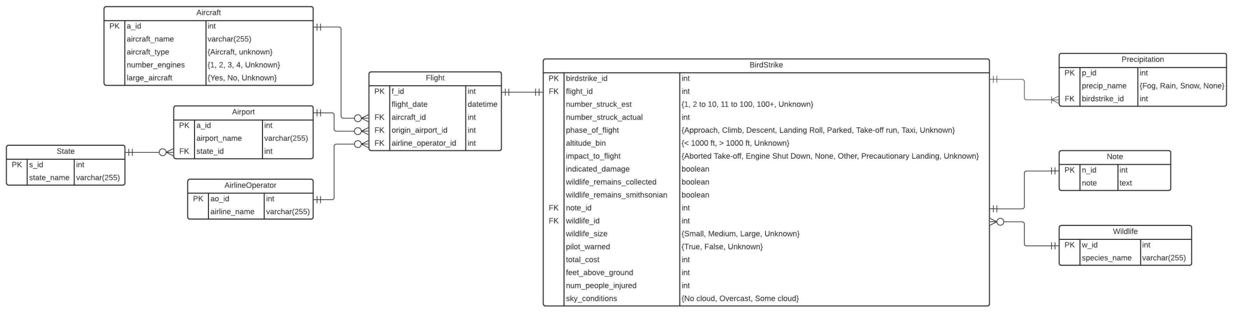
Practicum 1 DB

ERD


Aircraft
PK a_id int
aircraft_name varchar(255)
aircraft_type {Aircraft, unknown}
number_engines {1, 2, 3, 4, Unknown}
large_aircraft {Yes, No, Unknown}
Airport
PK a_id int
airport_name varchar(255)
FK state_id int
State
PK s_id int
state_name varchar(255)
AirlineOperator
PK ao_id int
airline_name varchar(255)
Flight
PK f_id int
flight_date datetime
FK aircraft_id int
FK origin_airport_id int
FK airline_operator_id int
BirdStrike
PK birdstrike_id int
FK flight_id int
number_struck_est {1, 2 to 10, 11 to 100, 100+, Unknown}
number_struck_actual int
phase_of_flight {Approach, Climb, Descent, Landing Roll, Parked, Take-off run, Taxi, Unknown}
altitude_bin {< 1000 ft, > 1000 ft, Unknown}
impact_to_flight {Aborted Take-off, Engine Shut Down, None, Other, Precautionary Landing, Unknown}
indicated_damage boolean
wildlife_remains_collected boolean
wildlife_remains_smithsonian boolean
FK note_id int
FK wildlife_id int
wildlife_size {Small, Medium, Large, Unknown}
pilot_warned {True, False, Unknown}
total_cost int
feet_above_ground int
num_people_injured int
sky_conditions {No cloud, Overcast, Some cloud}
Precipitation
PK p_id int
precip_name {Fog, Rain, Snow, None}
FK birdstrike_id int
Note
PK n_id int
note text
Wildlife
PK w_id int
species_name varchar(255)

https://lucid.app/lucidchart/invitations/accept/inv_5008b57e-c7a0-403b-acf6-4c4c93856462?viewport_loc=-1270%2C-1079%2C2519%2C1721%2C2qcOW-XE2nwP

Load Libraries

```
library(RMySQL)
```

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

```
library(sqldf)
```

```
## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

##
## Attaching package: 'RSQLite'

## The following object is masked from 'package:RMySQL':
##
##     isIdCurrent

## sqldf will default to using MySQL
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
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(data.table)
```

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

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

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
library(ggplot2)
library(tinytex)

options(sqldf.driver = "SQLite")
```

Connecting to the AWS database

```
db_user <- "admin"
db_password <- "Boston1234"
db_name <-"practicum1db"
db_host <- "practicum1.c9h321ihmn93.us-east-2.rds.amazonaws.com"
db_port <- 3306

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

```
# View tables in the database
dbListTables(mydb)
```

```
##  [1] "Aircraft"        "AirlineOperator" "Airport"         "BirdStrike"
##  [5] "Flight"          "Note"            "Precipitation"   "State"
##  [9] "VW_master_table" "Wildlife"
```

```
-- Turns off foreign key check
SET FOREIGN_KEY_CHECKS = 0;
```

Creating the tables

CREATE TABLE: Aircraft

```
DROP TABLE IF EXISTS Aircraft;
```

```sql
CREATE TABLE Aircraft(
  a_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  aircraft_name VARCHAR(255) UNIQUE NOT NULL,      -- TODO: We added unique here
  aircraft_type ENUM('Airplane', 'UNKNOWN') NOT NULL,
  number_engines ENUM('1', '2', '3', '4', '5', '6', '7', '8', 'UNKNOWN') NOT NULL,
  large_aircraft ENUM('Yes', 'No', 'UNKNOWN') NOT NULL
  );

-- assumption: we are assuming that each model for the airplane has only one engine configuration so we
```

```sql
SELECT * FROM Aircraft;
```

Table 1: 0 records

a_id	aircraft_name	aircraft_type	number_engines	large_aircraft

CREATE TABLE: State

```sql
DROP TABLE IF EXISTS State;
```

```sql
CREATE TABLE State(
  s_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  state_name VARCHAR(255) UNIQUE NOT NULL
  );
```

```sql
SELECT * FROM State;
```

Table 2: 0 records

s_id	state_name

CREATE TABLE: Airport

```sql
DROP TABLE IF EXISTS Airport;
```

```sql
CREATE TABLE Airport(
  a_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  airport_name VARCHAR(255) UNIQUE NOT NULL,
  state_id INTEGER NOT NULL,
  CONSTRAINT state_id_fk FOREIGN KEY (state_id) REFERENCES State(s_id)
  );
```

```sql
SELECT * FROM Airport;
```

Table 3: 0 records

a_id	airport_name	state_id

CREATE TABLE: AirlineOperator

```sql
DROP TABLE IF EXISTS AirlineOperator;
```

```sql
CREATE TABLE AirlineOperator(
  ao_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  airline_name VARCHAR(255) UNIQUE NOT NULL
  );
```

```sql
SELECT * FROM AirlineOperator;
```

Table 4: 0 records

ao_id	airline_name

CREATE TABLE: Flight

```sql
DROP TABLE IF EXISTS Flight;
```

```sql
CREATE TABLE Flight(
  f_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  flight_date DATE NOT NULL,
  aircraft_id INTEGER NOT NULL,
  origin_airport_id INTEGER NOT NULL,
  airline_operator_id INTEGER NOT NULL,
  CONSTRAINT aircraft_id_fk FOREIGN KEY (aircraft_id) REFERENCES Aircraft(a_id),
  CONSTRAINT origin_airport_id_fk FOREIGN KEY (origin_airport_id) REFERENCES Airport(a_id),
  CONSTRAINT airline_operator_id_fk FOREIGN KEY (airline_operator_id) REFERENCES AirlineOperator(ao_id)
  );
```

```sql
SELECT * FROM Flight;
```

Table 5: 0 records

f_id	flight_date	aircraft_id	origin_airport_id	airline_operator_id

CREATE TABLE: Wildlife

```sql
DROP TABLE IF EXISTS Wildlife;
```

```sql
CREATE TABLE Wildlife(
  w_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  species_name VARCHAR(255) UNIQUE NOT NULL
  );
```

```sql
SELECT * FROM Wildlife;
```

Table 6: 0 records

w_id	species_name

CREATE TABLE: Note

```
DROP TABLE IF EXISTS Note;
```

```
CREATE TABLE Note(
  n_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  note TEXT NOT NULL
  );
```

```
SELECT * FROM Note;
```

Table 7: 0 records

n_id	note

CREATE TABLE: BirdStrike

```
DROP TABLE IF EXISTS BirdStrike;
```

```
CREATE TABLE BirdStrike(
  birdstrike_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  flight_id INTEGER NOT NULL,
  number_struck_est ENUM('1', '2 to 10', '11 to 100', '100+', 'UNKNOWN') NOT NULL,
  number_struck_actual INTEGER NOT NULL,
  phase_of_flight ENUM('Approach', 'Climb', 'Descent', 'Landing Roll',
                       'Parked', 'Take-off run', 'Taxi', 'UNKNOWN') NOT NULL,
  altitude_bin ENUM('< 1000 ft', '> 1000 ft', 'UNKNOWN') NOT NULL,
  impact_to_flight ENUM('Aborted Take-off', 'Engine Shut Down', 'None',
                        'Other', 'Precautionary Landing', 'UNKNOWN') NOT NULL,
  indicated_damage BOOLEAN NOT NULL,
  wildlife_remains_collected BOOLEAN NOT NULL,
  wildlife_remains_smithsonian BOOLEAN NOT NULL,
  note_id INTEGER NOT NULL,
  wildlife_id INTEGER NOT NULL,
  wildlife_size ENUM('Small', 'Medium', 'Large', 'UNKNOWN') NOT NULL,
  pilot_warned ENUM('True', 'False', 'UNKNOWN') NOT NULL,
  total_cost INTEGER NOT NULL,
  feet_above_ground INTEGER NOT NULL,
  num_people_injured INTEGER NOT NULL,
  sky_conditions ENUM('No cloud', 'Overcast', 'Some cloud') NOT NULL,
  CONSTRAINT flight_id_fk FOREIGN KEY (flight_id) REFERENCES Flight(f_id),
  CONSTRAINT note_fk FOREIGN KEY (note_id) REFERENCES Note(n_id),
  CONSTRAINT wildlife_id_fk FOREIGN KEY (wildlife_id) REFERENCES Wildlife(w_id)
  );
```

```
SELECT * FROM BirdStrike;
```

Table 8: 0 records

birdstrike_id	flight_id	number_struck_est	number_struck_actual	phase_of_flight	altitude_bin	impact_to_flight	indicated_damage	wildlife_remains_collected	wildlife_remains_smithsonian	note_id	wildlife_id	wildlife_size	pilot_warned	total_cost	feet_above_ground	num_people_injured	sky_conditions

CREATE TABLE: Precipitation

```
DROP TABLE IF EXISTS Precipitation;
```

```
CREATE TABLE Precipitation(
  p_id INTEGER PRIMARY KEY AUTO_INCREMENT,
  precip_name ENUM('Fog', 'Rain', 'Snow', 'None') NOT NULL,
  birdstrike_id INTEGER NOT NULL,
  CONSTRAINT birdstrike_id_fk FOREIGN KEY (birdstrike_id) REFERENCES BirdStrike(birdstrike_id)
  );
```

```
SELECT * FROM Precipitation;
```

Table 9: 0 records

p_id	precip_name	birdstrike_id

Cleaning the Data

Load CSV File

```
# Saved a copy of birdstrikes.csv to our git repo
file <- "BirdStrikesData.csv"

# to remove the blanks and update with UNKNOWN
birdStrike_df <- read.csv(file, header = TRUE, stringsAsFactors = FALSE, na.strings=c("", " "))
birdStrike_df[is.na(birdStrike_df)] <- 'UNKNOWN'

# make the row names a column for each row, which will be needed for loading data
setDT(birdStrike_df, keep.rownames = TRUE)[]
```

```
##            rn Record.ID Aircraft..Type                   Airport..Name
##     1:      1    202152       Airplane                    LAGUARDIA NY
##     2:      2    208159       Airplane     DALLAS/FORT WORTH INTL ARPT
##     3:      3    207601       Airplane               LAKEFRONT AIRPORT
##     4:      4    215953       Airplane            SEATTLE-TACOMA INTL
##     5:      5    219878       Airplane                    NORFOLK INTL
##    ---
## 25554: 25554    321151       Airplane               REDDING MUNICIPAL
## 25555: 25555    319677       Airplane                    ORLANDO INTL
## 25556: 25556    319680        UNKNOWN                         UNKNOWN
## 25557: 25557    319679       Airplane DETROIT METRO WAYNE COUNTY ARPT
## 25558: 25558    319593       Airplane     ABRAHAM LINCOLN CAPITAL ARPT
##         Altitude.bin Aircraft..Make.Model Wildlife..Number.struck
##     1:    > 1000 ft            B-737-400                Over 100
##     2:    < 1000 ft                MD-80                Over 100
##     3:    < 1000 ft                C-500                Over 100
##     4:    < 1000 ft            B-737-400                Over 100
##     5:    < 1000 ft          CL-RJ100/200                Over 100
##    ---
## 25554:    > 1000 ft              EMB-120                       1
## 25555:    < 1000 ft                A-321                       1
## 25556:      UNKNOWN               EC-135                 UNKNOWN
## 25557:    < 1000 ft           B-757-200                      1
## 25558:    < 1000 ft           B-737-400                      1
##        Wildlife..Number.Struck.Actual Effect..Impact.to.flight      FlightDate
##     1:                            859        Engine Shut Down 11/23/2000 0:00
##     2:                            424                    None  7/25/2001 0:00
##     3:                            261                    None  9/14/2001 0:00
##     4:                            806    Precautionary Landing   9/5/2002 0:00
##     5:                            942                    None  6/23/2003 0:00
##    ---
## 25554:                              1                    None 12/30/2011 0:00
## 25555:                              1                    None 12/30/2011 0:00
## 25556:                              1                 UNKNOWN         UNKNOWN
## 25557:                              1                    None 12/31/2011 0:00
## 25558:                              1                    None 12/31/2011 0:00
##        Effect..Indicated.Damage Aircraft..Number.of.engines.
##     1:           Caused damage                            2
##     2:           Caused damage                            2
##     3:               No damage                            2
##     4:               No damage                            2
##     5:               No damage                            2
##    ---
## 25554:               No damage                            2
## 25555:               No damage                            2
## 25556:               No damage                      UNKNOWN
## 25557:               No damage                            2
## 25558:           Caused damage                            2
##        Aircraft..Airline.Operator Origin.State When..Phase.of.flight
##     1:               US AIRWAYS*     New York                  Climb
##     2:         AMERICAN AIRLINES        Texas           Landing Roll
##     3:                  BUSINESS    Louisiana               Approach
##     4:           ALASKA AIRLINES   Washington                  Climb
##     5:           COMAIR AIRLINES     Virginia               Approach
##    ---
## 25554:          SKYWEST AIRLINES   California               Approach
## 25555:                US AIRWAYS      Florida           Landing Roll
## 25556:                   UNKNOWN     Virginia                UNKNOWN
## 25557:           DELTA AIR LINES     Michigan           Landing Roll
## 25558:              XTRA AIRWAYS     Illinois          Take-off run
##        Conditions..Precipitation Remains.of.wildlife.collected.
##     1:                      None                          FALSE
##     2:                      None                          FALSE
##     3:                      None                          FALSE
##     4:                      None                           TRUE
##     5:                      None                          FALSE
##    ---
## 25554:                       Fog                          FALSE
## 25555:                      None                          FALSE
## 25556:                      None                          FALSE
## 25557:                      None                          FALSE
## 25558:                      None                           TRUE
##        Remains.of.wildlife.sent.to.Smithsonian
##     1:                                   FALSE
##     2:                                   FALSE
##     3:                                   FALSE
##     4:                                 FALSE
##     5:                                 FALSE
##    ---
## 25554:                                 FALSE
## 25555:                                 FALSE
## 25556:                                 FALSE
## 25557:                                 FALSE
## 25558:                                 FALSE
##
##     1:  FLT 753. PILOT REPTD A HUNDRED BIRDS ON UNKN TYPE. #1 ENG WAS SHUT DOWN AND DIVERTED TO EWR.
##     2:
##     3:
##     4: NOTAM WARNING. 26 BIRDS HIT THE A/C, FORCING AN EMERGENCY LDG. 77 BIRDS WERE FOUND DEAD ON RWY
##     5:
##    ---
## 25554:
## 25555:
## 25556:
## 25557:
## 25558:
##        Wildlife..Size Conditions..Sky     Wildlife..Species
##     1:         Medium        No Cloud Unknown bird - medium
##     2:          Small      Some Cloud           Rock pigeon
##     3:          Small        No Cloud     European starling
##     4:          Small      Some Cloud     European starling
##     5:          Small        No Cloud     European starling
##    ---
## 25554:          Large        Overcast  Unknown bird - large
## 25555:          Small      Some Cloud          Tree swallow
## 25556:        UNKNOWN        No Cloud  Unknown bird - small
## 25557:         Medium      Some Cloud Unknown bird - medium
## 25558:         Medium        No Cloud       Red-tailed hawk
##        Pilot.warned.of.birds.or.wildlife. Cost..Total.. Feet.above.ground
##     1:                                  N        30,736             1,500
##     2:                                  Y             0                 0
##     3:                                  N             0                50
##     4:                                  Y             0                50
##     5:                                  N             0                50
##    ---
## 25554:                                  N             0             1,500
## 25555:                                  Y             0                 0
## 25556:                            UNKNOWN             0           UNKNOWN
## 25557:                                  Y             0                 0
## 25558:                                  N             0                 0
##        Number.of.people.injured Is.Aircraft.Large.
##     1:                        0                Yes
##     2:                        0                 No
##     3:                        0                 No
##     4:                        0                Yes
##     5:                        0                 No
##    ---
## 25554:                        0                 No
## 25555:                        0                 No
## 25556:                        0            UNKNOWN
## 25557:                      0                 Yes
## 25558:                      0                 Yes
```

```
names(birdStrike_df)[names(birdStrike_df) == "rn"] <- "UniqueKey"
head(birdStrike_df)
```

```
##    UniqueKey Record.ID Aircraft..Type               Airport..Name Altitude.bin
## 1:         1    202152       Airplane                LAGUARDIA NY    > 1000 ft
## 2:         2    208159       Airplane DALLAS/FORT WORTH INTL ARPT    < 1000 ft
## 3:         3    207601       Airplane           LAKEFRONT AIRPORT    < 1000 ft
## 4:         4    215953       Airplane         SEATTLE-TACOMA INTL    < 1000 ft
## 5:         5    219878       Airplane                NORFOLK INTL    < 1000 ft
## 6:         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
##    Effect..Impact.to.flight      FlightDate Effect..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
##    When..Phase.of.flight Conditions..Precipitation
## 1:                 Climb                      None
## 2:          Landing Roll                      None
## 3:              Approach                      None
## 4:                 Climb                      None
## 5:              Approach                      None
## 6:          Take-off run                      None
##    Remains.of.wildlife.collected. Remains.of.wildlife.sent.to.Smithsonian
## 1:                          FALSE                                   FALSE
## 2:                          FALSE                                   FALSE
## 3:                          FALSE                                   FALSE
## 4:                           TRUE                                   FALSE
## 5:                          FALSE                                   FALSE
## 6:                          FALSE                                   FALSE
##
## 1:  FLT 753. PILOT REPTD A HUNDRED BIRDS ON UNKN TYPE. #1 ENG WAS SHUT DOWN AND DIVERTED TO EWR. SLI
## 2:
## 3:
## 4: NOTAM WARNING. 26 BIRDS HIT THE A/C, FORCING AN EMERGENCY LDG. 77 BIRDS WERE FOUND DEAD ON RWY/TW
## 5:
## 6:
```

```
##    Wildlife..Size Conditions..Sky     Wildlife..Species
## 1:         Medium        No Cloud Unknown bird - medium
## 2:          Small      Some Cloud           Rock pigeon
## 3:          Small        No Cloud     European starling
## 4:          Small      Some Cloud     European starling
## 5:          Small        No Cloud     European starling
## 6:          Small        No Cloud  Unknown bird - small
##    Pilot.warned.of.birds.or.wildlife. Cost..Total.. Feet.above.ground
## 1:                                  N        30,736             1,500
## 2:                                  Y             0                 0
## 3:                                  N             0                50
## 4:                                  Y             0                50
## 5:                                  N             0                50
## 6:                                  N             0                 0
##    Number.of.people.injured Is.Aircraft.Large.
## 1:                        0                Yes
## 2:                        0                 No
## 3:                        0                 No
## 4:                        0                Yes
## 5:                        0                 No
## 6:                        0                 No
```

Parsing Date

```
# If needed, this is used to drop the parsed_date column
# birdStrike_df = subset(birdStrike_df, select = -c(flight_date) )
```

```
# If value was uploaded as UNKNOWN, set the date to 1/1/1776. Otherwise, parse the date as is.
for (row in 1:nrow(birdStrike_df)){
  date_time_string <- unlist(birdStrike_df[row, "FlightDate"])
  if (date_time_string == "UNKNOWN"){
    date_time_parsed <- as.Date(as.character(as.POSIXct("1/1/1776 0:00", format="%m/%d/%Y %H:%M")))
  } else {
    date_time_parsed <- as.Date(as.character(as.POSIXct(date_time_string, format="%m/%d/%Y %H:%M")))
  }
  birdStrike_df[row, "flight_date"] <- date_time_parsed
}

# Assumption:
#  - For unknown dates, it is set to 1776
```

```
# SQL command to confirm that data was correctly pulled
date_confirmation <- sqldf('SELECT "UniqueKey", "FlightDate"
                           , "flight_date"
                           FROM birdStrike_df')
head(date_confirmation)
```

```
##   UniqueKey      FlightDate flight_date
## 1         1 11/23/2000 0:00  2000-11-23
## 2         2  7/25/2001 0:00  2001-07-25
## 3         3  9/14/2001 0:00  2001-09-14
## 4         4   9/5/2002 0:00  2002-09-05
## 5         5  6/23/2003 0:00  2003-06-23
## 6         6  7/24/2003 0:00  2003-07-24
```

Cleaning Wildlife

```
# Copy over size where "Unknown bird - SIZE" was listed. Also standardize "UNKNOWN"
birdStrike_df$Wildlife..Size[birdStrike_df$Wildlife..Species == "Unknown bird - small"] <- "Small"
birdStrike_df$Wildlife..Size[birdStrike_df$Wildlife..Species == "Unknown bird - medium"] <- "Medium"
birdStrike_df$Wildlife..Size[birdStrike_df$Wildlife..Species == "Unknown bird - large"] <- "Large"

birdStrike_df$Wildlife..Species[birdStrike_df$Wildlife..Species == "Unknown bird - small"] <- "UNKNOWN"
birdStrike_df$Wildlife..Species[birdStrike_df$Wildlife..Species == "Unknown bird - medium"] <- "UNKNOWN
birdStrike_df$Wildlife..Species[birdStrike_df$Wildlife..Species == "Unknown bird - large"] <- "UNKNOWN"
birdStrike_df$Wildlife..Species[birdStrike_df$Wildlife..Species == "Unknown bird or bat"] <- "UNKNOWN"
```

Cleaning BirdStrike

```
# Change Y or N to True or False
birdStrike_df$Pilot.warned.of.birds.or.wildlife.[birdStrike_df$Pilot.warned.of.birds.or.wildlife. == "Y
birdStrike_df$Pilot.warned.of.birds.or.wildlife.[birdStrike_df$Pilot.warned.of.birds.or.wildlife. == "N
```

Uploading the Data

PREPARE DATA: AirlineOperator

```
AirlineOperatorTable <- sqldf('SELECT DISTINCT "Aircraft..Airline.Operator" AS airline_name
                              FROM birdStrike_df
                              ORDER BY airline_name')
head(AirlineOperatorTable)
```

```
##                   airline_name
## 1 ABSA AEROLINHAS BRASILEIRAS
## 2                      ABX AIR
## 3                 ACM AVIATION
## 4            ADI SHUTTLE GROUP
## 5                   AER LINGUS
## 6                     AERO AIR
```

```
# Upload data
dbWriteTable(mydb, "AirlineOperator", AirlineOperatorTable, row.names = FALSE, append = TRUE)
```

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

PREPARE DATA: State

```
# Retrieve Data
StateTable <- sqldf('SELECT DISTINCT "Origin.State" AS state_name
                    FROM birdStrike_df
                    ORDER BY state_name')
head(StateTable)
```

```
##         state_name
## 1          Alabama
## 2           Alaska
## 3          Alberta
## 4          Arizona
## 5         Arkansas
## 6 British Columbia
```

```
dbWriteTable(mydb, "State", StateTable, row.names = FALSE, append = TRUE)
```

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

PREPARE DATA: Airport

```
# Query State Data from Database
stateDataFromDatabase <- dbGetQuery(mydb, "SELECT * FROM State;")

# Join query from Database with query from birdStrike_df
stateDataTemp <- sqldf('SELECT DISTINCT "Airport..Name" AS airport_name
                       , s_id AS state_id
                       FROM birdStrike_df
                       JOIN stateDataFromDatabase
                       ON "Origin.State" = state_name')

# Use SQLDF to query only columns needed, labeled as needed
airportTable <- sqldf('SELECT DISTINCT airport_name
                     , state_id
                     FROM stateDataTemp
                     ORDER BY airport_name')
head(airportTable)
```

```
##                   airport_name state_id
## 1         ABERDEEN REGIONAL AR       51
## 2        ABILENE REGIONAL ARPT       53
## 3 ABRAHAM LINCOLN CAPITAL ARPT       16
## 4   ADAMS COUNTY- LEGION FIELD       60
## 5             ADAMS FIELD ARPT        5
## 6         ADDINGTON FIELD ARPT       20
```

```
dbWriteTable(mydb, "Airport", airportTable, row.names = FALSE, append = TRUE)
```

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

PREPARE DATA: Aircraft

```
# Query df to generate list of aircraft and numengines
AircraftNumEnginesOnlyDF <- sqldf('SELECT "Aircraft..Make.Model" AS Aircraft,
                                  "Aircraft..Number.of.engines." AS NumEngines
                                  FROM birdStrike_df
                                  ORDER BY Aircraft')

# Create summary table and then convert it back to dataframe
AircraftNumEnginesOnlyTable <- table(AircraftNumEnginesOnlyDF)

# Convert the table back to df
AircraftNumEnginesOnlyDF2 <- as.data.frame.matrix(AircraftNumEnginesOnlyTable)

# Find the max values for each column and add as new column
max_values <- colnames(AircraftNumEnginesOnlyDF2)[max.col(AircraftNumEnginesOnlyDF2, ties.method = "fir
AircraftNumEnginesOnlyDF2$most_common_engine <- max_values
```

```r
# Add airplane name as column, not row identifier
setDT(AircraftNumEnginesOnlyDF2, keep.rownames = TRUE)[]
```

```
##                   rn 1   2 3 4 C UNKNOWN most_common_engine
##   1:           A-10A 0   0 0 0 0       5            UNKNOWN
##   2: A-23 MUSKATEER 0   0 0 0 0       1            UNKNOWN
##   3:           A-300 0 327 0 1 0       0                  2
##   4:           A-310 0  28 0 0 0       0                  2
##   5:           A-318 0  84 0 0 0       0                  2
##  ---
## 347:            T-1A 0   0 0 0 0       1            UNKNOWN
## 348:            T-38 0   0 0 0 0       4            UNKNOWN
## 349:           T-38A 0   0 0 0 0      28            UNKNOWN
## 350:           T-38N 0   0 0 0 0       1            UNKNOWN
## 351:    VOLPARE BE18 0   1 0 0 0       0                  2
```

```r
# Query above table to retrieve needed format
AircraftTableEnginesOnly <- sqldf('SELECT rn AS name
                                  , most_common_engine AS number_engines
                                  FROM AircraftNumEnginesOnlyDF2')

# Query dataframe to pull in large_aircraft attribute for each aircraft
AircraftTableLargeAircraft <- sqldf('SELECT DISTINCT "Aircraft..Make.Model" AS name,
                                    "Aircraft..Type" AS aircraft_type,
                                    "Is.Aircraft.Large." AS large_aircraft
                                    FROM birdStrike_df
                                    GROUP BY name')

# Union df that has engine setup with df that has aircraft_type and large_aircraft
AircraftTable <- sqldf('SELECT rest.name AS aircraft_name
                       , rest.aircraft_type
                       , engines.number_engines
                       , rest.large_aircraft
                       FROM AircraftTableLargeAircraft AS rest
                       JOIN AircraftTableEnginesOnly AS engines
                       ON rest.name = engines.name
                       ORDER BY rest.name')
head(AircraftTable)
```

```
##    aircraft_name aircraft_type number_engines large_aircraft
## 1          A-10A      Airplane        UNKNOWN             No
## 2 A-23 MUSKATEER      Airplane        UNKNOWN             No
## 3          A-300      Airplane              2             No
## 4          A-310      Airplane              2             No
## 5          A-318      Airplane              2             No
## 6          A-319      Airplane              2             No
```

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

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

PREPARE DATA: Flight

```r
# Query information from database, needed to join everything together
AircraftFromAWS <- dbGetQuery(mydb, "SELECT * FROM Aircraft;")
```

```
AirportFromAWS <- dbGetQuery(mydb, "SELECT * FROM Airport;")
AirlineOperatorFromAWS <- dbGetQuery(mydb, "SELECT * FROM AirlineOperator;")
StateFromAWS <- dbGetQuery(mydb, "SELECT * FROM State;")
```

```
# Start by joining in Aircraft data
FlightTableWithAircraft <- sqldf('SELECT "UniqueKey" as f_id, "Record.ID" AS record_id, flight_date
                                  , aircraft.a_id AS aircraft_id, aircraft.aircraft_name AS aircraft_nam
                                  , "Airport..Name" AS airport_name
                                  , "Origin.State" AS origin_state
                                  , "Aircraft..Airline.Operator"
                                  FROM birdStrike_df
                                  JOIN AircraftFromAWS AS aircraft
                                  ON "Aircraft..Make.Model" = aircraft.aircraft_name')

# Then add in state
FlightTableWithAircraft_State <- sqldf('SELECT *
                                        FROM FlightTableWithAircraft
                                        JOIN StateFromAWS
                                        ON origin_state = state_name')


# Then add in airline operator
FlightTableWithAircraft_AirlineOperator <- sqldf('SELECT *
                                                  FROM FlightTableWithAircraft_State
                                                  JOIN AirlineOperatorFromAWS AS airlineOperator
                                                   ON "Aircraft..Airline.Operator" = airlineOperator.airli

# Then add in airport
FlightTable <- sqldf('SELECT f_id, flight_date, aircraft_id, a_id AS origin_airport_id, ao_id AS airline
                                         FROM FlightTableWithAircraft_AirlineOperator
                                         JOIN AirportFromAWS AS airport
                                         ON FlightTableWithAircraft_AirlineOperator.airport_name = airpo
head(FlightTable)
```

```
##   f_id flight_date aircraft_id origin_airport_id airline_operator_id
## 1    1  2000-11-23          40               531                 275
## 2    2  2001-07-25         278               209                  46
## 3    3  2001-09-14         130               538                  70
## 4    4  2002-09-05          40               912                  36
## 5    5  2003-06-23         153               715                 101
## 6    6  2003-07-24           3               396                  46
```

```
dbWriteTable(mydb, "Flight", FlightTable, row.names = FALSE, append = TRUE)
```

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

PREPARE DATA: Note

```
NoteTable <- sqldf('SELECT "UniqueKey" AS n_id, "Remarks" AS note
                       FROM birdStrike_df')
head(NoteTable)
```

```
##   n_id
## 1    1
## 2    2
## 3    3
## 4    4
## 5    5
## 6    6
##
## 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
```

```
dbWriteTable(mydb, "Note", NoteTable, row.names = FALSE, append = TRUE)
```

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

PREPARE DATA: Wildlife

```
WildlifeTable <- sqldf('SELECT DISTINCT "Wildlife..Species" AS species_name
                        FROM birdStrike_df')
head(WildlifeTable)
```

```
##         species_name
## 1            UNKNOWN
## 2        Rock pigeon
## 3  European starling
## 4       Canada goose
## 5         Snow goose
## 6 Black-headed munia
```

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

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

PREPARE DATA: BirdStrike

```
# Query database to get necessary information to join birdstrike table
WildlifeFromAWS <- dbGetQuery(mydb, "SELECT * FROM Wildlife;")
```

```
BirdStrikeTable <- sqldf('SELECT "Record.ID"as birdstrike_id
                         , "UniqueKey" AS flight_id
                         , "Wildlife..Number.struck" as number_struck_est
                         , "Wildlife..Number.Struck.Actual" as number_struck_actual
                         , "When..Phase.of.flight" as phase_of_flight
                         , "Altitude.bin" as altitude_bin
                         , "Effect..Impact.to.flight" as impact_to_flight
                         , "Effect..Indicated.Damage" as indicated_damage
                         , "Remains.of.wildlife.collected." as wildlife_remains_collected
                         , "Remains.of.wildlife.sent.to.Smithsonian" as wildlife_remains_smithsonian
                         , "Wildlife..Size" as wildlife_size
                         , "UniqueKey" as note_id
                         , wildlife.w_id as wildlife_id
                         , "Pilot.warned.of.birds.or.wildlife." as pilot_warned
                         , "Cost..Total.." as total_cost
                         , "Feet.above.ground" as feet_above_ground
```

```
                            , "Number.of.people.injured" as num_people_injured
                            , "Conditions..Sky" as sky_conditions
                            FROM birdStrike_df
                            JOIN WildlifeFromAWS AS wildlife
                            ON "Wildlife..Species" = wildlife.species_name')
head(BirdStrikeTable)
```

```
##   birdstrike_id flight_id number_struck_est number_struck_actual
## 1        202152         1          Over 100                  859
## 2        208159         2          Over 100                  424
## 3        207601         3          Over 100                  261
## 4        215953         4          Over 100                  806
## 5        219878         5          Over 100                  942
## 6        218432         6          Over 100                  537
##   phase_of_flight altitude_bin      impact_to_flight indicated_damage
## 1           Climb    > 1000 ft      Engine Shut Down    Caused damage
## 2    Landing Roll    < 1000 ft                  None    Caused damage
## 3        Approach    < 1000 ft                  None        No damage
## 4           Climb    < 1000 ft Precautionary Landing        No damage
## 5        Approach    < 1000 ft                  None        No damage
## 6    Take-off run    < 1000 ft                  None        No damage
##   wildlife_remains_collected wildlife_remains_smithsonian wildlife_size note_id
## 1                          0                            0        Medium       1
## 2                          0                            0         Small       2
## 3                          0                            0         Small       3
## 4                          1                            0         Small       4
## 5                          0                            0         Small       5
## 6                          0                            0         Small       6
##   wildlife_id pilot_warned total_cost feet_above_ground num_people_injured
## 1           1        False     30,736             1,500                  0
## 2           2         True          0                 0                  0
## 3           3        False          0                50                  0
## 4           3         True          0                50                  0
## 5           3        False          0                50                  0
## 6           1        False          0                 0                  0
##   sky_conditions
## 1       No Cloud
## 2     Some Cloud
## 3       No Cloud
## 4     Some Cloud
## 5       No Cloud
## 6       No Cloud
```

```
dbWriteTable(mydb, "BirdStrike", BirdStrikeTable, row.names = FALSE, append = TRUE)
```

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

PREPARE DATA: Precipitation

```
strike_and_precip <- sqldf('SELECT "Record.ID" AS birdstrike_id, "Conditions..Precipitation" AS precip_
                               FROM birdStrike_df')
head(strike_and_precip)
```

```
##   birdstrike_id precip_name
## 1        202152        None
## 2        208159        None
## 3        207601        None
## 4        215953        None
## 5        219878        None
## 6        218432        None
```

```r
# Create an empty preciptable
PrecipTable <- data.frame(matrix(ncol = 2, nrow = 0))
col_names <- c("birdstrike_id", "precip_name")
colnames(PrecipTable) <- col_names

# iterate through each row in the df to parse multi-valued atributes to unique columms
for (row in 1:nrow(strike_and_precip)){
  record_id <- strike_and_precip[row, "birdstrike_id"]
  precip_parsed <- unlist(strsplit(strike_and_precip[row, "precip_name"], "\\,\ "))
  for (precip in precip_parsed){
    new_row <- c(record_id, precip)
    PrecipTable <- rbind(new_row, PrecipTable)
  }
}

# add back column names
colnames(PrecipTable) <- col_names
head(PrecipTable)
```

```
##   birdstrike_id precip_name
## 1        319593        None
## 2        319679        None
## 3        319680        None
## 4        319677        None
## 5        321151         Fog
## 6        319672        None
```

```r
dbWriteTable(mydb, "Precipitation", PrecipTable, row.names = FALSE, append = TRUE)
```

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

```sql
-- Turns on foreign key check now that data is uploaded
SET FOREIGN_KEY_CHECKS = 1;
```

Practicum Questions

4. (10 pts / 1 hr) Create a SQL query against your database to find the number of bird strike incidents for each airline upon take-off or climb. Include all airlines. You may either use a {sql} code chunk or an R function to execute the query.

```sql
SELECT COUNT(BirdStrike.birdstrike_id) AS incidents, airline_name
FROM BirdStrike
JOIN Flight
ON BirdStrike.flight_id = Flight.f_id
JOIN AirlineOperator
ON airline_operator_id = ao_id
WHERE phase_of_flight IN ('Take-off run', 'Climb') AND airline_name <> "UNKNOWN"
GROUP BY airline_name
```

```
ORDER BY incidents DESC
```

Table 10: Displaying records 1 - 10

incidents	airline_name
1544	SOUTHWEST AIRLINES
1287	BUSINESS
771	AMERICAN AIRLINES
517	DELTA AIR LINES
343	US AIRWAYS*
324	AMERICAN EAGLE AIRLINES
282	SKYWEST AIRLINES
240	JETBLUE AIRWAYS
232	US AIRWAYS
192	UNITED AIRLINES

5. (10 pts / 1 hr) Create a SQL query against your database to find the airports that had the most bird strike incidents (during any flight phase). Include all airlines. You may either use a {sql} code chunk or an R function to execute the query.

```
SELECT COUNT(BirdStrike.birdstrike_id) AS incidents
, airport_name
FROM BirdStrike
JOIN Flight
ON BirdStrike.flight_id = Flight.f_id
JOIN Airport
ON origin_airport_id = a_id
GROUP BY airport_name
ORDER BY incidents DESC
```

Table 11: Displaying records 1 - 10

incidents	airport_name
803	DALLAS/FORT WORTH INTL ARPT
676	SACRAMENTO INTL
479	SALT LAKE CITY INTL
476	DENVER INTL AIRPORT
452	KANSAS CITY INTL
442	PHILADELPHIA INTL
408	ORLANDO INTL
401	BALTIMORE WASH INTL
395	LOUISVILLE INTL ARPT
390	JOHN F KENNEDY INTL

6. (10 pts / 1 hr) Create a SQL query against your database to find the number of bird strike incidents by year. Include all airlines. You may either use a {sql} code chunk or an R function to execute the query.

```
SELECT COUNT(BirdStrike.birdstrike_id) AS incidents
, YEAR(Flight.flight_date) AS Year
FROM BirdStrike
```

```
JOIN Flight
ON BirdStrike.flight_id = Flight.f_id
WHERE YEAR(Flight.flight_date) <> "1776"
GROUP BY YEAR(Flight.flight_date)
ORDER BY Year
```

Table 12: Displaying records 1 - 10

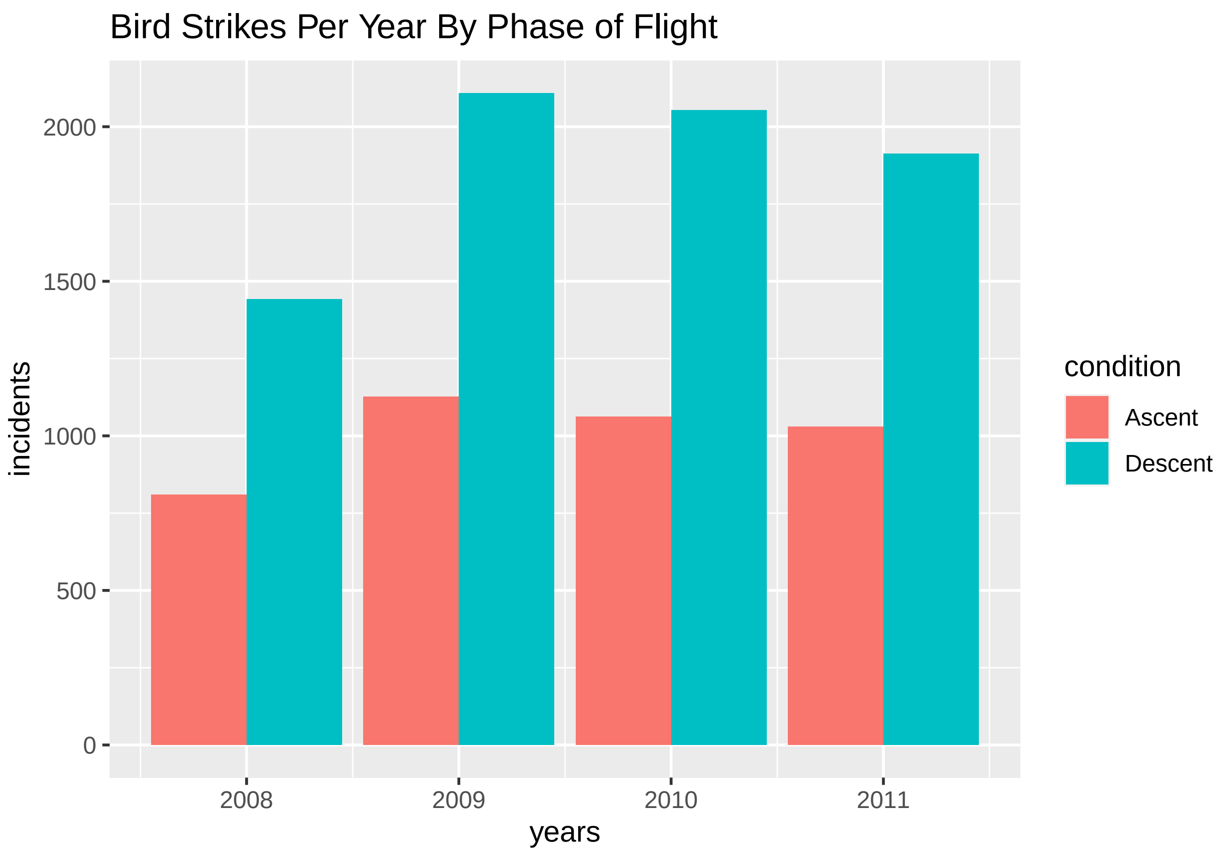
incidents	Year
1367	2000
1230	2001
1681	2002
1568	2003
1692	2004
1853	2005
2159	2006
2301	2007
2258	2008
3247	2009

7. (10 pts / 3 hrs) Using the above data, build a column chart that visualizes the number of bird strikes incidents per year from 2008 to 2011 during take-off/climbing and during descent/approach/landing. Adorn the graph with appropriate axis labels.

```
# pulling the data together into a data frame
BirdstrikesPerYear <- dbGetQuery(mydb,
                            'SELECT * FROM
                            (SELECT COUNT(BirdStrike.birdstrike_id) as incidents
                            , CASE
                              WHEN phase_of_flight = "Take-off run" THEN "Ascent"
                              WHEN phase_of_flight = "Climb" THEN "Ascent"
                              WHEN phase_of_flight = "Descent" THEN "Descent"
                              WHEN phase_of_flight = "Approach" THEN "Descent"
                              WHEN phase_of_flight = "Landing Roll" THEN "Descent"
                            END as phase
                            , YEAR(Flight.flight_date) as Year
                            FROM BirdStrike
                            JOIN Flight
                            ON BirdStrike.flight_id = Flight.f_id
                            WHERE YEAR(Flight.flight_date) >= 2008
                            and YEAR(Flight.flight_date) <= 2011
                            GROUP BY YEAR(Flight.flight_date), phase
                            ORDER BY Year) t
                            WHERE phase = ("Ascent" OR "Descent")')

#initializing the data
condition <- BirdstrikesPerYear$phase
years <- BirdstrikesPerYear$Year
incidents <- BirdstrikesPerYear$incidents

# adorning the bar chart
barChart <- ggplot(data = BirdstrikesPerYear, aes(fille = condition, x = years, y = incidents, fill = co
print(barChart)
```



8. (10 pts / 3 hrs) Create a stored procedure in MySQL (note that if you used SQLite, then you cannot complete this step) that removes a bird strike incident from the database. You may decide what you need to pass to the stored procedure to remove a bird strike incident, e.g., departure airport, airlines, or some ID. Show that the deletion worked as expected.

```
DROP PROCEDURE IF EXISTS Remove_BirdStrike;
```

```
CREATE PROCEDURE Remove_BirdStrike (
  IN birdstrike_id_to_delete INTEGER)
BEGIN
  -- Save the note_id, which will be needed for deletion later
  DECLARE delete_note_id INTEGER;
  SET delete_note_id = (SELECT BirdStrike.note_id
                        FROM BirdStrike
                        WHERE BirdStrike.birdstrike_id = birdstrike_id_to_delete);

  -- Remove the associated precipitation records
  DELETE FROM Precipitation WHERE Precipitation.birdstrike_id = birdstrike_id_to_delete;

  -- Remove the birdstrike record
  DELETE FROM BirdStrike WHERE BirdStrike.birdstrike_id = birdstrike_id_to_delete;

  -- Remove the note record
  DELETE FROM Note WHERE Note.n_id = delete_note_id;
```

```sql
    -- Remove the corresponding flight
  DELETE FROM Flight WHERE Flight.f_id = (SELECT BirdStrike.flight_id
                                          FROM BirdStrike
                                          WHERE BirdStrike.birdstrike_id = birdstrike_id_to_delete);

END;
```

```sql
-- Selecting record based on birdstrike_id
SELECT * FROM BirdStrike WHERE birdstrike_id = 315417;
```

Table 13: 1 records

birdstrike_id	flight_id	number_struck_est	number_struck_actual	phase_of_flight	altitude_bin	impact_to_flight	indicated_damage	wildlife_remains_collected	wildlife_remains_smithsonian	wildlife_id	note_id	wildlife_size	pilot_warned	total_cost	feet_above_ground	num_people_injured	sky_conditions
315417	243571		1	Take-off run	< 1000 ft	None	0	1	0	243572	2	Small	False	0	0	0	No cloud

```sql
-- Executing stored procedure on above birdstrike_id to remove record
CALL Remove_BirdStrike(315417)
```

```sql
-- Attempting to select record based on birdstrike_id. If stored procedure is successful, no record is
SELECT * FROM BirdStrike WHERE birdstrike_id = 315417;
```

Table 14: 0 records

birdstrike_id	flight_id	number_struck_est	number_struck_actual	phase_of_flight	altitude_bin	impact_to_flight	indicated_damage	wildlife_remains_collected	wildlife_remains_smithsonian	wildlife_id	note_id	wildlife_size	pilot_warned	total_cost	feet_above_ground	num_people_injured	sky_conditions

Create View

Recreate CSV file that was provided at beginning of assignment

```sql
DROP VIEW IF EXISTS VW_master_table
```

```sql
CREATE VIEW VW_master_table AS
SELECT BirdStrike.birdstrike_id, Aircraft.aircraft_type, Airport.airport_name
       , BirdStrike.altitude_bin, Aircraft.aircraft_name
       , BirdStrike.number_struck_est, BirdStrike.number_struck_actual,BirdStrike.impact_to_flight
       , Flight.flight_date, BirdStrike.indicated_damage
       , Aircraft.number_engines, AirlineOperator.airline_name
       , State.state_name, BirdStrike.phase_of_flight, Precipitation.precip_name
       , BirdStrike.wildlife_remains_collected
       , BirdStrike.wildlife_remains_smithsonian, Note.note
       , BirdStrike.wildlife_size, BirdStrike.sky_conditions
       , Wildlife.species_name, BirdStrike.pilot_warned, BirdStrike.total_cost
       , BirdStrike.feet_above_ground, BirdStrike.num_people_injured
       , Aircraft.large_aircraft
FROM BirdStrike
     JOIN Note
     ON BirdStrike.note_id = Note.n_id
     JOIN Wildlife
     ON BirdStrike.wildlife_id = Wildlife.w_id
```

```sql
    JOIN Precipitation
    ON BirdStrike.birdstrike_id = Precipitation.birdstrike_id
    JOIN Flight
    ON BirdStrike.flight_id = Flight.f_id
    JOIN Aircraft
    ON Flight.aircraft_id = Aircraft.a_id
    JOIN Airport
    ON Flight.origin_airport_id = Airport.a_id
    JOIN State
    ON Airport.state_id = State.s_id
    JOIN AirlineOperator
    ON Flight.airline_operator_id = AirlineOperator.ao_id
    ORDER BY birdstrike_id DESC;
```

```sql
SELECT * FROM VW_master_table;
```

Table 15: Displaying records 1 - 10

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
321909	Airplane	NEW ORLEANS INTL	< 1000 ft	B-717-200	1	1	None	2010-11-09	0	2	AIRTRAN AIRWAYS	Louisiana	Landing Roll	None	1	0	TWR ADZ O2 OF BIRD-STRIKE REPTD ON RWY 1/19 ABOUT 1000' REMAINIGN RWY 1. RWY SWEEP DID NOT FIND REMAINS. O2 REPT TO THE GATE WHERE THE A/C WAS AND SAW BIRD REMAINS STUCK TO WINDSHLD. UNABLE TO ID THE BIRD. NO DMG.	Small	Some cloud	UNKNOWN	True	0	0	No

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
321291	Airplane	SEATTLE-TACOMA INTL	< 1000 ft	A-320	2 to 10	10	None	2010-08-03	0	2	JETBLUE AIRWAYS	Washington	Approach	None	1	1	ID BY SMITHSONIAN, FAA 7439. TIME OUT OF SERVICE 1/2 HR.	Small	Some cloud	Barn swallow	False	0	200	0	No
321172	Airplane	CHARLOTTE/DOUGLAS INTL ARPT	< 1000 ft	DHC-8 DASH 8	2 to 10	10	Aborted Take-off	2010-11-23	0	2	PIEDMONT AIRLINES	North Carolina	Take-off run	None	1	0	A/C RETD TO GATE FOR INSPN. OPS FOUND 8 ROCK PIGEON CARCASSES ON RWY. REMAINS DISCARDED BEFORE SAMPLES COULD BE MADE.	Small	No cloud	Rock pigeon	True	0	0	0	No
321159	Airplane	CHARLOTTE/DOUGLAS INTL ARPT	< 1000 ft	CL-RJ100/200		1	None	2010-11-25	0	2	PSA AIRLINES	North Carolina	Approach	None	0	0	NO DMG REPTD.	Small	No cloud	Rock pigeon	True	0	200	0	No
321151	Airplane	REDDING MUNICIPAL	> 1000 ft	EMB-120		1	None	2010-12-30	0	2	SKYWEST AIRLINES	California	Approach	Fog	0	0	DUCK? NO DMG REPTD.	Large	Overcast	UNKNOWN	False	0	1	0	No
320702	Airplane	GREENVILLE DOWNTOWN ARPT	< 1000 ft	C-560	2 to 10	4	None	2010-10-19	0	2	BUSINESS	South Carolina	Climb	None	0	0	NO DMG NOTED. BITS OF BLOOD/FEATHERS FOUND ON LEADING EDGE AND LDG GEAR DOOR. (DATA ENTRY NOTE: FORM ARRIVED MAR 2012)	Small	Some cloud	UNKNOWN	False	0	20	0	No

bird[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]injured
320701	Airplane	FORT SMITH MUNICIPAL ARPT	< 1000 ft	CL-RJ100/200	1	1	None	2010-10-13	2	PINNACLE	Arkansas	Take-off run	None	0	0	PILOT REPTD NO DMG. NO REMAINS FOUND. (DATA ENTRY NOTE: FORM ARRIVED MAR 2012)	Small	No cloud	UNKNOWN	False	[illegible]	0	0	No
320296	Airplane	RALEIGH EXEC JETPORT AT SANFORD-LEE CNTY ARPT	< 1000 ft	BE-33	1	1	None	2010-11-13	1	BUSINESS	North Carolina	Climb	None	0	0	RT WING DAMAGED. SMALL DENT AND PAINT REMOVED.	Medium	No cloud	UNKNOWN	False	[illegible]	300	0	No
320200	Airplane	CHARLOTTE/DOUGLAS INTL ARPT	< 1000 ft	CL-RJ100/200	2 to 10	5	Aborted Take-off	2010-11-11	2	PSA AIRLINES	North Carolina	Take-off run	None	0	0	ATIS WARNING. A/C ABORTED T/O AND RETD TO GATE FOR INSPN. NO DMG REPTD.	Small	No cloud	Rock pigeon	True	0	0	0	No
320199	Airplane	CHARLOTTE/DOUGLAS INTL ARPT	> 1000 ft	CL-RJ100/200	1	1	None	2010-11-10	2	PINNACLE	North Carolina	Climb	None	0	0	UNKNOWN	Large	No cloud	UNKNOWN	True	[illegible]	3	0	No

Disconnect

```
dbDisconnect(mydb)
```

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