Final Assignment - Database Administration - Part 3
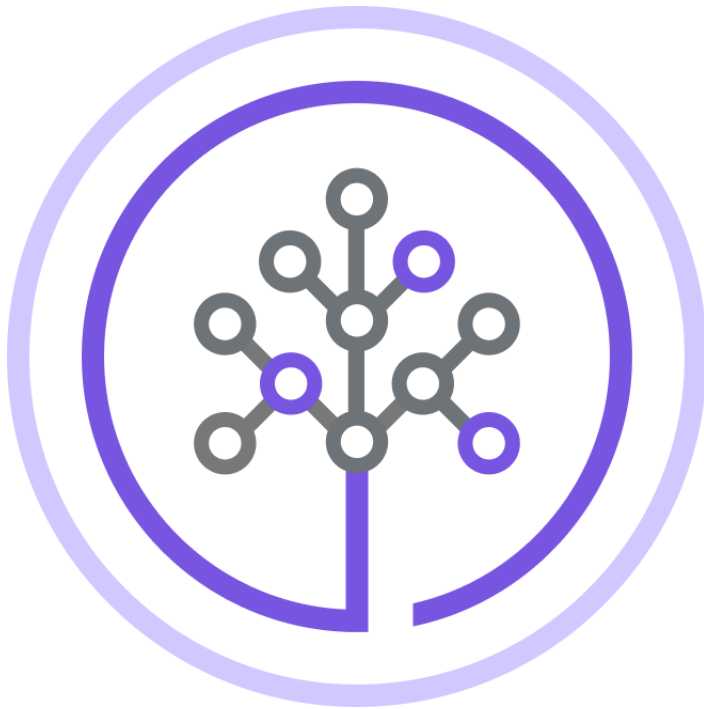

Estimated time needed: **45 minutes**.

Software Used in this Lab

In this lab, you will use Datasette, an open-source multi-tool for exploring and publishing data.

After launching the lab, you will see a view like the one below, with the Datasette Interface embedded within it.

Scenario

You have been assigned the work on Datasette and perform the tasks like restoration of data and index creation to improve the query performance. You will create views to make queries easier to write.

Objectives

In part 3 of this assignment, you will be working on the following aspects of Database Administration.

- Restore data
- Indexing
- View creation

Note - Screenshots

Throughout this lab, you will be prompted to take screenshots and save them on your device. These screenshots need to be uploaded for peer review in the next section of the course. You can use various free screen-grabbing tools to do this or use your operating system's shortcut keys to do this (for example, Alt+PrintScreen in Windows).

Dataset Used in this Lab

The dataset used in this lab is billing.csv.

Exercise 3.1 - Restore data

Task 3.1 - Restore the table billing

Use the billing.csv and restore the CSV file into a table named billing. Write a query to display the total number of rows imported.

Take a screenshot of the SQL statements you used and the output.

Name the screenshot as **restore-table.jpg**. (images can be saved with either .jpg or .png extension)

▶ click here for hint

Exercise 3.2 - Create a view

Task 3.2 - Create a view named basicbilldetails with the columns customerid, month, billedamount

Take a screenshot of the SQL statement used to create the view.

Name the screenshot as **create-view.jpg**. (images can be saved with either .jpg or .png extension)

Exercise 3.3 - Indexing

Task 3.3 - Baseline query performance

Write a query to find out all the rows with a billing amount of 19929.

Hint: Use the command SELECT strftime("%Y-%m-%d %H:%M:%f", "now"); before and after your query to display the run time.

Note: There will not be much difference in the run time as the number of records is less for the above-given condition.

Take a screenshot of the command you used, along with the query run time.

Name the screenshot as **query-base-line.jpg**. (images can be saved with either .jpg or .png extension). If you find it difficult to capture the entire screen in a single screenshot, please capture only the results section showing the commands you used and the output.

Task 3.4 - Create an index

Create an index that can make the query in the previous task faster. Name the index as billingamount.

Take a screenshot of the SQL statement you used and the output.

Name the screenshot as **index-creation.jpg**. (images can be saved with either .jpg or .png extension)

Task 3.5 - Document the improvement in query performance

Find out if the index has any impact on query performance.

Re-run the query to find out all the rows with a billing amount of 19929.

Take a screenshot of the command you used, the output, and the query run time.

You will NOT be evaluated on the query run time. However, you are encouraged to run the query multiple times and pick the result with the lowest query run time.

Name the screenshot as **query-after-index.jpg**. (images can be saved with either .jpg or .png extension). If you find it difficult to capture the entire screen in a single screenshot, please capture only the results section showing the commands you used and the output.

End of assignment - Part 3.

Author(s)

Ramesh Sannareddy

Other Contributor(s)

Rav ahuja

Changelog

Date	Version	Changed by	Change Description
2024-05-30	0.5	Akansha Yadav	Recreated and updated instruction
2023-05-11	0.4	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-04	0.3	Eric Hao & Vladislav Boyko	Republished
2022-06-15	0.2	Appalabhaktula Hema	Converted initial version to Datasette
2021-10-21	0.1	Ramesh Sannareddy	Created initial version