My Vacation & Timeoff Tracker application will leverage a Relational Database to store the necessary data that will be used for CRUD operations and also business logic processing. I chose a RDBMS model because of the many primary and foreign key relationships that I want to ensure are enforced using database CONSTRAINTs. I will need to explicitly insert data into the admin table for initial use; otherwise, nobody will be able to access the admin menu page within my application.

Below is the Entity-Relationship diagram for the tables in my schema, and I will leverage PostgreSQL to implement this database model:

\*\*The yellow highlighted items below indicate changes to the original diagram to better use PRIMARY KEY GUIDs, and these will be set as INT SERIAL in order to receive the next unique value from the database.

Several cylinder cylinders with text

Description automatically generated  
  
A cylinder with words on it

Description automatically generatedA white cylinder with black text

Description automatically generated A diagram of a couple cylinder

Description automatically generated



**student**

The purpose of this table is to store student information along with the student’s starting time-off balances. Starting time-off balances will be determined by the registration process and the student’s Master’s Degree start date. Business logic will then calculate how many vacation, sick, and personal days should be awarded based upon a tenure range. For example, I started by Master’s Degree journey on January 16, 2023, so my allotments will be awarded based upon 2 years of student tenure.

studentID is the only primary key as there can only be one student entry on this table.

**timeoff**

The purpose of this table is to store student’s timeoff requests that are submitted via the Vacation & Timeoff Tracker application. There will be no CRUD operations to deduct used hours from the student table’s original allotments. Business logic will determine available balances dynamically.

studentID, timeoffDate, and timeoffCd are the three primary key columns because there cannot be duplication of these entries. It is possible to take ½ day vacation and ½ day personal leave in a single day, and this will be allowed because timeoffCd will be different.

The student table relationship to timeoff is one-to-many since there can only be one student record, but a student can have many timeoff requests.

**timeoffCd**

This is basically a domain table that stores the timeoffCd and its corresponding description. This table will be used to translate ‘V’ into ‘Vacation’, ‘S’ into ‘Sick’, and ‘P’, into ‘Personal’ using a CASE…WHEN…THEN statement for better readability.

The timeoff table relationship to the timeoffCd table is one-to-one since there can only be one description for a timeoffCd value.

**approvals**

The purpose of this table is to store the key value pair student:manager relationships in the event the student submits a timeoff request that exceeds his/her balance and manager approval is required. Business logic will SUM() the timeoff requests for a student, and if that value exceeds the student’s original allocation, the approval workflow will be triggered.

The approvals table is one-to-many to the student table since there can only be one student record, but a student can have many timeoff request approvals.

**manager**

The purpose of this table is to store the manager’s e.g. professor’s information, to be used in business logic and reporting for better readability. Since a college course can have a professor and a professor’s aide, both of which are available to the student, I elected to use an activeInd to record past and present student:manager relationships.

The manager table is one-to-many to the student table since there can only be one student record, but a student can have more than one active or inactive manager that timeoff requests in need of approval can be send to.

The manager table is also a foreign key relationship to the timeoff table. I will use this relationship for reporting, charting, and graphing to display the manager’s information for better readability.

**preferences**

The purpose of this table is to store the student’s affinity towards vacation spots, interests, and activities, which will be used by business logic to recommend and/or solicit completed vacation destinations from other students. For example, if a student enjoys domestic travel (in the USA) and opts into the recommendation business logic, he/she will receive domestic travel suggestions from other students who have completed and enjoyed their vacations. And also to stay away from vacations that were not enjoyable.

The preferences table is one-to-one to the student table since there can only be one student record and only one set of preferences.

**admin**

The purpose of this table is to store admin userids that will be used to access admin menu pages within the Vacation & Timeoff Tracker application. This will be a stand-alone table that contains my Maryville userid and the professor’s.

**studentTrip**

The purpose of this table is to store evaluations of completed vacations by students in order to make recommendations to other students who have unused vacation and personal timeoff time. Sick leave will be excluded from the business logic to make these recommendations since students should not be using sick leave for vacations/trips. Students will rate their vacation on a scale of 1 (not enjoyable) to 5 (very enjoyable) so that business logic can determine which ones to recommend to students. I may even separate the recommendations into ‘take these trips’ and ‘don’t take these trips’ based upon the evaluations.

The studentTrip table is a one-to-many relationship to the student table since there can only be one student record, but a student can have many trips.

**tripPics**

The purpose of this table is to store any pictures uploaded by the student from his/her vacation.

The tripPics table is a many-to-many relationship to the studentTrip table since a student can have many vacations on file with many pictures. I may limit the upload utility to 5 pictures. This isn’t Walgreens or Facebook to just do a picture dump and fill up storage in my database.