2. THE ENTITY-RELATIONSHIP MODEL
(E-R Model)

- The E-R model was developed by Peter Chen in the mid-70's, and extended by him and others in 80's. The E-R model is still broadly used in logical (conceptual) database design.
- The "world" is described in terms of
  - entities
  - relationships
  - attributes
- A schema in E-R model can be visualized by creating an *E-R diagram*.
- Entity-Relationship model can be used as
  - a tool for data modeling and logical database design (specification of an enterprise schema),
  - a formal specification of overall system data structure,
  - a tool for newcomers to learn database concepts and structure,
  - a communication tool between designers,
  - a communication tool between designers and users.

2.1 Basic Concepts

- **Entity instance (e):** a *distinguishable* thing, person, or event ...
- **Entity set (E):** a set of entity instances of the same type.

  Examples:
  - students currently registered at UOG
  - flights offered by Air Canada
  - Sears credit card holders

- **Relationship (r):** represents the fact that certain entities are related to each other in a specified way.
- **Relationship set (R):** a set of relationships of the same type.

  Examples:

  - students *registered* in a course
  - passengers *booked* on a flight
  - books *on loan* to library patrons
  - parent-child relationship

  Mathematical representation of relationship set:

$$R = \{(e_1, e_2, ..., e_n) || e_1 \in E_1, e_2 \in E_2, ..., e_n \in E_n\}$$

$$Teaching = \{(p, s) || p \in Professor, s \in Section\}$$

  Note that a relationship set cannot exist unless all entity instances involved in the relationship exist.

- **Attribute:** describes a property of an entity or a relationship

  Each attribute is labeled with its corresponding name.

  Examples:

  - Student: ID, name, department, ...
  - Book on loan: loan date, due date, ...

  An attribute value is drawn from a certain *domain* or *value set*.

- **E-R diagram:**
  - An E-R diagram is a graphical representation of a schema in E-R model.
  - An E-R diagram utilizes the following notations:

    - * *Rectangles* represent entity sets;
    - * *Ellipses* represent attributes;
    - * *Diamonds* represent relationship sets;
    - * *Lines* link attributes to entity sets and entity sets to relationship sets.
  - No standard notations have been defined.

2.2 Relationship Degree and Cardinality

- **Relationship degree:**
  - Unary relationship: relationship between an entity set and itself
  - Binary relationship: relationship between two entity sets.
  - N-ary relationships: relationship between n entity sets.
  - Note that
    - * Both entity sets and relationship sets may have attributes;
    - * Entities must have keys; and
    - * Relationships can be identified by the (combined) keys of the participating entities.

- **Cardinality:**

  Mapping cardinalities express the number of entity instances to which another entity can be associated via a relationship set.

  - One to one (1:1)

    Each entity instance in A is related to at most one entity instance in B and similarly for each entity instance in B.

  - One-to-many (1:M)

    An entity instance in B is related to at most one entity instance in A, but an entity instance in A can be related to many entity instances in B.

  - Many-to-many (M:M)

    An entity instance in A can be related to many entity instances in entity set B, and vice versa.

  - In an E-R diagram, we can use directed lines (arrows) to specify cardinality. An entity set pointed by an arrow is a "one" entity set in a relationship set.

- Weak entity set and existence dependency:

  - The existence of entity set A depends on the existence of entity set B. A is said to be dependent on B. B is said to be a strong entity set (parent entity) and A is a **weak entity set** (child entity) in the relationship.
  - An entity in a weak entity set cannot exist unless some other entity exists (in the strong entity set)

- More than one relationship set may exist between two entity sets

  Note that relationships must be named to handle such situations.

2.3 Keys

- **Superkey**: a set of one or more attributes which, taken collectively, allow us to identifying an entity instance in an entity set uniquely.

  Example:

  - { student ID, name}
  - { social insurance number, address }
  - { row number, column number }

- **Candidate key:** a smallest possible superkey. i.e. a superkey for which no proper subset is a superkey.

  Example:

  - { student ID }
  - { social insurance number }
  - { row number, column number }

- **Primary key:** a candicate key chosen by database designer as the principal means of identifying entities within an entity set.

  Example:

  - { student ID }

  Primary key selection criteria:

  - not change over time;
  - no null value;

- A weak entity set does not have sufficient attributes to form a primary key (not globally unique), a discriminator is selected for the weak entity set.

  - The **discriminator** of a weak entity set is a set of attributes that allow us to distinguish entities that are all dependent on one particular strong entity set.

    Example (of discriminator):

    { Section number }

  - The **primary key of a weak entity set** can be formed by the primary key of the strong entity set on which it is existence dependent, plus its discriminator.

    Example:

    { Course name, section number }

    or, { Course number, section number }

2.4 Extensions of the E-R model (Special Relationships)

- **Generalization:**

- Generalization forms a new (higher-level) entity set (superset) as the union of two or more (lower-level) entity sets (subsets).
- Attributes that are common to all subsets entities are placed in the superset entities.
- A **superset** is a generic entity set that is subdivided into subsets. A **subset** is a subset of a superset which shares common attributes with other subsets (of the same superset).
- **ISA** relationship is the relationship between each subset and its superset.

- **Specialization (Categorization):**
  - Specialization forms a new entity subset by taking a subset of an entity superset.
  - Specialization and generalization are generated in the opposite directions.
  - **Exhaustive subsets:** All subsets are defined for a superset.
  - **Exclusive subsets:** Subsets have no overlaps. Each instance of the superset is required to be a member of exactly one subset.
  - **Nonexhaustive subset:** Some (not all) of the subsets have been defined for a superset.
  - **Nonexclusive subset:** Subsets may overlap. An instance of the superset may simultaneously belong to more than one subset.

- **Aggregation:**
  - Aggregation models the relationship of "is a part of".
  - It allows modelling of relationships between aggregated parts of a schema.

2.5 Logical data modeling in the E-R model

The general approach for logical modeling in the E-R model:

- Information for logical modeling comes from the problem statement, expert knowledge of the application domain, and general knowledge of the real world.
- Identify entity and relationship sets first.
- Next add attributes to further describe the basic network of entity and relationship sets.

- Then combine and organize entity sets using generalization (specialization) and aggregation.

In-Class Exercise 1

A university is designing a system to store information about courses, course offering, instructors and classrooms for this year.

A course may have prerequisites. A course may be offered once to three times, or not be offered. A course offering may be taught by one or more professors. A professor may teach zero, one or more course offerings. A course (offering) may be taught in one or more classrooms. A classroom may be used for zero, one or more course offerings.

In-Class Exercise 2

A university is designing a system to store information about courses, course offering, instructors and classrooms.

A course may have prerequisites. A course may be offered once to three times, or not be offered. A course (offering) may have a number of lecture sections. A lecture section may be taught by one or more professors. A professor may teach zero, one or more courses. A course may be taught in one or more classrooms. A classroom may be used for zero, one or more courses.

The database should store information about

- courses, including course number and course title,
- offering, including semester, year, and enrolment limit,
- sections, including section number and enrolment,
- professors, including, ID, name, phone number, office,
- classrooms, including room number, and building, and the number of seats.

Reading: Chapter 2.