E-Commerce Applications Development

Entity Relationship Diagram

Topics to cover

- Definition
- Basic Components of ERD
- ERD Representations
- Notation Symbols
  - Chen Notation
  - Crow’s foot Notation Symbols
- Type of Entities
- Types of Attributes
- Types of Relationships
  - Implementation of 1:1
  - Implementation of 1:M
  - Implementation of M:M

Entity Relationship Diagram

- To design a database, we need to develop first an Entity relationship diagram that depicts our entire database.
- ERDs depict the database's main components: entities, attributes, and relationships.
- The Relational Database Model (ERM-database containing tables) forms on the basis of an ERD.
- The ERD represents the conceptual database as viewed by the end user.
- ERM uses the ER diagram to represent the conceptual schema (synopsis/outline/diagram)

Entity Relationship Diagram

Why do we need Databases?

- To store physical records or manual records in computers. Database is simply a method through which we store data.

Database should provide two important functions:

- Storage of data
- Accurate and efficient retrieval of data.

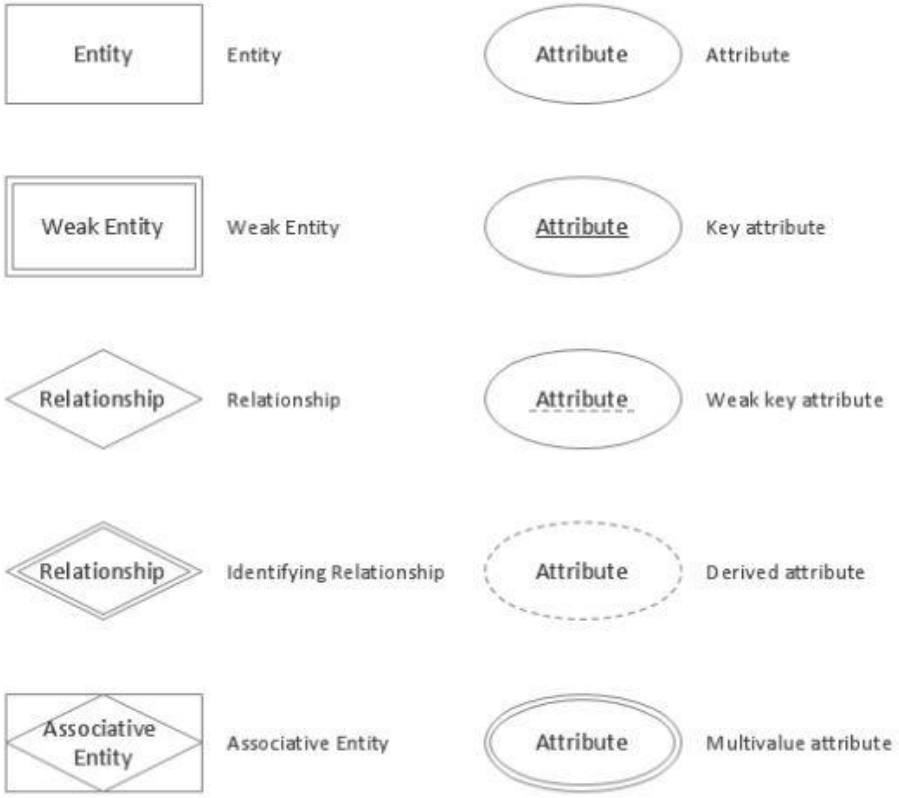
Basic Components of ERD

There are three basic components of ERD, these are:

- Entity
- Attributes
- Relationship

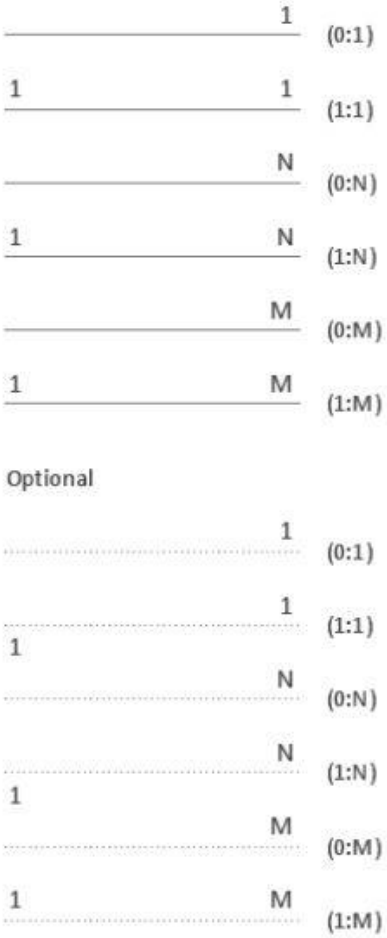
ERD Representations

Chen ERD



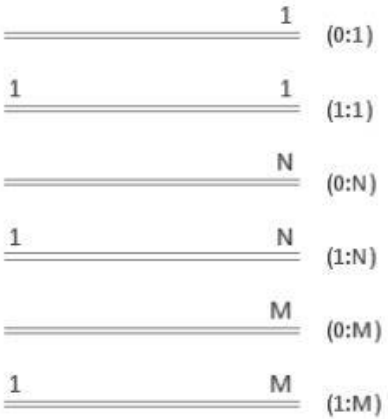
Participations

Cardinality can be shown or hidden



Recursive Relationship

Cardinality can be shown or hidden



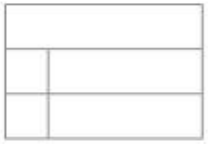
ERD Representations

Crow's Foot ERD

Entity
(with no attributes)

Entity
(with attributes field)

Entity
(attributes field with columns)



Entity
(attributes field with columns and variable number of rows)

Relationships
(Cardinality and Modality)

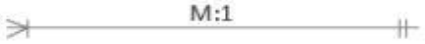
Zero or More

One or More

One and only One

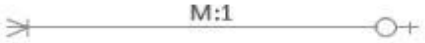
Zero or One

Many - to - One

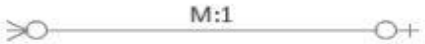


a one through many notation on one side of a relationship and a one and only one on the other

a zero through many notation on one side of a relationship and a one and only one on the other



a one through many notation on one side of a relationship and a zero or one notation on the other



a zero through many notation on one side of a relationship and a zero or one notation on the other

Many-to-Many

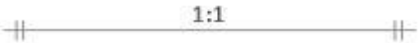
a zero through many on both sides of a relationship

a one through many on both sides of a relationship

a zero through many on one side and a one through many on the other

Many-to-Many

a one and only one notation on one side of a relationship and a zero or one on the other



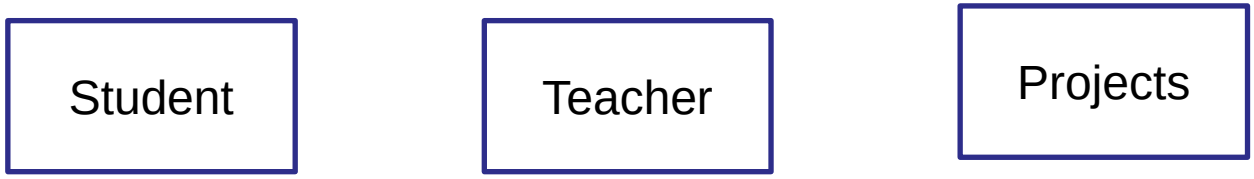
a one and only one notation on both sides

Basic Components of ERD

Entities:

- Entities are represented by means of rectangles.
- Rectangles are named with the entity set they represent.

- For Example: Entities in a school database

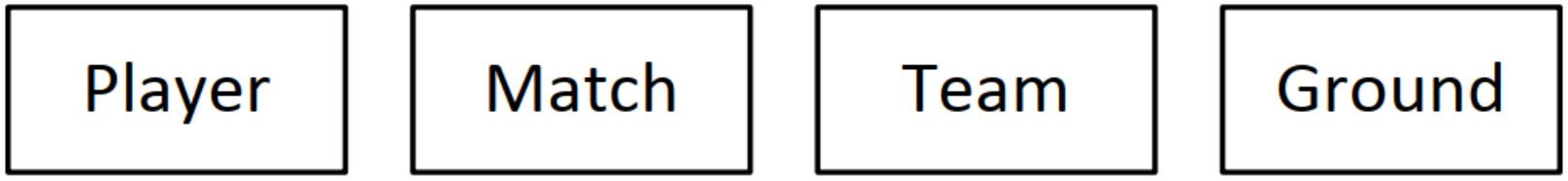


Basic Components of ERD

- **Entity**: A person, place, object, event, or concept in the user environment about which the organization wishes to maintain data.

- **Entity Type (or Entity Set)** – collection of entities
  - Often corresponds to a table.

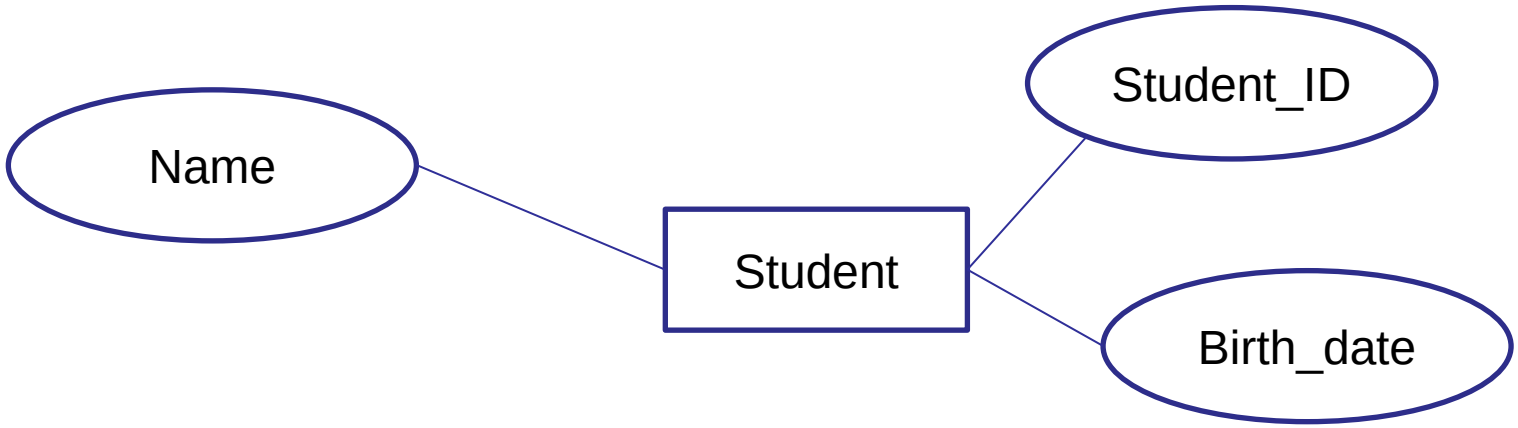
- **Entity instance** – A single occurrence of an entity type.
  - Often corresponds to a row in a table.



Basic Components of ERD

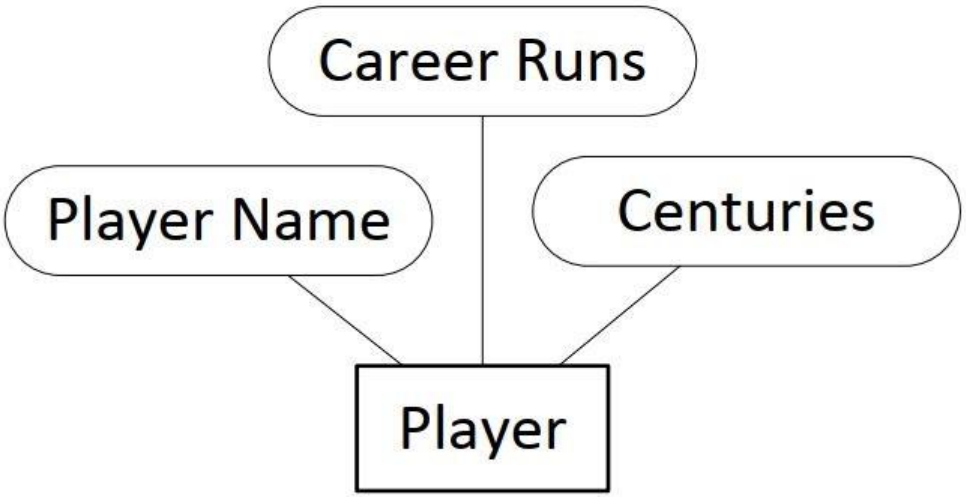
Attributes:

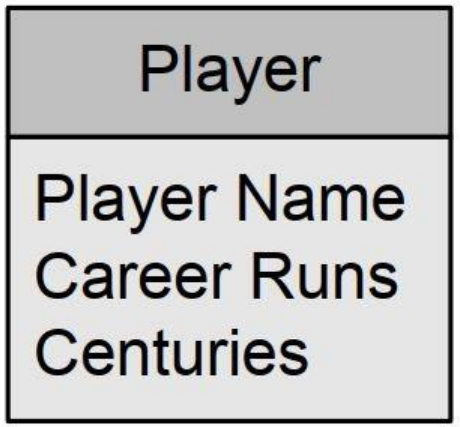
- Attributes are properties of entities.
- Attributes are represented by oval shape (ellipse).
- Every ellipse represents one attribute and is directly connected to its entity (rectangle)



Basic Components of ERD

- Property or characteristic of an entity.
  - An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
- Examples: Player name, Career runs, Number of centuries scored, etc.
- Domain: The set of permitted values for each attribute.
- Rules/Conventions for naming attributes too.



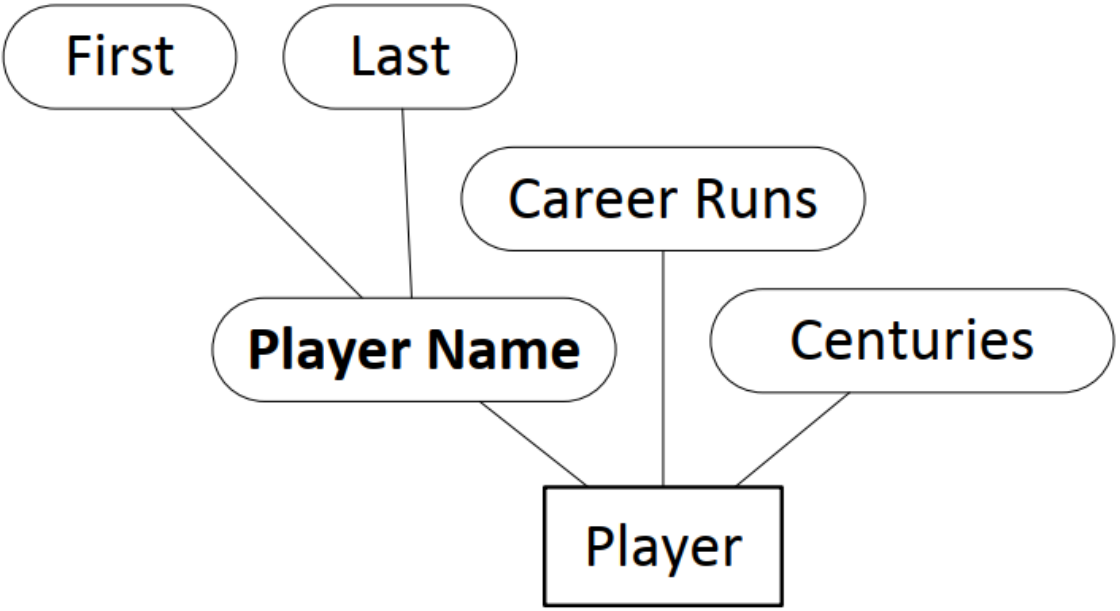


Attribute classes

- Simple vs. Composite Attribute
- Single-Valued vs. Multi-valued Attribute
- Stored vs. Derived Attributes
- Identifier Attributes
- Required vs. Optional Attributes

Composite Attributes

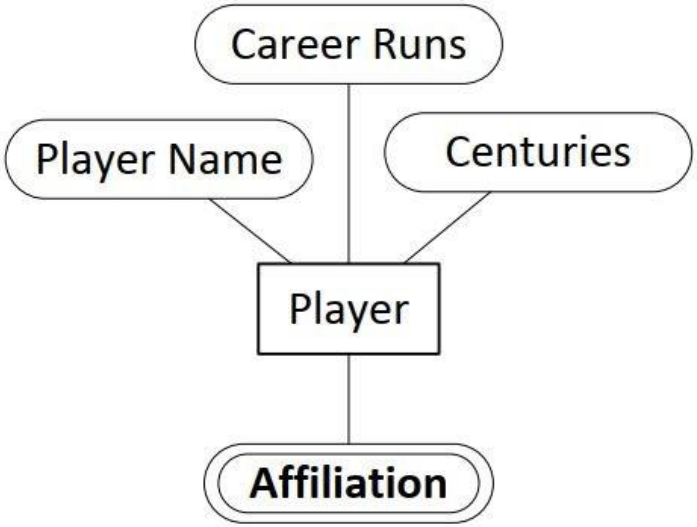
- An attribute broken into many parts: compound data values.



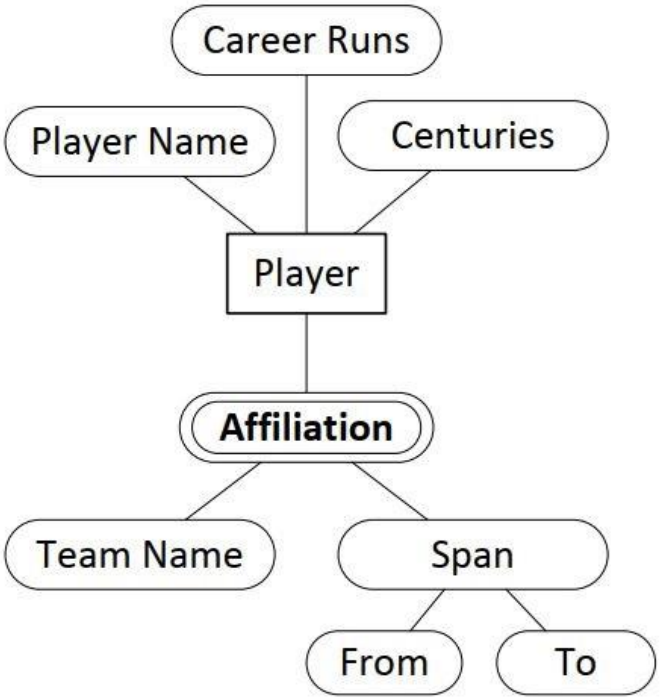
Player
Player Name (First, Last) Career Runs Centuries

Multi-valued attributes

- Multiple data values for one attribute are allowed



Player
Player Name Career Runs Centuries {Affiliation}



Player
Player Name Career Runs Centuries {Affiliation (TeamName, Span (From, To))}

Derived Attributes

- Value can be computed from other attributes
  - Example: Age, given Date of Birth