Homework 9

[Your Name Here]

Due Tuesday 11/16/2021

Classmates/other resources consulted: [type answer here]

```
library(DBI)
library(RSQLite)
```

Question 1 (12 points)

a. **Set up a connection to the database in the speciesdata.sqlite fil; the rest of this file will use this connection. This database has information from a long-term ecological study being conducted near Portal, Arizona (see https://www.biorxiv.org/content/10.1101/332783v3 for more information).**

```
db <- dbConnect(
          SQLite(), ## Dialect of SQL we will be using
          dbname = "speciesdata.sqlite" ## Name of database and where to find it
)
db
```

```
## <SQLiteConnection>
##   Path: /Users/razakee/cs36directory/speciesdata.sqlite
##   Extensions: TRUE
```

b. **What are the tables in the species database?**

```
dbListTables(db)
```

```
## [1] "plots"        "species"      "sqlite_stat1" "sqlite_stat4" "surveys"
```

c. **Set up a code chunk that can run SQL queries for this database. What are the columns of the species table?**

```
SELECT * FROM species
```

Table 1: Displaying records 1 - 10

species_id	genus	species	taxa
AB	Amphispiza	bilineata	Bird
AH	Ammospermophilus	harrisi	Rodent
AS	Ammodramus	savannarum	Bird
BA	Baiomys	taylori	Rodent
CB	Campylorhynchus	brunneicapillus	Bird
CM	Calamospiza	melanocorys	Bird
CQ	Callipepla	squamata	Bird
CS	Crotalus	scutalatus	Reptile
CT	Cnemidophorus	tigris	Reptile
CU	Cnemidophorus	uniparens	Reptile

species_id genus species taxa

d. **Consider the tables species, plots, and surveys. What is the number of rows and columns in each?**

species: 54 rows, 4 columns plots: 24 rows, 2 columns surveys: 35549 rows, 9 columns

Question 2 (24 points)

a. **Output just the year, month, day, and species_id columns from the surveys data set.**

```
SELECT year, month, day, species_id FROM surveys
```

Table 2: Displaying records 1 - 10

year	month	day	species_id
1977	7	16	NL
1977	7	16	NL
1977	7	16	DM
1977	7	16	DM
1977	7	16	DM
1977	7	16	PF
1977	7	16	PE
1977	7	16	DM
1977	7	16	DM
1977	7	16	PF

b. **Edit your code from part a so that your column names are capitalized and have spaces instead of underscores.**

```
SELECT year AS Year, month AS Month, day AS Day, species_id AS "Species ID" FROM surveys
```

Table 3: Displaying records 1 - 10

Year	Month	Day	Species ID
1977	7	16	NL
1977	7	16	NL
1977	7	16	DM
1977	7	16	DM
1977	7	16	DM
1977	7	16	PF
1977	7	16	PE
1977	7	16	DM
1977	7	16	DM
1977	7	16	PF

c. **How many different species IDs appear in this table?**

```
SELECT count(species_id) FROM surveys
```

Table 4: 1 records

count(species_id)
34786

d. **How many different combinations of month and day appear in this table?**

```
SELECT DISTINCT month,
day FROM surveys
```

Table 5: Displaying records 1 - 10

month	day
7	16
7	17
7	18
8	19
8	20
8	21
9	11
9	12
9	13
10	16

297 distinct combos

e. **Arrange the surveys table by species ID. When there is a tie in species ID, next arrange by year, then by month, then by day. Be sure you display the species ID, year, month, and day columns in your answer.**

```
SELECT species_id, year, month, day
  FROM surveys
  ORDER BY species_id, year, month, day
```

Table 6: Displaying records 1 - 10

species_id	year	month	day
NA	1977	10	17
NA	1977	10	17
NA	1977	10	17
NA	1977	11	13
NA	1977	11	13
NA	1977	11	13
NA	1977	11	14
NA	1977	11	14
NA	1977	11	14
NA	1977	11	14

f. **Arrange the surveys table by weight, from highest to lowest. There is no need to include any tiebreakers.**

```
SELECT *
  FROM surveys
  ORDER BY weight desc
```

Table 7: Displaying records 1 - 10

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
33049	11	17	2001	12	NL	M	33	280
12871	5	28	1987	2	NL	M	32	278
15459	1	11	1989	9	NL	M	36	275
2133	10	25	1979	2	NL	F	33	274
12729	4	26	1987	2	NL	M	32	270
13114	7	26	1987	2	NL	M	NA	269
30175	1	8	2000	2	NL	M	34	265
4962	11	22	1981	12	NL	F	NA	264
12602	4	6	1987	2	NL	M	34	260
13025	7	1	1987	2	NL	M	33	260

g. **Calculate the min, max, average, and total weights across all observations in the surveys table.**

```
SELECT min(weight)
  FROM surveys
```

Table 8: 1 records

min(weight)
4

```
SELECT max(weight)
  FROM surveys
```

Table 9: 1 records

max(weight)
280

```
SELECT avg(weight)
  FROM surveys
```

Table 10: 1 records

avg(weight)
42.67243

```
SELECT sum(weight)
  FROM surveys
```

Table 11: 1 records

sum(weight)
1377594

h. **In the tidyverse, if you try to take an average/sum/min/max/etc. of a column that contains NA values, by default it returns NA and you have to explicitly add the parameter na.rm = TRUE to remove the NA values from the calculations. Do these operations have different default behaviors in SQL? Reference the surveys table and your answer to part f in your explanation.**

```
SELECT *
  FROM surveys
  WHERE weight IS NULL
```

Table 12: Displaying records 1 - 10

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
1	7	16	1977	2	NL	M	32	NA
2	7	16	1977	3	NL	M	33	NA
3	7	16	1977	2	DM	F	37	NA

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
4	7	16	1977	7	DM	M	36	NA
5	7	16	1977	3	DM	M	35	NA
6	7	16	1977	1	PF	M	14	NA
7	7	16	1977	2	PE	F	NA	NA
8	7	16	1977	1	DM	M	37	NA
9	7	16	1977	1	DM	F	34	NA
10	7	16	1977	6	PF	F	20	NA

The default behavior is different in sql because When we average weight we get a number but there are null values in the weight columns.

Question 3 (8 points)

Filter the surveys table to find the following collections of observations. Only keep the record_id, species_id, weight, and hindfoot_length in your answers.

a. **All observations of species "CQ", which is a Scaled Quail, and species "CB", which is a cactus wren**

```
SELECT *
  FROM surveys
  WHERE (species_id == "CQ" or  species_id == "CB")
```

Table 13: Displaying records 1 - 10

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
3189	8	13	1980	23	CB	NA	NA	NA
3200	8	13	1980	19	CQ	NA	NA	NA
3336	10	11	1980	20	CQ	NA	NA	NA
3700	1	11	1981	23	CQ	NA	NA	NA
3717	1	11	1981	23	CQ	NA	NA	NA
3722	1	11	1981	23	CQ	NA	NA	NA
4707	7	30	1981	15	CB	NA	NA	NA
4743	7	31	1981	20	CQ	NA	NA	NA
6099	6	29	1982	23	CQ	NA	NA	NA
6185	7	25	1982	5	CQ	NA	NA	NA

b. **All observations where the species_id begins with the letter C**

```
SELECT *
  FROM surveys
  WHERE species_id LIKE "C%"
```

Table 14: Displaying records 1 - 10

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
3189	8	13	1980	23	CB	NA	NA	NA
3190	8	13	1980	9	CM	NA	NA	NA
3200	8	13	1980	19	CQ	NA	NA	NA
3208	8	13	1980	8	CM	NA	NA	NA
3250	9	7	1980	21	CM	NA	NA	NA
3263	9	7	1980	10	CM	NA	NA	NA
3277	9	7	1980	23	CM	NA	NA	NA
3285	9	7	1980	21	CM	NA	NA	NA
3291	9	7	1980	18	CM	NA	NA	NA
3306	9	8	1980	2	CM	NA	NA	NA

c. **All observations where the hindfoot_length is NA**

```
SELECT *
  FROM surveys
  WHERE hindfoot_length is NULL
```

Table 15: Displaying records 1 - 10

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
7	7	16	1977	2	PE	F	NA	NA
14	7	16	1977	8	DM	NA	NA	NA
19	7	16	1977	4	PF	NA	NA	NA
29	7	17	1977	11	PP	M	NA	NA
34	7	17	1977	17	DM	NA	NA	NA
57	7	18	1977	22	DM	NA	NA	NA
77	8	19	1977	4	SS	NA	NA	NA
93	8	20	1977	18	DM	NA	NA	42
106	8	20	1977	12	NL	NA	NA	NA
107	8	20	1977	18	NL	NA	NA	NA

d. **Observations in rows 20,001 to 20,500 in the original surveys data set. DO NOT use the record_ids column; use code that would work for a table that didn't have a row ID column.**

```
SELECT month, day, year
  FROM surveys
  LIMIT 500
  OFFSET 20000
```

Table 16: Displaying records 1 - 10

month	day	year
5	3	1992
5	3	1992
5	3	1992

month	day	year
5	3	1992
5	3	1992
5	3	1992
5	3	1992
5	3	1992
5	30	1992
5	30	1992

Question 4 (9 points)

a. **Filter the species table to find all animals where the genus includes the substring "mys", which is the Greek word for mouse**

```
SELECT *
  FROM species
  WHERE genus LIKE "%mys%"
```

Table 17: Displaying records 1 - 10

species_id	genus	species	taxa
BA	Baiomys	taylori	Rodent
DM	Dipodomys	merriami	Rodent
DO	Dipodomys	ordii	Rodent
DS	Dipodomys	spectabilis	Rodent
DX	Dipodomys	sp.	Rodent
OL	Onychomys	leucogaster	Rodent
OT	Onychomys	torridus	Rodent
OX	Onychomys	sp.	Rodent
PE	Peromyscus	eremicus	Rodent
PL	Peromyscus	leucopus	Rodent

b. **Filter the species table to find all *rodent* species where the genus does not include the substring "mys"**

```
SELECT *
  FROM species
  WHERE taxa == "Rodent" and genus NOT LIKE "%mys%"
```

Table 18: Displaying records 1 - 10

species_id	genus	species	taxa
AH	Ammospermophilus	harrisi	Rodent
NL	Neotoma	albigula	Rodent
NX	Neotoma	sp.	Rodent
PB	Chaetodipus	baileyi	Rodent

species_id	genus	species	taxa
PF	Perognathus	flavus	Rodent
PH	Perognathus	hispidus	Rodent
PI	Chaetodipus	intermedius	Rodent
PP	Chaetodipus	penicillatus	Rodent
PX	Chaetodipus	sp.	Rodent
SF	Sigmodon	fulviventer	Rodent

c. **Output a data table that has columns for record_id, species_id, the animal's weight in ounces (the original weight in the data set), and the animal's weight in pounds (the weight in ounces divided by 16). Filter out any NA values so that your results can be seen more clearly.**

```
SELECT record_id, species_id, weight AS "weight in ounces", (weight / 16) AS "weight in pounds"
  FROM surveys
  WHERE (record_id NOT NULL and species_id NOT NULL and weight NOT NULL)
```

Table 19: Displaying records 1 - 10

record_id	species_id	weight in ounces	weight in pounds
63	DM	40	2.5000
64	DM	48	3.0000
65	DM	29	1.8125
66	DM	46	2.8750
67	DM	36	2.2500
68	DO	52	3.2500
69	PF	8	0.5000
70	OX	22	1.3750
71	DM	35	2.1875
74	PF	7	0.4375

Question 5 (15 points)

a. (2 points) **What column(s) to the surveys and species tables have in common? What column(s) to the surveys and plots table have in common? What column(s) to the species and plots table have in common?**

```
SELECT *
  FROM surveys
```

Table 20: Displaying records 1 - 10

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
1	7	16	1977	2	NL	M	32	NA
2	7	16	1977	3	NL	M	33	NA
3	7	16	1977	2	DM	F	37	NA

record_id	month	day	year	plot_id	species_id	sex	hindfoot_length	weight
4	7	16	1977	7	DM	M	36	NA
5	7	16	1977	3	DM	M	35	NA
6	7	16	1977	1	PF	M	14	NA
7	7	16	1977	2	PE	F	NA	NA
8	7	16	1977	1	DM	M	37	NA
9	7	16	1977	1	DM	F	34	NA
10	7	16	1977	6	PF	F	20	NA

```
SELECT *
  FROM species
```

Table 21: Displaying records 1 - 10

species_id	genus	species	taxa
AB	Amphispiza	bilineata	Bird
AH	Ammospermophilus	harrisi	Rodent
AS	Ammodramus	savannarum	Bird
BA	Baiomys	taylori	Rodent
CB	Campylorhynchus	brunneicapillus	Bird
CM	Calamospiza	melanocorys	Bird
CQ	Callipepla	squamata	Bird
CS	Crotalus	scutalatus	Reptile
CT	Cnemidophorus	tigris	Reptile
CU	Cnemidophorus	uniparens	Reptile

```
SELECT *
  FROM plots
```

Table 22: Displaying records 1 - 10

plot_id	plot_type
1	Spectab exclosure
2	Control
3	Long-term Krat Exclosure
4	Control
5	Rodent Exclosure
6	Short-term Krat Exclosure
7	Rodent Exclosure
8	Control
9	Spectab exclosure
10	Rodent Exclosure

surveys and species species_id

surveys and plots plot_id

species and plots none

b.(3 points) **Join the survey and species table on their common columns to create a new table with columns for record_id, species_id, and the type of animal (rodent, reptile, bird, etc.). Do this join *without* using the join keyword.**

```
SELECT record_id, surveys.species_id, taxa AS "type of animal"
  FROM surveys , species
  WHERE (surveys.species_id = species.species_id)
```

Table 23: Displaying records 1 - 10

record_id	species_id	type of animal
1	NL	Rodent
2	NL	Rodent
3	DM	Rodent
4	DM	Rodent
5	DM	Rodent
6	PF	Rodent
7	PE	Rodent
8	DM	Rodent
9	DM	Rodent
10	PF	Rodent

c. (3 points) **Produce the same tibble as the previous part, but this time using the JOIN keyword and the USING keyword.**

```
SELECT record_id, species_id, taxa AS "type of animal"
  FROM surveys
    JOIN species USING (species_id)
```

Table 24: Displaying records 1 - 10

record_id	species_id	type of animal
1	NL	Rodent
2	NL	Rodent
3	DM	Rodent
4	DM	Rodent
5	DM	Rodent
6	PF	Rodent
7	PE	Rodent
8	DM	Rodent
9	DM	Rodent
10	PF	Rodent

d. (3 points) **Produce the same tibble as the previous two parts, but this time using the JOIN keyword and the ON keyword.**

```
SELECT  record_id, surveys.species_id, taxa AS "type of animal"
  FROM surveys
    JOIN species
    ON surveys.species_id = species.species_id
```

Table 25: Displaying records 1 - 10

record_id	species_id	type of animal
1	NL	Rodent
2	NL	Rodent
3	DM	Rodent
4	DM	Rodent
5	DM	Rodent
6	PF	Rodent
7	PE	Rodent
8	DM	Rodent
9	DM	Rodent
10	PF	Rodent

e. (2 points) **Explain when you can use the USING keyword, and when you can use the ON keyword.**

If the columns we're joining by have different names, use ON instead of USING.

f. (2 points) **Filter the tibble produced in parts (b)-(d) to only include animals that are not rodents. You can do the join any way you'd like.**

```
SELECT  record_id, surveys.species_id, taxa
  FROM surveys
    JOIN species
    ON surveys.species_id = species.species_id
    where taxa != "Rodent"
```

Table 26: Displaying records 1 - 10

record_id	species_id	taxa
779	SA	Rabbit
813	SA	Rabbit
3126	AB	Bird
3146	AB	Bird
3152	AB	Bird
3153	AB	Bird
3189	CB	Bird
3190	CM	Bird
3200	CQ	Bird
3208	CM	Bird

Question 6 (12 points)

a. **Create a column of all the species IDs that appear in surveys, and a column of all the species IDs that appear in species. Use them to make a column of all species_id that appear in *both* tibbles.**

```
SELECT  distinct surveys.species_id AS SPECIESID
  FROM surveys, species
  WHERE (surveys.species_id = species.species_id)
```

Table 27: Displaying records 1 - 10

SPECIESID
NL
DM
PF
PE
DS
PP
SH
OT
DO
OX

b. **Use code to check whether there are any species_ids that appear in species but not surveys.**

```
SELECT  count (distinct surveys.species_id), count (distinct species.species_id)
  FROM surveys, species
```

Table 28: 1 records

count (distinct surveys.species_id)	count (distinct species.species_id)
48	54

c. **Are there any species_ids that appear in surveys but not species? What does this tell you about what observations/rows might be missing from the joined tables you created in the previous question?**

There are 48 species that appear in both surveys and species. There are exactly 48 non-null species in surveys meaning all 48 of those non-nulls appear in species table as well. This means there are no non-null species ids in surveys that dont also appear in species table. na is the only species id present in surveys and not species

d. **Use an appropriate join to add the taxa information from species to the surveys table, keeping the record_id, species_id, and taxa column. Make sure your resulting table includes any rows you identified as missing in the previous part.**

```
SELECT surveys.record_id , surveys.species_id , taxa
  FROM surveys
    JOIN species
    ON surveys.species_id = species.species_id
```

Table 29: Displaying records 1 - 10

record_id	species_id	taxa
1	NL	Rodent
2	NL	Rodent
3	DM	Rodent
4	DM	Rodent
5	DM	Rodent
6	PF	Rodent
7	PE	Rodent
8	DM	Rodent
9	DM	Rodent
10	PF	Rodent

Question 7 (18 points)

a. **Create a table that has a column that contains each species_id that appears in surveys, and a column for how many times that species_id appears in surveys. Give this second column an appropriate name.**

```
SELECT surveys.species_id, count(surveys.species_id ) AS COUNT
  FROM surveys
  WHERE surveys.species_id NOT null
  Group by surveys.species_id
```

Table 30: Displaying records 1 - 10

species_id	COUNT
AB	303
AH	437
AS	2
BA	46
CB	50
CM	13
CQ	16
CS	1
CT	1
CU	1

b. **Create a table that has a row for each species ID that appears in surveys, and columns for the species_id, number of observations of that species, and average weight of that species. Give your columns appropriate names. Arrange your table by the average weight of each species, from largest to smallest.**

```
SELECT surveys.species_id, count(surveys.species_id ) AS COUNT, avg(surveys.weight) AS mean_weight
  FROM surveys
  WHERE surveys.species_id NOT null and surveys.weight NOT null
  Group by surveys.species_id
  ORDER BY mean_weight desc
```

Table 31: Displaying records 1 - 10

species_id	COUNT	mean_weight
NL	1152	159.24566
DS	2344	120.13055
SS	2	93.50000
SH	141	73.14894
SF	41	58.87805
SO	41	55.41463
DO	2904	48.87052
DM	10262	43.15786
PB	2810	31.73594
OL	970	31.57526

c. **Modify your code from the previous part so that the summary tibble only includes information on observations from 1990 or later.**

```
SELECT surveys.species_id, count(surveys.species_id ) AS COUNT, avg(surveys.weight) AS mean_weight
  FROM surveys
  WHERE surveys.species_id NOT null and surveys.weight NOT null and year >= 1990
  Group by surveys.species_id
  ORDER BY mean_weight desc
```

Table 32: Displaying records 1 - 10

species_id	COUNT	mean_weight
NL	328	161.26829
DS	89	113.01124
SH	46	69.28261
SF	36	59.44444
SO	41	55.41463
DO	1476	48.87195
DM	4781	43.88684
PB	2810	31.73594
PH	18	31.33333
OL	212	27.87736

d. **Modify your code from part (b) so that the summary tibble only includes species whose average weight is greater than 50.**

```
SELECT surveys.species_id, count(surveys.species_id ) AS COUNT, (avg(surveys.weight)>50)*avg(surveys.wei
  FROM surveys
  WHERE surveys.species_id NOT null and surveys.weight NOT null and year >= 1990
  Group by surveys.species_id
  HAVING mean_weight != 0
  ORDER BY mean_weight desc
```

Table 33: 5 records

species_id	COUNT	mean_weight
NL	328	161.26829
DS	89	113.01124
SH	46	69.28261
SF	36	59.44444
SO	41	55.41463

e. **Modify your code from (b) to have a separate column for the average weight of species across all the observations in 1990 or later; you should also still have a column for the average weight across all observations, regardless of year. Give your columns appropriate names.**

```
SELECT surveys.species_id,
count(surveys.species_id ) AS COUNT,
avg(surveys.weight) AS tot_mean_weight,
avg(surveys.weight) FILTER(WHERE(year >1990 )) AS mean_weight_after_1990

  FROM surveys
  WHERE surveys.species_id NOT null and surveys.weight NOT null
  Group by surveys.species_id
  HAVING tot_mean_weight > 50
  ORDER BY tot_mean_weight desc
```

Table 34: 6 records

species_id	COUNT	tot_mean_weight	mean_weight_after_1990
NL	1152	159.24566	161.94649
DS	2344	120.13055	111.21918
SS	2	93.50000	NA
SH	141	73.14894	67.43243
SF	41	58.87805	66.27778
SO	41	55.41463	55.41463

f. **Modify your code from (b) to also include a column for the taxa of each observation. Make sure you not not remove any observations; see the previous question.**

```
SELECT surveys.species_id, count(surveys.species_id ) AS COUNT, avg(surveys.weight) AS mean_weight, taxa
  FROM surveys
    JOIN species USING (species_id)
  WHERE surveys.species_id NOT null and surveys.weight NOT null
  Group by surveys.species_id
  ORDER BY mean_weight desc
```

Table 35: Displaying records 1 - 10

species_id	COUNT	mean_weight	taxa
NL	1152	159.24566	Rodent
DS	2344	120.13055	Rodent
SS	2	93.50000	Rodent
SH	141	73.14894	Rodent
SF	41	58.87805	Rodent
SO	41	55.41463	Rodent
DO	2904	48.87052	Rodent
DM	10262	43.15786	Rodent
PB	2810	31.73594	Rodent
OL	970	31.57526	Rodent

Question 8 (2 points)

Close your connection to the species data database.

```
dbDisconnect(db)
```