SQL CHEAT SHEET

Create DB - Terminal (SQL file and database in same location)
dropdb database_name delete a database
createdb database_name create database
psql -d database_name -f sql_file.sql link database to sql file (seeding the database)

View Tables & Fields in Terminal
psql -l list databases
psql -d database_name Get into database
\d list tables in database
\d+ table_name list fields in table
\q to get out of psql

R Studio

Example Connect to Database in R Studio

```
library(RPostgreSQL)
home_location <- Sys.getenv("HOME")
credentials_location <- paste0(home_location,"/credentials.R")
source(credentials_location)
db_connect <- dbConnect(drv = PostgreSQL(), user = postgresql_username, password =
postgresql_password, dbname = "zoo", host = "localhost")
```

rm(postgresql_username, postgresql_password) *removes u/n and p/w from global environment*

dbListTables(conn = db_connect) list tables in database
dbListFields(conn = db_connect, name = “table_name”) list fields in table

dbDisconnect(db_connect) disconnect database at end

SQL QUERY

SELECT AS / DISTINCT() / COUNT() / SUM() / MIN() / AVG() / MAX() / CASE WHEN & ELSE
FROM AS t1
JOIN table_name AS t2 (INNER JOIN, LEFT JOIN, FULL OUTER)
  ON t1.col_id = t2.col_id
WHERE AND / OR / BETWEEN / NOT / IS NOT NULL / IS NULL / IN / LIKE %
GROUP BY (Group by specific column then apply aggregate function)
  HAVING (filter groups by some value of aggregate function)
ORDER BY 1 / 2 / Column Name / ASC / DESC NULL LAST
LIMIT number of rows to return

CONSIDER
Primary Key Uniquely identifies row (only one of) **Foreign Key** reference in one tables records to another’s primary key
INNER JOIN - only where matches
LEFT JOIN - all in left table and matching in right
FULL OUTER JOIN - everything
UNION Stack tables on top of each other

Many to Many Table JOIN example

```
SELECT
  a.name AS animal_name,
  cs.day, k.name AS keeper_name
FROM
  (animals AS a INNER JOIN care_schedule AS cs
    ON a.id = cs.animal_id)
INNER JOIN keepers AS k
ON cs.keeper_id = k.id
ORDER BY a.name, cs.day
```

Useful tip - Check for duplicate joining keys

```
SELECT
  id,
  COUNT(*) as id_count
FROM table
GROUP BY id
HAVING count(*) > 1
```

Useful tip - finding entries which don't match in a JOIN

```
SELECT
  a.*
FROM animals AS a
LEFT JOIN diets AS d
ON a.diet_id = d.id
WHERE d.id IS NULL
```

USEFUL functions

CAST(column_name **AS INT)** change column type to integer (or other data type)
ORDER BY wont pick up both lower and upper case so make all **UPPER(**col_name**)** or **LOWER(**col_name**)**
ROUND(AVG(col_name, 2**))** round to specified number of spaces