How do we design the database for an application?

1. Analyze the problem.
   - Identify the entities, relationships, and attributes
   - Use Entity-Relationship model to capture design
2. Convert the E-R diagram into relational schema.
   - Check the schema for redundancies and anomalies – normalization
   - Input the schema into a DBMS
3. and Tuning:
   - Refine database based on expected usage

**Entity-Relationship Model**

- Proposed by Peter Chen in 1976 as a way to unify the network and relational database views.
- Conceptual data model that views the real world as entities and relationships.

contains:
- An object that exists and is distinguishable from other objects.
  - Example: specific person, company, event, plant
- have
  - Example: people have names and addresses
- An is a set of entities of the same type that share the same properties.
  - Example: set of all persons, companies, trees, holidays
### Entity Sets: customer and loan

<table>
<thead>
<tr>
<th>customer_id</th>
<th>customer name</th>
<th>street</th>
<th>city</th>
<th>loan number</th>
<th>amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>321-12-3123</td>
<td>Jones</td>
<td>Main</td>
<td>Harrison</td>
<td>L-17</td>
<td>1000</td>
</tr>
<tr>
<td>019-28-3746</td>
<td>Smith</td>
<td>North</td>
<td>Rye</td>
<td>L-23</td>
<td>2000</td>
</tr>
<tr>
<td>677-89-9011</td>
<td>Hayes</td>
<td>Main</td>
<td>Harrison</td>
<td>L-15</td>
<td>1500</td>
</tr>
<tr>
<td>555-55-5555</td>
<td>Jackson</td>
<td>Dupont</td>
<td>Woodside</td>
<td>L-14</td>
<td>1500</td>
</tr>
<tr>
<td>244-66-8800</td>
<td>Curry</td>
<td>North</td>
<td>Rye</td>
<td>L-19</td>
<td>500</td>
</tr>
<tr>
<td>963-96-3963</td>
<td>Williams</td>
<td>Nassau</td>
<td>Princeton</td>
<td>L-11</td>
<td>900</td>
</tr>
<tr>
<td>335-57-7991</td>
<td>Adams</td>
<td>Spring</td>
<td>Pittsfield</td>
<td>L-16</td>
<td>1300</td>
</tr>
</tbody>
</table>

### Relationship Sets

- **A relationship** is an association among several entities
  
  Example:
  
  Hayes **depositor** A-102
  
  *customer* entity relationship set *account* entity

- **A relationship set** is a mathematical relation among \( n \geq 2 \) entities, each taken from entity sets
  
  \[ \{(e_1, e_2, \ldots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\} \]

  where \((e_1, e_2, \ldots, e_n)\) is a
  
  Example:
  
  (Hayes, A-102) \(\in\) depositor

### Relationship Set: borrower

- **An attribute** can also be property of a
  
  For instance, the **depositor** relationship set between entity sets *customer* and *account* may have the attribute **access-date**

### Relationship Sets

- **An attribute** can also be property of a
  
  For instance, the **depositor** relationship set between entity sets *customer* and *account* may have the attribute **access-date**
Degree of a Relationship Set

- Refers to the number of entity sets that participate in a relationship set.
- Relationship sets that involve two entity sets are (or degree two). Generally, most relationship sets in a database system are binary.
- Relationship sets may involve more than two entity sets.
  - Example: Suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a ternary relationship set between entity sets employee, job, and branch.
- Relationships between more than two entity sets are rare. Most relationships are binary. (More on this later.)

Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
  - Example:
    
    \[
    \text{customer} = (\text{customer\_id, customer\_name, customer\_street, customer\_city}) \\
    \text{loan} = (\text{loan\_number, amount})
    \]
  - Domain - the set of permitted values for each attribute
  - Attribute types:
    - Simple and composite attributes.
    - Single-valued and multi-valued attributes
    - Example: multivalued attribute: phone_numbers attributes
    - Derived attributes
    - Can be computed from other attributes
    - Example: age, given date_of_birth

Composite Attributes

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one
  - One to many
  - Many to one
  - Many to many

Mapping

- Express the to which another entity can be associated via a relationship set.
- Most useful in describing relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - 
  - 
  - 
  - 
Mapping Cardinalities

- One to one
- Note: Some elements in A and B may not be mapped to any elements in the other set

- One to many

Mapping Cardinalities

- Many to one
- Note: Some elements in A and B may not be mapped to any elements in the other set

- Many to many

Keys

- A super key of an entity set is a set of one or more attributes whose values uniquely determine each entity.

- A candidate key of an entity set is a minimal super key
  - Customer_id is candidate key of customer
  - account_number is candidate key of account

- Although several candidate keys may exist, one of the candidate keys is selected to be the primary key.

Keys for Relationship Sets

- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
  - (customer_id, account_number) is the super key of depositor
  - NOTE: this means a pair of entity sets can have at most one relationship in a particular relationship set.

  Example: if we wish to track all access_dates to each account by each customer, we cannot assume a relationship for each access. We can use a multivalued attribute though.

- Must consider the mapping cardinality of the relationship set when deciding what are the primary keys in case of more than one candidate key.

- Need to consider semantics of relationship set in selecting the primary key in case of more than one candidate key.
E-R Diagrams

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes.
- Double ellipses represent multivalued attributes.
- Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes.

E-R Diagram With Composite, Multivalued, and Derived Attributes

Relationship Sets with Attributes

- The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works_for relationship set.
- Role labels are optional and are used to clarify semantics of the relationship.

Roles

- The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works_for relationship set.
- Role labels are optional and are used to clarify semantics of the relationship.
Cardinality Constraints

- We express cardinality constraints by drawing either a directed line (→), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.

- One-to-one relationship:
  - A customer is associated with at most one loan via borrower
  - A loan is associated with at most one customer via borrower

- In a one-to-many relationship, a loan is associated with at most one customer via borrower, a customer is associated with several (including 0) loans via borrower

- In a many-to-one relationship, a loan is associated with several (including 0) customers via borrower, a customer is associated with at most one loan via borrower

- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower
Participation of an Entity Set in a Relationship Set

- (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
  - E.g. participation of loan in borrower is
    - every loan must have a customer associated to it via borrower
- : some entities may not participate in any relationship in the relationship set
  - Example: participation of customer in borrower is

Alternative Notation for Cardinality Limits

Cardinality limits can also express participation constraints

E-R Diagram with a Ternary Relationship

Cardinality Constraints on Ternary Relationship

- We allow at most one arrow out of a ternary (or greater degree) relationship to indicate a cardinality constraint
- E.g.
  - If there is more than one arrow, of defining the meaning.
    - E.g a ternary relationship $R$ between $A$, $B$ and $C$ with arrows to $B$ and $C$ could mean
      1. each $A$ entity is associated with a unique entity from $B$ and $C$ or
      2. each pair of entities from $(A, B)$ is associated with a unique $C$ entity, and each pair $(A, C)$ is associated with a unique $B$
- Each alternative has been used in different formalisms
- To avoid confusion we
Design Issues

- Choice mainly depends on the structure of the enterprise being modeled, and on the semantics associated with the attribute in question.

Design Issues

- Choice mainly depends on the structure of the enterprise being modeled, and on the semantics associated with the attribute in question.

Possible guideline is to designate a relationship set to describe an action that occurs between entities.

Although it is possible to replace any nonbinary (n-ary, for \( n > 2 \)) relationship set by a number of distinct binary relationship sets, a n-ary relationship set shows more clearly that several entities participate in a single relationship.
Binary Vs. Non-Binary Relationships

- Some relationships that appear to be non-binary may be better represented using binary relationships
  - E.g. A ternary relationship parents, relating a child to his/her father and mother, is best replaced by two binary relationships, father and mother
  - Using two binary relationships allows (e.g. only mother being known)
  - But there are some relationships that are

    - Example: works_on

Converting Non-Binary Relationships to Binary Form

- In general, can be represented using binary relationships by creating an artificial entity set.
  - Replace $R$ between entity sets $A$, $B$ and $C$ by an entity set $E$, and three relationship sets:
    1. $R_A$, relating $E$ and $A$
    2. $R_B$, relating $E$ and $B$
    3. $R_C$, relating $E$ and $C$
  - Create a special identifying attribute for $E$
  - Add any attributes of $R$ to $E$
  - For each relationship $(a_i, b_i, c_i)$ in $R$, create:
    1. a new entity $e_i$ in the entity set $E$
    2. add $(e_i, a_i)$ to $R_A$
    3. add $(e_i, b_i)$ to $R_B$
    4. add $(e_i, c_i)$ to $R_C$

Converting Non-Binary Relationships

- Also need to translate constraints may not be possible
  - There may be instances in the translated schema that cannot correspond to any instance of $R$
  - We can avoid creating an identifying attribute by making $E$ identified by the three relationship sets (described later)
affect ER Design

- Can make access-date an attribute of account, instead of a relationship attribute, if each account can have only one customer.
- That is, the relationship from account to customer is many to one, or equivalently, customer to account is one to many.