SPS_Data607_Week2_2A - David Chen

SQL and R – Movie Ratings

Collect simple movie-rating data, store it in a SQL database, and analyze it in R.

Implementation Notes:

PostgreSQL is the recommended (and supported) relational database for this assignment. If you are unable to install and configure PostgreSQL in your environment, you may find SQLite easier to use.
Students experienced with relational databases may use any local or cloud-based database for their assignment. NEVER include passwords in your code.
For this assignment, the ability to reproduce your work in my environment is not required, but you need to provide all of the necessary code, include the code used to create and populate tables.

Task Description

Select six recent popular movies (or television episodes or books or songs or …) Ask at least five people to rate each movie they have seen on a 1–5 scale. Store the collected ratings in a SQL database of your choice. Load the data from SQL into R as a dataframe.

Approach

I need to demonstrate how to use an SQL connector in R and execute SQL code within R to read and write data in a database. It also shows how SQL can be used to perform queries. So I will have to know the libraries for connector and how to run SQL in R.

Installation

Created a new Ubuntu 25.10 CT in PVE node.

```
apt update && apt upgrade
apt install postgresql
sudo -u postgres psql
```

Adding login password

ALTER USER postgres PASSWORD 'YourStrongPassword';

```
nano /etc/postgresql/*/main/postgresql.conf
nano /etc/postgresql/*/main/pg_hba.conf
ufw allow 5432/tcp
```

update these 2 files and open firewall to allow pgadmin4 remote access to this CT.

also allowing password to login database.

Now ask ChatGPT to create this simple database based on the requirements.

```
CREATE TABLE movies (
    movie_id SERIAL PRIMARY KEY,
    title VARCHAR(100) NOT NULL
);
CREATE TABLE ratings (
    rating_id SERIAL PRIMARY KEY,
    movie_id INT REFERENCES movies(movie_id),
    rater_name VARCHAR(50) NOT NULL,
    rating INT CHECK (rating BETWEEN 1 AND 5)
);
INSERT INTO movies (title) VALUES
('Avatar: The Way of Water'),
('Oppenheimer'),
('Barbie'),
('Stranger Things S5'),
('The Marvels'),
('Killers of the Flower Moon');
INSERT INTO ratings (movie_id, rater_name, rating) VALUES
(1, 'Alice', 5),
(1, 'Bob', 4),
(1, 'Charlie', 3),
(1, 'David', 4),
(1, 'Eve', 5),

(2, 'Alice', 4),
(2, 'Bob', 5),
(2, 'Charlie', 4),
(2, 'David', 3),
(2, 'Eve', 4),

(3, 'Alice', 3),
(3, 'Bob', 4),
(3, 'Charlie', 5),
(3, 'David', 3),
(3, 'Eve', 4),

(4, 'Alice', 5),
(4, 'Bob', 5),
(4, 'Charlie', 4),
(4, 'David', 4),
(4, 'Eve', 5),

(5, 'Alice', 3),
(5, 'Bob', 4),
(5, 'Charlie', 3),
(5, 'David', 4),
(5, 'Eve', 3),

(6, 'Alice', 4),
(6, 'Bob', 4),
(6, 'Charlie', 5),
(6, 'David', 5),
(6, 'Eve', 4);
```

After this step , manually delete some values to null

Swtich to RStudio and connect to PostgreSQL

```
#install.packages("DBI")       # Generic database interface
#install.packages("RPostgres") # PostgreSQL driver
```

```r
library(DBI)
library(RPostgres)
```

```r
con <- dbConnect(
  RPostgres::Postgres(),
  dbname = "chatgpt_c",    # your database name
  host = "192.168.100.61",          # or server IP
  port = 5432,                  # default PostgreSQL port
  user = "postgres",            # your DB username
  password = "ubuntu"    # your DB password
)
```

```r
dbListTables(con)
```

```
[1] "movies"  "ratings"
```

```r
query <- "SELECT m.title, r.rater_name, r.rating
          FROM movies m
          JOIN ratings r ON m.movie_id = r.movie_id
          ORDER BY m.movie_id, r.rater_name;"

data <- dbGetQuery(con, query)
print(data)
```

```
                        title rater_name rating
1    Avatar: The Way of Water      Alice      5
2    Avatar: The Way of Water        Bob      4
3    Avatar: The Way of Water    Charlie      3
4    Avatar: The Way of Water      David      4
5    Avatar: The Way of Water        Eve     NA
6                 Oppenheimer      Alice      4
7                 Oppenheimer        Bob      5
8                 Oppenheimer    Charlie      4
9                 Oppenheimer      David      3
10                Oppenheimer        Eve      4
11                     Barbie      Alice      3
12                     Barbie        Bob     NA
13                     Barbie    Charlie      5
14                     Barbie      David      3
15                     Barbie        Eve      4
```

```
16         Stranger Things S5      Alice      5
17         Stranger Things S5        Bob      5
18         Stranger Things S5    Charlie      4
19         Stranger Things S5      David     NA
20         Stranger Things S5        Eve      5
21                The Marvels      Alice      3
22                The Marvels        Bob      4
23                The Marvels    Charlie      3
24                The Marvels      David      4
25                The Marvels        Eve      3
26 Killers of the Flower Moon      Alice     NA
27 Killers of the Flower Moon        Bob      4
28 Killers of the Flower Moon    Charlie      5
29 Killers of the Flower Moon      David      5
30 Killers of the Flower Moon        Eve      4
```

Option1: Replacing NA with "0"

```r
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
```

```r
library(ggplot2)
data <- data%>%
  mutate(rating = ifelse(is.na(rating), 0, rating))
print(data)
```

```
                        title rater_name rating
1     Avatar: The Way of Water      Alice      5
2     Avatar: The Way of Water        Bob      4
3     Avatar: The Way of Water    Charlie      3
```

```
4    Avatar: The Way of Water      David      4
5    Avatar: The Way of Water        Eve      0
6                 Oppenheimer      Alice      4
7                 Oppenheimer        Bob      5
8                 Oppenheimer    Charlie      4
9                 Oppenheimer      David      3
10                Oppenheimer        Eve      4
11                     Barbie      Alice      3
12                     Barbie        Bob      0
13                     Barbie    Charlie      5
14                     Barbie      David      3
15                     Barbie        Eve      4
16         Stranger Things S5      Alice      5
17         Stranger Things S5        Bob      5
18         Stranger Things S5    Charlie      4
19         Stranger Things S5      David      0
20         Stranger Things S5        Eve      5
21                The Marvels      Alice      3
22                The Marvels        Bob      4
23                The Marvels    Charlie      3
24                The Marvels      David      4
25                The Marvels        Eve      3
26 Killers of the Flower Moon      Alice      0
27 Killers of the Flower Moon        Bob      4
28 Killers of the Flower Moon    Charlie      5
29 Killers of the Flower Moon      David      5
30 Killers of the Flower Moon        Eve      4
```

```r
data %>%
  group_by(rater_name) %>%
  summarise(ratings_count = n())
```

```
# A tibble: 5 x 2
  rater_name ratings_count
  <chr>              <int>
1 Alice                  6
2 Bob                    6
3 Charlie                6
4 David                  6
5 Eve                    6
```

```r
data %>%
  group_by(title) %>%
  summarise(ratings_count = n())
```

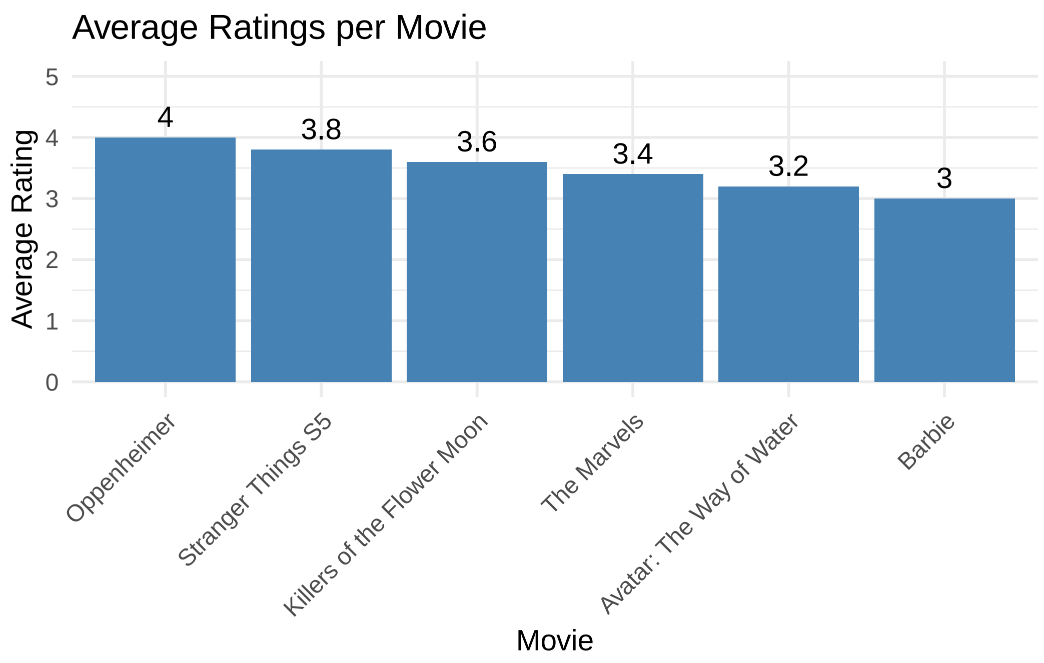
```
# A tibble: 6 x 2
  title                      ratings_count
  <chr>                              <int>
1 Avatar: The Way of Water               5
2 Barbie                                 5
3 Killers of the Flower Moon             5
4 Oppenheimer                            5
5 Stranger Things S5                     5
6 The Marvels                            5
```

Average rating per movie

```r
data %>%
  group_by(title) %>%
  summarise(avg_rating = round(mean(rating, na.rm = TRUE), 2))
```

```
# A tibble: 6 x 2
  title                      avg_rating
  <chr>                           <dbl>
1 Avatar: The Way of Water          3.2
2 Barbie                            3
3 Killers of the Flower Moon        3.6
4 Oppenheimer                       4
5 Stranger Things S5                3.8
6 The Marvels                       3.4
```

```r
avg_ratings <- data %>%
  group_by(title) %>%
  summarise(avg_rating = round(mean(rating, na.rm = TRUE), 2))
  ggplot(avg_ratings, aes(x = reorder(title, -avg_rating), y = avg_rating)) +
  geom_col(fill = "steelblue") +           # create the bars
  geom_text(aes(label = avg_rating), vjust = -0.5) +  # show value on top
  labs(title = "Average Ratings per Movie",
       x = "Movie",
       y = "Average Rating") +
  ylim(0, 5) +                             # rating scale 1-5
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

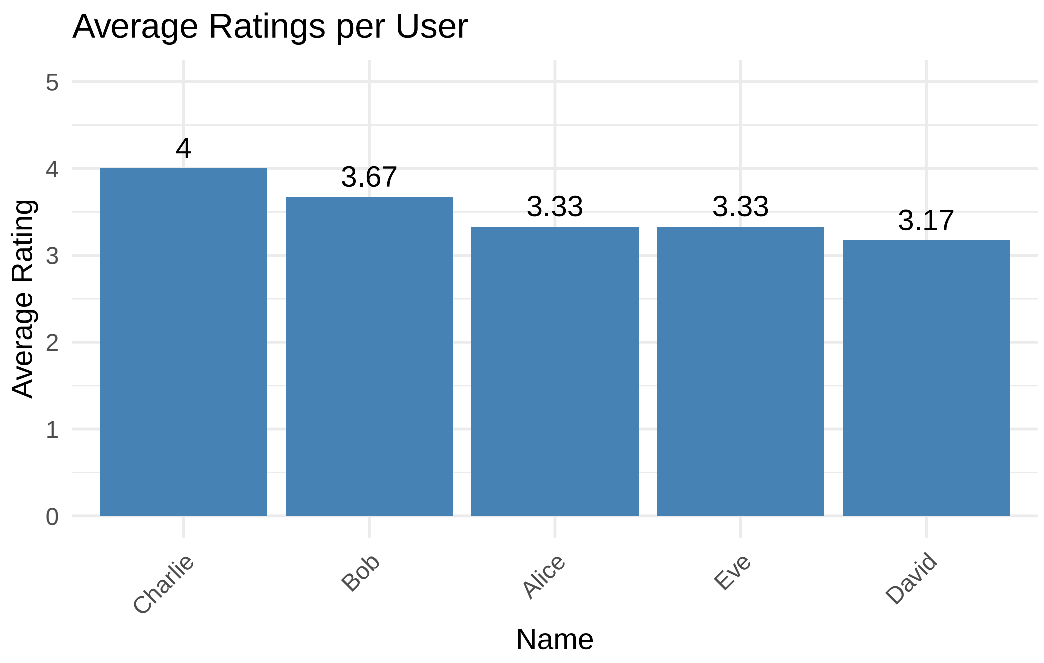


Average rating per user

```
data %>%
  group_by(rater_name) %>%
  summarise(avg_rating = round(mean(rating, na.rm = TRUE), 2))
```

```
# A tibble: 5 x 2
  rater_name avg_rating
  <chr>           <dbl>
1 Alice            3.33
2 Bob              3.67
3 Charlie          4
4 David            3.17
5 Eve              3.33
```

```
avg_ratings <- data %>%
  group_by(rater_name) %>%
  summarise(avg_rating = round(mean(rating, na.rm = TRUE), 2))
  ggplot(avg_ratings, aes(x = reorder(rater_name, -avg_rating), y = avg_rating)) +
  geom_col(fill = "steelblue") +          # create the bars
  geom_text(aes(label = avg_rating), vjust = -0.5) +  # show value on top
  labs(title = "Average Ratings per User",
       x = "Name",
       y = "Average Rating") +
  ylim(0, 5) +                            # rating scale 1-5
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Average Ratings per User
Average Rating
Name
Charlie
Bob
Alice
Eve
David