Data Mining Database Design
CAS Predictive Modeling Seminar

Jeffrey White, FCAS, MAAA

October 6-7, 2008

Topics Covered

- Why create specific Data Mining/Actuarial Data Store?
- Data Store Definitions and Tool Selection
- Considerations in the design of the Data Store

Why Data Mining/Actuarial Dedicated Data Store?

- Data Stores are used for different purposes
  - Operational
  - Financial
  - Analytic
- Data Stores can unintentionally apply to more than one purpose
- Data Mining/Actuarial needs are often not met by any of the above purposes
- Dedicated Data Store allows the department control over the environment to optimize flexibility, scalability, and uniformity

Dedicated Data Store - Flexibility

- A proper data store can be used for all actuarial purposes, not just data mining
- Analytics require multiple views of the data
- Traditional IT solutions can be too restrictive

Dedicated Data Store - Scalability

- Storing large amounts of actual data is only the tip of the iceberg
- Need to access and analyze the data efficiently
- Need to allow for data to double every two years
- Hundreds of iterations make predictive modeling very data intensive
- Predictive modeling is a continuous activity that lasts well beyond implementation
- Traditional IT designs will drop older data

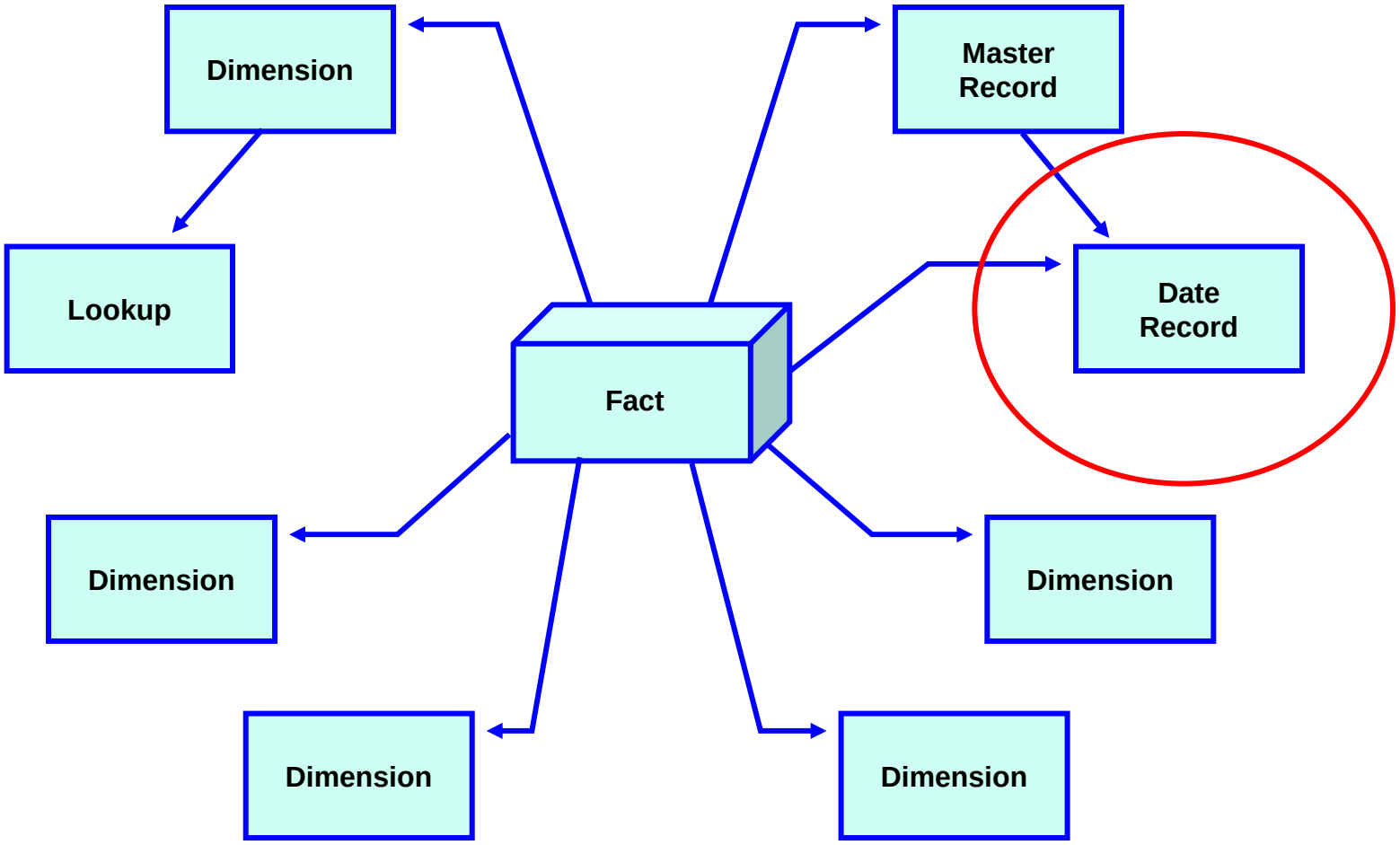
Dedicated Data Store - Uniformity

- Data comes from many internal sources
- Data comes from different external vendors (MVR, Geo Coding, etc.)
- Extract, translate, load required for efficient end user usage
- Traditional IT single source Data Warehouse or Data Mart
  - may not exist
  - may not contain all needed information (internal and external)
  - may not contain valid information
- Data Store design should combine flexibility, scalability, and uniformity considerations

Traditional IT Data Store Definitions

- Facts = values on which to operate
  - Includes premium, exposure, claim counts, loss dollars
- Attributes = information related to the values
  - Includes policy number, property location, date of loss, coverage
- Dimensions = Primary Attributes
  - Attributes used to segment the data for reporting
- Metadata = data about data
- Information can be both a fact and an attribute
  - Policy limit, policy term

Traditional IT Data Store Design - Star Schema



Traditional IT Data Store Tools

- Data Cubes
  - Like pivot tables
  - Dimensions allow drill down capability not found in standard pivot tables
  - Once designed, not easy to change
  - Single view of the data
- Business Intelligence Software
  - Flexible point and click access to data
  - Reporting capabilities
  - High level of maintenance
  - Single view of the data

Dedicated Data Store - Tool Selection

- Traditional Relational Databases have high overhead
  - Designed for operation, not analytic uses
  - Designed to get one record quickly, not manipulate data
  - Need to manage indexes and other database overheads
- Switch off or reduce unneeded features
  - Transaction logging and concurrency
  - Backup/recovery overhead
  - Manage specific security issues outside of the system
- Selecting the correct tool is critical
  - Use tools designed for analytics rather than operations
  - 80/20 solution to meet needs of all end users

Data Store Considerations

- Physical Design and Administration
- Aggregation of Data
- Transformation and Data Field Types

Data Store Considerations - Physical Design

- Physical versus Logical Structure
  - Do not let logical needs dictate physical structure
  - Use views to present the data to the end users
- Download data periodically from data sources
- Keep each period's data physically separate
- Keep physical data small and homogeneous
- Create summary data tables where appropriate
- Allow the data to be cumulated efficiently

Data Store Considerations - Star Schema

- Star Schema provides single view of data
- Data can be joined multiple ways (premium and claims)
- Requires overhead to manage and maintain table keys
- “Flat Files” are not bad
- Use only if software selection requires its use to maximize efficiency

Data Store Considerations - Administration

- Control your own data store administration
  - Indexing
  - Security
  - Metadata
- Keep data read-only
  - Do not let end users alter or delete data
  - Maintain reasonable backups for data
- Depersonalize non public personal information
  - Social Security numbers
  - Credit Card numbers
  - Drivers license numbers

Data Store Considerations - Aggregation

- Download as much detail as possible
  - Do not drop useful, populated fields from source data tables
  - Get one/get all concept
- Avoid summarizing the data, keep data transactional
- Except for key fields, avoid repeating data elements in different tables
  - Tempting for efficiency in coding
  - Mismatch problems
  - End user confusion

Data Store Considerations - Transformation

- Consistency is very important
- Data Field Names
  - Keep consistent across tables, sources
  - A good naming convention would be the one that is used in the predictive modeling implementation
- End user tables should look the “same”, regardless of the source that produced the data
  - Keep formats (type and length) the same across tables, sources
  - Remap codes to one standard mapping
- Employ reusable code as much as possible

Data Store Considerations - Date Fields

- Date fields are part of most data pulls
- Date fields are used both in the selection and filtering of data
- Many different kinds of date fields:
  - Accounting Date, Effective Date, Loss Date
- Date fields can be at the month, day, or time level
- Every record should have at least one date that shows when the record became effective
- Dates should be stored consistently regardless of data source

Data Store Considerations - Facts

- Do not store counts (policy, claim)
  - Unwieldy to store data at all necessary hierarchies
  - Obtain counts logically through code
- Store facts incrementally
  - Facts should be divisible across all attributes/dimensions
- Store facts in columns, not rows with a transaction code

Data Quality

- Clean data as much as possible
  - Make sure all data is valid (i.e. zip codes)
  - Solve data problems so exception coding is not needed
- Work with IT to solve problems at the source, do not correct downstream
- Be part of the solution, but not the entire solution