

BigQuery Practical Mastery – 90 Pure Practical Questions

This practice set is designed for Data Analytics professionals. There is NO theory. Solve each question practically in BigQuery. Completing all questions ensures strong BigQuery mastery.

- 1 Select only required columns (no SELECT *) and verify scanned data size.
- 2 Filter records using a DATE range on a TIMESTAMP column.
- 3 Find top 10 records per category using ORDER BY and LIMIT.
- 4 Calculate total, average, minimum, and maximum values.
- 5 Group data by multiple columns and apply HAVING conditions.
- 6 Use CASE WHEN to create value buckets.
- 7 Remove duplicate records using window functions.
- 8 Replace NULL values safely in calculations.
- 9 Join two large tables and calculate aggregated metrics.
- 10 Identify records present in table A but missing in table B.
- 11 Use LEFT JOIN and find unmatched rows.
- 12 Optimize JOIN queries by selecting only required columns.
- 13 Perform a self-join for time-based comparison.
- 14 Handle duplicate keys correctly during joins.
- 15 Rank records within each category.
- 16 Find the latest record per group using window functions.
- 17 Calculate running totals over time.
- 18 Calculate 7-day moving averages.
- 19 Compare current vs previous values using LAG.
- 20 Filter window function results using QUALIFY.
- 21 Find top N records per group without subqueries.
- 22 Rewrite a subquery using QUALIFY.
- 23 Use SAFE_DIVIDE to avoid division errors.
- 24 Use SAFE_CAST to convert data types safely.
- 25 Parse string dates into DATE format.
- 26 Extract year, month, day from TIMESTAMP.
- 27 Format TIMESTAMP into readable string.
- 28 Handle invalid data without query failure.
- 29 Query tables containing ARRAY fields.
- 30 UNNEST arrays and join back to parent rows.
- 31 Count elements inside ARRAY columns.
- 32 Filter records based on ARRAY contents.
- 33 Create STRUCT fields in query output.
- 34 Access nested STRUCT fields.
- 35 Aggregate values inside ARRAYS.
- 36 Rebuild ARRAYS after transformations.
- 37 Extract values from JSON strings.
- 38 Filter records using JSON attributes.
- 39 Convert JSON fields into STRUCT format.
- 40 Safely handle missing JSON keys.
- 41 Parse nested JSON arrays.
- 42 Combine JSON extraction with UNNEST.
- 43 Compare query cost between SELECT * and selective columns.
- 44 Rewrite inefficient queries to reduce scanned data.
- 45 Use partition filters correctly in WHERE clause.
- 46 Identify most expensive queries from job history.
- 47 Analyze query execution plan.
- 48 Detect and fix accidental cross joins.
- 49 Optimize queries using clustering columns.
- 50 Estimate query cost before execution.
- 51 Create partitioned tables using DATE column.
- 52 Query only required partitions.
- 53 Create clustered tables.
- 54 Compare clustered vs non-clustered performance.
- 55 Design partition strategy for large tables.
- 56 Rewrite queries to enable partition pruning.

- 57 Load CSV files into BigQuery tables.
- 58 Manually define schema vs auto-detect.
- 59 Append new data into existing tables.
- 60 Overwrite table data safely.
- 61 Create external tables from Cloud Storage.
- 62 Query external tables efficiently.
- 63 Export query results to Cloud Storage.
- 64 Create daily scheduled queries.
- 65 Store aggregated results in reporting tables.
- 66 Prevent duplicate records in scheduled queries.
- 67 Build incremental data loading logic.
- 68 Monitor and debug scheduled query failures.
- 69 Design idempotent scheduled pipelines.
- 70 Create a linear regression model using BigQuery ML.
- 71 Train a classification model using SQL.
- 72 Evaluate ML model performance metrics.
- 73 Generate predictions using trained models.
- 74 Store ML predictions in tables.
- 75 Use BigQuery ML for forecasting use cases.
- 76 Design optimized reporting tables for BI tools.
- 77 Build star schema using BigQuery.
- 78 Optimize BigQuery tables for Power BI/Tableau.
- 79 Create KPI summary tables.
- 80 Implement row-level security using SQL.
- 81 Detect anomalies using SQL logic.
- 82 Build alert-triggering metric queries.
- 83 Rewrite an expensive query to cut cost by 50%.
- 84 Explain and fix an inefficient execution plan.
- 85 Design schema for 1 billion row table.
- 86 Choose partition vs clustering for given scenario.
- 87 Handle late-arriving data efficiently.
- 88 Optimize dashboard queries for fast refresh.
- 89 Migrate traditional SQL to BigQuery-optimized SQL.