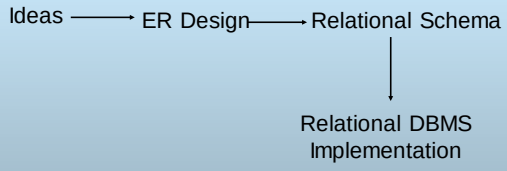
Dept. of Computer Science and Engineering
University of Rajshahi
www.ru.ac.bd

Dr. Shamim Ahmad

Database Modeling and Implementation Process



Two main activities:
Database design
Applications design

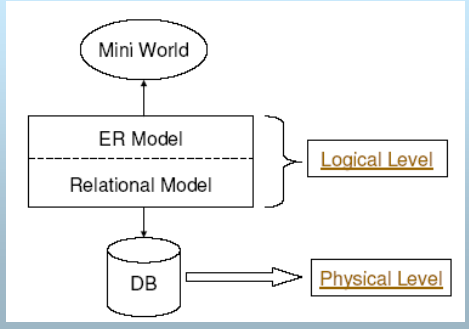
Focus on database design
To design the conceptual schema for a database

Applications design focuses on the programs interfaces that access the database
Generally considered part of software engineering

Entity Relationship Model **(ER Model)** is a popular high-level conceptual data model used for the conceptual design of database applications.

ER model has three main concepts:
Entities
Attributes
Relationships

Overview of Database Design



ER Model Concepts - Entity

- Entity
  - **Entities** are specific objects or things in the mini-world that are represented in the database.
    - *For example the EMPLOYEE John Smith, the Research DEPARTMENT SEEM, the Product Xbox*
  - **Entity set**: define a collection (or set) of entities that have the same attributes.
    - *Each entity type is described by its name and attributes.*

ER Model Concepts - Attribute

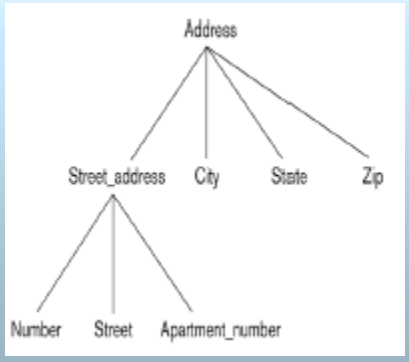
- Attribute
  - **Attributes** are properties used to describe an entity.
    - *For example an EMPLOYEE entity may have the attributes Name, ID, Sex, BirthDate*
  - A specific entity will have a **value** for each of its attributes.
    - *For example a specific employee entity may have Name='John Smith', ID='123456789', Sex='M', BirthDate='09-JAN-55'*
  - Each attribute has a *value set* (or **Domain** or **data type**) associated with it.
    - *For example. integer, string, subrange, enumerated type, …*

Types of Attributes (1)

Simple versus composite

Simple (or atomic): Each entity has a *single atomic value* for the attribute,
HKID or Sex.

Composite: Composition may form a hierarchy where some components are themselves composite.
Address(StreetAddress, City, State, Zip) or
Name((FirstName, MiddleName, LastName).



Types of Attributes (1)

Single-valued versus multivalued

Single-valued: a single value for a particular entity
Age is a single-valued attribute of person.

Multivalued: An entity may have multiple values for that attribute
Colors attribute for a car, or a Previous Degrees attribute for a person. Denoted as {Colors} or {PreviousDegrees}

Types of Attributes (2)

Stored versus derived

- For example, the Age and BirthDate attributes of a person. The value of Age can be determined from the current(today's) date and the value of that person's BirthDate.
- The Age attribute is called a ***derived*** attribute. (or be derivable from the BirthDate attribute.)
- The BirthDate attribute is called a ***stored*** attribute.

Types of Attributes (2)

In general, composite and multi-valued attributes may be **nested** arbitrarily to any number of levels, although this is rare.

- For example, PreviousDegrees of a PERSON is a composite multivalued attribute denoted by {PreviousDegrees (College, Year, Degree, Field)}
- Multiple PreviousDegrees values can exist
- Each has four subcomponent attributes: College, Year, Degree, Field

Types of Attributes (3)

Null Values: in some cases a particular entity may not have an applicable value for an attribute.

The meaning of Null can be classified into

Not applicable: A person with no college degree would have null for College Degree.

Unknown

- **Exists but is missing:** If the Height attribute of a person is listed as null.
- **Not known:** If the HomePhone attribute of a person is null.

9

Key Attributes

Super key: any set of attributes such that the values of the attributes (taken together) uniquely identify one entity in the entity set

- For example, HKID, SID, {NAME, SID}.
- **Candidate key:** Minimal super key -- a super key with no redundant attributes
- For example, HKID, SID.
- **Primary key**: A primary key is one of the candidate keys, designated by the database designer
- For example, SID.
- Every primary key is also a candidate key; every candidate key is also a super key, but not vice versa