COSC210
Database Mangement Sytems

Lecture 5 - Data Modelling using the Entity-Relationship Model

Dr. Mitchell Welch / Dr. Edmund Sadgrove

Reading

- Chapter 3 from *Fundamentals of Database Systems* by Elmazri and Navathe

Summary

* Database Design * The Entity-Relationship Model * Attributes * Entities * Relationships * Weak Entities * Higher-Degree Relationships * ER Diagram Notation * Example Transaction ERD

Database Design

- Database design goes through a series of stages:
  - **Requirement Collection** - acquire the data requirements.
  - **Functional Requirements** - user defined operations.
  - **Conceptual design** - developing the conceptual schema (high to mid level).
  - **Logical Design** - implement the database within a DBMS (database schema).
  - **Physical Design** - internal storage structures (e.g. indexing).
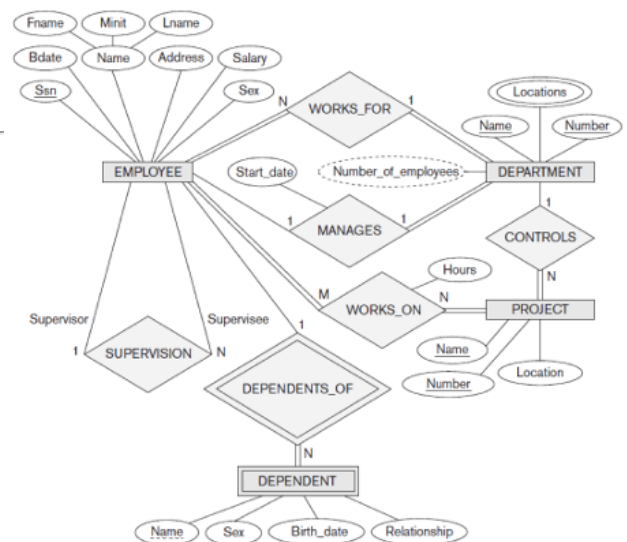
Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

The Entity-Relationship Model

- The **Entity-Relationship(ER) model** is a high-level, **conceptual design** tool.
  - **Entity**:
    - Tangible real-world thing (later tuple in relation or table).
      - E.g. cars, houses, people.
    - **Attributes** - data items that define the entity.
      - E.g. name, DOB, salary.
    - **Values** - listed separately - help describe the attribute.
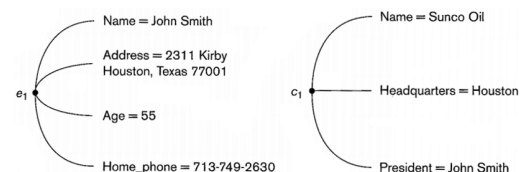
The Entity-Relationship Model - Entities

- With ER models we must define **entity types** with:
 - **Common attributes** - characteristics of a *real-world* entity.

- Defines an **entity set** - set of entities with common attributes.
- Can represent a relation or table.
- Denoted by a shaded rectangle.

The Entity-Relationship Model - Attributes

- Attributes can be:
  - **Simple**:
    - Atomic - indivisible.
    - Denoted by a single-oval.
  - **Composite**
    - Multiple data items.
    - Can form heirarchies (with subparts).
    - E.g. Name in Figure 7.2 (first, middle, last).
    - Denoted by a heirarchy of ovals.
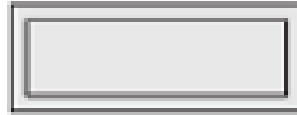
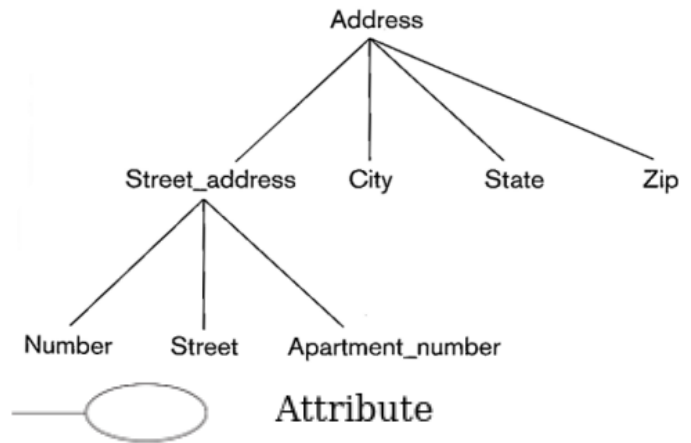# The Entity-Relationship Model - Attributes

- Attributes can be:
  - **Single-valued**:
    - Not part of a set.
    - Unbounded - not restricted by a set.
    - E.g. age, name.
    - ERD: single-oval.
  - **Multivalued**:
    - Contained set of values.
    - Upper bound and lower bound.
    - E.g. Car colour (set of colours).

* Denoted by a double-oval.

The Entity-Relationship Model - Attributes

- Attributes may be:
  - **Derived**:
    - Determined from another attribute.
    - Calculated *on-the-fly* (triggers).
    - E.g. age from birthdate.
    - Denoted by a dashed-oval.
  - **Stored**:
    - Self determined.
    - Inserted *as is*.
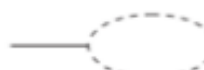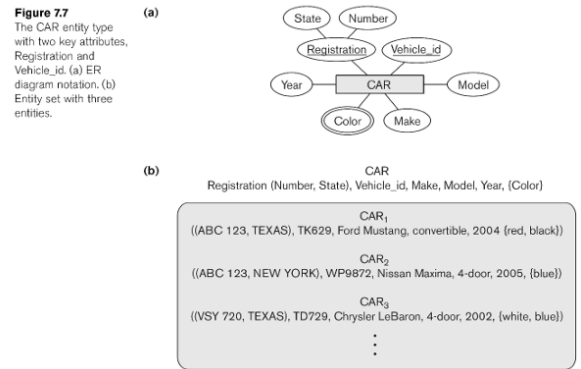   - E.g. birthdate.
    - Denoted by a single-oval.

The Entity-Relationship Model - Entities

- Entities of an **entity type** can have a **uniqueness constraint**:
  - Uniquness across all attributes.
  - **Key attribute** - similar to the relational model.
    - Uniquely identifies entities in an entity set.
    - Denoted by **underlining** the attribute name.
  - **Composite attributes** can be used.
    - Depicts a composite key.
   - Denoted by underlining the parent attribute.

Figure 7.7 The CAR entity type with two key attributes, Registration and Vehicle_id. (a) ER diagram notation. (b) Entity set with three entities.

The Entity-Relationship Model - Attributes

- Like the **relational model** each attribute will have a **domain**.
  - Formally: an attribute A of entity set E has a value set.
    - $A: E \rightarrow P(V)$ - where P is the power set.
    - The power set is all possible subsets of values.
  - Composite attibute value sets are the cartesian product of power sets.
    - $V = P(P(V_1) \times P(V_2) \times \ldots \times P(V_n))$
  - Typically ERD's do not depict value sets (including datatypes).

* A UML diagram is used instead.
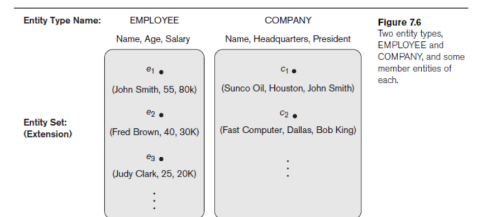
The Entity-Relationship Model - Entities

- Entity types may have multiple key attributes
  - ERD's have no concept of a **primary key**.
  - With multiple key attributes *each is a key in its own right*.
    - Each key is independently unique.
- *Weak* entities have no key attributes.

* More on these later on!

The Entity-Relationship Model - Relationships

- Referring to the diagram below:
- The attributes appear to refer to other entity types.
  - This is a **preliminary** design.
  - These will be converted to relationships in the final ERD.
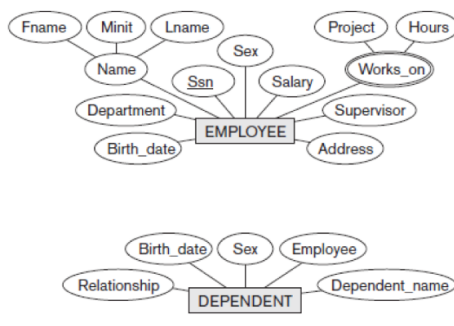
Figure 7.8 Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.
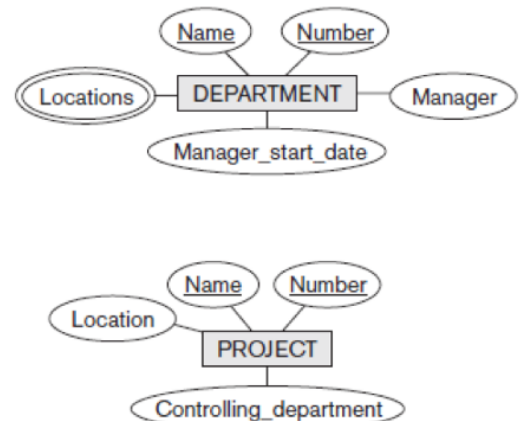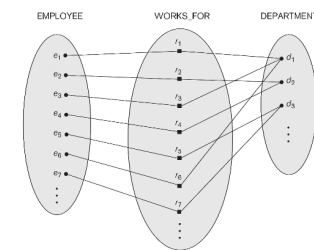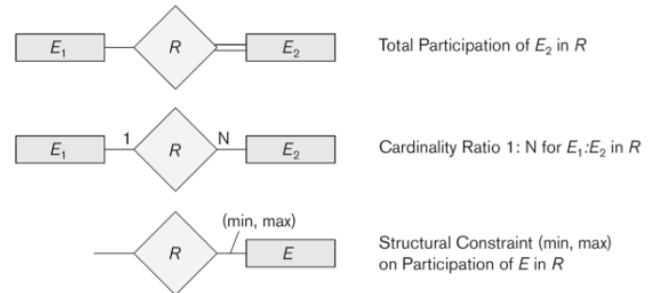
The Entity-Relationship Model - Relationships

- An ERD depicts relationships, this includes:
- A *Relationship Type* R can be defined between n entity types E.
- The is the *Relationship Set*:
    - R among n for $E_1, E_2, ..., E_n$
    - Where each entity is said to participate in the relationship R.
    - Representing an association between these entities.

The Entity-Relationship Model - Relationships

- The **Degree** of the relationship is the number of participating entity types.
    - **Binary**:
        - Common relationships between two entites (degree 2).
    - **Ternary**:
        - Relationships between three entities (degree 3).
- It is possible to represent relationships using attributes.
    - If we represent a **binary relationship**:
        - For example, the **EMPLOYEE entity** with a department attribute
        - Or **DEPARTMENT** with an employee attribute.
   - Both can represent the WORKS_FOR relationship.

Figure 7.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

Total Participation of E_2 in R

Cardinality Ratio 1: N for E_1:E_2 in R

Structural Constraint (min, max) on Participation of E in R

The Entity-Relationship Model - Relationships

- It is possible for relationships to be *recursive*
- For example, the EMPLOYEE entity has a *supervision* relationship with itself.
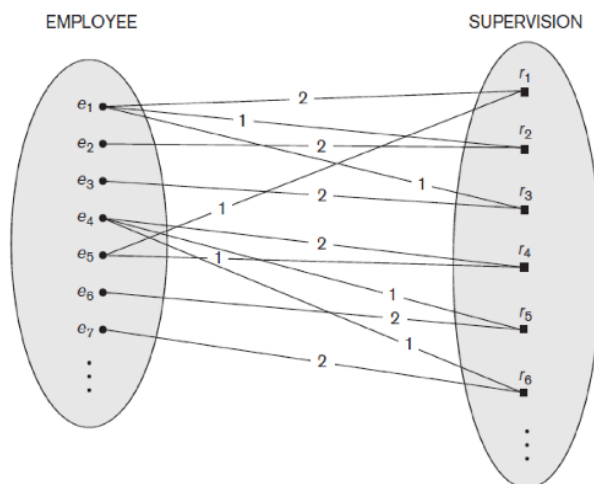
Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).
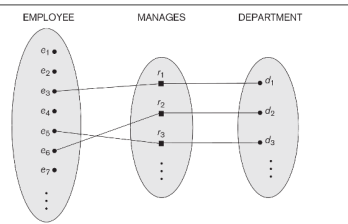
The Entity-Relationship Model - Relationships

- Relationships will have a **Cardinality**:
    - This specifies the number of relationship instances that an entity can participate in.
        - This can be a cardinality of:
        - **1:1, 1:N, N:1 or M:N**.
        - E.g. employee WORKS_FOR department will be 1:N.
- The cardinality of the relationship is displayed on the diamonds near the relationship links.

The Entity-Relationship Model - Relationships

- Entity type relationships can be either:
    - *Partial* participation:
        - Select entities participate.
        - E.g. not all employess are managers.
        - Depicted as a single line.
    - *Totally* participate:
        - Every entity participates.
        - E.g. all employees work for a department.
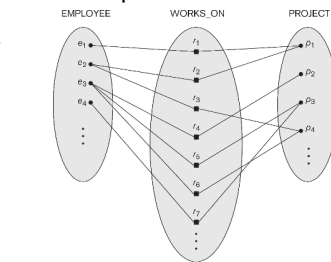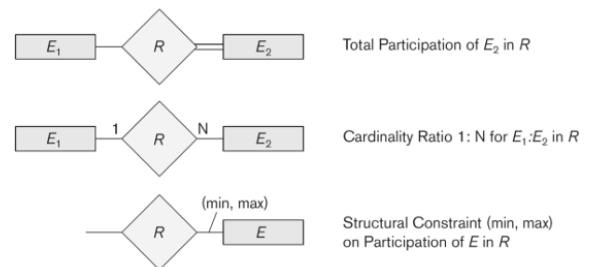
* Depicted as a double line.

The Entity-Relationship Model - Relationships

* Relationship types can have attributes. * In 1:1 and 1:N relationships: * **Attributes** can be **migrated** to one of the participating entities. * In a **1:1** relationship: * **Attributes** can be sent to either side. * In a **1:N** relationship: * Attribute need to be sent to the **'N' side** of the relationship. * In an **M:N** relationship: * Attribute must remain on the relationship. * Their value is determined by the combination of the participating entities.

The Entity-Relationship Model - Weak Entities

- **Entity types** can be **weak entities**:
    - They have no key attributes.
    - These entities cannot exist without another entity-type.
        - E.g. dependent entity relies on employee.
    - Typically have a **Partial Key**:
        - Denoted as a dashed underline.
        - Weak entities are depicted as a double-square.
        - Relationships are depicted with a double-diamond.
        - Always have total-participation (double-line).
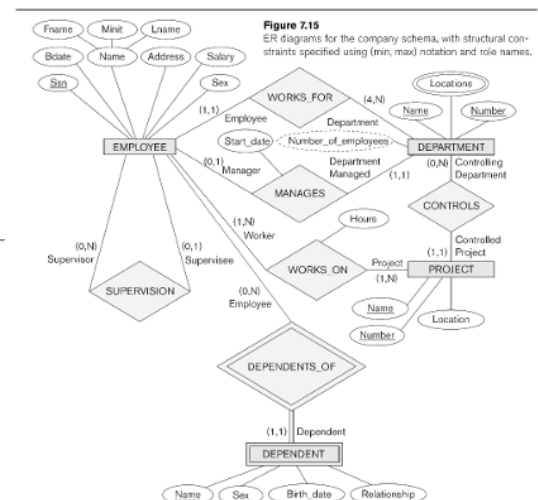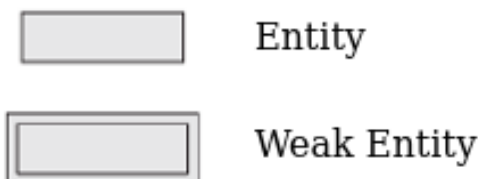- In **SQL** we would declare a foriegn key with `'ON DELETE CASCADE'`.

Higher-Degree Relationships

* We have focussed on **binary** relationships. * It is desirable to have **higher degree** relationships. * Where more than two entities are related. * **Higher degree** relationships have different meanings. * For clarity **higher-degree relationship** are

depicted alongside **binary** relationships.

ER Diagram Notation

- Summary of ER diagram notation.
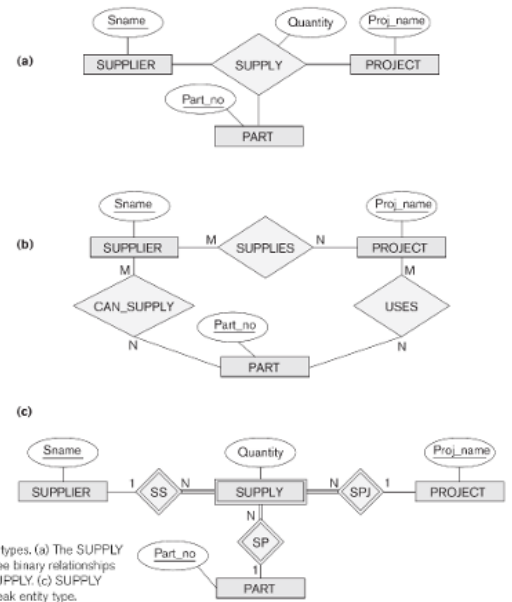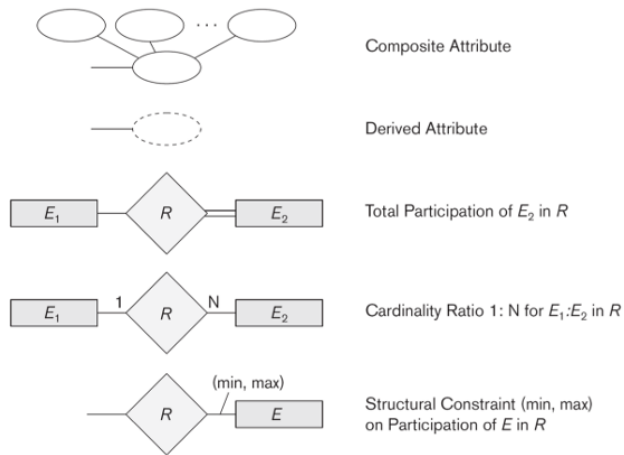
Figure 7.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

Transaction Example

- Example transaction ERD:
  - Entities:
    - Customer.
    - Account.
    - Third Party.
  - Relationships:
    - Transaction between parties.

Figure 7.14
Summary of the notation for ER diagrams.

Summary

- Database Design
- The Entity-Relationship Model
- The Entity-Relationship Model - Attributes
- The Entity-Relationship Model - Entities
- The Entity-Relationship Model - Relationships
- The Entity-Relationship Model - Weak Entities
- Higher-Degree Relationships
- ER Diagram Notation
- Transaction Example.

Questions?

Next Lecture

- Data Modelling using an Enhanced Entity Relationship model.

Reading

- Chapter 7 from *Fundamentals of Database Systems*
- Chapter 4 from *Fundamentals of Database Systems* for this weeks prac session.