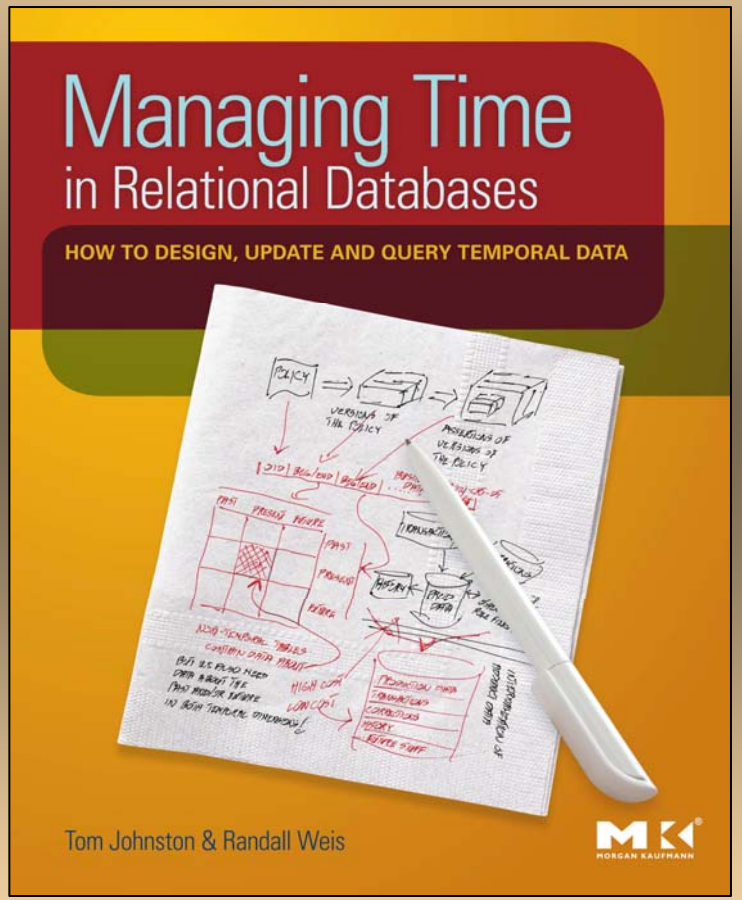
UNDERSTANDING BITEMPORAL DATA

Details of this story may be found in:



It all started with Aristotle.

But that's a story for another time….

Tom.Johnston@AssertedVersioning.com

Part I
Readings, Products

READINGS

Richard T. Snodgrass
Developing Time-Oriented Database Applications in SQL
(2000, Morgan-Kaufmann).

Tom Johnston, Randall Weis
Managing Time in Relational Databases:
How to Design, Maintain and Query Temporal Data
(2010, Morgan-Kaufmann).

And for temporal, but not bitemporal data:

C. J. Date, Hugh Darwen, Nikos Lorentzos.
Temporal Data and the Relational Model
(2002, Morgan-Kaufmann).

Anchor & Data Vault modeling.

Best practices using
history tables and versioning.

Good discussions can be found at the Bitemporal Data group and the Temporal Data Modeling group on LinkedIn.

A demo on maintaining and querying bitemporal data can be found at: AssertedVersioning.com

Thousands of academic publications on bitemporal data are available at the ACM Digital Library, at ACM.org.

A summary article of the material in this presentation can be found at Enterprise Systems Journal (esj.com)

PRODUCTS

DB2 10 for z/OS

A Matter of Time: Temporal Data Management in DB2 for z/OS
http://www-01.ibm.com/software/data/db2/zos/db2-10/

• **DB2 10 delivers the industry's first integrated bitemporal capabilities that are built directly into the database.**

PRODUCTS

Oracle 11g, Workspace Manager

R. T. Snodgrass
http://www.cs.arizona.edu/people/rts/sql3.html

The Oracle 10g Workspace Manager includes the period data type, valid-time support, transaction-time support, support for bitemporal tables, and support for sequenced primary keys, sequenced uniqueness, sequenced referential integrity, and sequenced selection and projection, in a manner quite similar to that proposed in SQL/Temporal.

PRODUCTS

Teradata 13.10

R. T. Snodgrass
http://www.cs.arizona.edu/people/rts/sql3.html

Teradata database 13.10 includes the period data type, valid-time support, transaction-time support, timeslices, temporal upward compatibility, sequenced primary key and temporal referential integrity constraints, nonsequenced queries, and sequenced projection and selection, in a manner almost identical to that proposed in SQL/Temporal.

PRODUCTS

The Asserted Versioning Framework

R. T. Snodgrass, back cover of Managing Time in Relational Databases.

"….. an original and comprehensive approach called Asserted Versioning ….. includes support for bitemporality and is a significant advance in the theory and practice of managing time-varying data."

BITEMPORAL DATA IS BECOMING MAINSTREAM!

DB2 10 for z/OS

Oracle 11g, Workspace Manager

Teradata 13.10

The Asserted Versioning Framework

So the DIY approach is no longer necessary, and was never desirable.

• Different DIYs for the same temporal requirements often mean different maintenance to achieve the same temporal results.

• Different DIYs for the same temporal requirements often mean different queries to retrieve the same temporal cases.

• Many DIY solutions are sub-optimal.

• Partial temporal solutions often require follow-on projects to support additional temporal requirements.

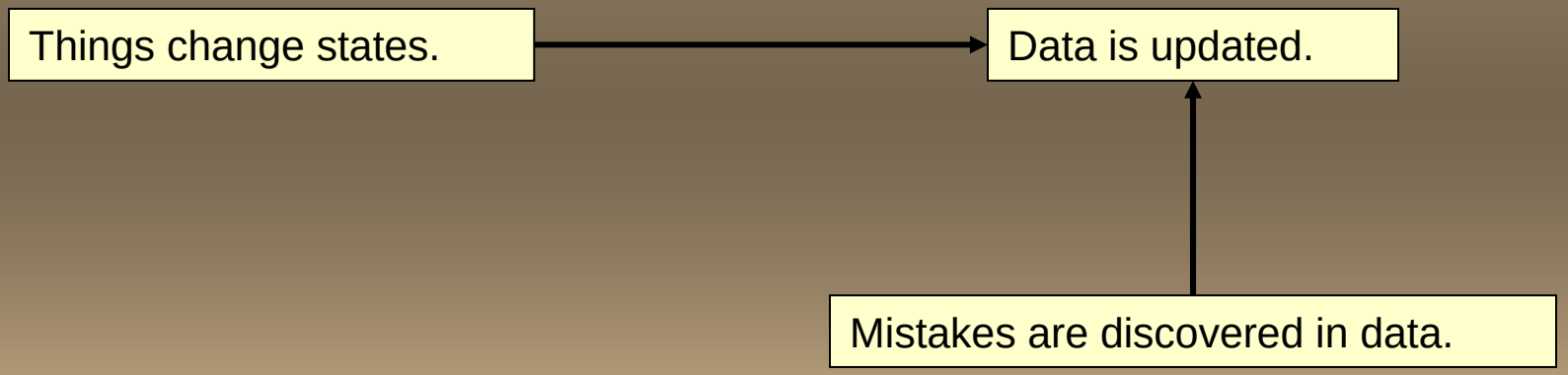
• Why DIY when a vendor can do it for you???

Part II
Two Kinds of Changes

TWO KINDS OF CHANGES

Changes that happen to things.

Changes that happen to data.



TWO KINDS OF CHANGES

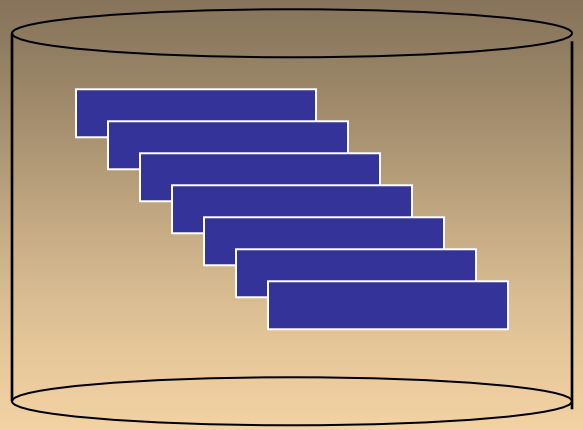
Things change states.

Data is updated.

Version tables and history tables keep a before-image of updated data.

Mistakes are discovered in data.

But they do not, and cannot, distinguish state changes from corrections.



This loses important information!

Bitemporal data preserves that information.

TWO "WHEN" QUESTIONS

When were things like that?

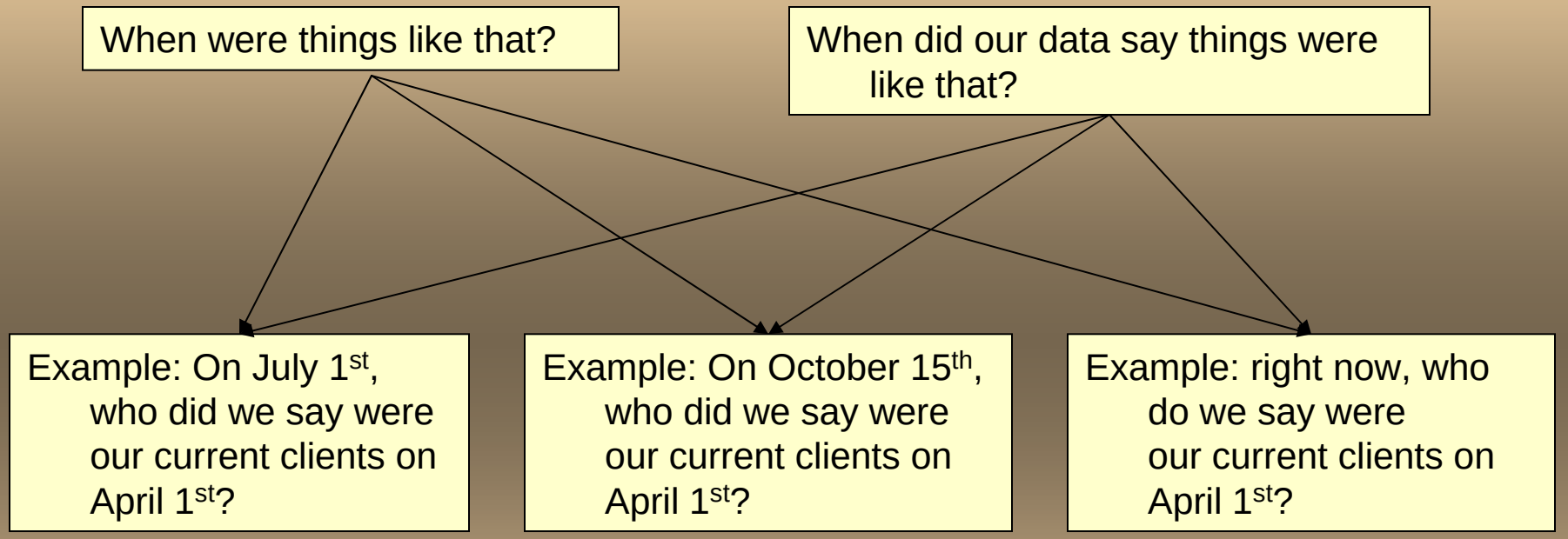
Example: On last April 1st, who were our then current clients?

When did our data say that things were like that?

Example: On last April 1st, who were listed in our Client table as current clients?

These are completely different questions!

TWO "WHEN" QUESTIONS



These are completely different questions!

Part III
Non-Temporal Data

NON-TEMPORAL DATA

clnt-id	txn-dt	name
C-335	09/01/10	Smith

Today is: 9/18/10.

Things This Data Doesn't Tell Us.

- When C-335 first became a client.
- Whether or not C-335 always had the name "Smith".
- How often data about C-335 has been modified.
- Whether or not C-335 was a client on, e.g. July 1st, 2010, and if so, what her name was then.

NON-TEMPORAL DATA

clnt-id	txn-dt	name
C-335	11/22/10	Jones

Today is: 11/22/10.
Update client C-335's name to Jones.

Things This Data Doesn't Tell Us.

- When C-335 first became a client.
- Whether or not C-335 always had the name "Jones".
- How often data about C-335 has been modified.
- Whether or not C-335 was a client on, e.g. July 1st, 2010, and if so, what her name was then.

NON-TEMPORAL DATA

clnt-id	txn-dt	name
C-335	11/22/10	Jones

Today is: 11/22/10.
Update client C-335's name to Jones.

Information That Has Been Lost.

- That C-335 was a client prior to this txn date.
- that C-335's name was "Smith" from 09/01/10 to 11/22/10.

NON-TEMPORAL DATA

clnt-id	txn-dt	name
C-335	11/22/10	Jones

Today is: 02/19/11.
Delete client C-335.

Things This Data Doesn't Tell Us.

- When C-335 first became a client.
- Whether or not C-335 always had the name "Smith".
- How often data about C-335 has been modified.
- Whether or not C-335 was a client on, e.g. July 1st, 2010, and if so, what her name was then.

NON-TEMPORAL DATA

clnt-id	txn-dt	name

Today is: 02/19/11.
Delete client C-335.

Information That Has Been Lost.

- That C-335 was a client prior to this txn date.
- that C-335's name was "Smith" from 09/01/10 to 11/22/10.

NON-TEMPORAL DATA

clnt-id	txn-dt	name

Today is: 02/19/11.
Delete client C-335.

Information That Has Been Lost.

- That there ever was a client C-335.

Part IV
Point in Time Versioning

POINT IN TIME VERSIONING

clnt-id	txn-dt	name
C-335	09/01/10	Smith
C-335	11/22/10	Jones

Today is: 11/22/10.
Update client C-335's name to Jones.

Information That is Available.

- When C-335 first became a client.
- Whether or not C-335 always had the name "Jones".
- How often data about C-335 has been modified.
- Whether or not C-335 was a client on, e.g. October 1st, 2010, and if so, what her name was then.

POINT IN TIME VERSIONING

clnt-id	txn-dt	name
C-335	09/01/10	Smith
C-335	11/22/10	Jones
C-335	9999	Jones

Today is: 12/21/10.
Delete client C-335.

Information That is Available.

- When C-335 first became a client.
- Whether or not C-335 always had the name “Jones”.
- How often data about C-335 has been modified.

POINT IN TIME VERSIONING

clnt-id	txn-dt	name
C-335	09/01/10	Smith
C-335	11/22/10	Jones
C-335	9999	Jones

Today is: 12/21/10.
Delete client C-335.

Information That is Not Available.

- When C-335 was deleted. And, as a result:
- Whether C-335 was a client on any date from 11/23/10 forwards to today.

POINT IN TIME VERSIONING

clnt-id	txn-dt	name
C-335	09/01/10	Smith
C-335	11/22/10	Jones
C-335	9999	Jones
C-335	03/22/11	Jones

Today is: 03/22/11.
Insert client C-335.

Information That is Available.

- When C-335 first became a client.
- Whether or not C-335 always had the name “Jones”.
- How often data about C-335 has been modified.

POINT IN TIME VERSIONING

clnt-id	txn-dt	name
C-335	09/01/10	Smith
C-335	11/22/10	Jones
C-335	9999	Jones
C-335	03/22/11	Jones

Today is: 03/22/11.
Insert client C-335.

Information That is Not Available.

- When C-335 was deleted. And, as a result:
- How long it took us to get C-335 back as a client.
- Whether C-335 was a client on any date from 11/23/10 to 3/21/11.

Part V
Delete Flag Versioning

DELETE FLAG VERSIONING

clnt-id	txn-dt	name	del-flg
C-335	09/01/10	Smith	N
C-335	11/22/10	Jones	N
C-335	12/21/10	Jones	Y
C-335	03/22/11	Jones	N

Information That is Available.

- When C-335 was deleted. And, as a result:
- How long it took us to get C-335 back as a client.
- Whether C-335 was a client on any date from 11/23/10 to 3/21/11.

DELETE FLAG VERSIONING

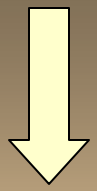
clnt-id	txn-dt	name	del-flg
C-335	09/01/10	Smith	N
C-335	11/22/10	Jones	N
C-335	12/21/10	Jones	Y
C-335	03/22/11	Jones	N

But Now Things are Getting Complicated.

- Queries for clients at a point in time must specify the delete flag as well as the point in time.
- For example: Show all clients at any time in November or December 2010:

DELETE FLAG VERSIONING

clnt-id	txn-dt	name	del-flg
C-335	09/01/10	Smith	N
C-335	11/22/10	Jones	N
C-335	12/21/10	Jones	Y
C-335	03/22/11	Jones	N

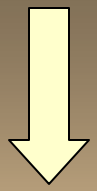


clnt-id	txn-dt	name
C-335	11/22/10	Jones
C-335	12/21/10	Jones

```
Select clnt-id, txn-dt, name
   From Clients
   Where '11/01/10' <= txn-dt
       And txn-dt < '01/01/11'
      And del-flg = 'N'
```


DELETE FLAG VERSIONING

clnt-id	txn-dt	name	del-flg
C-335	09/01/10	Smith	N
C-335	11/22/10	Jones	N
C-335	12/21/10	Jones	Y
C-335	03/22/11	Jones	N



clnt-id	txn-dt	name
C-335	11/22/10	Jones
C-335	12/21/10	Jones

But this is incomplete. C-335, with name “Smith”, was a client from November 1st to November 21st.

We could make this query work.

But what we really need is a better version table.

Part VI
Period of Time Versioning

PERIOD OF TIME VERSIONING

clnt-id	eff-beg-dt	eff-end-dt	name	txn-dt
C-335	09/01/10	11/22/10 9999	Smith	09/01/10
C-335	11/22/10	9999	Jones	11/22/10

Today is: 11/22/10.
Update client C-335's name to Jones.

PERIOD OF TIME VERSIONING

clnt-id	eff-beg-dt	eff-end-dt	name	txn-dt
C-335	09/01/10	11/22/10	Smith	09/01/10
C-335	11/22/10	12/21/10	Jones	11/22/10

Today is: 12/21/10.
Delete client C-335.

PERIOD OF TIME VERSIONING

clnt-id	eff-beg-dt	eff-end-dt	name	txn-dt
C-335	09/01/10	11/22/10	Smith	09/01/10
C-335	11/22/10	12/21/10	Jones	11/22/10
C-335	03/22/11	9999	Jones	03/22/11

Today is: 03/22/11.
Insert client C-335.

Period of time versioning seems to work pretty well.

This kind of data answers all the questions we've asked so far.

But suppose a version contains a mistake.

How do you correct a version?

Part VII
Versioning vs. Corrections

VERSIONS VS. CORRECTIONS

clnt-id	eff-beg-dt	eff-end-dt	name	txn-dt
C-335	09/01/10	11/22/10	Smith	09/01/10
C-335	11/22/10	12/21/10	Jones	11/22/10
C-335	03/22/11	9999	Jones	03/22/11

Today is: 03/29/11.
Make C-335's name "Brown" from 03/22/11.

How do you correct a version?

- When C-335 returned, on March 22nd, her name was "Brown", not "Jones".

VERSIONS VS. CORRECTIONS

clnt-id	eff-beg-dt	eff-end-dt	name	txn-dt
C-335	09/01/10	11/22/10	Smith	09/01/10
C-335	11/22/10	12/21/10	Jones	11/22/10
C-335	03/22/11	9999	Brown	03/29/11

Today is: 03/29/11.
Make C-335's name "Brown" from 03/22/11.

But now we've lost the information that, for a week, the database said that her name was still "Jones".

- We have as-is data about that past week.
- But we no longer have the original as-was data.
- We can no longer answer the question, e.g. "What did the report that was run on March 28th say that C-335's name was on March 25th?

Part VIII
Period of Time PK Versioning

PERIOD OF TIME PK VERSIONING

clnt-id	eff-beg-dt	eff-end-dt	name	txn-dt
C-335	09/01/10	11/22/10	Smith	09/01/10
C-335	11/22/10	9999	Jones	11/22/10

Today is: 11/22/10.
Update client C-335's name to Jones.

- We have already lost some information.
- Yesterday, a query would have said that C-335's name, today, will be "Smith".
- Today, that same query will say that C-335's name, today, is "Jones".
- We have lost as-was data about today.

Part IX
Bitemporal Data

BITEMPORAL DATA

clnt-id	eff-beg-dt	eff-end-dt	name	txn-dt
C-335	09/01/10	11/22/10	Smith	09/01/10
C-335	11/22/10	9999	Jones	11/22/10

- The time period we have keeps track of changes to client C-335. But it does not keep track of corrections to data about her.
- For that, we need a second time period.

BITEMPORAL DATA

clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	9999	Jones

Today is: 11/22/10.
Update client C-335's name to Jones.

BITEMPORAL DATA

clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones

Today is: 02/19/11.
Delete client C-335.

BITEMPORAL DATA

clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones
C-335	03/22/11	9999	03/22/11	9999	Jones

Today is: 03/22/11.
Insert client C-335.

BITEMPORAL DATA

clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones
C-335	03/22/11	9999	03/22/11	03/29/11	Jones
C-335	03/22/11	9999	03/29/11	9999	Brown

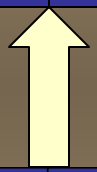
Today is: 03/29/11.
Make C-335's name "Brown" from 03/22/11.

Part X
Hiding the Complexities With Views

VIEWS: A VERSION TABLE

clnt-id	eff-beg-dt	eff-end-dt	name
C-335	09/01/10	11/22/10	Smith
C-335	11/22/10	02/19/11	Jones
C-335	03/22/11	9999	Jones

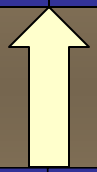
Today is: 3/28/11.



clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones
C-335	03/22/11	9999	03/22/11	03/29/11	Jones
C-335	03/22/11	9999	03/29/11	9999	Brown

VIEWS: A VERSION TABLE

clnt-id	eff-beg-dt	eff-end-dt	name
C-335	09/01/10	11/22/10	Smith
C-335	11/22/10	02/19/11	Jones
C-335	03/22/11	9999	Brown

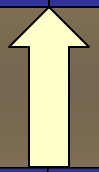


Today is: 3/29/11.

clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones
C-335	03/22/11	9999	03/22/11	03/29/11	Jones
C-335	03/22/11	9999	03/29/11	9999	Brown

VIEWS: A NON-TEMPORAL TABLE

clnt-id	name
C-335	Smith



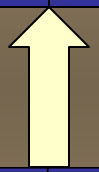
Today is: 10/13/10.

clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones
C-335	03/22/11	9999	03/22/11	03/29/11	Jones
C-335	03/22/11	9999	03/29/11	9999	Brown

VIEWS: A NON-TEMPORAL TABLE

Today is: 3/28/11.

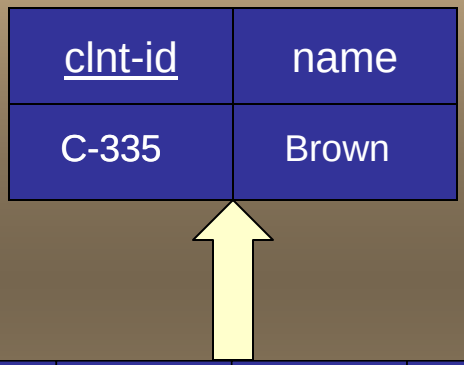
clnt-id	name
C-335	Jones



clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones
C-335	03/22/11	9999	03/22/11	03/29/11	Jones
C-335	03/22/11	9999	03/29/11	9999	Brown

VIEWS: A NON-TEMPORAL TABLE

Today is: 3/29/11.



clnt-id	name
C-335	Brown

clnt-id	eff-beg-dt	eff-end-dt	asr-beg-dt	asr-end-dt	name
C-335	09/01/10	9999	09/01/10	11/22/10	Smith
C-335	09/01/10	11/22/10	11/22/10	9999	Smith
C-335	11/22/10	9999	11/22/10	02/19/11	Jones
C-335	11/22/10	02/19/11	02/19/11	9999	Jones
C-335	03/22/11	9999	03/22/11	03/29/11	Jones
C-335	03/22/11	9999	03/29/11	9999	Brown

Part XI
Some Important Views

VIEWS: CONVENTIONAL TABLES

- In a bitemporal table, conventional data are all and only those rows which are current in both time periods.
  - In which Now() is >= their begin points in time, and
  - in which Now() is < their end points in time.

- This provides full upward compatibility for conventional tables which are converted to bitemporal tables.
  - Existing maintenance transactions are unaffected.
  - Existing queries are unaffected.

VIEWS: HISTORY/VERSION TABLES

- In a bitemporal table, historical versions are all and only those rows which are in the table with a past point or range in effective time, at one point in assertion time.
    - In which Now() is >= the assertion begin point in time, and
    - in which Now() is < the assertion end point in time.
- This works for current versions and future versions as well.

- This provides full upward compatibility for existing history/version tables which are converted to bitemporal tables.
    - Existing maintenance transactions are unaffected.
    - Existing queries are unaffected.

VIEWS: TABLE-SPECIFIC LOGFILE DATA

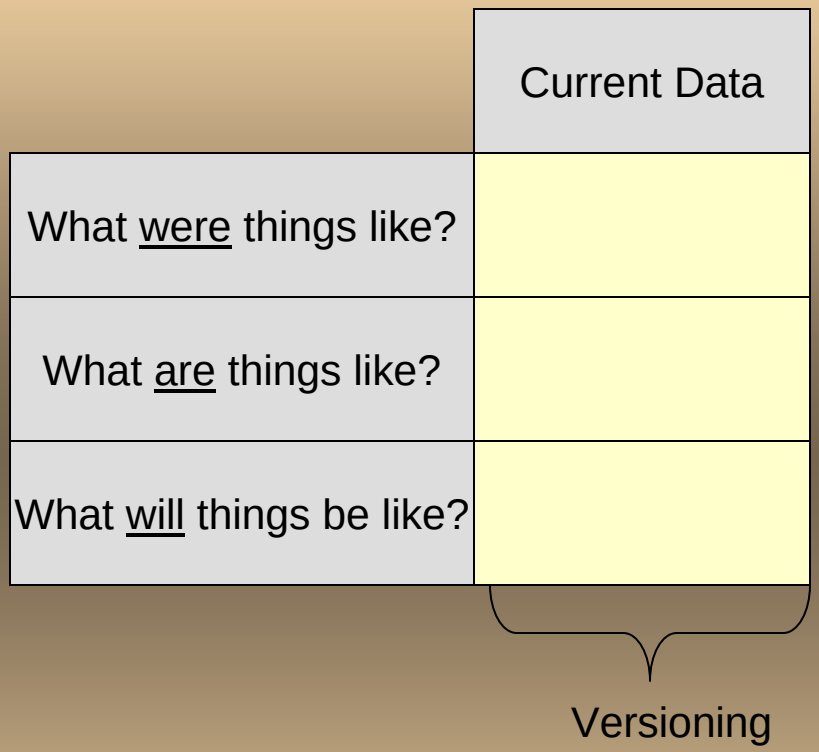
In a bitemporal table, logfile data (a reconstruction of the table as of a past point in time) are all and only those rows which are in the table as of that past point in assertion time. These rows may be non-temporal or versioned.

- In which the point in time is >= the assertion begin point in time.

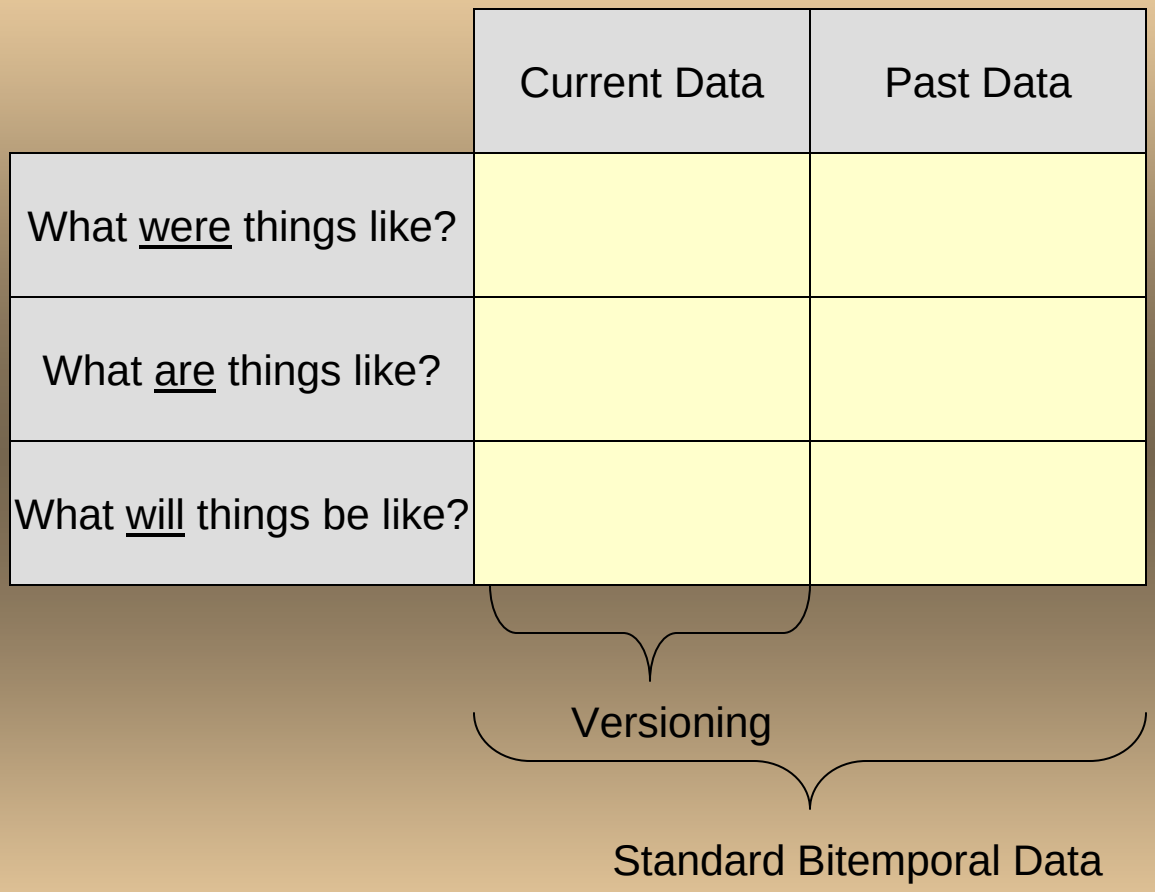
This provides the ability to run queries which support the re-creation of reports run in the past, about the past, present, or future, with the exact same results. Data for such re-creations is as-was data. Current data about the past, present, or future, is as-is data.

Part XII
Versioning,
Standard Bitemporal Data,
Extended Bitemporal Data

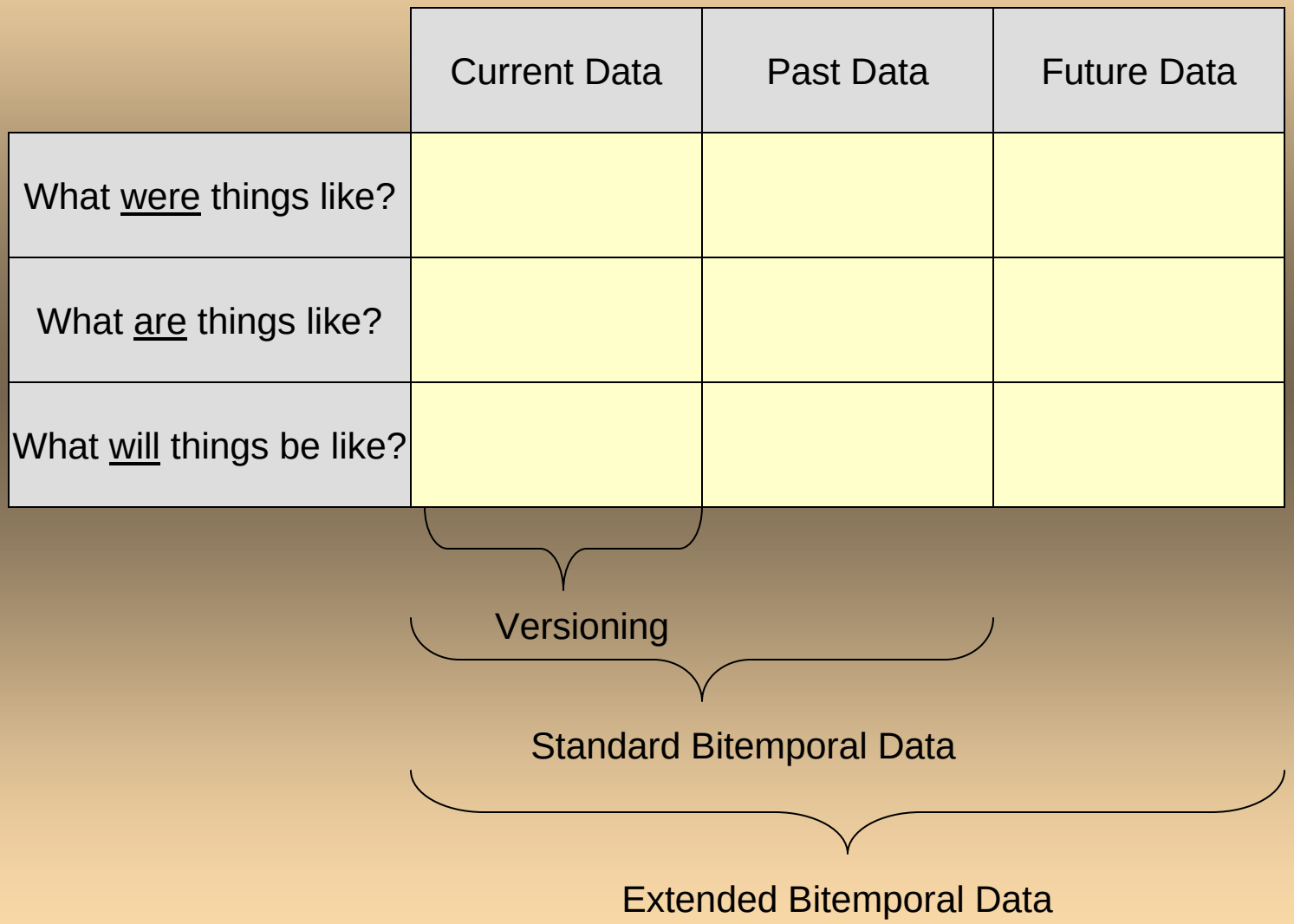
VERSIONING



STANDARD BITEMPORAL DATA



EXTENDED BITEMPORAL DATA



Part XIII
Conclusions

CONCLUSION

Bitemporal data management is becoming mainstream. DBMS and other vendors support it right now!

A vendor solution is an enterprise solution, replacing different home-grown ways of supporting temporal requirements.

The maintenance of bitemporal data is complex. One temporal transaction may take a dozen or more physical SQL transactions to complete.

Vendor solutions hide the complexity, by mapping the one temporal transaction to the multiple physical transactions “under the hood”.

QUESTIONS

1. How do Kimball's slowly changing dimensions compare to bitemporal data's way of keeping track of changes?

2. The maintenance of bitemporal data is complex. One temporal transaction may take a dozen or more physical SQL transactions to complete. Do we have to write those physical transactions? If not, how do we avoid it?

3. These bitemporal tables are complicated! How do we hide that complexity for queries against bitemporal data?

4. Rows in bitemporal tables contain a lot of extra columns. Will every change result in one or more new rows? If so, will storage space be a problem if we use bitemporal data?

5. How can bitemporal data be made to perform as well as non-temporal data?

6. How much work will be involved in swapping out DIY solutions for vendor-provided bitemporal solutions?

