**The E/R Model**

**E/R Basics: Entities & Relations**

1. High-Level E/R Model

2. Entities

3. Relations

**Database Design Process**

E/R is a Visual Syntax for a DB Design which is precise enough for technical points, abstracted enough for non-technical people.

Entities and Entity Sets

* Primitive unit of the E/R model

Entities are the *individual Objects*, which are members of entity sets

* **Example**: A specific person or product

Entity Sets are the classes or types of objects in our model

* **Example**: Person or Product

**Entities and Entity Sets**

An Entity Sets has attributes.

**Keys**

* A key is a minimal set of attributes that uniquely identifies an entity
* The E/R model forces us to designate a single *primary key*, we denote it by underling it in our E/R model

**The R in E/R: Relationships**

* A *relationship* is between two entities

**What Is a Relationships?**

* A Mathematical Definition:
  + Let A, B be sets and A x B (cross-product), we define a relationship to be a subset of A x B
* There can only be one relationship for every unique combination of entities, which means that the relationship is uniquely determined by the keys of its entities.
* Relationships may have attributes as well

**2. E/R Design Considerations**

1. Relationships Continued (Multiplicity, Multi-Way)
2. Design Considerations
3. Conversions to SQL
4. **Multiplicity of E/R Relationships**

One-to-One:

Many-to-One:

One-to-One:

Many-to-Many:

**3. From E/R Diagrams to Relational Schema**

**Lecture 2: Design Theory 1**

* Normal Forms and Functional Dependencies
  + Activity: Finding FDs
* Data Anomalies and Constraints
* Functional Dependencies
* FDs for Relational Schema Design

**Design Theory**

* Design Theory is about how to represent your data to avoid anomalies (not standard)

**Normal Forms**