Question Two – Normalizing

**Data stored in one single table**

Application-No + StudentID+ StudentName + Street + State + Zip-Code + ApplicYear + Reference-Name + RefInstitution + Reference-Statement + Prior-School-Id + PriorSchool-Addr + GPA

**Notes**

* Applications are only unique for a particular year, because it resets every year. I think the applicYear would be important as an id, as it would link to the application being unique that year.
* StudentID is unique and would be a primary key because it doesn’t change.
* Students may have more than one address because they might move. We might need to make a studentAddre table and put yearofMove and studentID
* There can’t be applications with different students, so application number/reference name is unique
* If I have reference name and ref institution together, it will be unique, not separate though
* Prior school id -> universityname, university location
* Prior school list may change across different application of the year, so it is better to have a prior school list table
* One gpa for from any school
* Application have one or more referees, so should do a table for referees
* Reference statement is different for different applications, so it the reference statement table, have an applicationID to identify each application

**When a student wants to apply to a graduate school:**

* They must have an application number and fill in all the required details in the table
* None can have null values
* Application No and Student ID are unique and nothing can be done without those, so they are automatically primary keys

**1NF**

* Remove repeating groups
* Data should be in most reduced form
* Data should be uniquely identified by primary key

**Application No and StudentID are candidate keys because they are both important. We need a student and an application to apply for a graduate programme, so both are needed to start off with.**

Application (App\_No, ApplicYear)

Student (StudentID, StudentName, Zip-Code, Reference-Name, RefInstitution, Reference-Statement, Prior-School-Id, Prior-School-Addr, GPA)

I want to remove State, Zip-Code and put it unto one data and call it address because it is repeating data, that can be grouped together and will make the table more reduced. I can take out the state and street because the zipcode is already unique enough for the address, no place has the same zipcode. In the end I think it is a better choice to remove street and state and use zip code only.

**2NF**

* All data should depend on the primary key
* Need to look at the functional dependencies and make decision

I looked at App\_No as the functional dependencies to make decisions about because all the information comes with the application and relies on the application being made to get the rest of the information, so the application is unique. I chose this over the studentID because the student doesn’t get a student ID, until they submit, they’re first application. That when the studentID becomes unique, so it is dependent on the application.

I thought it was important to move the references rather than the prior school information because the prior school goes with the student information and the references seem like an external thing from the personal detail of the student.

1. studentID -> App\_No , ApplicYear
2. studentID -> studentID, studentName, Prior-School-Id, Prior-School-Addr, GPA
3. studentID -> Reference-Name, RefInstitution, Reference-Statement

Student (StudentID, StudentName, Zip-Code, Prior-School-Id, Prior-School-Addr, GPA)

Application (StudentID, App\_No, ApplicYear)

References (StudentID, Reference-Name, RefInstitution, Reference-Statement)

**3NF**

* Remove transitive dependencies

1. Prior-school-Id -> Prior-School-Addr and GPA
2. Zip-Code-> Street, state

I put reference statement as unique because students can have the same reference name and the institution, they from, but the reference statement will always be different no matter the application or student the reference is writing an application for.

I decided to put back the state, street, so that I can put zip code as unique and put it into an address table.

Student (StudentID, StudentName, Zip-Code, Prior-School-Id)

Application (StudentID, App\_No, ApplicYear)

References (StudentID, App\_No, Reference-Name, RefInstitution, Reference-Statement)

School (Prior-School-Id, Prior-School-Addr, GPA)

Student Address (Street, State, Zip-Code)

StudentID is needed as a foreign key in application because each application must have a studentID and application number is a primary key for application table because the number is unique to that application.

StudentID is in references because each student must have atleast one reference and each reference have atleast one application, so App\_No connects references to the application table. Also, a referee can write application for different students, so it makes sense to include the studentID in the table. Both studentID and App\_No are foreign keys and reference-statement is the primary key because all application must have different reference statement even if its by the same referee.

**Questions**

* would it be better to do studentID instead of App\_No for the dependencies to look at? I think I should use studentID because it’s unique, but the studentID isn’t assigned until one application is submitted. Also, a student can have many applications no, but can only have one student ID, so it is better to do student ID as the dependency.
* Am I allowed remove things from the table, for example, I want to remove state and street and only leave the zip code? Is that allowed?
* Is it okay to not have any 1NF? Because I couldn’t find a data to reduce. I wanted to do it reduce the street and state because zip code is unique, but when doing the 3NF, I thought it would be better to make an address table and put those data in the table. I’m not sure which would be the better option, do it for 1NF or 3NF.