Cameron Gibson
Jeremy Wenzel
Rhett Calvert
Garret Mbaku
Anish Chhetri

Texas Tech Database Report

Problem Statement:

In order to help Texas Tech keep track of departments, instructors, students and courses our group has developed a database to keep track and organize information from rNumbers to a department's physical address. We broke this up into five main tables which we will cover in the next slides. Most of the relationships we saw when conceptually breaking down this problem were One-to-many or Many-to-many. Such as an instructor having one department but a department having many instructors, or many students to many courses.

Conceptual and Logical Design:

We broke the problem into five tables, Department, Instructor, Course, Student and Course_Student. The following images show all the tables constructed, showing different rows and data types.

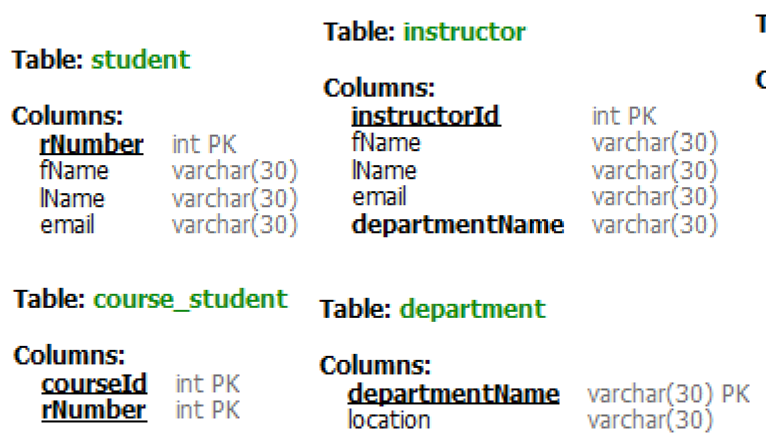


Table: student

Columns:

rNumber	int PK
fName	varchar(30)
lName	varchar(30)
email	varchar(30)

Table: instructor

Columns:

instructorId	int PK
fName	varchar(30)
lName	varchar(30)
email	varchar(30)
departmentName	varchar(30)

Table: course

Columns:

courseId	int PK
term	char(10)
courseName	varchar(30)
departmentName	varchar(30)
instructorId	int

Table: course_student

Columns:

courseId	int PK
rNumber	int PK

Table: department

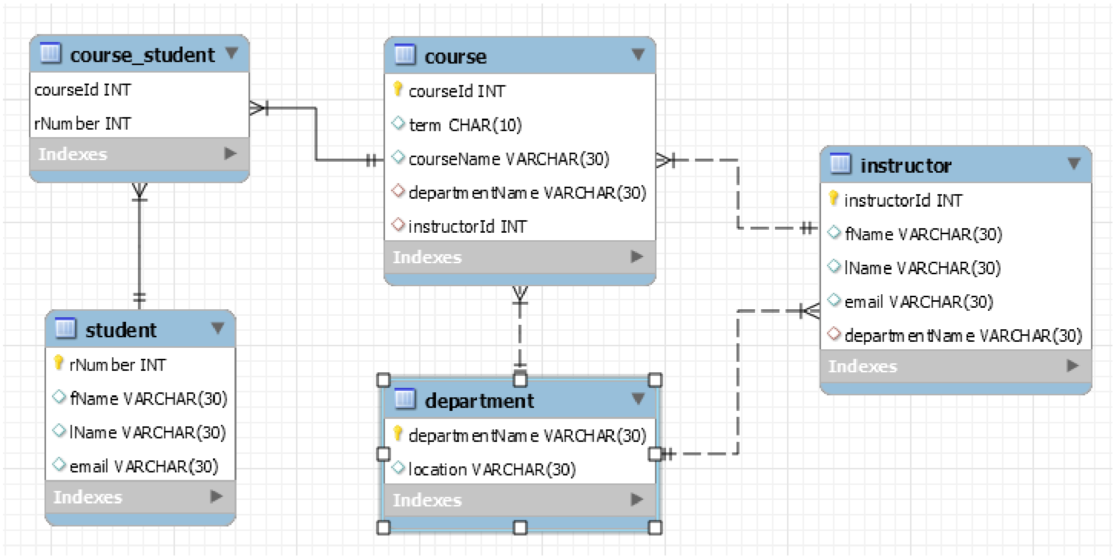
Columns:

departmentName	varchar(30) PK
location	varchar(30)

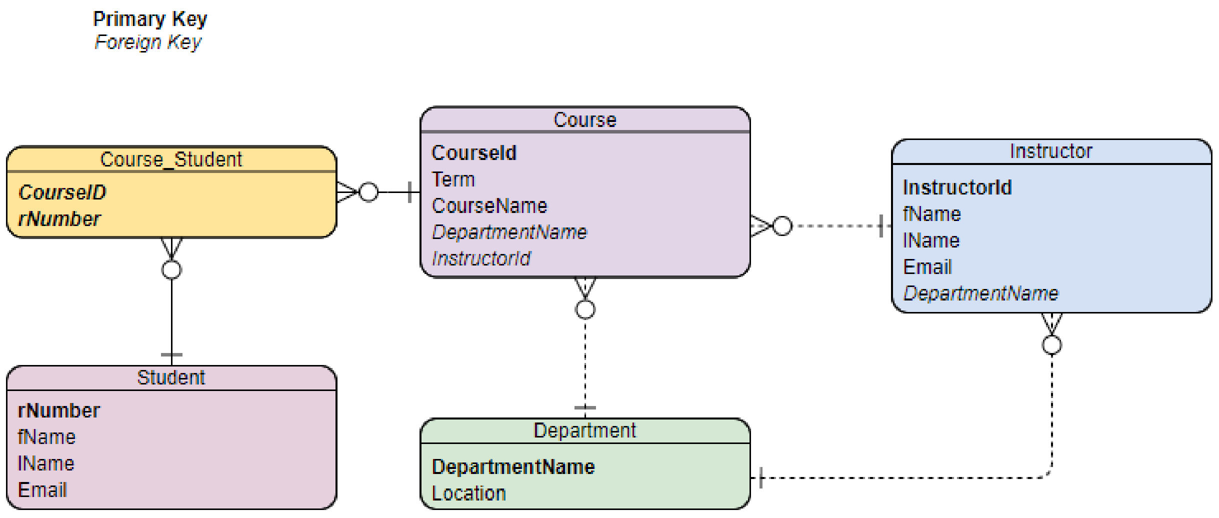
Normalization:

The department table originally had a varchar() limit of 60, but due to these being heavily called in the program, we made the decision to drop the City, State and Zip Code since all buildings reside in Lubbock, Texas on Texas Tech Campus. In a larger program, it can be added back for multiple university locations in different cities, states, or zip codes. We had to create a denormalized view to see the classes the students have enrolled in and their correlating rNumbers.

The entity relationships are shown below in the entity relationship diagram generated by MySql.



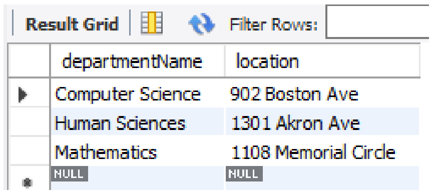
The table creation pictures above show the primary keys and foreign keys, here is a graph showing the logical design of the system.



Physical Design:

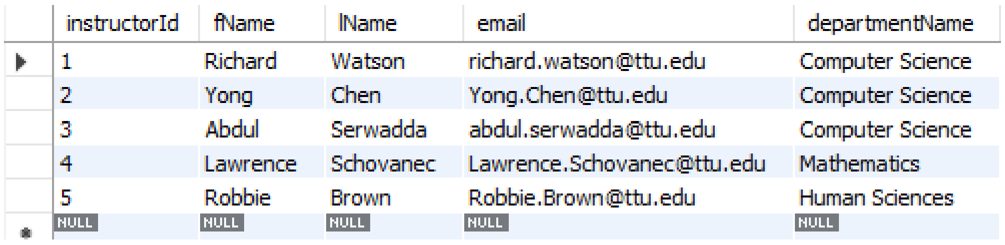
For the database, our group decided to use MySql Version 8.0.19, using the InnoDB engine, charset utf8mb4 and collation utf8mb4_0900_. The first step after we created the table structure for the scheme was to populate it. The images below show the tables and the information they contain after performing the written inserts.

Populating Departments:



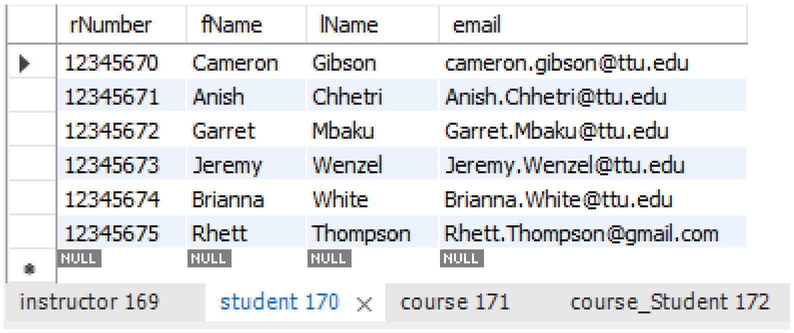
departmentName	location
Computer Science	902 Boston Ave
Human Sciences	1301 Akron Ave
Mathematics	1108 Memorial Circle
NULL	NULL

Instructors:



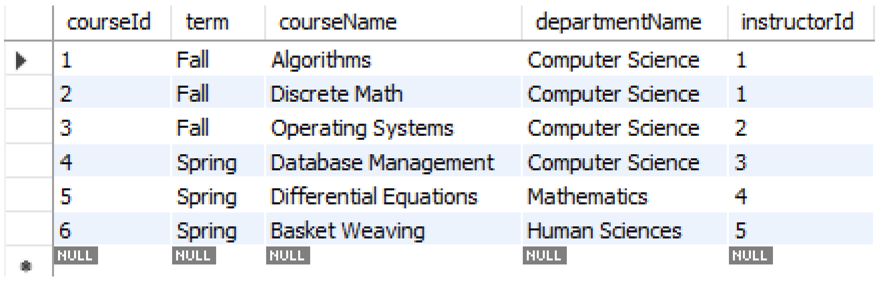
instructorId	fName	lName	email	departmentName
1	Richard	Watson	richard.watson@ttu.edu	Computer Science
2	Yong	Chen	Yong.Chen@ttu.edu	Computer Science
3	Abdul	Serwadda	abdul.serwadda@ttu.edu	Computer Science
4	Lawrence	Schovanec	Lawrence.Schovanec@ttu.edu	Mathematics
5	Robbie	Brown	Robbie.Brown@ttu.edu	Human Sciences
NULL	NULL	NULL	NULL	NULL

Students:



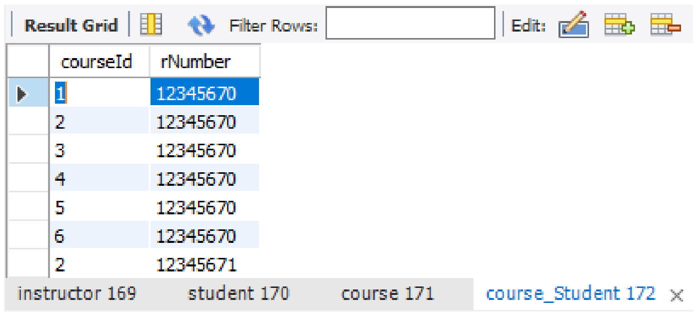
rNumber	fName	lName	email
12345670	Cameron	Gibson	cameron.gibson@ttu.edu
12345671	Anish	Chhetri	Anish.Chhetri@ttu.edu
12345672	Garret	Mbaku	Garret.Mbaku@ttu.edu
12345673	Jeremy	Wenzel	Jeremy.Wenzel@ttu.edu
12345674	Brianna	White	Brianna.White@ttu.edu
12345675	Rhett	Thompson	Rhett.Thompson@gmail.com
NULL	NULL	NULL	NULL

Courses:



courseId	term	courseName	departmentName	instructorId
1	Fall	Algorithms	Computer Science	1
2	Fall	Discrete Math	Computer Science	1
3	Fall	Operating Systems	Computer Science	2
4	Spring	Database Management	Computer Science	3
5	Spring	Differential Equations	Mathematics	4
6	Spring	Basket Weaving	Human Sciences	5
NULL	NULL	NULL	NULL	NULL

Course_Student (tracks the courseId and rNumber):



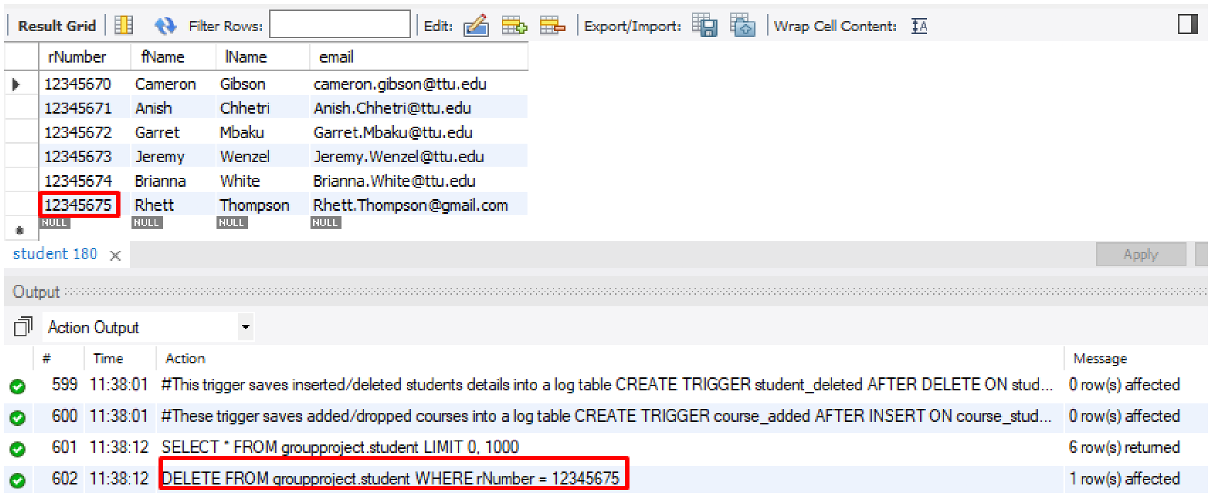
courseId	rNumber
1	12345670
2	12345670
3	12345670
4	12345670
5	12345670
6	12345670
2	12345671

Triggers:

Our group has designed four triggers, these are student_deleted, student_inserted, course_added, course_dropped. The student triggers are used to track the input and removal of students from the table and track in the log table: student_log. The course triggers are to track when a student drops or adds a class. A very practical use for this trigger is to determine if the student has hit their drop limit.

To test this trigger, we'll be dropping Rhett's details. Here is the view prior to the delete and the views after the delete:

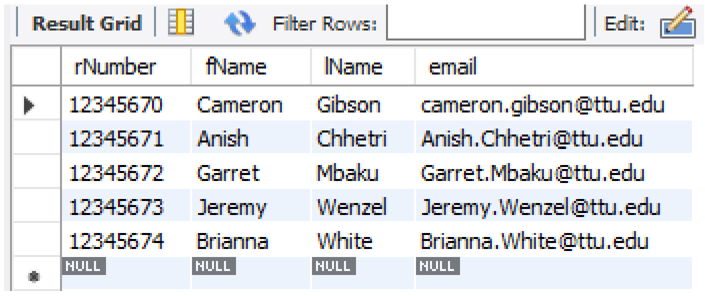
View Prior to Deletion:



rNumber	fName	lName	email
12345670	Cameron	Gibson	cameron.gibson@ttu.edu
12345671	Anish	Chhetri	Anish.Chhetri@ttu.edu
12345672	Garret	Mbaku	Garret.Mbaku@ttu.edu
12345673	Jeremy	Wenzel	Jeremy.Wenzel@ttu.edu
12345674	Brianna	White	Brianna.White@ttu.edu
12345675	Rhett	Thompson	Rhett.Thompson@gmail.com
NULL	NULL	NULL	NULL

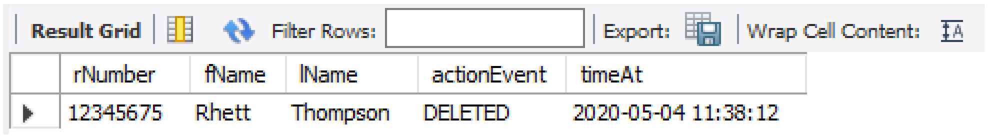
```
Output
Action Output
  #    Time      Action                                                                                                           Message
  599  11:38:01  #This trigger saves inserted/deleted students details into a log table CREATE TRIGGER student_deleted AFTER DELETE ON stud...  0 row(s) affected
  600  11:38:01  #These trigger saves added/dropped courses into a log table CREATE TRIGGER course_added AFTER INSERT ON course_stud...  0 row(s) affected
  601  11:38:12  SELECT * FROM groupproject.student LIMIT 0, 1000                                                                 6 row(s) returned
  602  11:38:12  DELETE FROM groupproject.student WHERE rNumber = 12345675                                                        1 row(s) affected
```

Student deleted from student table:



rNumber	fName	lName	email
12345670	Cameron	Gibson	cameron.gibson@ttu.edu
12345671	Anish	Chhetri	Anish.Chhetri@ttu.edu
12345672	Garret	Mbaku	Garret.Mbaku@ttu.edu
12345673	Jeremy	Wenzel	Jeremy.Wenzel@ttu.edu
12345674	Brianna	White	Brianna.White@ttu.edu
NULL	NULL	NULL	NULL

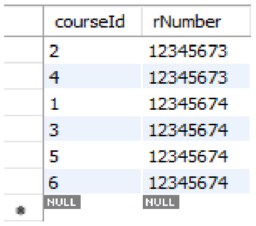
Student moved to log table:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

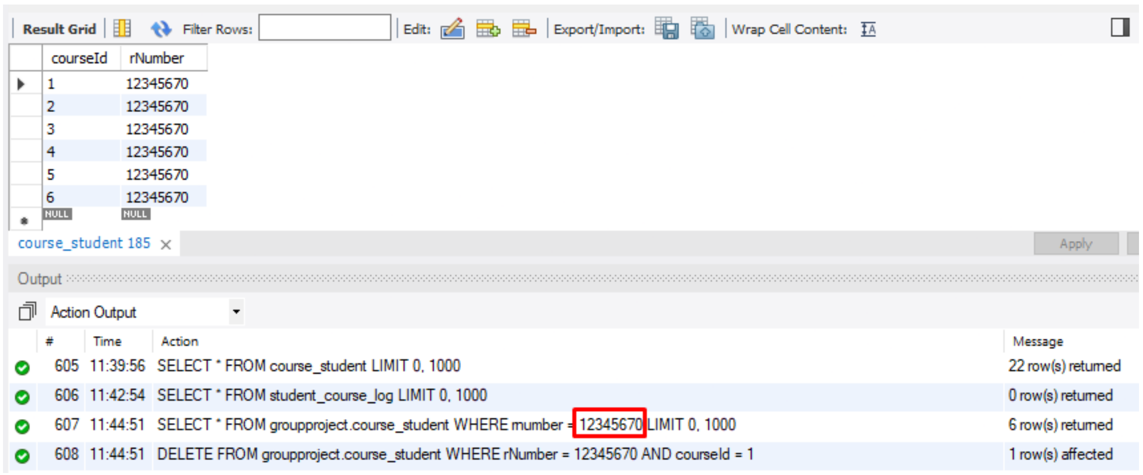
	rNumber	fName	lName	actionEvent	timeAt
▶	12345675	Rhett	Thompson	DELETED	2020-05-04 11:38:12

Student's enrolled classes dropped:



	courseId	rNumber
	2	12345673
	4	12345673
	1	12345674
	3	12345674
	5	12345674
	6	12345674
*	NULL	NULL

Testing of the course drop trigger, this view shows all of Cameron's classes and dropping the course with courseId set to one:



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	courseId	rNumber
▶	1	12345670
	2	12345670
	3	12345670
	4	12345670
	5	12345670
	6	12345670
*	NULL	NULL

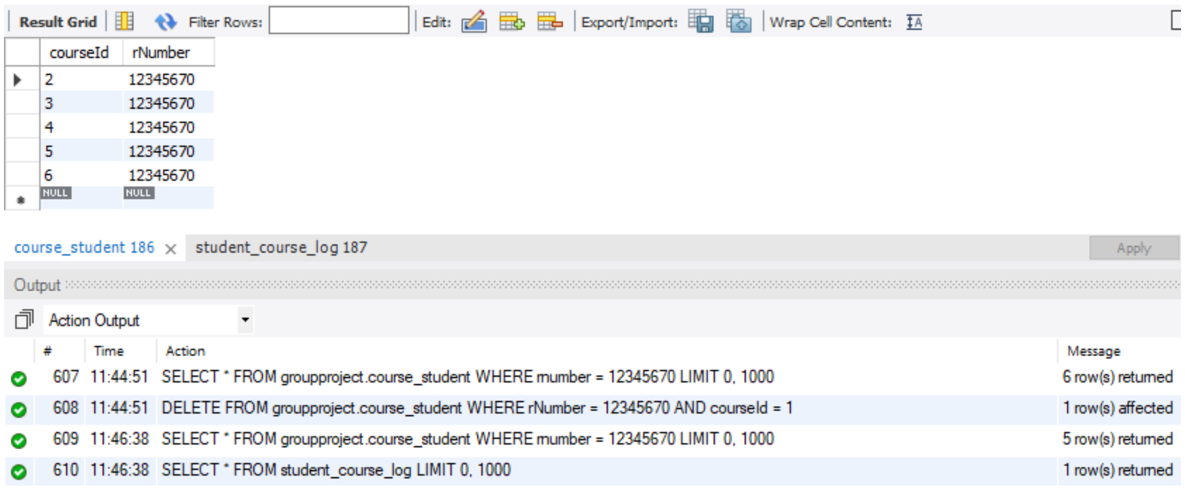
course_student 185 | Apply

Output: Action Output

	#	Time	Action	Message
✓	605	11:39:56	SELECT * FROM course_student LIMIT 0, 1000	22 row(s) returned
✓	606	11:42:54	SELECT * FROM student_course_log LIMIT 0, 1000	0 row(s) returned
✓	607	11:44:51	SELECT * FROM groupproject.course_student WHERE rnumber = 12345670 LIMIT 0, 1000	6 row(s) returned
✓	608	11:44:51	DELETE FROM groupproject.course_student WHERE rNumber = 12345670 AND courseId = 1	1 row(s) affected

After running the delete, you can see here that the class was dropped and added to the dropped courses table with the rNumber that dropped the class.

Cameron's enrolled classes:



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

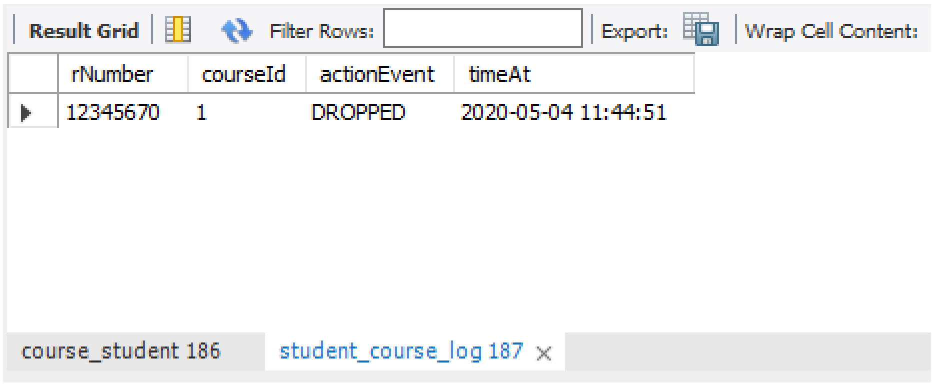
	courseId	rNumber
▶	2	12345670
	3	12345670
	4	12345670
	5	12345670
	6	12345670
*	NULL	NULL

course_student 186 | student_course_log 187 | Apply

Output: Action Output

	#	Time	Action	Message
✓	607	11:44:51	SELECT * FROM groupproject.course_student WHERE rnumber = 12345670 LIMIT 0, 1000	6 row(s) returned
✓	608	11:44:51	DELETE FROM groupproject.course_student WHERE rNumber = 12345670 AND courseId = 1	1 row(s) affected
✓	609	11:46:38	SELECT * FROM groupproject.course_student WHERE rnumber = 12345670 LIMIT 0, 1000	5 row(s) returned
✓	610	11:46:38	SELECT * FROM student_course_log LIMIT 0, 1000	1 row(s) returned

Dropped courses table:



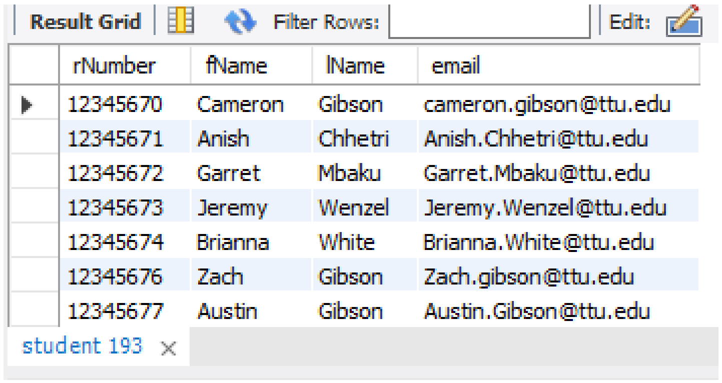
rNumber	courseId	actionEvent	timeAt
12345670	1	DROPPED	2020-05-04 11:44:51

Stored Procedure:

Our group has made a stored procedure to insert three students and the courses they're adding. We have built in error handling to handle SQL error 1062 which would be a duplicate rNumber, and 1048 which would be a null rNumber.

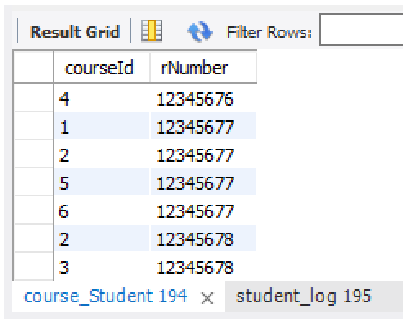
After calling the Stored Procedure, you can see the new students along with their details in the following tables.

Student Table:



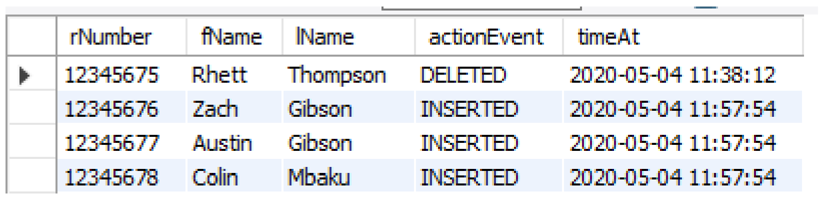
rNumber	fName	lName	email
12345670	Cameron	Gibson	cameron.gibson@ttu.edu
12345671	Anish	Chhetri	Anish.Chhetri@ttu.edu
12345672	Garret	Mbaku	Garret.Mbaku@ttu.edu
12345673	Jeremy	Wenzel	Jeremy.Wenzel@ttu.edu
12345674	Brianna	White	Brianna.White@ttu.edu
12345676	Zach	Gibson	Zach.gibson@ttu.edu
12345677	Austin	Gibson	Austin.Gibson@ttu.edu

Course_Student:



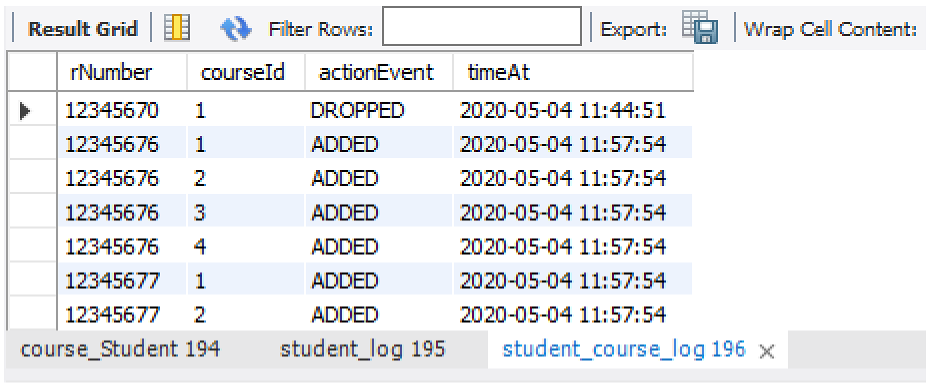
courseId	rNumber
4	12345676
1	12345677
2	12345677
5	12345677
6	12345677
2	12345678
3	12345678

Student Log:



	rNumber	fName	lName	actionEvent	timeAt
▶	12345675	Rhett	Thompson	DELETED	2020-05-04 11:38:12
	12345676	Zach	Gibson	INSERTED	2020-05-04 11:57:54
	12345677	Austin	Gibson	INSERTED	2020-05-04 11:57:54
	12345678	Colin	Mbaku	INSERTED	2020-05-04 11:57:54

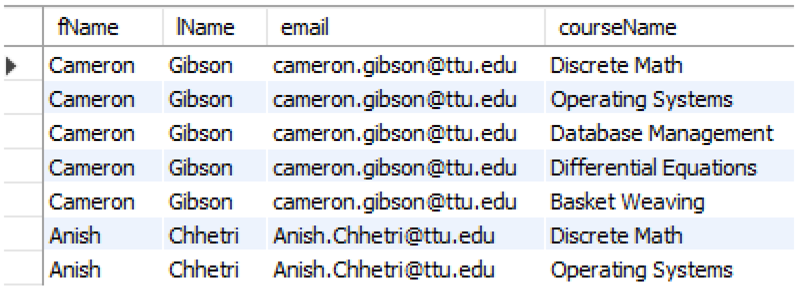
Course Log:



	rNumber	courseId	actionEvent	timeAt
▶	12345670	1	DROPPED	2020-05-04 11:44:51
	12345676	1	ADDED	2020-05-04 11:57:54
	12345676	2	ADDED	2020-05-04 11:57:54
	12345676	3	ADDED	2020-05-04 11:57:54
	12345676	4	ADDED	2020-05-04 11:57:54
	12345677	1	ADDED	2020-05-04 11:57:54
	12345677	2	ADDED	2020-05-04 11:57:54

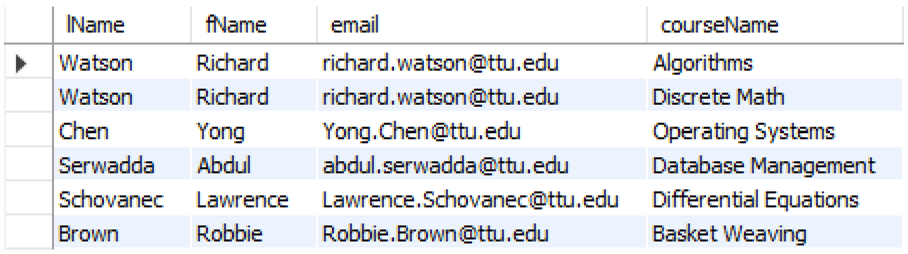
Views:

Our group has designed a variety of views to only show specific information, such as studentView to show the student's name, email and course names therefore hiding rNumber:



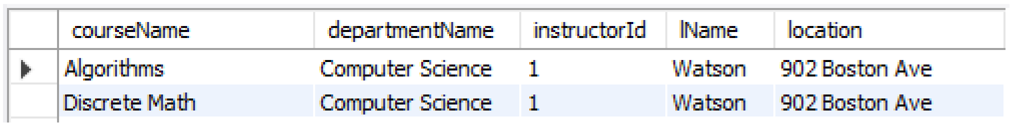
	fName	lName	email	courseName
▶	Cameron	Gibson	cameron.gibson@ttu.edu	Discrete Math
	Cameron	Gibson	cameron.gibson@ttu.edu	Operating Systems
	Cameron	Gibson	cameron.gibson@ttu.edu	Database Management
	Cameron	Gibson	cameron.gibson@ttu.edu	Differential Equations
	Cameron	Gibson	cameron.gibson@ttu.edu	Basket Weaving
	Anish	Chhetri	Anish.Chhetri@ttu.edu	Discrete Math
	Anish	Chhetri	Anish.Chhetri@ttu.edu	Operating Systems

Instructor view to see which classes an instructor teaches:



	lName	fName	email	courseName
▶	Watson	Richard	richard.watson@ttu.edu	Algorithms
	Watson	Richard	richard.watson@ttu.edu	Discrete Math
	Chen	Yong	Yong.Chen@ttu.edu	Operating Systems
	Serwadda	Abdul	abdul.serwadda@ttu.edu	Database Management
	Schovanec	Lawrence	Lawrence.Schovanec@ttu.edu	Differential Equations
	Brown	Robbie	Robbie.Brown@ttu.edu	Basket Weaving

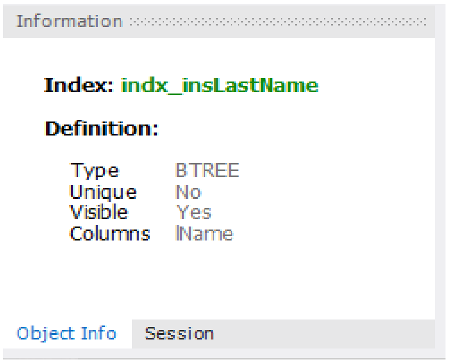
Course location to see all of the listed classes, departments, instructors id and corresponding locations:



courseName	departmentName	instructorId	lName	location
Algorithms	Computer Science	1	Watson	902 Boston Ave
Discrete Math	Computer Science	1	Watson	902 Boston Ave

Indexes:

Created indexes on course name, last name on student log, last name in student table, instructors last names, department locations, rNumbers and course Id for tables to enhance performance on data retrieval when running our queries. (On next Page)

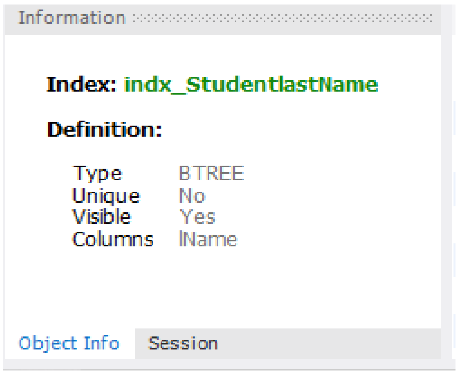


Information

Index: indx_insLastName

Definition:

Type	BTREE
Unique	No
Visible	Yes
Columns	lName

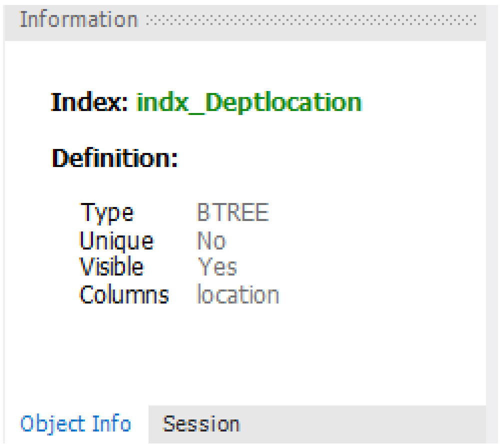


Information

Index: indx_StudentlastName

Definition:

Type	BTREE
Unique	No
Visible	Yes
Columns	lName

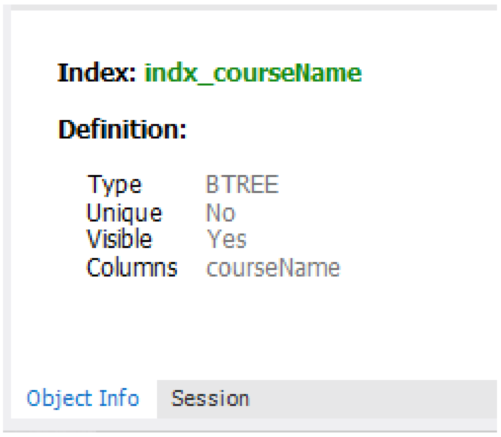


Information

Index: indx_Deptlocation

Definition:

Type	BTREE
Unique	No
Visible	Yes
Columns	location



Index: indx_courseName

Definition:

Type	BTREE
Unique	No
Visible	Yes
Columns	courseName

Contributions:

- Cameron Gibson: Created the tables, populated the tables, created the stored procedure and created the final document and slides.
- Jeremy Wenzel: Created the custom triggers and implementation of them. Showing examples on the affected tables, helped create the final document and slides.

- Rhett Calvert: Created custom views such as Student view which hides details such as rNumber, helped create the final document and slides.
- Garret Mbaku: Created the stored procedure implementation and examples in MySQL, also worked on index's
- Anish Chhetri: Helped create the idea for the student database and the basis problem, helped formulate the tables
- Oliver Jiang: We never heard from him. We just assume he dropped the course. Has contributed nothing to this document.