Section 2: ERD reflects current and possible future business needs, Conceptual Models facilitate conversation, describes information needs, prevents mistakes, forms basis for physical design, business rules, regulations and laws

Entity: something of significance, name for a set, noun usually

Instance: Something a part of the entity set (Ex: PERSON(entity) Barack Obama(instance)

Attributes: Describe an entity, Quantify an entity, Qualify an entity, Classify an entity, Specify an entity

Identifier: UID (unique) (primary key)

Implementation Free: stays same regardless of type of database being used for

ERM/ ERD: lists all attributes of entities and their relationships

There are four goals of ER modeling: Capture all required information, Ensure that information appears only once, Model no information that is derivable from other information already modeled, Locate information in a predictable, logical place

Section 3:

Relationships are bi-directional, represent importance to the business, exist only between entities or themselves, named at both ends, have optionality, cardinality (amount)

Mandatory: solid, optional: dashed  
Optionality phrased like must be or may be, cardinality phrased like one and only one / one or more

Entities in soft boxes, always singular with capital letters

Attributes under in smaller boxes,uid/pk marked by # (can be multiple uid attributes), mandatory marked by \*, optional marked by **o**, solid or dashed relationships and end in single toe or crow’s foot

The Components of ERDish

1. EACH

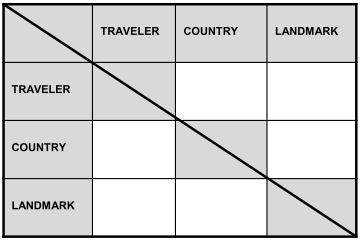
2. Entity A

3. OPTIONALITY (must be/may be)

4. RELATIONSHIP NAME (verbs)

5. CARDINALITY (one and only one/one or more)

6. Entity B

Matrix Diagrams help identify relationships and their names/ verbs, cardinality, and optionality:

Section 4:  
Supertype: “parent” entity i.e. EMPLOYEE  
Subtype: within parent, inherits all parent attributes + can have attributes of their own i.e. ACCOUNTANT, Inherits all attributes of the Supertype, Inherits all relationships of the Supertype, Usually has its own attributes or Relationships, Is drawn within the supertype, Never exists alone, May have subtypes of its own, Is also known as a "subentity“

Exhaustive: Every instance of the supertype is also an

instance of one of the subtypes. All subtypes are listed

without being left out.

Mutually Exclusive: Each instance of a supertype is an

instance of only one possible subtype.

Structural Business rules: types of info stored, how entities relate

Procedural Rules: workflow or process related like approvals must be made by a certain position, etc. (must be programmed/ documented)

Section 5: Non Transferable relationships (once documented, cannot be changed to a different entity i.e receipt to a customer) shown by a diamond on the entity with the mandatory side

One to One: modeled with roles typically, rarer

One to Many: most common  
Many to Many: common in early stages, weeded out in later stages with intersection entities

Redundant: can be derived from other relationships in the model

Resolving many to many: adding an attribute to one of the entities, intersection entities like assignments, records, schedules are typical names

CRUD: create (input), retrieve(view, look up), update (modify, alter), delete (four basic functions a database allows)

Section 6 (IMPORTANT) Simple UID: one attribute describes the UID, Composite UID: 2+ describes the UID. Artificial UID: created for sole purpose of identifying something, not naturally created i.e shoe ID

Barred relationship: UID is given by both entities, or inherited from one or more with the barred UID

Candidate UId: secondary and tertiary UIDs that would be able to be used

Normalization: tests to show if attributes are represented in the right place and only once, missing entities and relationships are identified, and redundancy is eliminated.

0 Normal Form: Every entity has a UID (barred, composite, simple, artificial, etc.)

1st Normal Form: every attribute must be single valued, no repeating attributes, finds missing entities (especially intersect), only one attribute name for the UID

2nd Normal Form: Attribute must be dependent on the entire UID, test with concatenated/ composite UIDs

3rd Normal Form: Attributes can’t have attributes of their own, can’t be dependent

Section 7: Arcs: help define exclusive ORs across relationships, aka entity can be related to one or the other

Hierarchies: one to many with optional “top” and mandatory “bottom”, shared aka barred UIDs common

Recursive Relationships: type of hierarchy where an entity gets a relationship back to itself i.e. manager and employees

Modeling Changes: create entities that hold history i.e. salary history, time worked, etc., audit trails(system that tracks transactions made)

(Section 9) modeling change: schedules(time), assignments, purchase(price), modifications (names for entities)

Section 10: Crows fly south and east, aka crows feet face down and right in the ERD, high volume entities go in lower right, most important entities go in the middle aka central entities, create space, limit crossing relationships

General modeling: looks at things from a distant perspective, simplifies the ERD, generic attributes

Section 11: Primary and Unique Keys, Foreign Key refers to other tables/ entities

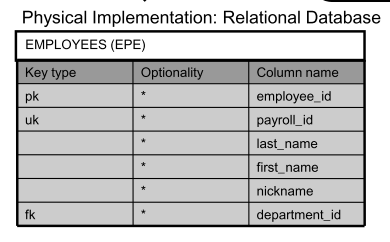
Naming: For single words: take first letter, first letter of second syllable or second letter then last letter

For entity names of more than one word, take the:

• First character of the first word

• First character of the second word

• Last character of the last word

Physical Model: 

• An entity becomes a table.

• An instance becomes a row.

• An attribute becomes a column.

• A primary unique identifier becomes a primary key.

• A secondary unique identifier becomes a unique key.

• A relationship is transformed into a foreign-key column

and a foreign key constraint.

Relationship mapping: subtypes in a similar way, all turned into tables and rows in the database

