COM1025 Web and Database Systems

Coursework Assignment

[Ice skating society]

Zoe Weston

1 Business Rules and Assumptions

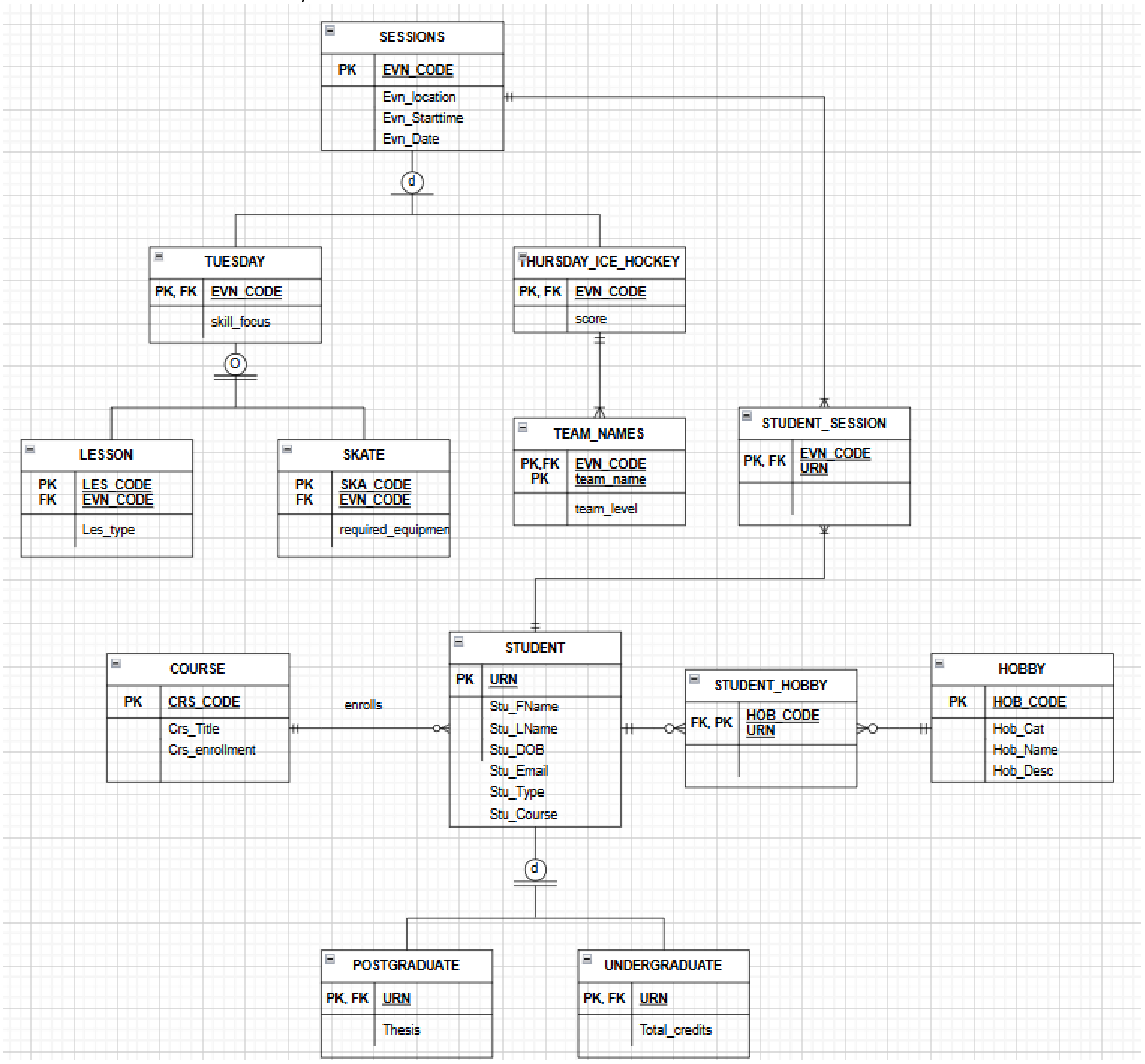
1. A university offers courses and has two categories of students: undergraduates and postgraduates.
    a. The university needs to store the following details about each student: name, date of birth, a phone number, and what course they are on.
    b. A student must be a postgraduate or an undergraduate, but they cannot be both at the same time.
    c. A student must be enrolled on one and only one course while a course can enrol no student or many students.
    d. A course has a course code, name and enrolment (total number of students on the course)
2. The university also wants to keep data on hobbies of students if they wish to provide that information so that appropriate clubs and societies can be suggested to them.
    a. A hobby has a name, a description and a category. Examples of categories and hobbies are shown below (these could be subjective and you can add/change items):

Category	Hobby Name
Sports	Football, tennis, swimming, pingpong, badminton, basketball, ice skating
Fitness	Yoga, Pilates, Taichi, hiking, climbing
Outdoor	Mountain climbing, fishing, kayaking, gardening
Intellectual	Reading, sudoku, puzzles, chess, coding, debating, poetry
Social	Sustainability campaigner, charity work, community work, animal rescue
Gaming	Multi-player games, Minecraft, FIFA
Construction	Modelling, 3D printing, Robots

    b. A student can have no hobby or can have many hobbies. A student does not have to provide any information on their hobbies if they choose not to.
    c. A hobby can be related to no student or to many students.
3. Within the ice skating society a student is able to be a part of many or no sessions. The database will record which sessions they attend.
    a. Sessions need a minimum of one student but can have many more.
    b. A session can be on one of two days, Tuesdays and Thursdays.
4. A Tuesday session will have a skills focus that the students will be primarily working on if they choose to attend a lesson or skate. A Tuesday session can be a lesson or a skate or both.
    a. A lesson will have a lesson type, such as a group lesson or an individual lesson.
    b. A skate will require them to bring their own equipment for the activity they are doing, other sessions will have the required equipment provided.
5. A Thursday session is only for ice hockey matches. For this a session level will need to be stored.
    a. In order for a session to run on a Thursday there must be more than one team to be playing, team_names will used as a multi-valued attribute to store this.
    b. Only teams of the same level are able to play against each other. The level of the teams needs to be stored.
6. A session does not have to be the regular Tuesday or Thursday sessions but will still need to be stored in the database with information such as location, start time and date.

2 Extended Entity Relationship Diagram (EERD)

I used draw.io to create my EERD



3 Logical Relational Database Schema

Course(Crs_Code, Crs_Title, Crs_enrollment)

PRIMARY KEY: Crs_Code

Student(URN, Stu_FName, Stu_LName, Stu_DOB, Stu_Email, Stu_Course, Stu_Type)

PRIMARY KEY: URN
FOREIGN KEY: Stu_Course REFERENCES **Course**(Crs_Code)

Postgraduate(URN, Thesis)

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

Undergraduate(URN, Total_credits)

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

Hobby(HOB_CODE, Hob_Cat, Hob_Name, Hob_Desc)

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

Student_Hobby(HOB_CODE, URN)

PRIMARY KEY: HOB_CODE, URN

FOREIGN KEY: URN REFERENCES **Student**(URN), HOB_CODE REFRENCES **Hobby**(HOB_CODE)

Student_session(EVN_CODE, URN)

PRIMARY KEY: EVN_CODE, URN

FOREIGN KEY: URN REFERENCES **Student**(URN), EVN_CODE REFRENCES **Sessions**(EVN_CODE)

Sessions(EVN_CODE, Evn_location, Evn_Starttime, Evn_Date)

PRIMARY KEY: EVN_CODE

Tuesday(EVN_CODE, Skill_Focus)

PRIMARY KEY: EVN_CODE
FOREIGN KEY: EVN_CODE REFRENCES **Sessions**(EVN_CODE)

Thursday_Ice_Hockey(EVN_CODE, score)

PRIMARY KEY: EVN_CODE
FOREIGN KEY: EVN_CODE REFERENCES **Sessions**(EVN_CODE)

Lesson(LES_CODE, EVN_CODE, Les_type)

PRIMARY KEY: LES_CODE
FOREIGN KEY: EVN_CODE REFERENCES **Tuesday**(EVN_CODE)

Skate(SKA_CODE, EVN_CODE, required_equipment)

PRIMARY KEY: SKA_CODE
FOREIGN KEY: EVN_CODE REFERENCES **Tuesday**(EVN_CODE)

Team_Names (EVN_CODE, TEAM_NAME, Team_Level)

PRIMARY KEY: EVN_CODE, TEAM_NAME
FOREIGN KEY: EVN_CODE REFERENCES **Thursday_Ice_Hockey** (EVN_CODE)

4 Website Working with MySQL Database

Index.js – written in JavaScript, provides middleware functionality, connects the database to the interface and contains the application routes.

Header.ejs – written in HTML, provides frontend functionality. Creates universal links the each page at the top of all pages

Index.ejs – written in HTML, provides frontend functionality, creates the homepage and displays values from database with corresponding headings

Courses.ejs – written in HTML, provides frontend functionality, displays table of all courses stored in the database. Shows their course code, course title and enrolment, as well as having a link to the edit page

Create.ejs – written in HTML, provides frontend functionality, includes input boxes for course code, course title and course enrolment with corresponding headings. Also includes a create course button.

Edit.ejs – written in HTML, provides frontend functionality, includes input boxes for course title and course enrolment with corresponding headings. As well as that, an update course button.

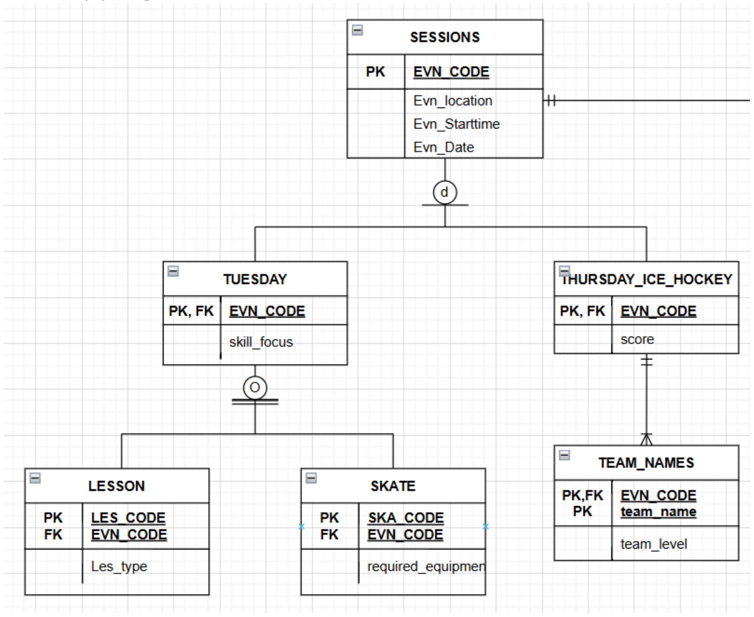
Main.css – written in CSS, provides uniform colour across all pages.

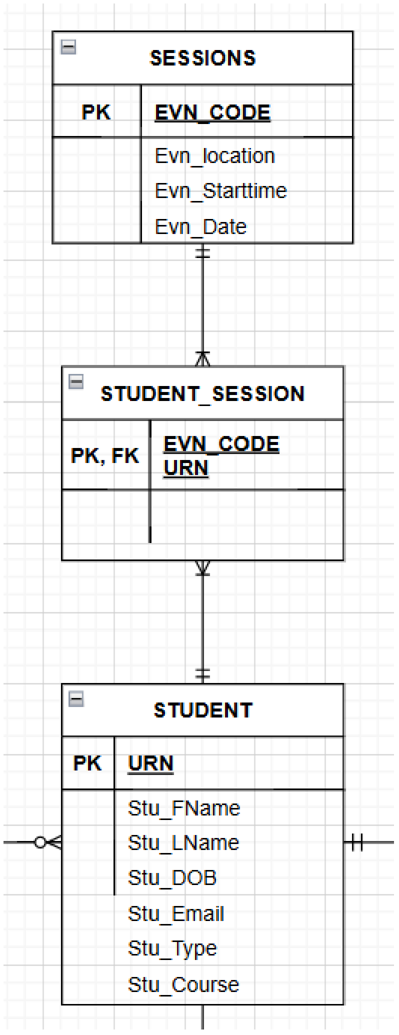
Db_setup.sql – stores the SQL commands to create the database which are referenced by the webpages, this provides the backend functionality of the website.

Delete.ejs – written in HTML, provides frontend functionality, this page was created to delete extra courses that were added for testing purposes. Includes a heading and a delete button. This page is not required and is not accessible from the other webpages so requires its URL searched directly.

5 Advanced Tasks

I included two specialisation hierarchies into my EERD. The connection from Sessions to Tuesday and Thursday is an disjoint partial so a session can be on either day with their specific requirements or it can be separate from both as a different type of event. The connection from Tuesday to lesson and skate is a overlapping complete constraint because if a session was happening on a Tuesday it must be either a lesson or a skate. For both of these any student is able to attend both so they are both overlapping constraints.





I included a many to many relationship between student and sessions because one student can attend many sessions and one session can have many students. In order to achieve this I included the bridge entity "Student_Session" with a composite primary key of "Evn_code" and "URN".

6 References

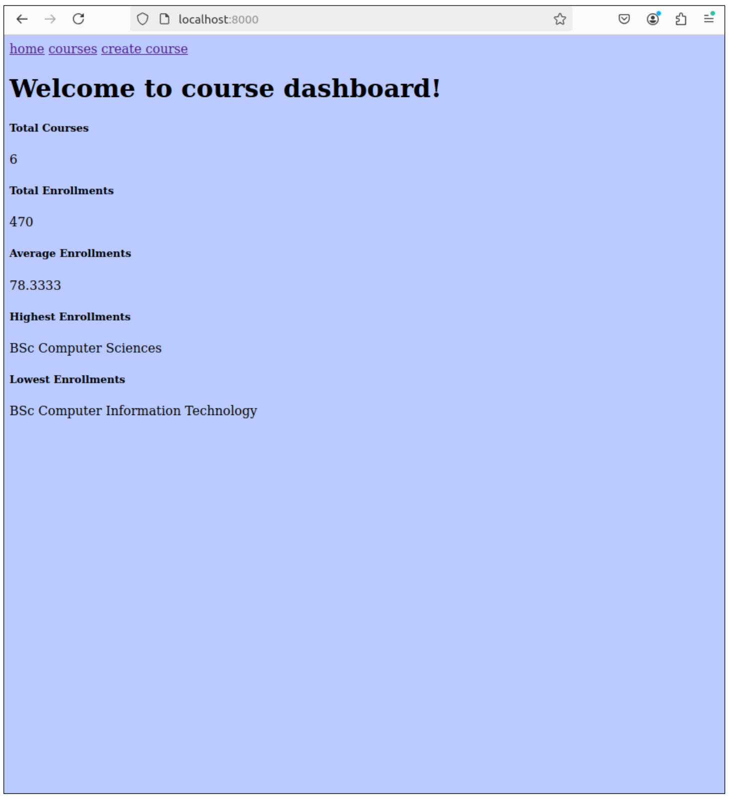
University of Surrey Ice Skating Society: https://surreyunion.org/IceSkating/ (accessed 10 December 2024) this was used to understand the basics of how the society is run and what events occur on a regular basis.

W3schools: https://www.w3schools.com/sql/default.asp (accessed 7 January 2025) this website was used as reference for the SQL section of the coursework.

University of Surrey SurreyLearn: https://surreylearn.surrey.ac.uk/d2l/le/lessons/267643/topics/3189235 (accessed 8 January 2025) lab 10 was referenced for the creation of the website.

Joeappleton18/WEB-AND-DATABASE-SYSTEMS week 10 lab solutions: https://github.com/joeappleton18/WEB-AND-DATABASE-SYSTEMS/tree/master/week-10/solutions (accessed 8 January 2025) refenced for the creation of functionality of the website.

7 Appendix: Screenshots of Website



localhost:8000

home courses create course

Welcome to course dashboard!

Total Courses

6

Total Enrollments

470

Average Enrollments

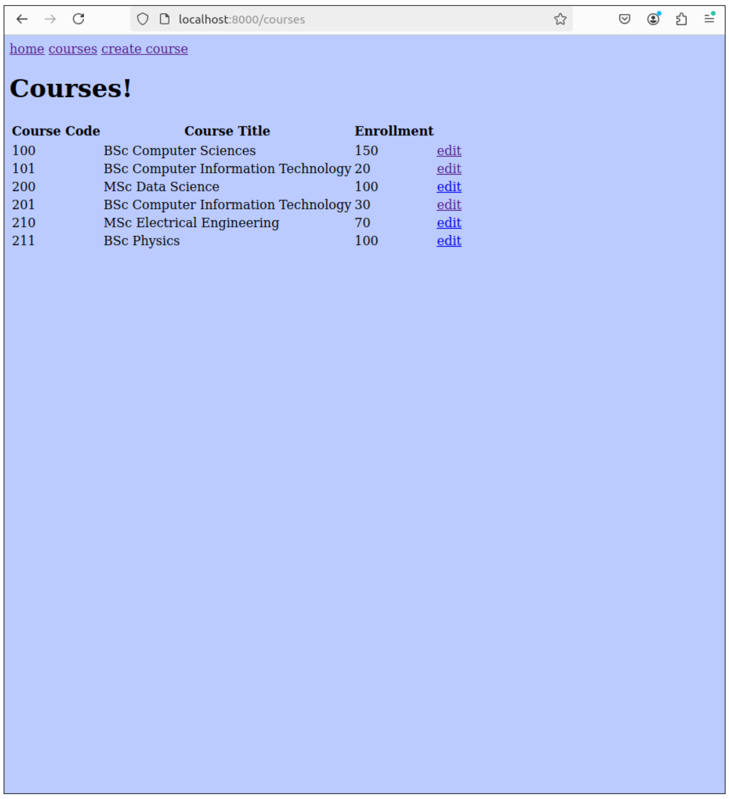
78.3333

Highest Enrollments

BSc Computer Sciences

Lowest Enrollments

BSc Computer Information Technology

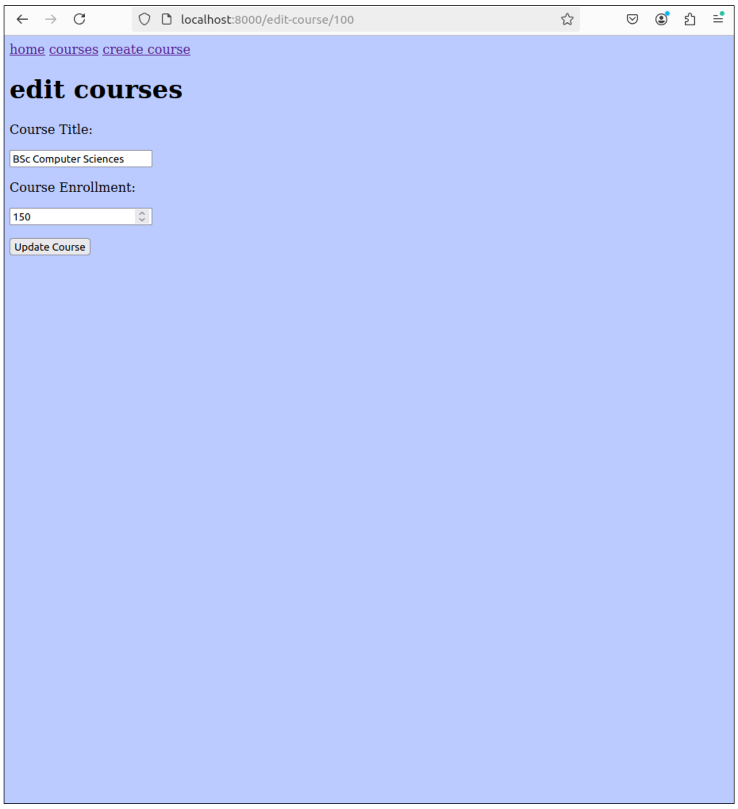


localhost:8000/courses

home courses create course

Courses!

Course Code	Course Title	Enrollment	
100	BSc Computer Sciences	150	edit
101	BSc Computer Information Technology	20	edit
200	MSc Data Science	100	edit
201	BSc Computer Information Technology	30	edit
210	MSc Electrical Engineering	70	edit
211	BSc Physics	100	edit



localhost:8000/edit-course/100

home courses create course

edit courses

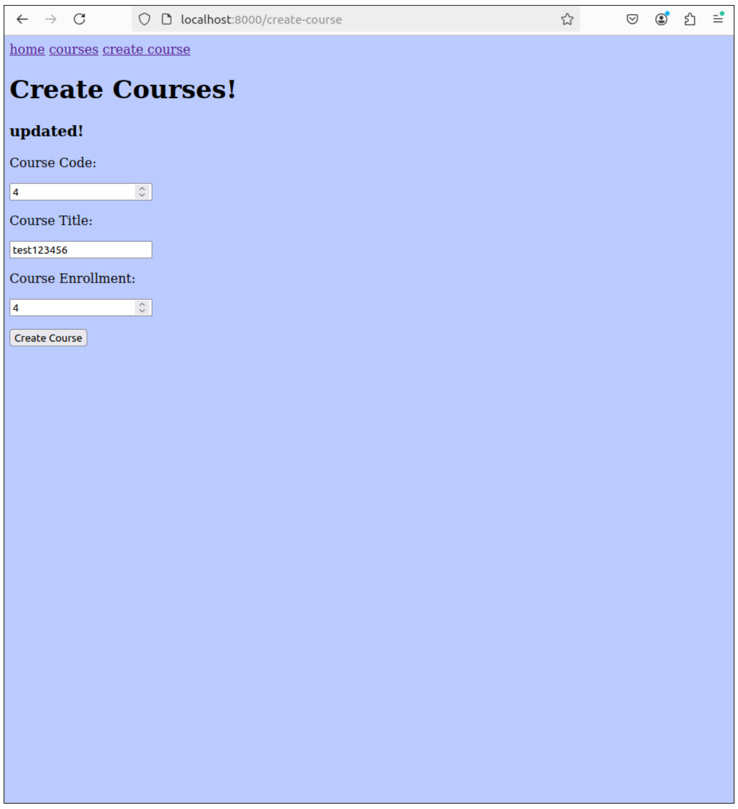
Course Title:

BSc Computer Sciences

Course Enrollment:

150

Update Course



localhost:8000/create-course

home courses create course

Create Courses!

updated!

Course Code:

4

Course Title:

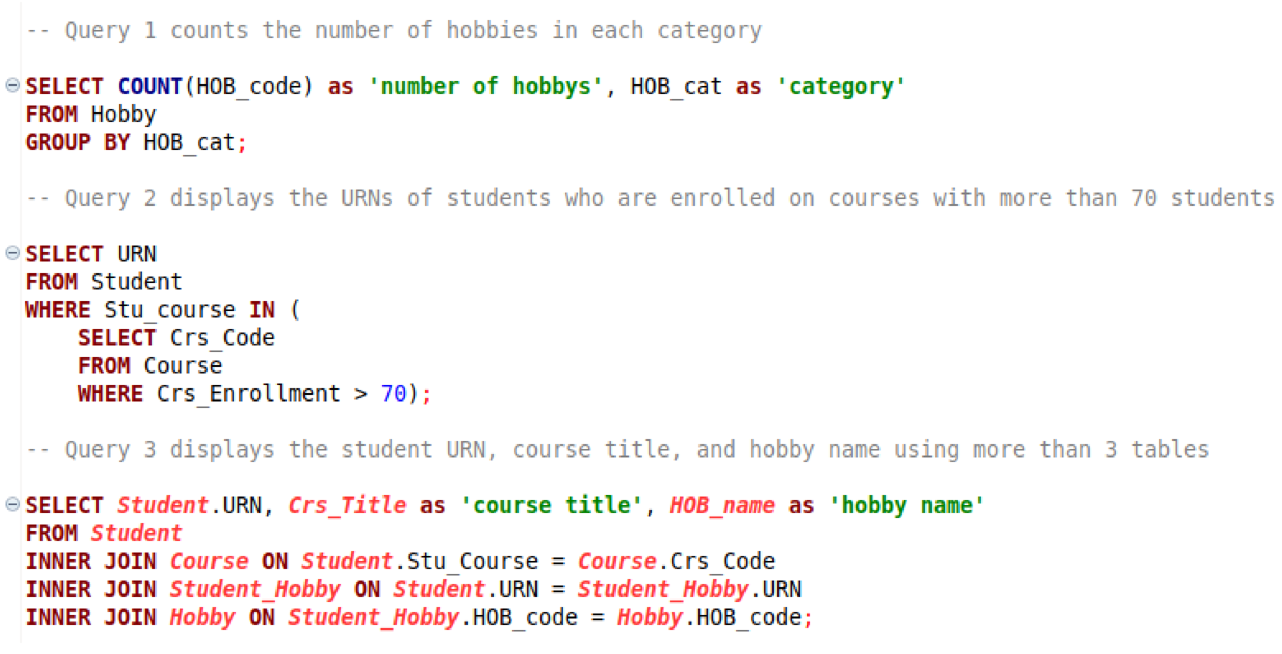
test123456

Course Enrollment:

4

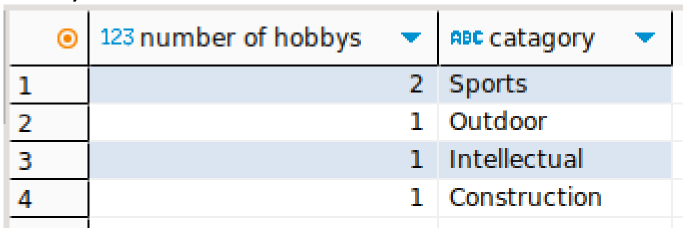
Create Course

8 Appendix: Screenshot of SQL queries



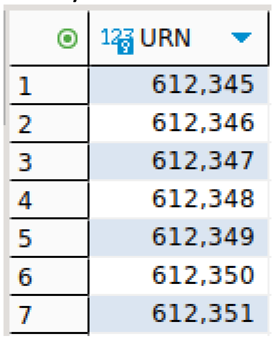
```
-- Query 1 counts the number of hobbies in each category

SELECT COUNT(HOB_code) as 'number of hobbys', HOB_cat as 'category'
FROM Hobby
GROUP BY HOB_cat;

-- Query 2 displays the URNs of students who are enrolled on courses with more than 70 students

SELECT URN
FROM Student
WHERE Stu_course IN (
    SELECT Crs_Code
    FROM Course
    WHERE Crs_Enrollment > 70);

-- Query 3 displays the student URN, course title, and hobby name using more than 3 tables

SELECT Student.URN, Crs_Title as 'course title', HOB_name as 'hobby name'
FROM Student
INNER JOIN Course ON Student.Stu_Course = Course.Crs_Code
INNER JOIN Student_Hobby ON Student.URN = Student_Hobby.URN
INNER JOIN Hobby ON Student_Hobby.HOB_code = Hobby.HOB_code;
```

Query 1



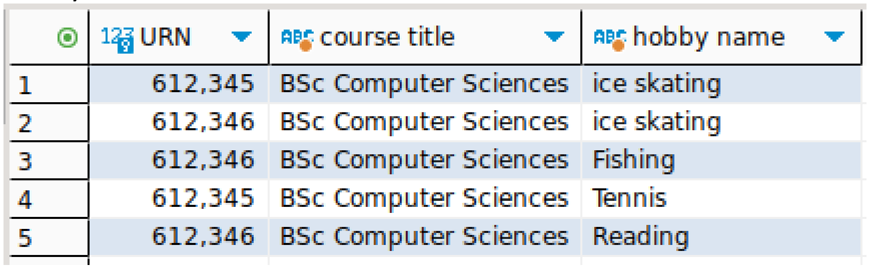
	number of hobbys	catagory
1	2	Sports
2	1	Outdoor
3	1	Intellectual
4	1	Construction

Query 2



	URN
1	612,345
2	612,346
3	612,347
4	612,348
5	612,349
6	612,350
7	612,351

Query 3



	URN	course title	hobby name
1	612,345	BSc Computer Sciences	ice skating
2	612,346	BSc Computer Sciences	ice skating
3	612,346	BSc Computer Sciences	Fishing
4	612,345	BSc Computer Sciences	Tennis
5	612,346	BSc Computer Sciences	Reading