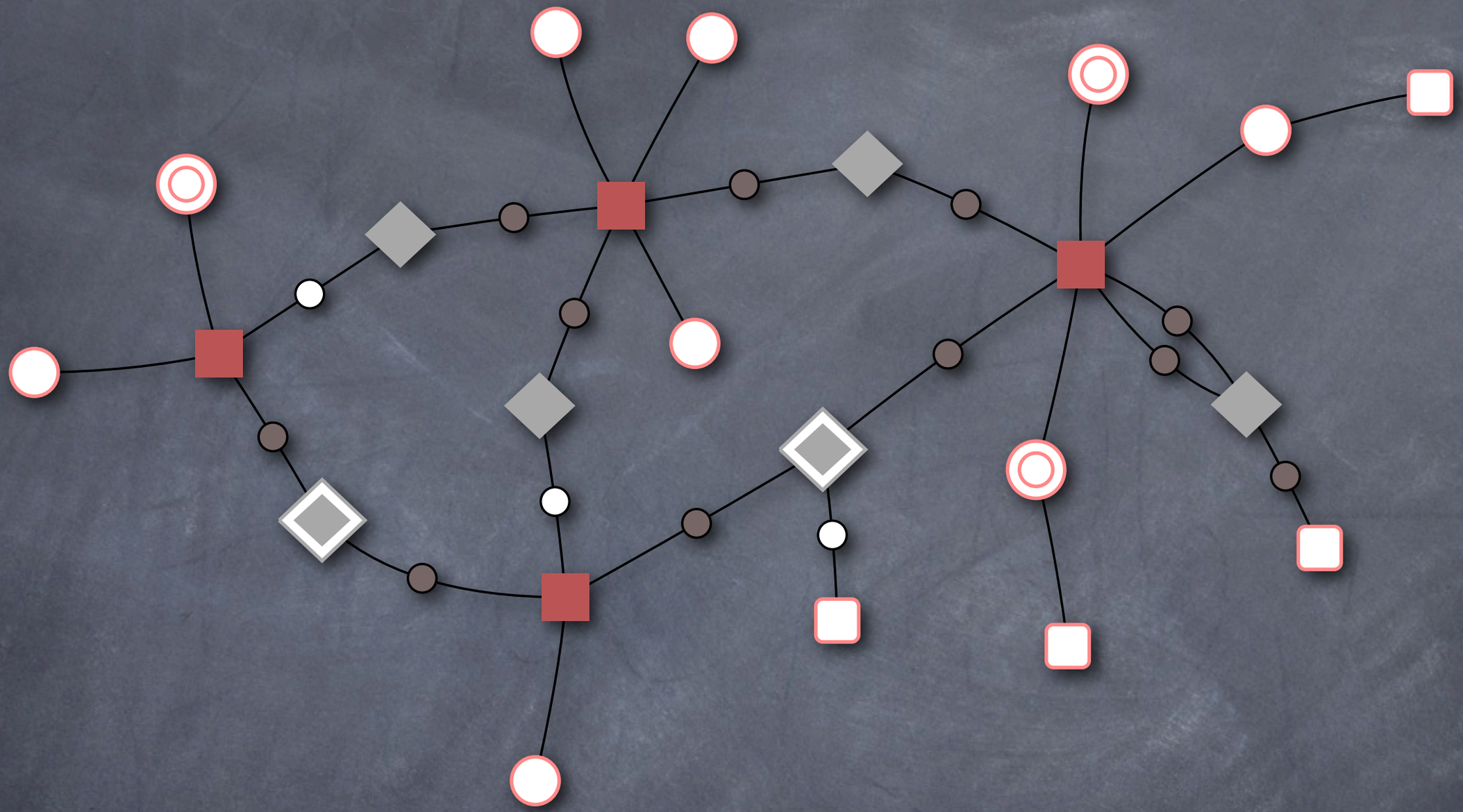


Anchor Modeling

A Technique for Information under Evolution

Lars Rönnbäck @Ordina 6/12, 2011

Anchor Modeling...

Pitches

- has a solid theoretical foundation.
- is based on well known principles.
- shortens implementation time.
- reduces maintenance costs.
- is agile.
- preserves old versions of the database.
- is temporal and optionally bi-temporal.
- is simple to learn.
- is hard to make mistakes with.
- often has better performance.
- is free and tools are Open Source.

Heraclitus
500.BC

"Panta rhei"
Everything flows

modeling = categorizing information into

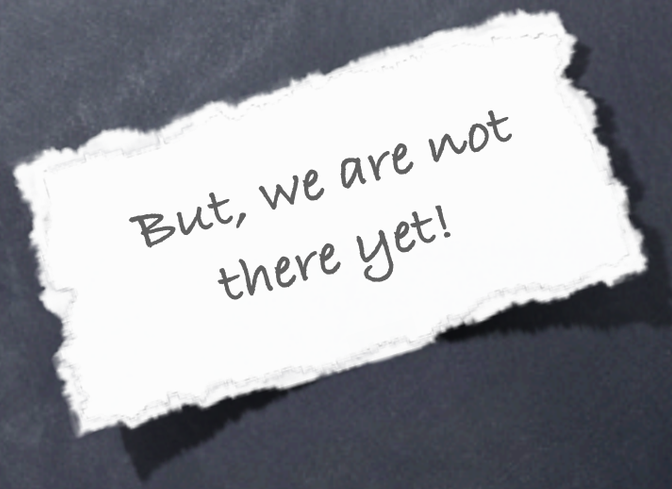
Imagine a satellite photo covering the area around this building...

- content
- structure
- constraints
- origins
- reliability
- resolution
- quality
- frequency
- cost
- interpretation

Information evolves in many ways...

What is a database?

- The purpose of a database is to store a body of information and allow searches over it.
- The purpose of a temporal database is to store a body of information under evolution and allow historical searches over it.



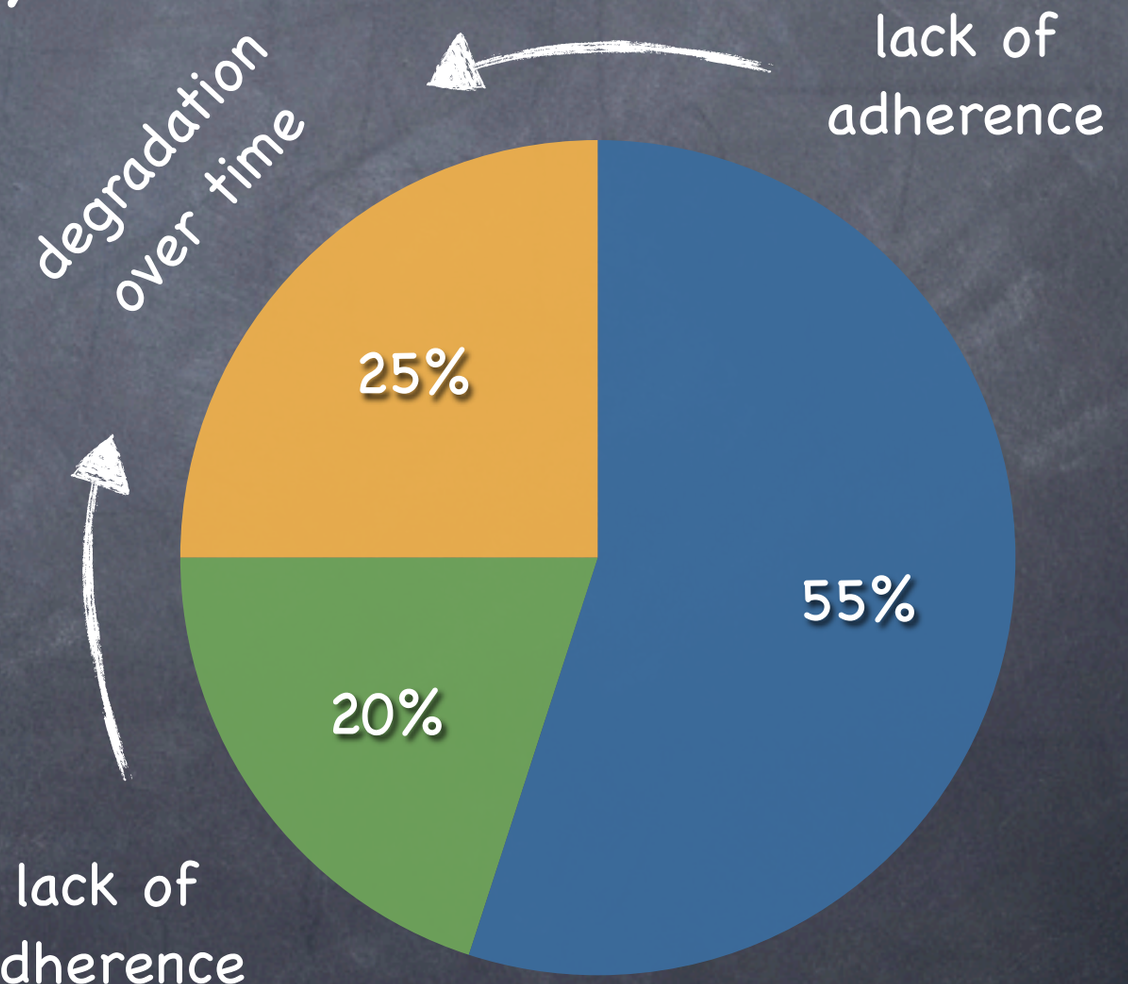
What is a Data Warehouse?

The most interesting questions are the ones we do not yet know we will ask!

- It is a database that:
  - integrates information from many sources
  - has a unified and well defined model
  - calculates and stores new information
  - provides means for asking complex questions
  - can do all this "fast enough"

The dilemma

- Many sources and many users naturally result in many changes.



Patch or Redo?

- Patching initially works, but...
- maintenance costs usually rise proportionally to the lifetime of the data warehouse.
- Meaning that:
  Redoing is unavoidable at some point!
  (and for dimensional modeling sometimes accounted for)
- Studies show that the average lifetime is five years

Don't let your **DW** turn into a **JBOT**
– Just a Bunch Of Tables

What is Anchor Modeling?

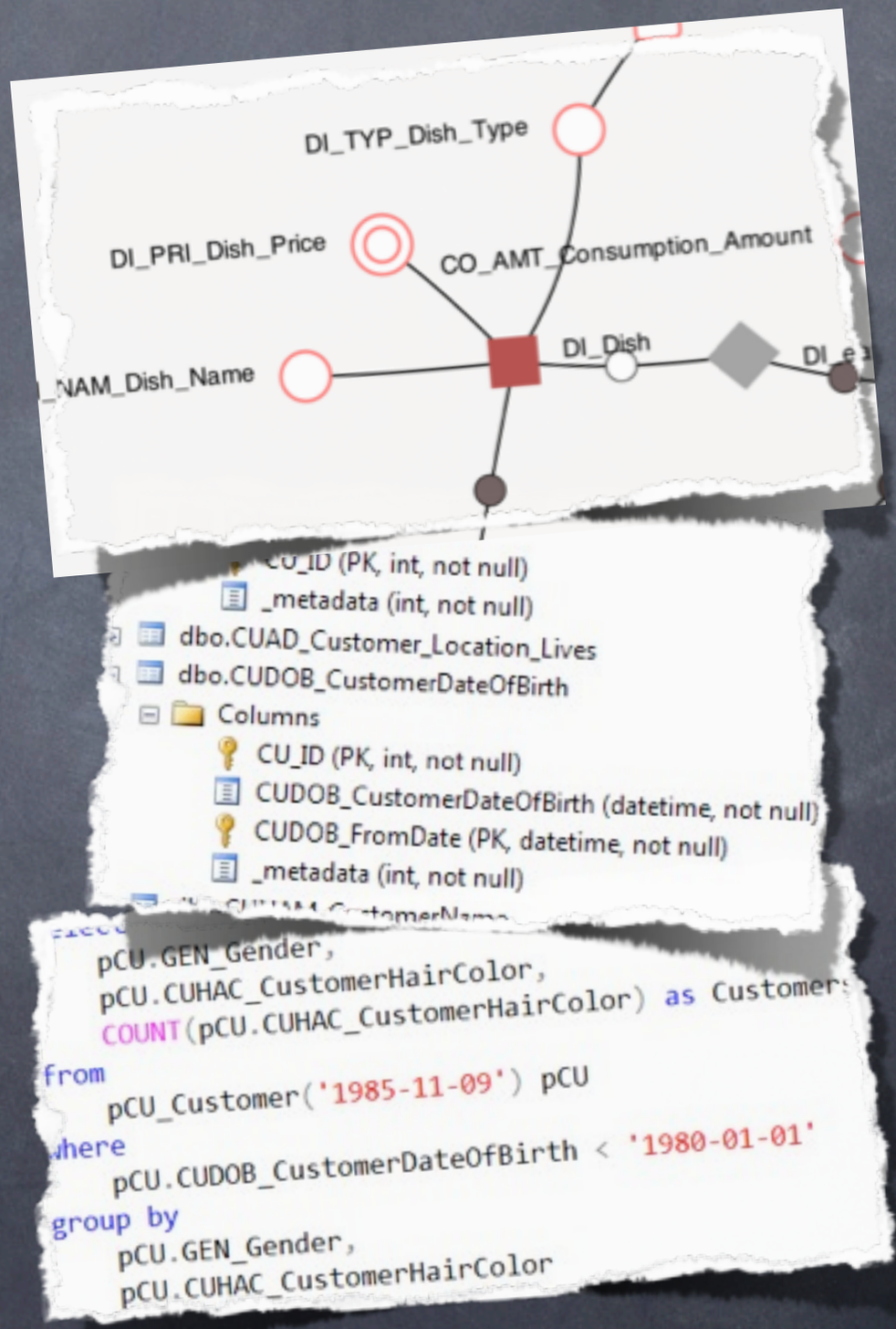
- Anchor Modeling combines normalization and emulation to provide an agile database modeling technique for evolving information that is implementable in current relational databases.
- Most, if not all, of what Anchor Modeling is doing in its physical (relational) representation could be "hidden" from the end-user in a true temporal database.

Technologies

- Entity-Relationship Modeling

one-to-one

- Sixth Normal Form Tables
- Temporal Database Emulation



History

Best Paper Award
@ ER'09

Paul Johannesson

Lars Rönnbäck

Olle Regardt

Maria Bergholtz

Petia Wohed

research

consulting

community

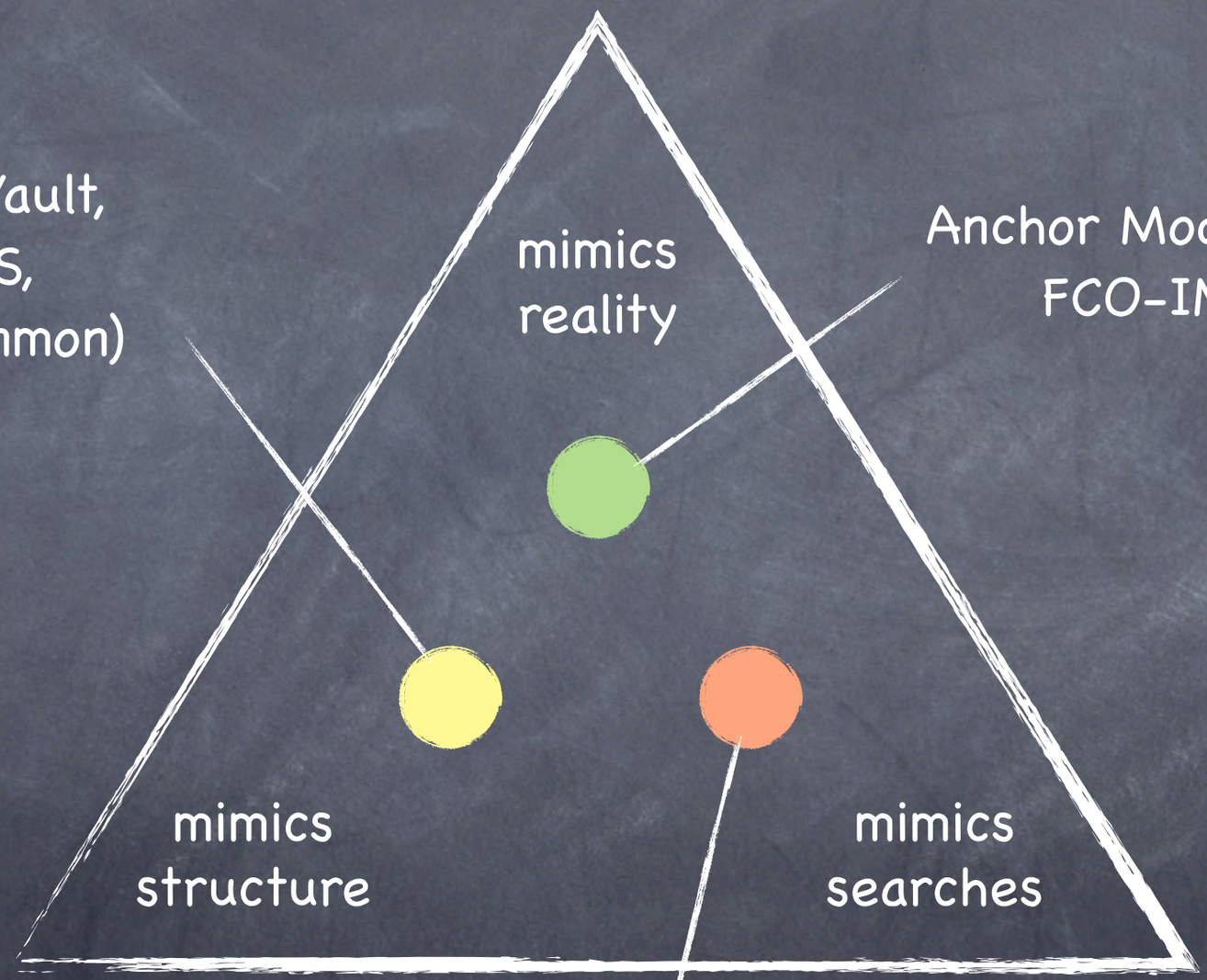
DW	MDM	EDW	DW	TDWI	SU WWW	DW ER09	DKE TOOL	DW GSE AMW	MVDW ?
03	04	05	06	07	08	09	10	11	12

Philosophy

- Make modeling free from assumptions
  (immutable surrogates, volatile naturals, query agnostic)
- Make modeling agile and iterative
  (non-destructive schema/content evolution)
- Do not duplicate information
  (normalization, decomposition, power types)
- Do not alter existing information
  (use only inserts, temporalization, concurrency)
- Provide a simple interface for queries
  (temporal perspectives, insert/update/delete triggers)
- Decouple metadata from the model
  (another anchor model referenced from data)

Positioning
Anchor Modeling

Domain driven modeling



Data Vault,
ODS,
3NF (Inmon)

Anchor Modeling,
FCO-IM

Data driven modeling

Use-case driven modeling

Basic Notions

Attributes - properties

Example: The surname of a Person

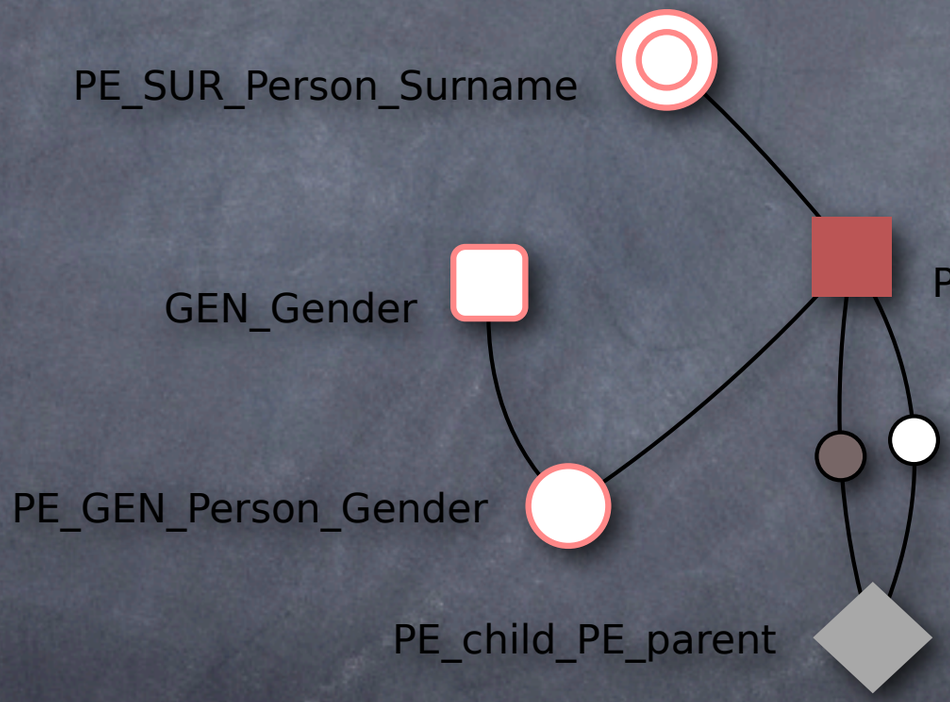
<#42, ‘Rönnbäck’, 2004-06-19>

Anchors - entities

Example: A Person

<#42>

Anchors are keepers of immutable identities.



filled = primary

Knots - shared properties

Example: The gender of a Person

<#1, ‘Male’> + <#42, #1>

Ties - relationships

Example: The children of a Person

<#42, #4711>

Naming convention

- **Anchors**
  Two letter mnemonic + descriptor
  `PE_Person`

  unique in the model

- **Knots**
  Three letter mnemonic + descriptor
  `GEN_Gender`

  unique on the anchor

- **Attributes**
  Inherited anchor mnemonic + three letter mnemonic + inherited anchor descriptor + descriptor
  `PE_SUR_Person_Surname`

- **Ties**
  Inherited anchor and knot mnemonics separated by the roles they play in the relationship
  `PE_child_PE_parent`

Temporal concepts

- **Changing time**
  (others: valid time, effective time)
  The time when entities change states, attributes change values or relationships change members.

"Your grade has been upgraded from C to A"

- **Recording time**
  (others: transaction time, assertion time)
  The period of time during which information about the domain was recorded in some kind of memory.

"Sorry, the A was meant for someone else"

- **Happening time**
  (others: user-defined time)
  The time of an event taking place in the domain being modeled.

"Your complaint on our grading has been duly noted"

Remember that satellite image?

The date when the photo was taken is:

Temporality depends on the domain being modeled

- **Happening time**
  If the photo is **IN** the domain

"We sell satellite imagery."

- **Changing time**
  (for what is depicted and different)
  If the photo is **OF** the domain

"We work with military intelligence."

- **Recording time**
  If the photo is **BY** the domain

"We write satellite operating systems."

Perspectives

They all look like **3NF**!

You interact with an anchor database using these.

- **Latest perspective**
  Shows the latest available information

- **Point-in-time perspective**
  Shows information as it was on the given timepoint

- **Interval perspective**
  Shows information changes that happened within the given interval

- **Natural perspective**
  Converts natural keys to surrogate identities
  (for composite keys these may span over several anchors)

Inserting data

```
insert into lPE_Person (
  PE_SUR_Person_Surname,
  PE_NAM_ChangedAt,
  PE_DOB_Person_DateOfBirth
) values ('Samuelsson', '1972-08-20', '1972-08-20');

update lPE_Person (
set
  PE_SUR_Person_Surname = 'Rönnbäck',
  PE_NAM_ChangedAt = '2004-06-19'
where
  PE_ID = 42;
```

An identity is created if not provided

Selecting data

```
select * from lPE_Person;
select * from pPE_Person('1999-12-31');
```

The **UPDATE** is translated to an **INSERT** and for bitemporal so is **DELETE**

Table elimination

- A table *T* can be removed from the execution plan if:

  a) no column from *T* is explicitly selected

  b) the number of rows in the returned data set is not affected by the join with *T*

The temporal perspectives regain all benefits from **6NF**!

Indexing

- All primary indexes are clustered indexes (index organized tables) which use no extra space

```
1|Peter    |2004-02-13 20:08
2|Paul     |2007-01-01 13:54
2|John     |2006-08-20 15:15
2|Matthew  |2002-10-15 13:20
2|Ringo    |2001-01-02 01:18
3|George   |2007-09-19 08:00
```

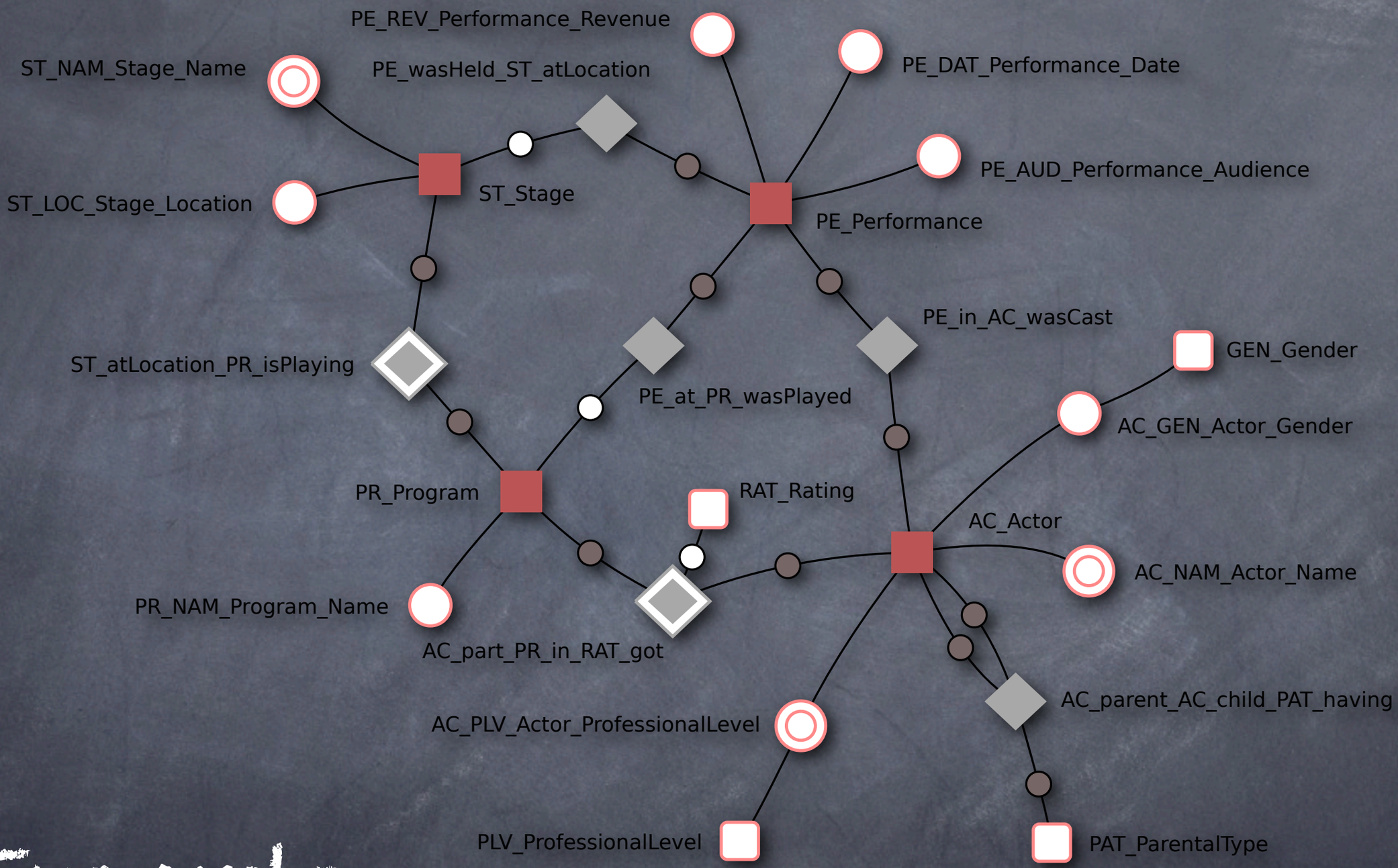
clustered index on identity + historization

- Secondary indexes are very rarely needed

Performance boosters

Performance gets comparatively better than less normalized techniques

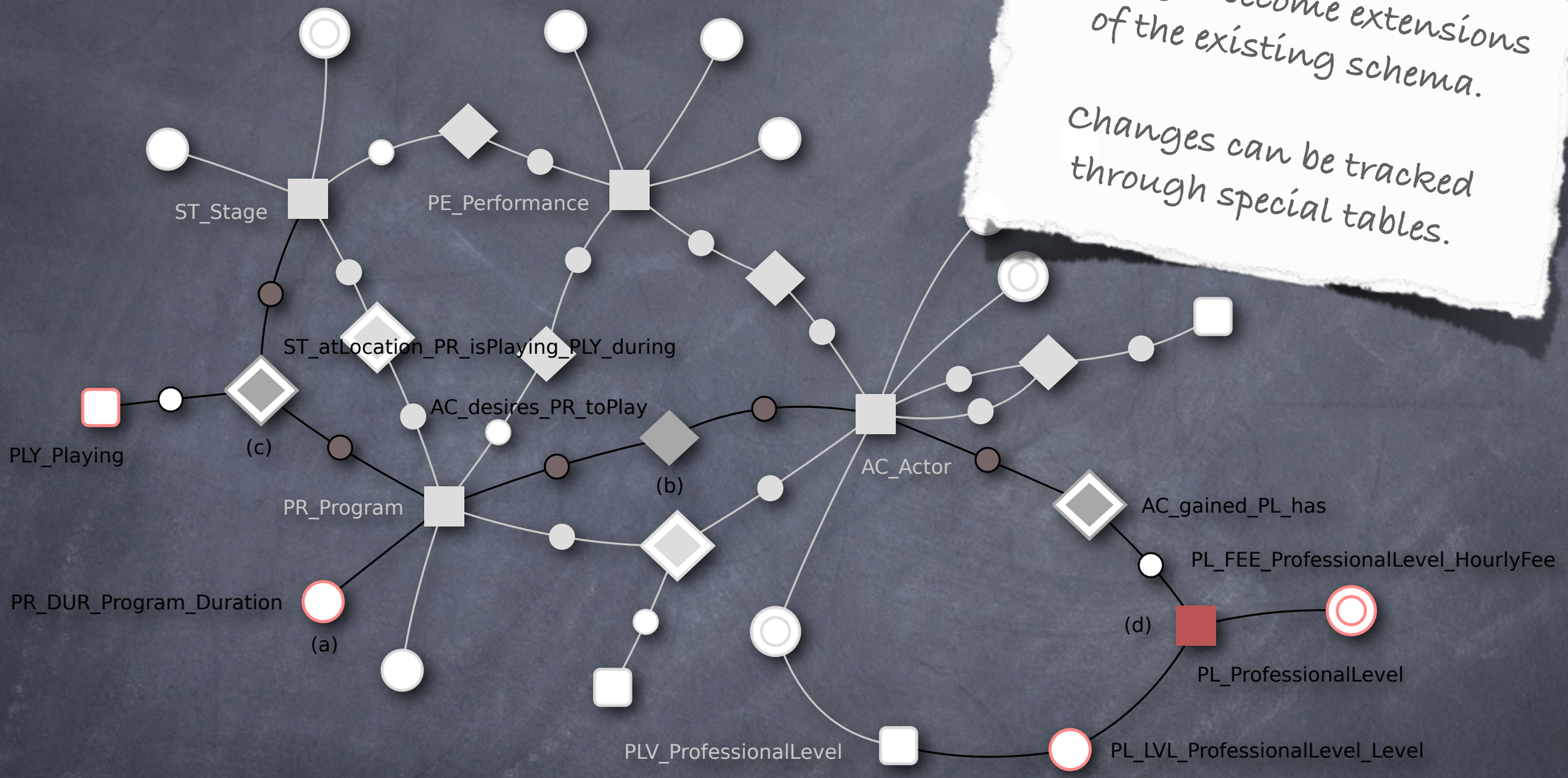
1. as models grow larger both in scope and volume.
2. when the content or structure is evolving over time.
3. when data is sparse (null values).
4. when the number of distinct values is small.
5. when table elimination can be utilized.
6. when intermediate result sets are small thanks to conditions in the query.



Example model

Changes become extensions of the existing schema.

Changes can be tracked through special tables.



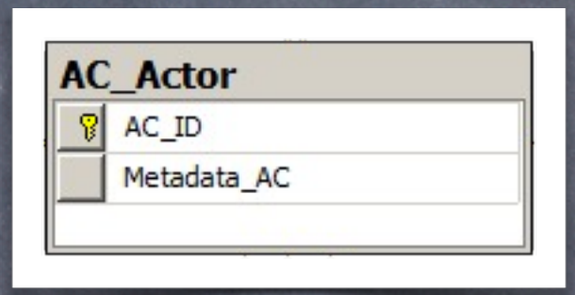
Schema evolution

all previous versions of the schema are available as subsets of the current schema

Modeling anchors

- **Guideline 1:** Use anchors for modeling core entities and transactions.

relational implementation:

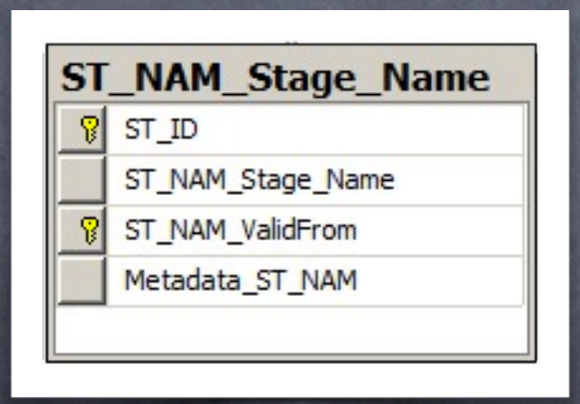


anchor

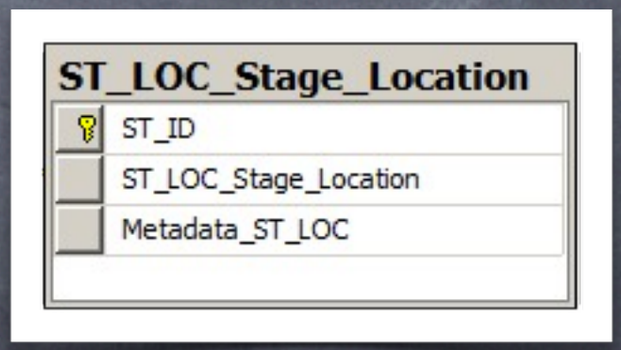
Modeling attributes

- **Guideline 2a:** Use a historized attribute if versioning of attribute values are of importance, otherwise use a static attribute.

relational implementation:



historized attribute

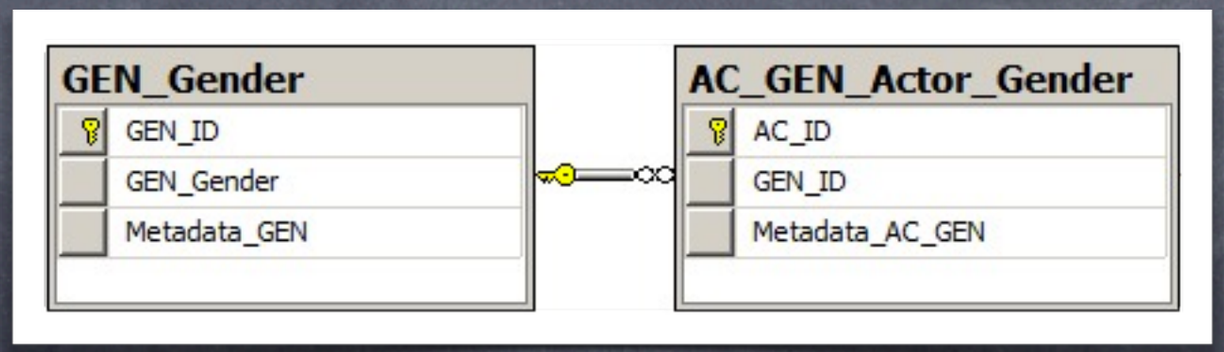


static attribute

Modeling attributes

- **Guideline 2b:** Use a knotted static attribute if attribute values represent categories or can take on only a fixed small set of values, otherwise use a static attribute.

relational implementation:

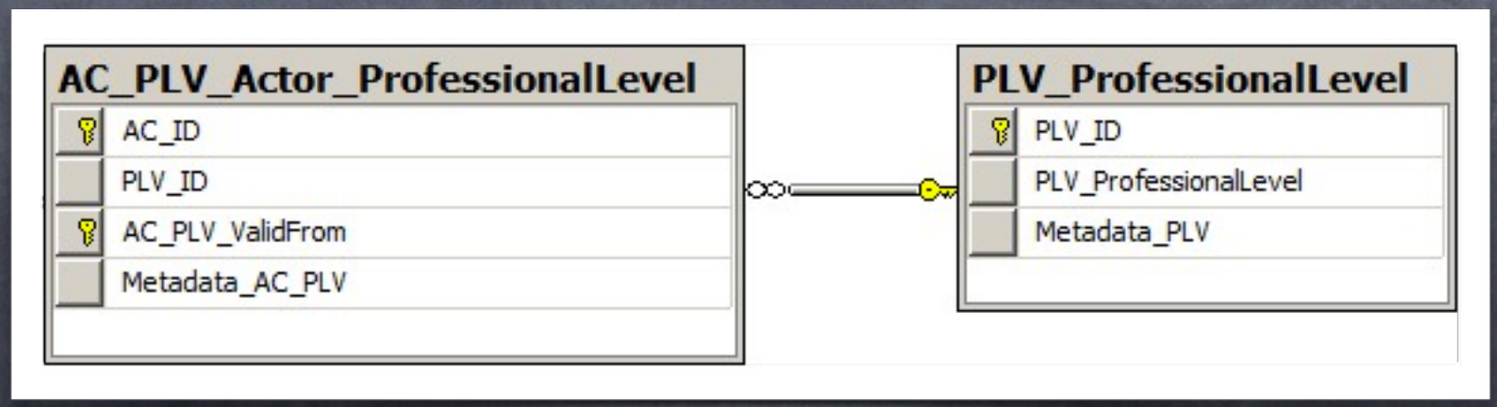


knotted static attribute

Modeling attributes

- **Guideline 2c:** Use a knotted historized attribute if attribute values represent categories or a fixed small set of values and the versioning of these are of importance.

relational implementation:



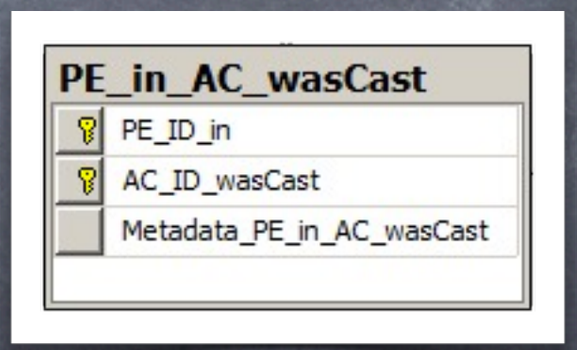
knotted historized attribute

Modeling ties

- **Guideline 3a:** Use a historized tie if a relationship may change over time, otherwise use a static tie.

relational implementation:

historized tie

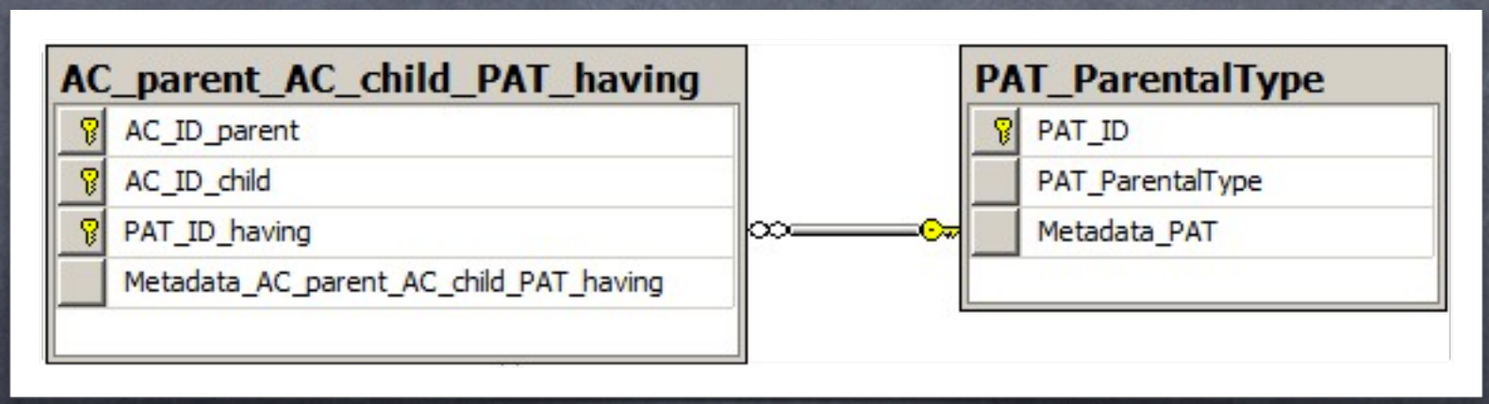


static tie

Modeling ties

- **Guideline 3b:** Use a knotted static tie if the instances of a relationship belong to certain categories, otherwise use a static tie.

relational implementation:

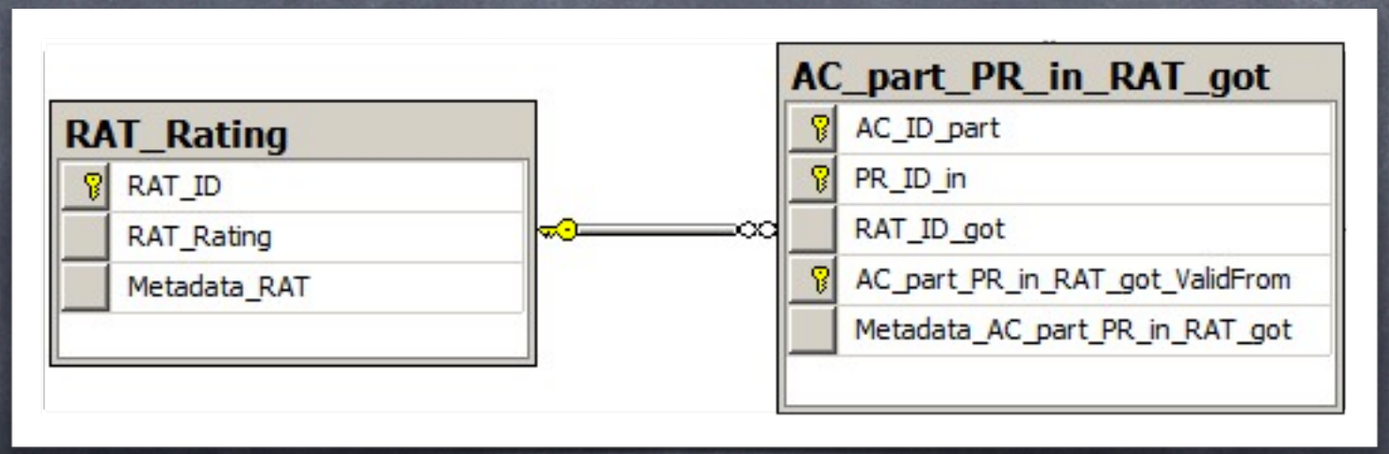


knotted static tie

Modeling ties

- **Guideline 3c:** Use a knotted historized tie if the instances of a relationship belong to certain categories and the relationship may change over time.

relational implementation:

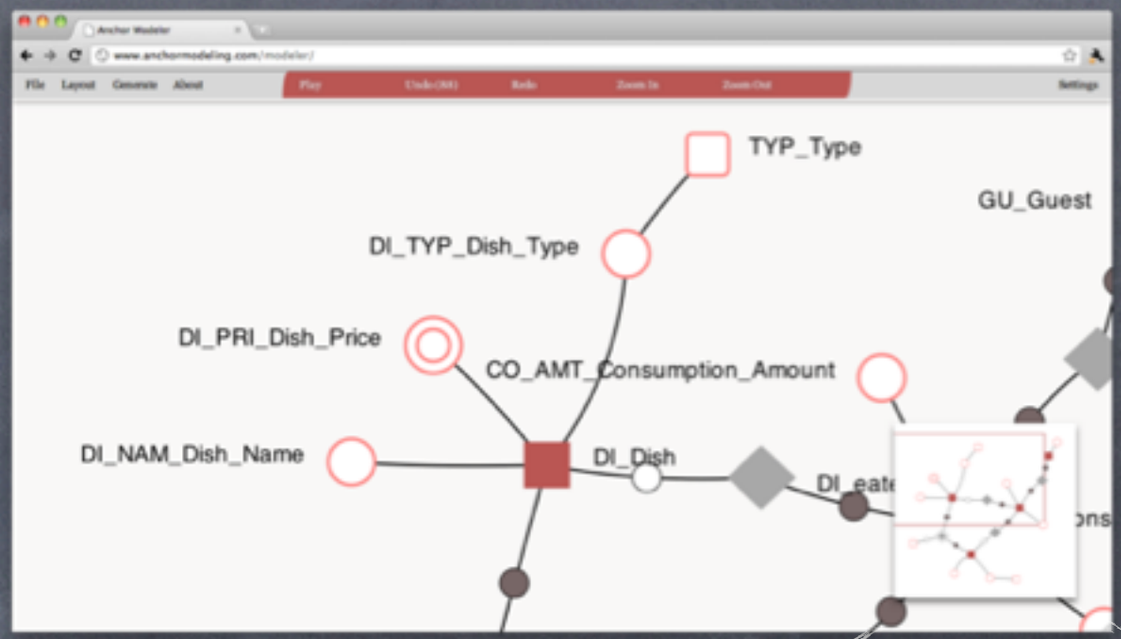


knotted historized tie

The Modeling Tool

- Open Source
- Online (**HTML5**)
- Free to use
- In the Cloud
- XML Interchange Format
- Automatic generation of SQL scripts
- Interactive (force-directed) Layout Engine

www.anchormodeling.com/modeler



DEMO!

Important Benefits

- Handles evolving information (keeping the integrity intact)
- Increases longevity (databases with long life expectancy)
- Simplifies modeling concepts (less prone to error)
- Enables modular and iterative development
- Needs no translation logic to the physical layer
- Automates generation of scripts
- No downtime when upgrading databases
- Scans only relevant data during searches
- Sparse data cause no gaps (no null values)

Future research

- Bitemporal Anchor Modeling
- Concurrent Anchor Modeling
- Case studies
- Benchmarking
- Tool development
  - Supporting other databases
  - More example models
  - Improved cloud functionality
    - collaboration, social features, rankings

Your chance to participate!

More Information

Homepage:
http://www.anchormodeling.com

Blog . Forum . Tutorials . Modeling Tool

Twitter: **anchormodeling**

E-mail: **lars.ronnback@anchormodeling.com**

LinkedIn Groups:
Anchor Modeling
Temporal Data Modeling
Temporal Data