Prep8S24

Sofia Guttmann

2024-04-05

Reminder: Prep assignments are to be completed individually. Upload a final copy of the .Qmd and renamed .pdf to your private repo, and submit the renamed pdf to Gradescope before the deadline (Tuesday night, 4/9/24, by midnight).

Reading

The associated reading for the week is Chapter 15 on SQL.

Practice 9 will contain questions about SQL and next week's content on iteration and simulation (Chapters 7 and 13). There is no Practice 8, as you should be working on your final project proposal.

1 - Chapter Basics

> part a - In your own words, explain what a relational database is, and why using one may be better than using a `flat file`.

Solution:
A relational database is a structured system for storing and managing data organized into tables, with each table made of rows and columns representing individual records and attributes, respectively. The strength of relational databases lies in their ability to establish relationships between tables based on common data elements, enabling efficient data retrieval and manipulation. Compared to flat files, relational databases offer numerous advantages including better data integrity through enforced constraints, scalability to handle large datasets, flexibility in querying and analyzing data using SQL, concurrency control mechanisms for managing multiple users, robust data security features, and the encouragement of data normalization to optimize storage and maintain consistency. These factors collectively make relational databases a superior choice for managing structured data in various applications, ensuring reliability, efficiency, and security throughout the data lifecycle.

> part b - What R package have we been using all semester that was structured to be similar to SQL?

Hint: We have usually not loaded this package directly, but it has been loaded when we load `tidyverse`.

Solution: The R package that has been structured to be similar to SQL and is typically loaded when we load the tidyverse package is **`dplyr`**. **`dplyr`** provides a set of functions for data manipulation that closely resemble SQL operations, such as filtering, sorting, joining, grouping, and summarizing data frames. This package allows for intuitive and efficient data manipulation workflows in R, making it a powerful tool for data analysis and transformation.

> part c - What two arguments are required for a SQL `select` query to run?

Hint: Many arguments can be provided in a `select` query. This is asking about the required two that a `select` query will not run without.

Solution: In a SQL SELECT query, the two required arguments are: Columns and Fields-This argument specifies the columns or fields from which you want to retrieve data. You need to specify at least one column to select data from. Tables specify the table or tables from which you want to retrieve data. You need to specify at least one table where the columns are located.

Without specifying these two arguments, the SELECT query will not be able to identify from which table(s) and which columns to retrieve data, thus it will not run successfully.

> part d - Comparing R and SQL, based on the arguments in the reading, which is better for data analysis? Which is better for data management?

Solution: SQL is typically better suited for data management tasks, especially when dealing with large volumes of structured data. SQL databases are designed for efficient data storage, retrieval, and management, offering features such as indexing, transaction management, concurrency control, and data integrity constraints. SQL is particularly powerful for tasks like data cleaning, data manipulation, and data aggregation, where structured querying and efficient data handling are essential. SQL databases also provide robust security and access control features, making them suitable for managing sensitive data in enterprise environments.

2 - Airline Flights in SQL

Learning SQL requires having a SQL server set up to access. Run the code below to get access to a server with the airline flights data. Then, use the provided code below to get a sense of the data and address a few questions.

```
# SQL commands
con <- dbConnect_scidb("airlines")
```

part a - How many tables are present?

```
query1 <- "SHOW TABLES"

dbGetQuery(con, query1)
```

```
  Tables_in_airlines
1           airports
2           carriers
3            flights
4             planes
```

Solution: There are 4 tables.

part b - What variables are present in the flights data? List some that may be of interest to you to explore.

```
query2 <- "DESCRIBE flights"

dbGetQuery(con, query2)
```

```
            Field        Type Null Key Default Extra
1            year smallint(4)  YES MUL    <NA>
2           month smallint(2)  YES        <NA>
3             day smallint(2)  YES        <NA>
4        dep_time smallint(4)  YES        <NA>
5  sched_dep_time smallint(4)  YES        <NA>
6       dep_delay smallint(4)  YES        <NA>
7        arr_time smallint(4)  YES        <NA>
8  sched_arr_time smallint(4)  YES        <NA>
9       arr_delay smallint(4)  YES        <NA>
```

```
10        carrier  varchar(2)   NO MUL
11        tailnum  varchar(6)  YES MUL    <NA>
12         flight smallint(4)  YES        <NA>
13         origin  varchar(3)   NO MUL
14           dest  varchar(3)   NO MUL
15       air_time smallint(4)  YES        <NA>
16       distance smallint(4)  YES        <NA>
17      cancelled  tinyint(1)  YES        <NA>
18       diverted  tinyint(1)  YES        <NA>
19           hour smallint(2)  YES        <NA>
20         minute smallint(2)  YES        <NA>
21      time_hour    datetime  YES        <NA>
```

Solution: Some variables included are year, month, day, dep_time, dep_delay, arr_time, arr_delay, carrier.

part c - How many flights went from Hartford (BDL) to Chicago (ORD) in 2014?

```
query3 <- "SELECT COUNT(*) as N
  FROM flights
  WHERE dest = 'ORD' AND year = 2014 AND origin = 'BDL'
"

dbGetQuery(con, query3)
```

```
     N
1 1690
```

Solution: 1690 flights went from Hartford to Chicago in 2014.

part d - Your turn! How many flights went from Chicago to Hartford in 2014?

Solution: 1623

```
query3 <- "SELECT COUNT(*) as N
  FROM flights
  WHERE dest = 'BDL' AND year = 2014 AND origin = 'ORD'
"

dbGetQuery(con, query3)
```

```
     N
1 1623
```

part e - Use more date info. How many domestic flights flew into Portland, Oregon (PDX) on May 14, 2014?

Solution: 0

```
query3 <- "SELECT COUNT(*) as N
  FROM flights
  WHERE dest = 'PDX' AND year = 2014 AND origin = 'BDL'
"

dbGetQuery(con, query3)
```

```
  N
1 0
```

part f - Design your own query. You can continue pulling from flights or use another table. Explain what you wanted the query to show (i.e. what question is it helping to answer?) and then provide an answer.

Solution: ORD to PDX? 2507

```
query3 <- "SELECT COUNT(*) as N
  FROM flights
  WHERE dest = 'PDX' AND year = 2014 AND origin = 'ORD'
"

dbGetQuery(con, query3)
```

```
     N
1 2507
```