Practicum I CS5200

Yawen Chi

Fall 2023

```
library(RMySQL)
```

```
##      DBI
```

Connect to Database

```
db_name_fh <- "sql12655763"
db_user_fh <- "sql12655763"
db_host_fh <- "sql12.freemysqlhosting.net"
db_pwd_fh <- "RyRdNuV3il"
db_port_fh <- 3306

# 3. Connect to remote server database
mydb.fh <-  dbConnect(RMySQL::MySQL(), user = db_user_fh, password = db_pwd_fh,
                      dbname = db_name_fh, host = db_host_fh, port = db_port_fh)

mydb <- mydb.fh
```

Create Database

```
drop table if exists strikes
```

```
drop table if exists conditions
```

```
drop table if exists flights
```

```
drop table if exists airports
```

Create airports table

```
CREATE TABLE airports (
  aid INTEGER AUTO_INCREMENT PRIMARY KEY,
  airportName TEXT,
  airportState TEXT(225),
  airportCode TEXT NULL
);
```

Create flights table

```
CREATE TABLE flights (
  fid INTEGER NOT NULL AUTO_INCREMENT,
  date DATE,
  origin INTEGER,
  airline TEXT,
  aircraft TEXT,
  altitude NUMERIC CHECK (altitude >= 0),
  heavy BOOLEAN NULL,
  PRIMARY KEY (fid),
  FOREIGN KEY (origin) REFERENCES airports(aid)
);
```

Create conditions table

```
CREATE TABLE conditions (
  cid INTEGER AUTO_INCREMENT PRIMARY KEY,
  sky_condition TEXT NOT NULL,
  explanation TEXT
);
```

Create strikes table

```
CREATE TABLE strikes (
  sid INTEGER AUTO_INCREMENT PRIMARY KEY,
  fid INTEGER,
  numbirds INTEGER,
  impact TEXT,
  damage BOOLEAN,
  altitude INTEGER CHECK (altitude >= 0),
  conditions INTEGER,
  FOREIGN KEY (conditions) REFERENCES conditions(cid),
  FOREIGN KEY (fid) REFERENCES flights(fid)
);
```

For checking only not evaluated

```
insert into airports(aid, airportName, airportState, airportCode)
values(1, "SEA-TAC International airport", "Washington", "SEA")
```

```
insert into flights(fid, date, origin, airline, aircraft, altitude, heavy)
values(2, "2022-10-10", 1, "EVA", "Airplane", 9999, TRUE)
```

```
insert into conditions(cid, sky_condition, explanation)
values(1, "No Cloud", "FLT 753. PILOT REPTD A HUNDRED BIRDS ON UNKN TYPE. #1 ENG WAS SHUT DOWN AND DIVE
```

```
insert into strikes(sid, fid, numbirds, impact, damage, altitude, conditions)
values(1, 2, 2, "Engine Shut Down", TRUE, 9999, 1)
```

```
select * from flights
```

```
select * from airports
```

```
select * from conditions
```

```
select * from strikes
```

Load the CSV file into a DataFrame

```
bds.raw <- read.csv("BirdStrikesData-V2.csv")
head(bds.raw)
```

```
##      rid aircraft                     airport        model wildlife_struck
## 1 202152 Airplane                LAGUARDIA NY    B-737-400             859
## 2 208159 Airplane DALLAS/FORT WORTH INTL ARPT        MD-80             424
## 3 207601 Airplane           LAKEFRONT AIRPORT        C-500             261
## 4 215953 Airplane         SEATTLE-TACOMA INTL    B-737-400             806
## 5 219878 Airplane                NORFOLK INTL CL-RJ100/200             942
## 6 218432 Airplane         GUAYAQUIL/S BOLIVAR        A-300             537
##                   impact     flight_date        damage           airline
## 1      Engine Shut Down 11/23/2000 0:00 Caused damage       US AIRWAYS*
## 2                  None  7/25/2001 0:00 Caused damage AMERICAN AIRLINES
## 3                  None  9/14/2001 0:00     No damage          BUSINESS
## 4 Precautionary Landing   9/5/2002 0:00     No damage   ALASKA AIRLINES
## 5                  None  6/23/2003 0:00     No damage   COMAIR AIRLINES
## 6                  None  7/24/2003 0:00     No damage AMERICAN AIRLINES
##       origin flight_phase remains_collected_flag
## 1   New York        Climb                  FALSE
## 2      Texas Landing Roll                  FALSE
## 3  Louisiana     Approach                  FALSE
## 4 Washington        Climb                   TRUE
## 5   Virginia     Approach                  FALSE
## 6        N/A Take-off run                  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
##   wildlife_size sky_conditions               species pilot_warned_flag
## 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
##   altitude_ft heavy_flag
## 1       1,500        Yes
## 2           0         No
## 3          50         No
## 4          50        Yes
## 5          50         No
## 6           0         No
```

Create data frame df.airports

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

df.airports <- sqldf::sqldf("select distinct 1 as aid, airport as airportName, origin as airportState, 
```

```
##
##    'RSQLite'
```

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

```
n.airports <- nrow(df.airports)
df.airports[,1] <- seq(1, n.airports)


head(df.airports)
```

```
##   aid                 airportName airportState airportCode
## 1   1                 LAGUARDIA NY     New York          NA
## 2   2 DALLAS/FORT WORTH INTL ARPT        Texas          NA
## 3   3            LAKEFRONT AIRPORT    Louisiana          NA
## 4   4          SEATTLE-TACOMA INTL   Washington          NA
## 5   5                 NORFOLK INTL     Virginia          NA
## 6   6          GUAYAQUIL/S BOLIVAR          N/A          NA
```

Create data frame df.flights

```
df.flights <- sqldf::sqldf("SELECT
  1 AS fid,
  rid AS lookup_id,
  flight_date AS date,
  airport AS origin,
  CASE
    WHEN not airline is not null THEN 'unknown'
    ELSE replace(trim(airline), '*', '')
  END AS airline,
  CASE
    WHEN not aircraft is not null THEN 'unknown'
    ELSE aircraft
  END AS aircraft,
  altitude_ft AS altitude,
  CASE
    WHEN lower(trim(heavy_flag)) == 'yes' THEN 1
    ELSE 0
  END AS heavy
FROM `bds.raw`;
")

n.flights <- nrow(df.flights)
df.flights[,1] <- seq(1, n.flights)


head(df.flights)
```

```
##   fid lookup_id            date                      origin           airline
## 1   1    202152 11/23/2000 0:00                 LAGUARDIA NY        US AIRWAYS
## 2   2    208159  7/25/2001 0:00 DALLAS/FORT WORTH INTL ARPT AMERICAN AIRLINES
## 3   3    207601  9/14/2001 0:00            LAKEFRONT AIRPORT          BUSINESS
## 4   4    215953   9/5/2002 0:00         SEATTLE-TACOMA INTL   ALASKA AIRLINES
## 5   5    219878  6/23/2003 0:00                 NORFOLK INTL   COMAIR AIRLINES
## 6   6    218432  7/24/2003 0:00         GUAYAQUIL/S BOLIVAR AMERICAN AIRLINES
##   aircraft altitude heavy
## 1 Airplane    1,500     1
## 2 Airplane        0     0
## 3 Airplane       50     0
## 4 Airplane       50     1
## 5 Airplane       50     0
## 6 Airplane        0     0
```

Create data frame df.conditions

```
df.conditions <- sqldf::sqldf("select distinct 1 as cid, sky_conditions as sky_condition, NULL as explan


n.conditions <- nrow(df.conditions)
df.conditions[,1] <- seq(1, n.conditions)


head(df.conditions)
```

```
##   cid sky_condition explanation
## 1   1      No Cloud          NA
## 2   2    Some Cloud          NA
## 3   3      Overcast          NA
```

Create data frame df.strikes

```
## Create data frame df.strikes
df.strikes <- sqldf::sqldf("SELECT
  1 AS sid,
  rid AS check_id,
  wildlife_struck AS numbirds,
  impact AS impact,
  altitude_ft AS altitude,
  CASE
    WHEN lower(trim(damage)) == 'caused damage' THEN 1
    ELSE 0
  END AS damage,
  sky_conditions AS conditions
FROM `bds.raw`;
")

n.strikes <- nrow(df.strikes)
df.strikes[,1] <- seq(1, n.strikes)


head(df.strikes)
```

```
##   sid check_id numbirds                 impact altitude damage conditions
## 1   1   202152      859       Engine Shut Down    1,500      1   No Cloud
## 2   2   208159      424                   None        0      1 Some Cloud
## 3   3   207601      261                   None       50      0   No Cloud
## 4   4   215953      806 Precautionary Landing       50      0 Some Cloud
## 5   5   219878      942                   None       50      0   No Cloud
## 6   6   218432      537                   None        0      0   No Cloud
```

clean up tables to avoid constraint failure

```
drop table if exists strikes
```

```
drop table if exists conditions
```

```
drop table if exists flights
```

```
drop table if exists airports
```

Bulk load data into conditions

```
dbWriteTable(mydb, "conditions", df.conditions, overwrite = T, row.names=FALSE)
```

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

Add constraints to tables

```
ALTER TABLE conditions
MODIFY cid INT AUTO_INCREMENT,
ADD PRIMARY KEY (cid);
```

Check tables

```
select * from conditions
```

Table 1: 3 records

cid	sky_condition	explanation
1	No Cloud	NA
2	Some Cloud	NA
3	Overcast	NA

Bulk load data into airports table

```
dbWriteTable(mydb, "airports", df.airports, overwrite = T, row.names=FALSE)
```

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

Add constraint for primary key

```
ALTER TABLE airports
MODIFY aid INT AUTO_INCREMENT,
ADD PRIMARY KEY (aid);
```

Check airports table

```
select airportName, count(*) from airports where airportName = ""
```

Table 2: 1 records

airportName	count(*)
	32

Bulk load data into temp_flights

```r
dbWriteTable(mydb, "temp_flights", df.flights, overwrite = T, row.names=FALSE)
```

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

```sql
drop table if exists flights
```

```sql
select * from temp_flights limit 10
```

Table 3: Displaying records 1 - 10

fid	lookup_id	date	origin	airline	aircraft	altitude	heavy
1	202152	11/23/2000 0:00	LAGUARDIA NY	US AIRWAYS	Airplane	1,500	1
2	208159	7/25/2001 0:00	DALLAS/FORT WORTH INTL ARPT	AMERICAN AIRLINES	Airplane	0	0
3	207601	9/14/2001 0:00	LAKEFRONT AIRPORT	BUSINESS	Airplane	50	0
4	215953	9/5/2002 0:00	SEATTLE-TACOMA INTL	ALASKA AIRLINES	Airplane	50	1
5	219878	6/23/2003 0:00	NORFOLK INTL	COMAIR AIRLINES	Airplane	50	0
6	218432	7/24/2003 0:00	GUAYAQUIL/S BOLIVAR	AMERICAN AIRLINES	Airplane	0	0
7	221697	8/17/2003 0:00	NEW CASTLE COUNTY	BUSINESS	Airplane	150	0
8	236635	3/1/2006 0:00	WASHINGTON DULLES INTL ARPT	UNITED AIRLINES	Airplane	100	0
9	207369	1/6/2000 0:00	ATLANTA INTL	AIRTRAN AIRWAYS	Airplane	0	0
10	204371	1/7/2000 0:00	ORLANDO SANFORD INTL AIRPORT	AIRTOURS INTL	Airplane	0	0

Insert data from temp_flight into flight with origin as fk reference airports aid

```sql
-- Create a new table
CREATE TABLE flights AS
SELECT
  fid AS fid,
  STR_TO_DATE(date, '%m/%d/%y') AS date,
  CASE
    WHEN CAST(REPLACE(altitude, ',', '') AS SIGNED) >= 0 THEN CAST(REPLACE(altitude, ',', '') AS SIGNED
    ELSE NULL
  END AS altitude,
  CASE
    WHEN origin = '' OR origin IS NULL THEN -1
    ELSE (SELECT aid FROM airports WHERE airportName = origin)
  END AS origin,
  CASE
```

```
    WHEN airline = '' THEN 'unknown'
    ELSE airline
  END AS airline,
  CASE
    WHEN aircraft = '' THEN 'unknown'
    ELSE REPLACE(aircraft, '"', '')
  END AS aircraft,
  heavy AS heavy
FROM temp_flights;
```

Add constraints to flights

```
ALTER TABLE flights
MODIFY fid INT AUTO_INCREMENT,
ADD PRIMARY KEY (fid);
```

```
ALTER TABLE flights
ADD CONSTRAINT altitude_check CHECK (altitude >= 0);
```

Check flights table

```
select * from flights limit 100
```

Table 4: Displaying records 1 - 10

fid	date	altitude	origin	airline	aircraft	heavy
1	2020-11-23	1500	1	US AIRWAYS	Airplane	1
2	2020-07-25	0	2	AMERICAN AIRLINES	Airplane	0
3	2020-09-14	50	3	BUSINESS	Airplane	0
4	2020-09-05	50	4	ALASKA AIRLINES	Airplane	1
5	2020-06-23	50	5	COMAIR AIRLINES	Airplane	0
6	2020-07-24	0	6	AMERICAN AIRLINES	Airplane	0
7	2020-08-17	150	7	BUSINESS	Airplane	0
8	2020-03-01	100	8	UNITED AIRLINES	Airplane	0
9	2020-01-06	0	9	AIRTRAN AIRWAYS	Airplane	0
10	2020-01-07	0	10	AIRTOURS INTL	Airplane	0

```
drop table if exists strikes
```

Bulk load data into temp_strikes

```
dbWriteTable(mydb, "temp_strikes", df.strikes, overwrite = T, row.names=FALSE)
```

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

Insert data from temp_strikes into strikes with fid and condition as fk reference flight fid and condition cid

```sql
-- Create a new table
CREATE TABLE strikes AS
SELECT
  sid AS sid,
  (SELECT fid FROM temp_flights WHERE check_id = lookup_id) AS fid,
  numbirds AS numbirds,
  impact AS impact,
  damage AS damage,
  altitude AS altitude,
  (SELECT cid FROM conditions WHERE sky_condition = conditions) AS conditions
FROM temp_strikes;
```

```sql
ALTER TABLE strikes
MODIFY sid INT AUTO_INCREMENT,
ADD PRIMARY KEY (sid);
```

Check strikes table

```sql
SELECT * FROM strikes LIMIT 100
```

Table 5: Displaying records 1 - 10

sid	fid	numbirds	impact	damage	altitude	conditions
1	1	859	Engine Shut Down	1	1,500	1
2	2	424	None	1	0	2
3	3	261	None	0	50	1
4	4	806	Precautionary Landing	0	50	2
5	5	942	None	0	50	1
6	6	537	None	0	0	1
7	7	227	Other	1	150	1
8	8	320	Other	1	100	2
9	9	9	Aborted Take-off	0	0	2
10	10	4	None	0	0	2

Drop temp_strikes table

```sql
DROP TABLE IF EXISTS temp_strikes
```

Drop temp_flights table

```sql
DROP TABLE IF EXISTS temp_flights
```

The top 10 states with the greatest number of bird strike incidents

```
SELECT
  a.airportState AS state,
  COUNT(s.sid) AS num_incidents
FROM
  strikes AS s
  JOIN flights AS f ON s.fid = f.fid
  JOIN airports AS a ON f.origin = a.aid
GROUP BY
  state
ORDER BY
  num_incidents DESC
LIMIT 10;
```

Table 6: Displaying records 1 - 10

state	num_incidents
California	2499
Texas	2445
Florida	2045
New York	1316
Illinois	1007
Pennsylvania	985
Missouri	956
Kentucky	806
Ohio	773
Hawaii	716

The airlines that had an above average number bird strike incidents.

```
SELECT
  airline AS airline_name,
  COUNT(s.sid) AS num_incidents
FROM
  strikes AS s
  JOIN flights AS f ON s.fid = f.fid
GROUP BY
  airline_name
HAVING
  num_incidents > (
    SELECT AVG(incident_count)
    FROM (
      SELECT COUNT(sid) AS incident_count
      FROM strikes
      GROUP BY fid
    ) AS avg_count
  )
ORDER BY
  num_incidents DESC;
```

Table 7: Displaying records 1 - 10

airline_name	num_incidents
SOUTHWEST AIRLINES	4628
BUSINESS	3074
AMERICAN AIRLINES	2058
DELTA AIR LINES	1349
US AIRWAYS	1337
AMERICAN EAGLE AIRLINES	932
SKYWEST AIRLINES	891
JETBLUE AIRWAYS	708
UPS AIRLINES	590
UNITED AIRLINES	506

the (total) number of birds that struck aircraft by month

```sql
SELECT
    DATE_FORMAT(date, '%m') AS month,
    SUM(numbirds) AS total_birds_struck
FROM
    strikes
    JOIN flights ON strikes.fid = flights.fid
GROUP BY
    month
ORDER BY
    month;
```

Table 8: Displaying records 1 - 10

month	total_birds_struck
00	141
01	3106
02	2602
03	3539
04	3802
05	4077
06	5209
07	9344
08	11013
09	9201

```r
result <- dbGetQuery(mydb, "
    SELECT
        DATE_FORMAT(date, '%m') AS month,
        SUM(numbirds) AS total_birds_struck
    FROM
        strikes
        JOIN flights ON strikes.fid = flights.fid
    GROUP BY
```

```
        month
    ORDER BY
        month
")
```

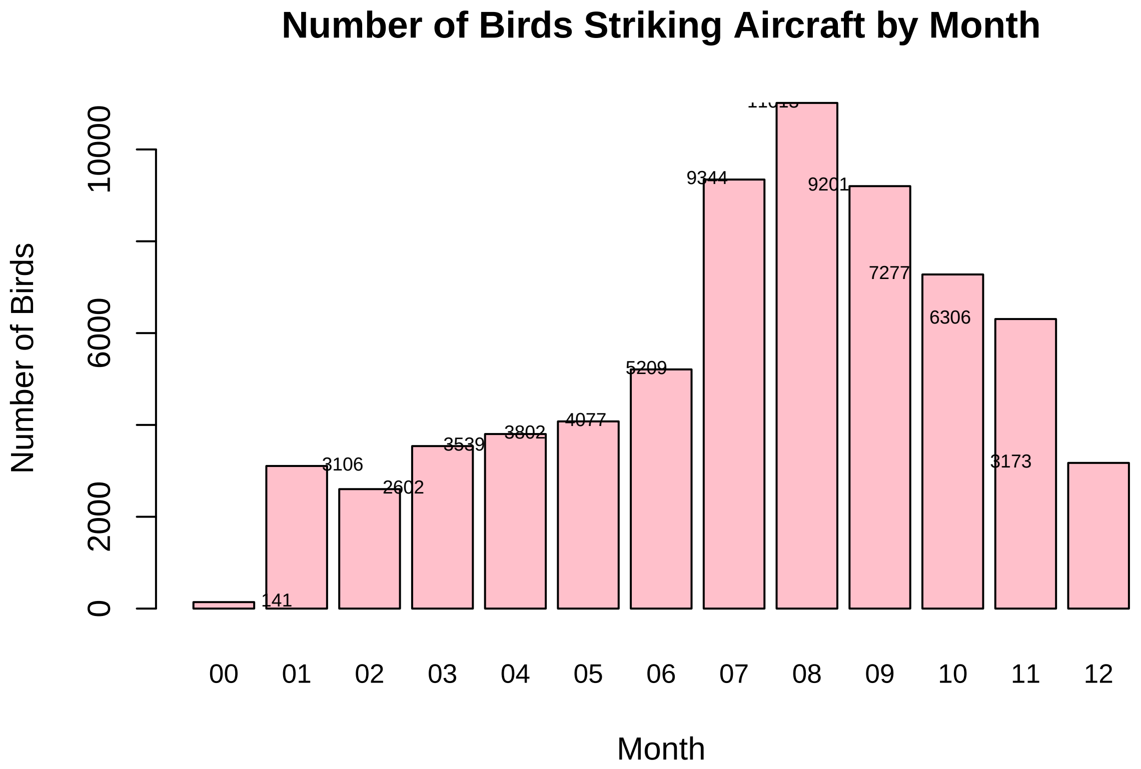
```
## Warning in .local(conn, statement, ...): Decimal MySQL column 1 imported as
## numeric
```

```
# Display the first six rows of the dataframe
head(result)
```

```
##   month total_birds_struck
## 1    00                141
## 2    01               3106
## 3    02               2602
## 4    03               3539
## 5    04               3802
## 6    05               4077
```

```
# Create the column chart
barplot(result$total_birds_struck, names.arg = result$month, xlab = "Month", ylab = "Number of Birds",
        main = "Number of Birds Striking Aircraft by Month",
        col = "pink", border = "black", cex.names = 0.8)

# Add data labels above the bars
text(x = 1:length(result$month), y = (result$total_birds_struck), labels = result$total_birds_struck,
     pos = 4, cex = 0.6)
```



```
drop PROCEDURE if exists AddBirdStrike
```

```
CREATE PROCEDURE AddBirdStrike(
    IN numBirdS INT,
    IN airport_name TEXT,
    IN airport_state TEXT,
    IN impact TEXT,
    IN damage BOOLEAN,
    IN altitude INT,
    IN `condition` TEXT,
    IN strike_date DATE,
    IN airline TEXT,
    IN aircraft TEXT,
    IN heavy BOOLEAN
)
BEGIN
    DECLARE aid INT;
    DECLARE fid INT;
    DECLARE cid INT;

    -- Check if the airport exists, and if not, insert a new airport
    SELECT distinct aid INTO aid FROM airports WHERE airportName = airport_name AND airportState = airpo

    IF aid IS NULL THEN
        INSERT INTO airports (airportName, airportState) VALUES (airport_name, airport_state);
        SET aid = LAST_INSERT_ID();
    END IF;

    -- Insert a new flight or use an existing one
    SELECT distinct fid INTO fid FROM flights WHERE origin = aid AND date = strike_date AND airline = a

    IF fid IS NULL THEN
        INSERT INTO flights (origin, date, airline, aircraft, altitude, heavy)
        VALUES (aid, strike_date, airline, aircraft, altitude, heavy);
        SET fid = LAST_INSERT_ID();
    END IF;

    -- Check if the condition exists, and if not, insert a new condition
    SELECT distinct cid INTO cid FROM conditions WHERE sky_condition = `condition`;

    IF cid IS NULL THEN
        INSERT INTO conditions (sky_condition) VALUES (`condition`);
        SET cid = LAST_INSERT_ID();
    END IF;

    -- Insert the bird strike incident
    INSERT INTO strikes (numbirds, fid, impact, damage, altitude, conditions)
    VALUES (numBirdS, fid, impact, damage, altitude, cid);
END;
```

```
CALL AddBirdStrike(5201314, 'LAGUARDIA NY', 'New York', 'None', 0, 550, 'Rain', '2022-03-04', 'EVA', 'A
```

```
select * from conditions
```

Table 9: 4 records

cid	sky_condition	explanation
1	No Cloud	NA
2	Some Cloud	NA
3	Overcast	NA
4	Rain	NA

```
SELECT * FROM strikes
order by sid desc limit 10
```

Table 10: Displaying records 1 - 10

sid	fid	numbirds	impact	damage	altitude	conditions
25559	25559	5201314	None	0	550	4
25558	25558	1	None	1	0	1
25557	25557	1	None	0	0	2
25556	25556	1		0		1
25555	25555	1	None	0	0	2
25554	25554	1	None	0	1,500	3
25553	25553	1	None	0	10	1
25552	25552	1	None	0	50	2

sid	fid	numbirds	impact	damage	altitude	conditions
25551	25551	1	None	0	0	1
25550	25550	1	None	0	0	3

```
dbDisconnect(mydb)
```

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