2020

COMS2003/2013 PROJECT

Arneev Singh & Kiara Gabriel

RUNTIME TERROR!

LECTRAC

**TABLE OF CONTENTS**

**PROBLEM STATEMENT1**

**EXTERNAL RESEARCH2**

Survey2

**DESCRIPTION OF PROCEDURES3**

**DATABASE DESIGN4**

Database Tables2

Business Rules2

Initial ERDs2

**Problem Statement**

We are currently facing a global pandemic and it seems as if everything has stopped, so in

order to stay relevant we need to adapt. Including the normality of attending university

campuses, which puts both students and lecturers in unfamiliar territory. Lectures are

unsure on how to provide an efficient online working environment and students are

finding it difficult to keep themselves organised.

Registered students do not have a platform where they are able to access all work material that they need to do and all work due date. Students need to scour multiple times a day for what needs to be done for each course via received emails, Sakai or Moodle. This creates problems for students who find it difficult to organise their time and will then miss due dates or try to complete tasks a few hours or minutes before it is due.

Our team has created a solution for this problem. We have designed a mobile app that allows students to access course related to-do lists so that they only need to go to one platform to find out all the work they need to do for each of their registered courses. The app also has a calendar feature that allow students to view any upcoming due dates for various course work.

Lecturers can access the app to add due dates to the calendar and add what work needs to be studied and completed to the course to-do list. Lecturers will also be able to broadcast messages to the students via the app and chat with students via the forum.

**External Research**

# **Survey**

1. Are you happy with the current online learning?
2. Do you miss due dates, laboratories or assignments because of the unorganized online working environment?
3. Do you wish that you had a single platform to get the due dates for all of your registered courses?
4. Have you created a work-related to-do list for yourself?
5. Do you think a to-do list would be beneficial to you?
6. Would you like a more organised system to check what work needs to be done rather than scrolling through emails and Sakai tabs?

**Description of Procedures**

1. Login System

This **login** is a security measure so that only students and lecturers who are registered at WITS university can access the app. The user needs to enter a student/ lecturer ID and a password, in which both inputs need to be correct to be allow access.

* Upon **registration** to the app, the user (student/lecturer) must provide a WITS ID, first name, last name, email address, nickname, and password. These personal details are saved to their respective databases (STUDENT/ LECTURER). The password field that needs to be filled in is, by default, the password that the university assigned to users.
* **Forgot password** allows for users to reset their passwords back to the default (university assigned passwords). If a user has forgotten his/her wits password, then the university needs to be contacted.

1. Calendar

The calendar feature allows for students to see all projects, labs and homework assignments and quizzes that are due for ALL the courses that they are registered for. It also allows lecturers to see when tasks they have set for themselves and students (in their teaching courses) are due. The user will be able to see which dates have a task due and when clicked, all tasks due for that date are displayed. The user can edit and delete tasks from the calendar which will reflect in the to-do list.

1. To-do List:

The to-do list is a list of tasks that tracks all tasks that the user needs to do. There are local tasks (saved to the user’s local database) and online tasks (saved to the LecTrac database). The tasks have a name, due date, due time and course.

* Both user types can **add** tasks to their to-do lists. A student can only add to his/her own (local) to-do list, whereas a lecturer has two choices: save to his/her own (local) to do list or to post an online task for all student in that course to see (course code must be chosen).
* Both users can **edit** the tasks in his/her local database. Only a lecturer can edit tasks that have been posted online.
* Both users can delete their own local tasks. If a student deletes an online task, a Boolean attribute, isDeleted, will be set to true and the student will not see the task. However, if a lecturer deletes an online task, it deletes from the database and all to-do lists.

1. Message Board

The message posts are broadcasted messages that a lecturer can add to their respective courses as announcements for students to read. Students can only read messages whereas lecturers can add and delete messages.

1. Test Marks

This feature allows students to access all test marks (course assessments) from their registered courses in one location. Students can view their previous results, the date the test/assessment was written/due and the name of the test/assessment so that they are aware of their performance and where they can better improve.

1. Courses

The courses tab in our app is for users to find information about the courses they are registered for. They can find the names of lecturers who teach the course and their email addresses. Lecturers can also find email addresses of their colleagues who teach the same courses as they do. This is added to make it easy for users to find who they need to speak to directly via email.

1. Push Notifications

This is so that students can get a notification or reminder that a project, lab or homework assignment is due. The notification will remind the student of the due date a day before it is due and on the day it is due. This feature is extremely beneficial because then students will be more aware of die dates and will hand in their work on time.

**Database Design**

# **Database Tables**

* **Local Database**

In our app we use *SQLite Database* to implement the local database and *DB Browser For SQLite* to view the tables. We have created the tables in Android Studio (reason for the inverted commas in the table declaration).

“SQLite is a relational database management system contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.”

The purpose of the local database is that if the user does not have an internet connection, they can still access the data that is stored offline. A lot of the functionalities of the app will still be functioning even in offline mode.

1. **USER TABLE** - The USER table is simply a table that stores user information like the User ID or Nickname. It is not related to any other database in terms of relationship and it should NOT be. It is used in conjunction with other local tables in order to do online queries. Think of the USER table as a shared preference from android studio. We were not aware of share preferences earlier on the project and only recently came to the knowledge of it, hence why it is there and not separated in share preferences.

A picture containing orange, player, room, holding

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. **LECTURER TABLE** - The LECTURER table is used to store information, well about the lecturers. It is used in conjunction with other tables and is often referenced (the Lecturer\_ID) as a foreign key. The information about the LECTURER table is also to display the courses details for the user’s course and show the user the lecturers email address for quick reference.

A picture containing table, orange, room, holding

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. **COURSE TABLE** - This table stores the user’s current course he/she is enrolled in. If the user is a lecturer, it will be assumed that the lecturer is “enrolled” in the course in order to provide the same functionality as it would for a student.

The course table stores the name and the course code of the user’s courses at WITS. The courses are gathered by the “WITS\_COURSE” Database and the user has NO influence in the courses that are provided by the app.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. **REGISTERED TABLE** - Linking the COURSE table and the LECTURER table together is the REIGISTERED table. The reason this table exists in the Local Database is so that when displaying the user’s course, the respective email address for each course’s lecturer will be displayed, naturally a lecturer can have more than one course to teach hence the existence of this table.

A picture containing indoor, photo, table, sitting

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. A screenshot of a cell phone

   Description automatically generated**MESSAGE TABLE** - Onto the fun tables! The MESSAGE table stores all “messages” that a lecturer adds (for the message board), which is uploaded to the Online Database when the student has internet access. He/she will receive all the messages sent out by the lecturer. Naturally, only lecturers can send out messages. Students cannot. (*They are basically more like informal emails).*

A screenshot of a computer

Description automatically generated

1. A close up of a screen

   Description automatically generated**TEST TABLE** - The make or break part of every student’s day… The TEST table stores the current users’ tests details. The lecture can however add marks for quizzes or other assessments. Well, this paragraph’s word count is as low as some of my test percentages. ☹ Ha! Just joking… hopefully.

A screenshot of a cell phone

Description automatically generated

1. A screen shot of a person

   Description automatically generated**USER\_TASK TABLE -** Stores any tasks made from the user, if a student made a task it will go here. If a lecturer chooses to store a task that only he wants to see, it is stored in this table. Only the user will have access to this task, and it will never ever see the light of the Online Database. Can contain a due date or due time and even course code!

A screenshot of a cell phone

Description automatically generated

1. **LECTURER\_TASK TABLE** - This is the USER\_TASK table’s more formal brother, LECTURER\_TASK gets all the attention of the Online Database. Literally every task in this database is from the Online Database. Naturally only a lecturer can add to this table.

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

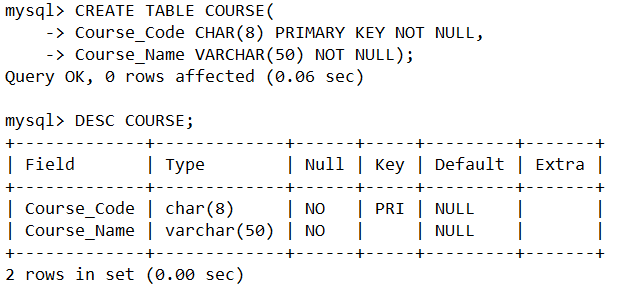
*You may now refer to the ERD of the local database on page*

Even though this Local Database is meant to only store data. We have ensured that the relationships are properly defined between them, even if the relationships are not exactly like that of its Online Database counterpart. All attributes for each table are properly defined to ensure that the data is not incorrectly inserted in any possible away.

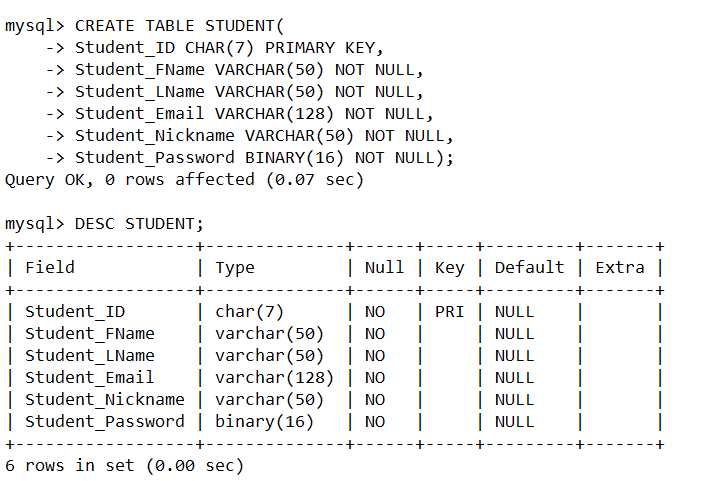
In my opinion, the Local Database fills its purpose of simply saving data from the Online Database. It ensures the integrity on the data due to the relationships. The Online Database is **not** **dependant** on any Local Database in any way. Therefore, it poses no security risk at all. The Local Database is **heavily** **dependent** on the Online Database. It regularly syncs whenever it has the chance too (well not literally whenever it can, but often enough).

* **Online Database (LecTrac)**

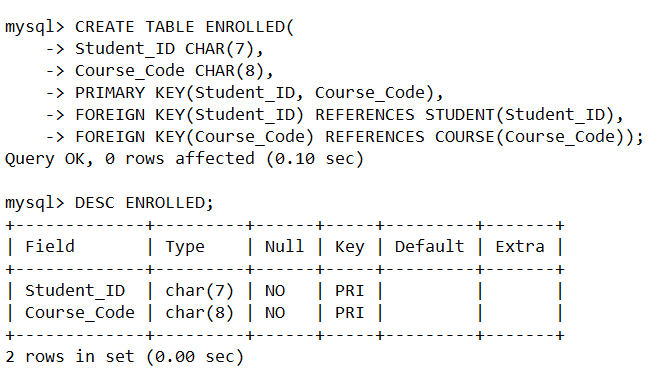
1. **COURSE TABLE** – The course table stores all courses that are offered at the university. This table is filled using the WITS\_COURSE table.



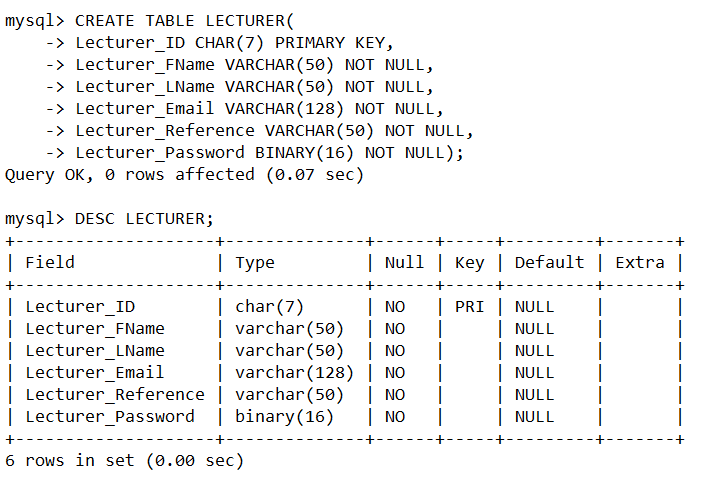
1. **STUDENT TABLE** – This table contains all students that have registered to the LecTrac app.

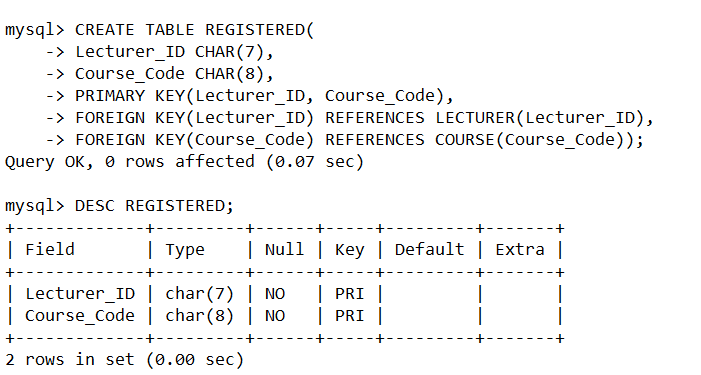


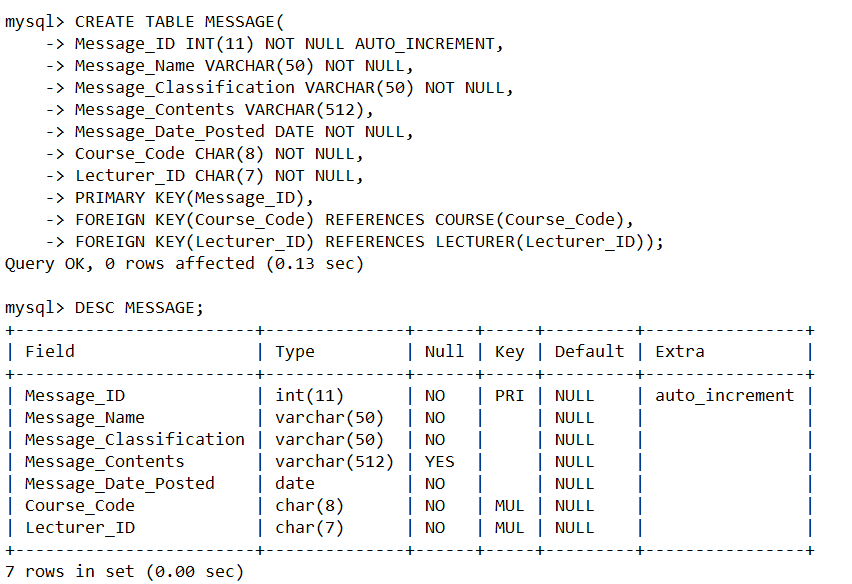
1. **ENROLLED TABLE** – This table connect the COURSE table and the STUDENT table (many-to-many relationship).

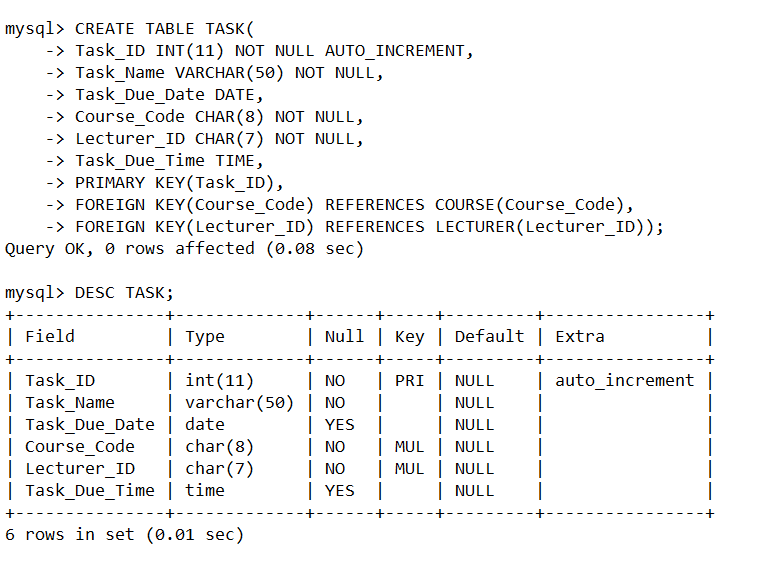


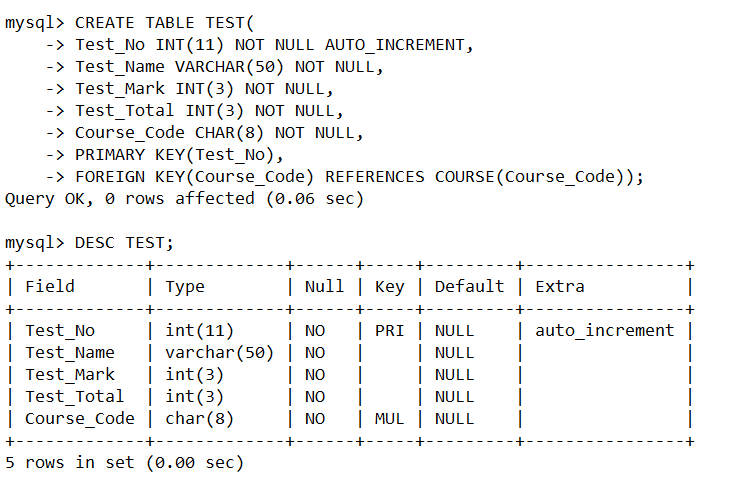
**LECTURER TABLE** – The details for all lecturers who have registered to use the LecTrac app are stored in this table.



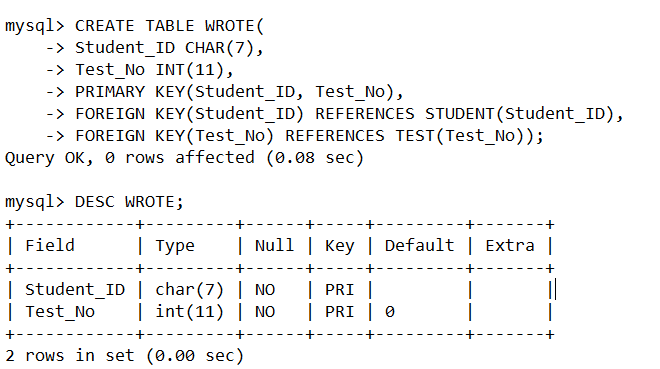
1. **REGISTERED TABLE** - This table connect the COURSE table and the LECTURER table (many-to-many relationship).
2. **MESSAGE TABLE** – All messages that lecturers post on the message board are stored in this table.



1. **TASK TABLE** – All tasks that lecturers post to students are stored in this table.
2. **TEST TABLE** – All test/assessment marks are taken from the WITS\_TEST table and transferred into this table using a trigger.

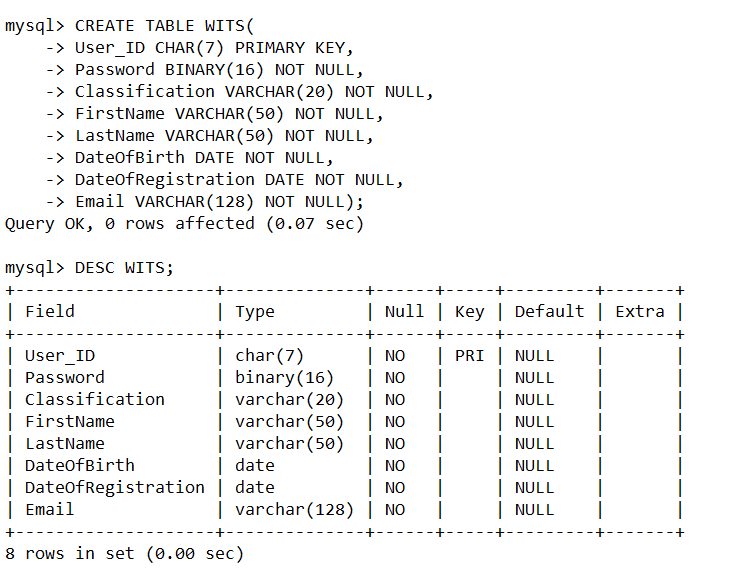


1. **WROTE TABLE** - This table connect the STUDENT table and the TEST table (many-to-many relationship).

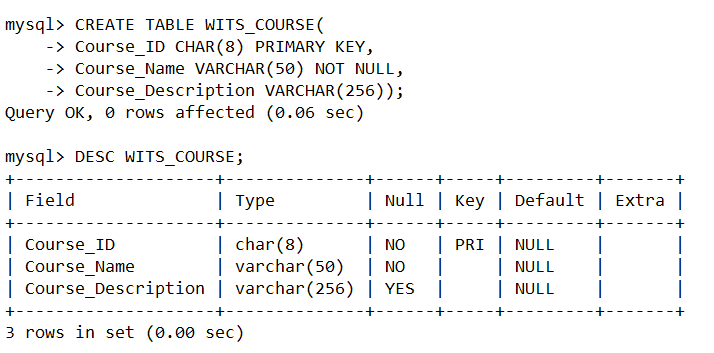


* **Online Database (Wits Dummy)**

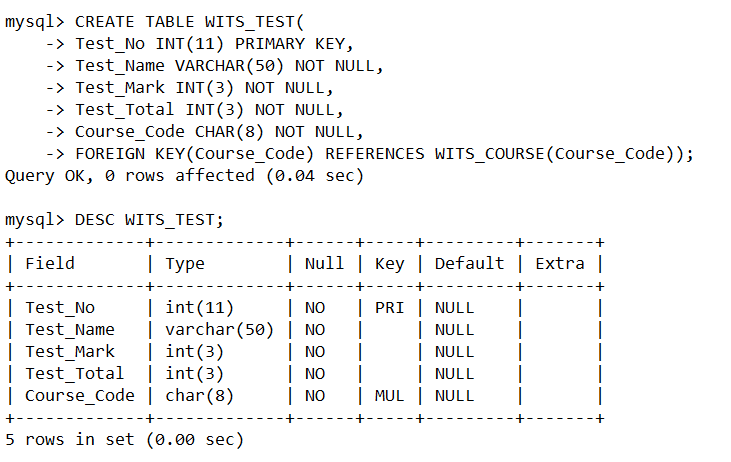
1. **WITS TABLE** – This table stores all student and lecturers. We created one table for both user types because this is not actually our database. We just need it to do verification checks for registration.



1. **WITS\_COURSE TABLE** – The table where all university courses are stored.



1. **WITS\_TEST TABLE** – All student marks are stored in this table.



# **Business Rules**

1. A student must be enrolled for at least one course.
2. A course can have many enrolled students.
3. A lecturer must be registered to teach at least one course.
4. A course can have many lecturers registered to teach it.
5. A course can have zero or many tasks.
6. A task can belong to only one course.
7. A student can have written many tests.
8. A test can be written by many students.
9. A course can generate many tests.
10. A test can be generated by only one course.
11. A lecturer can post zero or many tasks.
12. A task can only be posted by one lecturer.
13. A lecturer can post many messages.
14. A message can be posted by only one lecturer.
15. A course can have many messages.
16. A message can be about only one course.

# **Initial ERDs**

In our project, we have two ERD designs because we are using two databases that belong to out app. The first database is the *LecTrac* **local** database and the second database is the *LecTrac* **online** database.

We are also using the database of the Witwatersrand. *Please note* that in our case this is only a dummy database as we do not actually have access to the university database.

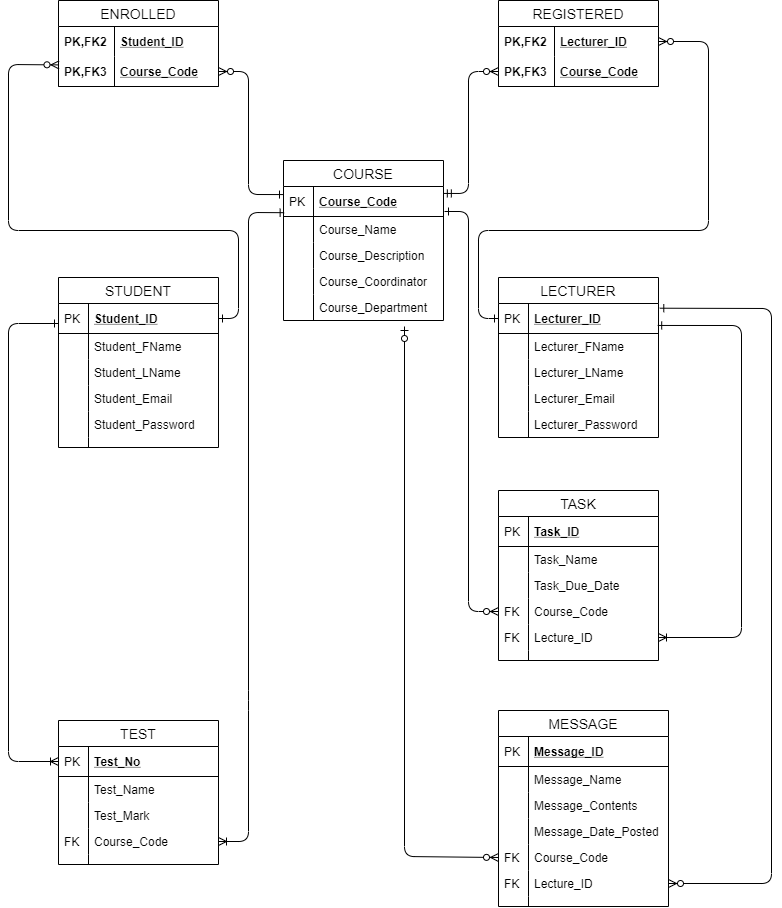
We use it as a security precaution in our app. When a student or lecturer registers to the app, it will confirm the Student\_ID or Lecturer\_ID with the actual Wits database. To make understanding easier, we have included a diagram to explain the relationship between our online LecTrac database and the university database.

* **LecTrac Local ERD:**

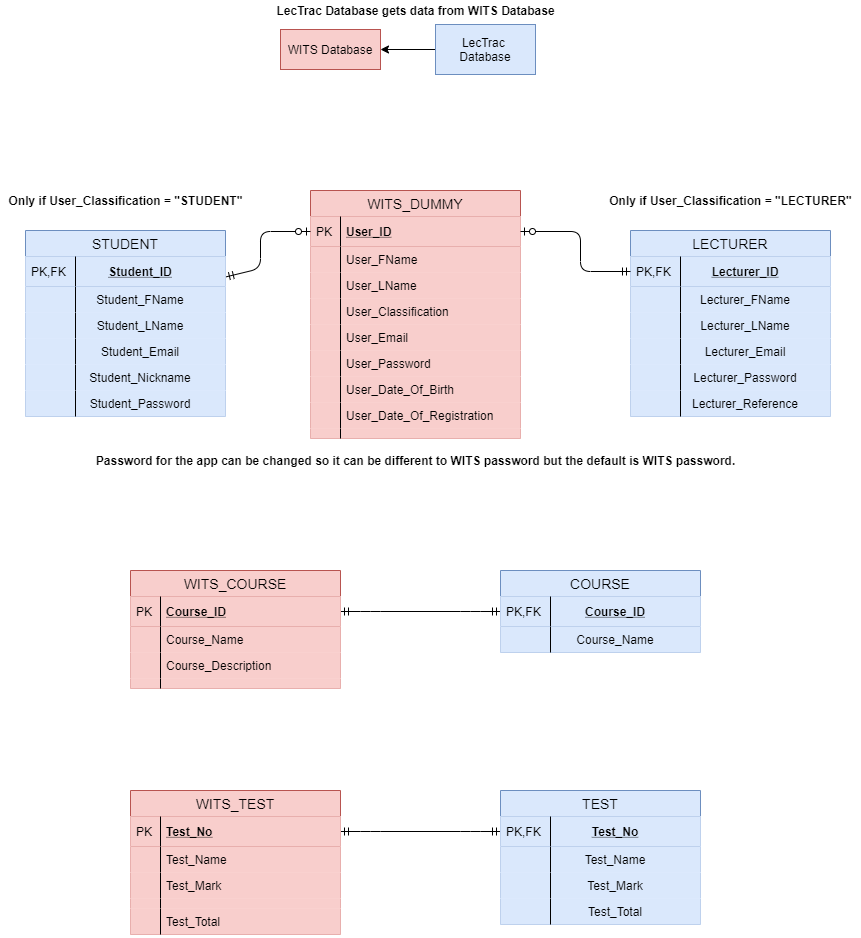
A screenshot of a cell phone

Description automatically generated

* **LecTrac Online ERD:**



* **External View Diagram:**



# **SQL Statements**

# **Final ERDs**

A close up of text on a white background

Description automatically generated

A screenshot of a cell phone

Description automatically generated

NOTE TO KEYATA

THIS LOOKS FREAKING AMAZEBOLLZ!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

CAN YOU ALSO NOTE THAT THE WAY THE PASSWORD HASHING WORK IS THAT IT HASED the password based on

“MD5” – yes all CAPS

Then it COPYS ONLY THE FIRST 16 CHARACTERS AND THE REASON IT ONLY TAJES 16 CHARS AND NOT MORE IS COZ THAT’S HOW IT WAS IN THE DB AND THEN I WAS TOO LAZY TO CHANGE IT………. SO YEA

TO HIMA AND PRAVESH

PLEASE NOTE THAT EVEN THOUGH SOME PHP QUERIES DON’T USE BIND PARAMS DUE TO THEIR LIMITATIONS, WE HAVE STEPS AND PUT THE DATA THROUGH A SANITIZATION PROCESS BEFORE ALLOWING IT ANY INTERACTIONS WITH A SQL QUERY

AND IM SURE YOU KNOW THIS BUT JUST POINTING IT OUT JUST IN CASE, THE CONTENTS PAGE DOESN’T WORK ☹