immediately.

Therefore, we did not have to split our entities into sub-entities at any point within our discussion of what the highest normal form is that our database satisfies.