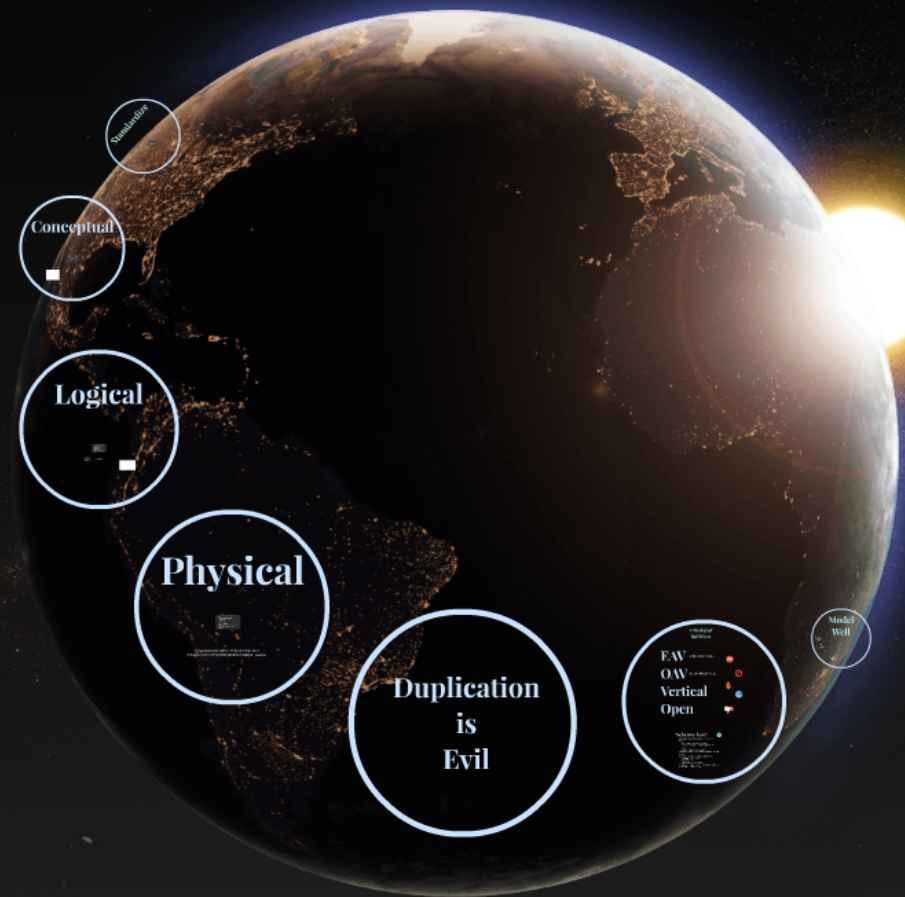


Save the Company by Modeling Data Right



Save the Company by Modeling Data Right



Bill Coulam

- dbsherpa.com and dbartisans.com
- Twitter: @billcoulam LinkedIn: billcoulam
- bill.coulam@dbsherpa.com
- C, C++, Java, JavaScript and PL/SQL since 1995
- Oracle data/database design and tuning since 1997
- Passionate about best programming/design practices
- Andersen Consulting - San Francisco, Denver, Herndon VA
- New Global Telecom - Denver, CO
- The Structure Group - Houston, TX
- Church of Jesus Christ of Latter Day Saints - SLC, UT
- Speaker at RMOUG, IOUG, ODTUG and UTOUG since 2001
- 2015 Oracle Developer of the Year nominee

Involve users early and often

Reduce and re-use.
Don't repeat.

Simplify.

Test with plenty
of dirty data

Get another pair of eyes

Document It !

Save the Company by Modeling Data Right



Standardize

- Establish a naming standard
- Establish modeling standard
- Choose a modeling tool
- Choose a versioning system
- Choose a DB devOps tool
- Ensure the standard, tool and systems are used



Standardize

- Establish a naming standard
- Establish modeling standard
- Choose a modeling tool
- Choose a versioning system
- Choose a DB DevOps tool
- Ensure the standard, tool and systems are used

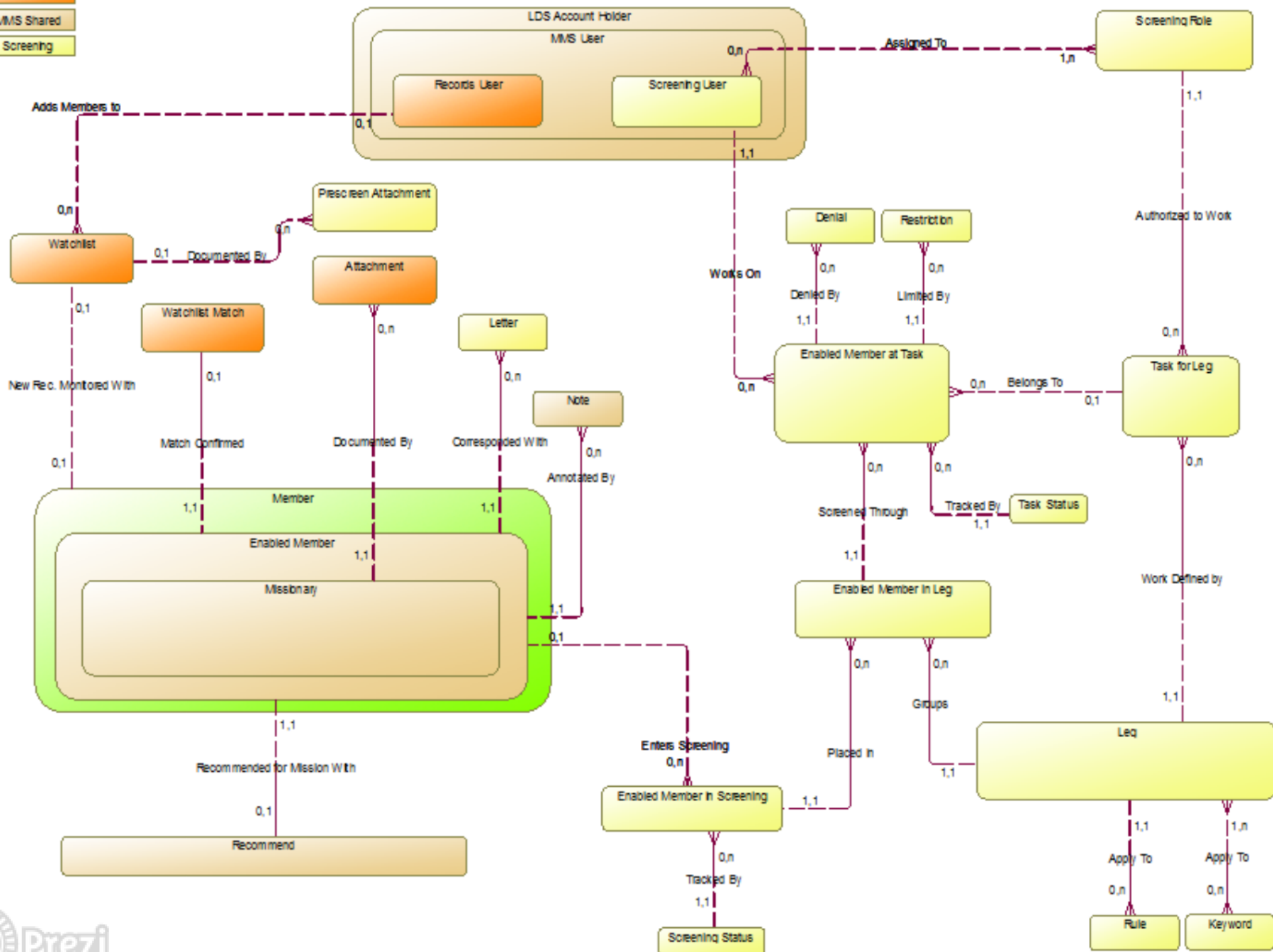
Conceptual

- aka Subject-Area or Business Models
- Only include the critical concepts within the scope of the model.
- The things mentioned over and over in conversations about the business.



- aka Subject-Area or Business Models
- Only include the critical concepts within the scope of the model.
 - The things mentioned over and over in conversations about the business.

- CMS
- Records
- MMS Shared
- Screening



Logical

- Also Relational Model
- The data rules behind an area of business
- Independent of DB implementation
- Uses normalization and abstraction



- aka Relational Model
- The data rules behind an area of business
- Independent of DB implementation
- Uses normalization and abstraction

Normalization

1nf - entities get PKs

- move repeating attributes to own entity
- separate multi-valued attributes

2nf - minimal PK to identify entity

3nf - no hidden dependencies

- remove derived data
- create lookup entities

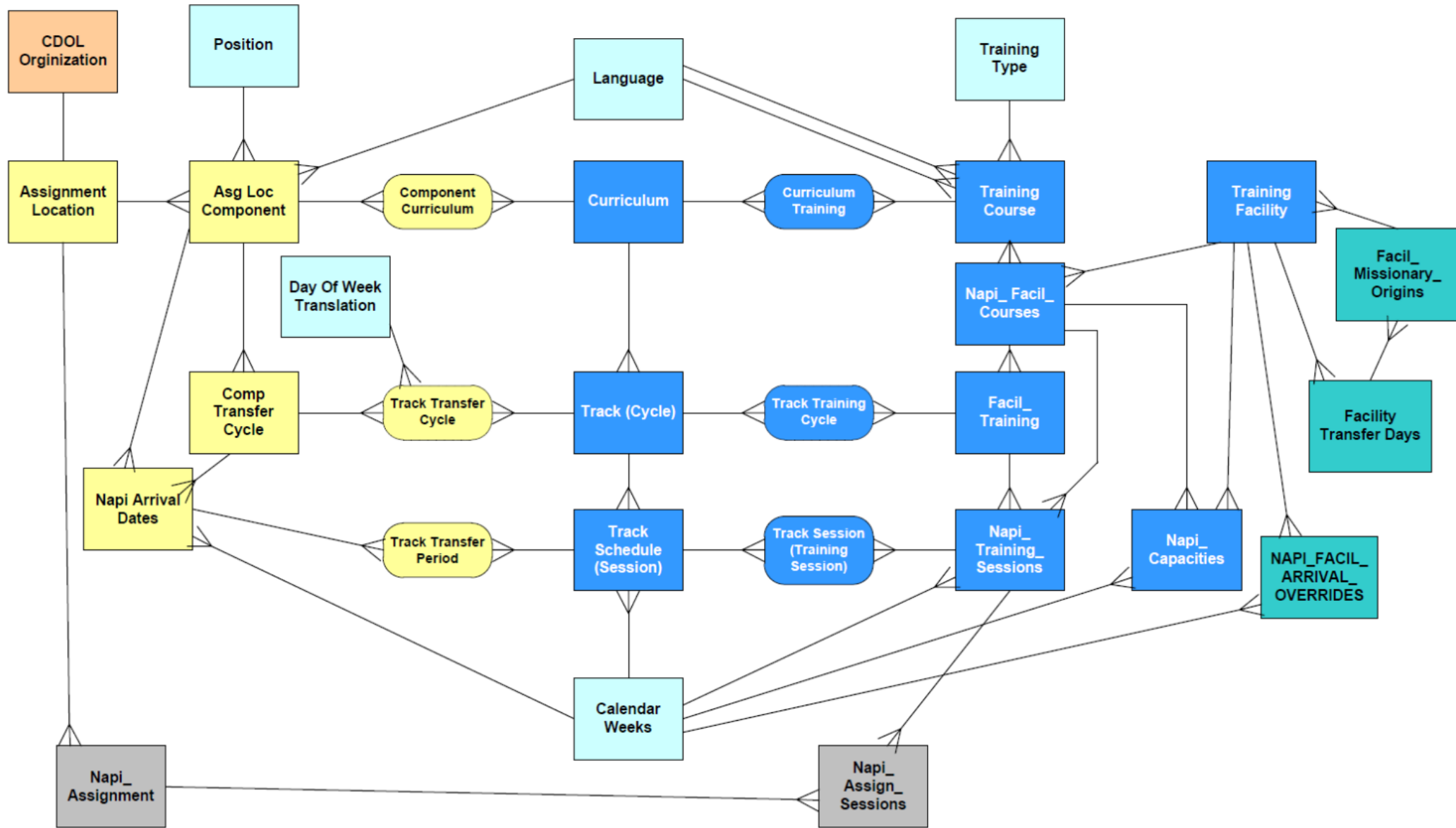
BCnf, 4nf, 5nf, 6nf - Never had the need

Abstraction

- Design in flexibility by eliminating specifics.
- In short, making the model more generic and re-usable.

Training Planning Entities & Relationships

23 Mar 2007



Physical

- Logical model optimized for the RDBMS implementation
- software
- hardware
- data usage
- performance
- Attributes and entities added to support tools, audit & history, reporting

< Live Demonstration of Physical Models >
(Images were removed to protect employer assets)

- Logical model optimized for the RDBMS implementation
 - software
 - hardware
 - data usage
 - performance
- Attributes and entities added to support tools, audit & history, reporting

RDBMS implementation

- software
- hardware
- data usage
- performance
- Attributes and entities added to support tools, audit & history, reporting

< Live Demonstration of Physical Models >
(Images were removed to protect employer assets)

**Duplication
is
Evil**

- multiple points of failure
- slower and error-prone
- inflexible
- unstable
- poor data quality
- costly to maintain

A Basket of Bad Ideas

EAV

entity-attribute-value



OAV

object-attribute-value



Vertical



Open



"Schema-Less"



- Praised as self-documenting and frees you from:
 - Joins & performance problems
 - Relational data constraints and issues
 - DBAs
 - Object-impedance mismatch
- However, total flexibility and no schema (model) means:
 - Duplication and redundancy everywhere
 - Missing, invalid data
 - Inconsistency
 - DBA duties are now yours
 - Objects and their data should still be analyzed, documented and modeled



Model Well

When defined users don't exist, the data model is the only thing that can help you understand the requirements better.

- The data model is the foundation of the system.
- Ensure the model is done properly the first time.
- Ignore those pleading for more flexibility: columns, queries, data access, derivation of joins.
- AFTER the model is correct, normalized, and reviewed, THEN make it flexible with views, materialized views, PL/SQL, or REST API, updatable views, virtual columns, etc.

Know Thy Data

- A data designer asks the right questions at the right time, determining and documenting:
- All the things
 - All the attributes of the things
 - All the relationships
 - Cardinality of the attributes and relationship
 - Data type of the attribute
 - Default value and valid values of the attribute
 - Minimum and maximum length of the attribute
 - Natural keys
 - Abstractness: "too specific?" \longleftrightarrow "too generic?"
 - Performance requirements
 - Security requirements
 - How will the data be used

The data model is the foundation of the system.

When defined users don't exist, the data model is the only thing that can help you understand the requirements better.

When defined users don't exist, the data model is the only thing that can help you understand the requirements better.

When defined users don't exist, the data model is the only thing that can help you understand the requirements better.

When defined users don't exist, the data model is the only thing that can help you understand the requirements better.

Victories and casualties...

- Convex models - US West
- EDV Issues - Structure Consulting
- Dev-friendly models - Statsoft Consulting
- Models of systems - Physical Fitness Class
- Dev-friendly Developer Model - Missoury
- Over-engineered and too flexible - Missouri
- Requirements Fuzzy - Missoury
- Cases: Health, Insurance, Medical - Statsoft
- MCT, Shuqun, LDIS Church - Statsoft

A data architect cares about clean data because it produces correct, actionable business intelligence instead of questionable business intelligence.



- The data model is the foundation of the system.
- Ensure the model is done properly the first time
- Ignore those pleading for more friendly columns, quicker data access, elimination of joins.
- **AFTER** the model is correct, normalized, and reviewed, **THEN make it friendly** with views, materialized views, PL/SQL or RESTful APIs, updateable views, virtual columns, etc.

Know Thy Data

A data designer asks the right questions at the right time, determining and documenting:

- All the things
- All the attributes of the things
- All the relationships
- Optionality of the attribute and relationship
- Cardinality of the attribute and relationship
- Data type of the attribute
- Default value and valid values of the attribute
- Minimum and maximum length of the attribute
- Natural keys
- Abstractions: Too specific? <-----> Too generic?
- Performance requirements
- Security requirements
- How will the data be used

"There are only two hard things in Computer Science: cache invalidation and naming things." - Phil Karlton (Principle Curmudgeon @ Netscape until his unfortunate passing in Italy in 1997)

Correctly naming the entities and attributes that will handle business change and the march of time should be a personal quest.



The name should accurately reflect what the thing contains.

Agile Data Modeling



- Be involved and engaged at the right time with stakeholders, business analysts and the dev team.
- Use a good modeling tool to capture, centrally locate and publish requirements and artifacts
- Data Model-Driven: No manual DDL. Generate DDL and new diagrams from the modeling tool. Validate, version and share.
- Use an automated DB migration framework

Modeling Tools



- Idera ER/Studio
- SAP PowerDesigner
- CA ERwin
- Oracle SQL Developer Data Modeler
- Several that are embedded in DB IDE tools

Victories and casualties...

- Convoluted models - US West
- EAV Horror - Structure Consulting
- Dev-Friendly models - Structure Consulting
- Models of screens - Physical Facilities Dept.
- Deadline-friendly Developer Model - Missionary
- Overengineered and too flexible - Missionary
- Requirements Fuzzy - Missionary
- **Clean, flexible, normalized models** - PacBell, NGT, Structure, LDS Church



bill.coulam@dbsherpa.com

dbsherpa.com (blog)

dbartisans.com (repo)

sourceforge.net/projects/plsqlframestart/

github.com/bcoulam/plsqlstarter

Save the Company by Modeling Data Right

