Module 22 Challenge

Start Assignment

Due Sep 25 by 11:59pm **Points** 100
Submitting a text entry box or a website url

In this challenge, you'll use your knowledge of SparkSQL to determine key metrics about home sales data. Then you'll use Spark to create temporary views, partition the data, cache and uncache a temporary table, and verify that the table has been uncached.

Before You Begin

1. Create a new repository for this project called, `Home_Sales`. **Do not add this homework to an existing repository**.
2. Clone the new repository to your computer.
3. Push your changes to GitHub.

Files

Download the following files to help you get started:

Module 22 Challenge files

Instructions

1. Rename the `Home_Sales_starter_code.ipynb` file as `Home_Sales.ipynb`.
2. Import the necessary PySpark SQL functions for this assignment.
3. Read the `home_sales_revised.csv` data in the starter code into a Spark DataFrame.
4. Create a temporary table called `home_sales`.
5. Answer the following questions using SparkSQL:
   - What is the average price for a four-bedroom house sold for each year? Round off your answer to two decimal places.
   - What is the average price of a home for each year it was built that has three bedrooms and three bathrooms? Round off your answer to two decimal places.
   - What is the average price of a home for each year that has three bedrooms, three bathrooms, two floors, and is greater than or equal to 2,000 square feet? Round off your answer to two decimal places.
   - What is the "view" rating for homes costing more than or equal to $350,000? Determine the run time for this query, and round off your answer to two decimal places.
6. Cache your temporary table `home_sales`.
7. Check if your temporary table is cached.
8. Using the cached data, run the query that filters out the view ratings with an average price of greater than or equal to $350,000. Determine the runtime and compare it to uncached runtime.
9. Partition by the "date_built" field on the formatted parquet home sales data.
10. Create a temporary table for the parquet data.
11. Run the query that filters out the view ratings with an average price of greater than or equal to $350,000. Determine the runtime and compare it to uncached runtime.
12. Uncache the `home_sales` temporary table.
13. Verify that the `home_sales` temporary table is uncached using PySpark.
14. Download your `Home_Sales.ipynb` file and upload it into your "Home_Sales" GitHub repository.

Support and Resources

Your instructional team will provide support during classes and office hours. You will also have access to learning assistants and tutors to help you with topics as needed. Make sure to take advantage of these resources as you collaborate with your partner on this project.

Requirements

1. A Spark DataFrame is created from the dataset. (5 points)
2. A temporary table of the original DataFrame is created. (10 points)
3. A query is written that returns the average price, rounded to two decimal places, for a four-bedroom house that was sold in each year. (5 points)
4. A query is written that returns the average price, rounded to two decimal places, of a home that has three bedrooms and three bathrooms. (5 points)
5. A query is written that returns the average price of a home with three bedrooms, three bathrooms, two floors, and is greater than or equal to 2,000 square feet for each year built rounded to two decimal places. (5 points)
6. A query is written that returns the view rating for the average price for homes that are greater than or equal to $350,000, rounded to two decimal places. (The output shows the run time for this query.) (10 points)
7. A cache of the temporary "home_sales" table is created and validated. (10 points)
8. The query from step 6 is run on the cached temporary table, and the run time is computed. (10 points)
9. A partition of the home sales dataset by the "date_built" field is created, and the formatted parquet data is read. (10 points)
10. A temporary table of the parquet data is created. (10 points)
11. The query from step 6 is run on the parquet temporary table, and the run time is computed. (10 points)
12. The "home_sales" temporary table is uncached and verified. (10 points)

This project will be evaluated against the requirements and assigned a grade according to the following table:

Grade	Points
A (+/-)	90+
B (+/-)	80–89
C (+/-)	70–79
D (+/-)	60–69
F (+/-)	< 60

Submission

You are required to submit the URL of your GitHub repository for grading.

IMPORTANT

> **It is your responsibility to include a note in the README section of your repo specifying code source and its location within your repo**. This applies if you have worked with a peer on an assignment, used code in which you did not author or create sourced from a forum such as Stack Overflow, or you received code outside curriculum content from support staff such as an Instructor, TA, Tutor, or Learning Assistant. This will provide visibility to grading staff of your circumstance in order to avoid flagging your work as plagiarized.
>
> If you are struggling with a Challenge or any aspect of the curriculum, please remember that there are student support services available for you:
>
> 1. Office hours facilitated by your TA(s)
> 2. Tutor sessions (sign up)
> 3. Ask the class Slack channel/get peer support
> 4. AskBCS Learning Assistants

References

Data for this dataset was generated by edX Boot Camps LLC, and is intended for educational purposes only.