```
rm(list=ls())
```

```
library(DBI)
```

```
## Warning: package 'DBI' was built under R version 3.6.3
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
con <- dbConnect(RSQLite::SQLite(), "dalt_database.db")
```

```
#list of tables in database
dbListTables(conn = con)
```

```
## [1] "broadband" "house"
```

```
broadband <- tbl(con, "broadband")

house <- tbl(con,"house")
```

```
#data in house table
glimpse(house)
```

```
## Observations: ??
## Variables: 103
## Database: sqlite 3.33.0 [C:\Users\bindu\Desktop\SQLite_Submission\dalt_database.db]
## $ `Region/Country_code` <chr> "E12000001", "E12000001", "E12000001", "E1200...
## $ `Region/Country_name` <chr> "North East", "North East", "North East", "No...
## $ Local_authority_code_ <chr> "E06000001", "E06000002", "E06000003", "E0600...
## $ Local_authority_name  <chr> "Hartlepool", "Middlesbrough", "Redcar and Cl...
## $ Year_ending_Dec_1995  <chr> "36500", "40000", "43000", "45250", "44000", ...
## $ Year_ending_Mar_1996  <chr> "37500", "41500", "42500", "46000", "43000", ...
## $ Year_ending_Jun_1996  <chr> "38500", "42000", "43563", "46950", "44000", ...
## $ Year_ending_Sep_1996  <chr> "39000", "43000", "44000", "47000", "43950", ...
## $ Year_ending_Dec_1996  <chr> "40000", "43000", "43500", "48000", "43500", ...
## $ Year_ending_Mar_1997  <chr> "40975", "43500", "44000", "48500", "44700", ...
## $ Year_ending_Jun_1997  <chr> "41752", "44000", "44000", "49500", "45500", ...
## $ Year_ending_Sep_1997  <chr> "41500", "45000", "44995", "49900", "46950", ...
## $ Year_ending_Dec_1997  <chr> "41450", "45000", "45000", "49500", "49000", ...
## $ Year_ending_Mar_1998  <chr> "41450", "45000", "45000", "49000", "49500", ...
## $ Year_ending_Jun_1998  <chr> "43000", "43000", "46000", "49950", "49950", ...
## $ Year_ending_Sep_1998  <chr> "44250", "42000", "45950", "49995", "50000", ...
## $ Year_ending_Dec_1998  <chr> "44000", "42513", "46000", "50000", "49995", ...
## $ Year_ending_Mar_1999  <chr> "44950", "43000", "45950", "51000", "49950", ...
## $ Year_ending_Jun_1999  <chr> "45375", "44850", "45000", "51500", "50000", ...
## $ Year_ending_Sep_1999  <chr> "46000", "46500", "45000", "52500", "50000", ...
## $ Year_ending_Dec_1999  <chr> "47375", "46000", "46000", "53375", "50675", ...
## $ Year_ending_Mar_2000  <chr> "48000", "45500", "47000", "54000", "50850", ...
## $ Year_ending_Jun_2000  <chr> "48500", "45000", "48000", "54250", "51350", ...
## $ Year_ending_Sep_2000  <chr> "47500", "45000", "49000", "55500", "52000", ...
## $ Year_ending_Dec_2000  <chr> "47000", "45000", "50000", "55500", "52566", ...
## $ Year_ending_Mar_2001  <chr> "45950", "44500", "50075", "55000", "53995", ...
## $ Year_ending_Jun_2001  <chr> "44950", "44000", "51000", "57500", "54000", ...
## $ Year_ending_Sep_2001  <chr> "47500", "44500", "51997.5", "58995", "55000"...
## $ Year_ending_Dec_2001  <chr> "48500", "45000", "52500", "59950", "56000", ...
## $ Year_ending_Mar_2002  <chr> "48997.5", "46000", "53150", "60000", "56500"...
## $ Year_ending_Jun_2002  <chr> "51950", "48000", "55000", "63000", "57950", ...
## $ Year_ending_Sep_2002  <chr> "53000", "47583", "56000", "65000", "59000", ...
## $ Year_ending_Dec_2002  <chr> "53000", "47000", "59000", "68000", "59500", ...
## $ Year_ending_Mar_2003  <chr> "53500", "45000", "59950", "70000", "59995", ...
## $ Year_ending_Jun_2003  <chr> "54950", "43000", "65000", "75500", "65000", ...
## $ Year_ending_Sep_2003  <chr> "54972.5", "46000", "69950", "80000", "71000"...
## $ Year_ending_Dec_2003  <chr> "56950", "49000", "73000", "85000", "79000", ...
## $ Year_ending_Mar_2004  <chr> "56975", "50500", "76000", "88500", "84000", ...
## $ Year_ending_Jun_2004  <chr> "54250", "60000", "81500", "95000", "90000", ...
## $ Year_ending_Sep_2004  <chr> "55000", "64950", "87000", "100000", "95000",...
## $ Year_ending_Dec_2004  <chr> "52975", "65000", "90000", "106000", "101000"...
## $ Year_ending_Mar_2005  <chr> "54000", "69000", "95000", "110000", "107500"...
## $ Year_ending_Jun_2005  <chr> "57500", "71725", "97500", "113000", "111000"...
## $ Year_ending_Sep_2005  <chr> "60000", "78000", "100000", "115000", "111750...
## $ Year_ending_Dec_2005  <chr> "69000", "85000", "105000", "118000", "110950...
## $ Year_ending_Mar_2006  <chr> "77000", "90176.5", "105000", "119785", "1119...
## $ Year_ending_Jun_2006  <chr> "83000", "96000", "110000", "120000", "112500...
## $ Year_ending_Sep_2006  <chr> "86000", "99000", "115000", "124000", "113995...
## $ Year_ending_Dec_2006  <chr> "90000", "101000", "119500", "125000", "11500...
## $ Year_ending_Mar_2007  <chr> "92500", "103000", "120000", "127972.5", "117...
```

```
## $ Year_ending_Jun_2007  <chr> "97500", "105000", "122000", "130995", "11800...
## $ Year_ending_Sep_2007  <chr> "100000", "108000", "123000", "132995", "1209...
## $ Year_ending_Dec_2007  <chr> "105000", "106000", "123500", "134000", "1215...
## $ Year_ending_Mar_2008  <chr> "109475", "105000", "123000", "134000", "1230...
## $ Year_ending_Jun_2008  <chr> "110000", "105000", "122500", "132000", "1239...
## $ Year_ending_Sep_2008  <chr> "110000", "100000", "120000", "132351", "1229...
## $ Year_ending_Dec_2008  <chr> "110000", "99999", "115000", "129950", "12495...
## $ Year_ending_Mar_2009  <chr> "105000", "100000", "115000", "130000", "1229...
## $ Year_ending_Jun_2009  <chr> "100000", "105000", "115000", "127500", "1187...
## $ Year_ending_Sep_2009  <chr> "100000", "110000", "115000", "125250", "1200...
## $ Year_ending_Dec_2009  <chr> "103500", "111000", "119950", "130000", "1230...
## $ Year_ending_Mar_2010  <chr> "108000", "110000", "120000", "130000", "1249...
## $ Year_ending_Jun_2010  <chr> "110000", "108000", "120000", "130500", "1250...
## $ Year_ending_Sep_2010  <chr> "112500", "108750", "119972.5", "130250", "12...
## $ Year_ending_Dec_2010  <chr> "110500", "105000", "117250", "128000", "1250...
## $ Year_ending_Mar_2011  <chr> "110000", "105000", "116000", "127997.5", "12...
## $ Year_ending_Jun_2011  <chr> "108475", "105000", "112000", "125000", "1249...
## $ Year_ending_Sep_2011  <chr> "105000", "102500", "112000", "125000", "1200...
## $ Year_ending_Dec_2011  <chr> "104000", "102000", "110000", "124000", "1200...
## $ Year_ending_Mar_2012  <chr> "101000", "102000", "111017", "124950", "1200...
## $ Year_ending_Jun_2012  <chr> "100000", "102553", "115000", "125000", "1200...
## $ Year_ending_Sep_2012  <chr> "102500", "105000", "115000", "125000", "1220...
## $ Year_ending_Dec_2012  <chr> "108000", "107000", "115000", "125000", "1190...
## $ Year_ending_Mar_2013  <chr> "110000", "105000", "115000", "125000", "1180...
## $ Year_ending_Jun_2013  <chr> "115000", "105000", "115000", "128245", "1199...
## $ Year_ending_Sep_2013  <chr> "116000", "108000", "116750", "127500", "1199...
## $ Year_ending_Dec_2013  <chr> "115747.5", "110000", "120000", "129995", "12...
## $ Year_ending_Mar_2014  <chr> "114950", "115000", "120000", "130000", "1210...
## $ Year_ending_Jun_2014  <chr> "116000", "118000", "120900", "130000", "1230...
## $ Year_ending_Sep_2014  <chr> "117000", "120000", "122500", "133000", "1249...
## $ Year_ending_Dec_2014  <chr> "120000", "120000", "123000", "134950", "1250...
## $ Year_ending_Mar_2015  <chr> "123000", "123475", "124995", "134500", "1300...
## $ Year_ending_Jun_2015  <chr> "124000", "124000", "125000", "134000", "1300...
## $ Year_ending_Sep_2015  <chr> "124250", "124000", "125000", "135000", "1310...
## $ Year_ending_Dec_2015  <chr> "124000", "129500", "127500", "135000", "1310...
## $ Year_ending_Mar_2016  <chr> "122750", "128000", "127000", "135000", "1320...
## $ Year_ending_Jun_2016  <chr> "120000", "130000", "129995", "135000", "1350...
## $ Year_ending_Sep_2016  <chr> "120000", "131000", "130000", "135000", "1332...
## $ Year_ending_Dec_2016  <chr> "119950", "130000", "130000", "137500", "1325...
## $ Year_ending_Mar_2017  <chr> "119950", "132500", "130000", "139950", "1326...
## $ Year_ending_Jun_2017  <chr> "123250", "129000", "130000", "142000", "1340...
## $ Year_ending_Sep_2017  <chr> "123000", "125000", "130000", "142500", "1379...
## $ Year_ending_Dec_2017  <chr> "124950", "125500", "130000", "144000", "1380...
## $ Year_ending_Mar_2018  <chr> "125000", "125500", "130000", "145000", "1380...
## $ Year_ending_Jun_2018  <chr> "124000", "130000", "130750", "144000", "1399...
## $ Year_ending_Sep_2018  <chr> "124000", "133000", "133750", "146000", "1390...
## $ Year_ending_Dec_2018  <chr> "125000", "133500", "132000", "145000", "1410...
## $ Year_ending_Mar_2019  <chr> "125000", "133500", "131995", "146500", "1429...
## $ Year_ending_Jun_2019  <chr> "126000", "133000", "130000", "147500", "1410...
## $ Year_ending_Sep_2019  <chr> "129950", "134950", "130000", "146500", "1400...
## $ Year_ending_Dec_2019  <chr> "129950", "135000", "131995", "149950", "1424...
## $ Year_ending_Mar_2020  <chr> "128000", "136000", "132950", "150000", "1427...
## $ Year_ending_Jun_2020  <chr> "128000", "134000", "132500", "147500", "1429...
```

```
house_df = data.frame(house)
house_df1 = data.frame(house)
house_columns <- colnames(house_df)
house_columns <- house_columns[5:103]
house_df[,house_columns] = data.frame(apply(house_df[house_columns], 2, as.numeric))
house_df$Region.Country_code <- house_df1$Region.Country_code
house_df$Region.Country_name <- house_df1$Region.Country_name
house_df$Local_authority_code_ <- house_df1$Local_authority_code_
house_df$Local_authority_name <- house_df1$Local_authority_name
```

```
#3.For a given ward in a particular district (e.g., City of Oxford, Cherwell, etc.) find the ave
rage prices of houses in a particular year such as 2018
avg <- house_df %>%
  filter(`Local_authority_name` == "City of London") %>%
  select(`Year_ending_Mar_2018`, `Year_ending_Jun_2018`, `Year_ending_Sep_2018`,`Year_ending_Dec
_2018`) %>%
  collect()
avg_2018 <- (avg$Year_ending_Mar_2018+avg$Year_ending_Jun_2018+avg$Year_ending_Sep_2018+avg$Year
_ending_Dec_2018)/4
avg_2018
```

```
## [1] 886069.5
```

```
#3.For a given ward in a particular district (e.g., City of Oxford, Cherwell, etc.) find the ave
rage prices of houses in a particular year such as 2019
avg <- house_df %>%
  filter(`Local_authority_name` == "City of London") %>%
  select(`Year_ending_Mar_2019`, `Year_ending_Jun_2019`, `Year_ending_Sep_2019`,`Year_ending_Dec
_2019`) %>%
  collect()
avg_2019 <- (avg$Year_ending_Mar_2019+avg$Year_ending_Jun_2019+avg$Year_ending_Sep_2019+avg$Year
_ending_Dec_2019)/4
avg_2019
```

```
## [1] 906851.2
```

```
#4. For a given ward in a particular district find the average increase in prices (in percent) b
etween two years
avg <- house_df %>%
  filter(`Local_authority_name` == "City of London") %>%
  select(`Year_ending_Mar_2017`, `Year_ending_Jun_2017`, `Year_ending_Sep_2017`,`Year_ending_Dec
_2017`) %>%
  collect()
avg_2017 <- (avg$Year_ending_Mar_2017+avg$Year_ending_Jun_2017+avg$Year_ending_Sep_2017+avg$Year
_ending_Dec_2017)/4
#AVerage increse in house price is
avg_2018 - avg_2017
```

```
## [1] 47332
```

```
#5.Considering all districts (in Oxfordshire), find a ward which has the highest house price in
 a particular (quarter of a) year, for example, Mar 2019 or Dec 2019
Max_price <- house_df %>%
  filter(`Region.Country_name` == "Yorkshire and The Humber") %>%
  select(`Region.Country_name`,`Local_authority_name`,`Year_ending_Mar_2019`) %>% collect()
require(data.table) ## 1.9.2
```

```
## Loading required package: data.table
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
group <- as.data.table(Max_price)
group %>% group_by(Region.Country_name) %>% top_n(1, Year_ending_Mar_2019)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 1 x 3
## # Groups:   Region.Country_name [1]
##   Region.Country_name      Local_authority_name Year_ending_Mar_2019
##   <chr>                    <chr>                                <dbl>
## 1 Yorkshire and The Humber Harrogate                           274950
```

```
#.6 Considering all districts (in Oxfordshire), find a ward which has the Lowest house price in
 a particular (quarter of a) year, for example, Mar 2019 or Dec 2019
group %>% group_by(Region.Country_name) %>% top_n(-1, Year_ending_Mar_2019)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 1 x 3
## # Groups:   Region.Country_name [1]
##   Region.Country_name      Local_authority_name        Year_ending_Mar_2019
##   <chr>                    <chr>                                      <dbl>
## 1 Yorkshire and The Humber Kingston upon Hull, City of               115000
```

```
broadband_df <- data.frame(broadband)
glimpse(broadband_df)
```

```
## Observations: 9,131
## Variables: 16
## $ Ward_Code                                          <chr> "E05000026", "E...
## $ Ward_Name                                          <chr> "Abbey", "Alibo...
## $ Local_Authority                                    <chr> "Barking and Da...
## $ Constituencies                                     <chr> "Barking", "Bar...
## $ Region._Country                                    <chr> "London", "Lond...
## $ Average_Download_Speed_Mbps                        <dbl> 42.1, 57.1, 49....
## $ SFBB_or_UFBB_Availability_..30_Mbps.               <dbl> 0.940, 0.988, 0...
## $ Fibre_to_the_property_availability                 <dbl> 0.133, 0.000, 0...
## $ Connections_Unable_to_receive_10_Mbps              <dbl> 0.000, 0.000, 0...
## $ Connections_Receiving_Under_2_Mbps                 <dbl> 0.004, 0.004, 0...
## $ Connections_Receiving_Under_10_Mbps                <dbl> 0.063, 0.066, 0...
## $ Connections_Receiving_Superfast_Speeds_....30_Mbps. <dbl> 0.412, 0.546, 0...
## $ Connections_Receiving_Over_300_Mbps                <dbl> 0.001, 0.002, 0...
## $ Average_speed_for_basic_broadband_lines_           <dbl> 17.1, 15.9, 8.1...
## $ Average_speed_for_superfast_lines                  <dbl> 77.5, 91.0, 84....
## $ Estimated_data_completeness                        <dbl> 0.995, 0.999, 0...
```

```
#7.Based on the data source (you use), find a broadband speed (average downland), or (superfast) broadband availability (%), in a particular ward (or a postcode) of a district
Average_Download_Speed_Alibon <- broadband_df %>%
  filter(`Ward_Name` == "Alibon") %>%
  select(`Average_Download_Speed_Mbps`) %>% collect()
Average_Download_Speed_Alibon
```

```
##   Average_Download_Speed_Mbps
## 1                        57.1
```

```
#7.Based on the data source (you use), find a broadband speed (average downland), or (superfast) broadband availability (%), in a particular ward (or a postcode) of a district
UFBB_Availability_Alibon <- broadband_df %>%
  filter(`Ward_Name` == "Alibon") %>%
  select(`SFBB_or_UFBB_Availability_..30_Mbps.`) %>% collect()
UFBB_Availability_Alibon
```

```
##   SFBB_or_UFBB_Availability_..30_Mbps.
## 1                                0.988
```

```
#8.Design a query of your choice using the broadband data. The query should retrieve certain dat
a from the database, which you think, can be useful to a user.
#Which ward is having highest average broadband speed
Average_Download_Speed <- broadband_df %>%
  select(`Ward_Name`,`Constituencies`,`Average_Download_Speed_Mbps`) %>% collect()
require(data.table) ## 1.9.2
group <- as.data.table(Average_Download_Speed)
group %>% top_n(1, Average_Download_Speed_Mbps)
```

```
##           Ward_Name          Constituencies Average_Download_Speed_Mbps
## 1 Upper Lune Valley Morecambe and Lunesdale                       472.8
```

```
#8.Highest speed Constituency
Average_Download_Speed <- broadband_df %>%
  select(`Constituencies`,`Average_Download_Speed_Mbps`) %>% collect()
group <- as.data.table(Average_Download_Speed)
group %>% top_n(1, Average_Download_Speed_Mbps)
```

```
##            Constituencies Average_Download_Speed_Mbps
## 1 Morecambe and Lunesdale                       472.8
```

```
#8.Highest speed Region
Average_Download_Speed <- broadband_df %>%
  select(`Region._Country`,`Average_Download_Speed_Mbps`) %>% collect()
group <- as.data.table(Average_Download_Speed)
group %>% top_n(1, Average_Download_Speed_Mbps)
```

```
##   Region._Country Average_Download_Speed_Mbps
## 1      North West                      472.8
```