IBM Software
White Paper

Improving data quality for exceptional business accuracy and compliance

Temporal data management with IBM DB2 Time Travel Query

Matthias Nicola
IBM Silicon Valley Laboratory
mnicola@us.ibm.com

What is time?

Albert Einstein once said, "The only reason for time is so that everything doesn't happen at once." After all, time is a measuring system that allows us to sequence events, measure the intervals between them and quantify rates of change. Indeed, it is the experience of change that makes time so important. For without the concept of time, there would be no change, no movement and no increase or decrease of any business metric. Organizations must track and capture time for the critical purpose of understanding and measuring change.

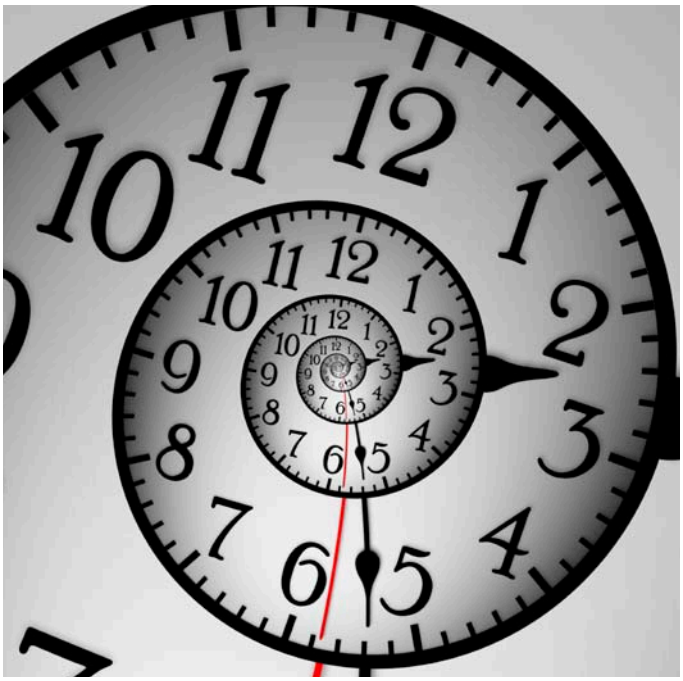
This white paper describes the need for time-aware, or *temporal*, data management. It explains the drawbacks of existing approaches and describes the advantages of the temporal capabilities provided by IBM® DB2® database software. These capabilities are collectively known as DB2 Time Travel Query and include temporal queries, temporal updates and temporal constraints, among others.

The need for time travel

Traveling through time to observe the past or take a peek into the future has always been a dream of mankind, because as humans we are inevitably stuck in the present. And yet the past and the future are of utmost importance to all our endeavors. IBM DB2 provides a "time machine for data" that enables organizations to move freely along one or more time axes of their business.

There are two key organizational drivers for this time travel:

- Governance, risk and compliance (GRC) initiatives and the often-associated need to meet auditing requirements
- Management of an organization's time dimension, such as past and future effective dates of business information for improved decision making



Meeting GRC and auditing requirements

Compliance and auditing regulations require many organizations to keep a history of data changes so that they can go back in time to verify data's existence at any given point. A prime example is the financial sector, where companies are subject to extensive compliance and auditing requirements and might need to answer questions like the following:

- What data was used when the price and risk of a given investment was assessed two months ago?
- Can we rerun the risk assessment based on the applicable past state of our data?
- Do we have full traceability of retroactive data corrections so that we can look at past data with and without the corrections?
- Can we prove that all parts of our firm have been using the same data at the same point in time?

Similar questions arise in other industries as well, such as healthcare. A healthcare provider might ask: What knowledge did we have about a patient's medical condition when we advised a change in treatment and new medication six months ago?

Failure to retain an accurate and complete history of data can have severe consequences. An incomplete data history may lead to costly errors, such as loss of investments in financial trades or misdiagnosis in healthcare. Additional consequences include fines for noncompliance, serious legal repercussions and damaged reputations.

To help organizations address compliance requirements, DB2 provides the following capabilities:

- Transparently maintain a system history of all data changes with system-generated time stamps, which are managed as *system time periods*
- Allow applications or end users to formulate simple queries that access data as of any desired point in the past, with picosecond precision in time
- Support the ISO/IEC 9075:2011 (SQL:2011) standards, which dramatically reduce temporal query complexity

Managing business time periods

Many organizations also need to manage the time dimension of their business, such as the effective dates or validity periods of business information. This *business time*, also known as *valid time*, describes the period of time during which business information is valid in the real world—as opposed to *system time*, which is a system-generated history of data changes.

Examples of business time include:

- The period of time for which a specific interest rate is valid for a given loan
- The time during which a specific address is valid for a given customer
- The period during which a product is on sale
- The effective dates of updated terms and conditions for a contract or insurance policy

Such business periods can be in the past, present and future. Hence, organizations often need to insert, modify or query business information in the past, present or future.

For example, the interest rate of a loan might increase by one percent on the first day of the next month and remain valid for one year. This is an update that is effective in the future. Or a customer notifies his insurance company that his address changed two weeks ago, which might result in a higher premium for his car insurance. The validity of his old address must be terminated as of two weeks ago, and the validity of the new address must start as of two weeks ago. This is an update of data in the past.

Another common requirement in this context is the enforcement of temporal constraints, such as uniqueness relative to time. For example, a bank needs to ensure that only one interest rate is valid for a loan at any given point in time, to avoid ambiguity. Similarly, a reservation system must make certain that no two guests book the same hotel room for overlapping periods of time.

"DB2 10 is revolutionary software that will greatly simplify database management. We tested several new features we expect to result in significant performance improvements for our customers. Those advancements include Time Travel Query, which will help customers store and retrieve data from particular time periods, a feature especially helpful to customers handling insurance policies, credit cards and loans where interest rates change over time."

—Jithen Bondada, Vice President Application Development/B2B, Miracle Software Systems, Inc.

DB2 enables companies to address these requirements by offering:

- Comprehensive capabilities to insert, modify and query effective dates in the past, present or future—managed as business time periods
- Optional enforcement of temporal constraints
- Support for the SQL:2011 standard, which greatly simplifies the management of business time in databases

The importance of bitemporal data management

Organizations often must manage business time periods while retaining a full history of all changes and corrections. Therefore, data needs to be managed along two dimensions of time: business time and system time. This powerful combination is known as *bitemporal data management*.

For example, only bitemporal data management can answer business questions such as "On January 1, what was our expected volume of loans for year-end?" In this question, "On January 1" refers to system time because it considers data that was recorded and current in the database in January. Conversely, "for year-end" refers to business time, which in this case is a validity date in the future.

Bitemporal data management provides comprehensive information about what is known and when it is known. It can also trace business data corrections in the past, present and future. As a result, bitemporal data management helps organizations reduce business risk and achieve thorough compliance and enhanced decision making.

DB2 is designed for bitemporal data management, integrating system time and business time features. It also supports the SQL:2011 standard, which helps simplify management of data along both dimensions of time.

The shortcomings of traditional methods of implementing time-aware processing logic

Organizations have traditionally addressed the need for temporal data management by implementing homegrown solutions. Such solutions typically require hand-coding of time-aware data processing logic in application code or custom-stored procedures and database triggers. However, this approach has inherent disadvantages:

- Implementing time-aware processing logic tends to be very complicated.
- Application development and maintenance typically incur high costs.
- Application performance is often suboptimal.
- Because custom code lacks standardization of temporal operations, different applications or departments within an organization often have different implementations of time-based processing logic.

These challenges make it almost impossible to ensure temporal consistency of data for a single application or across applications. As a result, organizations may end up with moderate data quality at best, as well as exposure to risk.

The following product pricing example illustrates the complexity of implementing time-aware operations. The row of data in Figure 1 states that the product with ID 123 has a price of $19.95 that is valid starting January 1, 2012, and expires on June 1, 2012. The effective_start date indicates the first day when the price is valid, and the effective_end date is the first day on which the price is no longer valid.

productID	price	effective_start	effective_end
123	$19.95	2012-01-01	2012-06-01

Figure 1: Sample row of data in product pricing example

If the product will be offered at a discounted price of $15.00 during the month of March, the data needs to be modified as shown in Figure 2.

productID	price	effective_start	effective_end
123	$19.95	2012-01-01	2012-03-01
123	$15.00	2012-03-01	2012-04-01
123	$19.95	2012-04-01	2012-06-01

Figure 2: Modified data in product pricing example

The first row in Figure 2 shows that the original price of $19.95 is still effective during January and February, but no longer effective on March 1. The second row represents the new price of $15.00 for the month of March, and the third row reflects a return to the original price for April and May.

"The use of standardized SQL syntax for temporal operations and the integration deep into the database engine make DB2 a leader in second-generation bitemporal data management—Bitemp 2.0!"

—Craig Baumunk, Principal, BitemporalData.com

The transformation of the data in Figure 1 to the data in Figure 2 is a nontrivial operation. The complexity stems from the fact that existing data must be modified and new rows might need to be generated, depending on how the existing dates (2012-01-01, 2012-06-01) relate to the newly provided dates (2012-03-01, 2012-04-01).

What happens to the Figure 2 data if the price for product 123 changes to $10.00 from March 15 to April 15? This new period overlaps with two of the existing periods in Figure 2, which makes the required transformation even more complicated. An application would first examine all existing rows to determine which ones are affected by the price change from March 15 to April 15. Then it would determine how to update these rows and which new rows to insert, if any. At the same time, the application must ensure that the different prices for product 123 never have overlapping periods—another complex task.

The benefits of temporal data management with DB2

With DB2, temporal operations such as the data modifications required in the product pricing example are no longer hand-coded as application logic. Instead, the application can express the desired changes in a declarative way—using standard SQL—and let DB2 determine and perform the necessary data modifications. Figure 3 shows how two simple SQL statements perform the price changes discussed in the product pricing example.

```
UPDATE product
  FOR PORTION OF BUSINESS_TIME FROM '2012-03-01' TO '2012-04-01'
SET price = 15.00
WHERE productID = 123;

UPDATE product
  FOR PORTION OF BUSINESS_TIME FROM '2012-03-15' TO '2012-04-15'
SET price = 10.00
WHERE productID = 123;
```

Figure 3: IBM DB2 enables the use of simple SQL statements to perform complex temporal data transformations

The FOR PORTION OF BUSINESS_TIME clause shown in Figure 3 is one of several language constructs in the SQL:2011 standard that simplify temporal data management. DB2 support for this SQL standard provides several advantages:

- Short, simple and declarative coding of temporal operations rather than complex, procedural application code
- Increased developer productivity and reduced cost of application development and maintenance
- Standardization of temporal processing logic across applications, which helps increase consistency and protect IT investments
- High-quality data across the enterprise and reduced business risk
- Exceptional performance, because temporal operations are deeply integrated in the DB2 engine

An IBM study[1] compared three different options for performing business time operations like the ones in the product pricing example:

1. DB2-based option, using standardized SQL for temporal operations
2. Homegrown option, implemented as a set of stored procedures
3. Homegrown option, implemented in Java

Although all three options implemented the same temporal operations, the cost savings provided by DB2—as inferred from the coding and testing efforts required—were striking. The DB2-based approach reduced coding requirements by more than 90 percent over both homegrown implementations (see Figure 4). Implementing just the core business-time logic in SQL stored procedures or in Java required 16 times and 45 times as many lines of code, respectively, as the equivalent simple SQL statements that use the new DB2 features.

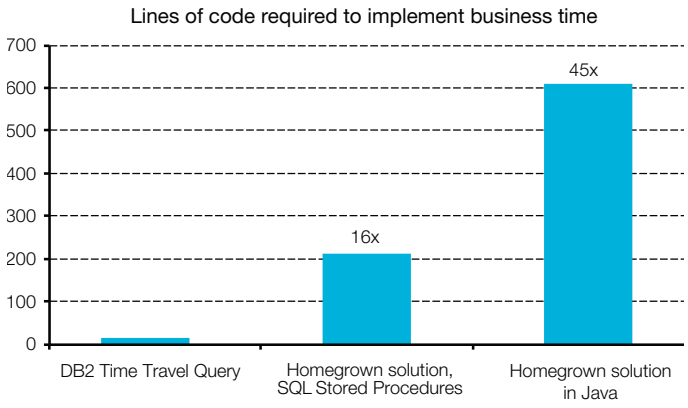


Figure 4: Lines of code provide a metric for application development and maintenance costs

Furthermore, IBM testers took less than one hour to develop and test the required SQL statements. By contrast, coding and testing one of the homegrown approaches took four to five weeks. Beyond the development cost, the length and complexity of the code also significantly affects the application maintenance cost.

Conclusion: Step into the DB2 time machine

IBM DB2 acts as the single time machine for applications that access the same data. Temporal data processing is deeply integrated into the DB2 engine, enabling outstanding performance and seamless integration of bitemporal data management with other database features and utilities.

Plus, DB2 supports short and efficient SQL:2011 coding of complex temporal operations, which helps increase developer productivity and reduce the cost to implement and maintain time-aware applications by up to 45 times, as shown by an IBM study.

Overall, DB2 helps improve data consistency and quality across the enterprise and provides a cost-effective means to address auditing and compliance issues. As a result, organizations can reduce their risk of noncompliance and achieve greater business accuracy.

For more information

For more information on the temporal capabilities of IBM DB2, please explore the following resources:

- Download and read the white paper "A Matter of Time: Temporal Data Management in DB2" at **ibm.com**/developerworks/data/library/techarticle/dm-1204db2temporaldata/index.html
- Visit the DB2 temporal discussion forum and ask your own questions: **ibm.com**/developerworks/forums/forum.jspa?forumID=2316
- Check out the DB2 Temporal Best Practices: **ibm.com**/developerworks/data/bestpractices/db2luw/#bestpractices

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
April 2012

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions. It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs. THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation. Statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

[1] IBM internal testing.

Please Recycle

IMW14607-USEN-00