Jack Dong, Alex Seletskiy, Ian Luck, Isabel Tomb

CSCI4797

Project - Task 3

*4707 Synopsis*

Our database design was able to pass a good amount of the requirements that we have tested. There were some requirements that did not pass though and would require design changes or additional features implemented like assertions, checks or triggers. Overall, our database held up well against the requirements and we are able to track a lot of the information that is needed or join back to get relevant information together.

The first design choice that caused some requirements to fail was with patients and their age. In our design, we forgot to include attributes like the date and age of a patient. This made it impossible for us to track if they actually needed a parent/guardian added as well or not. Simply adding this would have not been enough though, cause we would have needed to implement more to get it to work. If we would have had the date of birth included, we would then have additional assertions to check whether that patient needs a parent or not before insertion. If below, check if a Parent/Guardian was inserted, along with an entry to PARENT\_GUARDIAN\_FOR table to connect the two together. If the patient is over 18, have an assertion to make sure that no parent/guardian is associated with them as well. In addition, our design did not implement the PARENT\_GUARDIAN\_FOR relation properly. Instead of making both the attributes in that table the primary key, only the patients PID was made the primary key. This made it so that a patient could not have more than one parent. A parent could have as many patients assigned to them as they wanted as long that patient was not assigned to another parent as well. This would be a simple fix in our design by just adding the PGID from PARENT\_GUARDIAN into the primary key of PARENT\_GUARDIAN\_FOR table. For patients that turn 18, a trigger would have to be set up to detect those and delete the parents from both PARENT\_GUARDIANS and PARENT\_GUARDIANS\_FOR.

The second error in our design was with tracking insurance with the patient that had them. In our design, we did not include the start and end date in the INSURANCE\_FOR table. This made it impossible to be able to track current insurance and history of insurance. We made the InsNum and PID both primary keys, so a patient can have a track of which insurance they’ve had. Though we would not be able to tell which is the current and what is outdated. Adding both start date and end date, we would be able to get the current insurance information by finding the INSURANCE\_FOR entry with a NULL in the end date attribute. The history will be tracked by those to see what insurance the patient had at the time of the visit.

Another error found in our design was the lack of limitations on the values that can be entered for the pay type of an employee. The only possible values should be Hourly or Salary. Currently, the way we have the table set up, any value can be entered in this column. This could be fixed by adding a check for the column PayType in table EMPLOYEE to only allow for the values Hourly or Salary to be entered.

Our design choices for DIAGNOSIS\_FOR, VISIT, PATIENT, and SERVICE\_PROVIDER allowed patients to have multiple diagnoses and only allowed service providers to make diagnoses. The primary/foreign keys for the first three tables are visit IDs and patient IDs, which allowed for this feature to occur. Each diagnosis will need to be entered into DIAGNOSIS\_FOR each time for any queries to call all of them for each patient. For service providers, their primary/foriegn key has a one to many relationship to VISIT such that service providers and only service providers can make diagnoses on patients -which was tested in test7. We also allow patients to have more than one initial assessments total and limited to one per visit, because the primary key of VISIT corresponds to INITIAL\_ASSESSMENT’s foreign key. Then we tested to see if PATIENTS can have multiple VISITs, and tested they can. There is no key relation between patients and visits. We tested that intake clerks can document insurance information and copays of patients. INTAKE\_CLERK and INITIAL\_ASSESSMENT connect through INTAKE\_CLERK’s primary key CEmpID, though since NURSE is also related to INITIAL\_ASSESSMENT this means NURSE employees may also have access to INSURANCE\_FOR information. INITIAL\_ASSESSMENT relates to VISIT, which relates back to PATIENT which has INSURANCE\_FOR information.

The design decisions for NURSE, INITIAL\_ASSESSMENT, PATIENT, and VISIT made the initial assessments have access to the nurse’s information. In addition, a nurse was also able to perform multiple assessments for multiple patients and an initial assessment could have only been completed by one nurse. These features were tested in tests, 12, 13, and 14. The reason that these features were able to run smoothly were because INITIAL\_ASSESSMENT has the foreign keys NEmpID and VID that point to VISIT and NURSE and they were properly set up inside INITIAL\_ASSESSMENT. A NURSE was made to only be able to complete one INITIAL\_ASSESSMENT per PATIENT and multiple initial assessments can be performed on one PATIENT as long as it is a different VISIT. However, there is a design flaw in which the foreign key VID inside of INITIAL\_ASSESSMENT was not initially set as unique, and because it is not, multiple visits can be tied to an INITIAL\_ASSESSMENT. The design of PATIENT, VISIT, and INITIAL\_ASSESSMENT having the the primary keys: VID, PID, and AID alongside the foreign key found within VISIT, PID pointing back to PATIENT allows data to be gathered for a VISIT having information drawn from an INITIAL\_ASSESSMENT that was done during a certain VISIT. This design allows for the vitals of a PATIENT to be displayed using queries after a VISIT. However, there is a slight design flaw in the database because the column PID inside VISIT was not made to be a foreign key, but should be one. Overall the design of NURSE, INITIAL\_ASSESSMENT, PATIENT, and VISIT made tests 12, 13, 14, and 15 run smoothly showing that each query could retrieve the required information.

Our system does only allow one Service Provider in a visit for a patient, that was correctly implemented because of our design. We have the Service Provider table separate from the Employee, but foreign keyed to it so that only Service Providers and not Nurses or Intake Clerks would be able to be assigned to a visit.

By creating a table separate from VISIT called TREATMENT\_FOR this allows for there to be multiple tests or procedures requested by a doctor per visit. Similarly, by having the separate TREATMENT\_FOR table it allowed for there to be no tests or procedures for a patient during their visit. The cardinality ratio for service providers on visits is correct in that a service provider can work on many visits. Additionally the cardinality ratio of Intake Clerk on an initial assessment is correct because an intake clerk can log many initial assessments. The following primary keys are successful in linking our tables: VID, NEmpID, PID and SPEmpID. VID links the INITIAL\_ASSESSMENT and VISIT tables. NempID links the INITIAL\_ASSESSMENT and NURSE tables. PID links the VISIT and PATIENT tables. Lastly, SPEmpID links the SERVICE\_PROVIDER and VISIT tables.