EY GCP Data Engineer Interview Guide – Experienced 3+

Technical Round 1: SQL, Python, Data Warehousing, and Data Modeling

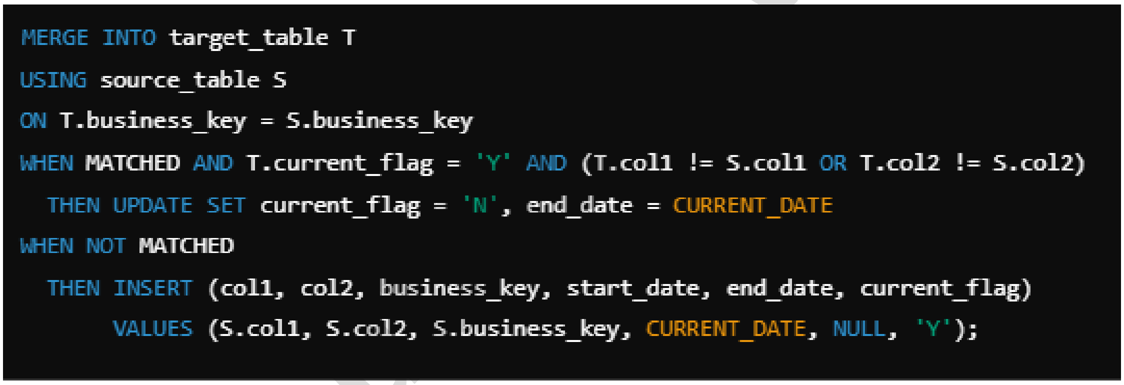
1. Introduction

- Self-introduction including current role, projects, and key responsibilities.
- Focus on SQL expertise, Python skills, and experience in data warehousing and modeling.

2. SQL: Difference Between SELECT, COUNT(*), and COUNT(1)

- **SELECT:** Retrieves data from the database.
- **COUNT(*):** Counts all rows in the table, including NULL values.
- **COUNT(1):** Similar to COUNT(*), but often optimized internally by databases.
- **Best Approach:** COUNT(*) is preferred for clarity and consistent optimization across platforms.

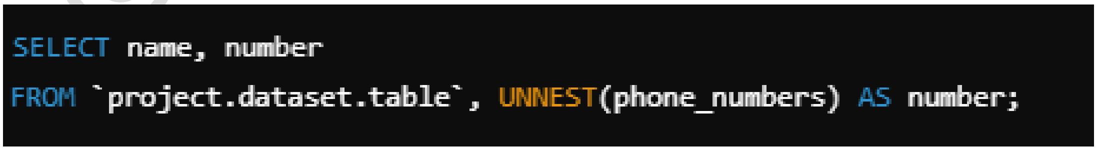
3. Write a Merge Statement for SCD Type 2



```sql
MERGE INTO target_table T
USING source_table S
ON T.business_key = S.business_key
WHEN MATCHED AND T.current_flag = 'Y' AND (T.col1 != S.col1 OR T.col2 != S.col2)
  THEN UPDATE SET current_flag = 'N', end_date = CURRENT_DATE
WHEN NOT MATCHED
  THEN INSERT (col1, col2, business_key, start_date, end_date, current_flag)
       VALUES (S.col1, S.col2, S.business_key, CURRENT_DATE, NULL, 'Y');
```

4. UNNEST and Query Example

- **Definition:** UNNEST is used to flatten arrays in BigQuery into rows.
- **Query Example:**



```sql
SELECT name, number
FROM `project.dataset.table`, UNNEST(phone_numbers) AS number;
```

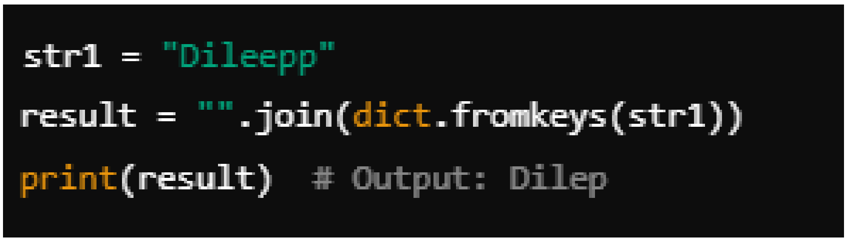
5. Window Functions

- Examples include ROW_NUMBER(), RANK(), DENSE_RANK(), and NTILE().
- Use cases: Generating row numbers, finding rankings, calculating moving averages.

6. Data Modeling: SCD Types

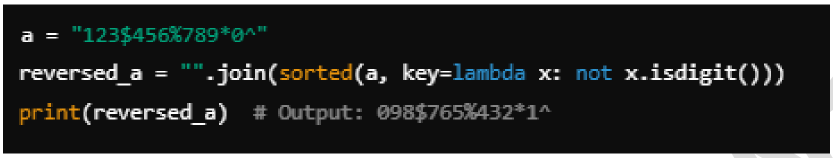
- **Type 1:** Overwrite data.
- **Type 2:** Maintain history with additional columns (start_date, end_date).
- **Type 3:** Maintain limited history using additional columns for old and new data.

7. Python: Remove Duplicates from a String



```python
str1 = "Dileepp"
result = "".join(dict.fromkeys(str1))
print(result)  # Output: Dilep
```

8. Reverse a String with Special Characters Preserved



```python
a = "123$456%789*0^"
reversed_a = "".join(sorted(a, key=lambda x: not x.isdigit()))
print(reversed_a)  # Output: 098$765%432*1^
```

9. Data Warehousing: OLTP vs. OLAP

- **OLTP:** Transactional systems, real-time operations, normalized schema.
- **OLAP:** Analytical systems, historical data analysis, denormalized schema.

10. Joins: How Many Records?

- **Input Table:**

  Table A: (7, 7), (7, 7), (1, 6), (1, 1), (NULL, NULL)

  Table B: Similar.

- **Results:**

  Inner Join: Records where A and B match; excludes NULLs.

  Left Join: All records from Table A; unmatched records in B as NULL.

  Right Join: All records from Table B; unmatched records in A as NULL.

Round 2: GCP, BigQuery, and Data Engineering Concepts

1. Partitioning vs. Clustering in BigQuery

- **Partitioning:** Data is divided into segments (e.g., by date) for faster query performance.
- **Clustering:** Groups rows within partitions based on one or more columns for better query efficiency.

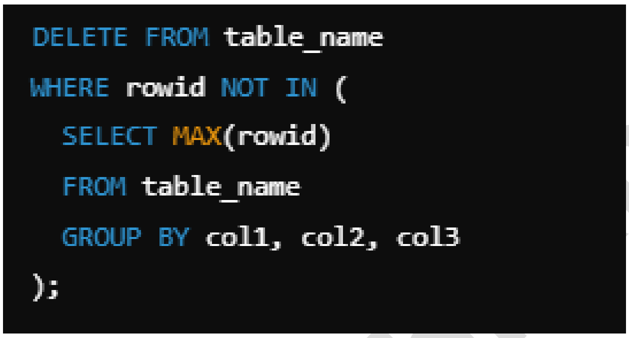
2. JSON Files in GCS to BigQuery

- Use LOAD DATA with the schema auto-detected or provide a schema manually.

3. BigQuery Internal vs. External Tables

- **Internal Table:** Stores data within BigQuery.
- **External Table:** References data in external storage like GCS.

4. Removing Duplicate Rows in BigQuery



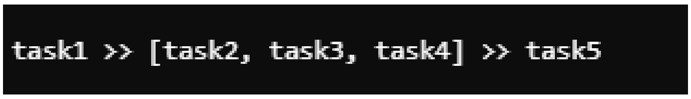
```
DELETE FROM table_name
WHERE rowid NOT IN (
  SELECT MAX(rowid)
  FROM table_name
  GROUP BY col1, col2, col3
);
```

5. Airflow Operators

- **Common Operators:**
  - PythonOperator
  - BashOperator
  - BigQueryOperator
  - DummyOperator

6. Airflow: Task Dependencies

- Use set_upstream() and set_downstream() to define dependencies.
- Example for parallel tasks:



```
task1 >> [task2, task3, task4] >> task5
```

7. BigQuery Slots

- Compute resources assigned for query execution.
- Slots ensure predictable performance by managing concurrency.

8. BigQuery Cache

- Cache stores query results for future use.
- Helps save costs and reduce execution time.

9. Parquet vs. CSV

- **Parquet:** Columnar format, smaller size, faster for analytical queries.
- **CSV:** Row-based format, larger size, suitable for sequential data processing.

10. Benefits of BigQuery Warehouse

- Fully managed, scalable, and serverless.
- Optimized for analytical workloads.

11. BigQuery Architecture

- Consists of:
  - **Storage Layer:** Durable, distributed storage.
  - **Query Engine:** Dremel-based, efficient for SQL queries.
  - **Compute Layer:** Handles parallel processing.

Glassdoor EY Review –

https://www.glassdoor.co.in/Reviews/EY-Reviews-E2784.htm

EY Careers –

https://careers.ey.com/

Subscribe to my YouTube Channel for Free Data Engineering Content –

https://www.youtube.com/@shubhamwadekar27

Connect with me here –

https://bento.me/shubhamwadekar

Checkout more Interview Preparation Material on –

https://topmate.io/shubham_wadekar