**Part 1: Short Answers (use 3-5 lines to answer the following questions)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Google BigQuery | Amazon Redshift | Hive |
| Description | Large scale data warehouse service with append-only tables | Large scale data warehouse service for use with business intelligence tools | Data warehouse software for querying and managing large distributed datasets, built on Hadoop |
| License | Commercial | Commercial | Open source |
| Cloud-based/OS | Yes/Hosted | Yes/Hosted | No  All OS with a Java VM |
| SQL | Yes | Yes | SQL Like |
| Partitioning methods | None | Sharding | Sharding |
| MapReduce | No | No | Yes |
| User concept | Access privileges (owner, writer, reader) for whole datasets, not for individual tables | Fine grained access rights according to SQL-standard | Access rights for users, groups and roles |

**Problem 1:** Why Google BigQuery is a fast analytics database?

Google BigQuery runs fast SQL queries on gigabytes to petabytes of data and makes it easy to join public or commercial datasets. BigQuery does not have the concept of machines or a cluster like HDInsight. There is no hardware configuration or administration that needs to be done. BigQuery is a massively parallel processing column store technology.

**Problem 2:** List at least two major differences between Hive and BigQuery?

Google BigQuery requires commercial license. Access privileges (owner, writer, reader) for whole datasets, not for individual tables. Immediate consistency.

Hive is open source. Access rights for users, groups and roles. Eventual consistency.

**Problem 3:** List at least two major differences between Redshift and BigQuery?

Redshift is built on massive parallel processing (MPP). Charged by the hour for each instance you spin up. Requires user to allocate different instance types and create clusters on thier own.

Hadoop is used for massive storage and batch processing of data. Hadoop is very mature and popular. Hadoop is not suitable for real-time querying and data analysis. Hardware is completely abstracted and not visible, even as virtualized units. Charged by storage, streaming inserts and queries

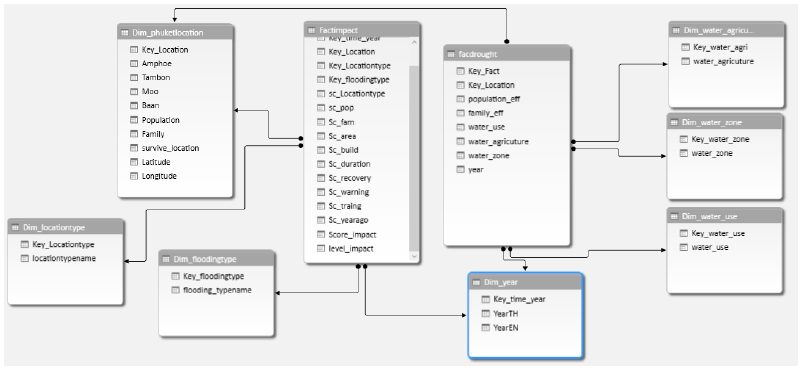
**Problem 4:** List at least two major differences between Hadoop and BigQuery?

Hadoop is an open-source framework. All common hardware failures will automatically be handled by the framework. Wide-column store based.

BigQuery does not automatically offer the ability to drill into your data. BigQuery is fully managed, has quick response time, low-cost, high scalability and can read and write data easily via Cloud Dataflow and Hadoop. Large scale data warehouse service with append-only tables.

**Problem 5:** Suppose you are planning a data warehousing, you have some different solutions to choose from, such as Google BigQuery and Amazon Redshift. Briefly describe your data warehouse, the specific solution you would choose, and the reasons behind why you would choose the specific solution.

I was tasked to design a data warehouse for research team at a public university. The data warehouse will contain various fact and dimension tables. Redshift would require the team to spend to much time optimizing data for fast queries BigQuery offers the flexibly for the research team to have a functional data warehouse that works without spending too much time tuning the database. The research team does not require extremely fast query response times as some of the data is rarely queried. BigQuery offers the best option so the research can focus on their research and not on managing the data warehouse.



**Part 2: Problem 6:** Study BigQuery and Hive syntax and do a side-by-side comparison to SQL and Hive’s query language. See comparison table below. You may research your own (be sure to include the reference) or use the following the links:

BigQuery: <https://cloud.google.com/bigquery/docs/reference/legacy-sql>

Hive: <https://docs.treasuredata.com/articles/hive>

|  |  |  |
| --- | --- | --- |
| Function | SQL | HIVE |
| Selecting a database | USE database; | USE database; |
| Listing databases | SHOW database ; | SHOW database ; |

Complete the following comparison table (the first row is already filled as an example):

**Retrieving Information:**

|  |  |  |
| --- | --- | --- |
| Function | SQL | HIVE |
| Retrieving Information (General) | SELECT columns  FROM table  WHERE conditions; | SELECT columns  FROM table  WHERE conditions; |
| Retrieving All Values | SELECT \*  FROM table; | SELECT \*  FROM table; |
| Retrieving Some Values | SELECT \*  FROM table  WHERE column\_name ="value"; | SELECT \*  FROM table  WHERE column\_name ="value"; |
| Retrieving With Multiple Criteria | SELECT \*  FROM table  WHERE column1\_name  ="value1" AND column2\_name  ="value2"; | SELECT \*  FROM table  WHERE column1\_name  ="value1" AND column2\_name  ="value2"; |
| Retrieving Specific Columns | SELECT column\_name  FROM table; | SELECT column\_name  FROM table; |
| Retrieving Unique Output | SELECT DISTINCT column\_name FROM table; | SELECT DISTINCT column\_name FROM table; |
| Sorting | SELECT col1, col2  FROM table  ORDER BY col1 ASC; | SELECT col1, col2  FROM table  ORDER BY col1 ASC; |
| Sorting Reverse | SELECT col1, col2  FROM table  ORDER BY col1 DESC; | SELECT col1, col2  FROM table  ORDER BY col1 DESC; |
| Counting Rows | SELECT COUNT(\*)  FROM table; | SELECT COUNT(\*)  FROM table; |
| Grouping With Counting | SELECT col\_name, COUNT(\*)  FROM table  GROUP BY col\_name; | SELECT col\_name, COUNT(\*)  FROM table  GROUP BY col\_name; |
| Maximum Value | SELECT MAX(col\_name) AS label FROM table; | SELECT MAX(col\_name) AS label FROM table; |
| Selecting from multiple tables (Join same table using alias w/”AS”) | SELECT t1.col\_name  FROM table1 AS t1 JOIN table2 AS t2 ON (t1. col\_name = t2.col\_name) | SELECT t1.col\_name  FROM table1 AS t1 JOIN table2 AS t2 ON (t1. col\_name = t2.col\_name) |

**Metadata:**

|  |  |  |
| --- | --- | --- |
| Listing tables in a database | USE database; | USE database; |
| Describing the format of a table | SHOW DATABASES; | SHOW DATABASES; |
| Creating a database | CREATE DATABASE db\_name; | CREATE DATABASE db\_name; |
| Dropping a database | DROP DATABASE db\_name; | DROP DATABASE db\_name (CASCADE); |