Practicum I CS5200

Summer Full 2023

Connect to Database

```
library(RMySQL)

host <- "sql9.freemysqlhosting.net"
username <- "sql9629158"
password <- "bXbHzkNEwi"
dbname <- "sql9629158"

dbcon <- dbConnect(
  MySQL(),
  user = username,
  password = password,
  dbname = dbname,
  host = host
)
```

Create Database

```
CREATE TABLE IF NOT EXISTS airports (
  aid INT PRIMARY KEY AUTO_INCREMENT,
  airportState TEXT,
  airportCode TEXT
);
```

```
CREATE TABLE IF NOT EXISTS flights (
  fid INT PRIMARY KEY,
  date DATE,
  origin INT,
  airline TEXT,
  aircraft TEXT,
  altitude INT,
  heavy BOOLEAN,
  FOREIGN KEY (origin) REFERENCES airports(aid)
);
```

```
CREATE TABLE IF NOT EXISTS conditions (
  cid INT PRIMARY KEY,
  sky_condition TEXT,
  explanation TEXT
);
```

```
CREATE TABLE IF NOT EXISTS strikes (
  sid INT PRIMARY KEY AUTO_INCREMENT,
  fid INT,
  numbirds INT,
  impact TEXT,
  damage BOOLEAN,
  altitude INT CHECK (altitude >= 0),
  conditions INT,
  FOREIGN KEY (conditions) REFERENCES conditions(cid),
  FOREIGN KEY (fid) REFERENCES flights(fid)
);
```

Manually Test Inserting Into Tables

```
INSERT INTO airports (airportState, airportCode)
VALUES ('California', 'LAX');
```

```
SELECT * FROM airports;
```

```
INSERT INTO flights (fid, date, origin, airline, aircraft, altitude, heavy)
VALUES (1, '2023-06-21', 1, 'Delta', 'Boeing 737', 35000, 1);
```

```
SELECT * FROM flights;
```

```
INSERT INTO conditions (cid, sky_condition, explanation)
VALUES (1, 'Clear', 'No clouds in the sky.');
```

```
SELECT * FROM conditions;
```

```
INSERT INTO strikes (fid, numbirds, impact, damage, altitude, conditions)
VALUES (1, 2, 'Engine failure', true, 30000, 1);
```

```
SELECT * FROM strikes;
```

Clean Data in Preparation for

```
DELETE FROM strikes;
```

```
DELETE FROM flights;
```

```
DELETE FROM airports;
```

```
DELETE FROM conditions;
```

```
ALTER TABLE airports AUTO_INCREMENT = 1;
```

```
ALTER TABLE strikes AUTO_INCREMENT = 1;
```

Create Dataframe

```
bds.raw <- read.csv('BirdStrikesData.csv')

csvData <- head(bds.raw, 100)
```

```
for (row in 1:nrow(datasubset)) {
  # airports table information
  airportState <- csvData[row, "origin"]

  # flights table information
  rid <- csvData[row, "rid"]
  date_string <- csvData[row, "flight_date"]
  date_object <- strptime(date_string, format = "%m/%d/%Y %H:%M")
  date <- format(date_object, format = "%Y-%m-%d")
  origin <- row
  airline <- csvData[row, "airline"]
  if (airline == "") {
    airline <- "unknown"
  }
  aircraft <- csvData[row, "aircraft"]
  altitude <- as.integer(gsub(",", "", csvData[row, "altitude_ft"]))
  damage <- csvData[row, "damage"]
  heavy <- 0
  if (damage == "Caused damage") {
    heavy <- 1
  }

  #strikes table information
  numbirds <- as.integer(csvData[row, "wildlife_struck"])
  impact <- csvData[row, "impact"]

  # conditions table information
  sky_condition <- csvData[row, "sky_conditions"]

  # Insert row information into airports table
  airports_insert_query <- sprintf("INSERT INTO airports (airportState) VALUES ('%s')", airportState)
  dbExecute(dbcon, airports_insert_query)

  # Insert row information into flights table
  flights_insert_query <- sprintf("INSERT INTO flights (fid, date, origin, airline, aircraft, altitude,
                                  VALUES (%d, DATE('%s'), %d, '%s', '%s', %d, %d)", rid, date, origin, a
  dbExecute(dbcon, flights_insert_query)

  # Insert row information into conditions table
  conditions_insert_query <- sprintf("INSERT INTO conditions (cid, sky_condition) VALUES (%d, '%s')", r
  dbExecute(dbcon, conditions_insert_query)
  # Insert row information into strikes table
  strikes_insert_query <- sprintf("INSERT INTO strikes (fid, numbirds, impact, damage, altitude, conditi
                                  VALUES (%d, %d, '%s', %d, %d, %d)", rid, numbirds, impact, heavy, alti
  dbExecute(dbcon, strikes_insert_query)
}
```

```
SELECT * FROM strikes;
```

Table 1: Displaying records 1 - 10

sid	fid	numbirds	impact	damage	altitude	conditions
1	202152	859	Engine Shut Down	1	1500	202152
2	208159	424	None	1	0	208159
3	207601	261	None	0	50	207601
4	215953	806	Precautionary Landing	0	50	215953
5	219878	942	None	0	50	219878
6	218432	537	None	0	0	218432
7	221697	227	Other	1	150	221697
8	236635	320	Other	1	100	236635
9	207369	9	Aborted Take-off	0	0	207369
10	204371	4	None	0	0	204371

```
SELECT * FROM flights;
```

Table 2: Displaying records 1 - 10

fid	date	origin	airline	aircraft	altitude	heavy
4064	2000-04-06	43	MILITARY	Airplane	1000	0
7379	2000-05-18	62	MILITARY	Airplane	1200	0
200031	2000-06-12	77	GREAT LAKES AIRLINES	Airplane	500	0
200068	2000-05-09	58	UPS AIRLINES	Airplane	75	0
200108	2000-07-10	94	UNITED AIRLINES	Airplane	0	0
200130	2000-03-23	37	NORTHWEST AIRLINES	Airplane	800	1
200155	2000-04-25	50	BUSINESS	Airplane	0	0
200219	2000-01-30	14	EXECUTIVE JET AVIATION	Airplane	1800	0
200220	2000-01-30	13	BUSINESS	Airplane	0	0
200247	2000-03-11	29	BUSINESS	Airplane	10	0

```
SELECT * FROM airports;
```

Table 3: Displaying records 1 - 10

aid	airportState	airportCode
1	New York	NA
2	Texas	NA
3	Louisiana	NA

aid	airportState	airportCode
4	Washington	NA
5	Virginia	NA
6	N/A	NA
7	Delaware	NA
8	DC	NA
9	Georgia	NA
10	Florida	NA

```
SELECT * FROM conditions;
```

Table 4: Displaying records 1 - 10

cid	sky_condition	explanation
4064	No Cloud	NA
7379	No Cloud	NA
200031	No Cloud	NA
200068	Overcast	NA
200108	No Cloud	NA
200130	Overcast	NA
200155	Some Cloud	NA
200219	Overcast	NA
200220	Overcast	NA
200247	No Cloud	NA

Query For Question 8

```
SELECT airportState AS state, COUNT(*) AS incidents
FROM strikes
JOIN flights ON strikes.fid = flights.fid
JOIN airports ON flights.origin = airports.aid
GROUP BY airportState
ORDER BY incidents DESC
LIMIT 10;
```

Table 5: Displaying records 1 - 10

state	incidents
California	9
Florida	8
DC	5
Georgia	5
Louisiana	4
Ohio	4
New York	4
Michigan	4
Illinois	4

state	incidents
N/A	4

Query For Question 9

```
SELECT airline, COUNT(*) AS incidents
FROM strikes
JOIN flights ON strikes.fid = flights.fid
JOIN airports ON flights.origin = airports.aid
GROUP BY airline
HAVING incidents > (SELECT AVG(incident_count) FROM (SELECT COUNT(*) AS incident_count FROM strikes JOI
ORDER BY incidents DESC;
```

Table 6: Displaying records 1 - 10

airline	incidents
BUSINESS	12
US AIRWAYS*	9
DELTA AIR LINES	8
UNITED AIRLINES	7
AMERICAN AIRLINES	6
NORTHWEST AIRLINES	4
COMAIR AIRLINES	4
AMERICA WEST AIRLINES	4
CONTINENTAL AIRLINES	3
AIRTRAN AIRWAYS	3

Query For Question 10

```
library(DBI)

query <- "SELECT DATE_FORMAT(date, '%Y-%m') AS month, SUM(numbirds) AS total_birds
  FROM strikes
  JOIN flights ON strikes.fid = flights.fid
  GROUP BY month
  ORDER BY month;"

result <- dbGetQuery(dbcon, query)

df <- as.data.frame(result)

head(df, 6)
```

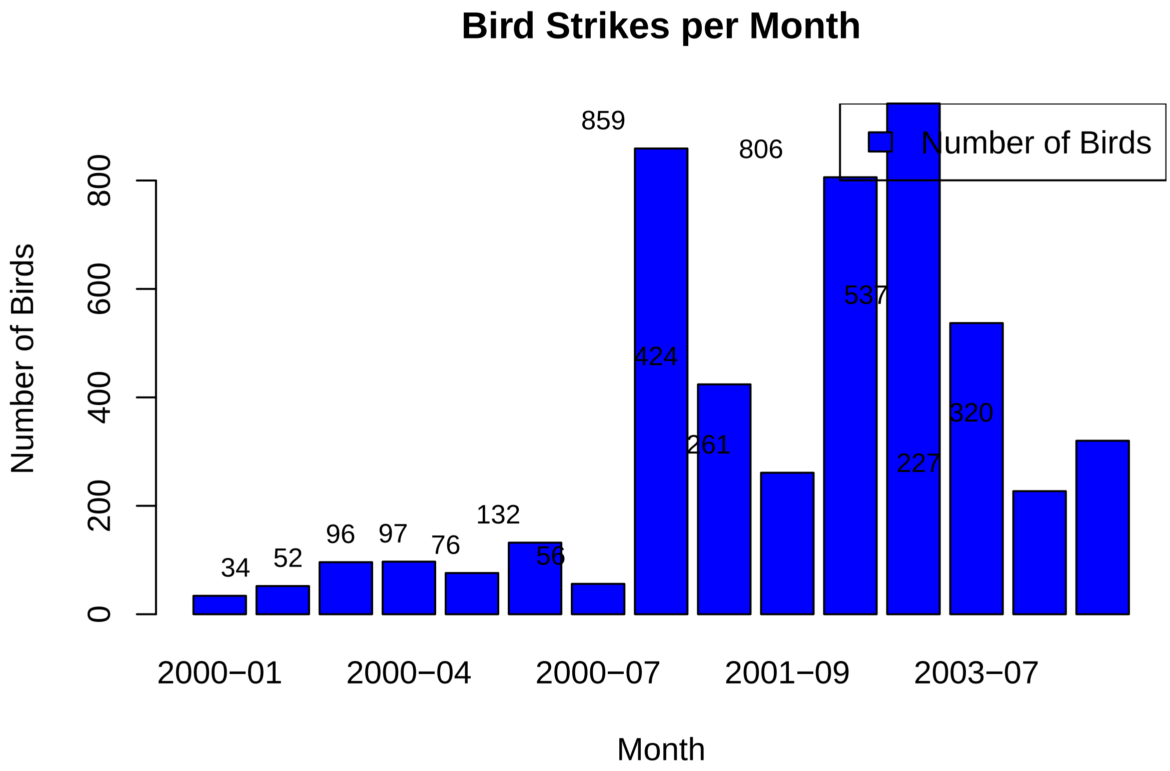
```
##     month total_birds
## 1 2000-01          34
## 2 2000-02          52
## 3 2000-03          96
## 4 2000-04          97
## 5 2000-05          76
## 6 2000-06         132
```

Data Graph For Question 11

```
# Create a column chart
barplot(df$total_birds, names.arg = df$month, xlab = "Month", ylab = "Number of Birds",
        main = "Bird Strikes per Month", col = "blue", ylim = c(0, max(df$total_birds)))

# Add data labels
text(x = 1:length(df$total_birds), y = df$total_birds, labels = df$total_birds, pos = 3, cex = 0.8)

# Add a legend
legend("topright", legend = c("Number of Birds"), fill = "blue")
```



```
# Adjust the plot margins
par(mar = c(5, 4, 4, 2) + 0.1)
```

Stored Procedure for Question 12

```
DROP PROCEDURE IF EXISTS AddNewStrike;
```

```
CREATE PROCEDURE AddNewStrike(
  IN fid INT,
  IN numbirds INT,
  IN impact TEXT,
  IN damage BOOLEAN,
  IN altitude INT,
  IN cid INT
)
BEGIN

  INSERT INTO strikes (fid, numbirds, impact, damage, altitude, conditions)
  VALUES (fid, numbirds, impact, damage, altitude, cid);
END;
```

```
query <- "CALL AddNewStrike(
  4064,
  10,
  'wing',
  0,
  5000,
  4064
)"

dbGetQuery(dbcon, query)
```

```
## data frame with 0 columns and 0 rows
```

```
check <- "SELECT *
FROM strikes
ORDER BY sid DESC
LIMIT 1;"

result <- dbGetQuery(dbcon, check)

print(result)
```

```
##   sid  fid numbirds impact damage altitude conditions
## 1 101 4064       10   wing      0     5000       4064
```

Disconnect From Database

```
dbDisconnect(dbcon)
```

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