Persistent | Google Cloud

Teradata to BigQuery Migration: Overview

Overview

You can use the BigQuery Data Transfer Service in combination with a special migration agent to copy your schema and data from Teradata to BigQuery. The migration agent connects to your local data warehouse communicates with the BigQuery Data Transfer Service to copy tables from your data warehouse to BigQuery.

The following steps describe the workflow for the migration process:

1.	2.	3.	4.
Download the migration agent.	Configure a transfer in the BigQuery Data Transfer Service.	Run the transfer job to copy table schema and data from your data warehouse to BigQuery.	Optional. Monitor transfer jobs by using the Google Cloud console.

Transfer Job Configuration

You can configure a transfer job to best suit your needs. Before setting up a data transfer from Teradata to BigQuery, consider the configuration options described in the following sections and decide what settings to use. Depending on the settings you choose, you might need to complete some prerequisites prior to starting the transfer job.

For most systems, especially those with large tables, you can get the best performance by following these steps:

1.	2.	3.
Partition your Teradata tables.	Use Teradata Parallel Transporter (TPT) for extraction.	Create a custom schema file and configure your target BigQuery clustering and partitioning columns.

This enables the migration agent to perform partition-by-partition extraction, which is the most efficient.

© 2025 Persistent Systems Ltd. All rights reserved.

Extraction Method

The BigQuery Data Transfer Service supports two extraction methods for transferring data from Teradata to BigQuery:

Use the Teradata Parallel Transporter (TPT) tbuild utility. This is the recommended approach. Using TPT typically results in faster data extraction.

In this mode, the migration agent attempts to calculate extraction batches using rows distributed by partitions. For each batch, the agent emits and executes a TPT extraction script, producing a set of pipe delimited files. It then uploads these files to a Cloud Storage bucket, where they are used by the transfer job. Once the files are uploaded to Cloud Storage, the migration agent deleted them from the local file system.

When you use TPT extraction **without** a partitioning column, your whole table is extracted. When you use TPT extraction **with** a partitioning column, the agent extracts sets of partitions.

In this mode, the migration agent doesn't limit the amount of space that the extracted files take up on the local file system. Make sure the local file system has more space than the size of your largest partition or your largest table, depending on whether you are specifying a partitioning column or not.

Extraction using a JDBC driver with FastExport connection. If there are constraints on the local storage space available for extracted files, or if there is some reason you can't use TPT, then use this extraction method.

In this mode, the migration agent extracts tables into a collection of AVRO files on the local file system. It then uploads these files to a Cloud Storage bucket, where they are used by the transfer job. Once the files are uploaded to Cloud Storage, the migration agent deletes them from the local file system.

In this mode, you can limit the amount of space used by the AVRO files on the local file system. If this limit is exceeded, extraction is paused until space is freed up by the migration agent uploading and deleting existing AVRO files.

Schema Identification

The BigQuery Data Transfer Service provides automatic schema detection and data type mapping during a data transfer from Teradata to BigQuery. Optionally, you can specify a custom schema file instead. We recommend schema customization in the following situations:

- You need to capture important information about a table, like partitioning, that would otherwise be lost in the migration. For example, incremental transfers should have a schema file specified so that data from subsequent transfers can be properly partitioned when loaded into BigQuery. Without a schema file specified, every time a transfer runs, the BigQuery Data Transfer Service automatically applies a table schema by using the source data being transferred, and all information about partitioning, clustering, primary keys and change tracking is lost.
- You need to change column names or data types during the data transfer.

© 2025 Persistent Systems Ltd. All rights reserved.

Custom Schema File

A custom schema file is a JSON file that describes database objects. The schema contains a set of databases, each containing a set of tables, each of which contains a set of columns. Each object has an originalName field that indicates the object name in Teradata, and a name field that indicates the target name for the object in BigQuery.

Columns have the following fields in addition:

- An originalType field that indicates the column data type in Teradata.
- A type field that indicates the target data type for the column in BigQuery.
- A usageType field to capture information about the way the column is used by the system, such as in clustering or partitioning. The following usage types are supported:

Clustering: You can annotate up to four columns in each target table with this usage type. The column order for clustering is determined based on the order in which they appear in the custom schema. The columns you select must meet the constraints for clustering in BigQuery. If a PARTITIONING field is specified for the same table, BigQuery uses these columns to create a clustered table.

Commit_timestamp: You can annotate only one column in each target table with this usage type. Use this usageType to identify an update timestamp column for incremental updates. This column is used to extract rows created / updated since the last transfer run. You can only use this usage type with column that has a TIMESTAMP or DATE data type.

Partitioning: You can annotate only one column in each target table with this usage type. This column is used in the partitioned table definition for the containing tables object. You can only use this usage type with column that has a TIMESTAMP or DATE data type.

Default: You can annotate multiple columns in one target table with this usage type. This usageType indicates that the column has no special use in the source system. This is the default value.

Primary_key: You can annotate columns in each target table with this usage type. Use this usage type to identify either just one column as the primary key or in the case of a composite key use the same usage type on multiple columns to identify the unique entities of a table. These columns work together with COMMIT_TIMESTAMP to extract rows created or updated since the last transfer run.

You can create a custom schema file manually, based on this example, or you can have the migration agent generate one for you when you initialize the agent.

© 2025 Persistent Systems Ltd. All rights reserved.

On-demand or Incremental Transfers

When migrating data from a Teradata database instance to BigQuery, the BigQuery Data Transfer Service supports both full transfers (on-demand transfer) and recurring transfers (incremental transfers). You designate the transfer as on-demand or incremental in the scheduling options when setting up a transfer.

- **On-demand transfer**: Use this mode to perform the full snapshot migration of schema and data from Teradata to BigQuery.
- **Scheduled transfer**: Use this mode to perform the full snapshot and regularly migrate new and modified data (incremental data) from Teradata to BigQuery. Incremental transfers requires customizing your schema to annotate columns with either of the below use cases:
  - Annotate columns with only COMMIT_TIMESTAMP usage type: In this transfer, new or modified rows in Teradata are appended to data in BigQuery. Updated rows in BigQuery tables might potentially have duplicate rows with old and new values.
  - Annotate columns with both COMMIT_TIMESTAMP and PRIMARY_KEY usage type: In this transfer, new rows are appended and modified rows are updated to the corresponding row in BigQuery. The column defined in PRIMARY_KEY is used to maintain uniqueness of the data in BigQuery.
  - The PRIMARY_KEY column defined in the schema does not have to be the PRIMARY_KEY in the Teradata table. It can be any column, but must contain unique data.

© 2025 Persistent Systems Ltd. All rights reserved.

Incremental Transfers

In incremental transfers, the first transfer always creates a table snapshot in BigQuery. All subsequent incremental transfers will adhere to the annotations defined in the custom schema file explained below.

For each transfer run, a timestamp of the transfer run is saved. For each subsequent transfer run, an agent receives the timestamp of a previous transfer run (T1) and a timestamp of when the current transfer run started (T2).

The following table describes how the migration agent handles Data Definition Language (DDL) and Data Manipulation Language (DML) operations in incremental transfers.

Teradata operation	Type	Teradata to BigQuery support
Create	DDL	A new full snapshot for the table is created in BigQuery.
Drop	DDL	Not supported
Alter (Rename)	DDL	A new full snapshot for the renamed table is created in BigQuery. The previous snapshot deleted from BigQuery. The user is not notified of the renamed table.
Insert	DML	New rows are added to the BigQuery table.
Update	DML	Rows are appended to the BigQuery table as new, similar to an INSERT operation if only COMMIT_TIMESTAMP is used. Rows are updated, similar to an UPDATE operation both COMMIT_TIMESTAMP and PRIMARY_KEY are used.
Merge	DML	Not supported. See instead Insert, Update, and Delete.
Delete	DML	Not supported

© 2025 Persistent Systems Ltd. All rights reserved.

Location Considerations

Your Cloud Storage bucket must be in a region or multi-region that is compatible with the region or multi-region of the destination dataset in BigQuery.

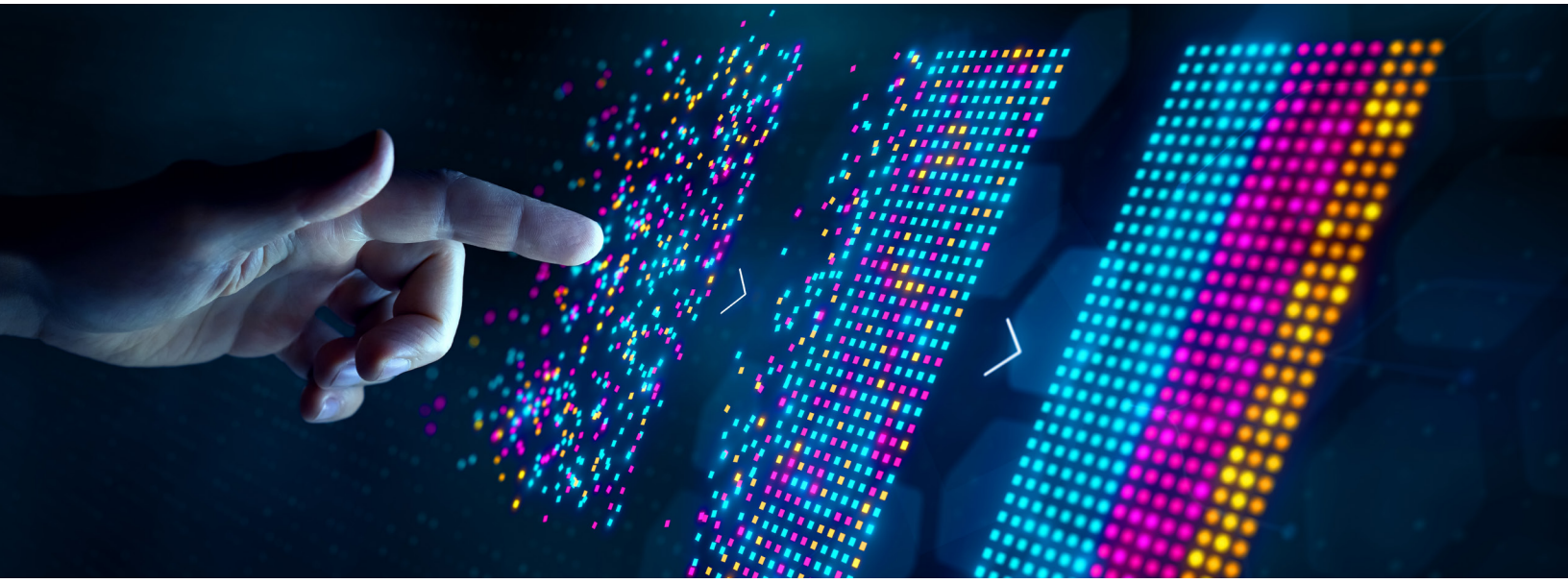
- If your BigQuery dataset is in a multi-region, the Cloud Storage bucket containing the data you're transferring must be in the same multi-region or in a location that is contained within the multi-region. For example, if your BigQuery dataset is in the `EU` multi-region, the Cloud Storage bucket can be located in the `europe-west1` Belgium region, which is within the EU.
- If your dataset is in a region, your Cloud Storage bucket must be in the same region. For example, if your dataset is in the `asia-northeast1` Tokyo region, your Cloud Storage bucket cannot be in the `ASIA` multi-region.

For detailed information about transfers and regions, see Dataset locations and transfers.

Pricing

The data transfer with BigQuery is free of charge. However, costs can be incurred outside of Google by using this service, such as platform outbound data transfer charges.

- Extraction, uploading to a Cloud Storage bucket, and loading data into BigQuery is free.
- Data is not automatically deleted from your Cloud Storage bucket after it is uploaded to BigQuery. Consider deleting the data from your Cloud Storage bucket to avoid additional storage costs. See Cloud Storage pricing.
- Standard BigQuery Quotas & limits on load jobs apply.
- Standard DML BigQuery Quotas & limits on incremental ingestion upserts will apply.
- After data is transferred to BigQuery, standard BigQuery storage and compute pricing applies.
- See our transfers Pricing page for details.



© 2025 Persistent Systems Ltd. All rights reserved.

Limitations

- One-time, on-demand transfers are fully supported. Incremental transfers are in Beta. DDL / DML operations in incremental transfers are partially supported.
- During data transfer, data is extracted to a directory on the local file system. Make sure there is adequate free space.
  - When using the FastExport mode of extraction, you can set the maximum storage space to be used, and the limit strongly enforced by the migration agent. Set the max-local-storage setting in the migration agent's configuration file when setting up a transfer from Teradata to BigQuery.
  - When using the TPT extraction method, make sure the file system has enough free space — larger than the largest table partition in the Teradata instance.
- The BigQuery Data Transfer Service converts schema automatically (if you don't supply a custom schema file) and transfers Teradata data to BigQuery. Data is mapped from Teradata to BigQuery types.
- Files are not automatically deleted from your Cloud Storage bucket after being loaded into BigQuery. Consider deleting the data from your Cloud Storage bucket after loading it into BigQuery, to avoid additional storage costs. See Pricing.
- The speed of the extraction is bounded by your JDBC connection.
- The data extracted from Teradata is not encrypted. Take appropriate steps to restrict access to the extracted files in the local file system, and ensure the Cloud Storage bucket is properly secured.
- Other database resources, such as stored procedures, saved queries, views, and user-defined functions are not transferred and not in the scope of this service.
- Incremental transfers does not support hard deletes. Incremental transfers does not sync any deleted rows in Teradata with BigQuery.

© 2025 Persistent Systems Ltd. All rights reserved.

Re(AI)magining™ the World

About Persistent

Persistent Systems (BSE & NSE: PERSISTENT) is a global services and solutions company delivering Digital Engineering and Enterprise Modernization to businesses across industries. With over 23,900 employees located in 19 countries, the Company is committed to innovation and client success. Persistent offers a comprehensive suite of services, including AI-enabled software engineering, product development, data and analytics, CX transformation, cloud computing, and intelligent automation. The Company is part of the MSCI India Index and is included in key indices of the National Stock Exchange of India, including the Nifty Midcap 50, Nifty IT, and Nifty MidCap Liquid 15 as well as several on the BSE such as the S&P BSE 100 and S&P BSE SENSEX Next 50. Persistent is also a constituent of the Dow Jones Sustainability World Index. The Company has achieved carbon neutrality, reinforcing its commitment to sustainability and responsible business practices. As a participant of the United Nations Global Compact, Persistent is committed to aligning strategies and operations with universal principles on human rights, labor, environment, and anti-corruption, as well as take actions that advance societal goals. With 327% growth in brand value since 2020, Persistent is the fastest-growing IT services brand in the 2024 Brand Finance India 100 Report.

USA
Persistent Systems, Inc.
2055 Laurelwood Road, Suite 210
Santa Clara, CA 95054
Tel: +1 (408) 216 7010
Fax: +1 (408) 451 9177
Email: info@persistent.com

India
Persistent Systems Limited
Bhageerath, 402
Senapati Bapat Road
Pune 411016
Tel: +91 (20) 6703 0000
Fax: +91 (20) 6703 0008

Persistent

www.persistent.com

© 2025 Persistent Systems Ltd. All rights reserved.