Jacob Dineen

IST 722  
Data Warehouse

Notes

**Logging on to school remote server:**

Sura VPN on.

Ist-rl-jafar.ad.syr.edu

[Jdineen@ad.syr.edu](mailto:Jdineen@ad.syr.edu)

Jake8579!

Access to the **ist-cs-dw1.ad.syr.edu** Microsoft SQL Server.

**Live Session 1- Breakout Notes:**

**Why is data an organizations more important asset?**

- without data we can't really do anything. Data Is used to look at the past, and to analyze the future (retrospective/forecasting). Used for business making, strategy. Record of what happened/happens.

**What are the 4 characteristics of a data warehouse and provide your own novel examples of each?**

subject oriented (sales can be a particular area),

time variant (data changes from time to time, every transaction is history),

nonvolatile (keeping historical data, never changed),

integrated (integrates data from multiple resources)

**You probably noticed we made a "copy" of the data from the source system to the data warehouse. Can you think of three reasons why the data must be a copy and you cannot just use the original data?**

The structure is different from the data warehouse from the source system. Warehouse is optimized for the querying.

Tuning is different. Writes/Reads vary from DBSM. One is for queries/reports

Serves as a backup (only point in time, not from time)

Staging Process – Way to introduce time variance to the data. Eliminates the load on the source system (don’t want to slow system down querying for data warehouse). Warehouse data comes from the staging process. Start the consolidation process.

**What is the difference between business intelligence and data warehouse?**

- Business Intelligence is about using data to answer business questions

- Data warehouse is about efficient data storage so we can perform business intelligence

Warehouse is the supplier of the data. BI is user facing (dash/report/chart)

Hard to do BI without data warehouse.

Nouns are master data

Verbs are business processes

Primary Key is a rule that makes data unique within a table.

Business Key is a user facing value that someone would recognize as being unique about a record – NetID, SUID, Email, etc..

**Asynchronous 2: Data Warehouse Architectures Notes**

**Technical Architecture**- Logical. How data moves around. ERP < ETL < DW

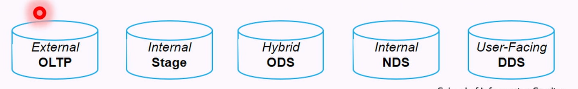
Data Store- Represented by the cylinder. Any place we can get data and that data is at rest.

User Facing- End User

Internal- Used by Data Warehouse. Not open

Hybrid- Combo of UF and Internal

External- Not part of Data Warehouse

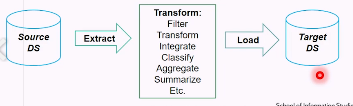


**System Architecture**- Physical Infrastructure. Hardware/Software.

**ETL**= Extract Transform Load

Moving data from one store to another.

Transformation may occur.



**Stage Data Store**

Internal data store. Not user facing

Stores extract from source system. Reduces contention with source system. Change Detection. Snapshot data of a point in time.

**Common Data Stores:**

NDS- Normalized Data Stores (Inmon).

Internal Data Store. Not user Facing

Stored in 3NF

Used as a source for data marts

ODS- Hybrid Data Store

Internal and user facing. Integrated, detailed, volatile. Current data.

Used in analytics that need to be represented in real time.

Consolidated from multiple sources.

DDS- Dimensional Data Store

User Facing.

Subject Oriented. Time Variant. Stored in dimensional format (Star Schema). Cube in multidimensional database.

Fact table is the middle of the star schema. Dimension tables are the outer layers.

DDS to MD DBMS (Cube)- Very fast query performance.

Metadata = Data about data.

Internal. Technical (Infrastructure)/Business (User) /Process (System)

Common Technical Architectures

Independent Data Marts – Source System < DDS(s).

Centralized- OLTP Source System < Stage < DDS. Consolidated into single data store. Lack integration.

Enterprise bus architecture- OLTP Source System < Stage < DDS. Conformed dimensions- Kimball technical architecture.

Hub and Spoke- Normalized data store introduced (Inman) OLTP< Stage < NDS < DDS(S)

Federated with ETL- Multiple Data Warehouse < ETL < FDW. Applicable during mergers.

Immon’s Corporate Information Factory

Inmon = Hub and Spoke architecture.

External World – Sourced from anywhere. Ex: Facebook Feed, Twitter Feed

I&T – Integration and Transformation Layer.

Takes nonintegrated data from sources and consolidates it. Staging data store and ETL. Structures unstructured data.

EDW, ODS, Data Marts

Inmon only calls EDW data warehouse.

Enterprise Data Warehouse – Subject oriented. Stored in 3NF. Use as a source for data marts and decision support systems. Heart of the CIF. Grows over time.

ODS V EDW – Very similar. ODS is point in time. EDW/NDS supports historical decision making. ODS gets updates/overwritten. Cannot coexist. Not uncommon to have one of each in company.

Data marts are sourced from the EDW. Stored in dimensional models. Source data for OLAP (Online Analytical Processing)

DSS Applications (Decision Support Systems)

Sourced from EDW. Used for Business Analytics/ Predictive Analytics

Cross Media Storage- Stores unfrequently accessed historical data.

Common System Architectures

SMP- Single System with multiple CPUs. Shared bus, memory, and IO. Microsoft, Oracle, IBM

MPP- Interconnected SMP nodes to form single cluster. Very vertical/high end.

Hadoop/Map Reduce- Commodity solution to MPP. Processing and Storage done independently.

**Synchronous (LS) 2: Data Warehouse Architectures**

Business Key – Unique in the source system.

DDS < Data Mart (Star Schema) < OLAP/Cube

**CIF**

Informatica – Cross Media Storage

PostgresSQL - EDW

Pentaho – ETL

Essbase- MOLAP

Birst- Consulting Data Warehouse company

Inmon – 3NF (NDS)

Kimball – Dimension and Fact Tables (ODS)

Homework 2- Make sense of the market.

**Project**

Group Members Emails

Denise Dunham ddunham@syr.edu

Qunfang Wu qwu114@syr.edu

Anug Dubey adubey@syr.edu

Building a warehouse (Kimball)

Total Orders by Customer/Which employees responsible for which order (Pick a couple to model)

Database given to group (Move data with ETL) < Build Star Schema < Build analytics

Profile Data – Master Data/Business Process

**Synchronous (LS) 3: Building the Data Warehouse**

Primary Development Methodologies

User Centric- About the people who use the data

Waterfall Model. Start with the users’ needs first. Linear in Nature. *Top Down approach. Starts with feasibility and requirements.*

Data Centric- About the data itself

Spiral Model. Data Centric. Starts with the data, end with the user. Iterative in nature. Bottom up approach. Starts with architecture.

Kimball – User Centric. Enterprise Bus TA. Waterfall model. Top down

Inmon- Data Centric. Hub and Spoke TA. Spiral Model. Bottom up

The data-centric approach focuses on what you have. It does not (at least initially) consider the needs of the users.

The enterprise bus technical architecture consists of dimensional models. Those dimensional models are built with a user-centric approach. They focus on the needs of the business users and not on the organizational data.

Therefore, it would be difficult to take a data-centric approach because the enterprise bus architecture supports the needs of the user.

DW Maturity Model

Prenatal, Infant, Child, Teenager, Adult, Sage

Productions Reporting, Spreadmarts, Datamarts, Data Warehouses, Enterprise DW, Analytic Services

We learned earlier that some technical architectures are more complex than others. As your data warehouse matures, so should your technical architecture.

For example: In the data marts phase you might have an independent data mart TA. To get to a legitimate data warehouse, your next TA might be centralized. To make it to enterprise data warehouse, you will need hub and spoke or enterprise bus.

Kimball Terminology

Program: Collection of coordinated projects. EG. Data marts for internet sales, store sales, and partner sales

Project: Single iteration of entire cycle. EG. Data marts for internet sales.

Program< Project < Data Mart

Charter: Define the scope and the background of the project. Identify success criteria. Business Justification

Project team: Business Lead, Project Manager, Business Analyst, Data architect, ETL architect, BI architect

Communication Plan: Keeping people informed. How often will you meet? Who needs to be present?

Create Project Plan and task list

Use change log or issue tracking system

Hold a kickoff meeting to get everyone on the same page.

Requirements Gathering

Event: A frequent activity within the business. Also known as a business process. EG. Student registers for a course

Status: Condition of an object at a point in time. Helps identify workflow. EG. Student has registered < Now receives grades

Level: Quantitative measurement of an object at a point in time. Period Snapshots: EG. Credit Card balance. Student GPA

Roles: The who/ What/ When of the event/status/level

Function requirement – Defines what system does or should do. Address needs of business users.

EG. Analyze sales overtime by geographic region, customer segment, or sales territory.

Nonfunctional requirement: guide and constrain the system architecture.

EG: Max query response time < 20 seconds

Key activities of req. gathering

Interviews/Observations

Data audits: data profiling to assess capabilities

Document interview write ups, Enterprise bus matrix, Prioritization grid, Issues list

^ What type of routine analysis do you perform? What data are used? What do you do with the data?

Functional Req. to Dimensional Model

Identify the bus process and the bus process type

Identify the facts of the process – Quant values we measure

Identify the attributes in dimensions of the bus process. Sort/Filter/Search/Aggregate

The roles by which we measure the bus process.

Creating Dimensional Models from Functional Requirements

Identifying 3 types of business processes

Events or Transactions- One row per line. Basic fact grain. Space and time. Once inserted, not revisited or updated. Rows inserted into fact able when a transaction occurs.

Workflows based on object status (accumulating snapshots)- Based on a status. Used to capture a business workflow. Fact row is initially inserted and then updated as milestones occur. Fact table has multiple dates that correspond to each milestone and records a change in status. Special facts: Milestone counters and lag facts for length of time between milestones. Order fulfillment, applicant tracking, rental cars.

Quant measurement of points in time based on level (Periodic Snapshots)- predetermined interval. Daily/weekly/monthly. Complements detailed trans. Facts but does not replace them. Financial reports, bank account values, semester class schedules

Business Processes end up being FACT tables in a ROLAP star schema – Design phase they are called dimensional models. Star schema is the real thing. Dim. Model is the design of that thing.

Identifying Facts of Business Process

Facts are quant numerical values associated with business processes. How much? How many? How long? How often? If not tied to the business process, is not a fact.

Point scored in a game < Fact

Player height < Not a fact

Additive- Facts can be summed across all dimensions

Semi additive- cannot be summed across all dimension, such as time periods

Non additive- cannot be summed across any dimension

Dimension provide context for our facts.

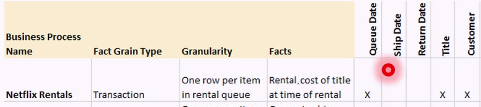
We can easily identify dimensions because of the by and/or for words. EG. Total Accounts Rec for the IT Department by month

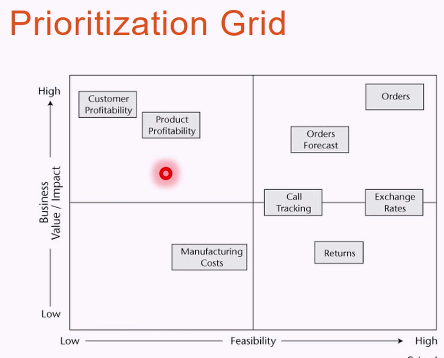
Dimensions have attributes that describe and categorize their values. EG. Student: Major, Year, Dormitory, Gender

Help to contain and summarize facts.

Document Functional Req.

Enterprise Bus Matrix –





**(LS) 3: Building the Data Warehouse**

Dimensional Modeling sheet for Homework 3 -- Northwind

**Asynchronous 4: Dimensional Modeling**

Dimensional Model Design

Dim Modeling – Logical Design Technique for structuring data.

Intuitive and Fast

Fact Table + Dimensions it requires

Implemented in relational DBMS as star schemas (Rolap). They exist in MOLAP databases as cubes

Fact Table: quantifiable performance measurements. Has FKs to each of the dimensions

Dimension Table: Table of contexts for the facts

Attribute: Characteristic of a dimension

Star Schema: Connections among facts and dimensions that define a business process

**Rules of Fact Table Design**

Primary Key should use min number of columns possible. No surrogate keys

Referential integrity is a must. Every foreign key in the fact table must have a value.

Avoid nulls in the foreign key by using flags, which are special values in the place of NULL

Granularity of Fact Table is at the lowest, most detailed grain.

Each fact should be additive across all dimensions, or redesigned to be as additive as possible

Each fact must be of the same granularity.

Should always use an integer as a foreign key.

**Factless Fact Table**

Business Processes that do not generate quant measurements (Student attendance, College Admission)

Can be easily converted into traditional fact tables by adding an attribute, count, which is always = 1

Consider adding facts for when the event did not happen

**Rules of dimension table design**

Verbose Attribute Values: Should be as descriptive as possible

Descriptive Columns: Should be easy to tell what the column means.

Complete: Every column should have data in it. No Null/Empty Values

Discrete Values: One business entity value per row

Quality Assured: Data are clean and consistent

Every Dimension should have a natural of business key from the source system + a surrogate primary key.

Surrogate Key – Autogenerated Numbers.

Yield best performance.

No dependency on the primary key in operational source data.

**Slowly Changing Dimensions**

Dimensional data changes infrequently.

Customer address change, or name change.

1. Overwrite existing attribute
   1. Correcting mistakes or errors
   2. Where historical doesn’t matter
   3. Old value has no significance
2. Add a new dimension row
   1. Most popular. Preserves History
   2. Natural key is repeated.
   3. Old and new values are stored along with effective dates and indicator to which row is current. (Type 2 metadata)
3. Add a new dimension attribute
   1. Infrequently used. Preserves history
   2. Old value is stored in new column. New value is written to the existing attribute. (Type 3 metadata)

**Rapidly Changing Dimensions**

Rapid is loosely defined as not often with no consistency. EG. Customer shirt at time of purchase.

**Degenerate dimensions**

Value that changes so frequently we store in the fact table.

Any attribute in the fact table that is not a dimension key or fact is considered a Degenerate dimension.

**Mini dimensions**

Rapidly changing dimensions and Slowly Changing dimensions in the same table.

**Conformed dimensions**

Mater or common reference dimensions.

Shared across business processes (fact tables) in the DQ

Reusable – Drill across

Lower time to develop next star schema

Identical dimension

Perfect subset

**Conformed Subset**

Business process should contain similar dimensions that match granularity of business process. Create month table from the date table if needing periodic snapshot

**Role-playing dimensions**

Same dimension is used more than once within business process.

Common among date dimensions.

Order/Shipped/Delivery Date

Multiple FKs back to same dimension table.

**Date and Time Dimensions**

Every Fact table has a date or time dimension

One row for every date

Usually generated programmatically during the ETL or imported from spreadsheet

Acceptable to use PJ in the form of YYYY/MMDD

Time of day = separate dimension for time

Time of day = only if meaningful textual descriptions of time (lunch, dinner, first shift, second shift)

Elapsed times intervals are facts

Time dimension = 1 row for each second of the day

**Junk Dimensions**

For misc low cardinality flag. Empty until transactions occur

**Handling Time Zones**

Store in UTC time

Also express in local time zone

Conformed dimension used across multiple business processes.

Roleplay dimension used within same business processes.

Date is both ^^

**Asynchronous 5: Dimensional Modeling Development**

Physical design = creation of actual artifacts as part of the design.

Implemented in RDBMS

Star Schema with Kimball architecture

Making tables keys and constraints

Setting up schemas synonyms and views

External Model: User’s view of implementation (Views)

Internal Model: Actual implementation itself (Tables)

Process: Design < Dev Environment < Test Environment

**Developing Standards**

Naming Conventions/Consistency

Stage Table < Stg

Dimension Table < Dim

Fact < Fact

Dimension keys end with ‘Key’. Don’t use ‘ID’

Schemas are namespaces, which facilitate the separation, management, and ownership of database objects.

Objects are securable by schema

Create SCHEMA name

Synonyms and Views – Logical extractions of tables and SELECT statements

Create View name AS

Create SYNONYM name FOR

**Physical Design Guidelines**

Limit the bits of data types.

CHAR = Character

VARCHAR = Character when you don’t know how long the length is

Nchar = UNICODE set

Reference PDFS saved in DW Folder for more

Reference 5.7 + 5.8 for details on how to code staging tables

**Asynchronous 6: ETL**

ETL: ETL is code buy not typically written as code. We use tooling to write the code for us.

ELT : Transformed at the DW server, rather than the ETL server.

**Four approaches to moving data:**

Pull from source: Most common. ETL system connect directly to OLTP database.

Push from source: Triggers in source system pushes change out. Useful for replaying transactions and changes.

Export and push: A batch process performs an export from the source. Typical when ETL cannot query the source because it is not a DBMS

Pull from logs: DBMS transaction log records changes. A log reader reads the logs.

Three Approaches to ETL Processing

ETL Server: Does not stress source of target. Best and most common

Data Warehouse: Save on licensing costs.

OLTP Source: Solution for real time data warehousing.

**All other files saved in Syr/DW/Week6**

**Asynchronous : ETL Development**

SSIS = Sequel Server Integration Tool (Visual Studio).

ETL Tooling is a visual depiction of how your data flows – Tooling writes the code for us

Understanding Analysis Services

Microsoft SQL Server Analysis Services allows us to build multidimensional OLAP solutions. We can deploy these solutions to a special database, which users can connect to and perform ad-hoc interactive reporting. The OLAP database differs from a traditional database in that it supports:

 Rich metadata in our dimensions—for example, we can sort by month number but display the month’s name so that the ordering of months is correct: Jan, Feb, Mar, etc.

 Fixed and ragged hierarchies.

 Detection of measures in our fact tables and their configuration as additive or semi-additive.

 Perspectives for limiting views into the OLAP database.

 Row-level security over our dimensional model.

 Aggregations—pre-calculations for performance improvements.

 Calculations, KPI’s, data mining and more!

Analysis services consist of two components:

1) **SQL Server Data Tools—**We use this to build our analysis services databases.

2) **SQL Server Analysis Services**—We deploy our analysis services databases to this server to test and deploy our packages

Terminology

 **Data source**—Connections to relational databases and other sources of data.

 **Data Source View**—The logical model of the schema used by analysis services. Typically comes from one source, but can be derived from several sources. You can create joins in your data sources that do not exist in your ROLAP model.

 **Cubes** —A set of measures and dimensions used to analyze data. This is the primary delivery mechanism for OLAP services.

 **Dimensions**—As the name implies, these are dimensions. Typically, there are additional rows in the dimension tables, such as audit columns and SCD columns, that we do not wish to include in the OLAP presentation. Also, we can rename our columns in our MOLAP dimensions, provide different display and sort values, and create hierarchies.

 **Mining Structures**—Defines a domain for a data mining models.

 **Roles**—Allows you to secure dimensions, measures, and/or values in the cubes. Your NetID requires admin access to your **ist722\_*yournetid*** database. Without it you will not be able to deploy the database.

 **Assemblies**—Code libraries that are required by the project. These typically include custom data sources or data mining models.