

Intro To Database

(Database Fundamental using MySQL)

Mohamed ELshafei





ABOUT Course

- **Objective** : this course is designed to introduce the fundamentals of Databases. Trainees will develop skills in the design, construction, modification, and use of databases. Structured Query Language (SQL) will be emphasized.

- **Course Duration :**

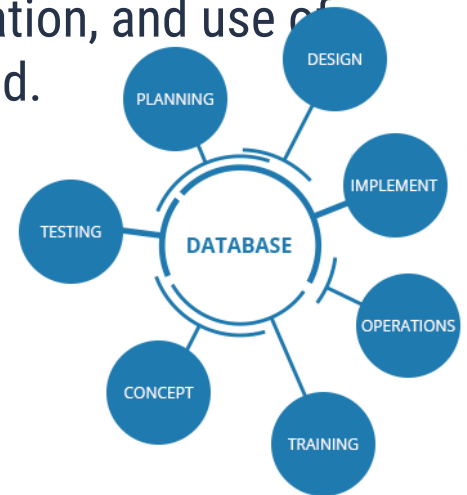
✓ Lectures: 15 hrs.

✓ Labs: 12 hrs.

- **Grading System:**

✓ Lab Work 40%

✓ Final Exam 60%





Agenda

- Data & Information.
- File Based System.
- What is Database, Database System?
- DBMS & its functions.
- Database Properties.
- Advantages and Disadvantages of Database Systems.
- DB Architecture.
- Who Deals with Database.
- Data Models.





HELLO!



Data & Information

- **Data** is the raw input (numbers, characters, images...) which when processed or arranged makes meaningful output (**Information**).
- **Data** is the lowest level of knowledge and **information** is the second level.
- **Data** by itself alone is not significant. **Information** is significant by itself.
- Observations and recordings are done to obtain **data**, while analysis and processing are done to obtain **information**.



File Based System

Early attempt to computerize the manual filing system , each Program defines and manages its own data.

- Separation & Isolation Of data (each user has a copy)
- Data Redundancy (Duplication of data)
- No Database integrity
- All programs maintain metadata for each file they use
- Incompatible File Formats
- Non-standard way to query





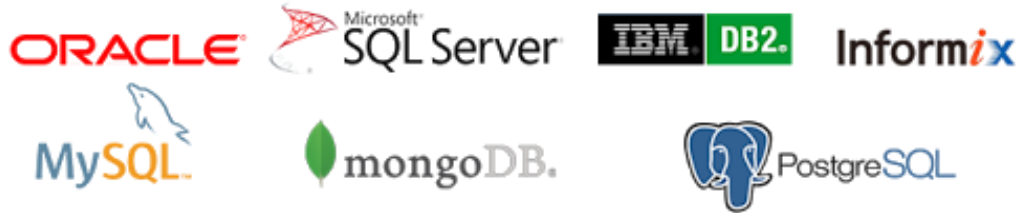
What is a database?

- **A database is an organized collection of related data.”**
- The data is typically organized to model relevant aspects of reality in a way that supports processes requiring this information



Database Management System(DBMS)

- It is the intermediate layer between the database and the programs that access the data.
- It is collection of programs that enables users to create and maintain a database.





DBMS Functions

- **Constructing Database.**
- **Manipulating Database.**
- **Data Security**
- **Data Integrity.**
- **Concurrency.**
- **Backup & Recovery.**
- **Data Dictionary (Meta Data).**
- **Performance.**





Advantages of Database

- Redundancy can be reduced.
- Inconsistency can be avoided.
- Data can be shared.
- Security restrictions can be applied.
- Enforcing Integrity Constraints.
- Providing Backup and Recovery



Disadvantages of Database

- It needs expertise to use.
- DBMS itself is expensive.
- DBMS may be incompatible with any other available DBMS.



Database System

Users



Database System

Application Programs



**DBMS
Software**

Software to process
Queries

Software to access Stored Data

Stored DB
Definition
(Metadata)



Stored
Database

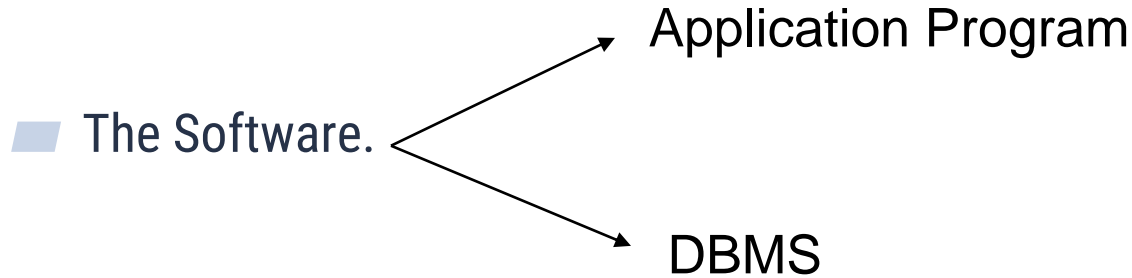




Database System

❑ **A database system is composed of :**

■ The database.

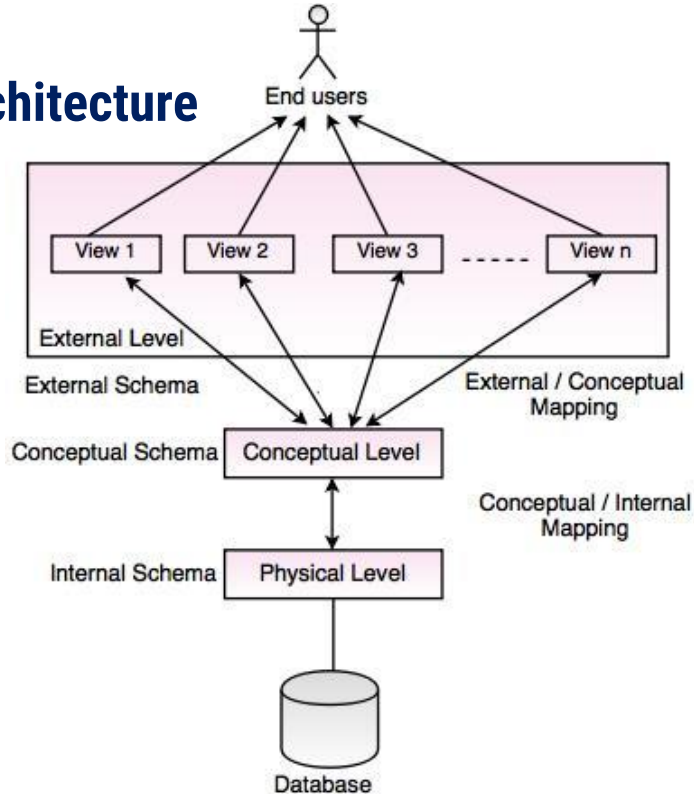




Three Level/Schema Architecture

The three levels of database architecture

1. Physical Level
2. Conceptual Level
3. External Level





Physical Level

- ☐ Physical level describes the physical storage structure of data in database.
- ☐ It is also known as Internal Level.
- ☐ At lowest level, it is stored in the form of bits with the physical addresses on the secondary storage device.
- ☐ At highest level, it can be viewed in the form of files.
- ☐ The internal schema defines the various stored data types. It uses a physical data model.



Conceptual Level

- ☐ Conceptual level describes the structure of the whole database for a group of users.
- ☐ It is also called as the data model.
- ☐ Conceptual schema is a representation of the entire content of the database.
- ☐ These schema contains all the information to build relevant external records.
- ☐ It hides the internal details of physical storage.



External Level

- ☐ External level is related to the data which is viewed by individual end users.
- ☐ This level includes a no. of user views or external schemas.
- ☐ This level is closest to the user.
- ☐ External view describes the segment of the database that is required for a particular user group and hides the rest of the database from that user group.



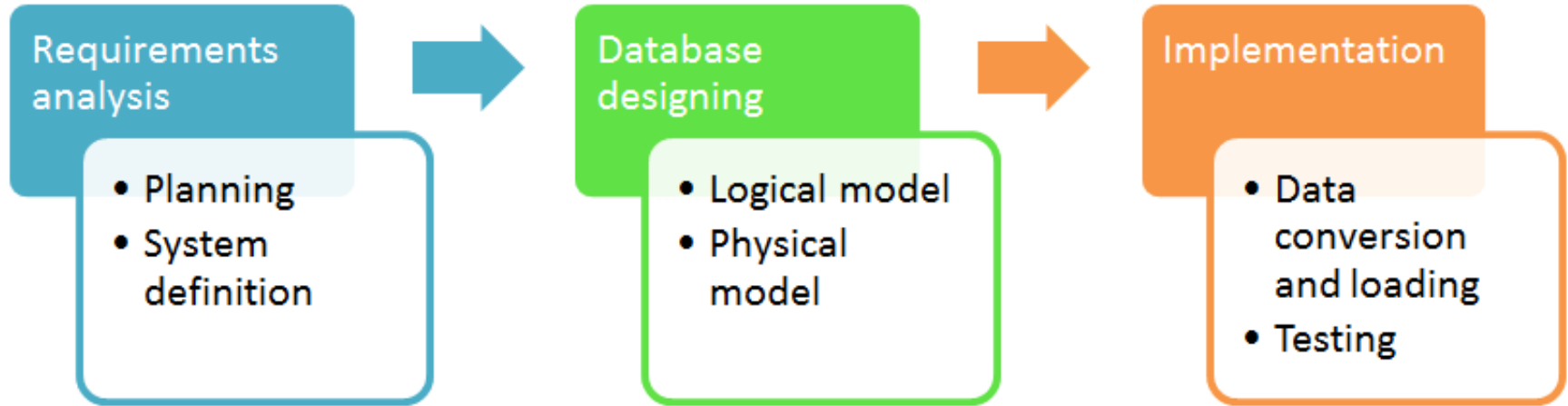
Database Users

- ☐ Database Administrator (DBA)
- ☐ Architect / Database Designer
- ☐ Database Developer
- ☐ Application programmers
- ☐ End users





Database development life cycle





Database Designing

■ **Logical model** - This stage is concerned with developing a database model based on requirements. The entire design is on paper without any physical implementations or specific DBMS considerations.

■ **Physical model** - This stage implements the logical model of the database taking into account the DBMS and physical implementation factors.

Two Types of Database Techniques

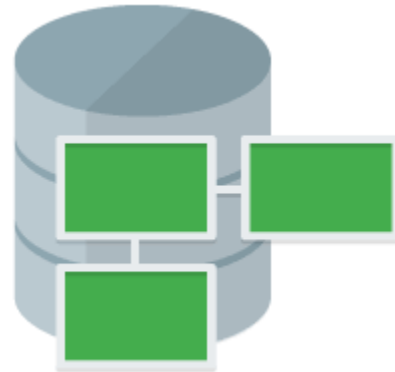
■ Normalization

■ ER Modeling



Data Model

- ❑ **A data model** is a way of finding the tools for both business and IT professionals, which uses a set of symbols and text to precisely explain a subset of real information to improve communication within the organization and thereby lead to a more flexible and stable application environment





Data Model

- A data model is an idea which describes how the data can be represented and accessed from software system after its complete implementation.
- It is a simple abstraction of complex real world data gathering environment.
- It defines data elements and relationships among various data elements for a specified system.
- The main purpose of data model is to give an idea that how final system or software will look like after development is completed.



Types of Data Model

- ☐ Hierarchical Model
- ☐ Relational Model
- ☐ Network Database Model
- ☐ Entity Relationship Model
- ☐ Object Model



Hierarchical Model

- Hierarchical model was developed by IBM and North American Rockwell known as Information Management System.
- It represents the data in a hierarchical tree structure.
- This model is the first DBMS model.
- In this model, the data is sorted hierarchically.
- It uses pointer to navigate between the stored data.

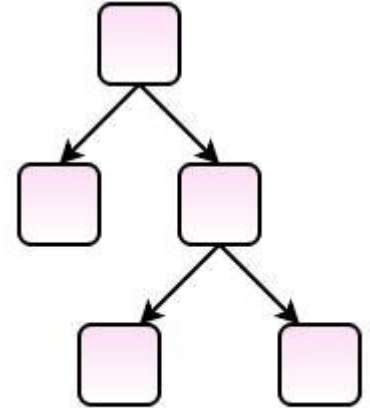
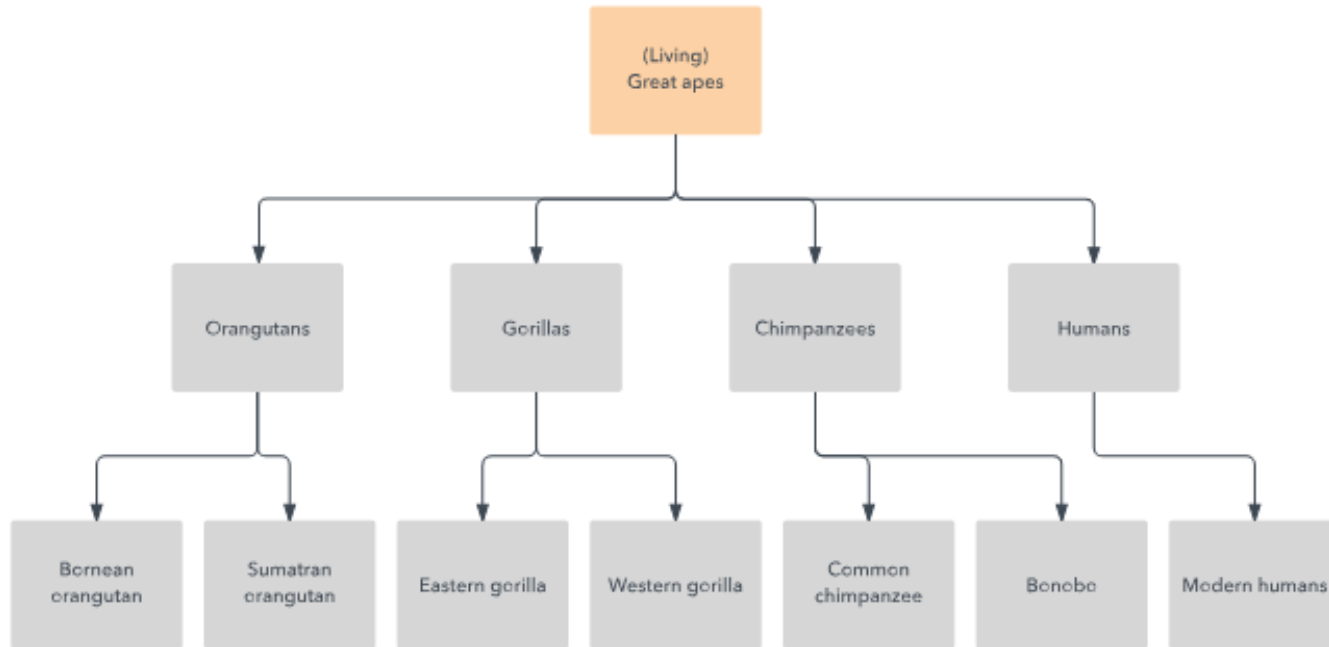


Fig. Hierarchical Model



Hierarchical Model





Relational Model

- Relational model is based on first-order predicate logic.
- This model was first proposed by E. F. Codd.
- It represents data as relations or tables.
- Relational database simplifies the database structure by making use of tables and columns.

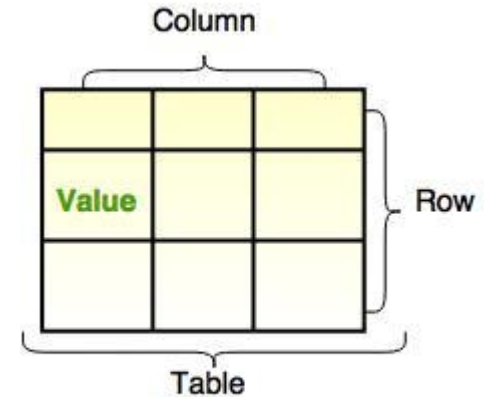
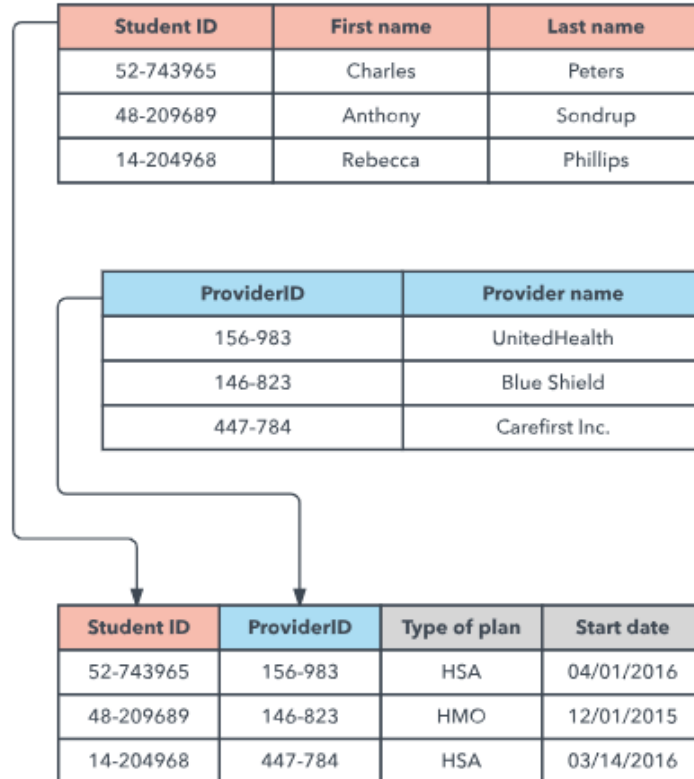


Fig. Relational Model



Relational Model





Network Database Model

- Network Database Model is same like Hierarchical Model, but the only difference is that it allows a record to have more than one parent.
- In this model, there is no need of parent to child association like the hierarchical model.
- It replaces the hierarchical tree with a graph.
- It represents the data as record types and one-to-many relationship.
- This model is easy to design and understand.

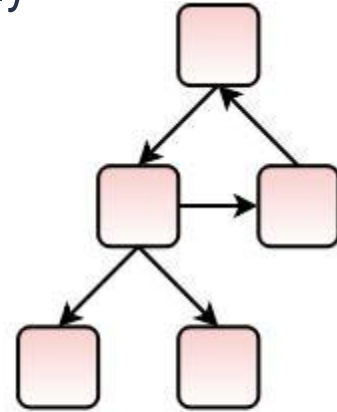
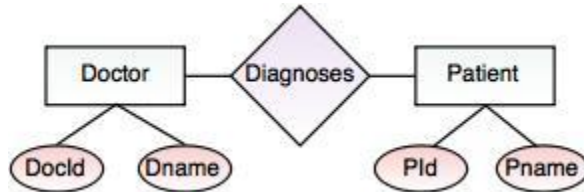


Fig. Network Model



Entity Relationship Model

- Entity Relationship Model is a high-level data model.
- It was developed by Chen in 1976.
- This model is useful in developing a conceptual design for the database.
- It is very simple and easy to design logical view of data.
- The developer can easily understand the system by looking at an ER model constructed.





Object Model

- Object model stores the data in the form of objects, classes and inheritance.
- This model handles more complex applications, such as Geographic Information System (GIS), scientific experiments, engineering design and manufacturing.
- It is used in File Management System.
- It represents real world objects, attributes and behaviors.
- It provides a clear modular structure.
- It is easy to maintain and modify the existing code.



Object Model

Object 1: Sales report

Month	
Product code	
Vendor	
Revenue	

**Object 1
instance**

01-15-16
54
154-234
\$887

Object 2: Sales activity

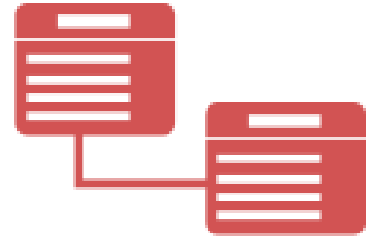
Customer	
Product code	
Product name	
Sales associate	
Date of sale	
Price	



ER Model

Entity Relationship Diagram

Concepts





ER Model

■ **ER Model** is a high-level data model, developed by Chen in 1976. This model defines the data elements and relationships for a specified system. It is useful in developing a conceptual design for the database & is very simple and easy to design logical view of data.



Importance of ER Model

- ERD stands for Entity Relationship diagram.
- ER diagram shows the relationship between objects, places, people, events etc. within that system.
- ER Model is a graphical representation for designing the structure.
- Without ER diagrams you cannot make a database structure & write production code.
- It displays the clear picture of the database structure.



Basic constructs of the E-R model

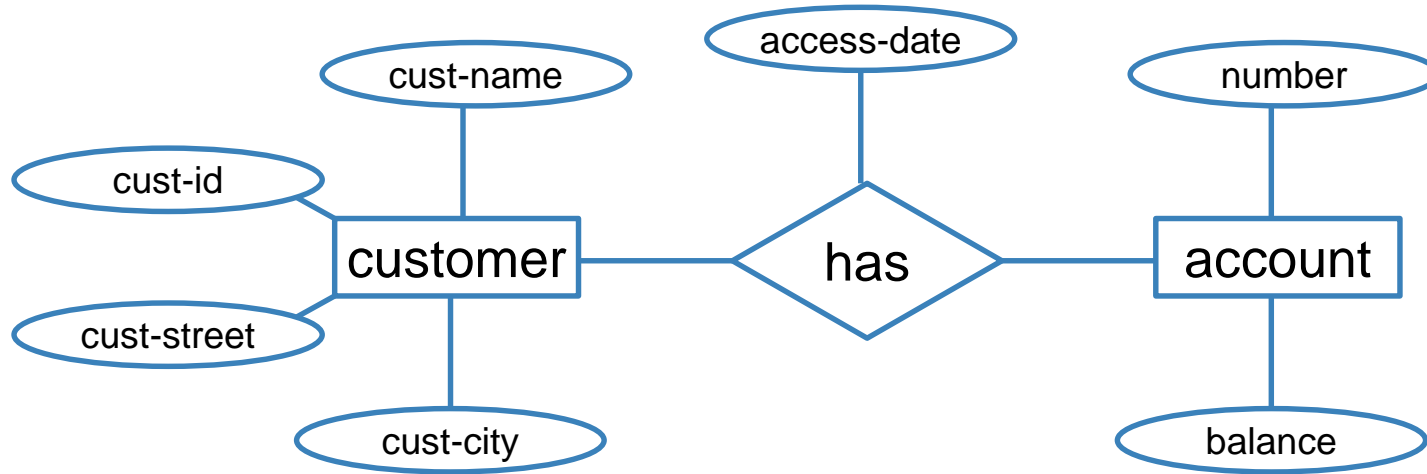
■ **Entities** - person, place, object, event, concept (often corresponds to a real time object that is distinguishable from any other object)

■ **Attributes** - property or characteristic of an entity type (often corresponds to a field in a table)

■ **Relationships** – link between entities (corresponds to primary key-foreign key equivalencies in related tables)



ER Diagram: Starting Example



- **Rectangles:** entity sets
- **Diamonds:** relationship sets
- **Ellipses:** attributes



Entity

- A definable thing—such as a person, object, concept or event—that can have data stored about it.
- Think of entities as nouns.
- Examples: a customer, student, car or product.
- Typically shown as a rectangle.

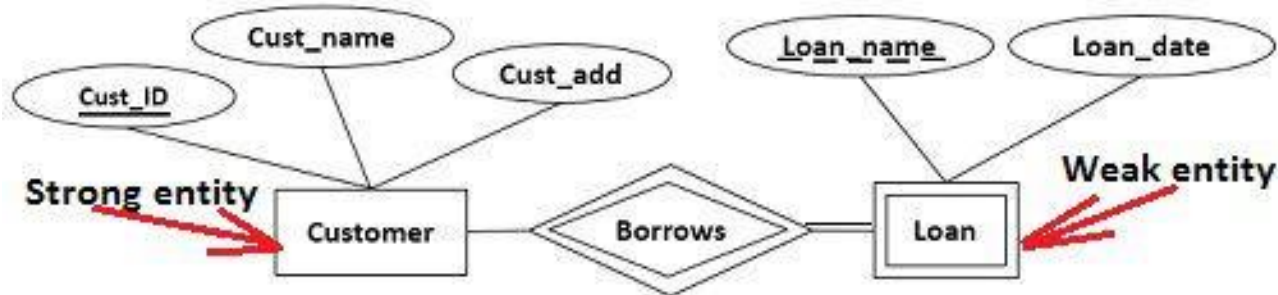


Entity



Entity categories

- **strong entity** can be defined solely by its own attributes.
- **weak entity** cannot be defined solely by its own attributes, An entity set that do not have sufficient attributes to form a primary key.





Entity keys

- Candidate Key
- Primary Key
- Foreign Key
- Composite Key
- Partial Key
- Alternate key
- Super Key



Primary Key

Identifier used to uniquely identify one particular instance of an entity.

- Can be one or more attributes.
- Must be unique .
- Value should not change over time.
- Must always have a value .



Entity keys

■ Candidate Key :

when multiple possible identifiers exist, each is a candidate key.

■ Foreign Keys :

Foreign keys reference a related table through the primary key of that related table.

■ Referential Integrity Constraint:

For every value of a foreign key there is a primary key with that value in the referenced table e.g. if student name is to be used in a dormitory table then that name must exist in the student table.



Attribute

■ A property or characteristic of an entity. Often shown as an oval or circle.

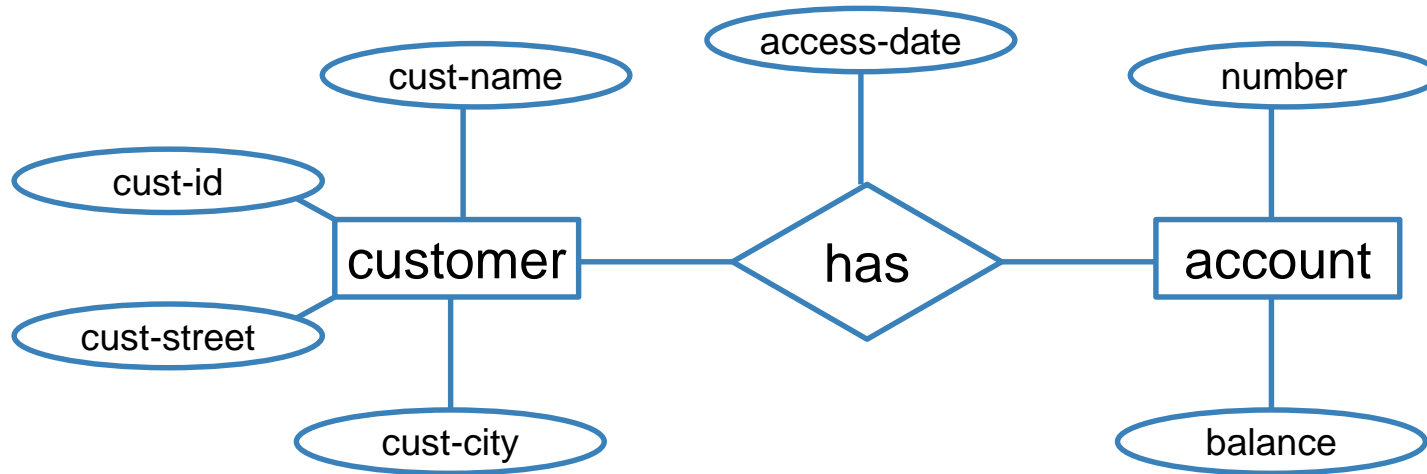
■ **Attribute categories:**

- ☐ Simple (Scalars) Attribute
- ☐ Composite Attribute
- ☐ Derived Attribute
- ☐ Multi-valued Attribute
- ☐ Complex Attribute



Simple Attribute

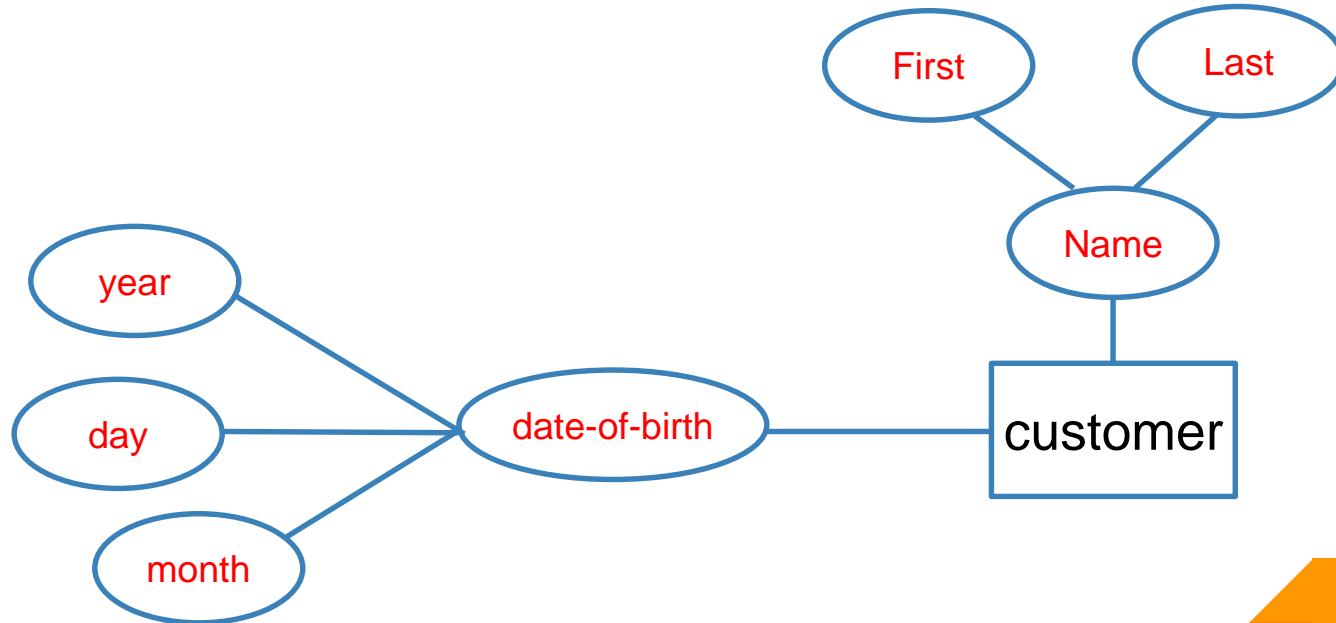
■ Means the attribute value is atomic and can't be further divided, such as a phone number.





Composite Attribute

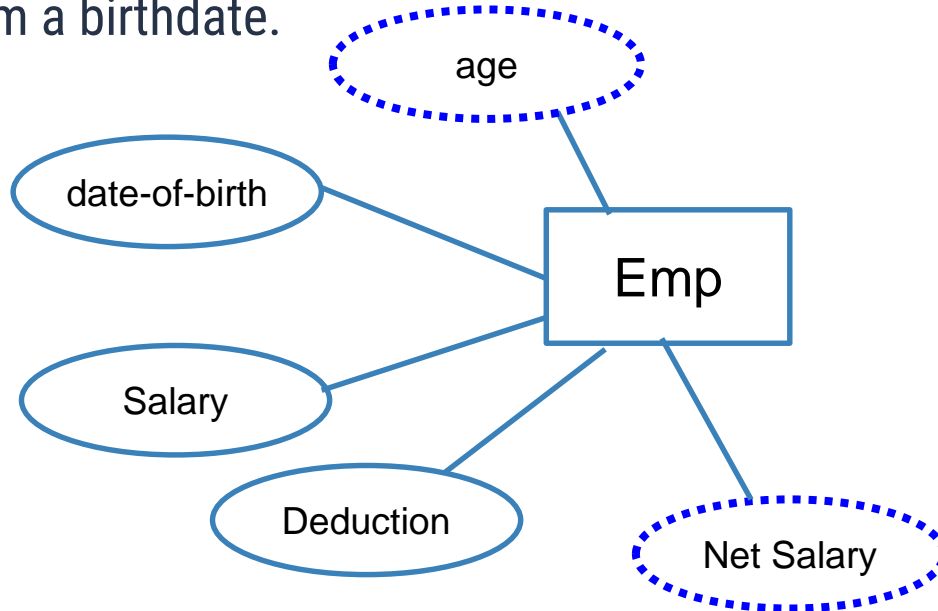
- Sub-attributes spring from an attribute





Derived Attribute

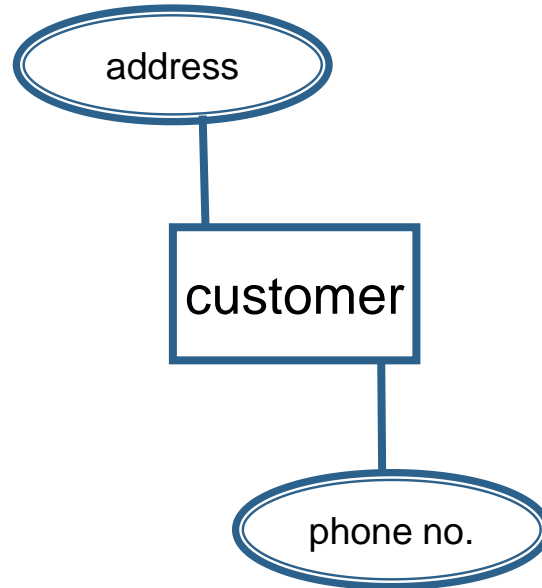
Attributed is calculated or otherwise derived from another attribute, such as age from a birthdate.





Multi-valued

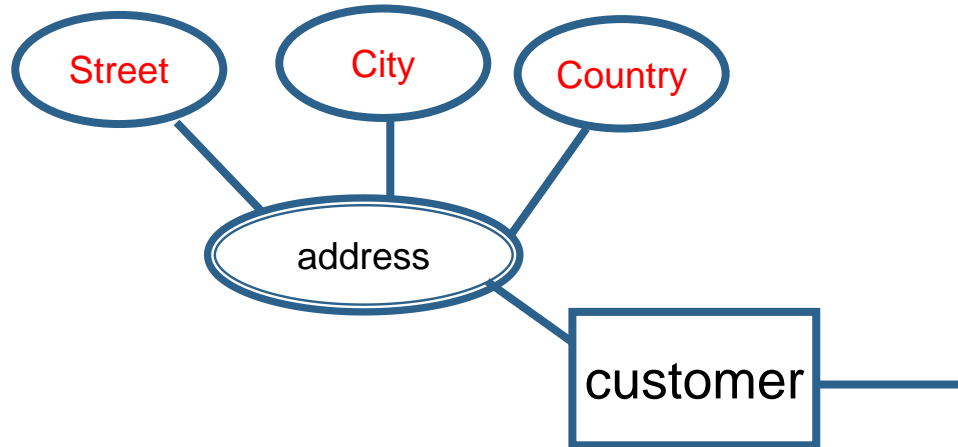
More than one attribute value is denoted, such as multiple phone numbers for a person.





Complex Attribute

■ multi-valued + Composite





Relationship

- A Relationship is an association among several entities.
- A relationship may also have attributes
- For example, consider the entity sets customer and loan and the relationship set borrower. We could associate the attribute date-issued to that relationship to specify the date when the loan was issued.

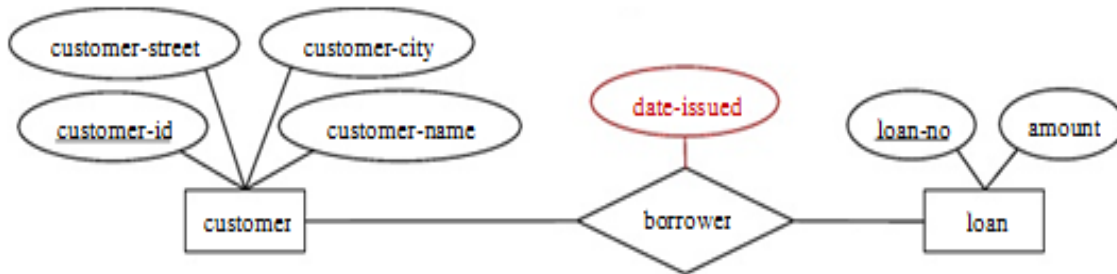


Figure: Descriptive attribute *date-issued*.



Relation Properties

Relation has three Properties:

- ☐ Degree of Relationships
- ☐ Cardinality Constraint
- ☐ Participation Constraint

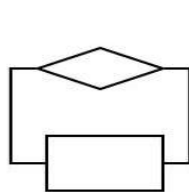


Degree of Relationships

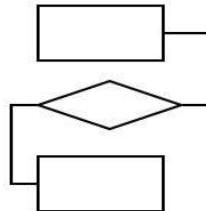
■ number of entity types that participate in a relationship

Three cases:

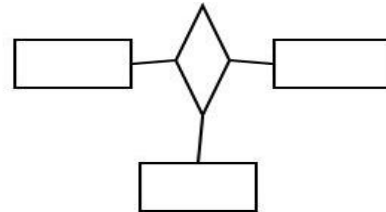
- ❑ **Unary:** between two instances of one entity type
- ❑ **Binary:** between the instances of two entity types
- ❑ **Ternary:** among the instances of three entity types



Unary



Binary

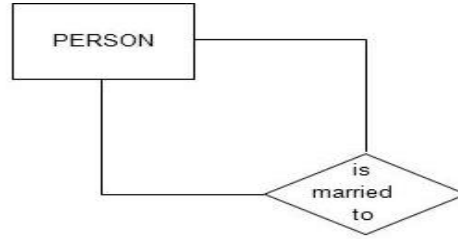


Ternary



Degree of Relationships

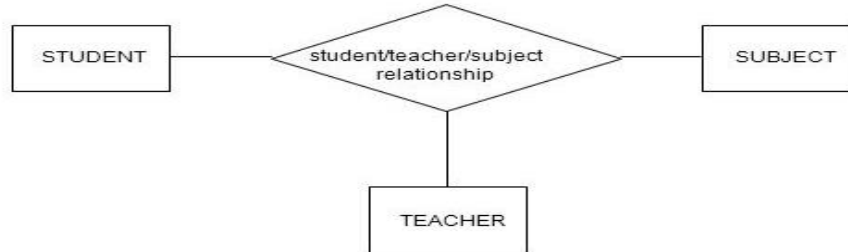
Unary



Binary



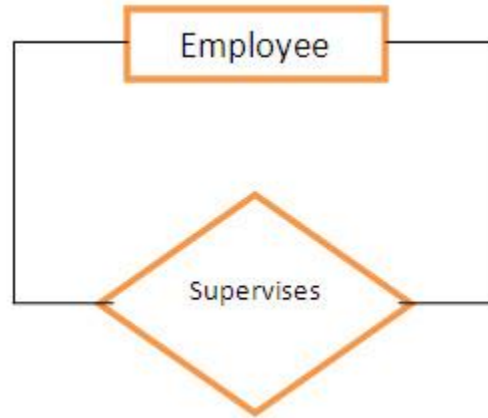
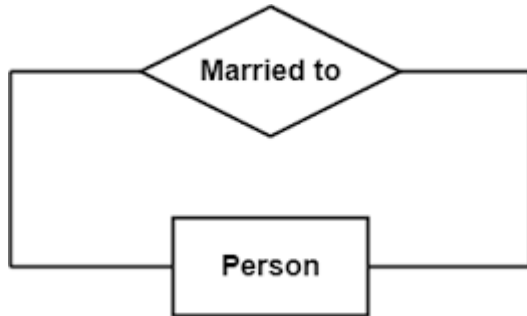
Ternary





Recursive Relationship (Unary)

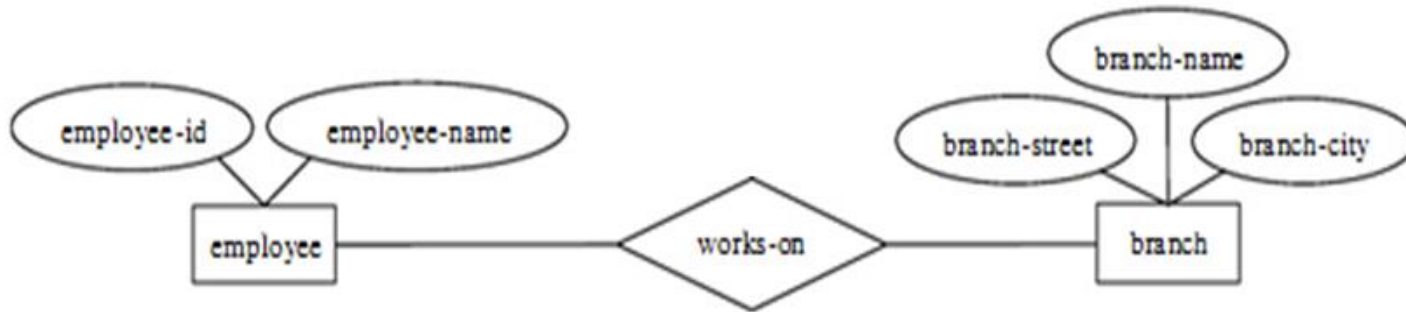
- ❑ **Recursive Relationships** - A relationship in which the same entity participates more than once.





Binary Relationship

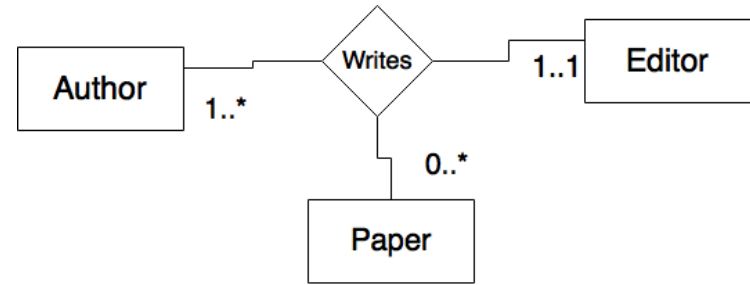
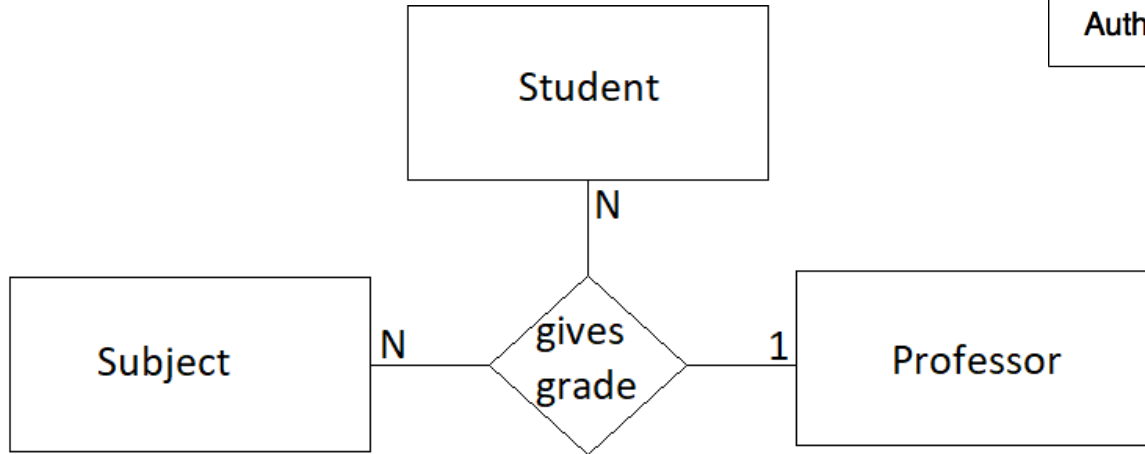
- A **binary relationship** set is of degree 2.





Ternary Relationship

■ Ternary relationship set is of degree 3.





Cardinality

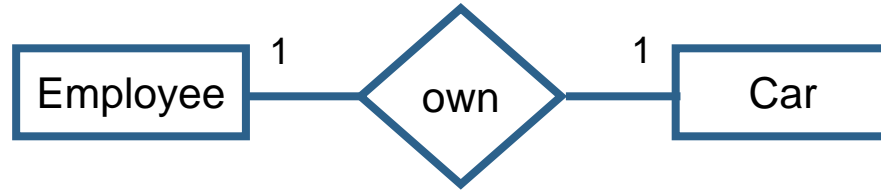
How many instances of one entity will or must be connected to a single instance from the other entities.

- One-One Relationship
- One-Many Relationship
- Many- Many Relationship

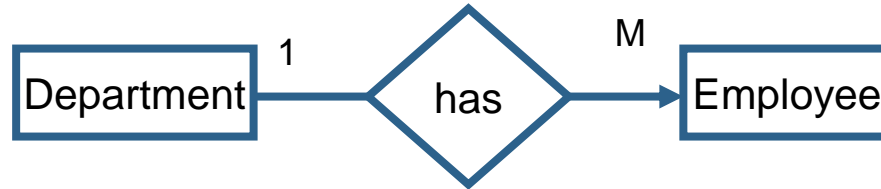


Mapping Cardinalities

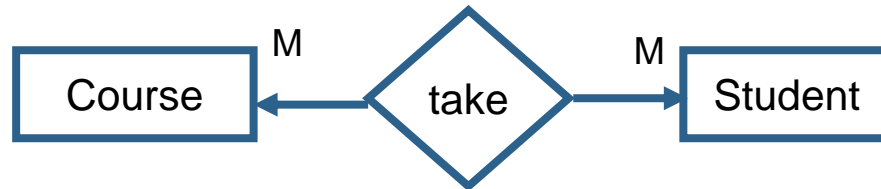
One-to-One



One-to-Many



Many-to-Many





PARTICIPATION CONSTRAINT

- An employee **MUST** work for a department

An employee entity can exist only if it participates in a WORKS_FOR relationship instance

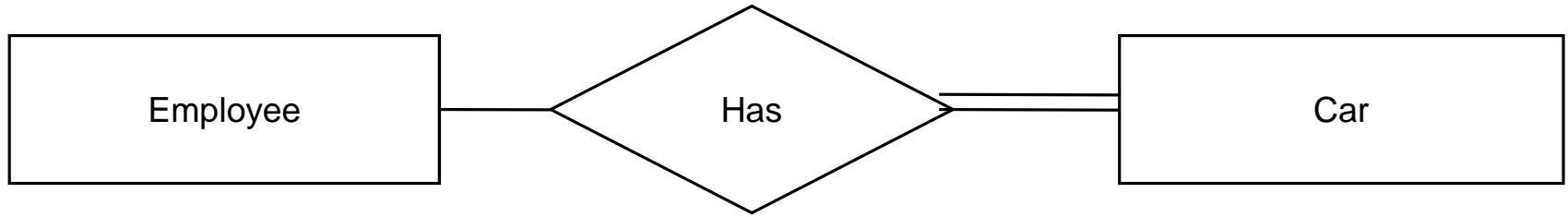
So this participation is **TOTAL**

- Only some employees manage departments

The participation is **PARTIAL**



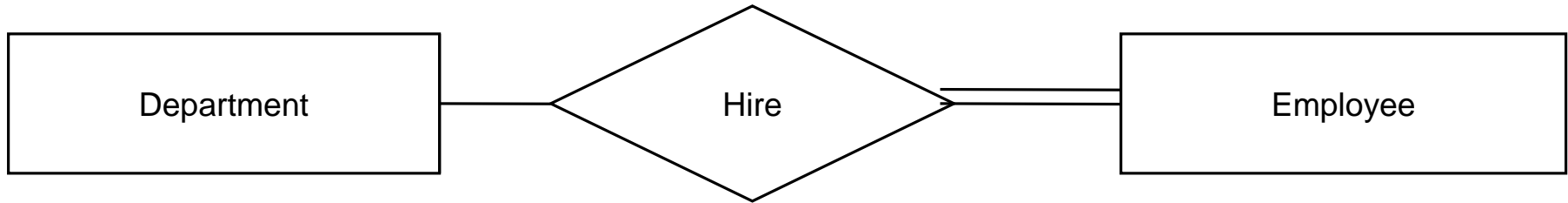
PARTICIPATION CONSTRAINT



- An Employee **may** have a car.
- A Car **must** be assigned to particular employee



PARTICIPATION CONSTRAINT



- A department **may** hire many employees (Zero or more)
 - An employee **must** be employed by a department
- (Department membership is **Optional**, Employee membership is **Mandatory**)



Summary of notation for ERD

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	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

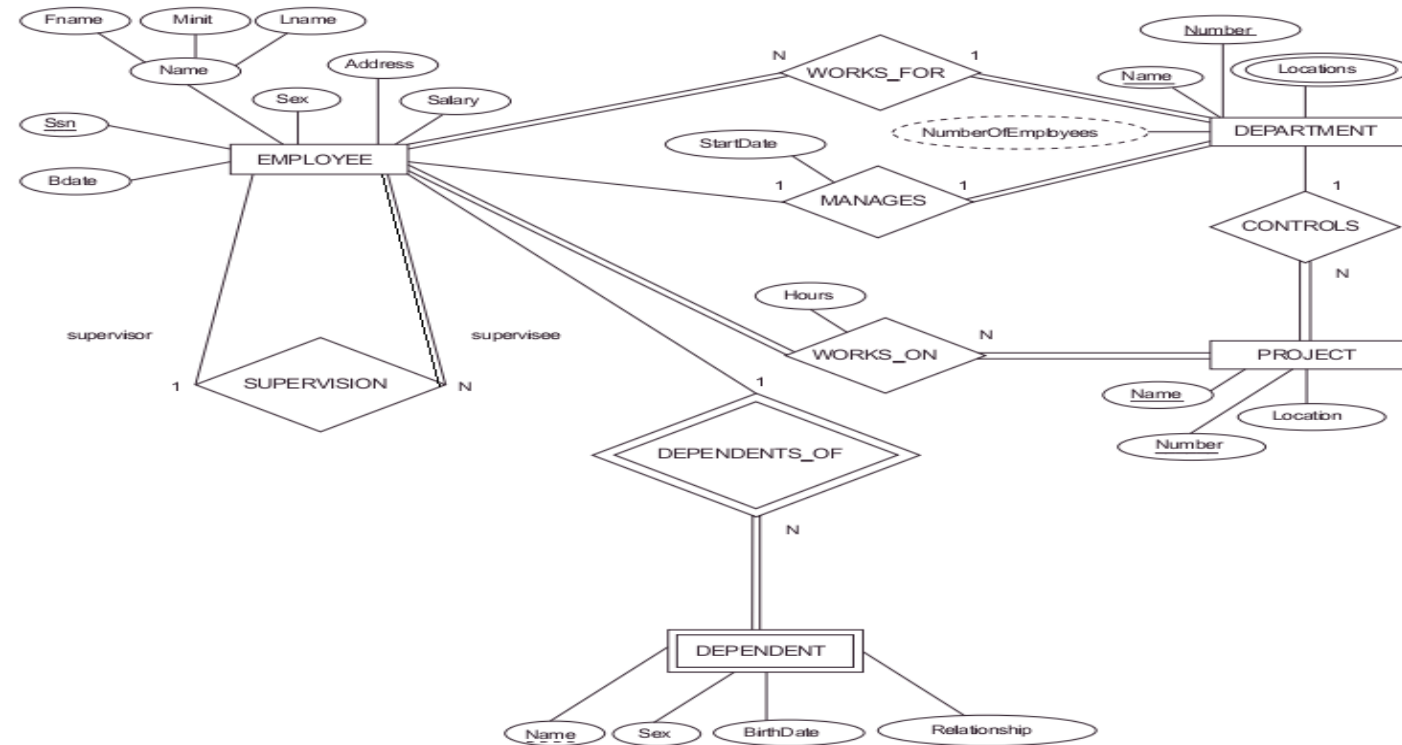


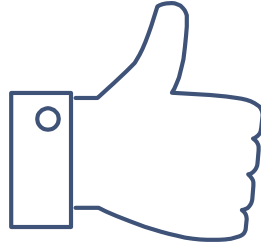
Case Study

Company



Result





THANKS!

Any questions?