Database Principles: Fundamentals of Design, Implementation, and Management

Tenth Edition

Chapter 7

Data Modeling with Entity Relationship Diagrams

Objectives

- In this chapter, students will learn:
  - The main characteristics of entity relationship components
  - How relationships between entities are defined, refined, and incorporated into the database design process
  - How ERD components affect database design and implementation

The Entity Relationship Model (ERM)

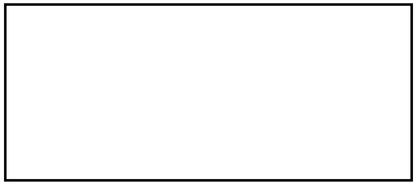
- ER model forms the basis of an ER diagram
- ERD represents conceptual database as viewed by end user
- ERDs depict database's main components:
  - Entities
  - Attributes
  - Relationships

Entities

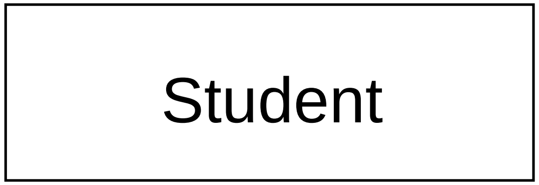
- Refers to entity set and not to single entity occurrence
- Corresponds to table and not to row in relational environment
- In Chen and Crow's Foot models, entity is represented by rectangle with entity's name
- The entity name, a noun, is written in capital letters

Entities

- Represented as a rectangle in E-R diagram



Example:



Attributes

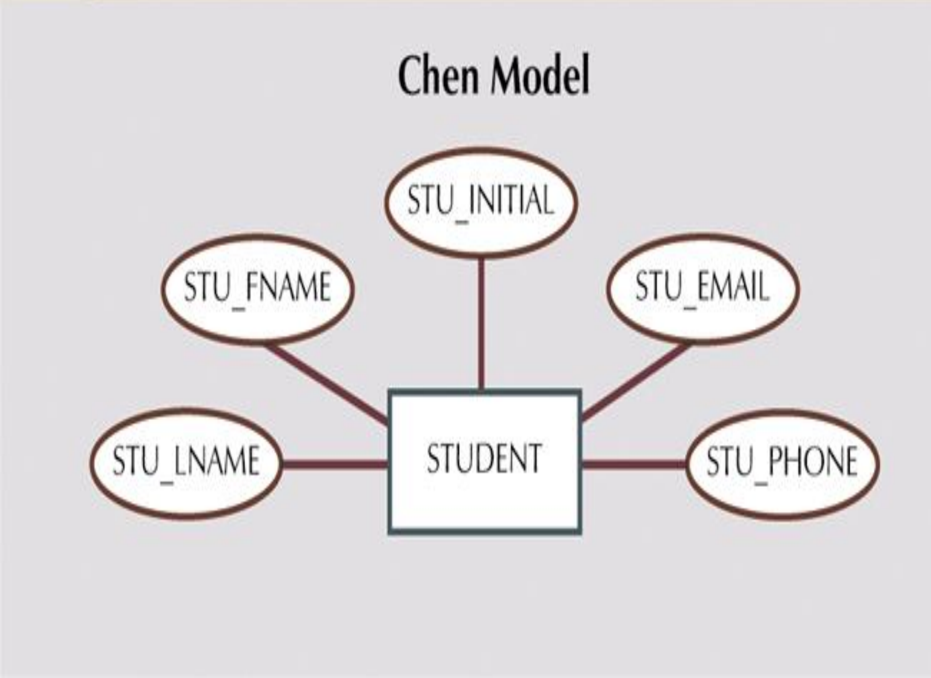
- Characteristics of entities
- Chen notation: attributes represented by ovals connected to entity rectangle with a line
  - Each oval contains the name of attribute it represents

- Crow's Foot notation: attributes written in attribute box below entity rectangle

Attributes

Example

FIGURE 4.1 The attributes of the STUDENT entity



Chen Model

Attribute Types

- **Simple/Atomic** or **Composite** (number of sub-parts):
  - ➢ Simple: one part
  - ➢ Composite: divided into sub-parts

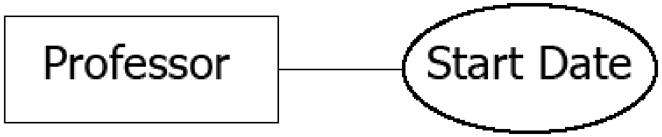
    e.g. Name = surname + first_name
- **Single-** or **multi-valued** (number of values per entity):
  - ➢ Single-valued: one value for a particular entity
  - ➢ Multi-valued: a set of values

    e.g. Student has a number of courses
- **Null attributes**: unknown, not applicable or missing
- **Derived / Calculated**: values of attributes can be derived from other attributes (not stored; computed when required) e.g: age from DOB

Attributes (cont'd.)

- **Composite attribute** can be subdivided
- **Simple attribute** cannot be subdivided

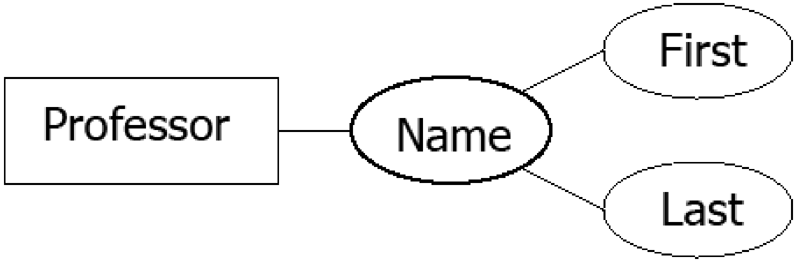
Simple



Professor

Start Date

Composite



Professor

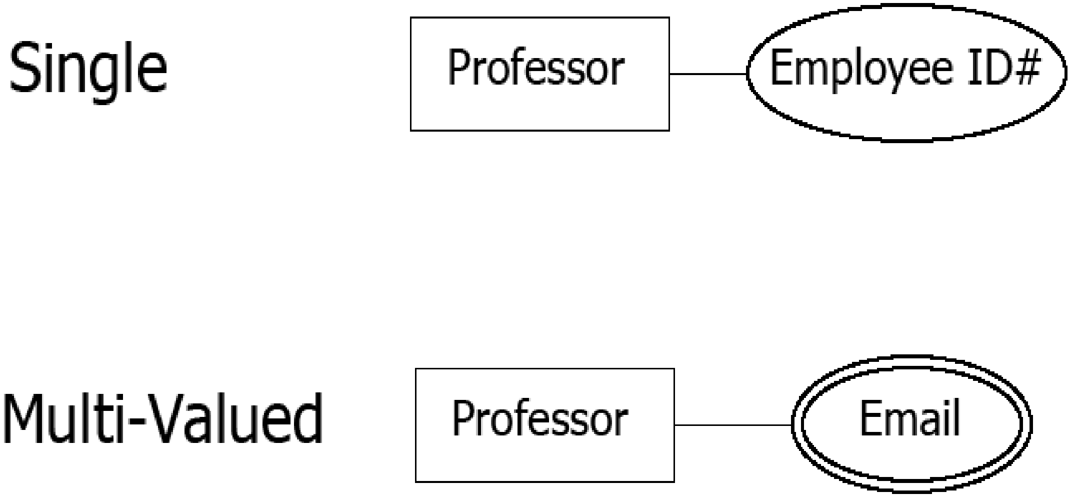
Name

First

Last

Attributes (cont'd.)

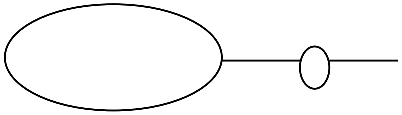
- **Single-value attribute** can have only a single value
- **Multivalued attributes** can have many values



Attribute Types (continued)

- **NULL attributes** have no value
  - – not 0 (zero)
  - – not a blank string
- Attributes can be “nullable” where a null value is allowed, or “not nullable” where they must have a value.

- Represented by:



Attributes (cont'd.)

- **Derived / Calculated Attributes**:
- M:N relationships and multivalued attributes should not be implemented
- Create several new attributes for each of the original multivalued attributes' components
- Create new entity composed of original multivalued attributes' components
- Derived attribute: value may be calculated from other attributes
- Need not be physically stored within database

Attribute Types (continued)

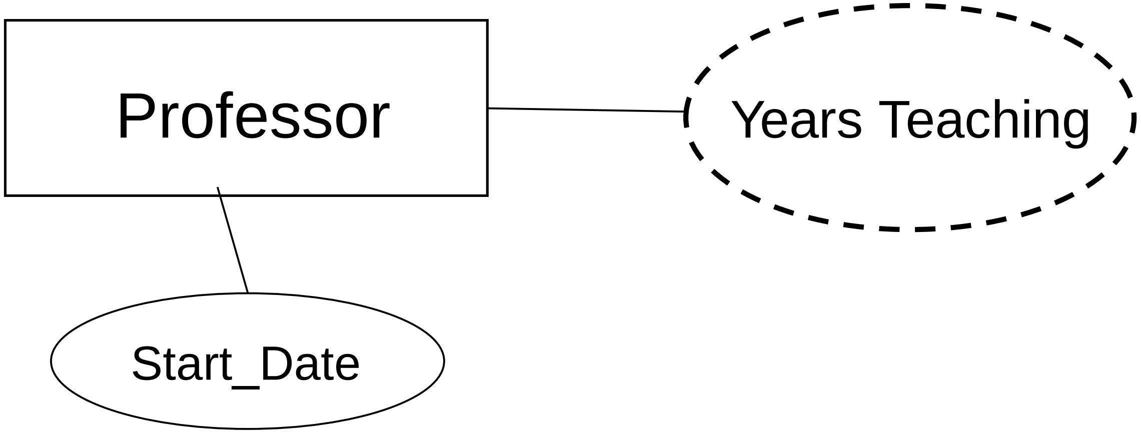
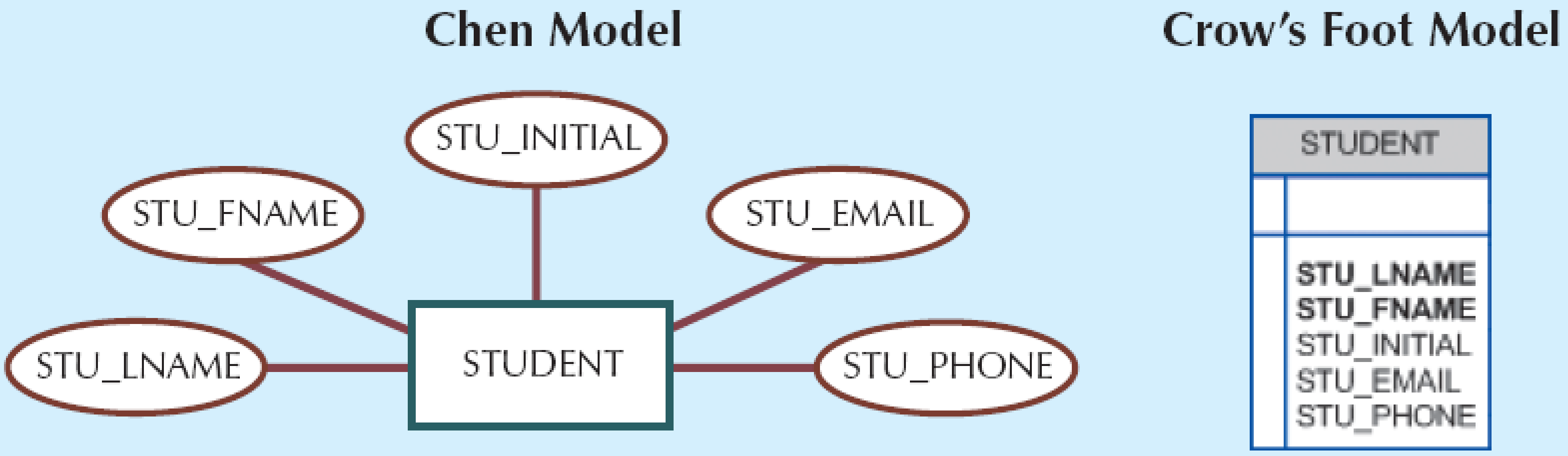


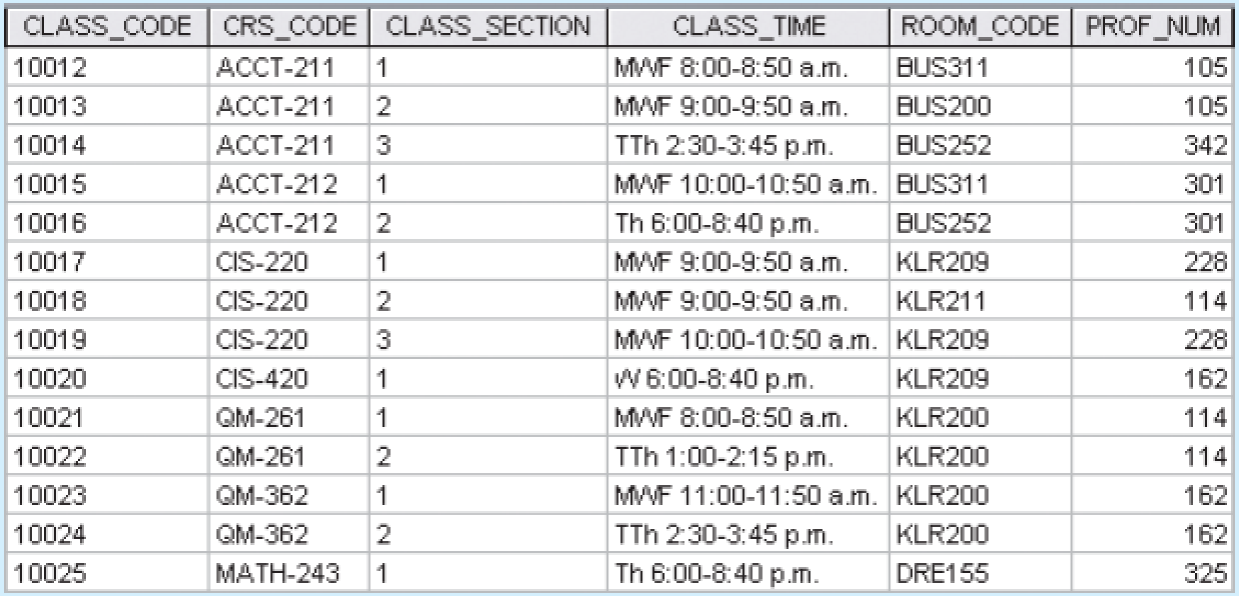
FIGURE 7.1 The attributes of the STUDENT entity: Chen and Crow's Foot



SOURCE: Course Technology/Cengage Learning

FIGURE 7.2 The CLASS table (entity) components and contents

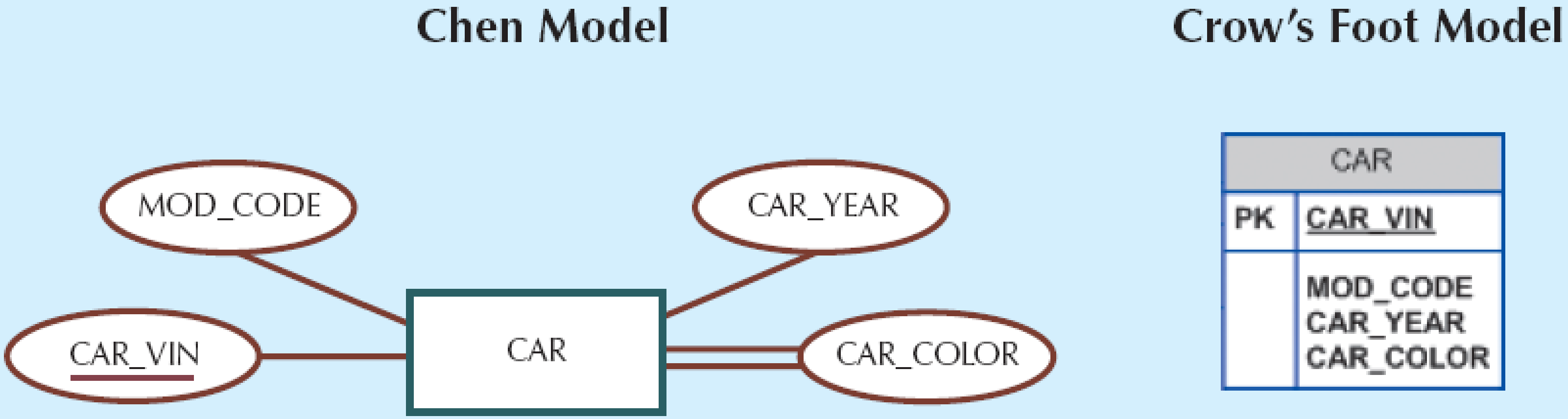
Database name: Ch07_TinyCollege



CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162
10025	MATH-243	1	Th 6:00-8:40 p.m.	DRE155	325

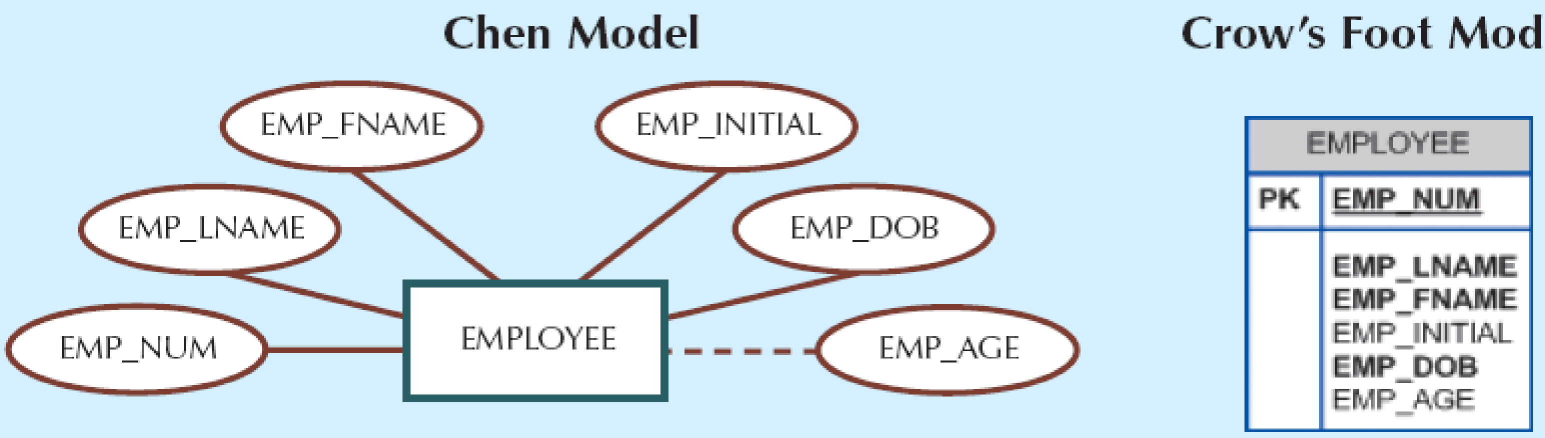
SOURCE: Course Technology/Cengage Learning

FIGURE 7.3 A multivalued attribute in an entity



SOURCE: Course Technology/Cengage Learning

FIGURE 7.6 Depiction of a derived attribute



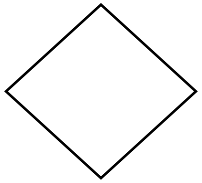
SOURCE: Course Technology/Cengage Learning

TABLE 7.2 Advantages and Disadvantages of Storing Derived Attributes

	DERIVED ATTRIBUTE	
	STORED	NOT STORED
Advantage	Saves CPU processing cycles Saves data access time Data value is readily available Can be used to keep track of historical data	Saves storage space Computation always yields current value
Disadvantage	Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change	Uses CPU processing cycles Increases data access time Adds coding complexity to queries

Relationships

- Association between entities
- Participants are entities that participate in a relationship
- Relationships between entities always operate in both directions
- Relationship can be classified as 1:M
- Relationship classification is difficult to establish if only one side of the relationship is known
- Represented by diamonds in E-R diagram



Connectivity and Cardinality

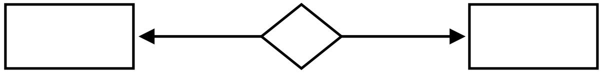
- **Connectivity**
  - Describes the relationship classification
- **Cardinality**
  - Expresses minimum and maximum number of entity occurrences associated with one occurrence of related entity
- Established by very short statements known as business rules

Connectivity and Cardinality

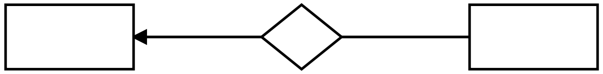
- Connectivity:

  Used to describe the relationship classification

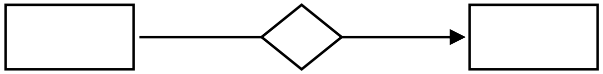
  – one to one

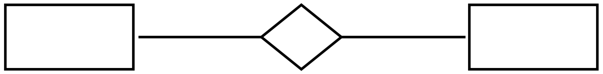
  – one to many

  – many to one

  – many to many

Connectivity and Cardinality

- Cardinality

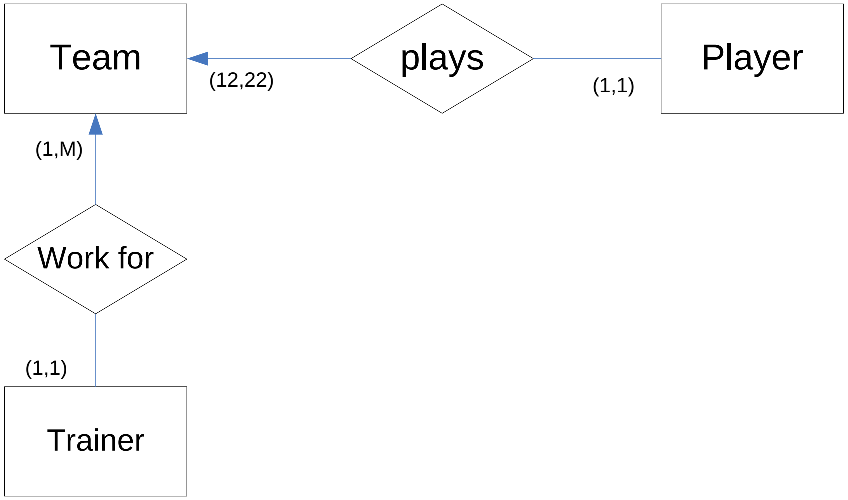
  Expresses minimum and maximum number of entity occurrences associated with one occurrence of related entity

Example:

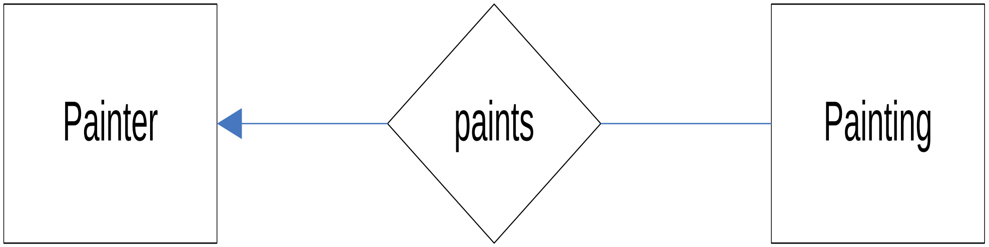
Each team has one or more trainers.

Each team has at least 12 at most 22 players.

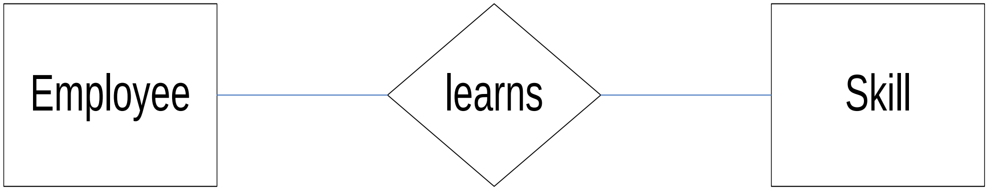
A player may play in only one team and a trainer may work for only one team.



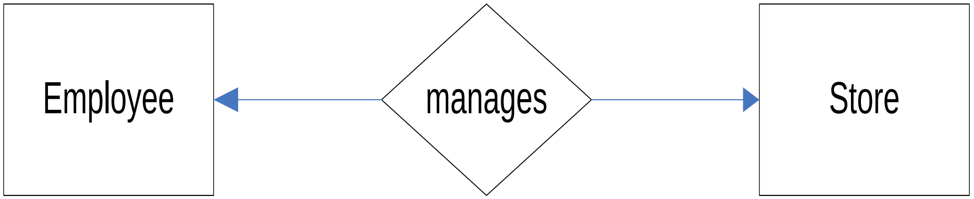
The Basic Chen ERD



A One-to-Many (1:M) Relationship:
A Painter can paint many Paintings;
Each Painting is painted by one Painter



A Many-to-Many (M:N) Relationship:
An Employee can learn many Skills;
each Skill can be learned by many Employees



A One-to-One (1:1) Relationship:
an Employee manages one Store;
each Store is managed by one Employee

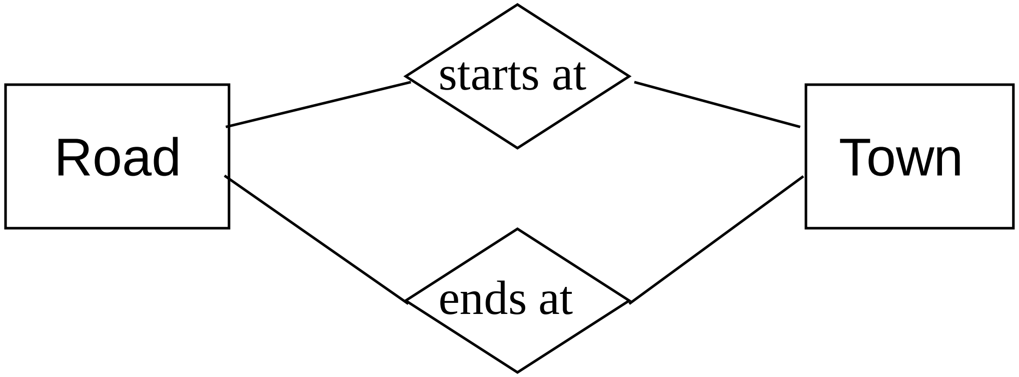
Degree of Relationships

Degree of a relationship is the number of entity types that participate in it

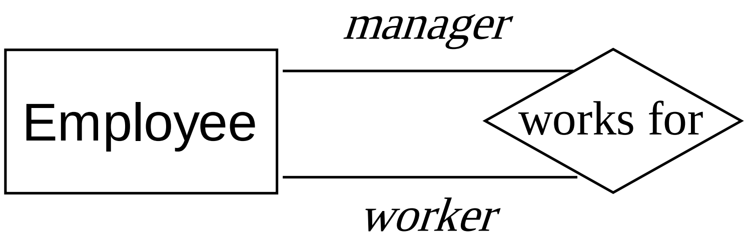
- Unary Relationship
- Binary Relationship
- Ternary Relationship

Examples

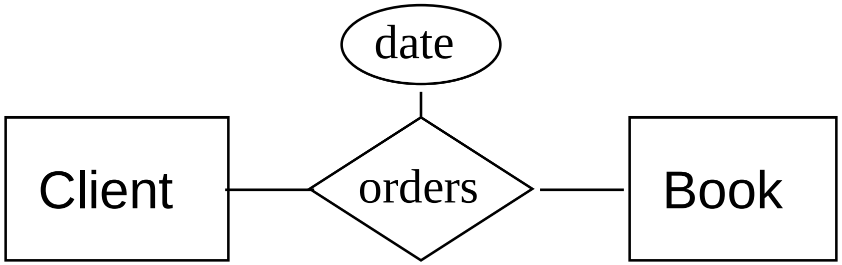
- Can have more than one relationship between entities.



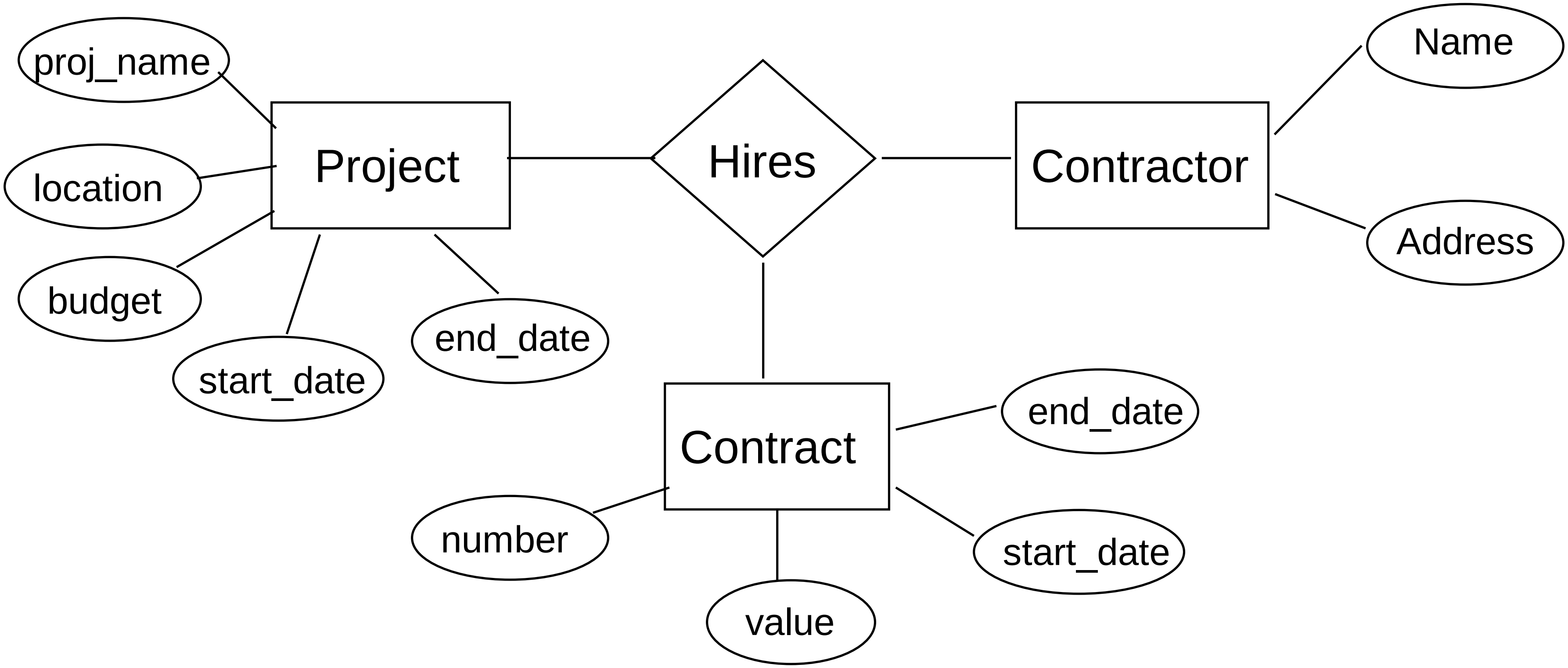
- Can have recursive (UNARY) relationships and can indicate roles for clarity.



- A relationship can also have descriptive attributes.



Ternary Relationship

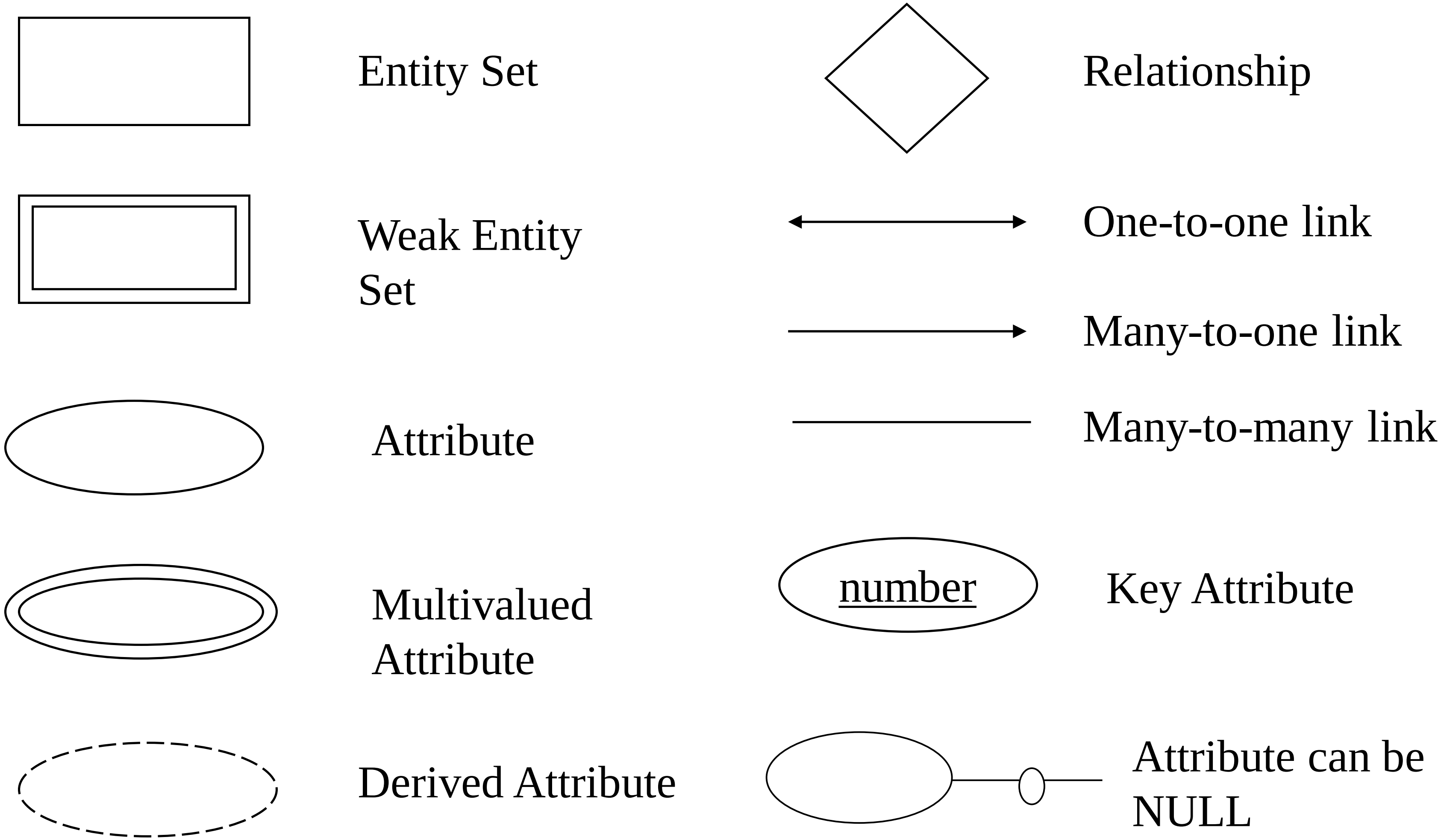


Keys in E/R Diagrams

- **Identifiers:** one or more attributes that uniquely identify each entity instance
- **Composite identifier:** primary key composed of more than one attribute

Identifiers (Primary Key) must be underlined!!!!

E-R Diagram Symbols



Entity Set

Weak Entity Set

Attribute

Multivalued Attribute

Derived Attribute

Relationship

One-to-one link

Many-to-one link

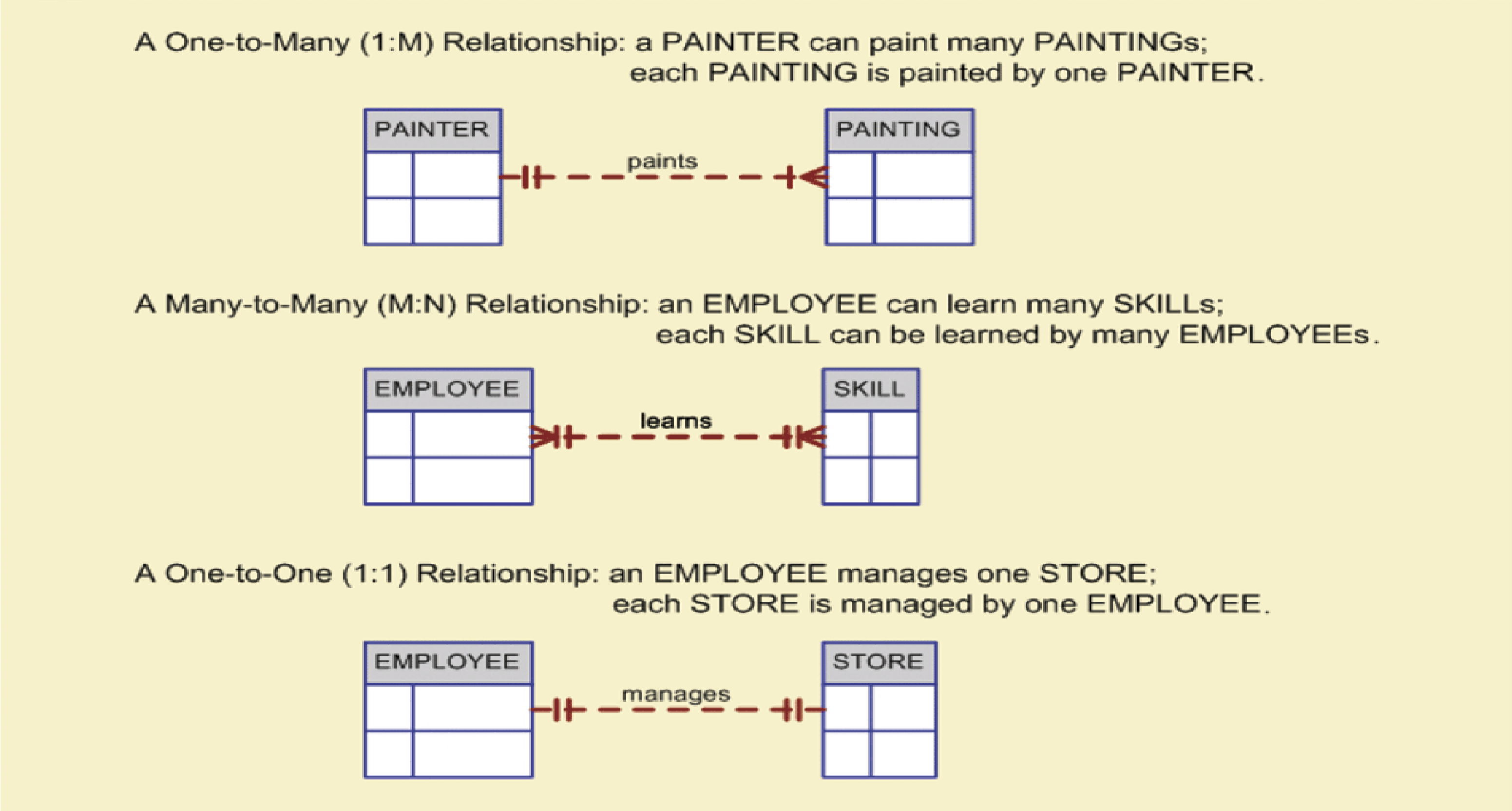
Many-to-many link

Key Attribute

Attribute can be NULL

The Entity Relationship Data Model – Extra Information

FIGURE 2.6 The basic Crow's foot ERD



A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

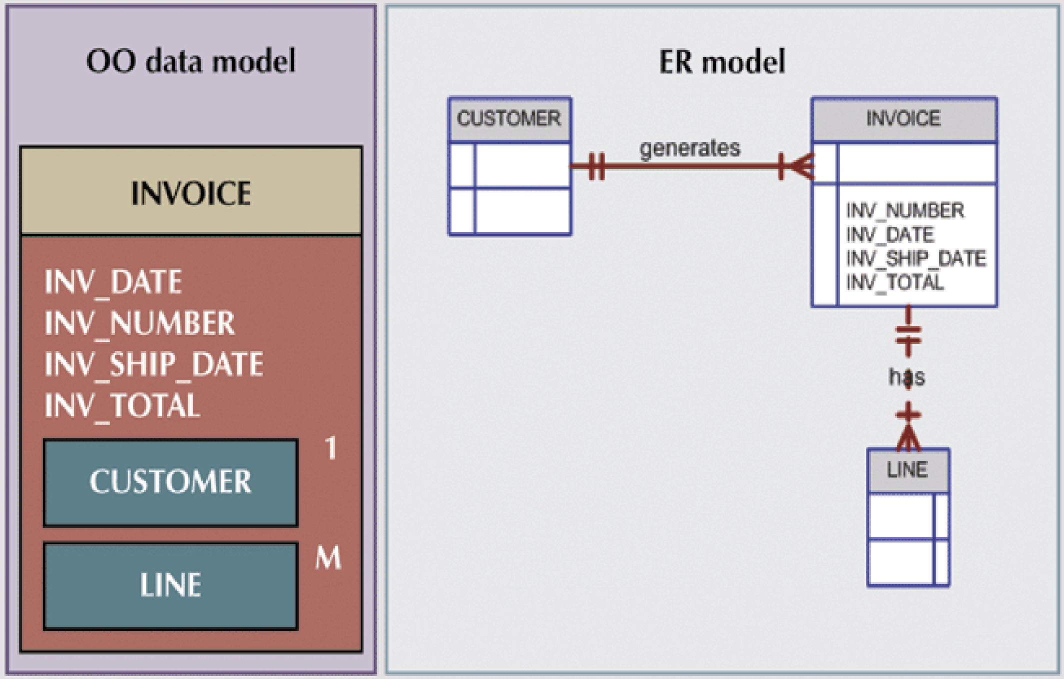
A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

The Object Oriented Model

- Modeled both data and their relationships in a single structure known as an object
- Object-oriented data model (OODM) is the basis for the object-oriented database management system (OODBMS)
- OODM is said to be a semantic data model
- Object is an abstraction of a real-world entity
- Attributes describe the properties of an object
- Objects that share similar characteristics are grouped in classes
- Classes are organized in a class hierarchy
- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of classes above it

The Object Oriented Model (continued)

FIGURE 2.7 A comparison of the OO model and the ER model



Developing an ER Diagram: Practices

Exercise 1:

Draw an ERD for the following specification.

Student: id, name, dob

Department: dno, dname

Course: ccode, title, credits

Rules:

A student is enrolled in one department
In one department, there are many students.

A student takes many courses
A course is taken by many students.

A course is offered by one department
A department offers many courses.

Practices

Exercise 2:

There are many activities. Each activity has a name, a fee, a maximum capacity and number of types (such as tennis, swimming,baseball,etc...). A student may participate in many activities and there may be many students participating in each activity. A student has name (first_name, last_name), a cgpa, and a dob. Each activity is supervised by one and only one coach. But a coach may supervise many activities. Each coach has a name, an identity number and a salary.

Clearly indicate the Primary Keys.

Practices

Exercise 3:

Draw an ERD for the Moon Hotel. There are many rooms in the hotel. Each room has a unique room number, type and price. There may be many customers in each room. But a customer may rent only one room. Each customer has a unique customer_id, name, date of birth and a phone number. There are many departments in the hotel. For each department you need to store deptno, and name. Departments are identified by deptno. There are many employees working for a department, but an employee works for only one department. Each employee has an emp_id, name and salary. One of the employees is the manager of a department. An employee may serve many rooms but only one employee serves each room.

Practices

Exercise 4:

Draw Entity – Relationship Diagram for the following scenario:

There are many branches. Each branch has the following attributes (branchno, address, fax_no, telno). There are many employees. Each employee has the following attributes (name, address, position, salary, gender, empno). Name of an employee is structure that contains two fields (last_name, first_name), address of employee is also a structure that contains (no, street, city). In each branch there are many employees working. Each employee works for a single branch. Each branch is managed by one and only one employee. And each employee manages one and only one branch.

Clearly indicate the Primary Keys.

Practices

Exercise 5:

Draw Entity – Relationship Diagram for the following scenario:

A company has a number of employees. The attributes of employee include employee_id (identifier), name, address, and birthdate. The company also has several projects. The attributes of a project include project_id (identifier), project_name, and start_date. Each employee may be assigned to one or more projects, or may not be assigned to a project. A project must have at least one employee assigned and may have any number of employees assigned. An employee's billing rate varies by project and the company wishes to record the applicable billing_rate (billing_rate) for each employee when assigned to a particular project.

Clearly indicate the Primary Keys.

Homeworks

Homework 1:

Draw Entity – Relationship Diagram for the following scenario:

The system is required to store information about movies for a Movie Hire Business. The users wish to keep track of the following data: Members have a username (no two members are allowed to have the same username), password and address. Every movie has a unique movie id, title, year released and category. A movie only has one director but a director may have directed more than one movie. The unique director id, first and last names as well as address is stored for each director. Address of director is composed of street, city and country. A movie can be hired by many members and a member may hire many movies. You should keep track of the date the movie is hired, as well as the date the movie is due to be returned.

Clearly indicate the Primary Keys.

Homeworks

Homework 2:

Draw Entity – Relationship Diagram for the following scenario:

A department employs up to 30 employees, but an employee is employed by one department. For each employee you need to store unique employee id, name, address and salary. Departments are identified by department id and also have a name. Some employees are not assigned to any department. A division operates many departments, but each department is operated by one division. An employee may be assigned at the most 3 projects, and a project may have at the most 6 employees assigned to it. A project may have at least one employee assigned to it. Each project is identified by unique name and has a budget. A project can be related to other projects. There can be many related projects. One of the employees manages each department, and each department is managed by only one employee. One of the employees runs each division, and each division is run by only one employee. For each division, store unique id and name.

Clearly indicate the Primary Keys.

Homeworks

Homework 3:

Draw Entity – Relationship Diagram for the following scenario:

There are many branches. Each branch has the following attributes (branch_no, address, fax_no, tel_no). There are many employees. Each employee has the following attributes (name, address, position, salary, gender, emp_no). Name of an employee is a structure that contains two fields (last_name, first_name), address of employee is also a structure that contains (no, street, city). The gender field is enumerated (FEMALE, MALE). In each branch there are many employees working. Each employee works for a single department. Each branch is managed by one and only one employee. And each employee manages one and only one department.

Clearly indicate the Primary Keys.

Homeworks

Homework 4:

Draw Entity – Relationship Diagram for the following scenario:

The firm has a number of sales offices in several states. We need to store unique office number and location for each sales office. Each sales office is assigned to one or more employees. You need to store the following information for each employee; unique employee id, employee name, and salary. An employee must be assigned to only one sales office. For each sales office, there is always one employee assigned to manage that office. An employee may manage only the sales office to which he/she is assigned. The firm lists property for sale. Attributes of property include property id (identifier) and Location (composed of Address, City, State, and Zip Code). Each property must be listed with one (and only one) of the sales offices. A sales office may have many number of properties listed, or may have no properties listed. Each unit of property has one or more owners. For each owner we need to store unique owner id and owner name. An owner may own one or more properties, and we need to keep the percentage of properties owned by each owner.

Clearly indicate the Primary Keys.