Data Modeling using the Entity Relationship Model
Contents

- High-level Conceptual Data Models for Database design
- Example Database Application
- Entity types, Entity Sets, Attributes and keys
- Relationship types, relationship sets, Roles and structural Constraints
- E-R Diagram, Naming Conventions and Design Issues
Data Modeling using the Entity Relationship Model

3

- REQUIREMENTS COLLECTION AND ANALYSIS
  - Functional Requirements
  - Data Requirements
  - CONCEPTUAL DESIGN
    - Conceptual Schema (In a high-level data model)
  - LOGICAL DESIGN (DATA MODEL MAPPING)
    - Logical (Conceptual Schema) (In the data model of a specific DBMS)
  - APPLICATION PROGRAM DESIGN
    - DBMS-independent
    - DBMS-specific
  - PHYSICAL DESIGN
    - Internal Schema
  - TRANSACTION IMPLEMENTATION
    - Application Programs

Minwold
The Company is organized into departments. Each Department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department has several locations.

A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
Example Database Application

- We store each employee’s name, social security number, address, salary, sex, and birthdate. An employee is assigned one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, sex, birth date, and relationship to the employee.
The basic object that the ER-Model represents is an entity, which is a “thing” in real-world with an independent existence.

An Entity may be an object with physical existence or may be an object with a conceptual existence.
Each Entity has attributes— the properties that describe it.

There are several types of attributes
- Simple versus Composite
- Single valued versus Multivalued
- Stored Versus Derived
Composite Versus Simple Attributes

- Composite attributes can be divided into smaller subparts, which represents more basic attributes with independent meaning.
- For example: Address can be subdivided into streetaddress, city, state, country, zip.
- Attributes that are not divisible are called Simple or atomic attributes.
Single Valued versus Multivalued Attributes

- Attributes that hold single value are called single-valued attributes.
  - For example: age

- Multivalued attributes hold more than one value.
  - For example: car with two colors, or phone no of a person
Attributes which are derived from some other attributes are called as derived attributes.

The attributes from which others attributes are derived are called stored attributes.

For ex: Age is derived from BirthDate, so age is derived attribute and Birth Date is stored attribute.
Null Values

- In some cases entity may not have an applicable value for an attribute.
  - For Example: Fax Number or apartment number.
- For such attributes an special attribute called null is created.
- Null Means unknown.
Complex Attributes

- The composition of composite and multivalued attributes is called as complex attributes.
- Composite attributes are represented by () parenthesis and multivalued attributes { }.
A Database usually contains groups of entities that are similar.

For Example, a company hiring hundreds of employees may want to store similar information

These entities share same attributes, but have their own values for each attribute.
An Entity Type defines a collection of entities that have same attributes.

The collection of all the entities of a particular entity type in the database at any point in time is called an entity set.

Entity type is represented as a Rectangular box in ER Diagrams enclosing entity type name.
Attributes names are enclosed in ovals and are attached to their entity type by straight lines.
Multivalued attributes are displayed in double ovals.
Key attributes

- An Entity type has an attribute whose values are distinct for each individual entity in the entity set. Such attributes are called key attributes.
- For example: Roll no in Student table.
- Sometimes several attributes together form a key, meaning that combination of the attribute will identify the entities in an entity set.
- Such an combination of attributes is called as Composite attribute.
Value Sets (Domains) of Attributes

- Each simple attribute in an entity set is associated with an value se or domain, which specifies the set of values that it may hold.
- Value sets are typically specified using the basic data types such as integer, boolean, floating, enumeration, sub range and so on.
- An entity type DEPARTMENT with attributes Name, Number, Locations, Manager, and ManagerStartDate.
- Locations is only multivalued attribute.
- We can specify both name and number as key attributes.
Initial Conceptual Design of Company Database

- An entity type PROJECT with attributes Name, Number, Location, and ControllingDepartment.
- Both Name and Number are key attributes.
Initial Conceptual Design of Company Database

- An Entity EMPLOYEE with attributes Name, SSN, Sex, Address, Salary, BirthDate, Department, and Supervisor.
- Both Name and address may be composite attributes; however this was not specified in requirements.
Initial Conceptual Design of Company Database

- An Entity type DEPENDENT with attributes Employee, DependentName, Sex, BirthDate and Relationship(to the employee)
Degree of a relationship type: The degree of a relationship type is the number of participating entity types.

For Example: WORKSFOR relationship is of degree two.

A relationship type of degree two is called binary.

A relationship type of degree three is called ternary.
It is sometimes convenient to think of a relationship type in terms of attributes.

ManagerStartDate is an attribute which is necessary for both Employee and Department.
Role Names

- The role name signifies the role that a participating entity from the entity type plays in each relationship instance and helps to explain what the relationship means.
- For example: In WORKS_FOR relationship type, EMPLOYEE plays the role of employee and DEPARTMENT plays a role of employer.
Recursive Relationships

- In some cases, the same entity type participates in a relationship type in different roles.
- Such relationships are called Recursive Relationships.
- For such relationships, the role name becomes essential for distinguishing the meaning of each participation.
Constraints of Relationship Types

- **Cardinality Ratios for Binary Relationship**
  - The cardinality ratio for a binary relationship specifies the maximum number of relationship instances that an entity can participate in.
  - For Example: Works _For Relationship is between EMPLOYEE and DEPARTMENT
  - Its Cardinality ratio is 1:N
  - Possible Cardinality ratios for binary relationships are 1:1, 1:N, N:1, and M:N.
Participation Constraint

- Specifies whether the existence of an entity depends on its being related to another entity via the relationship type.
- There are 2 possible participation constraints
  - Total
  - Partial
Weak Entity Types

- Entity types that do not have key attributes of their own are called weak entity types.
- In contrast regular entity types that do have key attributes are called strong entity types.
- Entities belonging to a weak entity type is are identified by strong entity types and the relationship is called identifying relationship.
ER Diagram Naming Conventions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ENTITY</td>
</tr>
<tr>
<td></td>
<td>WEAK ENTITY</td>
</tr>
<tr>
<td></td>
<td>RELATIONSHIP</td>
</tr>
<tr>
<td></td>
<td>IDENTIFYING RELATIONSHIP</td>
</tr>
<tr>
<td></td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>KEY ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>MULTIVALUED ATTRIBUTE</td>
</tr>
</tbody>
</table>
ER Diagram Naming Conventions

- Composite Attribute
- Derived Attribute
- Total Participation of $E_2$ in $R$
- Cardinality Ratio $1:N$ for $E_1:E_2$ in $R$
- Structural Constraint $(\text{min}, \text{max})$ on Participation of $E$ in $R$
Proper Naming of Schema Constructs

- Use Singular Names for Entity Types rather than plurals because entity type applies to each individual entity.
- Entity Type names and relationship types are written in uppercase.
- Attribute names are capitalized
- Role names are in lower case.
One place for all your IT Needs

- Computer Languages
- Website Development
- Project Development
- Software Development
- Corporate Training

Contact:
Sahaj Computer Solutions
3B, Hilltop Residency, Girnar Hills
Opp Gomatesh School, Hindwadi Belgaum-590011
Phone: 2481703, 4200864