COM1025 Web and Database Systems

Coursework Assignment

[University Database for Societies]

|  |  |
| --- | --- |
| URN | 6830924 |
| Username | rk01161 |
| Date | 30.12.2023 |

# Business Rules and Assumptions

1. A university has departments that offers courses and has two categories of students: undergraduates and postgraduates.
   1. The university needs to store the following details about each student: URN, full name, date of birth, phone number, student type (postgraduate-undergraduate) and what course they are on.
   2. A student must be a postgraduate or an undergraduate, but they cannot be both at the same time.
   3. A postgraduate student will have an URN and thesis while an undergraduate student will have an URN and total credits.
   4. A student must be enrolled on one and only one course while a course can enrol no student or many students.
   5. A course has a course code, title, a department and enrolment (total number of students on the course)
   6. A department has a department ID and a department name.
   7. A department can run no course or many courses, but a course must be run by one and only one department.
2. The university also wants to keep data on hobbies of students if they wish to provide that information so that appropriate societies can be suggested to them.
   1. A student can have no hobby or can have many hobbies.
   2. A hobby can be related to no student or to many students.
   3. A hobby has an ID, name and equipment which is a multivalued attribute. (e.g., knitting is a hobby that requires multiple equipment like needles, yarn, scissors, crochet hook)
   4. That multivalued attribute called equipment is stated in another table linked to hobby table and takes HOBBY\_ID as a PK. A hobby can have multiple equipment or none, but an equipment must be related to one and only one hobby.
   5. The current list of hobbies include reading, hiking, chess, knitting, ballroom dancing, football, tennis, rugby, climbing, rowing, music.
   6. A society can be related to no hobby or can be related to many hobbies.
   7. A hobby can have no society assigned to it or can have many societies assigned to it.
   8. A society has an ID, name, leader which is a student and a website.
   9. A society must have a student or many students.
   10. A student can be a member of no societies or many societies.
3. The university also keeps data of the events that societies run.
   1. A society can run no activities or many activities.
   2. An activity must be run by one and only one society.
   3. An activity has an ID, name, date, time and a location.

# Extended Entity Relationship Diagram (EERD)

A computer screen shot of a chart

Description automatically generated

# Logical Relational Database Schema

**Department**(DEPARTMENT\_ID, Department\_Name)

PRIMARY KEY: DEPARTMENT\_ID

**Course**(Crs\_Code, Crs\_Title, Crs\_Enrollment, Crs\_Department)

PRIMARY KEY: Crs\_Code

FOREIGN KEY: Crs\_Department REFERENCES **Department**(DEPARTMENT\_ID)

**Student**(URN, Stu\_FName, Stu\_LName, Stu\_DOB, Stu\_Phone, Stu\_Course, Stu\_Type)

PRIMARY KEY: URN

FOREIGN KEY: Stu\_Course REFERENCES **Course**(Crs\_Code)

**Undergraduate**(URN, Total\_credits)

PRIMARY KEY: URN

FOREIGN KEY: URN REFERENCES **Student**(URN)

**Postgraduate**(URN, Thesis)

PRIMARY KEY: URN

FOREIGN KEY: URN REFERENCES **Student**(URN)

**Hobbies**(HOBBY\_ID, Hobby\_Name)

PRIMARY KEY: HOBBY\_ID

**Equipment**(HOBBY\_ID, Equipment)

PRIMARY KEY: HOBBY\_ID, Equipment

FOREIGN KEY: HOBBY\_ID REFERENCES **Hobbies**(HOBBY\_ID)

**StuHob**(URN, HOBBY\_ID)

PRIMARY KEY: URN, HOBBY\_ID

FOREIGN KEY: URN REFERENCES **Student**(URN), HOBBY\_ID REFERENCES **Hobbies**(HOBBY\_ID)

**Societies**(SOCIETY\_ID, Society\_Name, Society\_Leader, Society\_Website)

PRIMARY KEY: SOCIETY\_ID

FOREIGN KEY: Society\_Leader REFERENCES **Student**(URN)

**Activities**(ACTIVITY\_ID, Activity\_Name, Activity\_Date, Activity\_Time, Activity\_Location, Activity\_Society)

PRIMARY KEY: ACTIVITY\_ID

FOREIGN KEY: Activity\_Society REFERENCES **Societies**(SOCIETY\_ID)

**StuSoc**(URN, SOCIETY\_ID)

PRIMARY KEY: URN, SOCIETY\_ID

FOREIGN KEY: URN REFERENCES **Student**(URN), SOCIETY\_ID REFERENCES **Societies**(SOCIETY\_ID)

**SocHob**(SOCIETY\_ID, HOBBY\_ID)

PRIMARY KEY: SOCIETY\_ID, HOBBY\_ID

FOREIGN KEY: SOCIETY\_ID REFERENCES **Societies**(SOCIETY\_ID), HOBBY\_ID REFERENCES **Hobby**(HOBBY\_ID)

# Website Working with MySQL Database

**A screenshot of a computer

Description automatically generatedstyle.css / styles.css:** They both are used to format/style the website. They have all the information about the layout of the website including text fonts, alignments, backgrounds, colours etc.

**activities.ejs:** This one has the *“Activities”* table and includes the functionality to view all and view one from the activity table which is given with rows. It also includes a navigation bar at the top that allows the user to go to other tables.

**course.ejs:** This one has the *“Course”* table and includes the view all and view one functionalities from the course table which is given with rows. It also includes a navigation bar at the top that allows the user to go to other tables.

**department.ejs:** This one has the *“Department”* table and includes the view all and view one functionalities from the department table which is given with rows. It also includes a navigation bar at the top that allows the user to go to other tables.

**hobbies.ejs:** This one has the *“Hobbies”* table and includes the view all and view one functionalities from the hobbies table which is given with rows. It also includes a navigation bar at the top that allows the user to go to other tables.

**home.ejs:** This one is the *“Home page”.* It has a header that includes a button that directs the user to “Societies” table, content navigation bar like the other files and an “About” part.

**societies.ejs:** This one has the *“Societies”* table and includes the view all and view one functionalities from the societies table which is given with rows. It also includes a navigation bar at the top that allows the user to go to other tables.

**student.ejs:** This one has the *“Student”* table and includes the view all and view one functionalities from the student table which is given with rows. Additionally, and unlike the other files, this one also includes update and delete functionalities. Update functionality allows the user to change any data about a student except their URN. This update route has validations that if the user tries to update a student’s information that is valid, text colour will be green and allow the user to update it. But if the information that is entered is not valid, text box will turn red, and user won’t be able to update it. It also includes a navigation bar at the top that allows the user to go to other tables.

**update.ejs:** This one has update route. It is only linked to student table, and it allows the user to click on the update button next to each student and edit their information. If the information that is edited is valid it will turn green, if not it will be red and won’t allow to update it. Each information that is edited requires to follow a specific pattern. This file also includes update button in it.

**index.js:** This one includes all the tables, validations, view all, view one, update and home routes. Each file accesses their table through here. What to access for each table is written with sql queries. Validations are specified with if statements through here as well.

# Advanced Tasks

**For Part 1:**

* Except from the given EERD, and the required hobby and two more tables in Section 2, I did one extra table. In total I have 6 main tables alongside with bridge entities and a table for the multivalued attribute called “Equipment” which is linked to Hobbies table. I translated all those tables from my EER Model to Relational Schema in Section 3.
* I also wrote 3 more queries as an extra challenge.

**For Part 2:**

* I added functionality to delete rows of data from “student”. (Both update and delete functionalities apply only for student table).

**6 References**

1. GitHub. (2023). *student-database-exercise/public/main.css at main · joeappleton18/student-database-exercise.* [online] Available at: https://github.com/joeappleton18/student-database-exercise/blob/main/public/main.css [Accessed 31 Dec. 2023].

<https://github.com/joeappleton18/student-database-exercise/blob/main/public/main.css>

1. GitHub. (2023). *space-park/styles.css at main · joeappleton18/space-park*. [online] Available at: https://github.com/joeappleton18/space-park/blob/main/styles.css [Accessed 31 Dec. 2023].

<https://github.com/joeappleton18/space-park/blob/main/styles.css>

1. www.foo.com. (2023). *Foo.com*. [online] Available at: http://www.foo.com [Accessed 31 Dec. 2023].

Based on the html template of <http://www.foo.com>

1. Surrey.ac.uk. (2023). Available at: https://www.ias.surrey.ac.uk/wp-content/uploads/2019/10/university-of-surrey-aerial-view.jpg [Accessed 31 Dec. 2023].

<https://www.ias.surrey.ac.uk/wp-content/uploads/2019/10/university-of-surrey-aerial-view.jpg>

# 7 Appendix: Screenshots of Website

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated