



Data Modeling for the Business

Donna Burbank, CA Technologies

A Handbook for Aligning the Business with IT using High-Level Data Models

DATA

MODELING FOR THE BUSINESS



STEVE HOBERMAN / DONNA BURBANK / CHRIS BRADLEY
CONTRIBUTOR - MONA POMRANING

Agenda

- Data Modeling for the Business
 - Starting with a High-Level Business View
 - Tips for Implementing your High-Level Model
 - Case Study

Who am I?

- Donna Burbank has more than more than 15 years of experience in the areas of data management, metadata management, and enterprise architecture.
 - Currently is the Senior Director of Product Marketing for CA's data modeling solutions.
 - Brand Strategy and Product Management roles at Computer Associates and Embarcadero Technologies
 - Senior consultant for PLATINUM technology's information management consulting division in both the U.S. and Europe.
 - Worked with dozens of Fortune 500 companies worldwide in the U.S., Europe, Asia, and Africa and speaks regularly at industry conferences.
 - Recently co-authored a new book entitled Data Modeling for the Business, with Steve Hoberman & Chris Bradley

Who Are You? Survey

- How would you describe your role?



Data Architect, Data Modeler, or Analyst



Businessperson or Business Analyst



DBA or Technical IT



A combination of the above



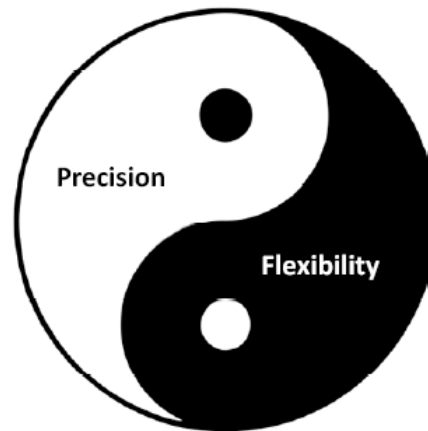
Other

The Challenge

- You've been tasked to assist in the creation of a Data Warehouse
- Trying to obtain a single view of **'customer'**
- Technical and political challenges exist
 - Numerous systems have been built already—different platforms and databases
 - Parties cannot agree on a single definition of what a 'customer' is
- Solution: Need to build a High-Level Data Model

What is a High-Level Data Model?

- A high-level data model (HDM) uses simple graphical images to describe core concepts and principles of an organization and what they mean
- The main audience of a HDM is **businesspeople**
- An HDM is used to facilitate **communication**
- It needs to be high-level enough to be intuitive, but still capture the rules and definitions needed to create database systems.



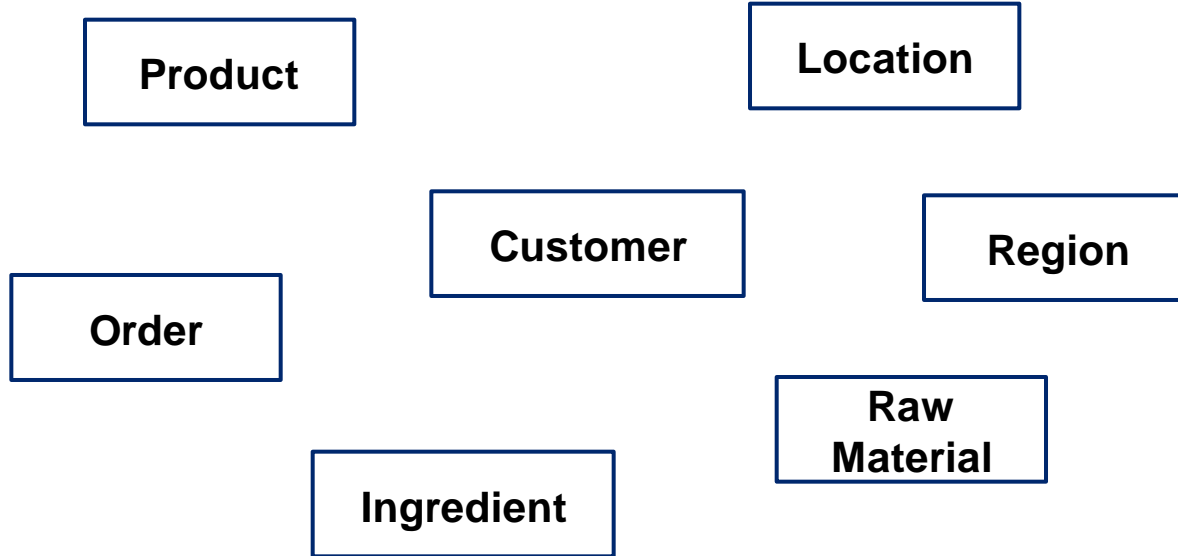
“A Picture is Worth a Thousand Words”

Examples of High-Level Data Models

Customer

“A Picture is Worth a Thousand Words”

Examples of High-Level Data Models

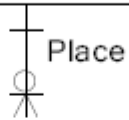


“A Picture is Worth a Thousand Words”

Examples of High-Level Data Models

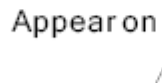
Customer

A person or organization who has purchased at least one of our products and has an active account.



Order

A contract to buy a quantity of product at a specific price.



Product

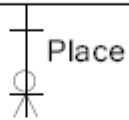
A physical good that is offered for sale. Does not include services.

“A Picture is Worth a Thousand Words”

Examples of High-Level Data Models

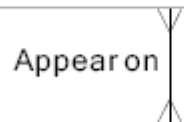
Customer

A person or organization who has purchased at least one of our products and has an active account.



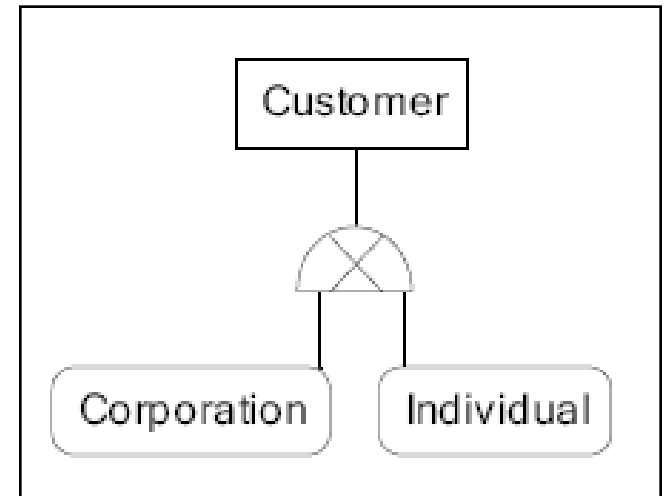
Order

A contract to buy a quantity of product at a specific price.



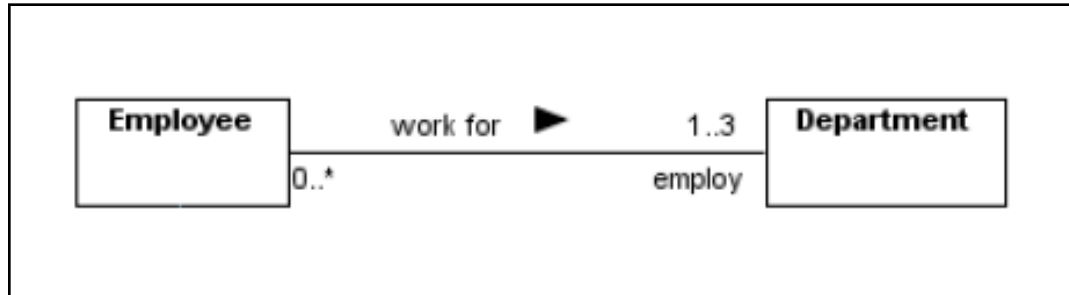
Product

A physical good that is offered for sale. Does not include services.



“A Picture is Worth a Thousand Words”

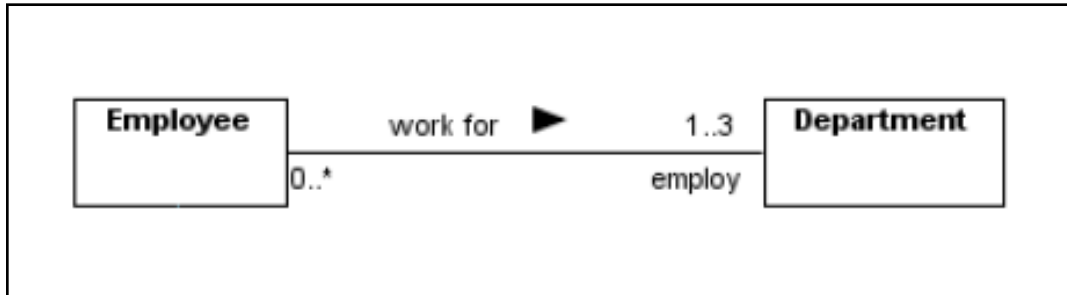
Examples of High-Level Data Models



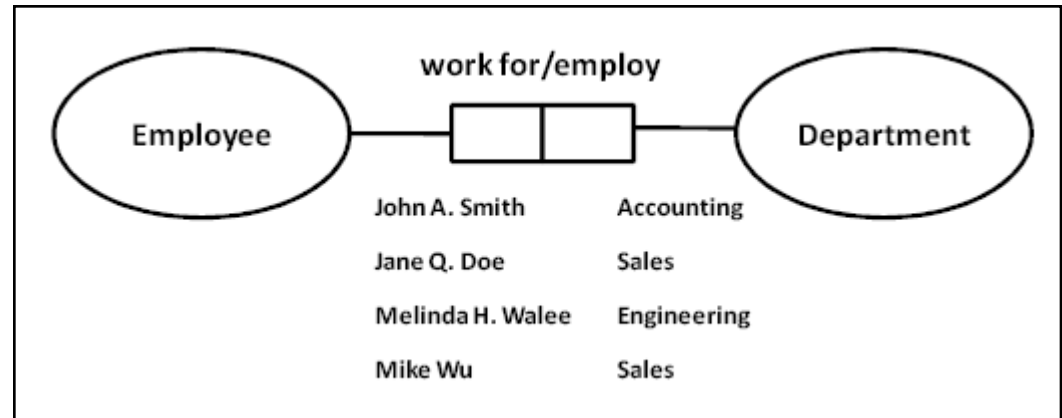
UML

“A Picture is Worth a Thousand Words”

Examples of High-Level Data Models



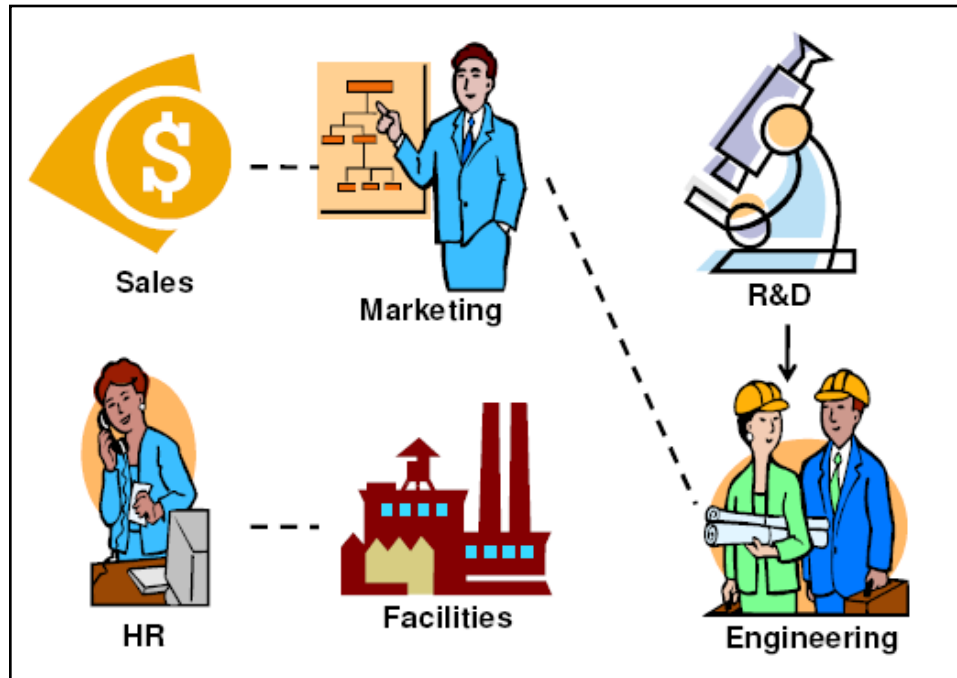
UML



ORM

“A Picture is Worth a Thousand Words”

Examples of High-Level Data Models



Name	Definition
Customer	The recipient and purchaser of our products.
Order	A contract to buy a quantity of product at a specified price.
Product	Anything we buy, sell, stock, move, or make. Any manufactured or purchased part, material, component, assembly, or product.
Associate	A person who is employed in a full or part time capacity by our company, to perform a service or function to our company.

“A Picture is Worth a Thousand Words”

Examples of High-Level Data Models

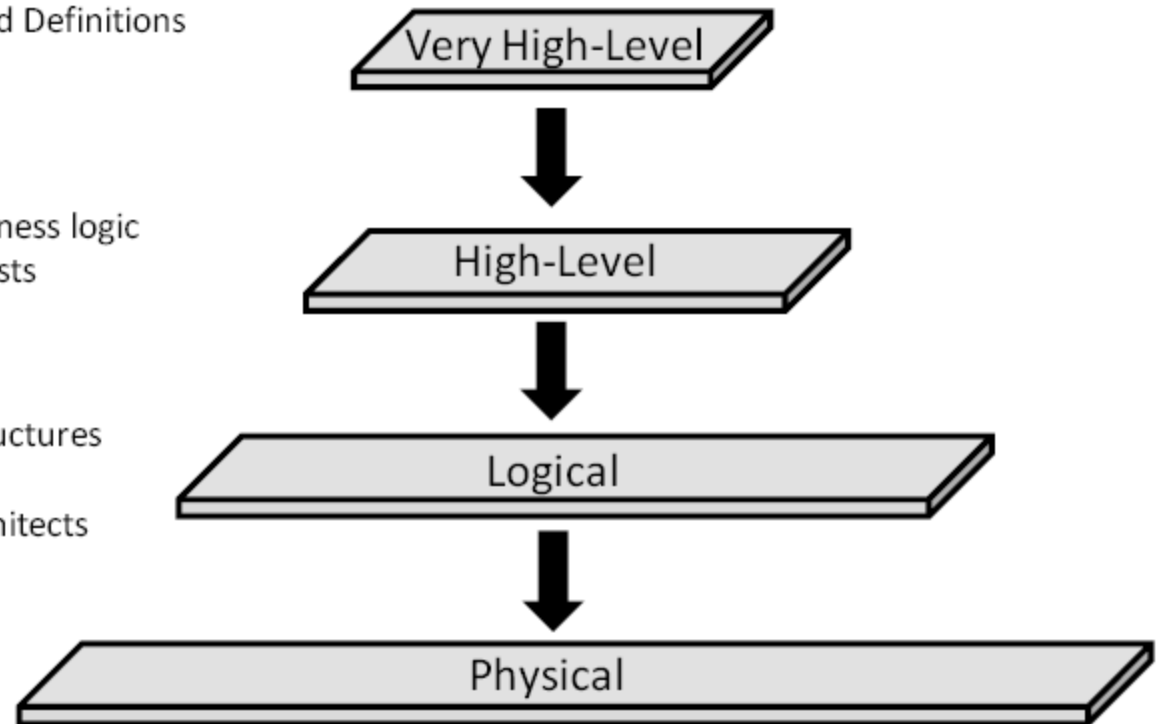


Is Notation Important?

- Many Notations can be used to express a high-level data model
- The choice of notation depends on purpose and audience
- For data-related initiatives, such as MDM and DW:
 - ER modeling using IE (Information Engineering) is our choice of notation (i.e. “crow’s feet”)
 - It is important that your high-level model uses a tool that can generate DDL, or can import/export with a tool that can
 - A repository-based solution helps with reuse and standards for enterprise-wide initiatives

Levels of Data Models

- One Pager
 - Agreement on 'Big Picture' Terms and Definitions
 - Audience = Business Users
- One Pager (maybe 2)
 - Defines core terms/definitions + business logic
 - Audience = Business Users and Analysts
- > 1 Page
 - Defines relational or dimensional structures independent of technology
 - Audience = Business Analysts & Architects
- > 1 Page
 - Defines relational or dimensional structures tuned for technology
 - Audience = Architects, DBAs, Developers



What is in a Name?

- The High-Level Data Model goes by many names

Very High-Level Data Model (VHDM)	High-Level Data Model (HDM)
Enterprise Data Model	Conceptual Data Model
Contextual Data Model	Subject Area Model
C-Level View	Business Data Model
'Picture'/ 'PowerPoint'	'One-Pager'

Many names for a High-Level Data Model

In a recent survey of data professionals, the most popular names for the high-level models included²:

Conceptual Data Model	59%
Subject Area Model	12%
Business Data Model	10%
Enterprise Data Model	6%
Other names:	13%

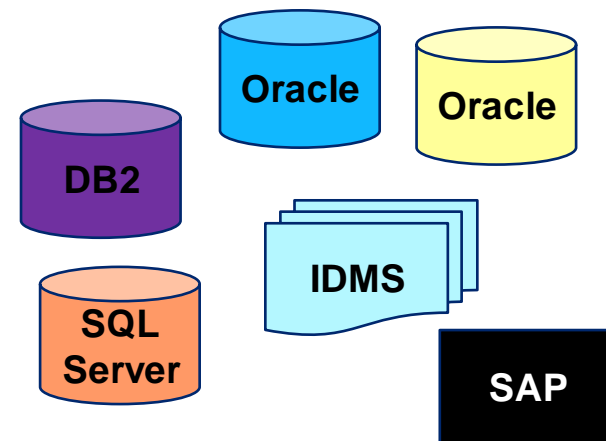
² Hoberman, Steve. "What's a Good Name for the High-Level Model?"
DMReview. 1 Dec. 2008.

How is this Different from a Logical Model?

VHDM	HDM	LDM
Defines the scope, audience, context for information	Defines key business concepts and their definitions	Represents core business rules and data relationships at a detailed level
Main purpose is for communication and agreement of scope and context	Main purpose is for communication and agreement of definitions and business logic	Provides enough detail for subsequent first cut physical design
Relationships optional. If shown, represent hierarchy.	Many-to-Many relationships OK	Many-to-Many relationships resolved
Cardinality not shown	Cardinality shown	Cardinality shown
No attributes shown	Attributes are optional. If shown, can be composite attributes to convey business meaning.	Attributes required and all attributes are atomic. Primary and foreign keys defined.
Not normalized (Relational models)	Not normalized (Relational models)	Fully normalized (Relational models)
Subject names should represent high-level data subjects or functional areas of the business	Concept names should use business terminology	Entity names may be more abstract
Subjects link to 1-M HDMs	Many concepts are supertypes, although subtypes may be shown for clarity	Supertypes all broken out to include subtypes
'One pager'	<i>Should</i> be a 'one pager'	May be larger than one page
Business-driven	Cross-functional & more senior people involved in HDM process with fewer IT.	Multiple smaller groups of specialists and IT folks involved in LDM process.
Informal notation	'Looser' notation required – some format construct needed, but ultimate goal is to be understood by a business user	Formal notation required
< 20 objects	< 100 objects	> 100 objects

Building a High-Level Data Model

- Let's go back to our challenge, to achieve a 'single version of the truth' for Customer information
- We have 6 different systems with customer information in them:
 - 2 on Oracle
 - 1 on DB2
 - 1 using legacy IDMS
 - 1 SAP system
 - 1 using MS SQL Server



Building a High-Level Data Model

- We start with a very simple HDM, with just one object on it, called “Customer”.
- We use an ER Model and show business definitions

Customer

A person or organization that has purchased at least one of our products and has an active account.

Too Simple??

Too simple?

- Our team thought so, so went ahead and focused on the technical integration, including:
 - Reverse engineering a physical model from each system
 - Creating ETL scripts
 - Migrating the data into a single warehouse
 - Building a reporting system off of the data

Focusing on the Business

- This implementation went “perfectly”, with no errors in the scripts, no data type inconsistencies, no delays in schedule, etc.
- We built a complex BI reporting system to show our upper management the results.
- We even sent out a welcome email to all of our customers, giving them a 50% off coupon, and thanking them for their support.

Focusing on the Business

- Until we showed the report to the business sponsor:
 - We can't have 2000 customers in this region! I know we only have around 400!
 - Why is Jones' Tire on this list? They are still evaluating our product! Sales was negotiating a 10% discount with them, and you just sent them a 50% coupon!?!?
 - You just spent all of that money in IT to build this report with bad data???



Back to the Drawing Board

- After doing an extensive review of the six source systems, and talking with the system owners we discovered that:
 - The DB2 system was actually used by *Sales* to track their prospective “customers”
 - These “customers” didn’t match our definition—they didn’t own a product of ours!!

Customer

A person or organization who does not currently own any of our products and who is potentially interested in purchasing one or more of our products.

Customer

A person or organization that has purchased at least one of our products and has an active account.

Oops!

- We were mixing current *customers*, with *prospects* (non-customers).
 - We just sent a discount coupon to 1600 of the wrong people!
 - We gave upper management a report showing the wrong figure for our total number of customers!
 - We are now significantly over budget to have to go back and fix this!!
- We started over, this time with a High-Level Data Model

Achieving Consensus

- We created a report of the various definitions of customer

Customer	A person or organization who does not currently own any of our products and who is potentially interested in purchasing one or more of our products.
Customer	A person or organization that has purchased at least one of our products and has an active account.

- > And verified with the various stakeholders that:
 - There were 2 (and only 2 definitions) of customer
 - Sales was OK with calling their “customer” a “prospect”

Resolving Differences

- Our new high-level data model looked like this:

Prospect

A person or organization who does not currently own any of our products and who is potentially interested in purchasing one or more of our products.



make a purchase to become

Customer

A person or organization who has purchased at least one of our products and has an account active.

Identify Model Stakeholders

- Make sure ALL relevant parties are involved in the design process
– *Get buy-in!*

Role	Builder – Business Focus	Builder – App Focus	User
Application Tester		✓	✓
Architect	✓	✓	✓
Business Analyst	✓		✓
Business User	✓		✓
DBA		✓	✓
Data Manager		✓	✓
Data Modeler	✓	✓	✓
Developer		✓	✓
Functional Analyst	✓	✓	✓
Project Manager		✓	✓
Project Sponsor	✓	✓	✓
Team Lead		✓	✓
Technical Analyst		✓	✓
Technical Support		✓	✓
Trainer			✓

Identify Model Stakeholders

- Make sure ALL relevant parties are involved in the design process – *Get buy-in!*

Role	Builder – Business Focus	Builder – App Focus	User
Application Tester		✓	✓
Architect	✓	✓	✓
Business Analyst	✓		✓
Business User	✓		✓
DBA			✓
Data Manager		✓	✓
Data Modeler	✓	✓	✓
Developer		✓	✓
Functional Analyst	✓	✓	✓
Project Manager		✓	✓
Project Sponsor	✓	✓	✓
Team Lead		✓	✓
Technical Analyst		✓	✓
Technical Support		✓	✓
Trainer			✓

Part of Your Job is Marketing!

A HDM Facilitates Communication

- A High-Level Data Model Facilitates Communication between Business and IT
 - Focus on your (business) audience
 - Intuitive display
 - Capture the business rules and definitions in your model
 - Simplicity does not mean lack of importance
 - A simple model can express important concepts
 - Ignoring the key business definitions can have negative affects
 - A model or tool is only *part* of the solution
 - Communication is key
 - Process and Best Practices are critical to achieve consensus and buy-in

Communication is the Main Goal of a High-Level Data Model

- Wouldn't it be helpful if we did this in daily life, too?
- i.e. "Let's go on a family holiday!"

Person	Concept	Definition
Father	Holiday	An opportunity to take the time to achieve new goals
Mother	Holiday	Time to relax and read a book
Jane	Holiday	A chance to get outside and exercise
Bobby	Holiday	Time to be with friends
Donna	Vacation	More time to build data models

Some Creative Ways to Facilitate Conversations with Stakeholders

- Food!
 - “Lunch and Learn”
 - Bring candy to meetings
- Force?
 - “No bathroom breaks until we reach consensus!”
- Active Listening
 - Understand **why** there is disagreement (e.g. “Ingredient” vs. Raw Material)
- Fit into **their** schedule
 - Webinars
 - The “5 minute rule” for business execs – small, bite-sized models or questions.

Example from ERworld Case Study: Scott Northrup's Implementation at Wells Fargo



!!! WIN Starbucks Coffee or Jamba Juice !!!



<== CLICK ... for more info ... CLICK ==>

Enticement!

Teradata MDS Logon

Click on a User Profile ==>
(Unless you are a designated
Administrator, select Power User)

Enter a password ==>
[Power User password = **metadata**],
then click the Logon button to access
Meta Data Services

Select a User Profile:

- Power User
- Administrator

User Name: CIA

Password: ●●●●●●●●

Logon

Last Updated: 2/2/06

Identify Model Purpose

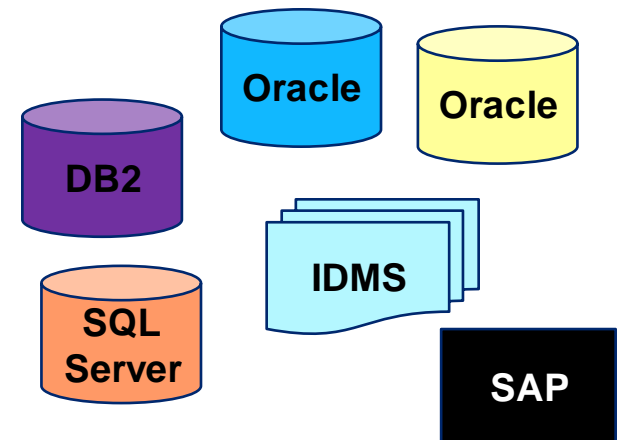
- Key to success of any project is *finding the right pain-point and solving it.*
- Make sure your model focuses on a particular pain point, i.e. migrating an application or understanding an area of the business

	Existing	Proposed
Business	“Today an Account can only be owned by one Customer.”	“By next quarter, an Account can be owned by more than one Customer.”
Application	“In the legacy Account Management system, we call the customer an Account Holder.”	“When we migrate to SAP/R3, Account Holder will be represented as Object.”

Managing the Technical Infrastructure

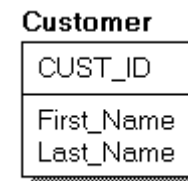
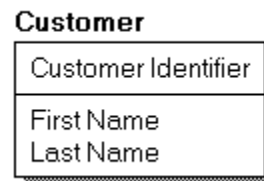
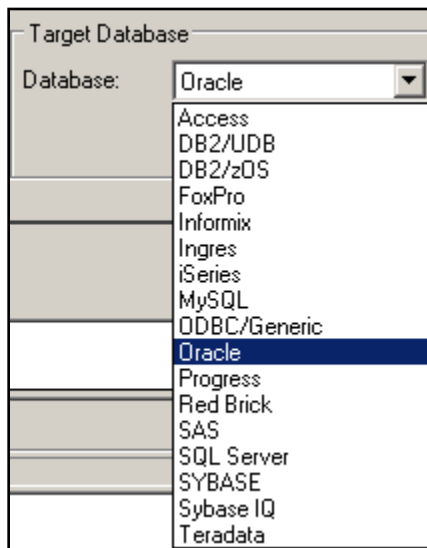
Why do you need a modeling tool, and not a drawing tool?

- Recall that we had multiple data sources on a variety of platforms:
 - 2 on Oracle
 - 1 on DB2
 - 1 using legacy IDMS
 - 1 SAP system
 - 1 using MS SQL Server

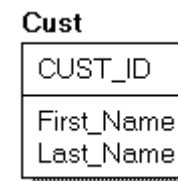


Creating a Data Inventory

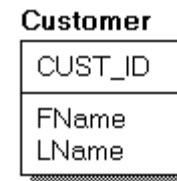
- “Design Once, Reuse Many Times” across heterogeneous platforms
- **Design layers** allow you to have a single high-level/logical model pointing to numerous physical model platforms.



Oracle

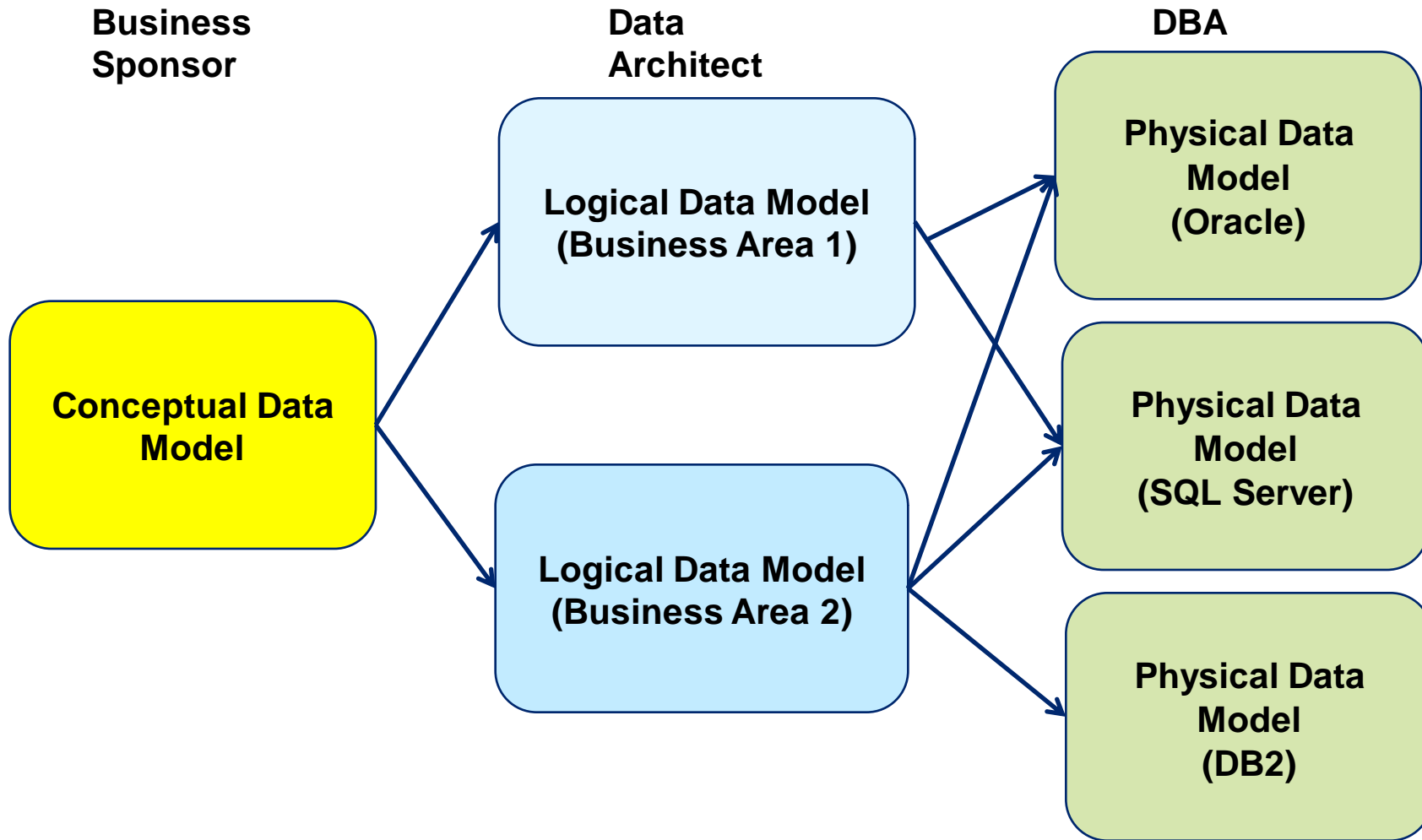


SQL Server



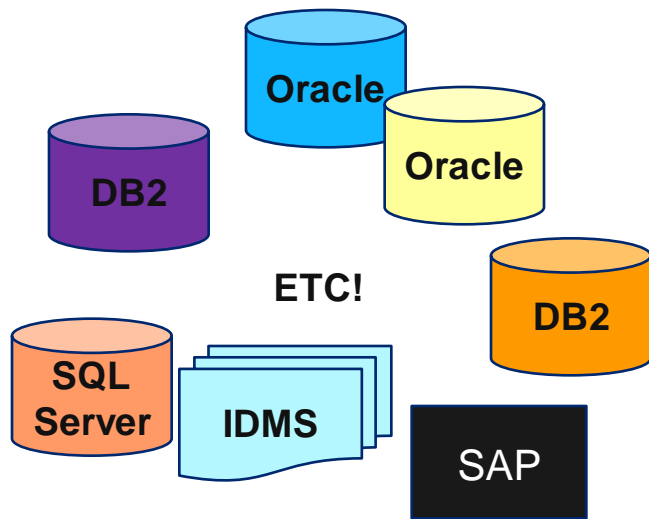
DB2

Design Layers Create both Business and Technical Designs



A Data Model can be your Filter

- A Data Model can add:
 - **Focus** – by Subject Area, by Platform, etc.
 - **Visualization** – Different Views for Different Audiences
 - **Translation** – to different DMBS AND to non DBMS formats such as UML, BI tools, Excel, XML, etc, etc.



Developers



Business Sponsors



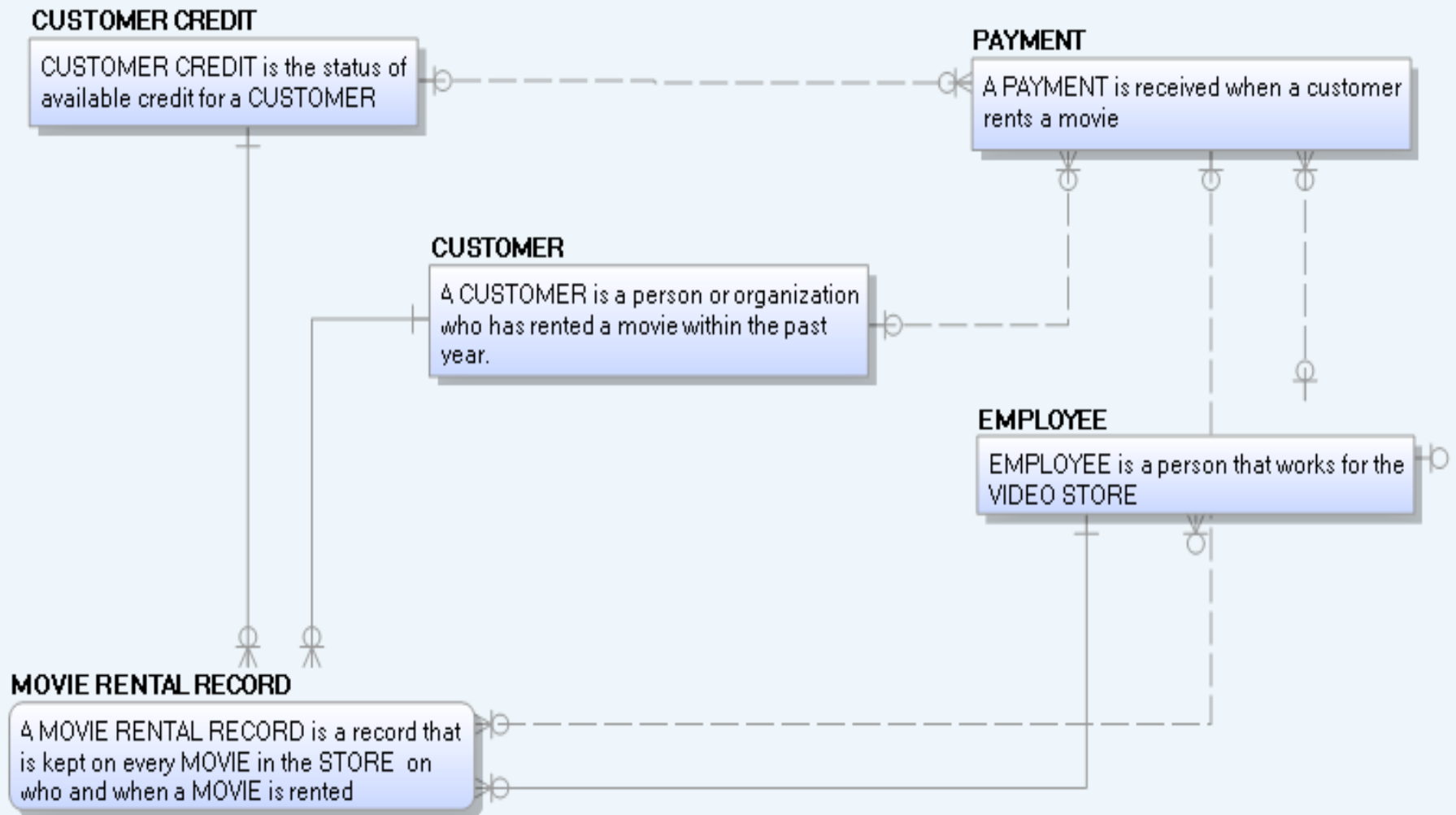
Data Architects

ETC!

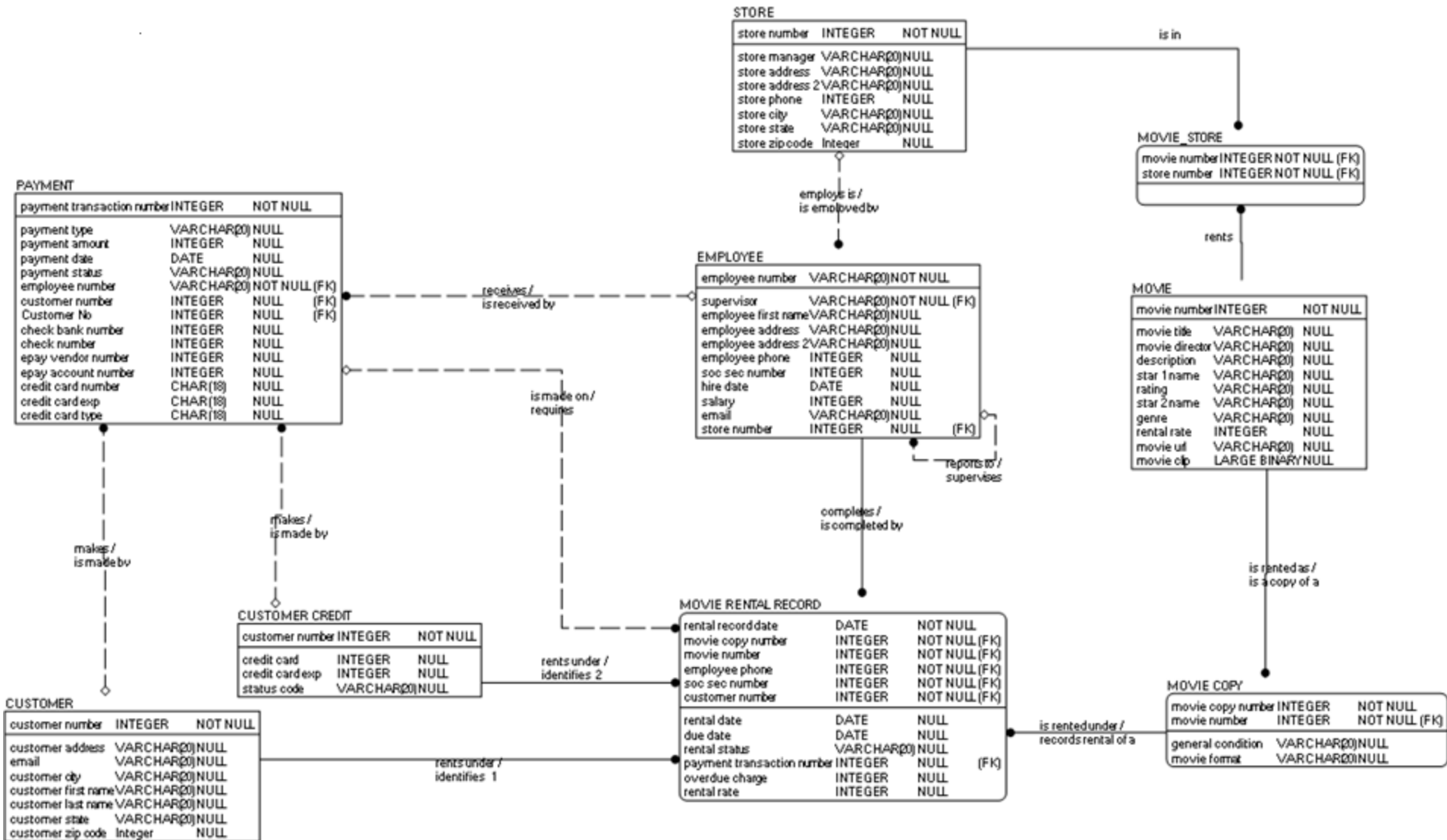


DBAs

Create Different Displays for Different Audiences: BUSINESS



Create Different Displays for Different Audiences: TECHNICAL



Generate Intuitive Reports for End Users

Many users want to see definitions, but not read a data model.

CRYSTAL REPORTS® 2008

Logical Dictionary

4/29/2010 1:46:15PM

Model : EMOVIES 7.3.3.1772

Subject Area : Customer

Entity Name : CUSTOMER

Attribute Name	Attribute Definition
customer number	A unique number associated with Customer
customer address	Street address of the customer
email	The email address associated with a customer
customer city	City associated with Customer's address
customer first name	First name of a Customer
customer last name	Last name of a Customer
customer state	The two-character state abbreviation associated with customer's address
customer zip code	zip-code associated with Customer's address

Entity Name : CUSTOMER CREDIT

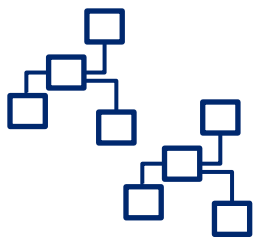
Entity Definition : CUSTOMER CREDIT is the status of available credit for a CUSTOMER

Attribute Name	Attribute Definition
customer number	A unique number associated with Customer
credit card	A credit card number associated with Customer
credit card exp	Expiration date on the customer's credit card
status code	Payment status associated with a Customer

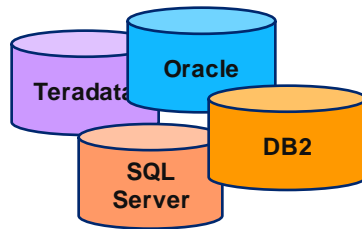
Managing the Data Inventory with a Central Repository

- A Central Model Store provides a single repository to store all of your data model assets
- A collaborative environment for multiple modeling teams.
- Metadata storage for: multiple models, multiple dbms platforms, multiple tools, multiple audiences

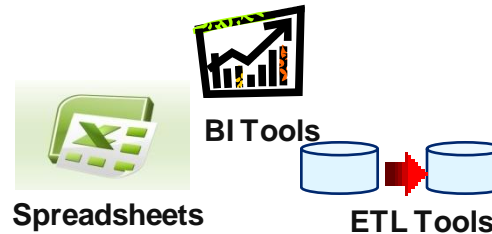
Multiple Models



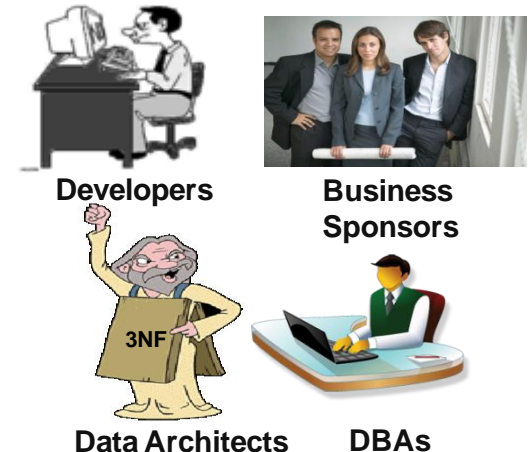
Multiple DBMSs



Multiple Tools



Multiple Audiences



Single Definition of
"Customer"

Central Model Store

Case Study – Major International Oil Company

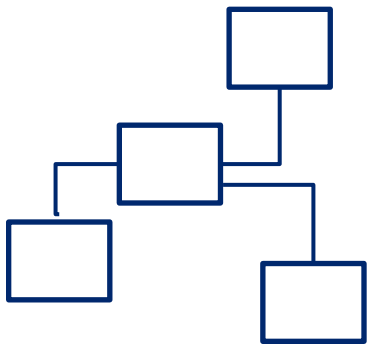


Corporate Culture

- A **diverse, federated organization** – culture encourages local decision making within a corporate framework
- Before high-level models were introduced:
 - Data architecture performed in different Segments & Functions
 - Variety of tools & techniques used
 - Projects encountered common cross-business data concepts, but largely created their own models & definitions
 - No overall context for models existed
 - Negative image regarding the term “models”

Repurpose “Models”

- In addition to using traditional “data models”, the team translated their high-level data models to
 - Excel Spreadsheets
 - Word Documents & Reports
 - HTML Pages on the web
- It was the *same information*, but translated into a *format the business users could understand*.



Repurpose Models – MS Excel

CM_PIdata.xls [Compatibility Mode] - Microsoft Excel

Home Insert Page Layout Formulas Data Review View

Clipboard Font Alignment Number Styles Cells Editing

A2 Asset

	A	B	C	D	E	F	G	H	I	J	K	L
	Entity Name	Attribute Name	Logical Rolename	Definition	Attribute Notes	Mandatory / Optional	Reference Value Name	Primary Key (PK) Foreign Key (FK)	Domain	Data Type	Width	Scale
1		AssetID	AssetID	Unique identifier for a defined BP asset. Set values same has in AHOU table in Production. Only set Asset field values for PI Tags (may need to add Platform code later).		Mandatory		(FK)		INTEGER		
2	Asset	BUID	BUID	Unique identifier for a Business Unit.		Mandatory		(FK)		VARCHAR	50	
3	Asset	AssetName	AssetName	Name of a defined BP asset.		Optional				VARCHAR	50	
4	Asset	AssetType	AssetType	Type of asset (one of Field, Platform, Well).		Optional				VARCHAR	50	
5	Asset	Parent	Parent	Parent Asset for specified asset entry.		Optional				VARCHAR	50	
6	Asset	Effective_Date	Effective_Date	Date defined asset came into existence.		Optional				DATE		
7	Asset	Expiry_Date	Expiry_Date	Date defined asset terminated existence.		Optional				DATE		
8	BusUnit	BUID	BUID	Unique identifier for a Business Unit.		Mandatory		(FK)		VARCHAR	50	
9	BusUnit	PUID	PUID	Unique identifier for the parent Performance Unit.		Mandatory		(FK)		VARCHAR	50	
10	BusUnit	Abbreviation	Abbreviation	An abbreviated term for the business unit.		Optional				VARCHAR	50	
11	BusUnit	Long_Name	Long_Name	Full (legal) description of the Business Unit within BP.		Optional				VARCHAR	50	
12	BusUnit	Effective_Dt	Effective_Dt	Date the business unit became operational with assigned responsibilities.		Optional				DATE		
13	BusUnit	Expiry_Dt	Expiry_Dt	Date the business unit ceased operational responsibilities, was decommissioned or reorganized into a new or existing business unit.		Optional				DATE		
14	Equip System	EquipSysID	EquipSysID	Unique identifier for a specified equipment system.		Mandatory		(FK)		INTEGER		
15	Equip System	SystemTypeID	SystemTypeID	Unique identifier for a group of equipment systems that support a major function for an asset.		Mandatory		(FK)		INTEGER		
16	Equip System	AssetID	AssetID	Unique identifier for a defined BP asset. Set values same has in AHOU table in Production. Only set Asset field values for PI Tags (may need to add Platform code later).		Mandatory		(FK)		INTEGER		

Submodels Entities Attributes Relationships Domains Reference Values

70%

Community of Interest

- The data architecture team created a “Community of Interest” to:
 - Share best practices inside company
 - Exchange ideas across projects
- The goal was to get ALL users involved via
 - “Lunch and Learn” meetings
 - Webinars
 - Training and Education
- **Both Business and Technical resources were invited**



Case study lessons

- Understand roles and motivations and work within the organization
 - Federated governance model
 - Avoid silo mentality
 - Communicate
 - Start small & document success
 - Make it easy to get hold of
 - Market, market, market!
- Follow up with a robust architecture
 - Common repository
 - Models appropriate for the audience
 - Defined ownership/stewardship
 - Unique definitions
 - “Repurpose” data for various audiences: via the web, Excel, DDL, XML, etc. It’s the **data** that’s important, not the format.



Case study lessons

- Understand roles and motivations and work within the organization
 - Federated governance model
 - Avoid silo mentality
 - Communicate
 - Obtain buy in by starting small & document success
 - Make it easy to get hold of
 - Market, market
- Follow up with a robust architecture
 - Common repository
 - Defined stewardship
 - Unique definitions
 - “Repurpose” data for various audiences: via the web, Excel, DDL, XML, etc. It’s the **data** that’s important, not the format.



Part of Your Job is Marketing!

Summary

- A high-level data model can help achieve a “single view of customer”
- Aim the HDM at the business user by being “generic” but keeping enough detail to make it meaningful
- Treat your model like a project
 - Identify pain points and solve them
 - Identify stakeholders and market
 - Document purpose and expected results
 - Follow an organized, repeatable process
- Using high-level models can help increase communication with the business and achieve better results

Data Modeling for the Business

- Available at:
 - Amazon.com
 - Technics Publications

The authors of Data Modeling for the Business do a masterful job at simply and clearly describing the art of using data models to communicate with business representatives and meet business needs. The book provides many valuable tools, analogies, and step-by-step methods for effective data modeling and is an important contribution in bridging the much needed connection between data modeling and realizing business requirements.

Len Silverston, author of The Data Model Resource Book series

