# Class Notes 10

Tobias Boggess

4/13/2022

## Section 15.2 Exercises

**Exercise 1: Now open a connection to the airlines database using the command involving dbConnect() (from "DBI") given above. Save the connection as, say, db_con**

Setup:

```
db_con <- dbConnect(drv = MySQL(),
                    dbname = "airlines",
                    host = "mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com",
                    user = "mdsr_public",
                    password = "ImhsmflMDSwR")
```

**a) What is the class of the object db_con? Use class().**
Code:

```
class(db_con)
```

```
## [1] "MySQLConnection"
## attr(,"package")
## [1] "RMySQL"
```

The class is labelled above.

**b) How many tables are in the airlines database and what are their name? Use dbListTables().**
Code:

```
dbListTables(db_con)
```

```
## [1] "airports" "carriers" "flights"  "planes"
```

There are four tables in the airlines database and the names are reported above.

**c) Recall that DESCRIBE is used to get a description of the contents of a table. Using dbGetQuery(), with statement = "DESCRIBE airports;", how many variables ("fields") are in the airports table?**
Code:

```
dbGetQuery(conn = db_con,
           statement = "DESCRIBE airports")
```

```
##     Field          Type Null Key Default Extra
## 1     faa    varchar(3)   NO PRI
## 2    name  varchar(255)  YES        <NA>
## 3     lat decimal(10,7)  YES        <NA>
## 4     lon decimal(10,7)  YES        <NA>
## 5     alt       int(11)  YES        <NA>
## 6      tz   smallint(4)  YES        <NA>
## 7     dst       char(1)  YES        <NA>
## 8    city  varchar(255)  YES        <NA>
## 9 country  varchar(255)  YES        <NA>
```

There are 8/9 variables in the table given the city and country is split up when the above command is run.

**d) Using dbGetQuery(), with statement = "DESCRIBE flights;", how many variables ("fields") are in the flights table?**
Code:

```
dbGetQuery(db_con, statement = "DESCRIBE flights")
```

```
##             Field        Type Null Key Default Extra
## 1            year smallint(4)  YES MUL    <NA>
## 2           month smallint(2)  YES        <NA>
## 3             day smallint(2)  YES        <NA>
## 4        dep_time smallint(4)  YES        <NA>
## 5  sched_dep_time smallint(4)  YES        <NA>
## 6       dep_delay smallint(4)  YES        <NA>
## 7        arr_time smallint(4)  YES        <NA>
## 8  sched_arr_time smallint(4)  YES        <NA>
## 9       arr_delay smallint(4)  YES        <NA>
## 10        carrier  varchar(2)   NO MUL
## 11        tailnum  varchar(6)  YES MUL    <NA>
## 12         flight smallint(4)  YES        <NA>
## 13         origin  varchar(3)   NO MUL
## 14           dest  varchar(3)   NO MUL
## 15       air_time smallint(4)  YES        <NA>
## 16       distance smallint(4)  YES        <NA>
## 17      cancelled  tinyint(1)  YES        <NA>
## 18       diverted  tinyint(1)  YES        <NA>
## 19           hour smallint(2)  YES        <NA>
## 20         minute smallint(2)  YES        <NA>
## 21      time_hour    datetime  YES        <NA>
```

According to the above command there are 10 variables.

## Section 15.3 Exercises

**Exercise 2: Use dbGetQuery() with SELECT and FROM to query the flights table to retrieve just the carrier and tailnum variables (columns). Report your R command(s) (or just your SQL statement).**

Code:

```
dbGetQuery(conn = db_con,
           statement = "SELECT carrier, tailnum FROM flights
           LIMIT 0, 5;")
```

```
##   carrier tailnum
## 1      XE  N11137
## 2      B6  N659JB
## 3      B6  N563JB
## 4      XE  N16559
## 5      OO  N908SW
```

**Exercise 3: Now use dbGetQuery() with SELECT and FROM to query the carriers table to retrieve all the variables (columns). Do the following.**

Setup:

```
my.carriers <- dbGetQuery(db_con,
           statement = "SELECT * FROM carriers
           LIMIT 0, 5;")
my.carriers
```

```
##   carrier                          name
## 1     02Q                 Titan Airways
## 2     04Q            Tradewind Aviation
## 3     05Q           Comlux Aviation, AG
## 4     06Q Master Top Linhas Aereas Ltd.
## 5     07Q           Flair Airlines Ltd.
```

**a) Confirm that the data set returned by dbGetQuery() is a data frame by typing:**
Code:

```
is.data.frame(my.carriers)
```

```
## [1] TRUE
```

**b) You can find out how much memory an object occupies in R using the object.size() function and its print() method.How much memory does my.carriers occupy? Find out by typing:**
Code:

```
print(object.size(my.carriers), units = "Kb")
```

```
## 1.6 Kb
```

```
print(object.size(my.carriers), units = "Mb")
```

```
## 0 Mb
```

**Exercise 4: Now use dbGetQuery() with SELECT and FROM to form a query to retrieve all the variables (columns) from the airports table. Save the data returned as, say, my.airports in R.**

Data frame:

```
my.airports <- dbGetQuery(conn = db_con,
           statement = "SELECT * FROM airports")
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 3 imported as
## numeric
```

**a) Each observation (row) in the my.airports data frame is an airport. How many rows does the data frame have? Use str() or nrow() or dim().**
Code:

```
nrow(my.airports)
```

```
## [1] 1458
```

There are 1,458 rows in the airport data frame.

**b) How many columns (variables) does the data frame have?**
Code:

```
ncol(my.airports)
```

```
## [1] 9
```

There are 9 variables in the data frame.

**Exercise 5: Recall that we can use dbGetQuery() with SELECT and FROM to form and select new columns from existing ones in a table, similar to using mutate() in "dplyr". Form a new column containing the travel speeds (in mph) of the flights (distance divided by air_time, then multiplied by 60), using AS to give it the name trvl_speed.**

Code:

```
dbGetQuery(db_con,
           statement = "SELECT *,
           (distance / air_time) * 60 AS trvl_speed
           FROM flights
           LIMIT 0,10;")
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 21 imported as
## numeric
```

```
##    year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1  2010    10   1        1           2100       181      159           2320
## 2  2010    10   1        1           1920       281      230           2214
## 3  2010    10   1        3           2355         8      339            334
## 4  2010    10   1        5           2200       125       41           2249
## 5  2010    10   1        7           2245        82      104           2347
## 6  2010    10   1        7             10        -3      451            500
## 7  2010    10   1        7           2150       137      139           2337
## 8  2010    10   1        8             15        -7      538            537
## 9  2010    10   1        8             10        -2      643            645
## 10 2010    10   1       10           2225       105      831            642
##    arr_delay carrier tailnum flight origin dest air_time distance cancelled
## 1        159      XE  N11137   2558    EWR  OMA      162     1133         0
## 2        256      B6  N659JB    562    FLL  SWF      131     1119         0
## 3          5      B6  N563JB    701    JFK  SJU      196     1597         0
## 4        112      XE  N16559   5982    IAD  BNA       82      542         0
## 5         77      OO  N908SW   6433    LAX  FAT       37      209         0
## 6         -9      AA  N3FRAA    700    LAX  DFW      150     1235         0
## 7        122      DL  N347NW   1752    ATL  IAD       70      533         0
## 8          1      CO  N73283   1740    SMF  IAH      193     1609         0
## 9         -2      DL  N333NW   2344    LAS  CVG      196     1678         0
## 10       109      B6  N585JB    174    SJC  JFK      293     2570         0
##    diverted hour minute           time_hour trvl_speed
## 1         0   21      0 2010-10-01 21:00:00   419.6296
## 2         0   19     20 2010-10-01 19:20:00   512.5191
## 3         0   23     55 2010-10-01 23:55:00   488.8776
## 4         0   22      0 2010-10-01 22:00:00   396.5854
## 5         0   22     45 2010-10-01 22:45:00   338.9189
## 6         0    0     10 2010-10-01 00:10:00   494.0000
## 7         0   21     50 2010-10-01 21:50:00   456.8571
## 8         0    0     15 2010-10-01 00:15:00   500.2073
## 9         0    0     10 2010-10-01 00:10:00   513.6735
## 10        0   22     25 2010-10-01 22:25:00   526.2799
```

## Section 15.4 Exercises

**Exercise 6: Report R command(s) (or just the SQL statements) involving dbGetQuery() and the logical operators (AND, OR, and NOT) with the flights table to retrieve flights meeting the following conditions.**

**a) Had an arrival delay of more than hours (arr_delay more than 120 minutes).**
Code:

```
dbGetQuery(db_con,
           statement = "SELECT *
                        FROM flights
                        WHERE arr_time > 120
                        LIMIT 0,10;")
```

```
##    year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1  2010    10   1        1           2100       181      159           2320
## 2  2010    10   1        1           1920       281      230           2214
## 3  2010    10   1        3           2355         8      339            334
## 4  2010    10   1        7             10        -3      451            500
## 5  2010    10   1        7           2150       137      139           2337
## 6  2010    10   1        8             15        -7      538            537
## 7  2010    10   1        8             10        -2      643            645
## 8  2010    10   1       10           2225       105      831            642
## 9  2010    10   1       12           2045       207      136           2209
## 10 2010    10   1       17           2319        58      758            732
##    arr_delay carrier tailnum flight origin dest air_time distance cancelled
## 1        159      XE  N11137   2558    EWR  OMA      162     1133         0
## 2        256      B6  N659JB    562    FLL  SWF      131     1119         0
## 3          5      B6  N563JB    701    JFK  SJU      196     1597         0
## 4         -9      AA  N3FRAA    700    LAX  DFW      150     1235         0
## 5        122      DL  N347NW   1752    ATL  IAD       70      533         0
## 6          1      CO  N73283   1740    SMF  IAH      193     1609         0
## 7         -2      DL  N333NW   2344    LAS  CVG      196     1678         0
## 8        109      B6  N585JB    174    SJC  JFK      293     2570         0
## 9        207      B6  N267JB   1329    BOS  BWI       61      370         0
## 10        26      UA  N556UA    792    SFO  IAD      259     2419         0
##    diverted hour minute           time_hour
## 1         0   21      0 2010-10-01 21:00:00
## 2         0   19     20 2010-10-01 19:20:00
## 3         0   23     55 2010-10-01 23:55:00
## 4         0    0     10 2010-10-01 00:10:00
## 5         0   21     50 2010-10-01 21:50:00
## 6         0    0     15 2010-10-01 00:15:00
## 7         0    0     10 2010-10-01 00:10:00
## 8         0   22     25 2010-10-01 22:25:00
## 9         0   20     45 2010-10-01 20:45:00
## 10        0   23     19 2010-10-01 23:19:00
```

**b) Flew to Houston (i.e. had a dest of IAH or HOU).**
Code:

```
dbGetQuery(db_con,
           statement = "SELECT *
                        FROM flights
                        WHERE dest = 'IAH' OR dest = 'HOU'
                        LIMIT 0,10;")
```

```
##    year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1  2010    10   1      559            605        -6      700            715
## 2  2010    10   1      602            605        -3      746            810
## 3  2010    10   1      615            615         0      710            715
## 4  2010    10   1      616            620        -4      702            710
## 5  2010    10   1      619            620        -1      836            905
## 6  2010    10   1      625            625         0      932            950
## 7  2010    10   1      629            630        -1      724            730
## 8  2010    10   1      630            635        -5      727            735
## 9  2010    10   1      630            630         0      756            800
## 10 2010    10   1      639            640        -1      729            735
##    arr_delay carrier tailnum flight origin dest air_time distance cancelled
## 1        -15      WN  N527SW    613    MSY  HOU       50      303         0
## 2        -24      WN  N347SW     10    BNA  HOU       92      670         0
## 3         -5      WN  N262WN   1628    DAL  HOU       42      239         0
## 4         -8      WN  N660SW      6    CRP  HOU       33      187         0
## 5        -29      WN  N741SA    368    MDW  HOU      125      937         0
## 6        -18      WN  N791SW    400    DEN  HOU      113      883         0
## 7         -6      WN  N350SW    201    DAL  HOU       41      239         0
## 8         -8      WN  N612SW      8    HRL  HOU       46      276         0
## 9         -4      WN  N303SW   1091    TUL  HOU       72      453         0
## 10        -6      WN  N388SW   1102    SAT  HOU       37      192         0
##    diverted hour minute           time_hour
## 1         0    6      5 2010-10-01 06:05:00
## 2         0    6      5 2010-10-01 06:05:00
## 3         0    6     15 2010-10-01 06:15:00
## 4         0    6     20 2010-10-01 06:20:00
## 5         0    6     20 2010-10-01 06:20:00
## 6         0    6     25 2010-10-01 06:25:00
## 7         0    6     30 2010-10-01 06:30:00
## 8         0    6     35 2010-10-01 06:35:00
## 9         0    6     30 2010-10-01 06:30:00
## 10        0    6     40 2010-10-01 06:40:00
```

**c) Were operated by United, American, or Delta (i.e. had a carrier of UA, AA, or DL).**
Code:

```
dbGetQuery(db_con,
           statement = "SELECT *
                        FROM flights
                        WHERE carrier = 'UA' OR
                        carrier = 'AA' OR
                        carrier = 'DL'
                        LIMIT 0,10;")
```

```
##    year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1  2010    10   1        7             10        -3      451            500
## 2  2010    10   1       21             25        -4      537            535
## 3  2010    10   1       43             45        -2      528            535
## 4  2010    10   1      119             35        44      545            500
## 5  2010    10   1      538            535         3      900            845
## 6  2010    10   1      541            540         1      924            940
## 7  2010    10   1      542            545        -3      920            925
## 8  2010    10   1      543            545        -2      903            900
## 9  2010    10   1      550            545         5      809            755
## 10 2010    10   1      553            600        -7      642            700
##    arr_delay carrier tailnum flight origin dest air_time distance cancelled
## 1         -9      AA  N3FRAA    700    LAX  DFW      150     1235         0
## 2          2      AA  N5EGAA   2281    SFO  DFW      174     1464         0
## 3         -7      AA  N3FSAA   2408    LAX  DFW      149     1235         0
## 4         45      AA  N579AA   2385    LAS  DFW      129     1055         0
## 5         15      AA  N611AA   1860    JFK  MIA      167     1090         0
## 6        -16      AA  N5DPAA    462    ORD  MIA      149     1197         0
## 7         -5      AA  N619AA   1711    DFW  MIA      139     1121         0
## 8          3      AA  N5ELAA   1037    BOS  MIA      181     1258         0
## 9         14      AA  N4XAAA   2320    DFW  ORD      111      802         0
## 10       -18      AA  N481AA   1492    AUS  DFW       35      190         0
##    diverted hour minute           time_hour
## 1         0    0     10 2010-10-01 00:10:00
## 2         0    0     25 2010-10-01 00:25:00
## 3         0    0     45 2010-10-01 00:45:00
## 4         0    0     35 2010-10-01 00:35:00
## 5         0    5     35 2010-10-01 05:35:00
## 6         0    5     40 2010-10-01 05:40:00
## 7         0    5     45 2010-10-01 05:45:00
## 8         0    5     45 2010-10-01 05:45:00
## 9         0    5     45 2010-10-01 05:45:00
## 10        0    6      0 2010-10-01 06:00:00
```

**d) Departed in summer (July, August, or September, i.e. month 7, 8, or 9).**
Code:

```
dbGetQuery(db_con,
           statement = "SELECT *
                        FROM flights
                        WHERE month IN (7, 8, 9)
                        LIMIT 0,10;")
```

```
##    year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1  2010     7   1        1           2359         2      506            517
## 2  2010     7   1        2             10        -8      447            500
## 3  2010     7   1        3              3         0      638            645
## 4  2010     7   1        4             10        -6      715            717
## 5  2010     7   1        5           2359         6      449            459
## 6  2010     7   1        6           2330        36      651            626
## 7  2010     7   1        6           2201       125       13           2230
## 8  2010     7   1        8           2300        68       55           2359
## 9  2010     7   1        9           2345        24      550            545
```

```
## 10 2010     7   1        9           2200       129      113           2314
##    arr_delay carrier tailnum flight origin dest air_time distance cancelled
## 1        -11      US  N122US   1110    PHX  ORD      169     1440         0
## 2        -13      AA  N3DPAA   2400    LAX  DFW      144     1235         0
## 3         -7      DL  N3763D   2344    LAS  CVG      198     1678         0
## 4         -2      AS  N596AS    148    ANC  DEN      294     2405         0
## 5        -10      US  N826AW    358    PHX  MSP      148     1276         0
## 6         25      DL  N361NW   2216    LAS  DTW      207     1750         0
## 7        103      EV  N975EV   5110    ATL  EVV       52      350         0
## 8         56      AA  N3AHAA    343    MIA  MSY       90      674         0
## 9          5      AA  N3ELAA   1522    SFO  ORD      204     1846         0
## 10       119      US  N820AW    640    ONT  PHX       44      325         0
##    diverted hour minute           time_hour
## 1         0   23     59 2010-07-01 23:59:00
## 2         0    0     10 2010-07-01 00:10:00
## 3         0    0      3 2010-07-01 00:03:00
## 4         0    0     10 2010-07-01 00:10:00
## 5         0   23     59 2010-07-01 23:59:00
## 6         0   23     30 2010-07-01 23:30:00
## 7         0   22      1 2010-07-01 22:01:00
## 8         0   23      0 2010-07-01 23:00:00
## 9         0   23     45 2010-07-01 23:45:00
## 10        0   22      0 2010-07-01 22:00:00
```

**e) Departed between midnight and 6:00 AM, inclusive (dep_time at 2400 or between 0 and 600, inclusive).**
Code:

```
dbGetQuery(db_con,
           statement = "SELECT * FROM flights
           WHERE dep_time = 2400 OR
           dep_time BETWEEN 0 AND 600
           LIMIT 0,10;")
```

```
##    year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1  2010    10   1        1           2100       181      159           2320
## 2  2010    10   1        1           1920       281      230           2214
## 3  2010    10   1        3           2355         8      339            334
## 4  2010    10   1        5           2200       125       41           2249
## 5  2010    10   1        7           2245        82      104           2347
## 6  2010    10   1        7             10        -3      451            500
## 7  2010    10   1        7           2150       137      139           2337
## 8  2010    10   1        8             15        -7      538            537
## 9  2010    10   1        8             10        -2      643            645
## 10 2010    10   1       10           2225       105      831            642
##    arr_delay carrier tailnum flight origin dest air_time distance cancelled
## 1        159      XE  N11137   2558    EWR  OMA      162     1133         0
## 2        256      B6  N659JB    562    FLL  SWF      131     1119         0
## 3          5      B6  N563JB    701    JFK  SJU      196     1597         0
## 4        112      XE  N16559   5982    IAD  BNA       82      542         0
## 5         77      OO  N908SW   6433    LAX  FAT       37      209         0
## 6         -9      AA  N3FRAA    700    LAX  DFW      150     1235         0
## 7        122      DL  N347NW   1752    ATL  IAD       70      533         0
```

```
## 8          1      CO  N73283   1740    SMF  IAH      193     1609         0
## 9         -2      DL  N333NW   2344    LAS  CVG      196     1678         0
## 10       109      B6  N585JB    174    SJC  JFK      293     2570         0
##    diverted hour minute           time_hour
## 1         0   21      0 2010-10-01 21:00:00
## 2         0   19     20 2010-10-01 19:20:00
## 3         0   23     55 2010-10-01 23:55:00
## 4         0   22      0 2010-10-01 22:00:00
## 5         0   22     45 2010-10-01 22:45:00
## 6         0    0     10 2010-10-01 00:10:00
## 7         0   21     50 2010-10-01 21:50:00
## 8         0    0     15 2010-10-01 00:15:00
## 9         0    0     10 2010-10-01 00:10:00
## 10        0   22     25 2010-10-01 22:25:00
```

**f) Were operated by United (carrier UA), departed in July (month 7), and had an arrival delay of more than two hours (arr_delay more than 120 minutes).**
Code:

```
dbGetQuery(db_con,
           statement = "SELECT * FROM flights
           WHERE carrier = 'UA' AND
           month = 7 AND
           arr_delay > 120
           LIMIT 0,10;")
```

```
##    year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1  2010     7   1       23           2130       173      148           2305
## 2  2010     7   1     1600           1216       224     1843           1502
## 3  2010     7   1     2049           1631       258     2210           1756
## 4  2010     7   1     2107           1415       412       18           1744
## 5  2010     7   1     2148           1735       253     2320           1936
## 6  2010     7   1     2205           1805       240       15           2024
## 7  2010     7   2       45           2154       171      249           2359
## 8  2010     7   2      943            600       223     1113            731
## 9  2010     7   2     1205            834       211     2039           1720
## 10 2010     7   2     1341           1109       152     1848           1646
##    arr_delay carrier tailnum flight origin dest air_time distance cancelled
## 1        163      UA  N581UA    467    DEN  SJC      132      949         0
## 2        221      UA  N427UA    217    IAD  SFO      301     2419         0
## 3        254      UA  N453UA    930    SAN  SFO       67      447         0
## 4        394      UA  N572UA    168    SJC  DEN      114      949         0
## 5        224      UA  N582UA    951    IAD  LAS      252     2066         0
## 6        231      UA  N507UA    240    IAD  SAN      293     2253         0
## 7        170      UA  N470UA    332    ORD  CMH       46      296         0
## 8        222      UA  N568UA    510    LAS  SFO       68      414         0
## 9        199      UA  N556UA    662    SFO  BOS      299     2704         0
## 10       122      UA  N824UA    508    SFO  MSP      170     1589         0
##    diverted hour minute           time_hour
## 1         0   21     30 2010-07-01 21:30:00
## 2         0   12     16 2010-07-01 12:16:00
## 3         0   16     31 2010-07-01 16:31:00
## 4         0   14     15 2010-07-01 14:15:00
```

```
## 5         0   17     35 2010-07-01 17:35:00
## 6         0   18      5 2010-07-01 18:05:00
## 7         0   21     54 2010-07-02 21:54:00
## 8         0    6      0 2010-07-02 06:00:00
## 9         0    8     34 2010-07-02 08:34:00
## 10        0   11      9 2010-07-02 11:09:00
```

## Section 15.5 Exercises

**Exercise 7: We know there were 65 flights that left Bradley Airport on June 26th, 2013, but what was the shortest departure delay for each airline carrier? What was the longest?**

**a) Use dbGetQuery() with GROUP BY and MIN() to retrieve the shortest departure delay for each carrier. Report your R command(s) (or just your SQL statement).**
Code:

```
dbGetQuery(db_con,
           statement = "SELECT carrier, MIN(dep_delay)
           AS min_dep_delay
           FROM flights
           WHERE origin = 'BDL' AND month = 6 AND day = 26 AND year = 2013
           GROUP BY carrier
           LIMIT 0,10;")
```

```
##    carrier min_dep_delay
## 1       9E            -7
## 2       AA            -1
## 3       B6            -3
## 4       DL            -5
## 5       EV            -9
## 6       MQ             0
## 7       UA             0
## 8       US            -7
## 9       WN             0
## 10      YV             0
```

**b) GROUP BY can summarize more than one variable at a time. Modify your command(s) from part a to use MIN() and MAX() to retrieve the shortest and longest departure delays for each carrier. Report your R command(s) (or just your SQL statement).**
Code:

```
dbGetQuery(db_con,
           statement =
           "SELECT carrier, MIN(dep_delay) AS min_dep_delay,
           MAX(dep_delay) AS max_dep_delay
           FROM flights
           WHERE origin = 'BDL'AND year = 2013 AND month = 6 AND day = 26
           GROUP BY carrier
           LIMIT 0,6;")
```

```
##   carrier min_dep_delay max_dep_delay
```

```
## 1      9E           -7            0
## 2      AA           -1           94
## 3      B6           -3           77
## 4      DL           -5           20
## 5      EV           -9          211
## 6      MQ            0          174
```

**Exercise 8: This problem concerns the GROUP BY and AVG() functions. It uses the flights table.**

**a) Explain in words what the following command does (recall that dest is the destination of the flight):**
Code:

```
dbGetQuery(
  conn = db_con,
  statement = "SELECT carrier, dest, AVG(arr_delay) AS meanArrDelay
FROM flights
WHERE year = 2013 AND month = 6 AND day = 26 AND origin = 'BDL'
GROUP BY dest;"
)
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
##    carrier dest meanArrDelay
## 1       DL  ATL       5.5000
## 2       WN  BWI      38.6250
## 3       US  CLT      17.2000
## 4       9E  CVG      -2.6667
## 5       US  DCA      -9.0000
## 6       WN  DEN      74.0000
## 7       AA  DFW      23.6667
## 8       DL  DTW     -16.2000
## 9       EV  EWR     -13.5000
## 10      B6  FLL       2.0000
## 11      YV  IAD       0.0000
## 12      WN  LAS     -11.0000
## 13      WN  MCO      43.4000
## 14      WN  MDW     197.3333
## 15      AA  MIA      -6.0000
## 16      DL  MSP       0.3333
## 17      YV  ORD     102.4444
## 18      B6  PBI      61.0000
## 19      US  PHL      14.0000
## 20      9E  RDU      -4.0000
## 21      B6  SJU      22.0000
## 22      WN  TPA      -5.5000
```

This command will show the carriers, destinations, and average arrival delay from the flights data set that happened on June 26 of 2013 from the BDL airport and group them based on the destination airport.

**b) Recall that GROUP BY can summarize more than one variable at a time. Explain in words what the following command does:**
Code:

```
dbGetQuery(
  conn = db_con,
  statement = "SELECT carrier, dest, AVG(arr_delay) AS meanArrDelay,
AVG(distance) AS meanDist
FROM flights
WHERE year = 2013 AND month = 6 AND day = 26 AND origin = 'BDL'
GROUP BY dest;"
)
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric

## Warning in .local(conn, statement, ...): Decimal MySQL column 3 imported as
## numeric

##    carrier dest meanArrDelay meanDist
## 1       DL  ATL       5.5000      859
## 2       WN  BWI      38.6250      283
## 3       US  CLT      17.2000      644
## 4       9E  CVG      -2.6667      661
## 5       US  DCA      -9.0000      313
## 6       WN  DEN      74.0000     1671
## 7       AA  DFW      23.6667     1471
## 8       DL  DTW     -16.2000      549
## 9       EV  EWR     -13.5000      116
## 10      B6  FLL       2.0000     1173
## 11      YV  IAD       0.0000      326
## 12      WN  LAS     -11.0000     2297
## 13      WN  MCO      43.4000     1050
## 14      WN  MDW     197.3333      777
## 15      AA  MIA      -6.0000     1194
## 16      DL  MSP       0.3333     1050
## 17      YV  ORD     102.4444      783
## 18      B6  PBI      61.0000     1133
## 19      US  PHL      14.0000      196
## 20      9E  RDU      -4.0000      532
## 21      B6  SJU      22.0000     1666
## 22      WN  TPA      -5.5000     1111
```

This command will show the carrier, destination, average arrival delay, and average distance from the flights dataset on June 26, 2013 at the BDL airport and group them by the destination.

## Section 15.6 Exercises

**Exercise 9: This problem concerns ORDER BY combined with GROUP BY and AVG(). It uses the flights table. There were 22,258 flights that left Bradley Airport in the year 2013.**

**a) Use GROUP BY with AVG() and ORDER BY to determine the destination dest for which the average travel time (air_time) from Bradley Airport was shortest in 2013. Report the**

**(abbreviated) destination (dest) name.**
Code:

```
dbGetQuery(con = db_con,
           statement = "SELECT dest, AVG(air_time) AS avg_air_time
           FROM flights
           WHERE origin = 'BDL' AND year = 2013
           GROUP BY dest
           ORDER BY avg_air_time ASC
           LIMIT 0,6;")
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 1 imported as
## numeric
```

```
##   dest avg_air_time
## 1  EWR      32.8673
## 2  BWI      52.6590
## 3  PHL      53.3492
## 4  IAD      57.5096
## 5  DCA      65.5490
## 6  CLE      82.2982
```

**b) Now use GROUP BY with AVG() and ORDER BY to determine the destination dest for which the average travel time (air_time) from Bradley Airport was longest in 2013. Report the (abbreviated) destination (dest) name.**
Code:

```
dbGetQuery(con = db_con,
           statement = "SELECT dest, AVG(air_time) AS avg_air_time
           FROM flights
           WHERE origin = 'BDL' AND year = 2013
           GROUP BY dest
           ORDER BY avg_air_time DESC
           LIMIT 0,6;")
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 1 imported as
## numeric
```

```
##   dest avg_air_time
## 1  LAX     341.7165
## 2  LAS     310.8626
## 3  DEN     236.4110
## 4  DFW     205.8531
## 5  SJU     203.8560
## 6  RSW     176.7582
```

**Exercise 10: This problem concerns ORDER BY combined with GROUP BY and COUNT(). It uses the flights table. There were 22,258 flights that left Bradley Airport in the year 2013.**

**a) Use GROUP BY with COUNT() and ORDER BY to determine to determine which dest (i.e. which destination) was flown to the most times out of Bradley Airport in 2013. Report the (abbreviated) destination (dest) name.**
Code:

```
dbGetQuery(con = db_con,
           statement = "SELECT dest, COUNT(*) AS numFlights
           FROM flights
           WHERE year = 2013 AND origin = 'BDL'
           GROUP BY dest
           ORDER BY numFlights DESC
           LIMIT 0,6;")
```

```
##   dest numFlights
## 1  ORD       2657
## 2  BWI       2613
## 3  ATL       2277
## 4  CLT       1842
## 5  MCO       1789
## 6  DTW       1523
```

**b) Use GROUP BY with COUNT() and ORDER BY to determine which tailnum (i.e. which individual airplane) flew the most times out of Bradley Airport in 2013. Report the tailnum value of the airplane.**
Code:

```
dbGetQuery(con = db_con,
           statement = "SELECT tailnum, COUNT(*) AS numFlights
           FROM flights
           WHERE year = 2013 AND origin = 'BDL'
           GROUP BY tailnum
           ORDER BY numFlights DESC
           LIMIT 0,6;")
```

```
##   tailnum numFlights
## 1      NA         97
## 2  N128UW         36
## 3  N505MJ         35
## 4  N504MJ         35
## 5  N522LR         34
## 6  N503MJ         34
```

## Sectiion 15.7 Exercises

**Exercise 11: There were 22,258 flights that left Bradley Airport in the year 2013. There were 25 destinations of those flights from Bradley:**

Setup:

```
dbGetQuery(
  conn = db_con,
  statement = "SELECT dest, COUNT(*) AS numFlights,
AVG(arr_delay) AS avg_arr_delay
FROM flights
WHERE year = 2013 AND origin = 'BDL'
```

```
GROUP BY dest;"
)
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
##    dest numFlights avg_arr_delay
## 1   ATL       2277        4.4704
## 2   BWI       2613        5.0325
## 3   CLE         57      -13.0702
## 4   CLT       1842       -0.1205
## 5   CVG        708       -7.3701
## 6   DCA        204       -2.8971
## 7   DEN        365       -1.2767
## 8   DFW       1062        0.7495
## 9   DTW       1523       -2.1477
## 10  EWR        437       13.1968
## 11  FLL       1011        0.2770
## 12  IAD        622        4.9084
## 13  LAS        262       -0.3092
## 14  LAX        127      -10.3071
## 15  MCO       1789        8.3784
## 16  MDW        974        8.5144
## 17  MIA        404       -3.2723
## 18  MSP        981       -3.6636
## 19  ORD       2657        7.3643
## 20  PBI        365        8.4466
## 21  PHL        358        4.3352
## 22  RDU        256        0.9570
## 23  RSW        273        6.0879
## 24  SJU        375        7.0427
## 25  TPA        716        6.0740
```

**a) Alter the command above using HAVING so that it only returns destinations for which the average arrival delay is positive (avg_arr_delay > 0).Report your R command(s) (or just your SQL statement).**
Code:

```
dbGetQuery(
  conn = db_con,
  statement = "SELECT dest, COUNT(*) AS numFlights,
AVG(arr_delay) AS avg_arr_delay
FROM flights
WHERE year = 2013 AND origin = 'BDL'
GROUP BY dest
HAVING avg_arr_delay > 0;"
)
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
##    dest numFlights avg_arr_delay
```

```
## 1   ATL       2277        4.4704
## 2   BWI       2613        5.0325
## 3   DFW       1062        0.7495
## 4   EWR        437       13.1968
## 5   FLL       1011        0.2770
## 6   IAD        622        4.9084
## 7   MCO       1789        8.3784
## 8   MDW        974        8.5144
## 9   ORD       2657        7.3643
## 10  PBI        365        8.4466
## 11  PHL        358        4.3352
## 12  RDU        256        0.9570
## 13  RSW        273        6.0879
## 14  SJU        375        7.0427
## 15  TPA        716        6.0740
```

**b) Now alter your command from part a so that it only returns destinations for which the average arrival delay is positive (avg_arr_delay > 0) and the total number of flights was more than 1,000 (numFlights > 1000). Report your R command(s) (or just your SQL statement).**
Code:

```
dbGetQuery(
  conn = db_con,
  statement = "SELECT dest, COUNT(*) AS numFlights,
AVG(arr_delay) AS avg_arr_delay
FROM flights
WHERE year = 2013 AND origin = 'BDL'
GROUP BY dest
HAVING avg_arr_delay > 0 AND numFlights > 1000;"
)
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
##   dest numFlights avg_arr_delay
## 1  ATL       2277        4.4704
## 2  BWI       2613        5.0325
## 3  DFW       1062        0.7495
## 4  FLL       1011        0.2770
## 5  MCO       1789        8.3784
## 6  ORD       2657        7.3643
```

**Exercise 12: There were 22,258 flights that left Bradley Airport in the year 2013. Those flights from Bradley were made by 12 airline carriers. Use dbGetQuery() with SELECT, FROM, WHERE, GROUP BY, and HAVING to query the flights table to do the following. Report your R command(s) (or just your SQL statement).**

1. Group the flights from Bradley in 2013 by airline carrier (carrier).
2. Find the average departure delay for each carrier (use AVG() with dep_delay).
3. Retrieve just the carriers whose average departure delays were longer than 10 minutes.
   Code:

```
dbGetQuery(conn = db_con,
           statement = "SELECT carrier, AVG(dep_delay) AS avg_dep_delay
           FROM flights
           WHERE origin = 'BDL' and year = 2013
           GROUP BY carrier
           HAVING avg_dep_delay > 10;")
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 1 imported as
## numeric
```

```
##   carrier avg_dep_delay
## 1      EV       10.7332
## 2      MQ       12.8813
## 3      WN       10.9046
```

## Section 15.8 Exercises

**Exercise 13: How would you modify the query below so that it returns instead the 6th-9th flight destinations?**

Original Query:

```
dbGetQuery(
  conn = db_con,
  statement = "SELECT dest, COUNT(*) AS numFlights, AVG(arr_delay) AS avg_arr_delay
FROM flights
WHERE year = 2013 AND origin = 'BDL'
GROUP BY dest
HAVING numFlights > 365 * 2
ORDER BY avg_arr_delay ASC
LIMIT 3, 4;"
)
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
##   dest numFlights avg_arr_delay
## 1  FLL       1011        0.2770
## 2  DFW       1062        0.7495
## 3  ATL       2277        4.4704
## 4  BWI       2613        5.0325
```

Modified Query:

```
dbGetQuery(
  conn = db_con,
  statement = "SELECT dest, COUNT(*) AS numFlights, AVG(arr_delay) AS avg_arr_delay
FROM flights
WHERE year = 2013 AND origin = 'BDL'
```

```
GROUP BY dest
HAVING numFlights > 365 * 2
ORDER BY avg_arr_delay ASC
LIMIT 5, 4;"
)
```

```
## Warning in .local(conn, statement, ...): Decimal MySQL column 2 imported as
## numeric
```

```
##   dest numFlights avg_arr_delay
## 1  ATL       2277        4.4704
## 2  BWI       2613        5.0325
## 3  ORD       2657        7.3643
## 4  MCO       1789        8.3784
```

## Section 15.9 Exercises

**Exercise 14: This exercise involves a JOIN clause. There were 65 flights that left Bradley Airport on June 26th, 2013. For this query, you'll need to join the airports table onto the flights table, matching the destination airport code (dest column) of flights to the airport code (faa column) in airports. You'll also need to see the destinations, flight numbers, and airline carrier codes (dest, flight, and carrier columns) from the flights table and the full airport names (name column) from the airports table in the result set. Use dbGetQuery() to answer the following problem. List the full name of the destination airport of flight EV 4714 from from Bradley on June 26th, 2013.**

Code:

```
dbGetQuery(conn = db_con,
           statement = "SELECT carrier, dest, flight, airports.name
           FROM flights
           JOIN airports ON flights.dest = airports.faa
           WHERE origin = 'BDL' AND year = 2013 AND month = 6 AND day = 26
           HAVING carrier = 'EV' AND flight = 4714;")
```

```
##   carrier dest flight                name
## 1      EV  EWR   4714 Newark Liberty Intl
```

**Exercise 15: This exercise involves a JOIN clause.There were 65 flights that left Bradley Airport on June 26th, 2013. For this query, you'll need to join the carriers table onto the flights table, matching the airline carrier code (carrier column) of flights to the carrier code (carrier column) in carriers. You'll also need to see the destinations and flight numbers and (dest and flight columns) from the flights table and the full carrier names (name column) from the carriers table in the result set. Use dbGetQuery() to answer the following problems.**

**a) List the full airline carrier name and flight number for all flights between Bradley Airport (BDL) and MSP on June 26th, 2013.**
Code:

```
dbGetQuery(conn = db_con,
           statement = "SELECT origin, dest, flight, flights.carrier, carriers.name
           FROM flights
           JOIN carriers ON flights.carrier = carriers.carrier
           WHERE year = 2013 AND month = 6 AND day = 26 AND origin = 'BDL'
           HAVING dest = 'MSP';")
```

```
##   origin dest flight carrier                 name
## 1    BDL  MSP    797      DL Delta Air Lines Inc.
## 2    BDL  MSP   3338      9E    Endeavor Air Inc.
## 3    BDL  MSP   1226      DL Delta Air Lines Inc.
```

**b) List the destination airport codes and flight numbers for the three flights from Bradley made by Mesa Airlines Inc. on June 26th, 2013.**
Code:

```
dbGetQuery(conn = db_con,
           statement = "SELECT origin, dest, flight, flights.carrier, carriers.name AS name
           FROM flights
           JOIN carriers ON flights.carrier = carriers.carrier
           WHERE year = 2013 AND month = 6 AND day = 26 AND origin = 'BDL'
           HAVING name = 'Mesa Airlines Inc.';")
```

```
##   origin dest flight carrier               name
## 1    BDL  ORD   3755      YV Mesa Airlines Inc.
## 2    BDL  ORD   3737      YV Mesa Airlines Inc.
## 3    BDL  IAD   3745      YV Mesa Airlines Inc.
```

**Exercise 16: This exercise concerns LEFT JOIN. There were 50 flights that left Palm Beach International Airport ('PBI') on June 26th, 2013. Explain in words why, below, LEFT JOIN returns all 50 flights (with NA as one of the destination airport names), but JOIN only returns 49.**

Code:

```
dbGetQuery(conn = db_con,
           statement = "SELECT origin, dest,
           name AS dest_name, flight, carrier
           FROM flights
           LEFT JOIN airports ON flights.dest = airports.faa
           WHERE year = 2013 AND month = 6 AND
           day = 26 AND origin = 'PBI';")
```

```
##    origin dest                          dest_name flight carrier
## 1     PBI  ATL    Hartsfield Jackson Atlanta Intl   2271      DL
## 2     PBI  JFK                John F Kennedy Intl    154      B6
## 3     PBI  PHL                  Philadelphia Intl   1894      US
## 4     PBI  HPN                     Westchester Co    572      B6
## 5     PBI  EWR                Newark Liberty Intl   1264      UA
## 6     PBI  CLT             Charlotte Douglas Intl   1410      US
```

```
## 7     PBI  ATL    Hartsfield Jackson Atlanta Intl    688      DL
## 8     PBI  BOS General Edward Lawrence Logan Intl     22      B6
## 9     PBI  DFW             Dallas Fort Worth Intl    499      AA
## 10    PBI  ATL    Hartsfield Jackson Atlanta Intl    456      FL
## 11    PBI  EWR                Newark Liberty Intl   1741      UA
## 12    PBI  LGA                         La Guardia   2358      DL
## 13    PBI  ATL    Hartsfield Jackson Atlanta Intl   1184      DL
## 14    PBI  EWR                Newark Liberty Intl   1147      UA
## 15    PBI  ATL    Hartsfield Jackson Atlanta Intl   1134      DL
## 16    PBI  BWI          Baltimore Washington Intl   1225      WN
## 17    PBI  PHL                  Philadelphia Intl    280      FL
## 18    PBI  BDL                       Bradley Intl   1142      B6
## 19    PBI  LGA                         La Guardia   1922      DL
## 20    PBI  CLT             Charlotte Douglas Intl   1652      US
## 21    PBI  DCA      Ronald Reagan Washington Natl   1126      US
## 22    PBI  EWR                Newark Liberty Intl    544      B6
## 23    PBI  ATL    Hartsfield Jackson Atlanta Intl    697      DL
## 24    PBI  ISP             Long Island Mac Arthur   2422      WN
## 25    PBI  SJU                                <NA>   1327      B6
## 26    PBI  LGA                         La Guardia    362      B6
## 27    PBI  CLT             Charlotte Douglas Intl   1412      US
## 28    PBI  DTW             Detroit Metro Wayne Co   1930      DL
## 29    PBI  PHL                  Philadelphia Intl   1247      US
## 30    PBI  ATL    Hartsfield Jackson Atlanta Intl   1852      DL
## 31    PBI  JFK                John F Kennedy Intl     54      B6
## 32    PBI  LGA                         La Guardia   1174      DL
## 33    PBI  CLT             Charlotte Douglas Intl   1203      US
## 34    PBI  ATL    Hartsfield Jackson Atlanta Intl    768      DL
## 35    PBI  EWR                Newark Liberty Intl    249      UA
## 36    PBI  IAH       George Bush Intercontinental   6123      EV
## 37    PBI  HPN                    Westchester Co     1972      B6
## 38    PBI  ATL    Hartsfield Jackson Atlanta Intl   1034      DL
## 39    PBI  ISP             Long Island Mac Arthur    197      WN
## 40    PBI  DFW             Dallas Fort Worth Intl   1483      AA
## 41    PBI  LGA                         La Guardia    834      DL
## 42    PBI  JFK                John F Kennedy Intl   1254      B6
## 43    PBI  LGA                         La Guardia   1262      B6
## 44    PBI  ATL    Hartsfield Jackson Atlanta Intl   1053      DL
## 45    PBI  ATL    Hartsfield Jackson Atlanta Intl    151      FL
## 46    PBI  BOS General Edward Lawrence Logan Intl    122      B6
## 47    PBI  HPN                    Westchester Co     1072      B6
## 48    PBI  JFK                John F Kennedy Intl    454      B6
## 49    PBI  BWI          Baltimore Washington Intl    416      WN
## 50    PBI  ORD                 Chicago Ohare Intl   1759      AA
```

```
dbGetQuery(conn = db_con,
           statement = "SELECT origin, dest,
           name AS dest_name, flight, carrier
           FROM flights
           JOIN airports ON flights.dest = airports.faa
           WHERE year = 2013 AND month = 6 AND
           day = 26 AND origin = 'PBI';")
```

```
##    origin dest                         dest_name flight carrier
```

```
## 1     PBI  ATL    Hartsfield Jackson Atlanta Intl   2271      DL
## 2     PBI  JFK                John F Kennedy Intl    154      B6
## 3     PBI  PHL                  Philadelphia Intl   1894      US
## 4     PBI  HPN                     Westchester Co    572      B6
## 5     PBI  EWR                Newark Liberty Intl   1264      UA
## 6     PBI  CLT             Charlotte Douglas Intl   1410      US
## 7     PBI  ATL    Hartsfield Jackson Atlanta Intl    688      DL
## 8     PBI  BOS General Edward Lawrence Logan Intl     22      B6
## 9     PBI  DFW             Dallas Fort Worth Intl    499      AA
## 10    PBI  ATL    Hartsfield Jackson Atlanta Intl    456      FL
## 11    PBI  EWR                Newark Liberty Intl   1741      UA
## 12    PBI  LGA                          La Guardia   2358      DL
## 13    PBI  ATL    Hartsfield Jackson Atlanta Intl   1184      DL
## 14    PBI  EWR                Newark Liberty Intl   1147      UA
## 15    PBI  ATL    Hartsfield Jackson Atlanta Intl   1134      DL
## 16    PBI  BWI          Baltimore Washington Intl   1225      WN
## 17    PBI  PHL                  Philadelphia Intl    280      FL
## 18    PBI  BDL                       Bradley Intl   1142      B6
## 19    PBI  LGA                          La Guardia   1922      DL
## 20    PBI  CLT             Charlotte Douglas Intl   1652      US
## 21    PBI  DCA      Ronald Reagan Washington Natl   1126      US
## 22    PBI  EWR                Newark Liberty Intl    544      B6
## 23    PBI  ATL    Hartsfield Jackson Atlanta Intl    697      DL
## 24    PBI  ISP             Long Island Mac Arthur   2422      WN
## 25    PBI  LGA                          La Guardia    362      B6
## 26    PBI  CLT             Charlotte Douglas Intl   1412      US
## 27    PBI  DTW             Detroit Metro Wayne Co   1930      DL
## 28    PBI  PHL                  Philadelphia Intl   1247      US
## 29    PBI  ATL    Hartsfield Jackson Atlanta Intl   1852      DL
## 30    PBI  JFK                John F Kennedy Intl     54      B6
## 31    PBI  LGA                          La Guardia   1174      DL
## 32    PBI  CLT             Charlotte Douglas Intl   1203      US
## 33    PBI  ATL    Hartsfield Jackson Atlanta Intl    768      DL
## 34    PBI  EWR                Newark Liberty Intl    249      UA
## 35    PBI  IAH       George Bush Intercontinental   6123      EV
## 36    PBI  HPN                     Westchester Co   1972      B6
## 37    PBI  ATL    Hartsfield Jackson Atlanta Intl   1034      DL
## 38    PBI  ISP             Long Island Mac Arthur    197      WN
## 39    PBI  DFW             Dallas Fort Worth Intl   1483      AA
## 40    PBI  LGA                          La Guardia    834      DL
## 41    PBI  JFK                John F Kennedy Intl   1254      B6
## 42    PBI  LGA                          La Guardia   1262      B6
## 43    PBI  ATL    Hartsfield Jackson Atlanta Intl   1053      DL
## 44    PBI  ATL    Hartsfield Jackson Atlanta Intl    151      FL
## 45    PBI  BOS General Edward Lawrence Logan Intl    122      B6
## 46    PBI  HPN                     Westchester Co   1072      B6
## 47    PBI  JFK                John F Kennedy Intl    454      B6
## 48    PBI  BWI          Baltimore Washington Intl    416      WN
## 49    PBI  ORD                 Chicago Ohare Intl   1759      AA
```

Left join will include all 50 flights because the left join function shows all the flights in the flights destination airports. Join will only show rows that are the same based on destination airport and the faa code for the airports dataset.

## Section 15.10 Exercises

**Exercise 17: This exercise concerns the use of UNION. Describe in words which sets of flights will be combined to comprise the result set of the following query, then check your answer.**

Code:

```
dbGetQuery(conn = db_con,
           statement = "(SELECT year, month, day, origin,
           dest, flight, carrier
           FROM flights
           WHERE year = 2013 AND month = 6 AND day = 26
           AND origin = 'BDL' AND dest = 'ORD')
           UNION
           (SELECT year, month, day, origin, dest,
           flight, carrier
           FROM flights
           WHERE year = 2013 AND month = 6 AND day = 26
           AND origin = 'MSP' AND dest = 'JFK');")
```

```
##    year month day origin dest flight carrier
## 1  2013     6  26    BDL  ORD   3755      YV
## 2  2013     6  26    BDL  ORD   3133      MQ
## 3  2013     6  26    BDL  ORD   3127      MQ
## 4  2013     6  26    BDL  ORD   3645      MQ
## 5  2013     6  26    BDL  ORD   3842      EV
## 6  2013     6  26    BDL  ORD   3146      MQ
## 7  2013     6  26    BDL  ORD   3737      YV
## 8  2013     6  26    BDL  ORD   3706      MQ
## 9  2013     6  26    BDL  ORD   1740      UA
## 10 2013     6  26    MSP  JFK   3541      9E
## 11 2013     6  26    MSP  JFK   3356      9E
## 12 2013     6  26    MSP  JFK   3543      9E
```

The command above will combine rows from the year 2013, month of June, and the 26th day from the flights dataset to include flights that occured from BDL to ORD and from MSP to JFK airports. Its similar to doing the dbGetQuery command twice but shortens it down to a single command.

## Section 15.11 Exercises

**Exercise 18: In this section, we used a subquery to determine whether Bradley Airport had any flights to Alaska or Hawaii (time zone, tz, less than -8) in 2013. Did Bradley have any flights to airports in the Pacific time zone (tz less than -7) in 2013? If so, which airports (three-letter codes) in the Pacific time zone were the flights' destinations?**

Code:

```
head(dbGetQuery(conn = db_con,
           statement = "SELECT dest
                        FROM flights
```

```
                          WHERE year = 2013
                            AND origin = 'BDL'
                            AND dest IN
                              (SELECT faa
                               FROM airports
                               WHERE tz < -7);"), 2)
```

```
##   dest
## 1  LAX
## 2  LAS
```

The result from the command above shows the airports in the pacific time zone that left Bradley airport. Only LAX and LAS resulted from the above.

**Exercise 19: In this section, we used a subquery to determine whether Bradley Airport had any flights to Alaska or Hawaii (time zone, tz, less than -8) in 2013. Did John F Kennedy Airport (JFK) have any flights to Alaska or Hawaii (time zone, tz, less than -8) in 2013? If so, which airport(s) (three-letter codes) in Alaska or Hawaii were the flights' destinations?**

Code:

```
head(dbGetQuery(conn = db_con,
           statement = "SELECT dest
                        FROM flights
                        WHERE year = 2013
                          AND origin = 'JFK'
                          AND dest IN
                            (SELECT faa
                             FROM airports
                             WHERE tz < -8);"), 1)
```

```
##   dest
## 1  HNL
```

The command above shows the airports flown to from JFK. The only airport flown from JFK to the pacific time zone was HNL which is Honolulu airport. It was flown to about 356 times in the year 2013.

## Section 15.12 Exercises

**Exercise 20: In Exercise 2, SELECT and FROM were used to query the flights table to retrieve just the carrier and tailnum variables (columns). Write a command using "dplyr" that performs this same query. Do not use dbGetQuery(). Report your R command(s).**

Code:

```
flights <- tbl(db_con, "flights")
carriers <- tbl(db_con, "carriers")
```

```
my.query <- filter(.data = flights, year == 2013 & month == 6 & day == 6 & origin == "BDL")

my.query.flights <- select(.data = flights, c(carrier, tailnum))

# my.query.carriers <- select(.data = carriers, c(name, tailnum))

head(my.query.flights, 5)
```

```
## # Source:   lazy query [?? x 2]
## # Database: mysql 5.7.33-log
## #   [@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:/airlines]
##   carrier tailnum
##   <chr>   <chr>
## 1 XE      N11137
## 2 B6      N659JB
## 3 B6      N563JB
## 4 XE      N16559
## 5 OO      N908SW
```

```
# head(my.query.carriers)
```

**Exercise 21: Report R command(s) involving "dplyr" and the logical operators & ("and"), | ("or"), and ! ("not") with the flights table to map to flights meeting the conditions:**

- a) Had an arrival delay of more than hours (arr_delay more than 120 minutes).
  Code:

```
head(filter(.data = flights, arr_delay > 120), 5)
```

```
## # Source:   lazy query [?? x 21]
## # Database: mysql 5.7.33-log
## #   [@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:/airlines]
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <int>    <int>          <int>
## 1  2010    10     1        1           2100       181      159           2320
## 2  2010    10     1        1           1920       281      230           2214
## 3  2010    10     1        7           2150       137      139           2337
## 4  2010    10     1       12           2045       207      136           2209
## 5  2010    10     1       20           2145       155      305             27
## # ... with 13 more variables: arr_delay <int>, carrier <chr>, tailnum <chr>,
## #   flight <int>, origin <chr>, dest <chr>, air_time <int>, distance <int>,
## #   cancelled <int>, diverted <int>, hour <int>, minute <int>, time_hour <chr>
```

- b) Flew to Houston (i.e. had a dest of IAH or HOU).
  Code

```
head(filter(.data = flights, dest == 'IAH' | dest == 'HOU'), 5)
```

```
## # Source:   lazy query [?? x 21]
## # Database: mysql 5.7.33-log
## #   [@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:/airlines]
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <int>    <int>          <int>
## 1  2010    10     1      559            605        -6      700            715
## 2  2010    10     1      602            605        -3      746            810
## 3  2010    10     1      615            615         0      710            715
## 4  2010    10     1      616            620        -4      702            710
## 5  2010    10     1      619            620        -1      836            905
## # ... with 13 more variables: arr_delay <int>, carrier <chr>, tailnum <chr>,
## #   flight <int>, origin <chr>, dest <chr>, air_time <int>, distance <int>,
## #   cancelled <int>, diverted <int>, hour <int>, minute <int>, time_hour <chr>
```

- c) Were operated by United, American, or Delta (i.e. had a carrier of UA, AA, or DL).
  Code:

```
head(filter(.data = flights, carrier == 'UA' | carrier == 'AA' | carrier == 'DL'), 5)
```

```
## # Source:   lazy query [?? x 21]
## # Database: mysql 5.7.33-log
## #   [@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:/airlines]
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <int>    <int>          <int>
## 1  2010    10     1        7             10        -3      451            500
## 2  2010    10     1       21             25        -4      537            535
## 3  2010    10     1       43             45        -2      528            535
## 4  2010    10     1      119             35        44      545            500
## 5  2010    10     1      538            535         3      900            845
## # ... with 13 more variables: arr_delay <int>, carrier <chr>, tailnum <chr>,
## #   flight <int>, origin <chr>, dest <chr>, air_time <int>, distance <int>,
## #   cancelled <int>, diverted <int>, hour <int>, minute <int>, time_hour <chr>
```

- d) Departed in summer (July, August, or September, i.e. month 7, 8, or 9).
  Code:

```
head(filter(.data = flights, month == 7 | month == 8 | month == 9), 5)
```

```
## # Source:   lazy query [?? x 21]
## # Database: mysql 5.7.33-log
## #   [@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:/airlines]
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <int>    <int>          <int>
## 1  2010     7     1        1           2359         2      506            517
## 2  2010     7     1        2             10        -8      447            500
## 3  2010     7     1        3              3         0      638            645
## 4  2010     7     1        4             10        -6      715            717
## 5  2010     7     1        5           2359         6      449            459
## # ... with 13 more variables: arr_delay <int>, carrier <chr>, tailnum <chr>,
## #   flight <int>, origin <chr>, dest <chr>, air_time <int>, distance <int>,
## #   cancelled <int>, diverted <int>, hour <int>, minute <int>, time_hour <chr>
```

- e) Departed between midnight and 6:00 AM, inclusive (dep_time at 2400 or between 0 and 600,inclusive).
  Code:

```
head(filter(.data = flights, dep_time == 2400 | dep_time >= 0 & dep_time <= 600), 5)
```

```
## # Source:   lazy query [?? x 21]
## # Database: mysql 5.7.33-log
## #   [@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:/airlines]
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <int>    <int>          <int>
## 1  2010    10     1        1           2100       181      159           2320
## 2  2010    10     1        1           1920       281      230           2214
## 3  2010    10     1        3           2355         8      339            334
## 4  2010    10     1        5           2200       125       41           2249
## 5  2010    10     1        7           2245        82      104           2347
## # ... with 13 more variables: arr_delay <int>, carrier <chr>, tailnum <chr>,
## #   flight <int>, origin <chr>, dest <chr>, air_time <int>, distance <int>,
## #   cancelled <int>, diverted <int>, hour <int>, minute <int>, time_hour <chr>
```

- f) Were operated by United (carrier UA), departed in July (month 7), and had an arrival delay of more than two hours (arr_delay more than 120 minutes).
  Code:

```
head(filter(.data = flights, carrier == 'UA' & month == 7 & arr_delay > 120), 5)
```

```
## # Source:   lazy query [?? x 21]
## # Database: mysql 5.7.33-log
## #   [@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:/airlines]
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <int>    <int>          <int>
## 1  2010     7     1       23           2130       173      148           2305
## 2  2010     7     1     1600           1216       224     1843           1502
## 3  2010     7     1     2049           1631       258     2210           1756
## 4  2010     7     1     2107           1415       412       18           1744
## 5  2010     7     1     2148           1735       253     2320           1936
## # ... with 13 more variables: arr_delay <int>, carrier <chr>, tailnum <chr>,
## #   flight <int>, origin <chr>, dest <chr>, air_time <int>, distance <int>,
## #   cancelled <int>, diverted <int>, hour <int>, minute <int>, time_hour <chr>
```

## Section 15.13 Exercises

**Exercise 22: Create an in-memory database, then add two tables, the who and population data frames.**

Setup:

```
# Connect to the computer's memory using dbConnect():
db_con <- dbConnect(drv = SQLite(),
                    dbname = ":memory:")
```

```
#Load the who table:
dbWriteTable(conn = db_con,
             name = "whoTable",
             value = who)
#Load the population table:
dbWriteTable(conn = db_con,
             name = "popTable",
             value = population)
```

**a) Now us dbGetQuery() with SELECT and FROM to query the who table to retrieve just the country, year, and new_sp_m014 variables (columns). Report your R command(s) (or just your SQL statement).**
Code:

```
head(dbGetQuery(conn = db_con,
           statement = "SELECT country, year,
           new_sp_m014
           FROM whoTable;"), 5)
```

```
##       country year new_sp_m014
## 1 Afghanistan 1980          NA
## 2 Afghanistan 1981          NA
## 3 Afghanistan 1982          NA
## 4 Afghanistan 1983          NA
## 5 Afghanistan 1984          NA
```

**b) Now modify your query from part a using a WHERE clause so that it only retrieves rows corresponding to the U.S. ("United States of America") more recently than 2009 (year >= 2010) . Report your R command(s) (or just your SQL statement).**
Code:

```
dbGetQuery(conn = db_con,
           statement = "SELECT country, year,
           new_sp_m014
           FROM whoTable
           WHERE country = 'United States of America' AND
           year >= 2010;")
```

```
##                    country year new_sp_m014
## 1 United States of America 2010           5
## 2 United States of America 2011          12
## 3 United States of America 2012          10
## 4 United States of America 2013          NA
```

**Exercise 23: Use GROUP BY with AVG() and ORDER BY to determine the country for which the average number of new TB cases for the (new_sp_m014) group was lowest in the year 2013. Note: It's okay if you end up with NAs. Report your R command(s) (or just your SQL statement).**

Code:

```
head(dbGetQuery(conn = db_con,
                statement = "SELECT country, new_sp_m014,
                AVG(new_sp_m014) AS avgNew_sp_m014
                FROM whoTable
                WHERE year = 2013
                GROUP BY country
                ORDER BY avgNew_sp_m014 ASC;"), 5)
```

```
##          country new_sp_m014 avgNew_sp_m014
## 1    Afghanistan          NA             NA
## 2        Albania          NA             NA
## 3        Algeria          NA             NA
## 4 American Samoa          NA             NA
## 5        Andorra          NA             NA
```