**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)**

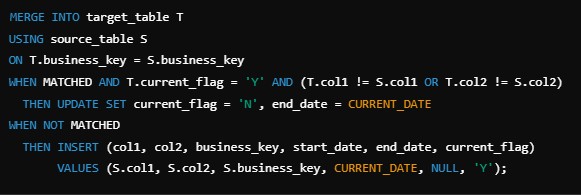
 **SELECT:** Retrieves data from the database.

 **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**



**4. UNNEST and Query Example**

 **Definition:** UNNEST is used to flatten arrays in BigQuery into rows.

 **Query Example:**



**5. Window Functions**

 Examples include ROW\_NUMBER(), RANK(), DENSE\_RANK(), and NTILE().

 Use cases: Generating row numbers, finding rankings, calculating moving averages.

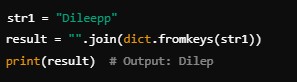
**6. Data Modeling: SCD Types**

 **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**



**8. Reverse a String with Special Characters Preserved**



**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**

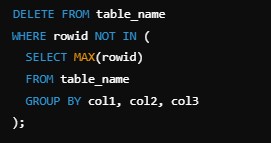
 Use LOAD DATA with the schema auto-detected or provide a schema manually.

**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**



**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:



**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.