Candidate: 44445

Schema 1:

1. Candidate key(s):

   (stuId, unit)

2. The key and non-key attributes:

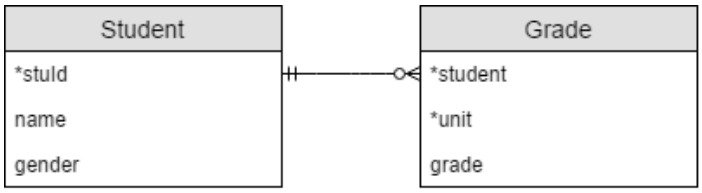
   Key attributes: stuId, unit
   Non-key attributes: name, gender, grade

3. **1NF:**
   Basically, I think the table in Schema 1 is in 1NF. In 1FN, each column can contain only one value (Atomic Value) and the column with the same characteristic cannot repeat.

   2NF:
   However, this table not fulfills 2NF. As every non-key attributes must have "fully functional dependency" on the primary key, like "gender" and "name" have no relationships with "unit". Here is the proof which "partial functional dependency" might cause a series of problems. Firstly, you cannot add a student record who does not have any units as the "unit" cannot be NULL. Secondly, you cannot delete a grade data, because other information such as "unit" and "stuId" will be destroyed as well. Thirdly, there will have too much information that has to be synchronized if you revise one key. For instance, if student 101 changes his name, you will just have to change three records once a time.

   3NF & BCNF:
   It is evident that this table does not meet 2NF. Therefore it is not in 3NF or BCNF either.

4. I consider the best of BCNF is to ensure that all attributes of each table are functionally dependent on their candidate keys. Following is the update ER diagram, which shows that every attribute is relied on their primary keys (*stuId => name, *stuId => gender, *stuId *unit => grade). For now, this table will spare more efforts when updating a student record. Thus meeting BCNF.



Schema 2:

1. Candidate key(s):

   city

2. The key and non-key attributes:

   Key attribute: city
   Non-key attributes: country, pop, co_pop, capital

3. **1NF:**
   I do believe that this table satisfies 1NF because there is no attribute which contains more than one value.

   2NF:
   According to Schema 2, I reckon that this table is in 2NF. On account of the fact that user can add or remove the city without affecting other attributes.

   3NF:
   However, the “co_pop” only has dependency on “country”, so they have a “transitive dependency”. Although the user can insert or delete a record freely, still if “co_pop” changes, there will be loads of records needed to be updated at the same time. It shows that we need to eliminate transitive dependency to meet 3NF, so we have to make sure that there is no completely functional dependency between any of non-key attributes. As a result, the “country” and “co_pop” should be split from the original table. Finally, the user can update country and city freely.

4. As for BCNF, if the primary key is a single attribute, I think there is no need to concern about BCNF. As we can follow the 3NF diagram in the following picture.