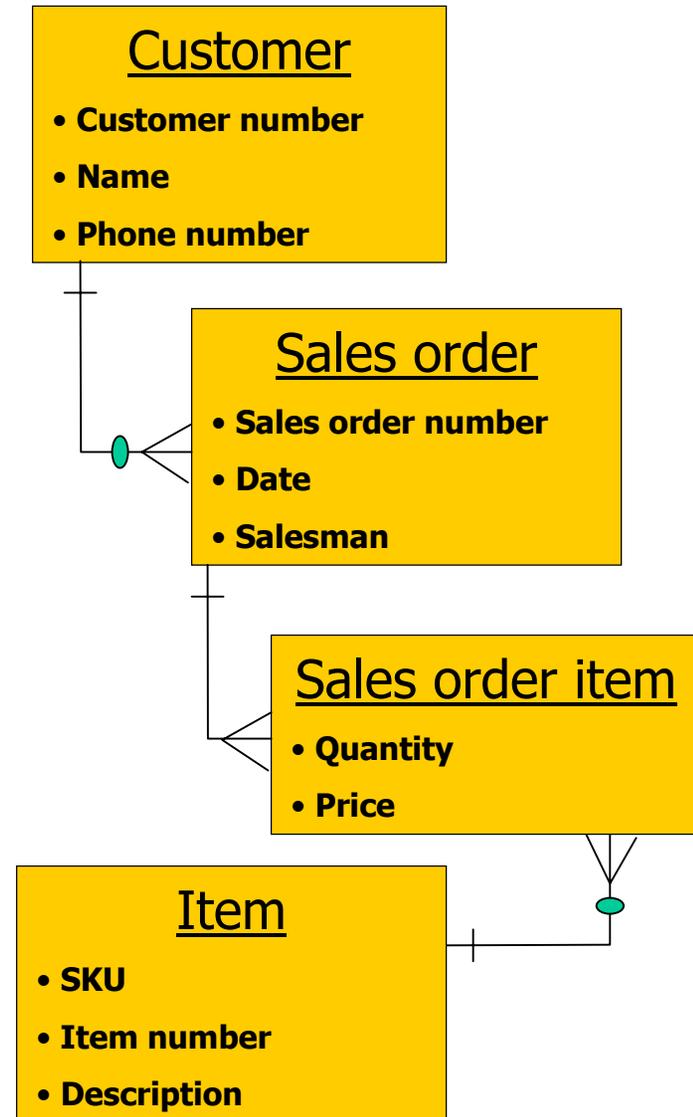


What is a data model?

- A model is a symbolic or abstract representation of something real or imagined
- A data model is an aid to help us visualize data structures and how completely and accurately they reflect our information system problem space
- A data model focuses on business meaning and solutions rather than technical implementation

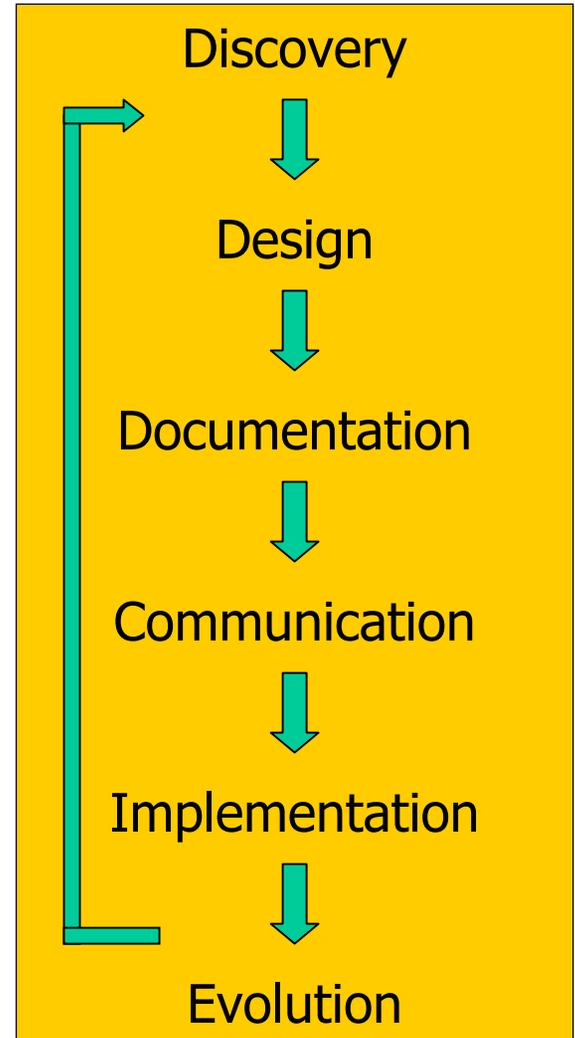


Why model data?

A sound database design **costs less!**

- Better match to requirements = happier users, less maintenance
- More understandable solutions = more utilization, less maintenance
- Fewer construction errors = less debugging, less maintenance
- More system flexibility = less maintenance
- Greater reusability of ideas = less maintenance

A data model gives us a chance to make design changes which cost less than code changes



Data model and MMIS?

Typically, one of the first applications an organization develops is one to facilitate some transactional nature of its business

Characteristics of a data model serving an MMIS application are:

- high degree of normalization which translates into
 - ➡ efficient transaction processing
 - ➡ increased quality of data
- maintenance of business rules regarding the integrity of the data via:
 - ➡ key definitions
 - ➡ constraint definitions

Why RDBMS for MMIS?

- Relational databases hide where the data is and how it is accessed = application developers can devote more focus to business and presentation layers
-

- RDBMS handles MMIS demands for many types of transactions
- Data is shared across major MMIS functional groups independent of their environment or platform
- Access to tables or subsets of data can be restricted for particular users depending on their defined role
- Changes to the database objects can be made with minimal or no impact to existing application processes

Why RDBMS for MMIS?

RDBMS provide tools and features that are essential in monitoring and tuning applications.

- System catalog = tracks details and statistics about user tables, columns, indexes, constraints
- Database logs = accurate, dependable recovery model
- Locks = provide transactional consistency to systems issuing many update requests
- Program explains = a performance analysis report

Performance edge

- Continuous SQL training for DBA and application development staff via:
 - local User Group meetings
 - Conferences
 - In-house training classes
- DBA review of SQL and access path information (walk-thrus)
- DBA use of monitoring tools to identify problem processes
- DBA staff being an important part of system design efforts
- Creation of a multi-functional group devoted to analyzing and improving program performance
- Staying current on new releases of system software
- Taking advantage of new functionality in system software