**Analysis Report: Home Sales Data**

**Introduction:**

This report details the analysis performed on home sales data using SparkSQL, conducted on Google Colab due to local installation issues. The primary objectives were to compute key metrics related to home sales, optimize query performance, and verify the functionality of various Spark features such as caching and partitioning.

**Methodology:**

1. **Data Preparation**:
   * **Data Import**: The home\_sales\_revised.csv file was read into a Spark DataFrame.
   * **Temporary Table Creation**: A temporary table named home\_sales was created from the DataFrame for SQL operations.
2. **Queries Execution**:
   * Queries were formulated to determine:
     1. The average price of four-bedroom houses sold per year.
     2. The average price of homes with three bedrooms and three bathrooms for each year the home was built.
     3. The average price of homes meeting additional criteria (three bedrooms, three bathrooms, two floors, and ≥ 2,000 square feet) by year built.
     4. The average price of homes per view rating, filtered to include only those with an average price of at least $350,000.
3. **Performance Optimization**:
   * **Caching**: The home\_sales table was cached to improve query performance.
   * **Runtime Comparison**: The runtime of the query for average home price per view rating was compared between cached and uncached data.
   * **Partitioning**: The dataset was partitioned by the date\_built field, and the parquet format was used to further enhance performance.
   * **Query Execution on Parquet Data**: The query was run on the parquet-formatted data, and runtime was compared with cached and uncached queries.
4. **Validation**:
   * Verified the caching status of the home\_sales table.
   * Ensured that the table was successfully uncached and that all results were as expected.

**Analysis:**

**1. Average Price for Four-Bedroom Houses (Per Year)**

* The average prices for four-bedroom houses were computed for each year:
  + 2019: $300,263.70
  + 2020: $298,353.78
  + 2021: $301,819.44
  + 2022: $296,363.88

The results show minor fluctuations in average prices over the years, with 2021 experiencing the highest average price.

**2. Average Price of Homes with 3 Bedrooms and 3 Bathrooms (By Year Built)**

* The average prices for homes with three bedrooms and three bathrooms were:
  + 2010: $292,859.62
  + 2011: $291,117.47
  + 2012: $293,683.19
  + 2013: $295,962.27
  + 2014: $290,852.27
  + 2015: $288,770.30
  + 2016: $290,555.07
  + 2017: $292,676.79

The average prices were relatively stable across the years, with a slight increase in later years.

**3. Average Price of Homes with Specific Conditions (By Year Built)**

* For homes with three bedrooms, three bathrooms, two floors, and ≥ 2,000 square feet:
  + 2010: $285,010.22
  + 2011: $276,553.81
  + 2012: $307,539.97
  + 2013: $303,676.79
  + 2014: $298,264.72
  + 2015: $297,609.97
  + 2016: $293,965.10
  + 2017: $280,317.58

There was notable variability in prices, with peaks in 2012 and 2013, indicating market changes over time.

**4. Average Price per 'View' Rating**

* For homes with an average price ≥ $350,000:
  + View ratings and average prices were recorded, showing a strong correlation between view ratings and home prices, with higher ratings associated with higher average prices.

**5. Performance Comparison**

* **Cached vs. Uncached**:
  + The query runtime on cached data was 0.568 seconds.
  + The runtime on uncached data was 1.828 seconds.

Caching improved query performance significantly.

* **Parquet Data**:
  + Query runtime on parquet data was 0.725 seconds, which was faster than uncached but slower than cached.

**Conclusion:**

The assignment objectives were successfully achieved through the following:

* **Data Processing**: Accurate creation and manipulation of Spark DataFrames and temporary tables.
* **Query Execution**: Effective extraction of key metrics related to home prices.
* **Performance Optimization**: Successful implementation of caching and partitioning techniques to enhance query performance.
* **Verification**: Ensured correct implementation of caching, uncaching, and runtime comparisons.

The use of Google Colab was instrumental in overcoming local installation issues, allowing for the successful completion of the analysis. All university requirements were met, including data processing, query execution, performance optimization, and verification of results.