Step into SQL

Enayetur Raheem

11/15/2022

Table of contents

Preface

Step into SQL is a beginner-friendly book to learn SQL. The goal is to make you feel comfortable working with relational databases for those who never wrote a single line of SQL query ever.

After completing this book, you should be able to write SQL query in your dreams.

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Working with Dates

In all practical problems, we have to manipulate or otherwise work with date data types. So, in this chapter we will build the basic understanding of date data-types from ground up and then learn how to work with them. Finally, we will have many exercises to practice what we learn in this chapter. Let's go.

First let us connect to our Postgres SQL instance in R. If you are working with DBeaver or IDE that directly connects to the PostgreSQL server, then you can ignore the connection settings below. This arrangement is to faciliate writing this book using Quarto tools.

Once connected, we list the available tables in the database. Of them, we will be using the covid19 tables

```
select * from covid19.us_states limit 10
```

Table 2.1: Displaying records 1 - 10

date	state	fips	cases	deaths
2020-01-21	Washington	53	1	0
2020-01-22	Washington	53	1	0
2020-01-23	Washington	53	1	0
2020-01-24	Illinois	17	1	0
2020-01-24	Washington	53	1	0
2020-01-25	California	6	1	0
2020-01-25	Illinois	17	1	0
2020-01-25	Washington	53	1	0
2020-01-26	Arizona	4	1	0
2020-01-26	California	6	2	0

3 Explore Employee table

```
select * from employee limit 3
```

Table 3.1: 3 records

emp_id	emp_name	emp_dept	emp_salary	emp_exp_yr
100	Alex John	HR	70000	10
101	John Doe	HR	73000	4
102	Kabir Uddin	HR	60000	3

3.1 Practical Challenge

Use the COVID dataset to answer the following questions.

3.1.1 Create a new column to the existing table to show the daily deaths for each state

```
select *,
  deaths - lag(deaths) over(partition by state order by date) as daily_deaths
  from covid19.us_states limit 3
```

Table 3.2: 3 records

date	state	fips	cases	deaths	daily_deaths
2020-03-13	Alabama	1	6	0	NA
2020-03-14	Alabama	1	12	0	0
2020-03-15	Alabama	1	23	0	0

3.1.2 Create a table showing the highest deaths on a single day and the

corresponding date for that state

3.1.3 Calculate the number of days it took to reach the highest daily deaths for each state

3.1.4 Rank the states based

on the time it took to reach the highest deaths on a single day for each state Create a table that shows three columns: state name, date of first death (or more), date of 1000th death (or more) Calculate the time it took to reach the 1000th death for each state Rank the states from severest to the least severe in terms of the days it took to reach the 1000th death. Show all states if applicable Bring population data in each state from external sources. Population data would include at least, the total population in the state, and other relevant data Calculate per capita death rates (total deaths divided by total population) per state per month Calculate the overall per capita death rate for each state Rank the states according to the overall per capita death rates (the highest one at the top)

3.2 Learning objective

In Practical Challenge 3.1, we practiced several problems to test our understanding and skills to manipulate tables in a relational data base systems. Is the cross-referencing working as expected?

Reference for 3.1.1

```
dbDisconnect(con)
```

```
[1] TRUE
```

4 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```

References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.