Practicum1.CS5200Su21

Authors: Sunjit Dhillon [dhillon.su@northeastern.edu] MManzur Morshed [morshed.mm@northeastern.edu]

Link to LucidChart Diagram: https://lucid.app/lucidchart/05c683b2-bfc9-456a-8236-e299310aca73/edit?invitationId=inv_cea08688-d283-4f6b-9021-73ff624cccb3

```
# Load the required libraries
library(RMySQL)
```

```
## Loading required package: DBI
```

```
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
```

```
library(ggplot2)
```

```
# Settings
db_user <- 'root'
db_password <- 'sunjit22'
db_name <- 'Practicum'

db_host <- 'localhost'
db_port <- 3306

# Connect to DB
mydb <-  dbConnect(MySQL(), user = db_user, password = db_password,
                 dbname = db_name, host = db_host, port = db_port)
```

```
# Read data from .csv file into a dataframe
path <- "/Users/sunjitdhillon/Downloads"
fn <- "BirdStrikesData.csv"

fileName <- paste(path, fn, sep = "/")

df <- read.csv(fileName, header = TRUE, stringsAsFactors = FALSE)
```

```
set global local_infile=true;
```

```
# Create a temporary dataframe and rename column names
temp <- df
temp <- rename(temp, record_id = Record.ID,
               aircraft_type=Aircraft..Type,
               airport_name=Airport..Name,
               altitude_bin=Altitude.bin,
               aircraft_make_model=Aircraft..Make.Model,
               wildlife_number_struck=Wildlife..Number.struck,
               wildlife_number_struck_actual=Wildlife..Number.Struck.Actual,
               impact_to_flight=Effect..Impact.to.flight,
               flight_date=FlightDate,
               indicated_damage=Effect..Indicated.Damage,
               aircraft_number_of_engines=Aircraft..Number.of.engines.,
               aircraft_airline_operator=Aircraft..Airline.Operator,
               origin_state=Origin.State,
               phase_of_flight=When..Phase.of.flight,
               conditions_precipitation=Conditions..Precipitation,
               conditions_sky=Conditions..Sky,
               remains_collected=Remains.of.wildlife.collected.,
               remains_sent_to_smithsonian=Remains.of.wildlife.sent.to.Smithsonian,
               is_aircraft_large=Is.Aircraft.Large.,
               species=Wildlife..Species,
               size=Wildlife..Size,
               pilot_warned_of_birds_or_wildlife=Pilot.warned.of.birds.or.wildlife.,
               total_cost_in_dollars=Cost..Total..,
               feet_above_ground=Feet.above.ground,
               number_of_people_injured=Number.of.people.injured,
               remarks = Remarks
)
head(temp)
```

```
##   record_id aircraft_type                airport_name altitude_bin
## 1    202152      Airplane                LAGUARDIA NY    > 1000 ft
## 2    208159      Airplane DALLAS/FORT WORTH INTL ARPT    < 1000 ft
## 3    207601      Airplane           LAKEFRONT AIRPORT    < 1000 ft
## 4    215953      Airplane         SEATTLE-TACOMA INTL    < 1000 ft
## 5    219878      Airplane                NORFOLK INTL    < 1000 ft
## 6    218432      Airplane         GUAYAQUIL/S BOLIVAR    < 1000 ft
##   aircraft_make_model wildlife_number_struck wildlife_number_struck_actual
## 1           B-737-400               Over 100                           859
## 2               MD-80               Over 100                           424
## 3               C-500               Over 100                           261
## 4           B-737-400               Over 100                           806
## 5        CL-RJ100/200               Over 100                           942
## 6               A-300               Over 100                           537
##        impact_to_flight     flight_date indicated_damage
## 1      Engine Shut Down 11/23/2000 0:00    Caused damage
## 2                  None  7/25/2001 0:00    Caused damage
## 3                  None  9/14/2001 0:00        No damage
## 4 Precautionary Landing   9/5/2002 0:00        No damage
## 5                  None  6/23/2003 0:00        No damage
## 6                 None  7/24/2003 0:00        No damage
##   aircraft_number_of_engines aircraft_airline_operator origin_state
## 1                          2               US AIRWAYS*     New York
## 2                          2         AMERICAN AIRLINES        Texas
## 3                          2                  BUSINESS    Louisiana
## 4                          2           ALASKA AIRLINES   Washington
## 5                          2           COMAIR AIRLINES     Virginia
## 6                          2         AMERICAN AIRLINES          N/A
##   phase_of_flight conditions_precipitation remains_collected
## 1           Climb                     None             FALSE
## 2    Landing Roll                     None             FALSE
## 3        Approach                     None             FALSE
## 4           Climb                     None              TRUE
## 5        Approach                     None             FALSE
## 6    Take-off run                     None             FALSE
##   remains_sent_to_smithsonian
## 1                       FALSE
## 2                       FALSE
## 3                       FALSE
## 4                       FALSE
## 5                       FALSE
## 6                       FALSE
##
## 1  FLT 753. PILOT REPTD A HUNDRED BIRDS ON UNKN TYPE. #1 ENG WAS SHUT DOWN AND DIVERTED TO EWR. SLIG
## 2
## 3
## 4 NOTAM WARNING. 26 BIRDS HIT THE A/C, FORCING AN EMERGENCY LDG. 77 BIRDS WERE FOUND DEAD ON RWY/TWY
## 5
## 6
##     size conditions_sky               species pilot_warned_of_birds_or_wildlife
## 1 Medium       No Cloud Unknown bird - medium                                 N
## 2  Small     Some Cloud           Rock pigeon                                 Y
## 3  Small       No Cloud     European starling                                 N
## 4  Small     Some Cloud     European starling                                 Y
## 5  Small       No Cloud     European starling                                 N
## 6  Small       No Cloud  Unknown bird - small                                 N
##   total_cost_in_dollars feet_above_ground number_of_people_injured
## 1                30,736             1,500                        0
## 2                     0                 0                        0
## 3                     0                50                        0
## 4                     0                50                        0
## 5                     0                50                        0
## 6                     0                 0                        0
##   is_aircraft_large
## 1               Yes
## 2                No
## 3                No
## 4               Yes
## 5                No
## 6                No
```

Data Cleaning: The aircraft_number_of_engines refers to the number of engines in an aircraft, which must be an integer value. Based on analysis of data, we found that value of aircraft_number_of_engines corresponding to record id 206990 is ‘C’. For uniformity of data type, we assume that aircraft_number_of_engines corresponding to aircraft_make_model = ‘RKWLTRBO 690’ is 2.

```
r<-temp[which(temp$aircraft_make_model=='RKWLTRBO 690'), ]
r <- r %>% select(record_id, aircraft_make_model, aircraft_number_of_engines)
r
```

```
##       record_id aircraft_make_model aircraft_number_of_engines
## 209      206414        RKWLTRBO 690                          2
## 2520     253426        RKWLTRBO 690                          2
## 3761     308571        RKWLTRBO 690                          2
## 3776     308605        RKWLTRBO 690                          2
## 4448     224822        RKWLTRBO 690                          2
## 5031     202354        RKWLTRBO 690                          2
## 6641     207676        RKWLTRBO 690                          2
## 6665     206990        RKWLTRBO 690                          C
## 7270     214807        RKWLTRBO 690                          2
## 9280     223234        RKWLTRBO 690                          2
## 10129    231030        RKWLTRBO 690                          2
## 11973    236934        RKWLTRBO 690                          2
## 12085    237755        RKWLTRBO 690                          2
## 13140    235072        RKWLTRBO 690                          2
## 13166    244728        RKWLTRBO 690                          2
## 13201    239327        RKWLTRBO 690                          2
## 15089    249710        RKWLTRBO 690                          2
## 15194    245154        RKWLTRBO 690                          2
## 15856    252612        RKWLTRBO 690                          2
## 15947    252105        RKWLTRBO 690                          2
## 18168    263108        RKWLTRBO 690                          2
## 19856    267165        RKWLTRBO 690                          2
## 20209    269448        RKWLTRBO 690                          2
## 22156    305312        RKWLTRBO 690                          2
## 23145    310904        RKWLTRBO 690                          2
```

```
# Replace 'C' with 2 in temp dataframe.
temp["aircraft_number_of_engines"][temp["aircraft_number_of_engines"] ==
                                     'C'] <- 2
```

```
# Create dataframe Aircraft_df to store all distinct entries of aircrafts

Aircraft_df <- select(temp, aircraft_make_model, aircraft_number_of_engines,
                      aircraft_type, is_aircraft_large)
Aircraft_df <- distinct(Aircraft_df)
aircraft_id <- seq_len(nrow(Aircraft_df))
Aircraft_df <- cbind(aircraft_id, Aircraft_df)
```

```
DROP TABLE IF EXISTS Aircraft
```

```
CREATE TABLE Aircraft(
               aircraft_id INTEGER NOT NULL,
       aircraft_make_model TEXT,
aircraft_number_of_engines INTEGER,
             aircraft_type TEXT,
         is_aircraft_large TEXT,
CONSTRAINT ck_categorical_aircraft_size CHECK (is_aircraft_large IN ("Yes","No", null)),
PRIMARY KEY (aircraft_id)
)
```

```
# Write data from dataframe Aircraft_df to table Aircraft

dbWriteTable(mydb, "Aircraft", Aircraft_df, append = TRUE, row.names = FALSE)
```

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

```
SELECT * FROM Aircraft LIMIT 10
```

Table 1: Displaying records 1 - 10

aircraft_id	aircraft_make_model	aircraft_number_of_engines	aircraft_type	is_aircraft_large
1	B-737-400	2	Airplane	Yes
2	MD-80	2	Airplane	No
3	C-500	2	Airplane	No
4	CL-RJ100/200	2	Airplane	No
5	A-300	2	Airplane	No
6	LEARJET-25	2	Airplane	No
7	A-320	2	Airplane	No
8	DC-9-30	2	Airplane	No
9	A-330	2	Airplane	No
10	FOKKER F100	2	Airplane	No

```
# Create dataframe Airport_df to store all distinct entries of airports.

Airport_df <- select(temp, airport_name, origin_state)
Airport_df <- distinct(Airport_df)
airport_id <- seq_len(nrow(Airport_df))
Airport_df <- cbind(airport_id, Airport_df)
```

```
DROP TABLE IF EXISTS Airport
```

```
CREATE TABLE Airport(
  airport_id INTEGER NOT NULL,
airport_name TEXT,
origin_state TEXT,
PRIMARY KEY(airport_id)
)
```

```
# Write data from dataframe Airport_df to table Airport
dbWriteTable(mydb, "Airport", Airport_df, append = TRUE, row.names = FALSE)
```

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

```
SELECT * FROM Airport LIMIT 10
```

Table 2: Displaying records 1 - 10

airport_id	airport_name	origin_state
1	LAGUARDIA NY	New York
2	DALLAS/FORT WORTH INTL ARPT	Texas
3	LAKEFRONT AIRPORT	Louisiana
4	SEATTLE-TACOMA INTL	Washington
5	NORFOLK INTL	Virginia
6	GUAYAQUIL/S BOLIVAR	N/A
7	NEW CASTLE COUNTY	Delaware
8	WASHINGTON DULLES INTL ARPT	DC
9	ATLANTA INTL	Georgia
10	ORLANDO SANFORD INTL AIRPORT	Florida

```
# Create dataframe Wildlife_df to store all distinct entries of Wildlife species.

Wildlife_df <- select(temp, species, size)
Wildlife_df <- distinct(Wildlife_df)
wildlife_id <- seq_len(nrow(Wildlife_df))
Wildlife_df <- cbind(wildlife_id, Wildlife_df)
```

```
DROP TABLE IF EXISTS Wildlife
```

```
CREATE TABLE Wildlife(
wildlife_id INTEGER NOT NULL,
    species TEXT,
       size TEXT,
CONSTRAINT ck_categorical_size CHECK (size IN ("Small","Medium","Large", null)),
PRIMARY KEY (wildlife_id)
)
```

```
# Write data from dataframe Wildlife_df to table Wildlife

dbWriteTable(mydb, "Wildlife", Wildlife_df, append = TRUE, row.names = FALSE)
```

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

```
SELECT * FROM Wildlife LIMIT 10
```

Table 3: Displaying records 1 - 10

wildlife_id	species	size
1	Unknown bird - medium	Medium
2	Rock pigeon	Small
3	European starling	Small
4	Unknown bird - small	Small
5	Canada goose	Large

wildlife_id	species	size
6	Snow goose	Large
7	Black-headed munia	Small
8	Ring-billed gull	Medium
9	Sandhill crane	Large
10	Western meadowlark	Small

```r
# Add aircraft_id column to temp dataframe

for(i in 1:dim(temp)[1]) {
  for(j in 1:dim(Aircraft_df)[1]) {

    # Compare values of aircraft_make_model, aircraft_number_of_engines,
    # aircraft_type, is_aircraft_large in Aircraft_df and temp dataframe
    if (temp$aircraft_make_model[i]==Aircraft_df$aircraft_make_model[j]
        & temp$aircraft_number_of_engines[i]==Aircraft_df$aircraft_number_of_engines[j]
        & temp$aircraft_type[i]==Aircraft_df$aircraft_type[j]
        & temp$is_aircraft_large[i]==Aircraft_df$is_aircraft_large[j]) {

      temp$aircraft_id[i] <- j
      break
    }
  }
}
```

```r
# Add airport_id column to temp dataframe

for(i in 1:dim(temp)[1]) {
  for(j in 1:dim(Airport_df)[1]) {

    # Compare values of airport_name, origin_state in Airport_df & temp dataframe
    if (temp$airport_name[i]==Airport_df$airport_name[j] &
        temp$origin_state[i]==Airport_df$origin_state[j]) {
      temp$airport_id[i] <- j
      break
    }
  }
}
```

```r
# Create dataframe Flight_Detail_df
Flight_Detail_df <- select(temp, record_id, aircraft_id, airport_id,
                           flight_date, aircraft_airline_operator,
                           pilot_warned_of_birds_or_wildlife)
# Change the flight_date format
Flight_Detail_df$flight_date <- as.Date(Flight_Detail_df$flight_date,"%m/%d/%Y")
# Assume missing values to be a default date '1900-01-01'
Flight_Detail_df$flight_date[is.na(Flight_Detail_df$flight_date)] <- '1900-01-01'
```

```sql
DROP TABLE IF EXISTS Flight_Detail
```

```
CREATE TABLE Flight_Detail(
                        record_id INTEGER NOT NULL,
                      aircraft_id INTEGER NOT NULL,
                       airport_id INTEGER NOT NULL,
                      flight_date DATE DEFAULT(DATE_FORMAT('%Y-%m-%d','1900-01-01')) NOT NULL,
        aircraft_airline_operator TEXT,
pilot_warned_of_birds_or_wildlife TEXT,
CONSTRAINT ck_categorical_pilot_warned CHECK (pilot_warned_of_birds_or_wildlife
IN ("Y","N", null)),
PRIMARY KEY (record_id),
FOREIGN KEY (aircraft_id) REFERENCES Aircraft(aircraft_id) ON DELETE CASCADE,
FOREIGN KEY (airport_id) REFERENCES Airport(airport_id) ON DELETE CASCADE
)
```

```
# Write data from dataframe Flight_Detail_df to table Flight_Detail

dbWriteTable(mydb, "Flight_Detail", Flight_Detail_df, append = TRUE,
             row.names = FALSE)
```

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

```
SELECT * FROM Flight_Detail LIMIT 10
```

Table 4: Displaying records 1 - 10

record_id	aircraft_id	airport_id	flight_date	aircraft_airline_operator	pilot_warned_of_birds_or_wildlife
1195	29	37	2002-11-13	MILITARY	Y
3019	87	717	2002-10-10	MILITARY	Y
3500	29	37	2001-05-15	MILITARY	Y
3504	29	37	2001-05-23	MILITARY	Y
3597	83	123	2001-04-18	MILITARY	Y
4064	29	37	2000-04-06	MILITARY	Y
4074	123	180	2002-07-15	MILITARY	Y
4076	29	37	2002-07-15	MILITARY	Y
4090	80	114	2001-07-02	MILITARY	Y
4091	92	114	2001-07-07	MILITARY	Y

```
# Create dataframe Strike_Impact_df
Strike_Impact_df <- select(temp, record_id, impact_to_flight, indicated_damage,
                           number_of_people_injured, total_cost_in_dollars,
                           remarks)
```

```
DROP TABLE IF EXISTS Strike_Impact
```

```
CREATE TABLE Strike_Impact(
               record_id INTEGER NOT NULL,
        impact_to_flight TEXT,
        indicated_damage TEXT,
number_of_people_injured INTEGER,
   total_cost_in_dollars INTEGER,
                 remarks TEXT,
CONSTRAINT ck_categorical_impact CHECK (impact_to_flight IN ("Aborted Take-off",
"Engine Shut Down","None", "Other","Precautionary Landing", null)),
CONSTRAINT ck_categorical_damage CHECK (indicated_damage IN ("Caused damage",
"No damage", null)),
PRIMARY KEY (record_id),
FOREIGN KEY (record_id) REFERENCES Flight_Detail(record_id) ON DELETE CASCADE
)
```

```
# Write data from dataframe Strike_Impact_df to table Strike_Impact
dbWriteTable(mydb, "Strike_Impact", Strike_Impact_df, append = TRUE,
             row.names = FALSE)
```

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

```
SELECT * FROM Strike_Impact LIMIT 10
```

Table 5: Displaying records 1 - 10

record_id	impact_to_flight	indicated_damage	number_of_people_injured	total_cost_in_dollars	remarks
1195	None	No damage	0	0	None.
3019	Precautionary Landing	No damage	0	0	
3500	Precautionary Landing	No damage	0	0	
3504	Precautionary Landing	No damage	0	0	
3597	None	No damage	0	0	
4064	None	No damage	0	0	A bird struck the left inboard flap and one was ingested into the #7 engine intake.
4074	None	No damage	0	0	
4076	Precautionary Landing	No damage	0	0	During touch and go bird struck the top of the nose radome between the #! and #2 window.
4090	Precautionary Landing	No damage	0	0	
4091	Aborted Take-off	No damage	0	0	

```
# Create dataframe Strike_Condition_df
Strike_Condition_df <- select(temp, record_id, altitude_bin, feet_above_ground,
                              conditions_sky, conditions_precipitation,
                              phase_of_flight)
```

```
DROP TABLE IF EXISTS Strike_Condition
```

```
CREATE TABLE Strike_Condition(
             record_id INTEGER NOT NULL,
          altitude_bin TEXT,
     feet_above_ground INTEGER,
        conditions_sky TEXT,
conditions_precipitation TEXT,
       phase_of_flight TEXT,
CONSTRAINT ck_categorical_altitude_bin CHECK (altitude_bin IN ("< 1000 ft","> 1000 ft", null)),
CONSTRAINT ck_categorical_conditions_sky CHECK (conditions_sky IN ("No Cloud","Some Cloud","Overcast", n
CONSTRAINT ck_categorical_phase CHECK (phase_of_flight IN ("Approach","Climb", "Descent", "Landing roll
PRIMARY KEY (record_id),
FOREIGN KEY (record_id) REFERENCES Flight_Detail(record_id) ON DELETE CASCADE
)
```

```
# Write data from dataframe Strike_Condition_df to table Strike_Condition

dbWriteTable(mydb, "Strike_Condition", Strike_Condition_df, append = TRUE,
             row.names = FALSE)
```

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

```
SELECT * FROM Strike_Condition LIMIT 10
```

Table 6: Displaying records 1 - 10

record_id	altitude_bin	feet_above_ground	conditions_sky	conditions_precipitation	phase_of_flight
1195	> 1000 ft	2	Overcast	None	Approach
3019	< 1000 ft	400	No Cloud	None	Climb
3500	< 1000 ft	1	No Cloud	None	Approach
3504	> 1000 ft	1	No Cloud	None	Approach
3597	< 1000 ft	200	Some Cloud	None	Approach
4064	< 1000 ft	1	No Cloud	None	Approach
4074	< 1000 ft	0	No Cloud	None	Take-off run
4076	< 1000 ft	500	Some Cloud	None	Climb
4090	< 1000 ft	50	Some Cloud	None	Climb
4091	< 1000 ft	0	Some Cloud	None	Take-off run

```
# Add wildlife_id column to temp dataframe
for(i in 1:dim(temp)[1]) {
  for(j in 1:dim(Wildlife_df)[1]) {

    # Compare values of species and size in Wildlife_df & temp dataframe
    if (temp$species[i]==Wildlife_df$species[j] & temp$size[i]==Wildlife_df$size[j]) {
      temp$wildlife_id[i] <- j
      break
    }
  }
}
```

```
# Create dataframe Wildlife_Strike_df

Wildlife_Strike_df <- select(temp, record_id, wildlife_id, wildlife_number_struck,
                             wildlife_number_struck_actual, remains_collected,
                             remains_sent_to_smithsonian)
```

```
DROP TABLE IF EXISTS Wildlife_Strike
```

```
CREATE TABLE Wildlife_Strike(
                    record_id INTEGER NOT NULL,
                  wildlife_id INTEGER NOT NULL,
       wildlife_number_struck TEXT,
wildlife_number_struck_actual INTEGER,
            remains_collected TEXT,
  remains_sent_to_smithsonian TEXT,
CONSTRAINT ck_categorical_number_struck CHECK (wildlife_number_struck IN ("1","2 to 10","11 to 100","Ove
CONSTRAINT ck_categorical_remains_collected CHECK (remains_collected IN ("TRUE","FALSE", null)),
CONSTRAINT ck_categorical_remains_sent_to_smithsonian CHECK (remains_sent_to_smithsonian IN ("TRUE","FAL
PRIMARY KEY (record_id),
FOREIGN KEY (wildlife_id) REFERENCES wildlife(wildlife_id) ON DELETE CASCADE,
FOREIGN KEY (record_id) REFERENCES Flight_Detail(record_id) ON DELETE CASCADE
)
```

```
# Write data from dataframe Wildlife_Strike_df to table Wildlife_Strike

dbWriteTable(mydb, "Wildlife_Strike", Wildlife_Strike_df, append = TRUE,
             row.names = FALSE)
```

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

```
SELECT * FROM Wildlife_Strike LIMIT 10
```

Table 7: Displaying records 1 - 10

record_id	wildlife_id	wildlife_number_struck	wildlife_number_struck_actual	remains_collected	remains_sent_to_smithsonian
1195	14	2 to 10	9	FALSE	FALSE
3019	14	1	1	FALSE	FALSE
3500	14	1	1	FALSE	FALSE
3504	14	2 to 10	8	FALSE	FALSE
3597	28	1	1	TRUE	TRUE
4064	14	2 to 10	10	FALSE	FALSE
4074	26	2 to 10	5	TRUE	TRUE
4076	41	1	1	TRUE	TRUE
4090	14	2 to 10	5	FALSE	FALSE

record_id	wildlife_id	wildlife_number_struck	wildlife_number_struck_actual	remains_collected	remains_sent_to_smithsonian
4091	14	2 to 10	2	FALSE	FALSE

Ques 4.

```
SELECT aircraft_airline_operator, COUNT(DISTINCT record_id) AS count_bird_strikes
FROM Flight_Detail
WHERE record_id IN (SELECT record_id FROM Strike_Condition WHERE phase_of_flight IN ("Take-off run", "C
GROUP BY aircraft_airline_operator;
```

Table 8: Displaying records 1 - 10

aircraft_airline_operator	count_bird_strikes
ABX AIR	51
ACM AVIATION	1
ADI SHUTTLE GROUP	5
AER LINGUS	2
AEROMEXICO	1
AIR AMERICA/TOTAL AIR	1
AIR BC	2
AIR CANADA	34
AIR CANADA JAZZ	20
AIR CARGO CARRIERS	3

Ques 5.

```
SELECT airport_name, COUNT(f.record_id) as count
FROM Flight_Detail AS f
NATURAL JOIN Airport AS a
GROUP BY a.airport_name
HAVING count = (
SELECT MAX(x.count) FROM
(SELECT a.airport_name AS airport_name, count(f.record_id) as count
FROM Flight_Detail AS f
NATURAL JOIN Airport AS a
GROUP BY a.airport_name) x)
```

Table 9: 1 records

airport_name	count
DALLAS/FORT WORTH INTL ARPT	803

Ques 6.

```
SELECT EXTRACT(YEAR FROM flight_date) AS year, COUNT(record_id) AS count_bird_strikes
FROM Flight_Detail
GROUP BY year
ORDER BY year
```

Table 10: Displaying records 1 - 10

year	count_bird_strikes
1900	129
2000	1367
2001	1230
2002	1681
2003	1568
2004	1692
2005	1853
2006	2159
2007	2301
2008	2258

Ques 7.

```
# Create a dataframe containing counts of bird strike incidents grouped by year and phase of flight

sqlCmd = "SELECT EXTRACT(YEAR FROM f.flight_date) AS year, s.phase_of_flight, COUNT(*) AS count
FROM Flight_Detail AS f
NATURAL JOIN Strike_Condition AS s
GROUP BY year, s.phase_of_flight
HAVING year >= 2008 AND  year <= 2011
AND phase_of_flight IN ('Take-off run', 'Climb', 'Descent', 'Approach', 'Landing roll')"

df = dbGetQuery(mydb, sqlCmd)
df
```

```
##    year phase_of_flight count
## 1  2009           Climb   547
## 2  2008    Landing Roll   459
## 3  2008           Climb   398
## 4  2008    Take-off run   412
## 5  2008         Descent   103
## 6  2008        Approach   880
## 7  2009         Descent    97
## 8  2009    Take-off run   580
## 9  2009    Landing Roll   694
## 10 2009        Approach  1318
## 11 2010         Descent    80
## 12 2010        Approach  1291
## 13 2010    Landing Roll   682
## 14 2010           Climb   479
## 15 2010    Take-off run   583
## 16 2011    Take-off run   537
## 17 2011        Approach  1277
## 18 2011           Climb   493
## 19 2011    Landing Roll   604
## 20 2011         Descent    32
```

```
# Group the flight phases into 'Take-off/Climbing' and 'Descent/Approach/Landing'
```

```
 for(i in 1:dim(df)[1]) {
    if (df$phase_of_flight[i]=='Take-off run' ||  df$phase_of_flight[i]=='Climb') {
      df$phase_of_flight[i] <- 'Take-off/Climbing'
    }

   if (df$phase_of_flight[i]=='Approach' ||  df$phase_of_flight[i]=='Landing Roll'
       || df$phase_of_flight[i]=='Descent') {
      df$phase_of_flight[i] <- 'Descent/Approach/Landing'

    }
}
df
```

```
##    year          phase_of_flight count
## 1  2009        Take-off/Climbing   547
## 2  2008 Descent/Approach/Landing   459
## 3  2008        Take-off/Climbing   398
## 4  2008        Take-off/Climbing   412
## 5  2008 Descent/Approach/Landing   103
## 6  2008 Descent/Approach/Landing   880
## 7  2009 Descent/Approach/Landing    97
## 8  2009        Take-off/Climbing   580
## 9  2009 Descent/Approach/Landing   694
## 10 2009 Descent/Approach/Landing  1318
## 11 2010 Descent/Approach/Landing    80
## 12 2010 Descent/Approach/Landing  1291
## 13 2010 Descent/Approach/Landing   682
## 14 2010        Take-off/Climbing   479
## 15 2010        Take-off/Climbing   583
## 16 2011        Take-off/Climbing   537
## 17 2011 Descent/Approach/Landing  1277
## 18 2011        Take-off/Climbing   493
## 19 2011 Descent/Approach/Landing   604
## 20 2011 Descent/Approach/Landing    32
```

```
# Group the bird strike incidents by their total sum per year (grouped by flight phase)

df2 <-df %>%
  group_by(year, phase_of_flight) %>%
  summarise(count=sum(count))
```

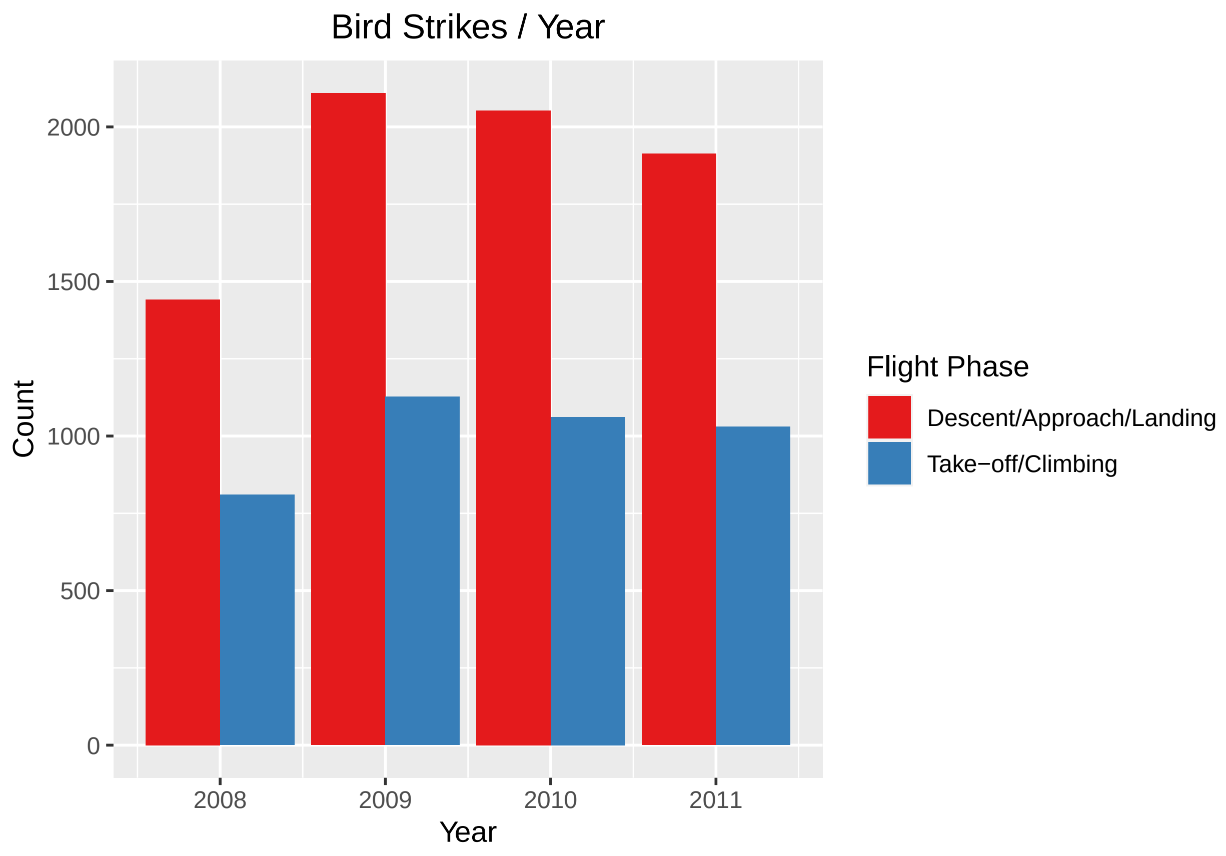
```
## `summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
```

```
df2
```

```
## # A tibble: 8 x 3
## # Groups:   year [4]
##    year phase_of_flight          count
##   <int> <chr>                    <dbl>
## 1  2008 Descent/Approach/Landing  1442
## 2  2008 Take-off/Climbing          810
## 3  2009 Descent/Approach/Landing  2109
## 4  2009 Take-off/Climbing         1127
## 5  2010 Descent/Approach/Landing  2053
## 6  2010 Take-off/Climbing         1062
## 7  2011 Descent/Approach/Landing  1913
## 8  2011 Take-off/Climbing         1030
```

```
# Plot the dataframe df2 to form a grouped bar chart

ggplot(df2, aes(year, count, fill = phase_of_flight)) +
  geom_bar(stat="identity", position = "dodge") +
  scale_fill_brewer("Flight Phase", palette = "Set1") +
   labs(y="Count", x = "Year") +
ggtitle("Bird Strikes / Year") +
  theme(plot.title = element_text(hjust = 0.5))
```



Ques 8.

```
DROP PROCEDURE IF EXISTS Delete_Flight_Detail
```

// The procedure Delete_Flight_Detail deletes bird strike incident record // corresponding to the record id entered as a parameter

```
CREATE PROCEDURE Delete_Flight_Detail(IN id_to_delete INTEGER)
    BEGIN
      DELETE FROM Flight_Detail
      WHERE record_id=id_to_delete;
    END
```

// Before calling the procedure:

```
SELECT * FROM Flight_Detail where record_id = 1195
```

Table 11: 1 records

record_id	aircraft_id	airport_id	flight_date	aircraft_airline_operator	pilot_warned_of_birds_or_wildlife
1195	29	37	2002-11-13	MILITARY	Y

// Call the procedure

```
CALL Delete_Flight_Detail(1195)
```

// After calling the procedure, the record corresponding to record id 1195 has // been deleted

```
SELECT * FROM Flight_Detail where record_id = 1195
```

Table 12: 0 records

record_id	aircraft_id	airport_id	flight_date	aircraft_airline_operator	pilot_warned_of_birds_or_wildlife

```
dbDisconnect(mydb)
```

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