Easy - Creating SQLite Database and populating dummy data from csv files

The database is named university.db and contains two tables, namely “Students” and “Enrolled”. Students has a list of students with their roll numbers and names. Enrolled has a row for every course a student is enrolled in.

```
library(RSQLite)
conn<-dbConnect(RSQLite::SQLite(), 'university.db')
students<- read.csv('Student.csv')
enrolled<- read.csv('Enrolled.csv')
dbWriteTable(conn, "Students", students, fileEncoding = "UTF-8")
dbWriteTable(conn, "Enrolled", enrolled, fileEncoding = "UTF-8")
dbListTables(conn)
```

```
## [1] "Enrolled" "Students"
```

A glimpse at the tables:

```
head(students, 10)
```

```
##    roll_no                name
## 1        1            qhgvlwtqk
## 2        2            niuughyhf
## 3        3              yddxlq
## 4        4     bdfaagposdiqgmi
## 5        5        znvvrdvsmpbj
## 6        6               fcmpu
## 7        7                okcl
## 8        8          jiorfkvgyio
## 9        9              raasmv
## 10      10 shddxznwfizmnznugb
```

```
head(enrolled, 5)
```

```
##   roll_no class_name
## 1       1     CSE455
## 2       2     CSE455
## 3       3     CSE455
## 4       4     CSE455
## 5       5     CSE455
```

Medium - Function with multiple search parameters

The following function takes multiple search parameters, creates a query and filters out results on the basis of that query. The names of variables are fairly intuitive and their function are commented. All variables except conn and distinct take vector inputs. Some clarifications -

1. ordering takes vector input with values “asc” or “desc” depending on how user wants to sort on the basis of respective column
2. ordering_parameters is a vector of column names on which sorting is applied.
3. check_null is a vector input with values TRUE or FALSE. TRUE means the respective column would be checked with IS NULL. FALSE means it will be checked for IS NOT NULL

The function works by separately generating query parts for SELECT, FROM and WHERE statements and then pasting them together finally in the variable “result_query”.

```
search <- function(conn = NULL, #connection to SQLite database
                   distinct = FALSE, # adds "DISTINCT" in the sql query if true
                   display_columns = NULL, # columns to be selected
```

```
                  table_names = NULL, # tables from which the columns should be selected
                  search_parameters = NULL, # column names on the basis of which
                                            # to filter out results
                  search_parameters_values = NULL, # values for the respective search
                                                   # parameters
                  search_parameters_operators = NULL, # supports >, <, ==, >=, <=
                  ordering = NULL, # adds ORDER BY statement to query to sort the results
                  ordering_parameters = NULL, # columns by which to sort
                  check_null=NULL, # adds a NOT NULL or NULL statement to the query
                  check_null_parameters = NULL # columns which should be tested
                                               # for IS NULL or IS NOT NULL
                  ){

 select_query<-""
if(isTRUE(distinct))
  select_query<-"SELECT DISTINCT "
else
  select_query<-"SELECT "

from_query<-"FROM "
where_query<-"WHERE "
ordering_query<-"ORDER BY "
result_query<-""

for(i in 1:length(display_columns)){
  if(i!=length(display_columns))
    select_query<- paste(select_query, display_columns[i], ", ", sep = "")
  else
    select_query<- paste(select_query, display_columns[i], " ", sep = "")
}

for(i in 1:length(table_names)){
  if(i!=length(table_names))
    from_query<- paste(from_query, table_names[i], ", ", sep = "")
  else
    from_query<- paste(from_query, table_names[i], " ", sep = "")
}


for(i in 1:length(search_parameters)){
  if(i!=length(search_parameters))
    where_query<- paste(where_query, search_parameters[i], " ",
                        search_parameters_operators[i], " ",
                        search_parameters_values[i], " AND ", sep = "")
  else
    where_query<- paste(where_query, search_parameters[i], " ",
                        search_parameters_operators[i], " ",
                        search_parameters_values[i], sep = "")
}

if(length(search_parameters!=0)&&length(check_null)!=0)
  where_query<- paste(where_query, "AND ")
```

```r
  for(i in 1:length(check_null)){
    if(i!=length(check_null)){
      if(check_null[i] == TRUE)
        where_query<- paste(where_query, check_null_parameters[i], " IS NULL AND ", sep = "")
      else
        where_query<- paste(where_query, check_null_parameters[i], " IS NOT NULL AND ", sep = "")
    }
    else{
      if(check_null[i] == TRUE)
        where_query<- paste(where_query, check_null_parameters[i], " IS NULL ", sep = "")
      else
        where_query<- paste(where_query, check_null_parameters[i], " IS NOT NULL ", sep = "")
    }
  }

  for(i in 1:length(ordering)){
    if(i!=length(ordering))
      ordering_query<- paste(ordering_query, ordering_parameters[i],
                             " ", ordering[i], ", ", sep = "")
    else
      ordering_query<- paste(ordering_query, ordering_parameters[i],
                             " ", ordering[i], sep = "")
  }

  result_query<- paste(select_query, from_query, where_query, ordering_query, sep = "\n")
  cat(paste("The resulting query created is:", result_query, sep = "\n"))
  result <- dbGetQuery(conn, result_query)
  return(result)
}
```

Example for filtering out results

Defining the parameters -

```r
display_columns<- c('roll_no', 'name')
table_names<- c('students')
search_parameters<- c('roll_no')
search_parameters_values<- c(5)
search_parameters_operators<- c('>=')
ordering = c('DESC')
ordering_parameters = c('roll_no')
check_null<-c(FALSE)
check_null_parameters<-c('name')
```

Results of the query -

```r
res<- search(conn = conn,
                   distinct = TRUE,
                   display_columns = display_columns,
                   table_names = table_names,
                   search_parameters = search_parameters,
                   search_parameters_values = search_parameters_values,
                   search_parameters_operators = search_parameters_operators,
                   ordering = ordering,
                   ordering_parameters = ordering_parameters,
                   check_null = check_null,
                   check_null_parameters = check_null_parameters)
```

```
## The resulting query created is:
## SELECT DISTINCT roll_no, name
## FROM students
## WHERE roll_no >= 5 AND name IS NOT NULL
## ORDER BY roll_no DESC
```

```
res
```

```
##   roll_no               name
## 1      10 shddxznwfizmnznugb
## 2       9             raasmv
## 3       8        jiorfkvgyio
## 4       7               okcl
## 5       6              fcmpu
## 6       5       znvvrdvsmpbj
```