Whitemarsh
Information Systems Corporation

DC Area Business Objects Crystal User Group (DCABOCUG)

Data Warehouse Architectures for Business Intelligence Reporting

April 14, 2009

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: Whitemarsh@wiscorp.com
Web: www.wiscorp.com

Topics

1. What is Business Intelligence ? 1

2. Fundamental Data Management Terms 2

3. Data is Executed Policy (core principles) 5

4. The World of Data Architectures 7

5. Two Data Warehouse Strategies 15
   5.1 ER-Model Based (Inmon) 16
   5.2 Star Schema Based (Kimball) 17
   5.3 Key “Data Warehouse Architecture” Differences 19
   5.4 Key Data Modeling Differences 22
   5.5 Key Build Strategy Differences 25

6. Conclusions and Recommendations 29

1. What is Business Intelligence ?

2. Fundamental Data Management Terms

Term	Definition
Data	The persistent evidences of policy executions, that is, the execution of the procedures of the enterprise.
Database	A highly organized repository of enterprise-data that contains its own internal structure ensuring that data can be entered, updated, reported, and held secure.
Database Management	The strategies and procedures for managing enterprise data within databases in an orderly and well-reasoned manner.
Database Management System (DBMS)	An IT software system that accomplishes the definition, instantiation, evolution, maintenance, evolution, and reporting of databases. Example: Oracle, DB2.
Data Element	A Data Element is a context independent business fact semantic template that can be employed as the meaning and rules basis for an attribute, column, field, screen element, etc.

Term	Definition
Data Model	A model of the data that is to be contained in a database. A complete data model consists of: 1) data record structure (a.k.a., tables), 2) relationships among the tables, and 3) formally defined operations on the rows of data and the relationships.
Concepts Data Model	A type of data model that represents the data models (see above) of individual concepts. E.g., Person, Address, Invoice, and Customer. Commonly this type of data model is high-level and likely only contains entities, attributes, and relationships. Possible 100s of data models of concepts. Data structure pattern of a concept. “Ether-land”
Logical Data Model	A database data model that consists of a schema, tables, columns, and relationships. These are typically constructed through the employment of data models of concepts within the database’s data model. These models are DBMS independent. Commonly these models are in third normal form.
Physical Data Model	A database models tuned to the needs of a specific application and DBMS. Commonly these models are derived from one or more logical data models and may not be in third normal form.

Term	Definition
Metadata	An imprecise and vague term to mean the specifications of real objects. Example: if a database's data is "real," the database's schema is metadata. Simply put, metadata is the abstraction-level about the considered object. Book example: while the chapters are "real," the table of contents and index are metadata.
Metadata Management System	A software application, commonly with a database that is managed by a DBMS that stores, manages, and reports metadata.
Business Intelligence	The sets of data that a business declares is needed to understand its past, and undertake tactical and strategic operations about its current, and future activities.

3. Data is Executed Policy (core principles)

- The persistent evidences of policy executions, that is, the execution of the procedures of the enterprise.
- Processes are the procedures through which policy is executed and data is created.
- Policy and procedures run the business.
- Policy executions produce data.
- Data stored in databases are the persistent memory of the organization.

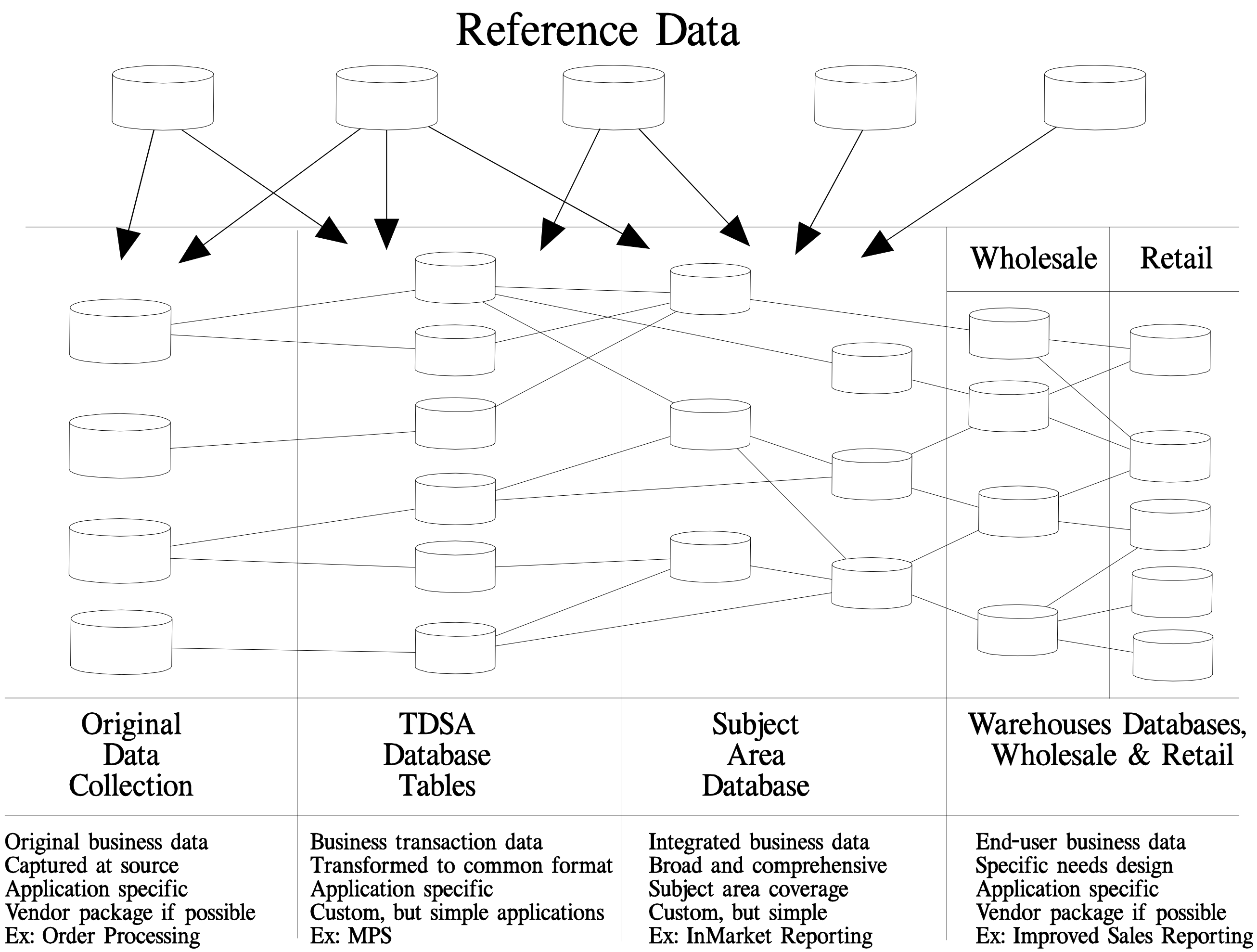
Data is Executed Policy (cont.)

- "Data" specifications are **policy definitions**.
- “Process” specifications are **procedure definitions**.
- All data (i.e., policy) specifications are metadata.
- All process (i.e., procedure) specifications are metadata.
- A metadata database, i.e. a metabase, is a database for all Policy and Procedure Specifications.
- Data Management manages both metadata and data.

4. The World of Data Architectures

An Architecture: An engineered and recognized style that is replicated across many instances in support of a well-defined purpose.

Five Data Architecture Classes Seen in Enterprises



Data Architecture Persistent Data Classifications and Characteristics				
Persistent Data Classification	**Persistent Data Characteristics**	**Process Characteristics**	**User Considerations**	**Technical Considerations**
Original Data Capture	Detailed atomic data Accurate as of the last update Well defined, long lasting database designs Normalized database designs Uses reference data No invalid data updates allowed	Tuned for transaction capture, storage and update Application oriented Transaction driven Processing supported by well known data integrity and business processing rules Understands, creates, and maintains TDSA databases through Original Data Capture system extract and TDSA loading software. Source data for TDSA	Original data source entry personnel High availability Supports day-to-day operations	Amount of data for processing is small Multiple vendor packages Package specific May or may not be controlled by SQL DBMS

Data Architecture
Persistent Data Classifications and Characteristics

Persistent Data Classification	Persistent Data Characteristics	Process Characteristics	User Considerations	Technical Considerations
TDSA: Transaction Data Staging Area	Transient data/short lived Foundation data source for all operational data systems Enterprise-wide standard semantics Package independent database designs Denormalized Full business transaction Does not use reference data	Accepts, stores, and then pushes forward function Only refreshed with changes from previous version Operations application data system daily update event driven Translation and transformation	Users cannot access	Multiple platforms Interface monitoring Applications insulation SQL DBMS controlled

Data Architecture Persistent Data Classifications and Characteristics				
Persistent Data Classification	**Persistent Data Characteristics**	**Process Characteristics**	**User Considerations**	**Technical Considerations**
ODS: Operational Data Store "Subject Area Databases"	Detail level data May be lightly summarized Current or nearly current Rolling histories Broad subject area database scope Normalized database designs Redundant data from across enterprise May contain derived data from "outside" Uses reference data May receive and/or send data to databases within class Data source for all warehouse databases	Updated daily via TDSA data transaction files Accepts and stores data from TDSA Supports comprehensive reporting and generalized ad hoc query	End-user detailed level analysis Used for up to the minute decisions Used for detailed decision making	Requires fast response time Large volume SQL DBMS controlled

Data Architecture Persistent Data Classifications and Characteristics				
Persistent Data Classification	**Persistent Data Characteristics**	**Process Characteristics**	**User Considerations**	**Technical Considerations**
Warehouse: Wholesale **Inmon data warehouse**	Detail and some summarized. Rolling Histories Load/replace, no end-user update Enterprise-wide standard semantics Address one or more subject areas Redundant data from across enterprise May contain internal derived data Reference data fully embedded May receive and/or send data to databases within class Data source for all retail data warehouses	No end-user updating Regular, periodic updates Supports standardized, on-demand reports Supports general complex business data analyses such as trends and forecasting Views data from multiple subject areas Supports data mining	Supports power and analytical users community Used for broad direction and positioning Used to formulate and assess long term decisions	Availability not on business' critical path User workstation access Large data volumes per query High processing power required SQL DBMS controlled

Data Architecture Persistent Data Classifications and Characteristics				
Persistent Data Classification	**Persistent Data Characteristics**	**Process Characteristics**	**User Considerations**	**Technical Considerations**
Warehouse: Retail **Kimball Data Mart**	Light to highly summarized and some detail Rolling histories Load/replace, no end-user update Enterprise-wide standard semantics Denormalized and highly designed to specifically favor one or more reporting formats Redundant data from across enterprise May contain internal derived data Reference data fully embedded May receive and/or send data to databases within class	Availability not on business' critical path Regular, periodic updates Highly designed, end-user on-demand reports Supports self-service end-user reporting Supports very specific simple to complex business data analyses Views data from multiple subject areas	Supports managerial community Cannot update Used for direction and positioning Supportive of long term decision making Specific reporting need	Relaxed availability User workstations Large volume High processing power SQL DBMS controlled

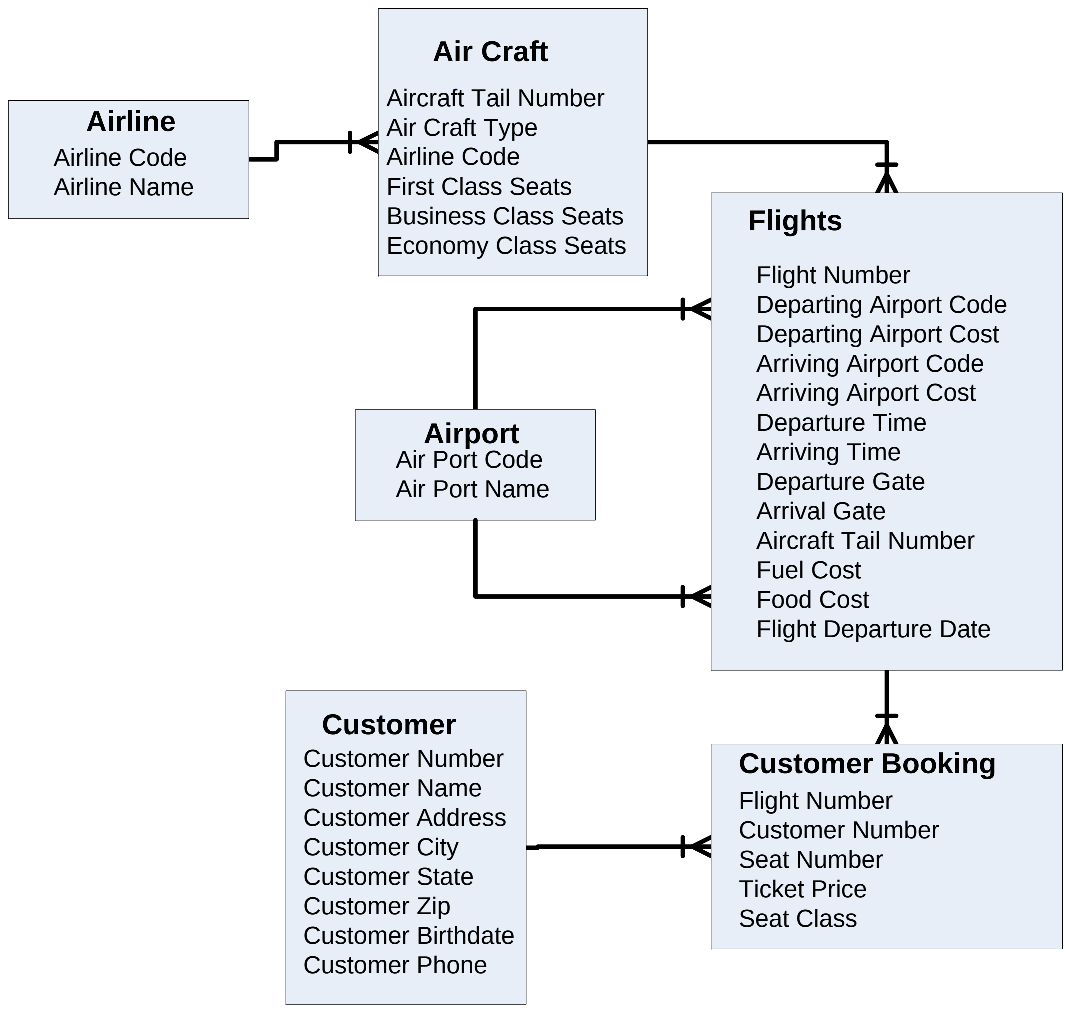
Data Architecture Persistent Data Classifications and Characteristics				
Persistent Data Classification	**Persistent Data Characteristics**	**Process Characteristics**	**User Considerations**	**Technical Considerations**
Reference Data	Durable codes and long value alternatives with policy definitions and full descriptions Enterprise-wide standard semantics Source of all valid and invalid values including alternatives for different countries and languages Multiple group data field constructors suitable for different countries and languages Definitive source for multi-use data in all other databases Changed data history supported by conversion mappings Long lasting, seldom updated	Simple updates Update mappings required for reference data value migration	Needed by all levels in the organization Used by all systems Enables understanding and conversion of historical data	Supports the concept of single source Integration with all data store types

5. Two Data Warehouse Strategies

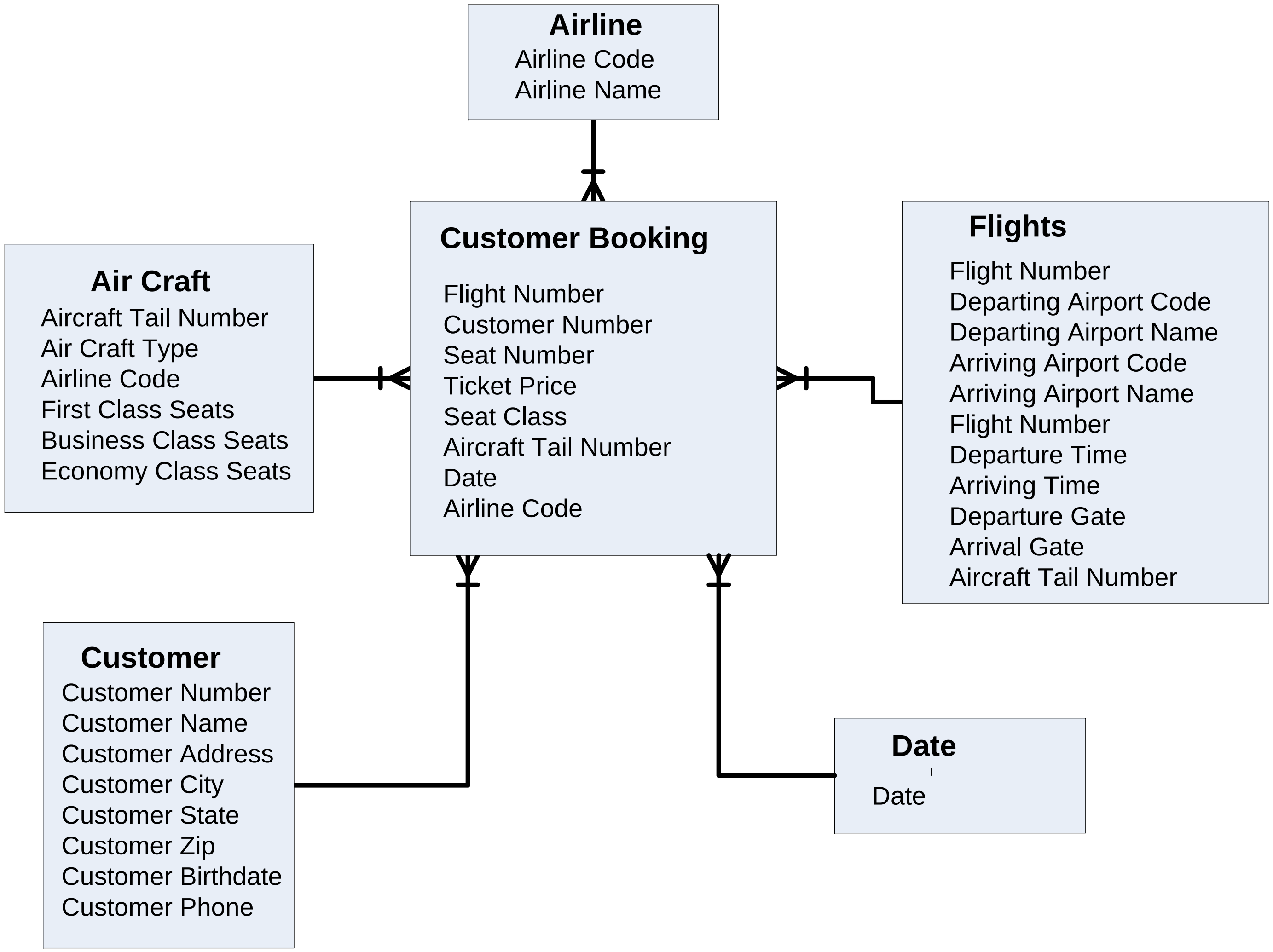
Bill Inmon – ER Model -based

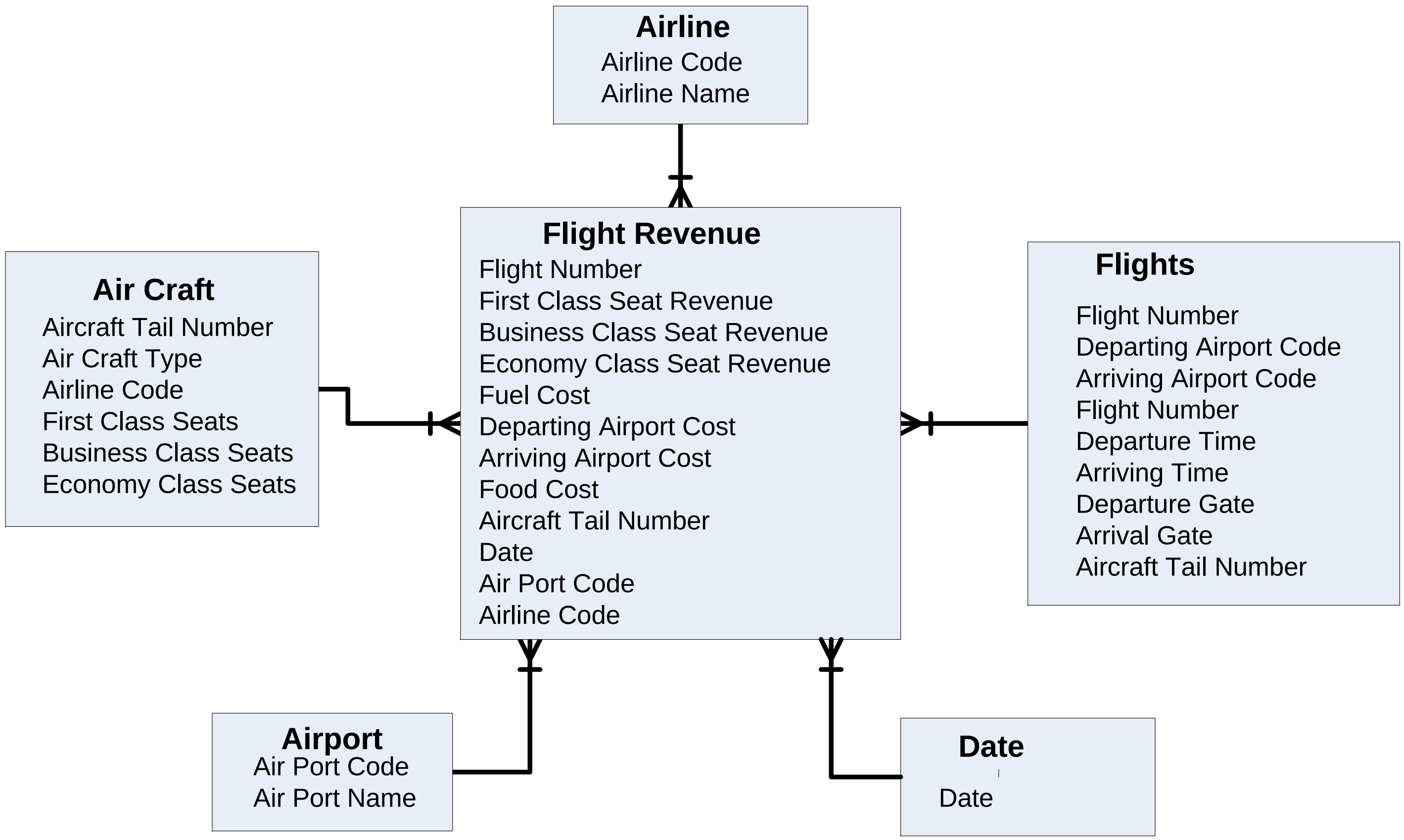
Ralph Kimball – Star Schema-based

5.1 ER-Model Based (Inmon)



5.2 Star Schema Based (Kimball)




Airline
Airline Code
Airline Name
Air Craft
Aircraft Tail Number
Air Craft Type
Airline Code
First Class Seats
Business Class Seats
Economy Class Seats
Flight Revenue
Flight Number
First Class Seat Revenue
Business Class Seat Revenue
Economy Class Seat Revenue
Fuel Cost
Departing Airport Cost
Arriving Airport Cost
Food Cost
Aircraft Tail Number
Date
Air Port Code
Airline Code
Flights
Flight Number
Departing Airport Code
Arriving Airport Code
Flight Number
Departure Time
Arriving Time
Departure Gate
Arrival Gate
Aircraft Tail Number
Airport
Air Port Code
Air Port Name
Date
Date

5.3 Key "Data Warehouse Architecture" Differences

1. Kimball approach has many Star Schemas.... Sort of a "have a need? Make a Start-Schema Data Mart!" Multiple Star Schemas share "dimensions."

2. Inmon is a top-down approach. Data Marts "pop" out of the bottom of a cohesive Enterprise-wide (or at least subject-wide) database architecture.

3. Kimball would say: The Corporate Data Warehouse is the collection of all Data Marts in which each corresponds to the past and current information about a discrete decision domain.

4. Inmon would say: The Corporation Data Warehouse is a unified, integrated, and non-redundant enterprise-level operational, tactical, and strategic database. A Data Mart is a materialized discrete data-based view of a small decision domain.

Quotes from Kimball and Inmon

"… The data warehouse is nothing more than the union of all the data marts …" **Ralph Kimball** Dec. 29, 1997.

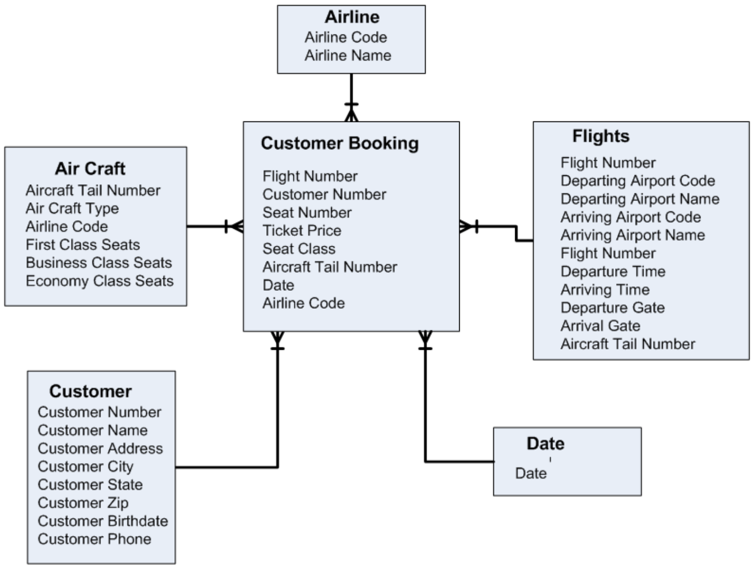
"You can catch all the minnows in the ocean and stack them together and they still do not make a whale." **Bill Inmon** Jan. 8, 1998.

In short –

Kimball--First Data Marts–The “Sum” of which ***is*** the Enterprise Data Warehouse.

Inmon---First the Enterprise Data Warehouse--Later the Data Marts.

5.4 Key Data Modeling Differences



1. Each dimension is a "flat" table of collapsed hierarchies. Hence very un-normalized.

2. Each dimension with history has a "Current" flag for the most recent row.

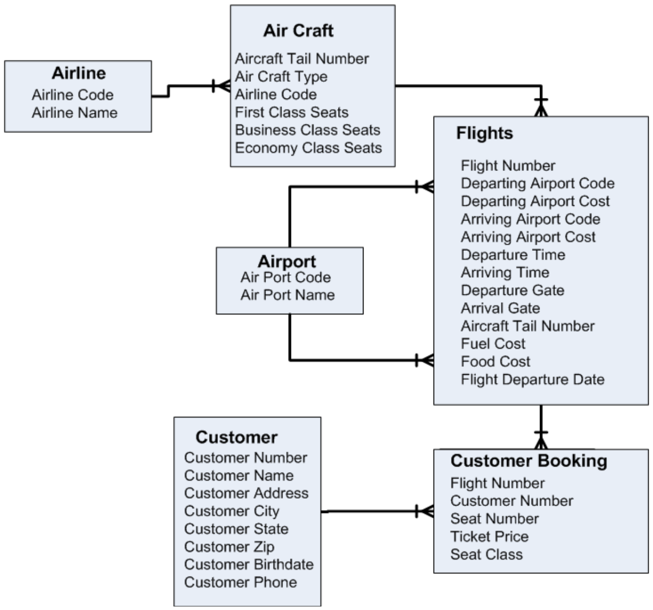
3. Time-sequence is managed by a special Time-Dimension

4. Fact tables record state changes or fact measurements. Facts are selectable or distinguishable by related dimension information.

5. Granularity, precision, and timeliness is set to the need of the decision domain of the specific star schema.

6. Dimensions can be shared between and among fact tables thus giving the ability to "sort of" report different measurements based on common "dimensions."

ER-Schema Data Models



1. Every table is normalized to the maximum degree possible.

2. No special treatment of history other that what is “naturally” engineered into database tables.

3. Time- sequence is represent by time-stamped columns

3. No “manufactured” dimensions. No “manufactured” fact tables.

4. Granularity, precision, and timeliness is set to that of the enterprise, not of the data warehouse tables.

5. Star Schemas Data Marts are built from corporate data warehouse.

Key Data Modeling Differences Summary

1. Inmon's approach is NOT for directly building data marts. Rather it's to build Enterprise Data Warehouses from which data marts are generated.

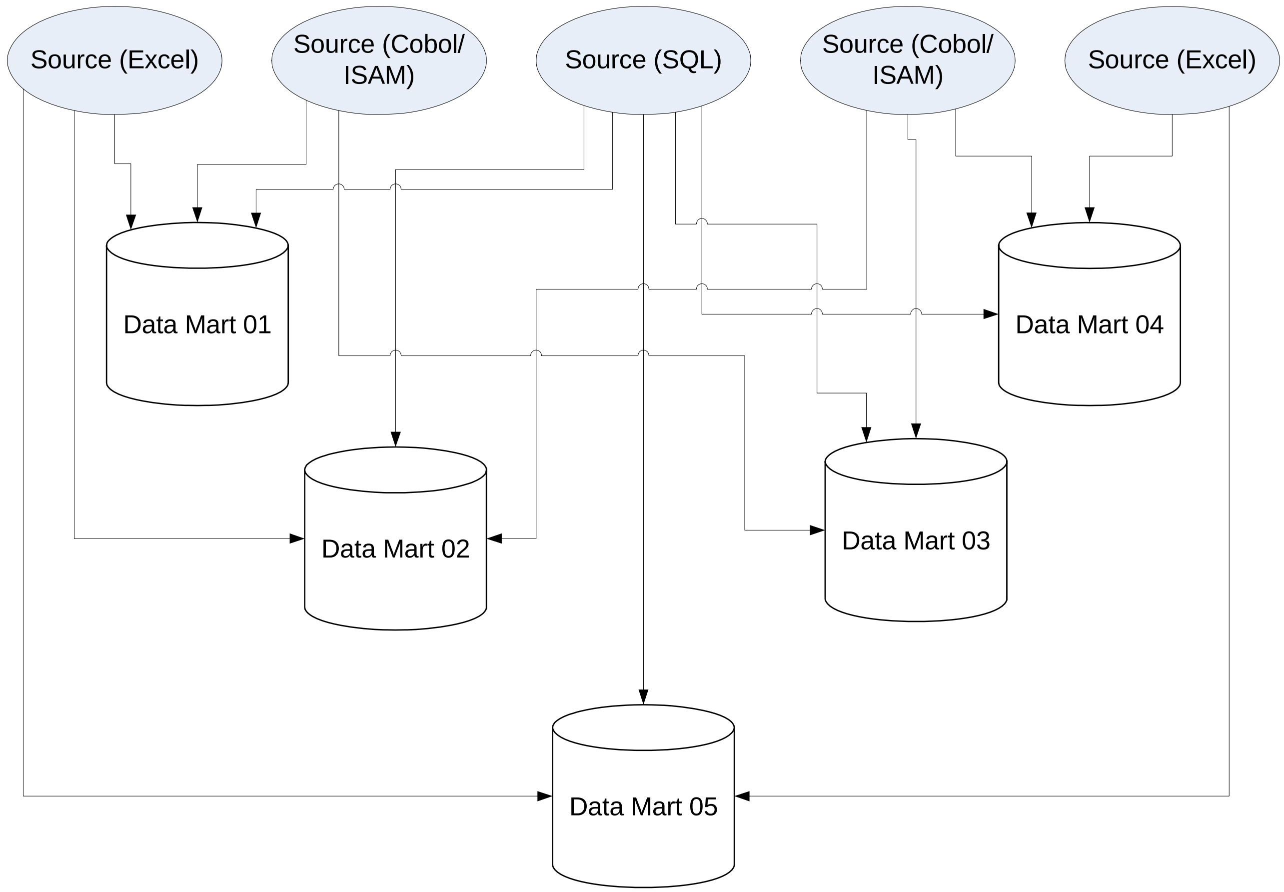
2. Kimball's approach is to build collections of Star Schema data marts with shared dimensions. The Kimball EDW is THIS collection.

3. So really, arguing for a Kimball or Inmon approach is almost like arguing which is better, a car's engine or its transmission. An argument based on a false premise.

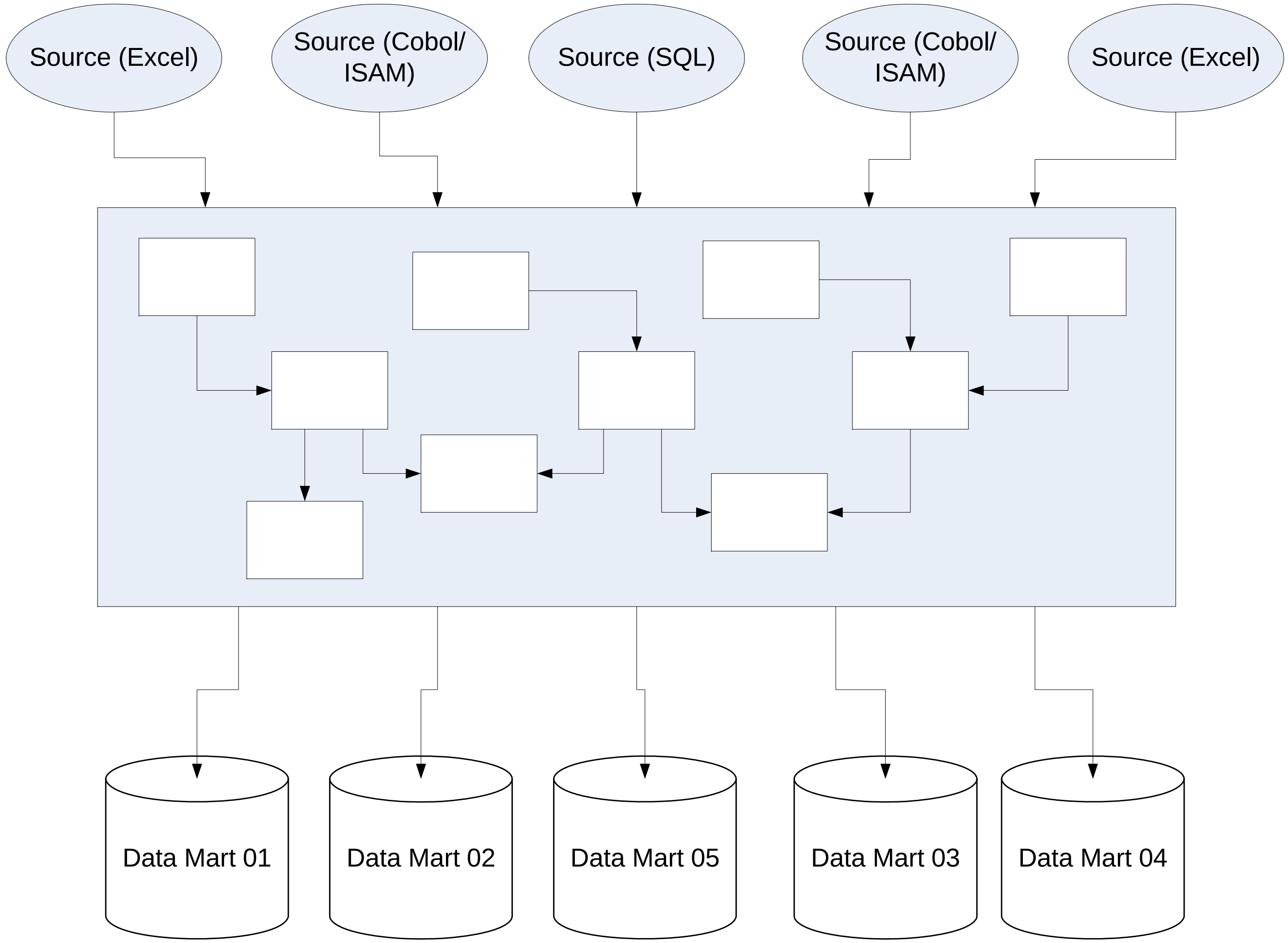
But if you are going to build Data Marts, the Build Strategy Differences are worth noting.

5.5 Key Build Strategy Differences

Star Schema Extract Transform Load (ETL) Strategy (worst case?)



ER-Model Extract Transform Load (ETL) Strategy



Critical Statistics	
Kimball	**Inmon**
15 sets of business rules for granularity, precision, and timeliness.	10 sets of business rules for granularity, precision, and timeliness.
15 sets of value domain engineering.	5 sets of value domain engineering.
15 sets of ETL software processes.	10 sets of ETL software processes.
Hi probability of conflicting Business rules for the 15 ETL Scripts.	Low probability of conflicting Business rules for the 10 ETL Scripts.

Counts drawn from strategies contained on Slides 25 and 26.

Issue	Critical Comparison	
	Kimball	**Inmon**
Consistent Semantics	Risk if each ETL and DM has its own.	Resolved and able to be controlled.
Synchronized Granularity and Precision	Risk if each ETL and DM has its own.	Resolved and able to be controlled.
Agreed-upon Timeliness	Risk if each ETL and DM has its own.	Resolved and able to be controlled.
Quality engineered two-way relationships versus collapsed hierarchies	Risk because there's not highly engineered pair-related sets to draw from.	Resolved and able to be controlled.

Comparisons drawn from strategies contained on Slides 25 and 26.

6. Conclusions and Recommendations

1. Kimball Data Marts have value and are needed within narrow, well-defined decision domains.

2. Kimball Data Marts have an intuitive way of perceiving and processing data by end users.

3. Kimball's Data Marts built directly from sources are accomplished faster (i.e., Source to Data Mart) than Source-to-Inmon-to-Kimball.

4. Direct population of Kimball Data Marts from Data Sources creates risks in the areas of semantics, granularity, precision, timeliness, and uncontrolled value domains.

5. Proliferation of Data Marts can cause more data and semantics stove-pipes if there's no enterprise-wide data-centric engineering and architecture.

Conclusions and Recommendations (cont.)

6. While a Inmon-Kimball strategy will take longer than one-off Kimball data marts, it has positive effects on synchronization of semantics, granularity, precision, timeliness, and controlled value domains across Kimball-based data marts.

7. Direct querying an Inmon Data warehouse requires more specialized and advanced database knowledge than from Kimball Data Marts.

8. Inmon data warehouse design is optimized for data loading and update.

9. An Inmon-Kimball architecture has greater stability, flexibility, and evolution capability over either architecture individually.

Bottom Line Recommendation

1. Build Inmon data warehouses at the Subject Area database scope.

2. Build Kimball Star-Schema data marts from Inmon data warehouses.

3. Inmon-Kimball is preferred over either Inmon or Kimball alone.