Assignment No: 11

**Title:** Write a program that interacts with the weather database. Find the day and the station with the maximum snowfall in 2013

**Theory:**

RSQLite is the easiest way to use a database from R because the package itself contains [SQLite](https://www.sqlite.org/index.html); no external software is needed. This vignette will walk you through the basics of using a SQLite database.

RSQLite is a DBI-compatible interface which means you primarily use functions defined in the DBI package, so you should always start by loading DBI, not RSQLite:

Library(DBI)

**Creating a new database**

To create a new SQLite database, you simply supply the filename to dbConnect():

mydb <- dbConnect(RSQLite::SQLite(), "my-db.sqlite")

dbDisconnect(mydb)

unlink("my-db.sqlite")

If you just need a temporary database, use either "" (for an on-disk database) or ":memory:" or "file::memory:" (for a in-memory database). This database will be automatically deleted when you disconnect from it.

mydb <- dbConnect(RSQLite::SQLite(), "") dbDisconnect(mydb)

**dbSendQuery():**

* The dbSendQuery() method only submits and synchronously executes the SQL query to the database engine. It does not extract any records --- for that you need to use the dbFetch() method, and then you must call dbClearResult() when you finish fetching the records you need. For interactive use, you should almost always prefer dbGetQuery()
* dbSendQuery() returns an S4 object that inherits from DBIResult. The result set can be used with dbFetch() to extract records. Once you have finished using a result, make sure to clear it with dbClearResult(). An error is raised when issuing a query over a closed or invalid connection, or if the query is not a non-NA string. An error is also raised if the syntax of the query is invalid and all query parameters are given (by passing the params argument) or the immediate argument is set to TRUE. Details

**Program:**

library(RSQLite)

library(DBI)

db<-dbConnect(SQLite(),dbname="test3")

dbSendQuery(conn=db,"CREATE TABLE weather\_real(year INT,day TEXT,station TEXT,snowfall INT,temp INT)")

dbSendQuery(conn=db,"INSERT INTO weather\_real (year,day,station,snowfall,temp )VALUES(2014,'tuesday','aa',10,30),(2015,'friday','bb',10,20),(2013,' saturday','cc',9,10),(2015,'tuesday','rr',22,30),(2013,'tuesday','bb' ,16,30),(2011,'saturday','oo',10,30),(2018,'tuesday','aa',10,23),(2017,'tuesday','aa',20,20),(2016,'friday','aa',18,12),(2018,'monday','aa ',10,21),(2013,'sunday','aa',10,24),(2016,'tuesday','pp',10,35)")

dbListTables(db)

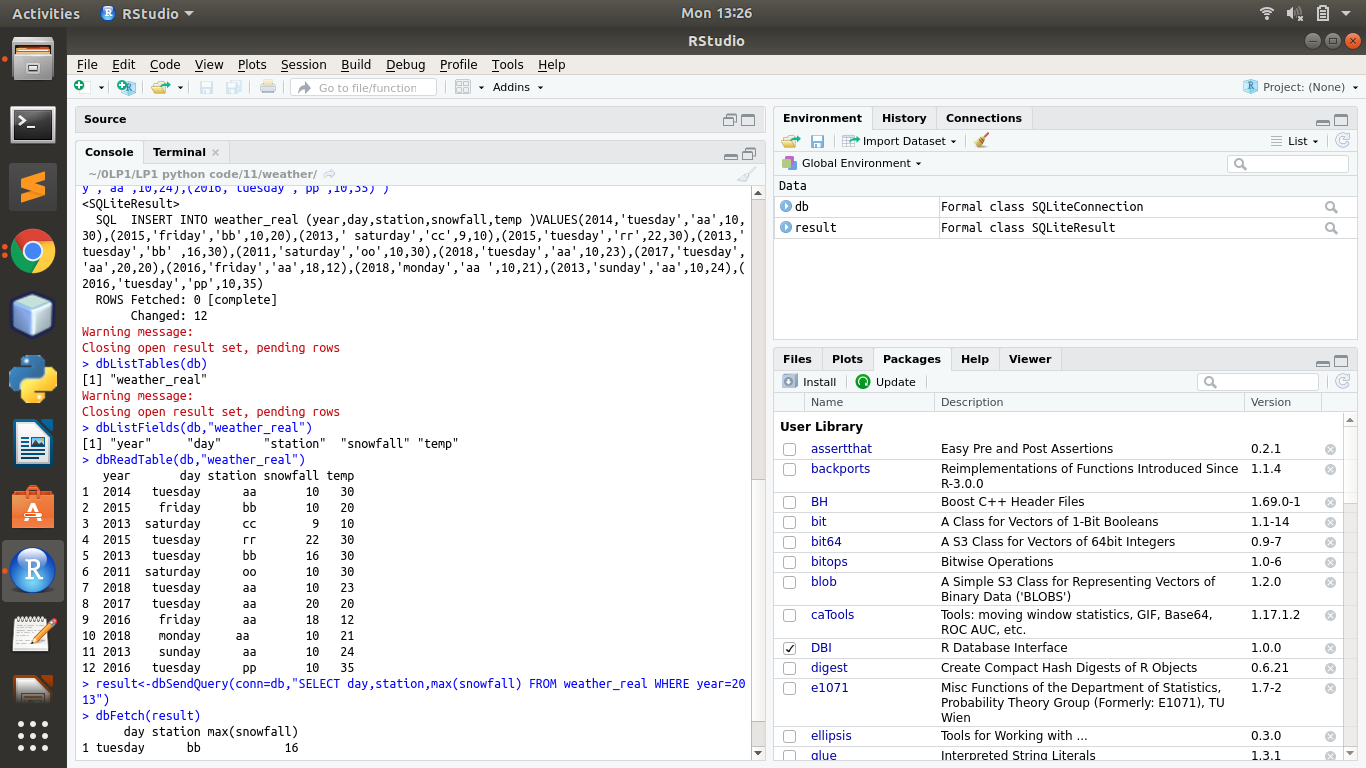
dbListFields(db,"weather\_real")

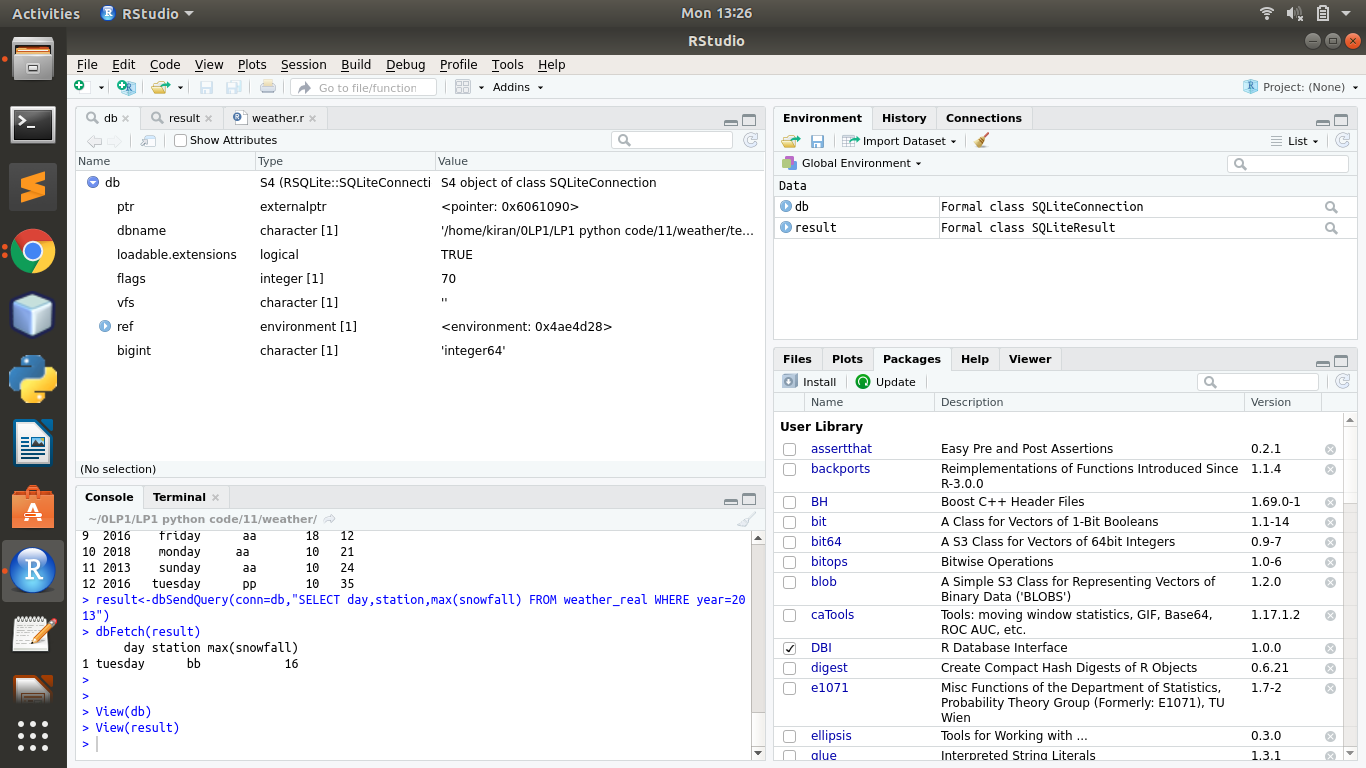
dbReadTable(db,"weather\_real")

result<-dbSendQuery(conn=db,"SELECT day,station,max(snowfall) FROM weather\_real WHERE year=2013")

dbFetch(result)

**Output:**

****

****