STAT 240 - Assignment 3

Problem 1

```
query1 = "SELECT COUNT(DISTINCT year) AS DistinctYears FROM WinterO"
dbGetQuery(dbcon, query1)
```

```
##   DistinctYears
## 1            21
```

Problem 2

```
##    Height_m
## 1     14.50
## 2      9.19
## 3      8.79
## 4      7.01
## 5      6.91
## 6      6.50
## 7      6.20
## 8      5.79
## 9      5.41
## 10     5.21
## 11     5.00
## 12     4.50
## 13     4.19
## 14     3.99
## 15     3.81
## 16     3.71
## 17     3.51
## 18     3.30
## 19     3.20
## 20     3.00
## 21     2.90
## 22     2.79
## 23     2.69
## 24     2.59
## 25     2.49
## 26     2.39
## 27     2.31
## 28     2.21
## 29     2.11
## 30     2.01
## 31     1.91
## 32     1.80
## 33      1.70
## 34      1.60
## 35      1.50
## 36      1.40
## 37      1.30
## 38      1.19
## 39      1.09
## 40      0.99
## 41      0.89
## 42      0.84
## 43      0.79
## 44      0.71
## 45      0.61
## 46      0.51
## 47      0.41
## 48      0.30
## 49      0.20
## 50      0.10
```

Problem 3

a

```
query3 = "SELECT * FROM CA"
system.time(for(i in 1:10000) {dbSendQuery(dbcon, query3)} )
```

```
##    user  system elapsed
##   3.732   0.198   3.978
```

b

```
query_out3 = dbSendQuery(dbcon, query3)
system.time(for(i in 1:10000) {dbFetch(query_out3, 1)} )
```

```
##    user  system elapsed
##   0.738   0.008   0.755
```

```
dbClearResult(query_out3)
```

c

'dbFetch' method was much faster than 'dbSendQuery'. And that's because 'dbfetch' gets a single row at a time from the query result whereas 'dbSendQuery' sends the query each time.

Problem 4

The LIKE operator matches the given pattern with strings in the rows of the query result. Special characters such as % , _ , and \ along with other arbitrary characters are used to specify the requirements. For example, % character stands for 0 or more arbitrary characters, _ character stands for exactly 1 arbitrary characters, and \ character is used to escape the special characters % and _ .

Selects all rows from table CA where column Geographic_names /postal code are 3 letters long and starts with 'V4' and then one single character.

```
query41 = "SELECT * FROM CA WHERE Geographic_name LIKE 'V4_'"
query_out41 = dbSendQuery(dbcon, query41)
dbFetch(query_out41, 5)
```

```
##    ID Country Geographic_name             Region         Province Prov_acr
## 1 230      CA             V4A   Surrey Southwest British Columbia       BC
## 2 231      CA             V4B         White Rock British Columbia       BC
## 3 232      CA             V4C    Delta Northeast British Columbia       BC
## 4 233      CA             V4E         Delta East British Columbia       BC
## 5 234      CA             V4G Delta East Central British Columbia       BC
##   Latitude Longitude Region_Index
## 1  49.0374 -122.8299            1
## 2  49.0259 -122.8058            4
## 3  49.1551 -122.9124            1
## 4  49.1197 -122.9056            1
## 5  49.1448 -122.9950            1
```

```
dbClearResult(query_out41)
```

Selects all rows from table CA where column Region has brackets '()' in the Region / city names.

```
query42 = "SELECT * FROM CA WHERE Region LIKE '%(%)'"
query_out42 = dbSendQuery(dbcon, query42)
dbFetch(query_out42, 5)
```

```
##   ID Country Geographic_name                             Region Province
## 1  1      CA             T0A         Eastern Alberta (St. Paul)  Alberta
## 2  2      CA             T0B        Wainwright Region (Tofield)  Alberta
## 3  3      CA             T0C         Central Alberta (Stettler)  Alberta
## 4  4      CA             T0E           Western Alberta (Jasper)  Alberta
## 5  5      CA             T0G North Central Alberta (Slave Lake)  Alberta
##   Prov_acr Latitude Longitude Region_Index
## 1       AB  54.7660 -111.7174           NA
## 2       AB  53.0727 -111.5816           NA
## 3       AB  52.4922 -112.8113           NA
## 4       AB  53.4021 -117.2308           NA
## 5       AB  55.6993 -114.4529           NA
```

```
dbClearResult(query_out42)
```