

TERADATA TEMPORAL CAPABILITY

PUT TIME ON YOUR SIDE

Change and evolution is woven into the fabric of every business. But too often, companies lack critical insight into their most vital business activities. Why? Because unless they can keep comprehensive historical detailed data as items change and evolve over time, they can't accurately capture how business changes affect the business. The result? Apples-to-oranges comparisons, incomplete analyses, and misleading reporting that can lead to poor business decisions.

With most databases, maintaining such a complete history is a cumbersome, complex, and error-prone challenge. That's because the ability to represent the state of your business at any point in time requires that you have all your historical data at your fingertips. But current temporal solutions lead to data quality issues and more complex data maintenance and queries.

THE TEMPORAL DIFFERENCE

Temporal means the time dimension has been added to data management and query processing. It lets the data warehouse track not just transactions over time, but how data and the business change and evolve over time.

A temporal database includes special support for the time dimension. It's a system that provides special facilities for storing, querying, and updating historical data. Rather than just changing data as updates are made, a temporal database retains the history represented by the old value and allows users to use that history to look at the business attribute as of any point in time. Popular database systems have not provided this facility, so users had to manage the history at the ETL and application level. Without the automation built into the database, data maintenance is complicated, and application and query development is extended and more error prone if the functionality can be provided at all.

As you add effective dates to the data and keep historical rows showing changes over time, even queries about the current state of the business are complicated and difficult to compose. Or, if you capture database snapshots periodically, your analytical ability is limited by the spotty nature of your historical data.

REPLACE COMPLEXITY WITH FLEXIBILITY

No one understands the unique requirements of the decision-making and analytics environment better than Teradata Corporation, the global leader in data warehousing and analytics.

Teradata includes a unique temporal database option with Teradata® Database. This temporal option is another pioneering innovation in our ongoing effort to help you make smarter, faster decisions.

With the in-database temporal option, Teradata Database can automatically handle the update complexities related to temporal data. In addition, you can easily specify full bi-temporal queries with no knowledge of effective date fields and complex condition clauses. Plus, in addition to simplifying the complexities of temporal processing, the temporal option includes many optimizations for reduced I/O and efficient processing of the more detailed history.

That means the temporal option in Teradata Database lets you:

- ~ Gain richer, more meaningful analysis that reflects changes in your business and environment over time.
- ~ Eliminate complexity that increases development costs, delays applications, and complicates queries, ETL, and data maintenance.
- ~ Take advantage of your database to automatically track history and understand data changes over time.

Just as important, the temporal capability brings you and your business an array of additional benefits, including improved data quality for users everywhere in your organization and added insight for stronger business analytics. And because the option also lowers application development and maintenance effort, you'll see faster time to value as well.

ANSWER YOUR MOST CRUCIAL QUESTIONS

With Teradata Database's temporal capability, your business users will be able to ask more sophisticated questions—and receive more insightful answers than ever before. Questions that were previously so time consuming or complicated that they were too costly and cumbersome to even ask, can now be handled quickly and cost effectively.

Teradata temporal enables you to ask these kinds of vital business questions:

- ~ What was the profitability by segment of Flight 123 three days before departure? Or seven or 14 days?
- ~ What was the BOM for a given manufactured item at time of failure?
- ~ How many wireless subscribers were won or lost from April through June?
- ~ How was a replenishment decision made for an item on August 23? And how would new rules change the decision?
- ~ What is the railroad crew utilization to car volume comparison by time interval?
- ~ How many customers did my records show were active as of December 31 when I calculated year-end bonuses last January 31?
- ~ How did my sales of products in product category 12 compare March of last year vs. March of this year using the product category definitions in effect at the time the sales were made?

The answers to these and other mission-critical business questions open up new opportunities for more in-depth analysis of every facet of your operations, analysis you can use to create accurate reports and make better decisions that can lead to reduced expenses, improved customer service—and a more profitable business.

ACCURATE AND AUTOMATIC

Teradata temporal integrates time dimension processing throughout the Teradata Database. That integration makes the database time aware and optimized for all phases of processing temporal tables. In true Teradata fashion, temporal is fully parallel, scalable, easy to use with industry standard SQL, and highly available. The option supports Valid Time, Transaction Time, and fully bi-temporal tables, including both valid and transaction time dimensions (i.e., when a fact was true in the world the data represents, and when the database became aware of the occurrence).

With the temporal option, whenever data are updated, the database automatically produces the required

THINGS YOU NO LONGER HAVE TO DO IN THE TIME-AWARE TERADATA DATABASE

Data Maintenance

- ~ Update effective dates in data.
- ~ Specify “transaction time” timestamp when data are entered or updated.
- ~ Insert new data rows for updates.
- ~ Manage temporal history storage space.

Query and Application Development

- ~ Write complex period-based condition clauses.
- ~ Add effective date qualifications to “current queries.”
- ~ Train users and application developers in effective date columns and processing for “AS OF” query development.
- ~ Save old reports or database snapshots to facilitate looking back for analysis, compliance reporting, or audit trails.

number of entries and tracks the required effective date information. So you and your business can gain a complete view of the data over time—data that are automatically maintained.

Existing queries, those standard queries about current situations in existing applications, will continue to execute unchanged. But queries with a temporal aspect (those based on a particular time, or a query combining some elements at one time and other elements at another) can be specified much more simply, often by placing a prefix in front of a standard query. This allows minimal new SQL knowledge or changes and makes it easier to use business intelligence (BI) tools that aren't temporal aware for temporal analysis.

OPTIMIZED FOR PERFORMANCE

We've optimized our temporal processing capability for efficiency. For example, we've integrated temporal intelligence into the multi-level partitioned primary index feature of Teradata Database so you can keep a *current* partition of all current state information. The Teradata optimizer will then automatically eliminate processing of historical data for the common queries that ask about the current state of the business. Partitions with historical data about the past are ignored. And that dramatically improves query efficiency while reducing I/O.

SEQUENCED QUERY—TEMPORAL JOIN

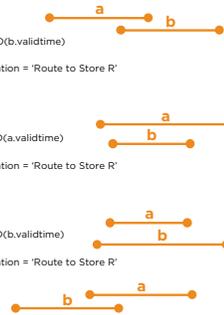
A laptop (150,101) was found to be broken due to shock. Find out other items that were on the same route (same truck) at the same time.

WITH TEMPORAL SUPPORT

```
VALIDTIME
SELECT b.item_id,b.item_serial_num
FROM objectlocation a,
     objectlocation b
WHERE
  a.item_id = 150
  AND a.item_serial_num = 101
  AND a.location = 'Route to Store R'
  AND b.location = a.location
```

WITHOUT TEMP SUPPORT WITH STATE TABLE

```
SELECT b.item_id, b.item_serial_num, BEGIN(b.validtime), END(a.validtime)
FROM state_objectlocation a, state_objectlocation b
WHERE a.item_id = 150 and a.item_serial_num = 101 and a.location = 'Route to Store R'
  AND b.location = a.location
  AND BEGIN(a.validtime) <= BEGIN(b.validtime)
  AND END(a.validtime) > BEGIN(b.validtime)
  AND END(a.validtime) <= END(b.validtime)
UNION ALL
SELECT b.item_id, b.item_serial_num, BEGIN(b.validtime), END(b.validtime)
FROM state_objectlocation a, state_objectlocation b
WHERE a.item_id = 150 and a.item_serial_num = 101 and a.location = 'Route to Store R'
  AND b.location = a.location
  AND BEGIN(a.validtime) <= BEGIN(b.validtime)
  AND END(a.validtime) >= END(b.validtime)
  AND END(a.validtime) <= END(b.validtime)
UNION ALL
SELECT b.item_id, b.item_serial_num, BEGIN(a.validtime), END(a.validtime)
FROM state_objectlocation a, state_objectlocation b
WHERE a.item_id = 150 and a.item_serial_num = 101 and a.location = 'Route to Store R'
  AND b.location = a.location
  AND BEGIN(a.validtime) >= BEGIN(b.validtime)
  AND END(a.validtime) <= END(b.validtime)
  AND END(a.validtime) <= END(b.validtime)
UNION ALL
SELECT b.item_id, b.item_serial_num, BEGIN(b.validtime), END(b.validtime)
FROM state_objectlocation a, state_objectlocation b
WHERE a.item_id = 150 and a.item_serial_num = 101 and a.location = 'Route to Store R'
  AND b.location = a.location
  AND BEGIN(a.validtime) >= BEGIN(b.validtime)
  AND BEGIN(a.validtime) < END(b.validtime)
  AND END(a.validtime) >= END(b.validtime)
```



CURRENT UPDATE

Denim pants (125,101) arrive at store R, as scanned by RFID scanner.

WITH TEMPORAL SUPPORT

```
UPDATE objectlocation
  SET location = 'Store R'
  WHERE item_id = 125
  AND item_serial_num = 101
```

WITHOUT TEMPORAL SUPPORT

```
INSERT INTO objectlocation
  SELECT 125,101, 'Store R', CURRENT_DATE,END(vt)
FROM objectlocation
WHERE item_id = 125
  AND item_serial_num = 101
  AND BEGIN(vt) <= CURRENT_DATE
  AND END(vt) > CURRENT_DATE;
UPDATE objectlocation
  SET END(vt) = CURRENT_TIME
  WHERE item_id = 125
  AND item_serial_num = 101
  AND BEGIN(vt) < CURRENT_TIME
  AND END(vt) > CURRENT_TIME;
UPDATE objectlocation
  SET location = 'Store R'
  WHERE item_id = 125
  AND item_serial_num = 101
  AND BEGIN(vt) > CURRENT_TIME;
```

Figure 1. Two examples of how queries can be simplified when you have a time-aware database.

Those optimizations and enhanced technical capabilities make it much easier to implement your temporal logic and business applications. And they also make it easier to derive much more value from your data by providing users throughout your enterprise a more complete view of your business over time.

CUT COST AND COMPLEXITY

Teradata Temporal reduces the cost and complexity of implementing a temporal data warehouse. It also opens up a new analytical dimension to you, a dimension that

will provide you with new insights and a better understanding of your business by being able to clearly see how things have changed over time or how they looked at any point in time.

FOR MORE INFORMATION

To find out more about how the temporal option in Teradata Database can help track changes in your business over time so you can make smarter decisions and grow your business, contact your Teradata representative or visit Teradata.com.

