

Introduction

Indeed, a bitemporal database is much more powerful than a temporal database. While a temporal database tracks valid time, capturing data "as it happened in the real world," a bitemporal database involves valid time and system time. System time records when something was recorded to the database.

In other words, bitemporal databases capture information "as it actually was." For example, a temporal database can tell us where John Smith lived on Dec. 6, but a bitemporal database can tell us where John Smith lived on Dec. 6 as we knew it on Dec. 15. With a bitemporal database, organizations can answer the critical questions: What did you know, and when did you know it? It helps ensure that there is always a full and accurate picture of data at every point in time.

Bitemporal design is about historical data, but it's really much more than that. It's about strong data governance, strategic planning, risk management, and competitive advantage.

Credit: <http://www.dbta.com/Editorial/Trends-and-Applications/The-Role-of-Bitemporality-in-Data-Governance-and-Compliance-117-412.aspx>

Bitemporality Merits

- Regulatory requirements, enabling companies to avoid fines
- Audits, preserving the history of all data, including the changes made to it
- Investigations and intelligence, maintaining information so it is possible to see exactly how data was updated based on what was known at the time
- Business analytics, enabling companies to run more complex queries based on time-based data

Cost of Doing Bitemporal Business

While data analytics capabilities can come at a premium, bitemporal technology can actually save companies money and provide opportunities for increased revenue.

The cost per gigabyte of data is decreasing, but organizations today are spending more on storing historical data because they are dealing with so much more of it—for regulatory reasons, but also because companies tend to hoard data. Bitemporal design helps keep storage in check because it avoids the need to set up additional databases for historical data.

Bitemporality Database

April

Orderno	Customer	Status	Changdate
1	A	New	1/1/2013
2	B	Accepted	3/3/2013
3	C	Delivery	14/4/2013

↓

Datawarehouse

Orderno	Customer	Status	ValidFrom	ValidUntil
1	A	New	14/1/2013	31/12/9999
2	B	New	13/2/2013	2/3/2013
2	B	Accepted	3/3/2013	31/12/9999
3	C	New	18/1/2013	5/2/2013
3	C	Accepted	6/2/2013	6/3/2013
3	C	OrderPick	7/3/2013	13/4/2013
3	C	Delivery	14/4/2013	31/12/9999

Barriers to Bitemporality

Traditional relational databases are not well-suited to handle bitemporal implementations. Here are some of the reasons:

- **Integrity constraints:** Relational databases have referential and entity integrity constraints, not to mention defined schemas. When bitemporal columns are added to a relational table, they can wreak havoc on the relational data model.
- **Schema evolution:** It is difficult to change schema in a relational database in general, let alone when adding bitemporal data.
- **Multiple data models:** Bitemporality involves integrating multiple data models and data silos into a single source of truth, a difficult task for schema-bound relational databases.
- **Performance decline:** Bitemporal queries consider multiple axes of time and often span multiple servers.
- **Add-on costs:** Bitemporal is often not built into the database, it is an add-on component that comes with additional costs.