Using Temporary Tables to Improve Performance for SQL Data Services
Abstract

A temporary table is a table in a relational database that can persist frequently queried SQL data service data. Complex queries store and retrieve large amounts of intermediate, temporary data. When you enable temporary tables to persist this temporary data, you can improve query performance. This article describes how to use temporary tables to increase the performance of queries that use intermediate data.

Supported Versions

- Data Services 9.5.1, 9.6.0, 9.6.1

Table of Contents

- Overview ................................................................. 2
- Persisting Virtual Data in Temporary Tables .................... 2
- Temporary Table Implementation .................................. 3
- Temporary Table Operations ........................................ 3
- Rules and Guidelines for Temporary Tables .................... 4

Overview

Complex queries commonly store and retrieve large amounts of intermediate data. You can improve the performance of SQL data service queries by persisting this intermediate data in temporary tables.

Persisting Virtual Data in Temporary Tables

A temporary table is a table in a relational database that stores intermediate, temporary data. Complex queries commonly require storage for large amounts of intermediate data, such as information from joins. When you implement temporary tables, business intelligence tools can retrieve this data from the temporary table instead of the SQL data service. This results in an increase in performance.

Temporary tables also provide increased security in two ways. First, only the user of the active session can access the tables. Also, the tables persist while a session is active, and the database drops the tables when the connection closes.

To implement temporary tables, an administrator must create the Data Integration Service, and then configure the Table Storage Connection in the SQL Connection properties of the Data Integration Service. After the administrator configures the connection, a developer uses the Informatica ODBC or JDBC driver to configure a connection between a business intelligence tool and the Informatica SQL data service. When these connections are configured, the business intelligence tool can create and use temporary tables.

Temporary tables for all SQL data services in a Data Integration Service use the same relational database connection. When the connection to the SQL data service is active, you can connect to the SQL data service through a JDBC or ODBC client. The relational database drops temporary tables when the session ends. If the Data Integration Service unexpectedly shuts down, the relational database drops temporary tables on the next Data Integration Service startup.
Temporary Table Implementation

You can store intermediate query result set data in temporary tables when complex queries produce large amounts of intermediate data. For example, temporary tables can store frequently used join results. Business intelligence tools can query the temporary table instead of the SQL data service, resulting in increased performance.

To implement temporary tables, the Informatica administrator and the business intelligence tool user perform the following separate tasks:

**Step 1. The Informatica administrator creates a connection for the data integration service.**

In the Administrator tool, create a connection to the SQL data service. Edit the SQL Properties of the Data Integration Service and select a relational database connection for the Table Storage Connection property. Recycle the Data Information Service.

**Step 2. The business intelligence tool user creates a connection for the SQL data service.**

In a business intelligence tool, create a connection to the SQL data service. The connection uses the Informatica ODBC or JDBC driver.

**Step 3. Queries from the business intelligence tool create and use temporary tables.**

While the connection is active, the business intelligence tool issues queries to the SQL data service. These queries create and use temporary tables to store large amounts of data that the complex query produces. When the connection ends, the database drops the temporary table.

Temporary Table Operations

After you create the SQL data service connection, you can use SQL operations to create, populate, select from, or drop a temporary table. You can issue these commands in a regular or stored SQL statement.

You can perform the following operations:

Create a temporary table.

To create a temporary table on the relational database, use the following syntax:

```sql
CREATE TABLE emp (empID INTEGER PRIMARY KEY, eName char(50) NOT NULL,)
```

You can specify the table name in the SQL data service.

**Note:** Use `CREATE TABLE`, not `CREATE TEMPORARY TABLE`. The use of `CREATE TEMPORARY TABLE` is not supported.

Create a temporary table from a source table.

You can create a temporary table with or without data from a source table.

The following syntax is supported in Informatica Data Services version 9.5.1:

```sql
CREATE TABLE emp.backup as select * from emp
```

Where `emp` is an existing schema in the SQL data service that you connected to.

The following syntax is supported in Informatica Data Services version 9.6.0 and 9.6.1:

```sql
CREATE TABLE emp.backup as select * from emp [ LIMIT n ]
```

Where `emp` is an existing schema in the SQL data service that you connected to.

When you create a temporary table with data, the Data Integration Service populates the table with the data. The `CREATE AS` operator copies columns from a database table into the temporary table.

You cannot maintain foreign key or primary key constraints when you use `CREATE AS`.

You can cancel a request before the Data Integration Service copies all the data.
Note: The Informatica administrator must create a connection, and then configure it in SQL Properties as the Table Storage Connection, before you create the temporary table.

Insert data into a temporary table.

To insert data into a temporary table, use the `INSERT INTO <temp_table>` statement. You can insert literal data and query data into a temporary table.

The following table shows examples of SQL statements that you can use to insert literal data and query data into a temporary table:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literal data</td>
<td>Literals describe a user or system-supplied string or value that is not an identifier or keyword. Use strings, numbers, dates, or boolean values when you insert literal data into a temporary table. Use the following statement format to insert literal data into a temporary table:</td>
</tr>
<tr>
<td></td>
<td><code>INSERT INTO &lt;TABLENAME&gt; &lt;OPTIONAL COLUMN LIST&gt; VALUES (&lt;VALUE LIST&gt;),</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;VALUE LIST&gt;)</code></td>
</tr>
<tr>
<td></td>
<td>For example, <code>INSERT INTO temp_dept (dept_id, dept_name, location) VALUES</code></td>
</tr>
<tr>
<td></td>
<td>(2, 'Marketing', 'Los Angeles').</td>
</tr>
<tr>
<td>Query data</td>
<td>You can query an SQL data service and insert data from the query into a temporary table. Use the following statement format to insert query data into a temporary table:</td>
</tr>
<tr>
<td></td>
<td><code>INSERT INTO &lt;TABLENAME&gt; &lt;OPTIONAL COLUMN LIST&gt; &lt;SELECT QUERY&gt;</code></td>
</tr>
<tr>
<td></td>
<td>For example, <code>INSERT INTO temp_dept(dept_id, dept_name, location) SELECT</code></td>
</tr>
<tr>
<td></td>
<td>dept_id, dept_name, location from dept where dept_id = 99.</td>
</tr>
<tr>
<td></td>
<td>You can use a set operator, such as <code>UNION</code>, in the SQL statement when you insert query data into a temporary table. Use the following statement format when you use a set operator:</td>
</tr>
<tr>
<td></td>
<td><code>INSERT INTO &lt;TABLENAME&gt; &lt;OPTIONAL COLUMN LIST&gt; (&lt;SELECT QUERY&gt; &lt;SET OPERATOR&gt; &lt;SELECT QUERY&gt;)</code></td>
</tr>
<tr>
<td></td>
<td>For example, <code>INSERT INTO temp_dept select * from north_america_dept UNION</code></td>
</tr>
</tbody>
</table>
|                  | select * from asia_dept.`

Select from a temporary table.

You can query the temporary table with the `SELECT ... from <table> statement`.

Drop a temporary table.

To drop a temporary table from the relational database, use the following syntax:

```
DROP TABLE <tableName>
```

If the table is not dropped on the physical database, the SQL data service drops the table the next time the Data Integration Service starts, if the table still exists.

Rules and Guidelines for Temporary Tables

Consider the following rules and guidelines for creation and use of temporary tables:

- You can specify schema and default schema for a temporary table.
- You can place the primary key, NULL, NOT NULL, and DEFAULT constraints on a temporary table.
- You cannot place a foreign key or CHECK and UNIQUE constraints on a temporary table.
- You cannot issue a query that contains a common table expression or a correlated subquery against a temporary table.
• CREATE AS statements cannot contain a correlated subquery.

Author

Mark Pritchard
Principal Technical Writer