We used two databases: PostgreSQL and SQLite. The metadata is stored in a simple PostgreSQL database, with the user\_id as a primary key. The client program has some basic login functionality and a SQLite database was used to store usernames and passwords as well as the user’s local emails and mailbox structure. Both are open source and were best for our project from the limited options that are available. PostgreSQL is good for concurrency and was used for our server to support concurrent connections and operations. SQLite is easy to set up and work with, so it’s an ideal solution for our client program. Digital Ocean virtual machine “droplet” was used for testing our server database. Digital Ocean droplets are inexpensive, reliable, and can be snapshotted for easy rollbacks should something go wrong during development. Our Digital Ocean runs on Ubuntu for easy terminal access over SSH.

PostgreSQL was used to manage the server database. Database ‘coredb’ was created to maintain the Adept mail server. The database coredb has three tables: users, mailboxes and emails. Currently the server is running on a command line prompt using sequel queries. The users table contains six attributes: user\_id, email, password, certificate, last\_login and isadmin. The user\_id attribute is the primary key for this table. This allows us to distinguish each person in the Adept server. The user\_id has a serial datatype so it auto increments as each user registers. The email attribute is unique for each user and lower case index was added to ignore case. The email attribute uses character varying datatype which limits the email to hundred and fifty characters. The password attribute also users character varying and limits the password of the user to hundred characters. Password-hash was supposed to be implemented, but was unable to due to time constraints. Certificate attribute is set to character varying with no limit it stores the SMPT certificate. The users table contains a last\_login attribute. A timestamp attribute with time zone datatype was used which allows the users to see when they last logged in. Unlike other database PostgreSQL allows us to implement date and time values very easily and efficiently. The last attribute in the users table was the isadmin attribute which is a boolean attribute, which flags to see if the user is an admin or not. All the attributes except for certificate are not NULL. The mailboxes table consists of three attributes: mailbox\_id, mailbox, and owner. The attribute mailbox\_id is the primary key and is being auto incremented. The attribute mailbox being the name of the mailbox such as inbox, spam, etc. The owner attribute of this table determines that different users could have the same mailbox name but it might belong to some other owner. Therefore, owner attribute is linked to user\_id in the users table as a foreign key. The mailbox\_id and owner are not NULL. The emails table consists of nine attributes: email\_id, owner, mailbox, date, to, from, subject, body and read. Where email\_id is the primary key and is auto incremented. The owner attribute here is also linked to the user\_id in the users table as a foreign key. The mailbox attribute is linked to mailbox attribute in the mailboxes table as a foreign key. Together the database keeps track of the user data by their user\_id, owner, and mailbox\_id attributes along with their written emails.

Figure 0.0: Relational diagram for the PostgreSQL database ‘coredb’

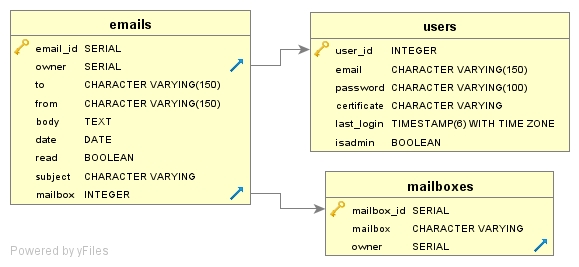


Figure 0.0 : Sample data for the users table

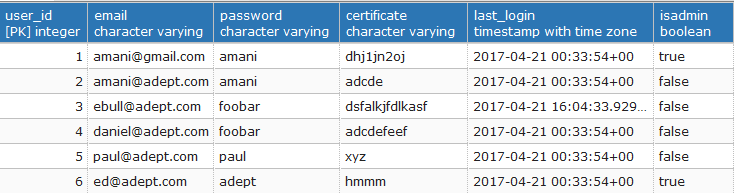


Figure 0.0 : Sample data for the mailboxes table

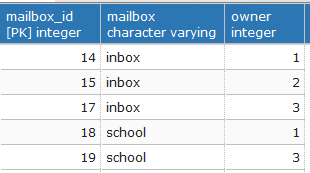
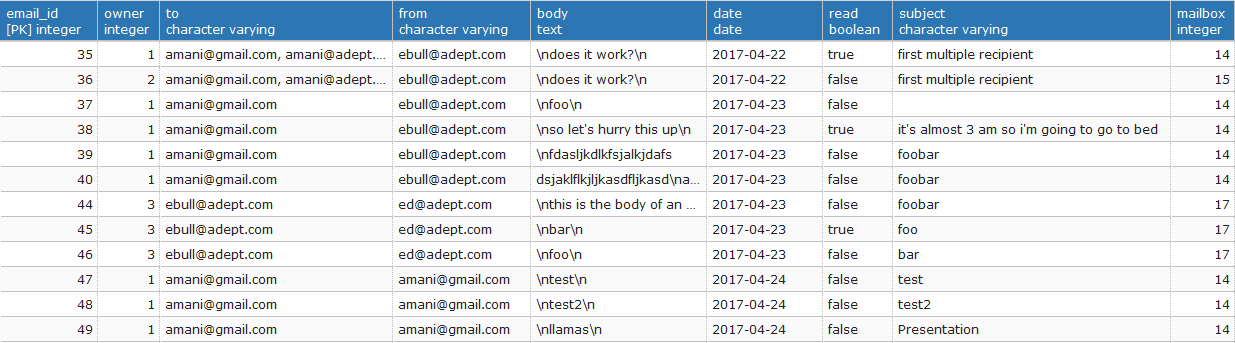


Figure 0.0 : Sample data for the emails table



SQLite will be used to manage the database. One database will be created to maintain four separate tables used for the Alliance Hotel System. The four separate tables will be as follows: Employee/User login, Hotel Rooms, Room Service Requests, and Customer Information. A graphical user interface will be created in order for the user to interact with the database. Employees will be able to access, add and process data stored within the database. The customers will have login privileges for the room service portion of the software. They will also have access to their room and billing information. The employee will have their own personal login information, access to the hotel services, and the customer's information. Manager will have the privilege to access and manipulate all data contained within the database. T