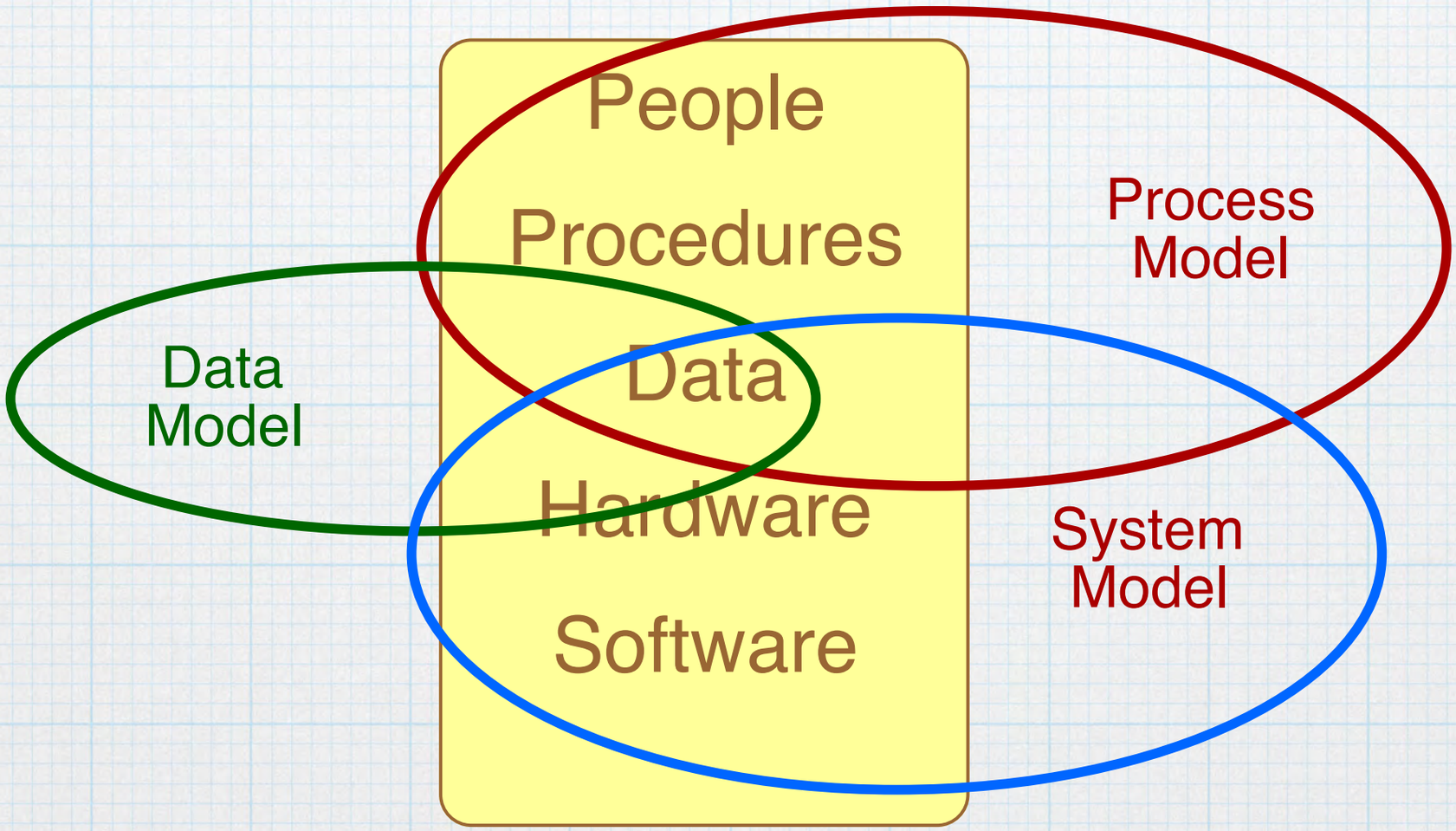
CS360

Business Information Systems Analysis and Modeling

System Models and Entity Relationship Models

Analysis, Modeling and Design

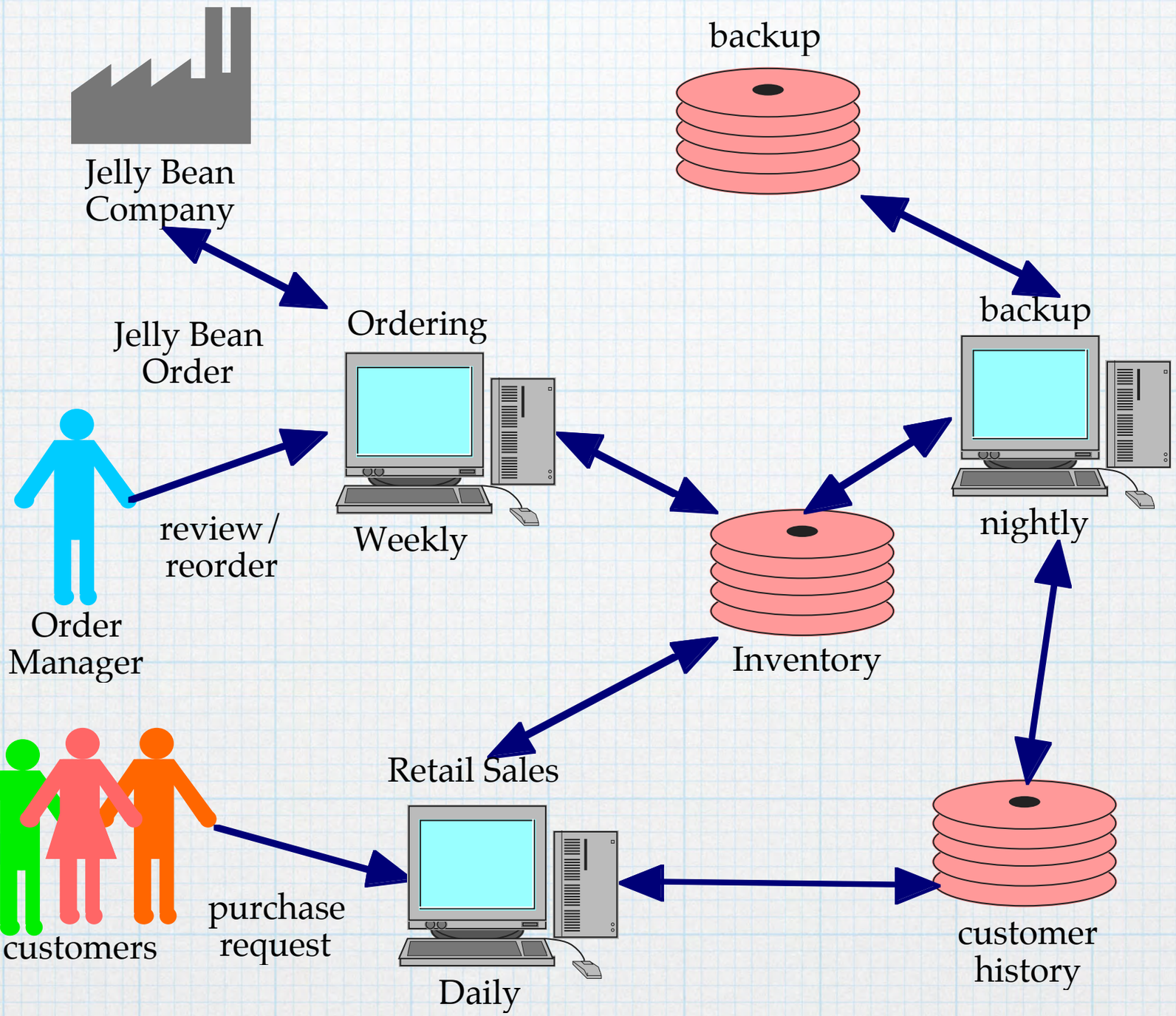
* Modeling
  * the capture of a subset of system characteristics relevant to a level or understanding or detail



System Models

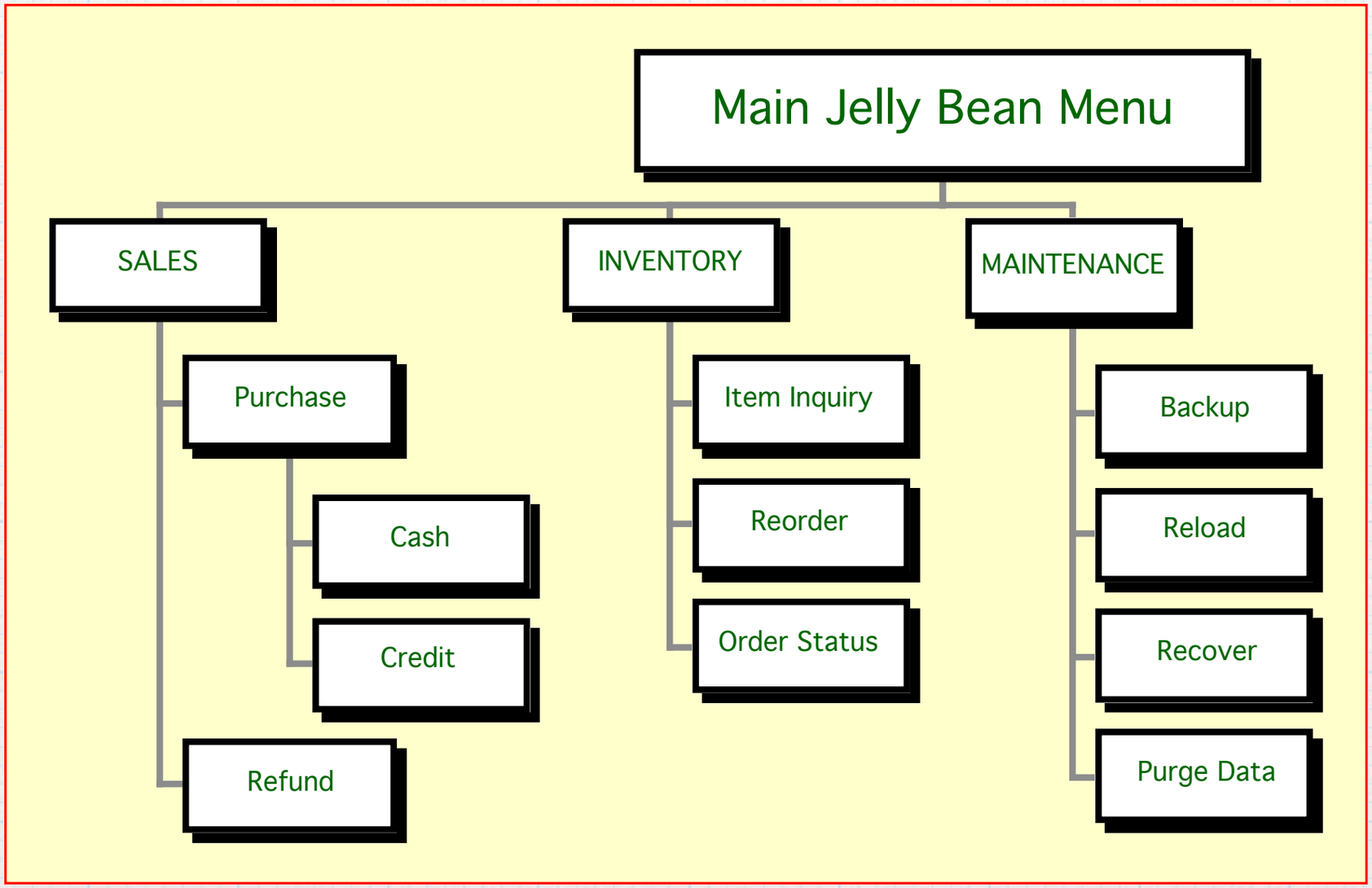
* System Models emphasize processing user interfaces
  * User's System Diagram
    * icon based
    * clearly describe user familiar system interfaces / controls
    * a gentle introduction to formal system specifications
  * Menu Trees
    * based on commonly used GUI control interfaces
    * focuses on actions / commands available to the user
    * categorizes and groups related system functions
  * System Flow Charts
    * ISO system components
    * focuses on relationships between data and software components
    * identifies configuration management items that must be maintained and controlled

User's System Diagram



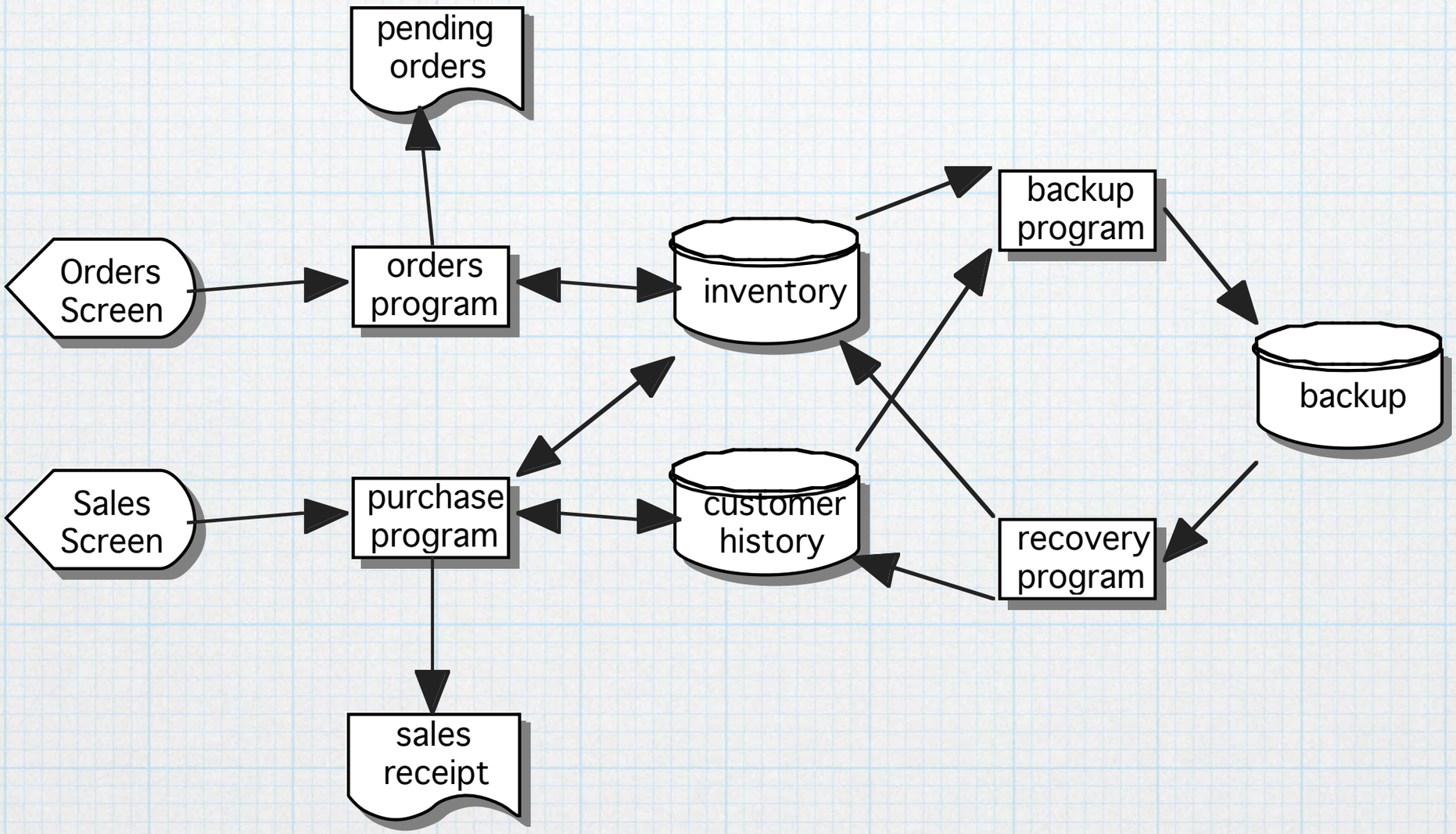
Menu Tree

* User interface function list



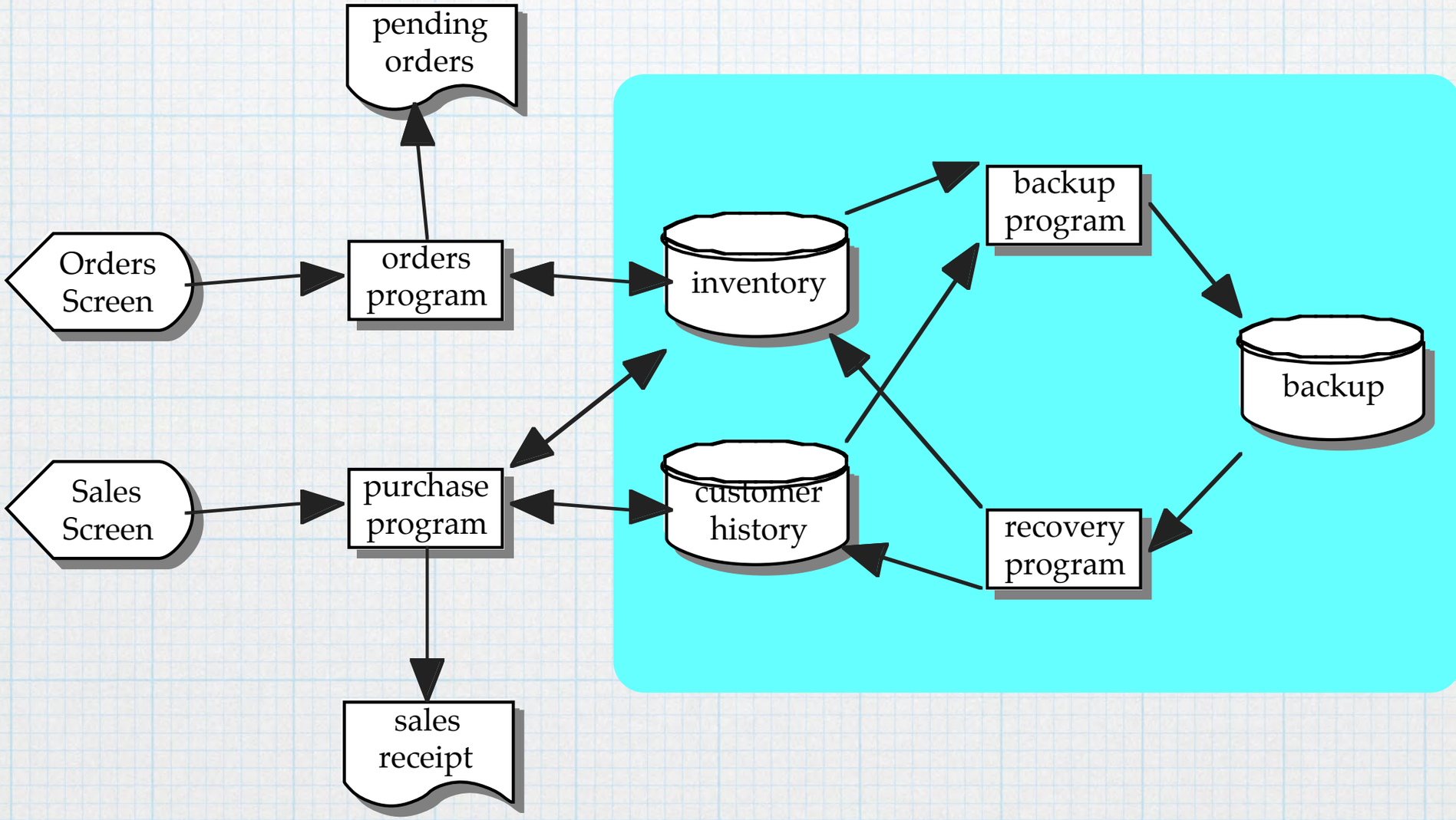
System Flowchart

* Relationships between data and software inputs and outputs



Subsystem Detailing

* Subsystems allow more focus on related operations in the context of the whole



Data Modeling

- Information Structure Support
  - facts
    - measurements / description
    - state designation
    - identifying
  - relationship
    - association
    - dependency
    - aggregation / membership
- Information Processing Support
  - collection
    - validation (existence, metric, range)
    - distinction (replacement, addition of facts)
  - retrieval
    - by identity
    - by classification
- maintenance
  - by identity and classification

Data Modeling Paradigms

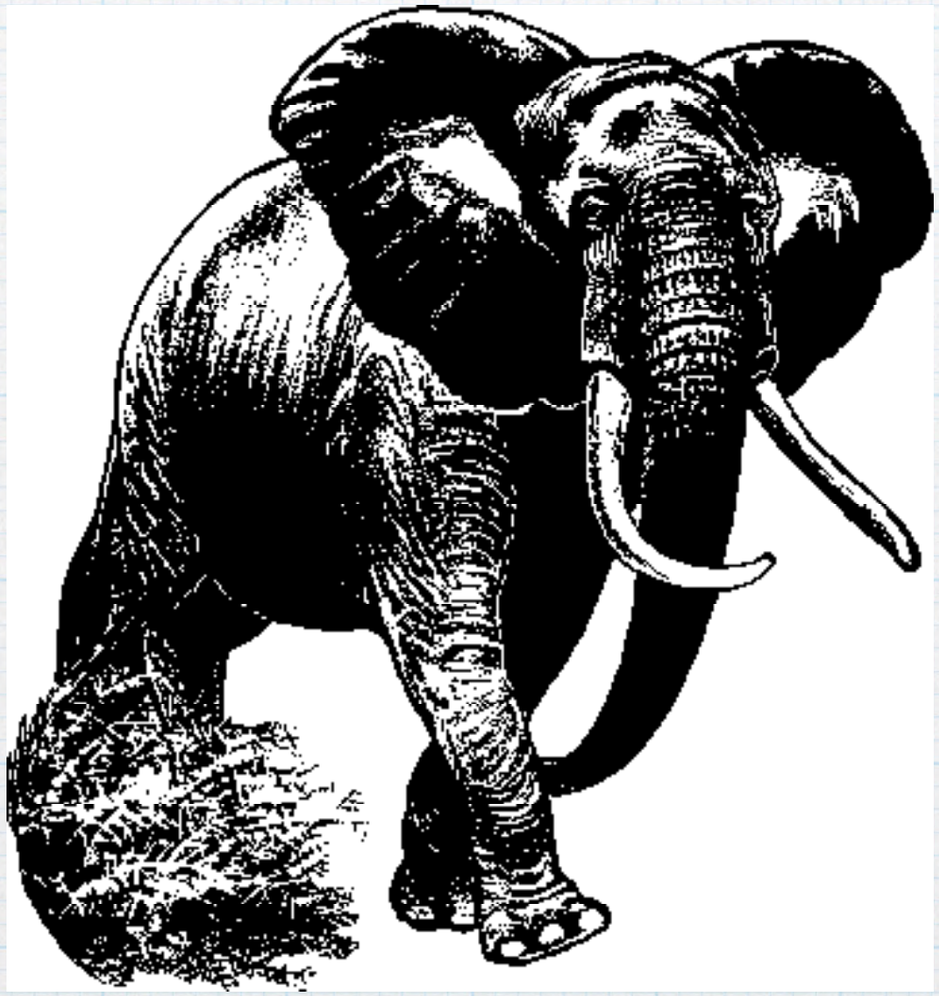
- Hierarchical
  - relationships based on parent/child
    - organizational chart
    - family tree
- Relational
  - relationships based upon information content
    - shared / common descriptive characteristics
    - identifying facts
    - formally consistent table structure
  - formal mathematics for information operations
    - relational algebra / calculus
    - non-procedural relational languages (SQL, QUEL, QBE)
- Network
  - relationships based upon association
    - telephone network
    - network of friends or business associates

Relational Data Model

* Formal Data Structuring Discipline
  * columns representing simple facts (attributes)
  * rows representing related facts (instances)
  * related facts defining an "entity"
  * rows + columns form TABLE aka RELATION
* Mathematical Operations on Relations
  * Projection: copying columns from tables
  * Selection: copying rows from tables
  * Join: building new rows by copying rows from two tables based on shared information
* Information Protection through Integrity Constraints
  * attributes are atomic
  * instances must be "distinguished" in a table
  * joins limited to "join compatible" attributes
  * attributes "clusters" based on functional dependency

Zoo Database

- Zoo keeper
  - feeds animals according to their diet
  - animals are housed in designated cages
  - meals are determined by a ration of foods
  - rations are taken from available food stores



Relational Table

* Tables, attributes, keys, instances

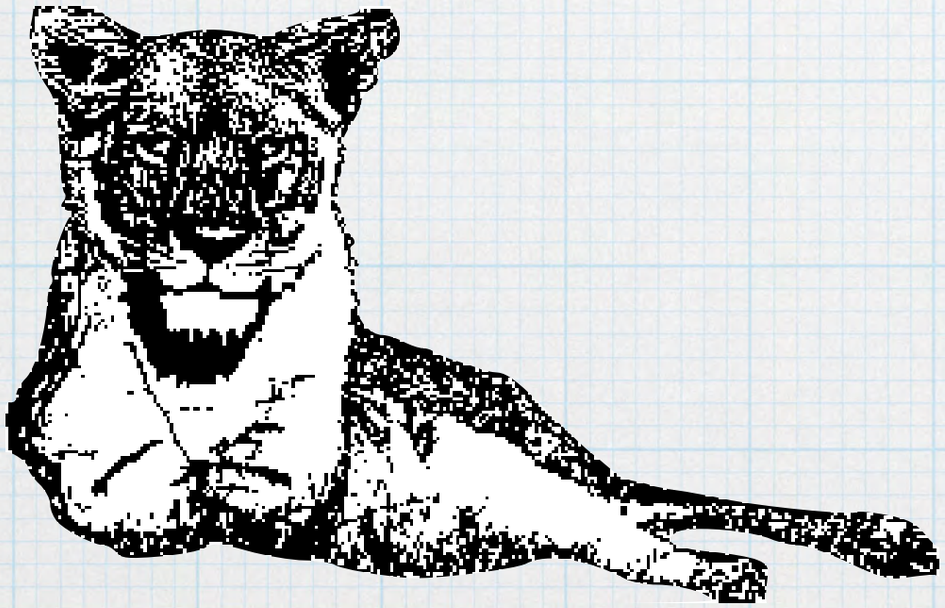
ANIMAL

Entity

Key →

ANIMAL
ANAME
ASPECIES
AGENDER
CLOC

Entity w/attributes

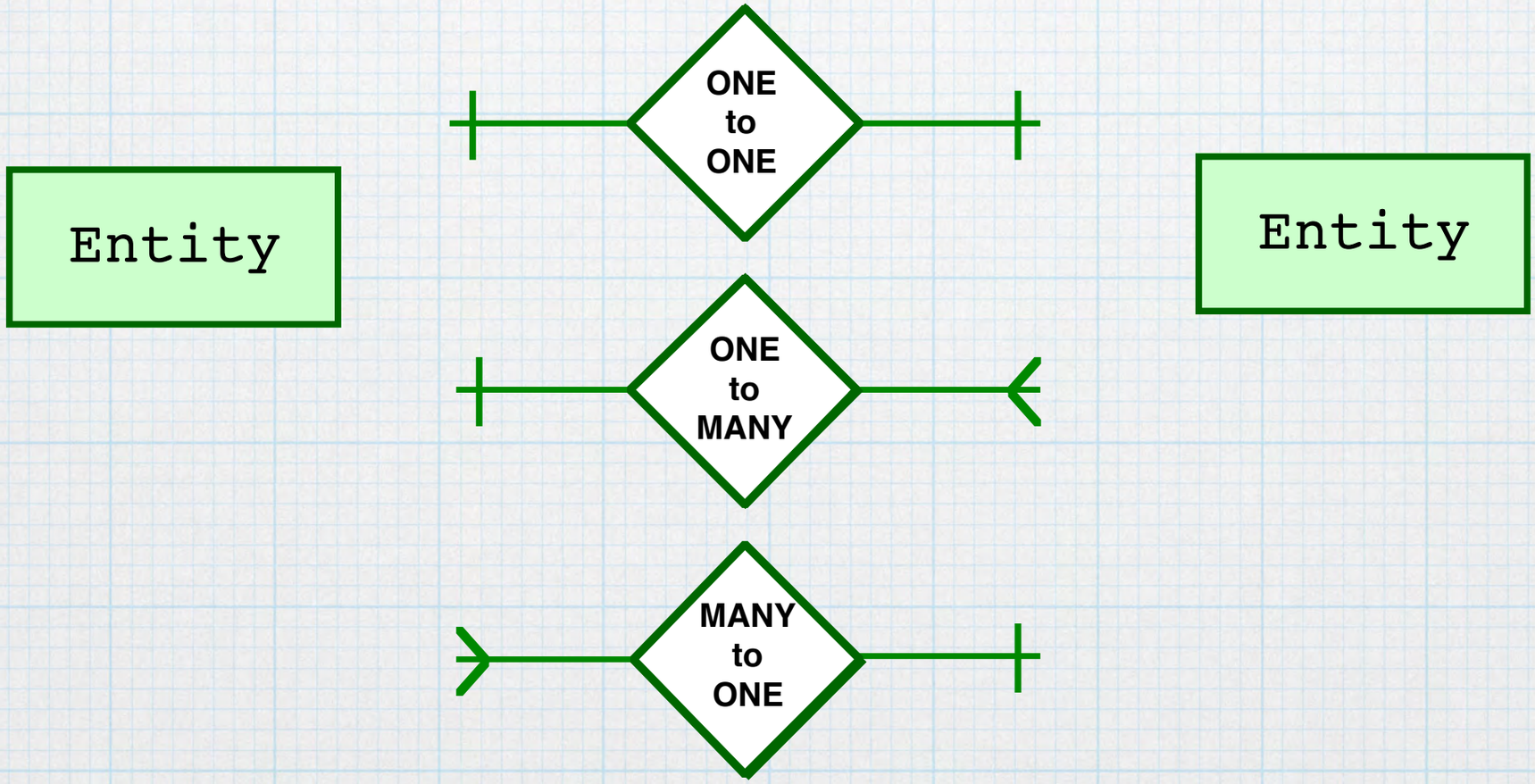


ANAME	ASPECIES	A	CLOC
LEO	LION	M	B2.1
MEO	LION	F	B2.2
ZEO	LION		B2.2
BLACKIE	PANTHER	M	B2.3
MOLLY	PANTHER	F	B2.3
JEFFREY	GIRAFFE	M	C1.1
MOLLY	DONKEY	F	C1.1
CONRAD	CAMEL	M	C1.1
FANNY	ELEPHANT	F	A2.1
MORRIS	ELEPHANT		A2.1
CHUBBLES	HIPPO	F	B4.1
SUZY	MONKEY	F	A3.1
SALLY	MONKEY	F	A3.1
SAM	MONKEY	M	A3.1
BOSS	APE	M	A3.2

Relational Table w/instances

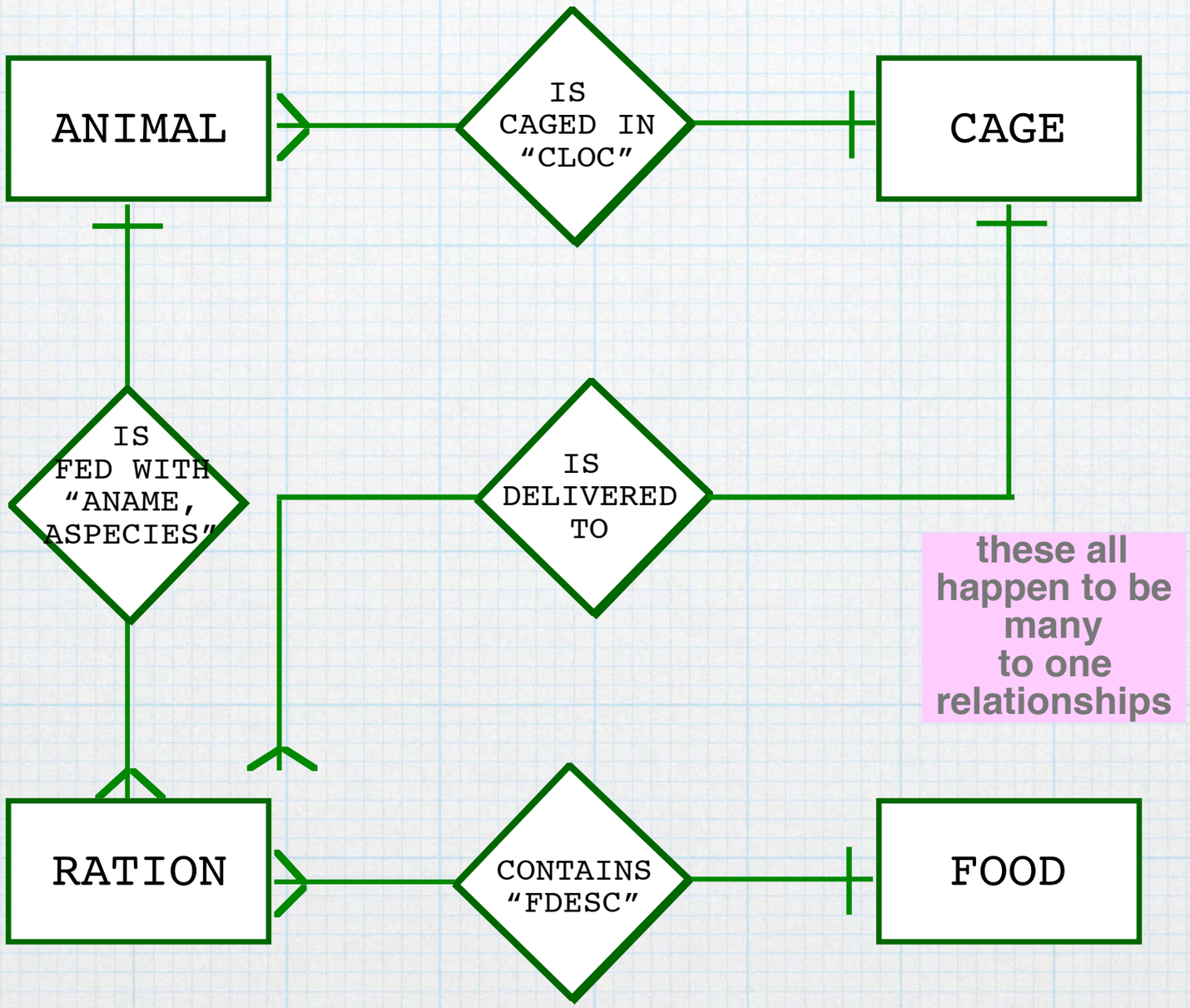
Relationships

* Relationships associate information found in two (or more) tables and define "cardinality" of the relationship



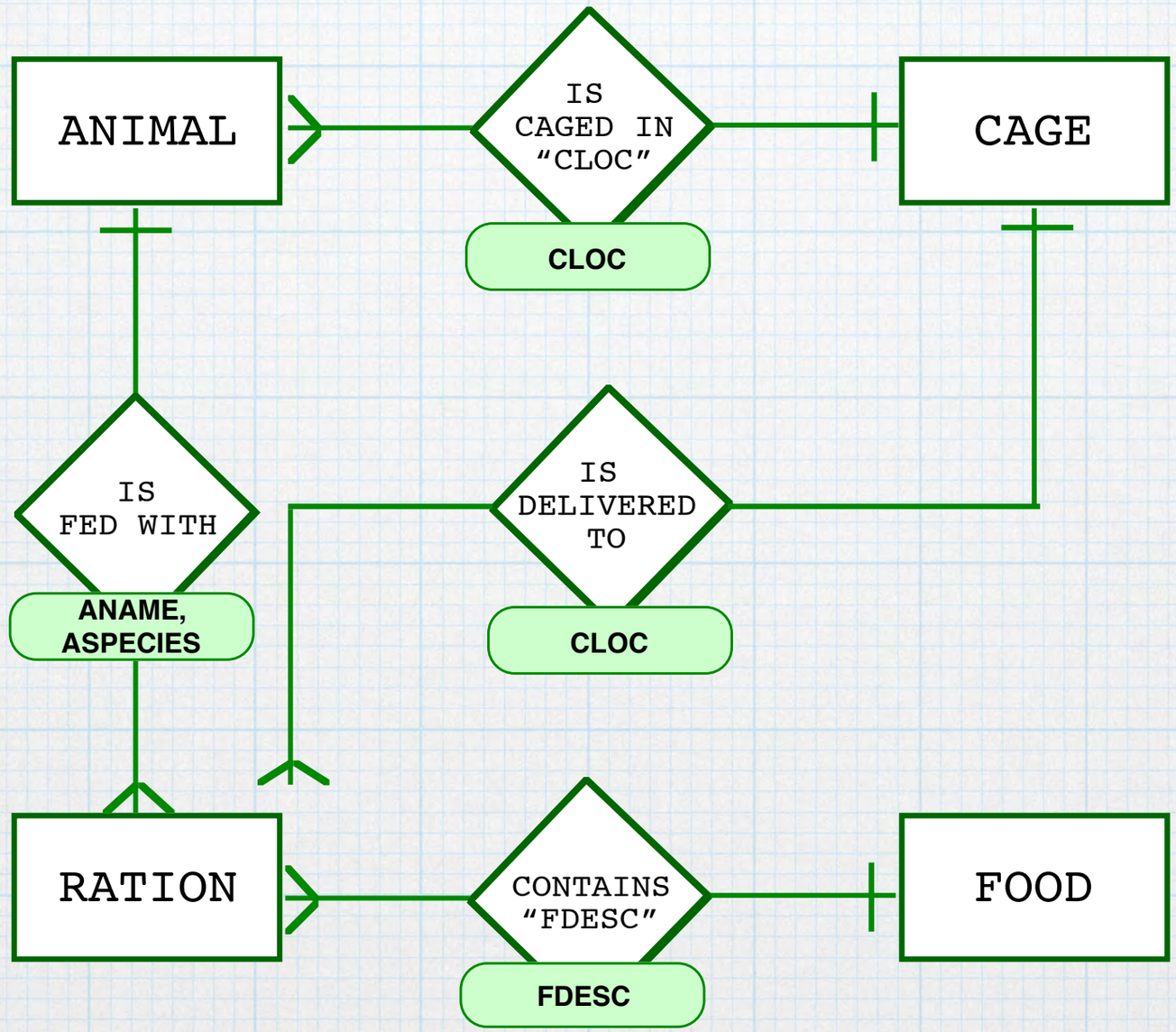
Relating Tables

* Keys logically connect information



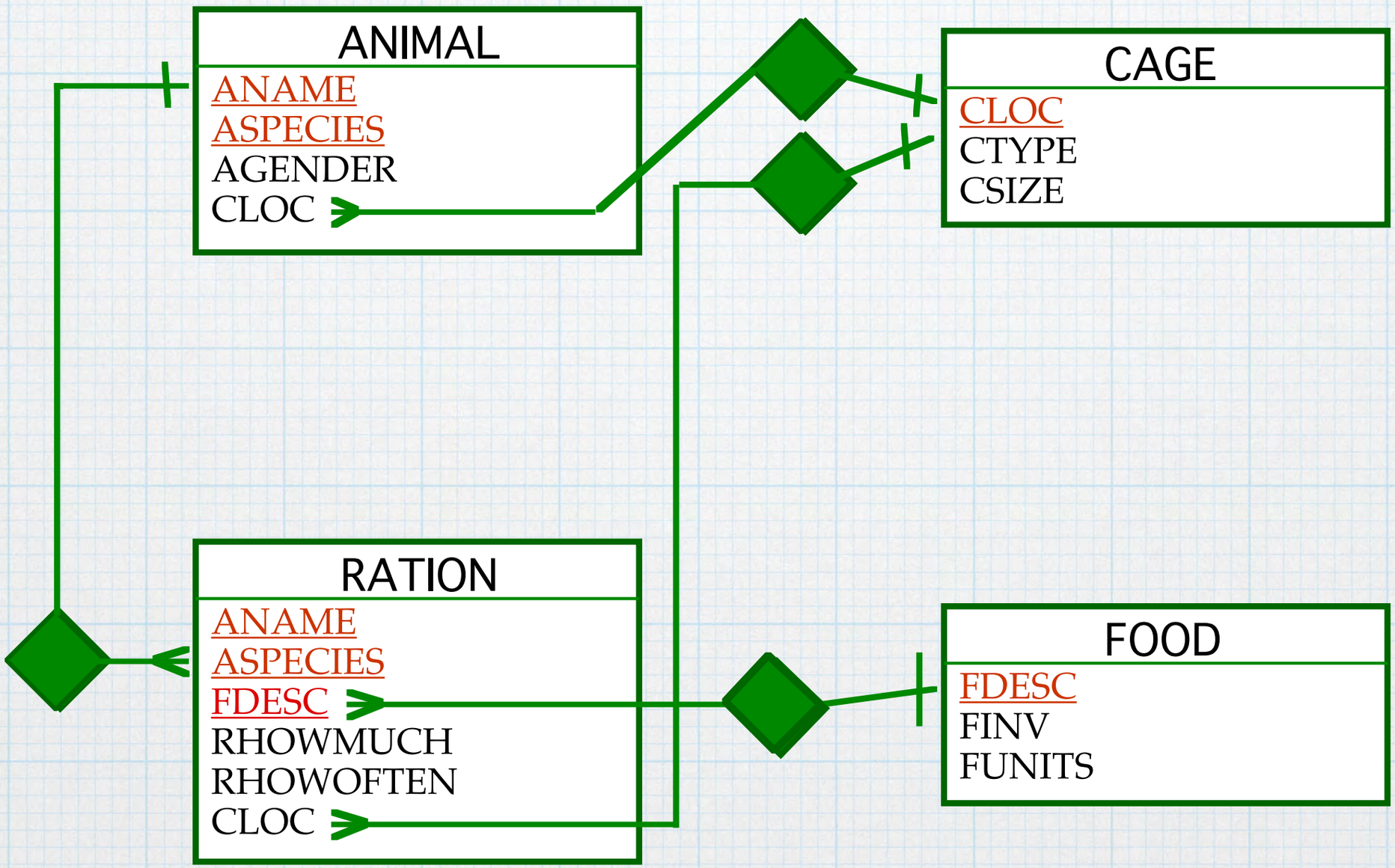
Relating Tables

* Linkage notation
  * note the join columns on the relationship symbol



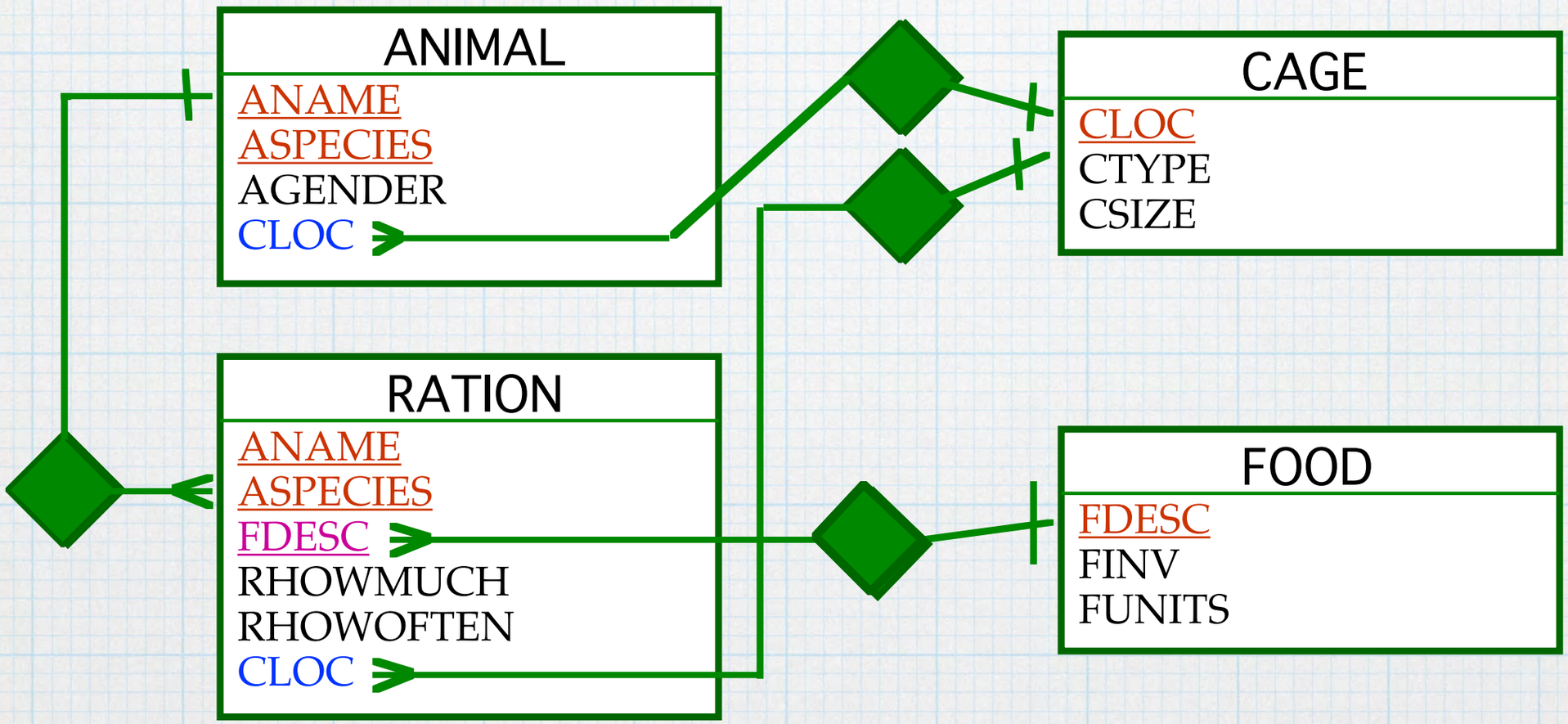
Relatability / Join Compatible

* All connections are based on table contents



Primary / Foreign Keys

* **Primary** keys uniquely distinguish rows in a table
* **Foreign** keys identify rows in another table for the purposes of joining



Data Dictionary

- Capture the full breadth of data definition
  - catalogs tables
    - purpose
    - authority
    - data sources
  - catalogs attributes
    - type
    - valid value ranges
    - formatting
- Defines database
  - Oracle
  - Access
  - Filemaker
  - etc.

Animal =
aname +
aspecies +
agender +
cloc

Food =
fdesc +
finv +
funits

Ration =
aname +
aspecies +
fdesc +
rhowmuch +
rhowoften

Cage =
cloc +
ctype +
csize

Normalization

* Normal forms confirm that the structure of the relational tables protects against various anomalies in retrieval / update
* First normal form
  * all attributes are "atomic"
  * no repeating groups
* Second normal form
  * all non-key attributes are determined by the primary key of that table
  * if the primary key is composite: the whole key
* Third normal form
  * there are no "transitive" relationships in any table
  * dependencies are on the primary key only

CASE Support

* Data base administration
  * many case tools not only allow the analyst to document the structure and contents of the data flows and stores in their model but define the data representation in production database tools
  * maintaining current documentation of database structure and contents is essential to protecting the information assets that a database represents
  * many case tools provide migration tools to move an applications data from one database tool to another with operational obsolescence become a problem
  * visual programming environment require an automated data dictionary to support the "drag and drop" programming of queries and reports