AAiT
ADDIS ABABA INSTITUTE OF TECHNOLOGY
አዲስ አበባ ቴክኖሎጂ ኢንስቲትዩት
ADDIS ABABA UNIVERSITY
አዲስ አበባ ዩኒቨርሲቲ

Department of Information Technology & Scientific Computing (Software Engineering)

Course name: - Big Data Analysis
Lab Report - I

Prepared By: -

Abdulhamid Abdo	ATR/8526/07
Abrham Kindu	ATR/6689/07
Samuel Mussie	ATR/9258/07

Submitted to: - Alazar Alemayehu

Dec 2018

ABSTRACT

Connecting to a database from the R environment can be extremely useful. It allows us direct access to the data without have to first export it from a database and then import it from a csv file or entering it directly into R. This assignment documentation will showcase the processes and steps to set up R to connect to a database. This documentation will show step-by-step instructions and provides the R code on how to query the database once connectivity is established.

Introduction

We all know that R is a very powerful when working with big data. We can manipulate and deal with small datasets that easily fit into your computer's memory. But what if we encounter datasets that are too large for your computer to handle as a whole? In this case, storing the data outside of R and organizing it in a database will become handy. Connecting to the database allows you to retrieve only the chunks needed for the current analysis. R can connect to almost any existing database type. Most common database types have R packages that allow you to connect to them. Examples could be RSQLite, RMySQL, RODBC, etc.). We will be using RODBC to connect to our database and perform CRUD operations on our database.

Problem Description

Performing CRUD (Create, Read, Update, Delete) operations using R to a database.

(A) Design a simple database containing 3 tables with relationships.

- *We have created a database called 'BIGDATA' with three tables namely 'USER', 'GRADES', and 'ADDRESS'.*
- *USER table basic information of a user which has 'id', 'first_name', 'last_name' and 'sex' attributes.*
- *GRADES table contains information about a user's grade and has 'id', 'user_id','course_name' and 'course_grade' attributes. It has 'many to many' relationship with table 'USER' as many grades can be possessed by a user.*
- *ADDRESS table contains information about a user's address and has 'id', 'user_id', 'house_no', 'region' and 'city' attributes. It has 'one to one' relationship with table 'USER' assuming a user has only one address to live on.*

The ER diagram for the database is shown below.

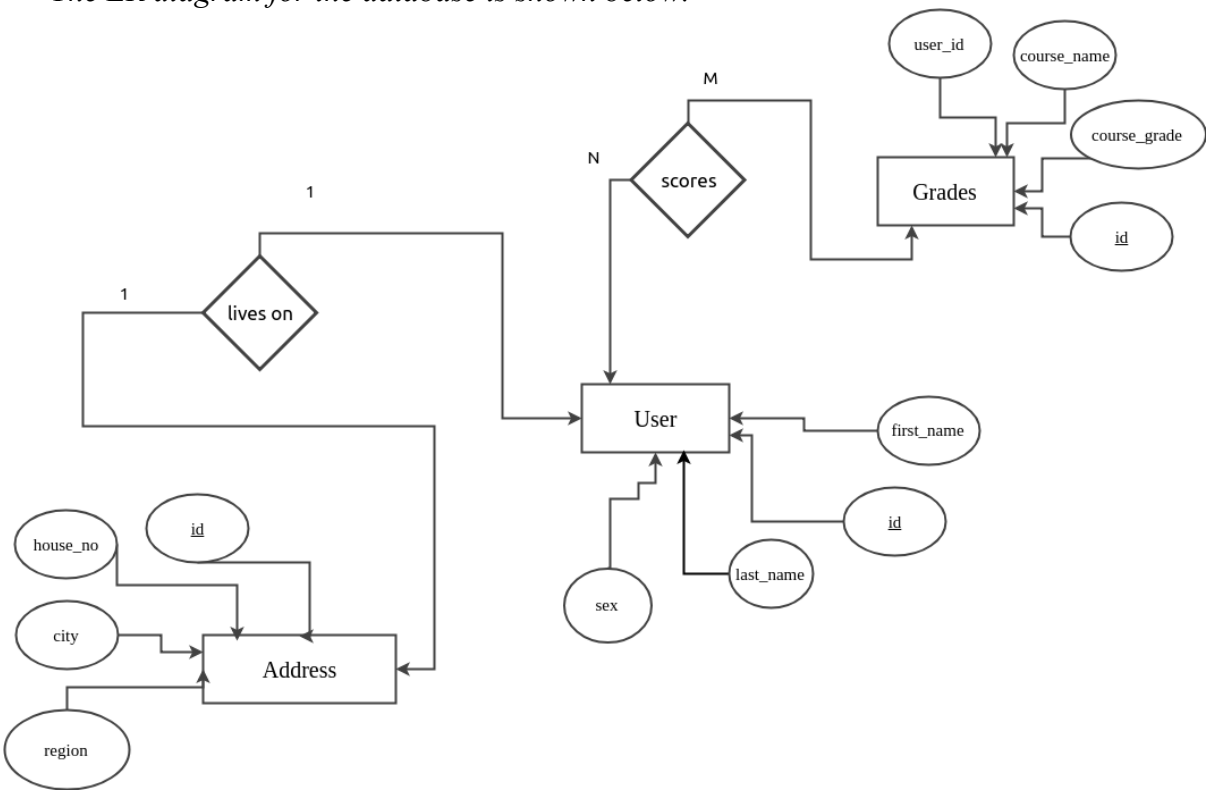


Figure 1 ER diagram for database BIGDATA

(B, C, D & E) Create a table from R program and demonstrate CRUD operations.

- *We will be showing step by step processes of creating the tables by using screenshots.*

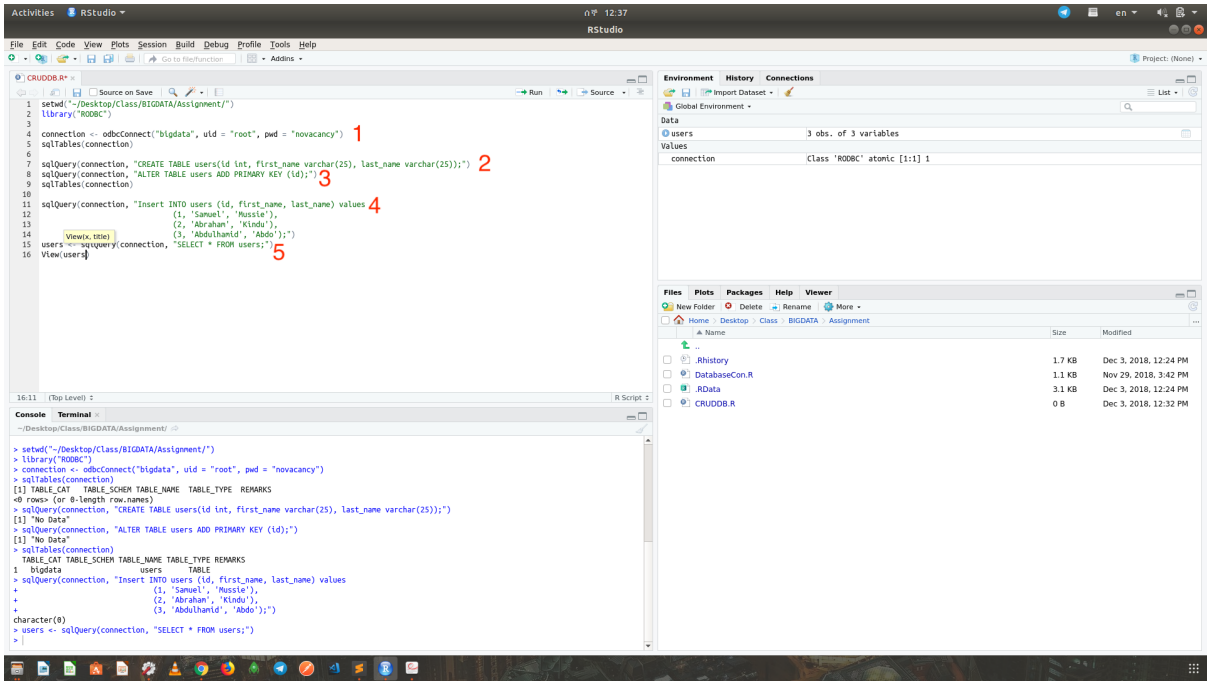
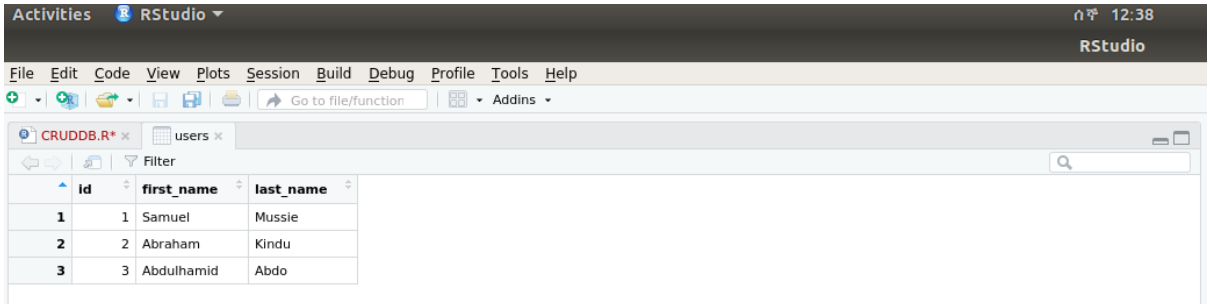


Figure 2 Creating and adding fields to a table using R.

- *The above screenshot shows the steps to create a table and add fields to a table using R inside RStudio.*
- *What the code numbered as 1 on the screenshot does is connecting to a database named 'bigdata' using the provided credentials. The code designated as 2 creates a table named 'users' with the provided attributes. Number 3 adds a 'PRIMARY KEY' 'id' to table 'users'. Number 4 inserts data in to table 'users'. And finally, number 5 assigns the result from the select statement to a variable user in order to view it inside RStudio. The result from executing code number 5 is shown in the figure below.*



	id	first_name	last_name
1	1	Samuel	Mussie
2	2	Abraham	Kindu
3	3	Abdulhamid	Abdo

Figure 3 Executing View(users).

- *Executing update operation.*

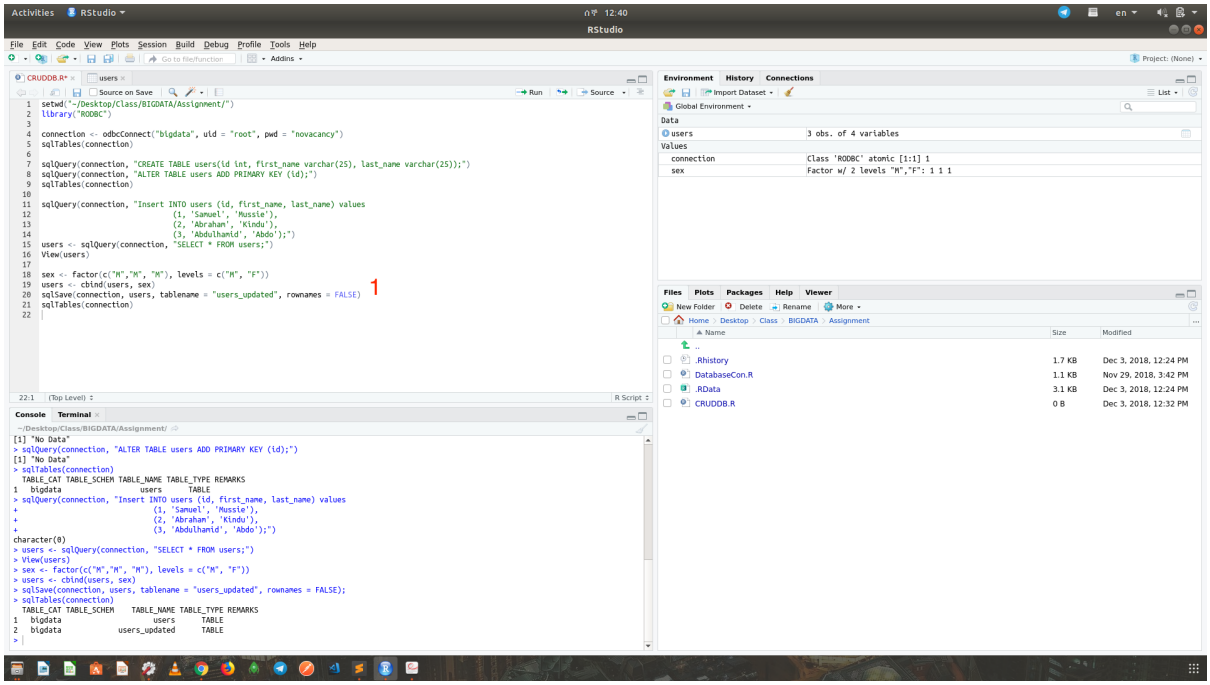


Figure 4 update operation.

- *The codes marked as 1 will factor and add a 'sex' attribute. Then cbind is called to bind 'sex' attribute and then we add it as a column in 'user_updated' table. The contents of the table are shown in the figure below.*

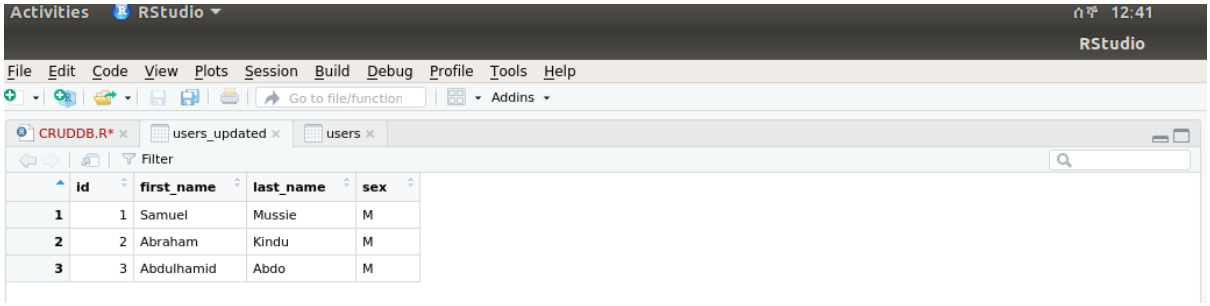


Figure 5 Viewing the updated table.

- In similar manners, we create table ‘addresses’ and ‘grades’.
  - Creating table addresses.

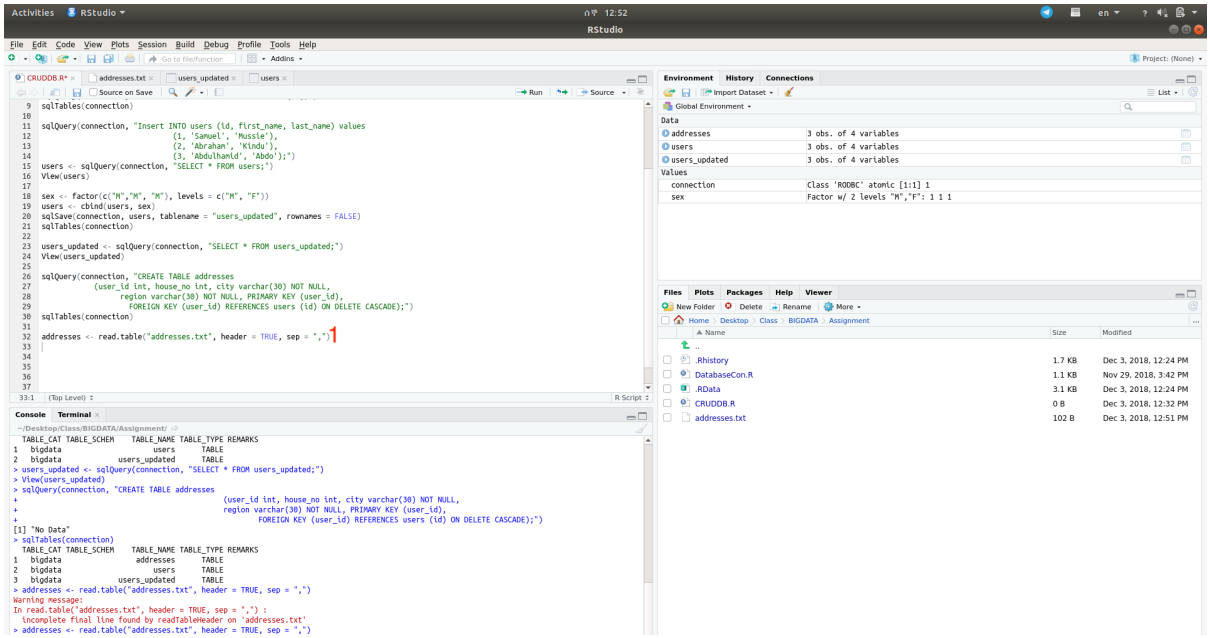


Figure 6 Creating and inserting data to table addresses.

- Code number 1 reads addresses.txt and adds them to table address. The contents of addresses.txt is shown in the figure below.

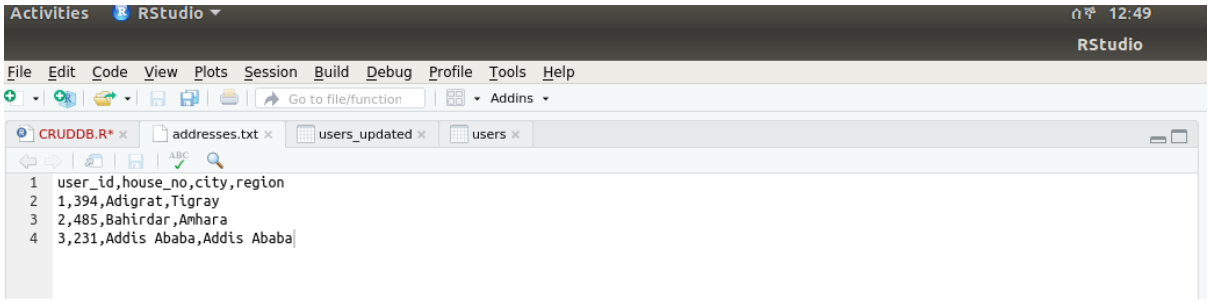


Figure 7 Addresses.txt

- Creating table ‘grades’.

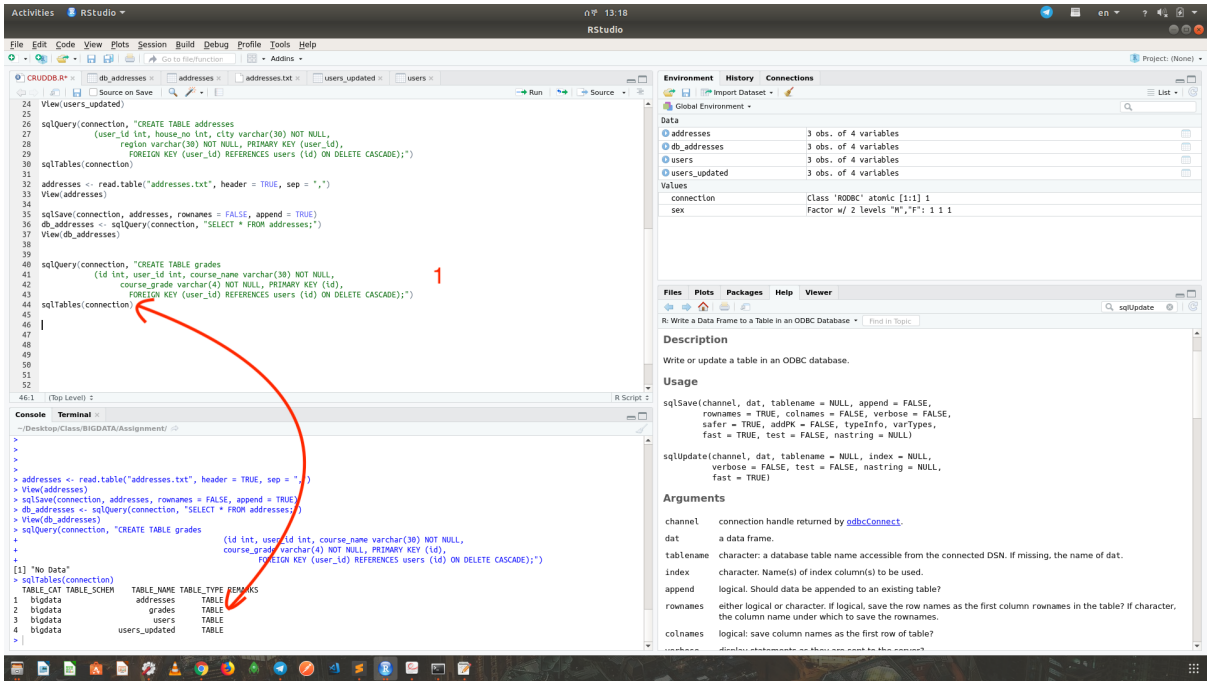


Figure 8 Creating table grades.

- Code numbered as 1 creates a table named grades with the provided attributes. We can see that, after the execution of sqltables(connection), a new table named grades is added as shown in the console.
- Adding and updating table grades.

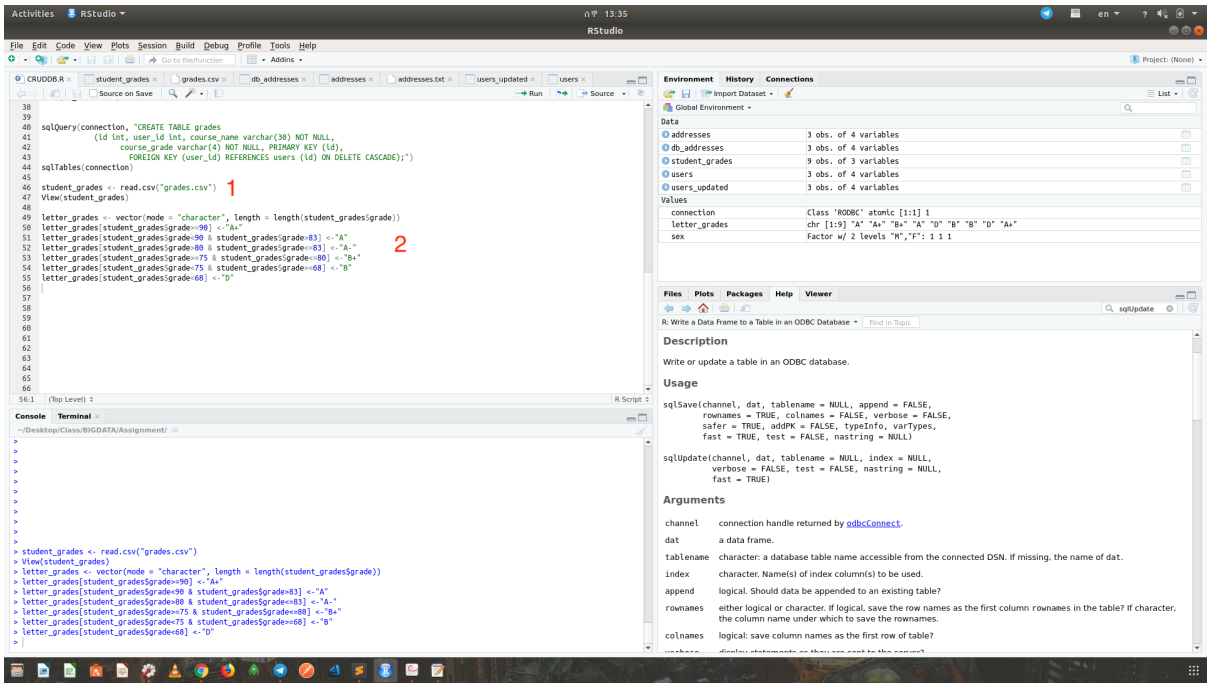
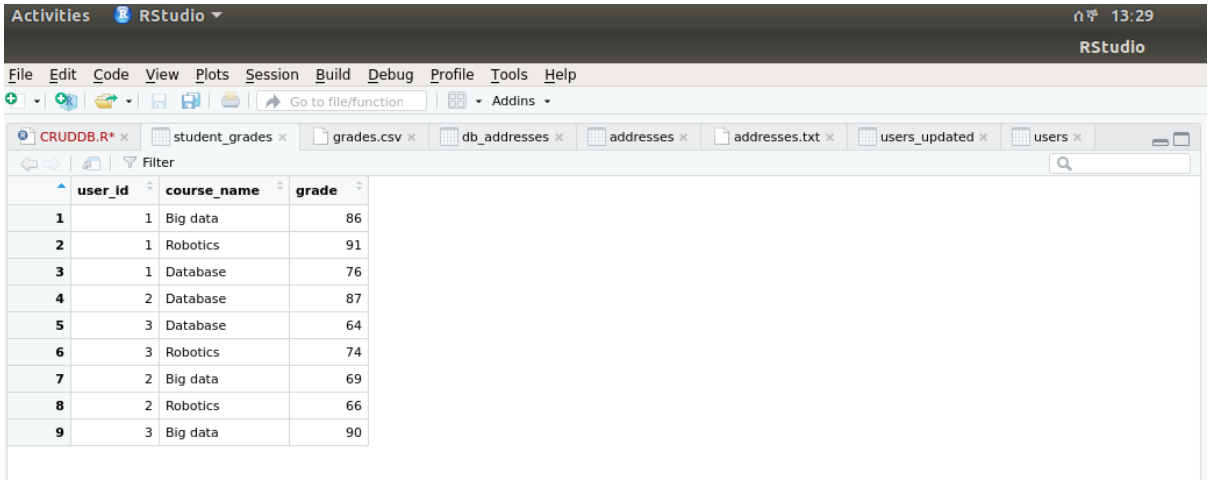


Figure 9 adding and updating table grades.

- Code numbered as 1 reads grades from file grades.csv and assigns it to a variable named student_grades. Codes designated as 2 filters student grades and assigns letter grades based on grade values.

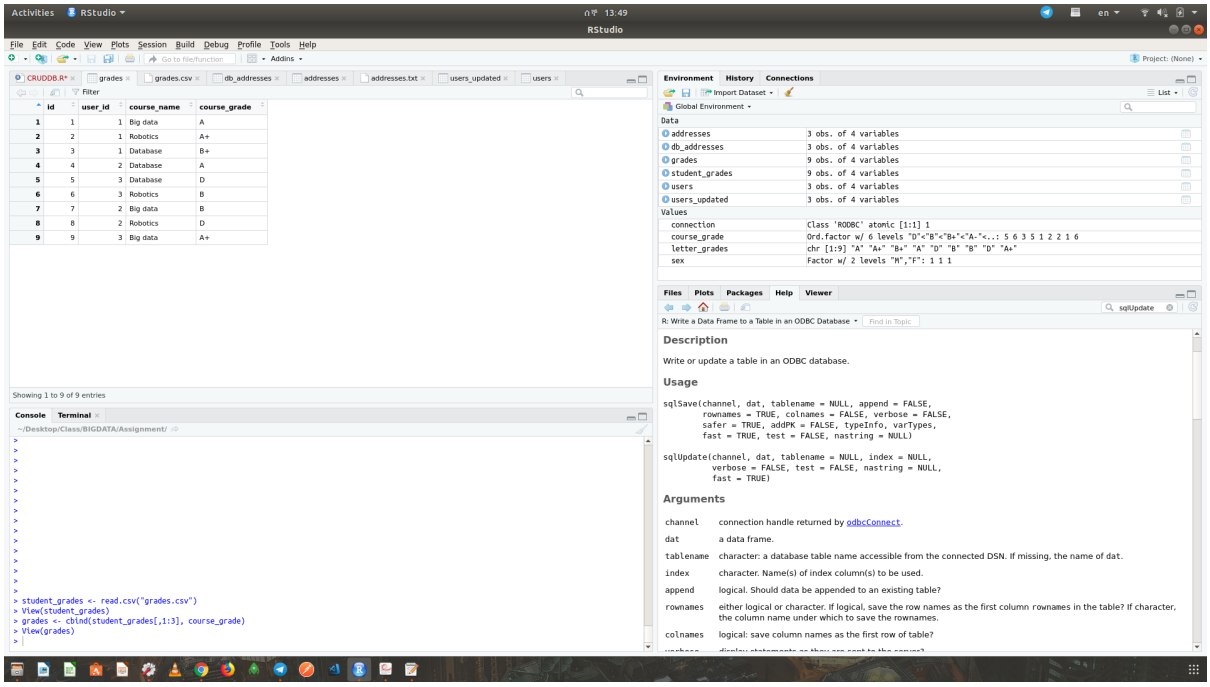
- Contents of student_grades before adding letter grades are shown in the figure below.



	user_id	course_name	grade
1	1	Big data	86
2	1	Robotics	91
3	1	Database	76
4	2	Database	87
5	3	Database	64
6	3	Robotics	74
7	2	Big data	69
8	2	Robotics	66
9	3	Big data	90

Figure 10 Student_grades

- The content of grades table is shown in the figure below.



	id	user_id	course_name	course_grade
1	1	1	Big data	A
2	2	1	Robotics	A+
3	3	1	Database	B+
4	4	2	Database	A
5	5	3	Database	D
6	6	3	Robotics	B
7	7	2	Big data	B
8	8	2	Robotics	D
9	9	3	Big data	A+

Figure 11 Grades with letter grades

- *Update and delete operations*

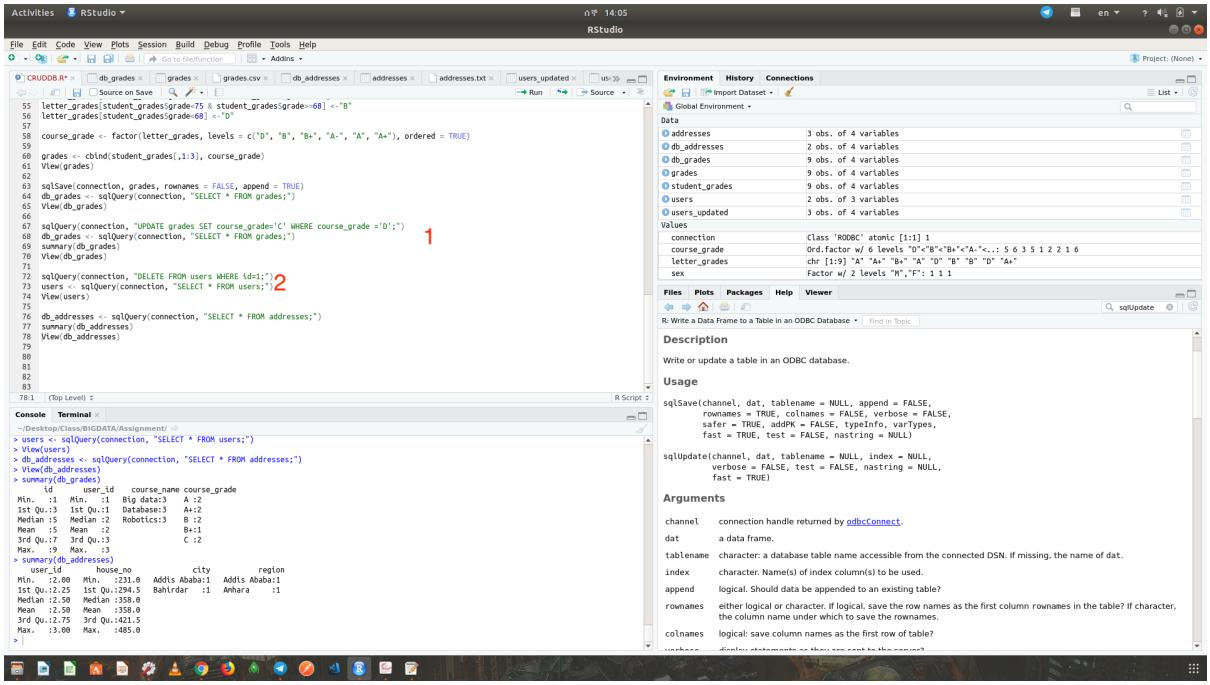


Figure 12 Update and delete operations.

- The code numbered as 1 updates tables grades by looking for course_grades with values ‘D’ and setting them to value ‘C’.
- The code numbered as 2 deletes a user with id 0f 1.
- Updated table and deleted table are shown respectively.

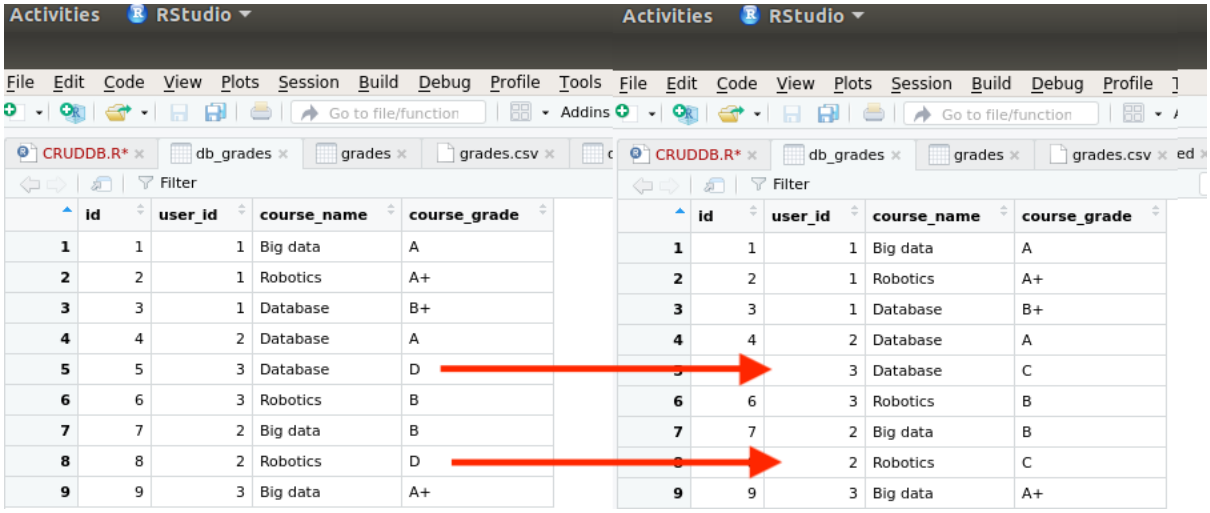


Figure 13 Updated grades table values

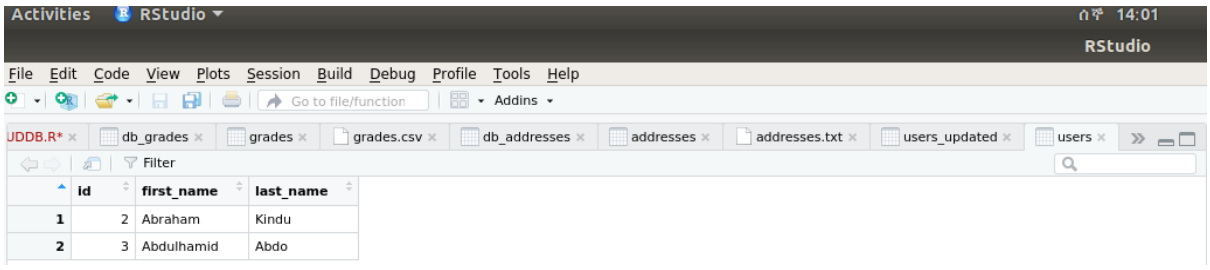


Figure 14 Deleted User with id 1

- *Showing cascading delete from the grades table too. No grade is shown with user id of 1 as user with an id if 1 is deleted his grades will also be automatically deleted.*

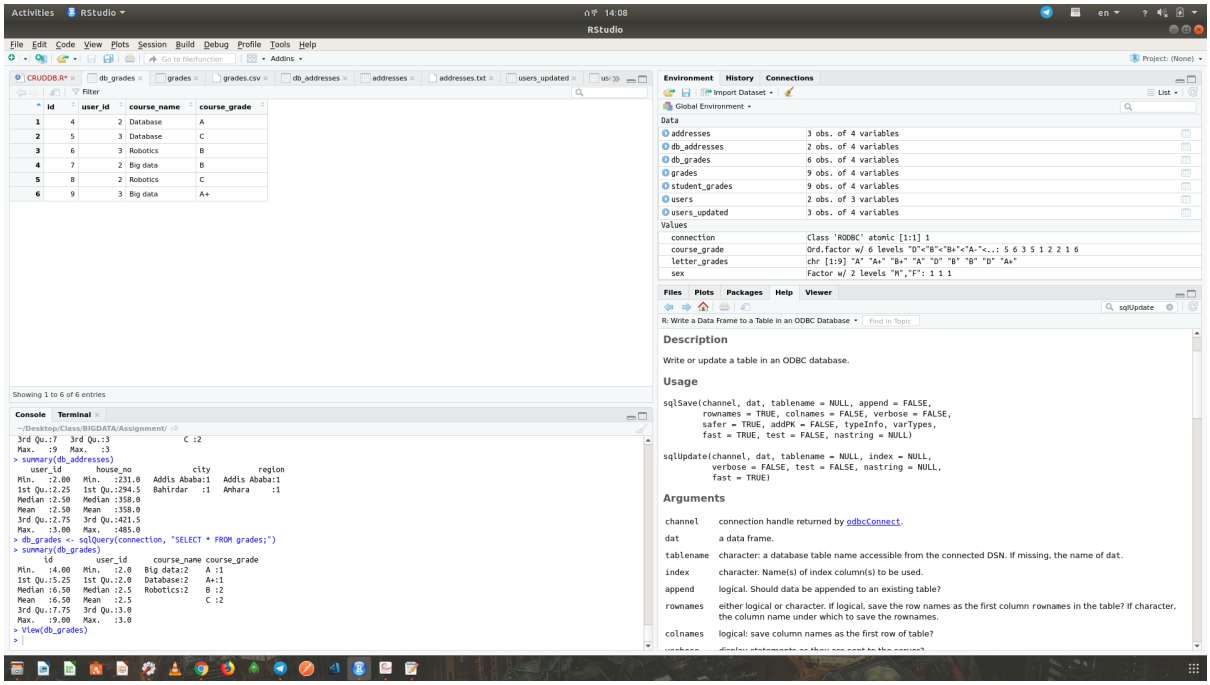


Figure 15 Demonstrating cascading delete