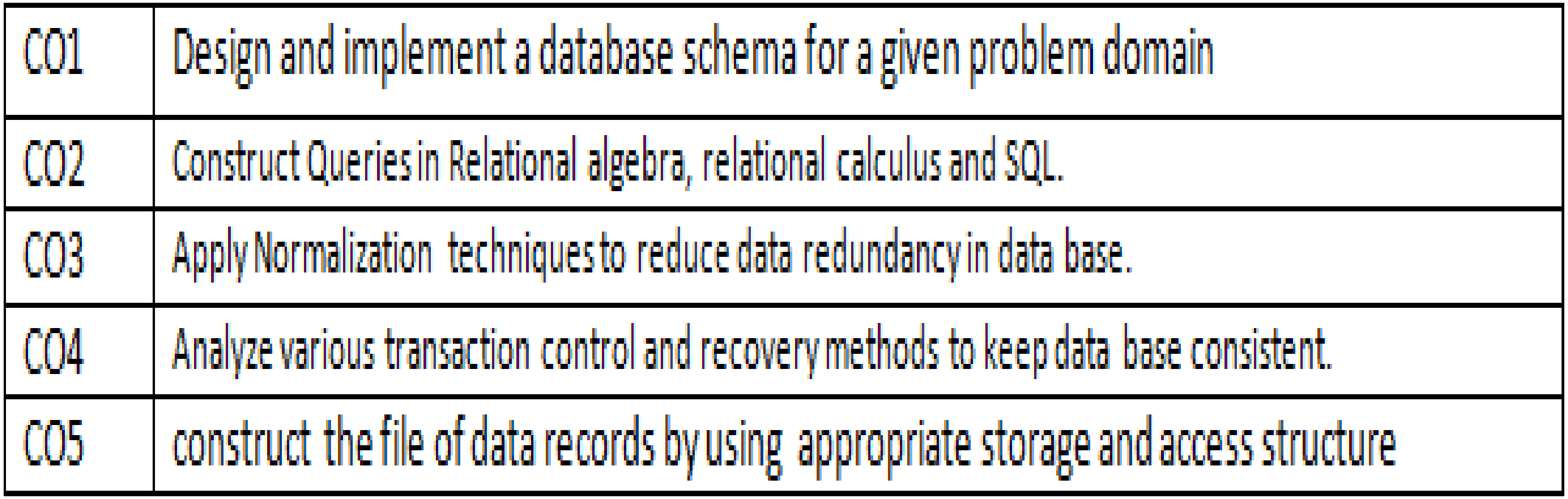
Course Objectives

- To understand purpose of database management system
- Apply concepts like database design and database languages in managing data
- Importance of normalization in dbms and SQL in implementation of database access
- Knowledge of transaction control , recovery strategies , storage and indexing etc

Course Outcomes



CO1	Design and implement a database schema for a given problem domain
CO2	Construct Queries in Relational algebra, relational calculus and SQL.
CO3	Apply Normalization techniques to reduce data redundancy in data base.
CO4	Analyze various transaction control and recovery methods to keep data base consistent.
CO5	construct the file of data records by using appropriate storage and access structure

Text Book

1. Database Management Systems,

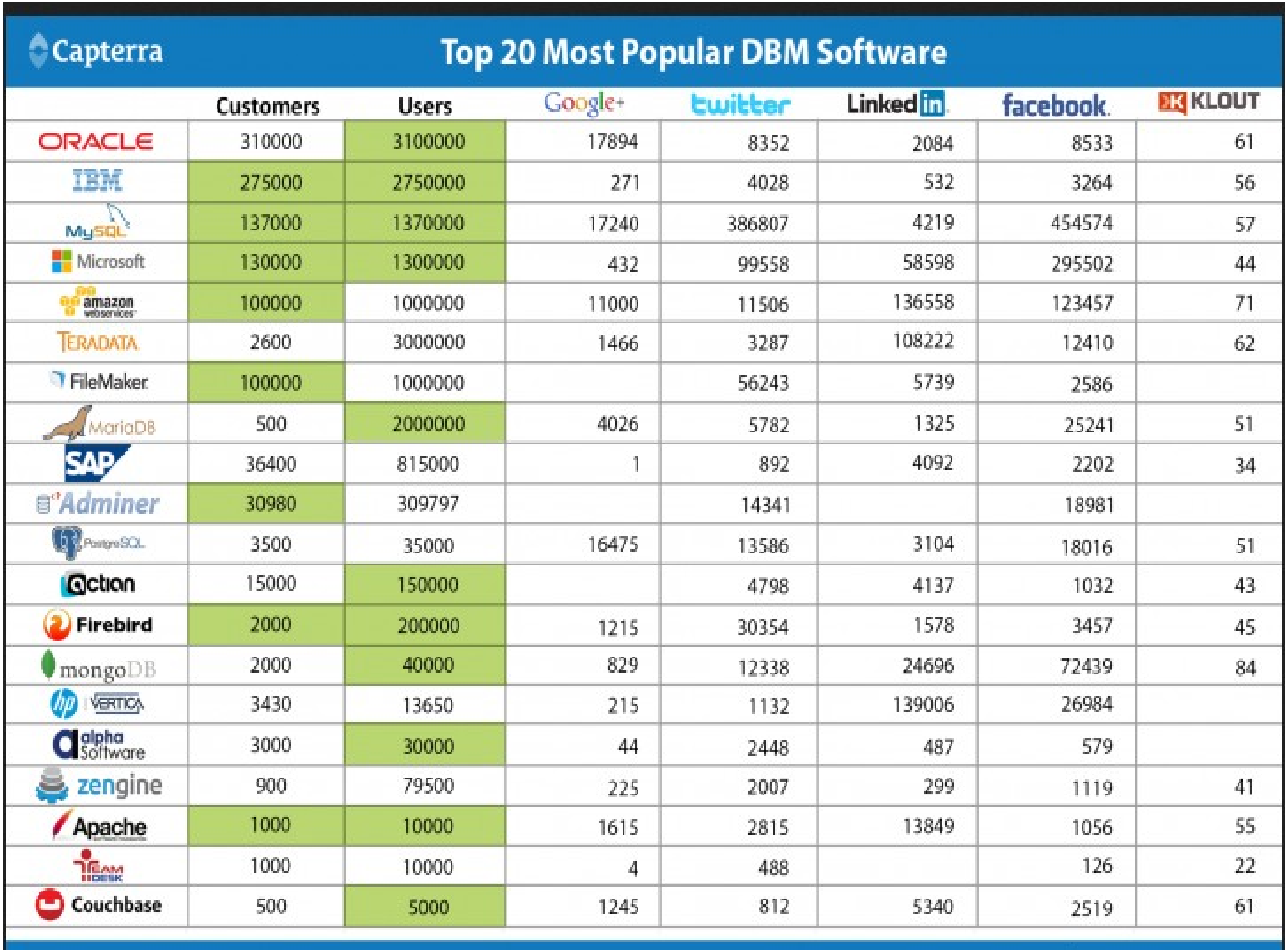
   Raghurama Krishnan, Johannes Gehrke (2007),

   3rd Edition, Tata McGraw-Hill, New Delhi, India

2. Database System Concepts,

   Abraham Silberschatz, Henry F. Korth, S.Sudarshan (2010),

   6th Edition, McGraw-Hill, New Delhi, India.



Capterra

Top 20 Most Popular DBM Software

	Customers	Users	Google+	twitter	LinkedIn	facebook	KLOUT
ORACLE	310000	3100000	17894	8352	2084	8533	61
IBM	275000	2750000	271	4028	532	3264	56
MySQL	137000	1370000	17240	386807	4219	454574	57
Microsoft	130000	1300000	432	99558	58598	295502	44
amazon web services	100000	1000000	11000	11506	136558	123457	71
TERADATA	2600	3000000	1466	3287	108222	12410	62
FileMaker	100000	1000000		56243	5739	2586	
MariaDB	500	2000000	4026	5782	1325	25241	51
SAP	36400	815000	1	892	4092	2202	34
Adminer	30980	309797		14341		18981	
PostgreSQL	3500	35000	16475	13586	3104	18016	51
Actian	15000	150000		4798	4137	1032	43
Firebird	2000	200000	1215	30354	1578	3457	45
mongoDB	2000	40000	829	12338	24696	72439	84
hp VERTICA	3430	13650	215	1132	139006	26984	
alpha Software	3000	30000	44	2448	487	579	
zengine	900	79500	225	2007	299	1119	41
Apache	1000	10000	1615	2815	13849	1056	55
TeamDesk	1000	10000	4	488		126	22
Couchbase	500	5000	1245	812	5340	2519	61

Roles

- Data Engineer
- Data Scientist etc

History of database systems

- **Integrated Data Store**, First general purpose DBMS by charles bachman at General Electric.[Network Model]
- IBM’s **Information Management System(IMS)[**hierachial data model]
- **SABRE system** for making airline reservations By American airlines and IBM
- In 1970,at IBM’s San Jose Research laboratory proposed **relational data model**
- In 1980’s, **SQL** for relational databases by IBM
- **Database transaction management** james gray 1999
- IBM’s **DB2, Oracle 8, Informix UDS**
- **ERP** and **MRP**
- **DBMS with internet**
- **Multimedia databases, streaming data, digital libraries ,NASA’s earth observation system project**
- **Decision making and mining data repositories**

Introduction to database management systems

- **Data:** Data represents known facts or raw information in unorganized form (such as alphabets or numbers or symbols) .
- **Database:** Database is a organized collection of data describing the activities of one or more related organizations.
- **Database management system(DBMS)** is a software designed to assist in storing, maintaining and utilizing large collections of data.

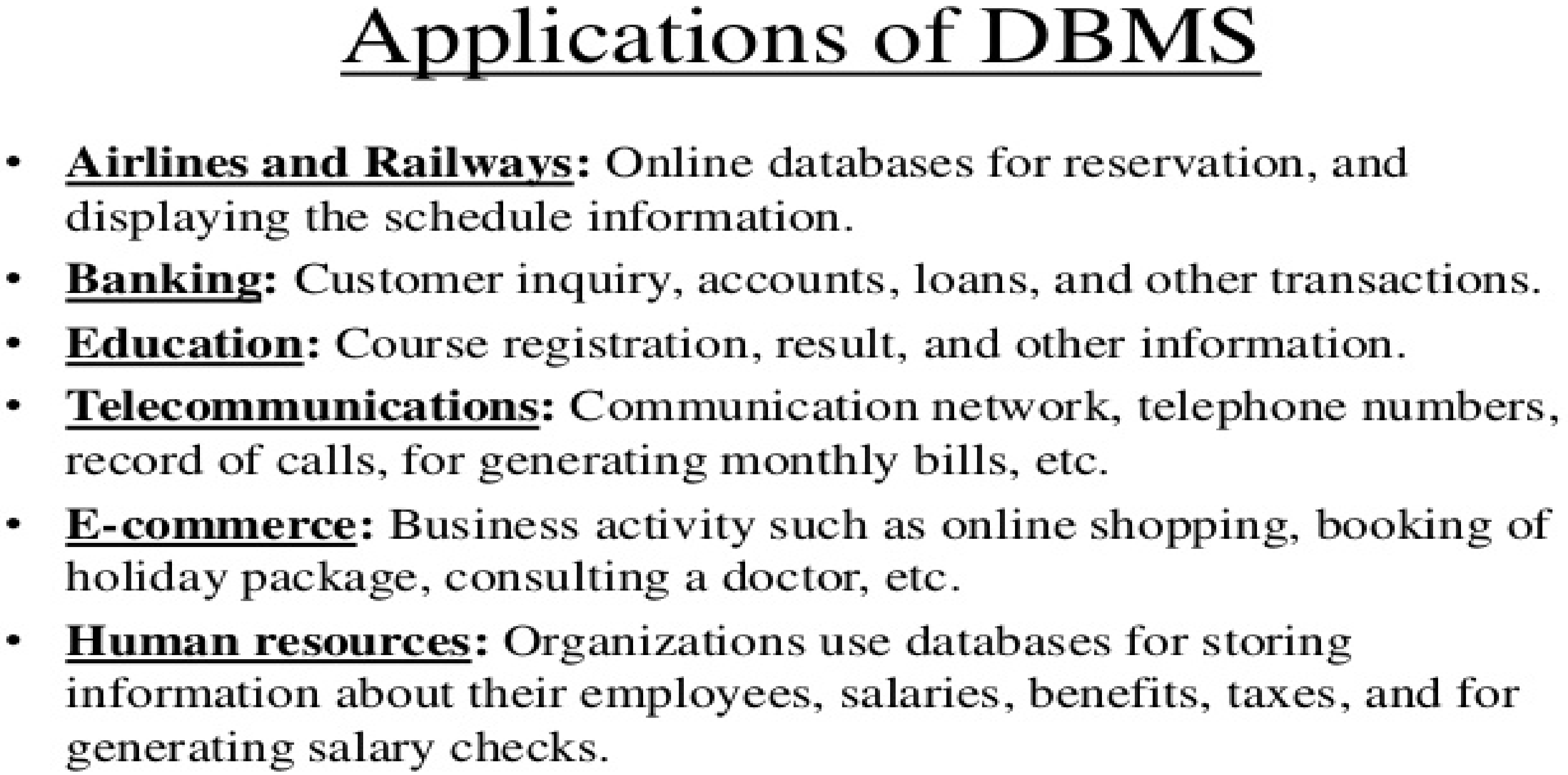
or

A **database-management system (DBMS)** is a computer-software application which interacts with end-users, other applications, and the database itself to access,update,manage and analyze data with the help of set of application programs.

Advantages of Database management systems

- Data Independence
- Efficient data access
- Data integrity and security
- Data administration
- Concurrent access and crash recovery
- Reduced application development time

Database systems applications



Applications of DBMS

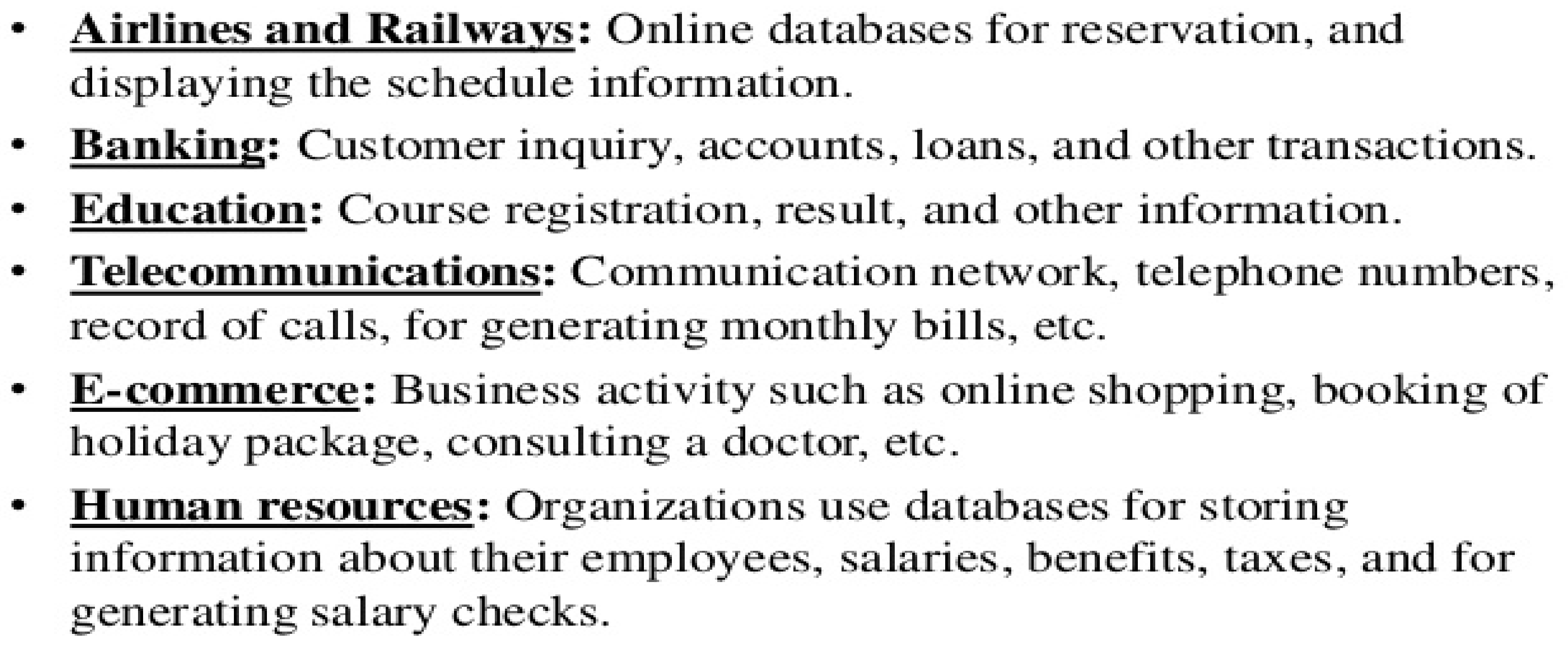
- **Airlines and Railways:** Online databases for reservation, and displaying the schedule information.
- **Banking:** Customer inquiry, accounts, loans, and other transactions.
- **Education:** Course registration, result, and other information.
- **Telecommunications:** Communication network, telephone numbers, record of calls, for generating monthly bills, etc.
- **E-commerce:** Business activity such as online shopping, booking of holiday package, consulting a doctor, etc.
- **Human resources:** Organizations use databases for storing information about their employees, salaries, benefits, taxes, and for generating salary checks.

Applications of Database Management systems

Enterprise Information

- **Sales**
- **Accounting**
- **Human resources**
- **Manufacturing**
- **Online retailers**

Finance and Universities



- **Airlines and Railways:** Online databases for reservation, and displaying the schedule information.
- **Banking:** Customer inquiry, accounts, loans, and other transactions.
- **Education:** Course registration, result, and other information.
- **Telecommunications:** Communication network, telephone numbers, record of calls, for generating monthly bills, etc.
- **E-commerce:** Business activity such as online shopping, booking of holiday package, consulting a doctor, etc.
- **Human resources:** Organizations use databases for storing information about their employees, salaries, benefits, taxes, and for generating salary checks.

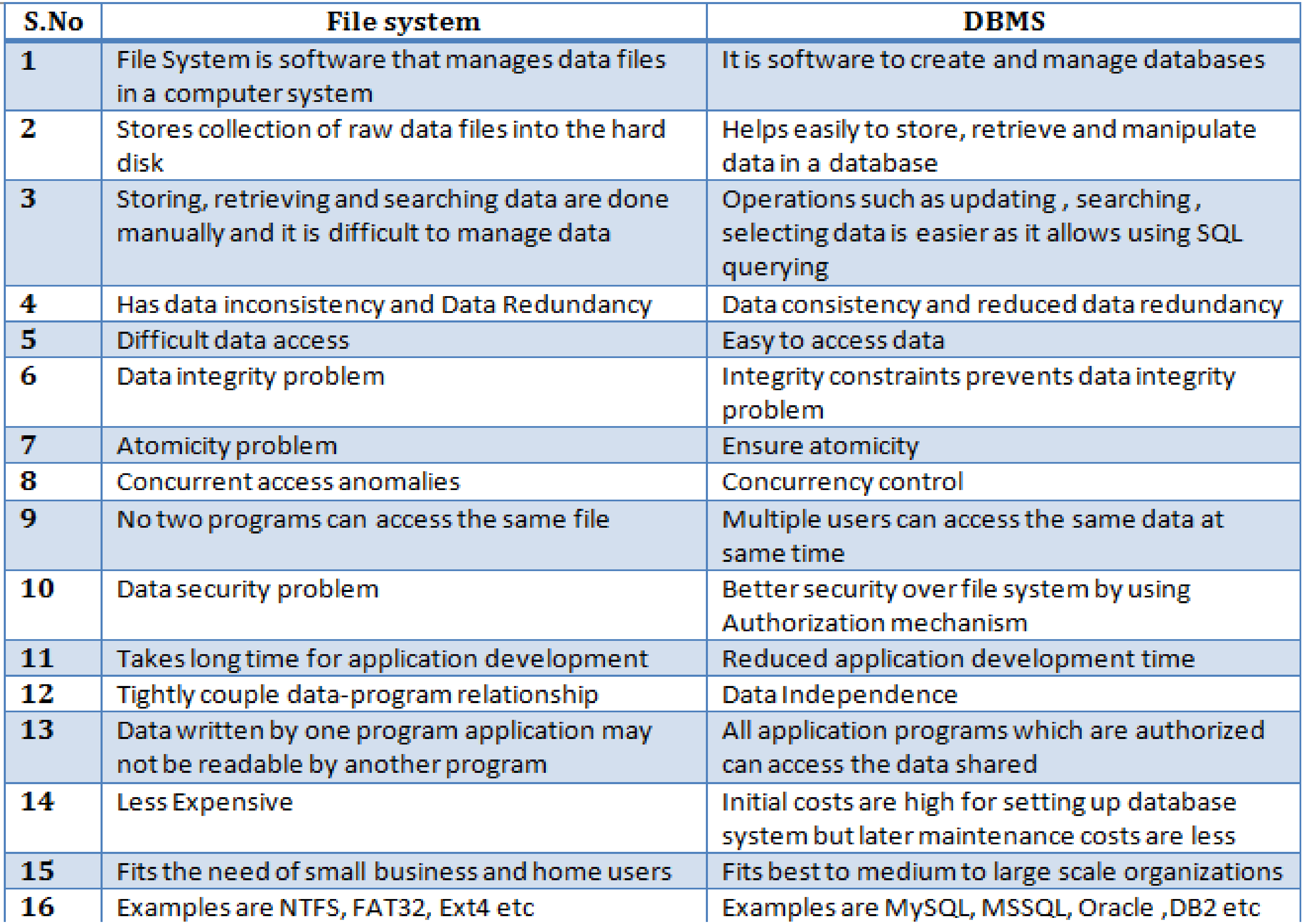
Database systems Vs file systems

- File storage refers to a collection of operating system files.

- DBMS features to manage the data in a robust and efficient manner.

Advantages of dbms over file system

- No redundant data – Redundancy removed by data normalization
- Data Consistency and Integrity – data normalization takes care of it too
- Secure – Each user has a different set of access
- Privacy – Limited access
- Easy access to data
- Easy recovery
- Flexible



S.No	File system	DBMS
1	File System is software that manages data files in a computer system	It is software to create and manage databases
2	Stores collection of raw data files into the hard disk	Helps easily to store, retrieve and manipulate data in a database
3	Storing, retrieving and searching data are done manually and it is difficult to manage data	Operations such as updating , searching , selecting data is easier as it allows using SQL querying
4	Has data inconsistency and Data Redundancy	Data consistency and reduced data redundancy
5	Difficult data access	Easy to access data
6	Data integrity problem	Integrity constraints prevents data integrity problem
7	Atomicity problem	Ensure atomicity
8	Concurrent access anomalies	Concurrency control
9	No two programs can access the same file	Multiple users can access the same data at same time
10	Data security problem	Better security over file system by using Authorization mechanism
11	Takes long time for application development	Reduced application development time
12	Tightly couple data-program relationship	Data Independence
13	Data written by one program application may not be readable by another program	All application programs which are authorized can access the data shared
14	Less Expensive	Initial costs are high for setting up database system but later maintenance costs are less
15	Fits the need of small business and home users	Fits best to medium to large scale organizations
16	Examples are NTFS, FAT32, Ext4 etc	Examples are MySQL, MSSQL, Oracle ,DB2 etc

View of data

- Abstraction
- Data Abstraction
- Instance and schema

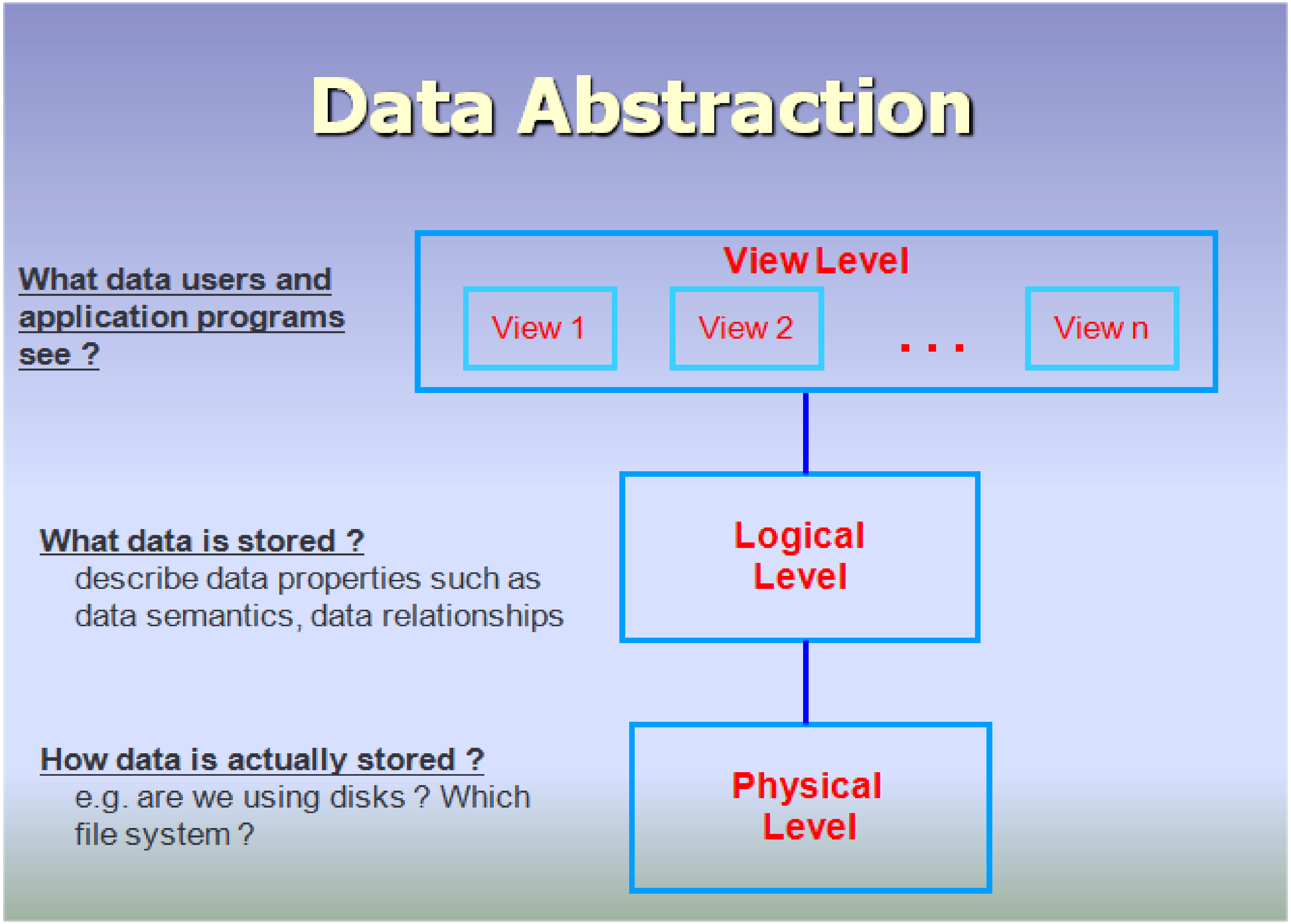
- **Abstraction :**Hiding unnecessary details from the user and providing abstract view of data which is required to users.

- **Data Abstraction:** Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.

Schema

- Design of a database is called schema.
- 3 types:
  - Physical schema
  - Logical schema
  - View schema

Three levels of Data Abstraction



Data Abstraction

What data users and application programs see ?

What data is stored ?
describe data properties such as data semantics, data relationships

How data is actually stored ?
e.g. are we using disks ? Which file system ?

View Level

View 1

View 2

. . .

View n

Logical Level

Physical Level

Data models

- Data models determines the logical structure of a database
- A **Data Model** is a logical structure of Database. It describes the design of database to reflect entities, attributes, relationship among data, constrains etc.

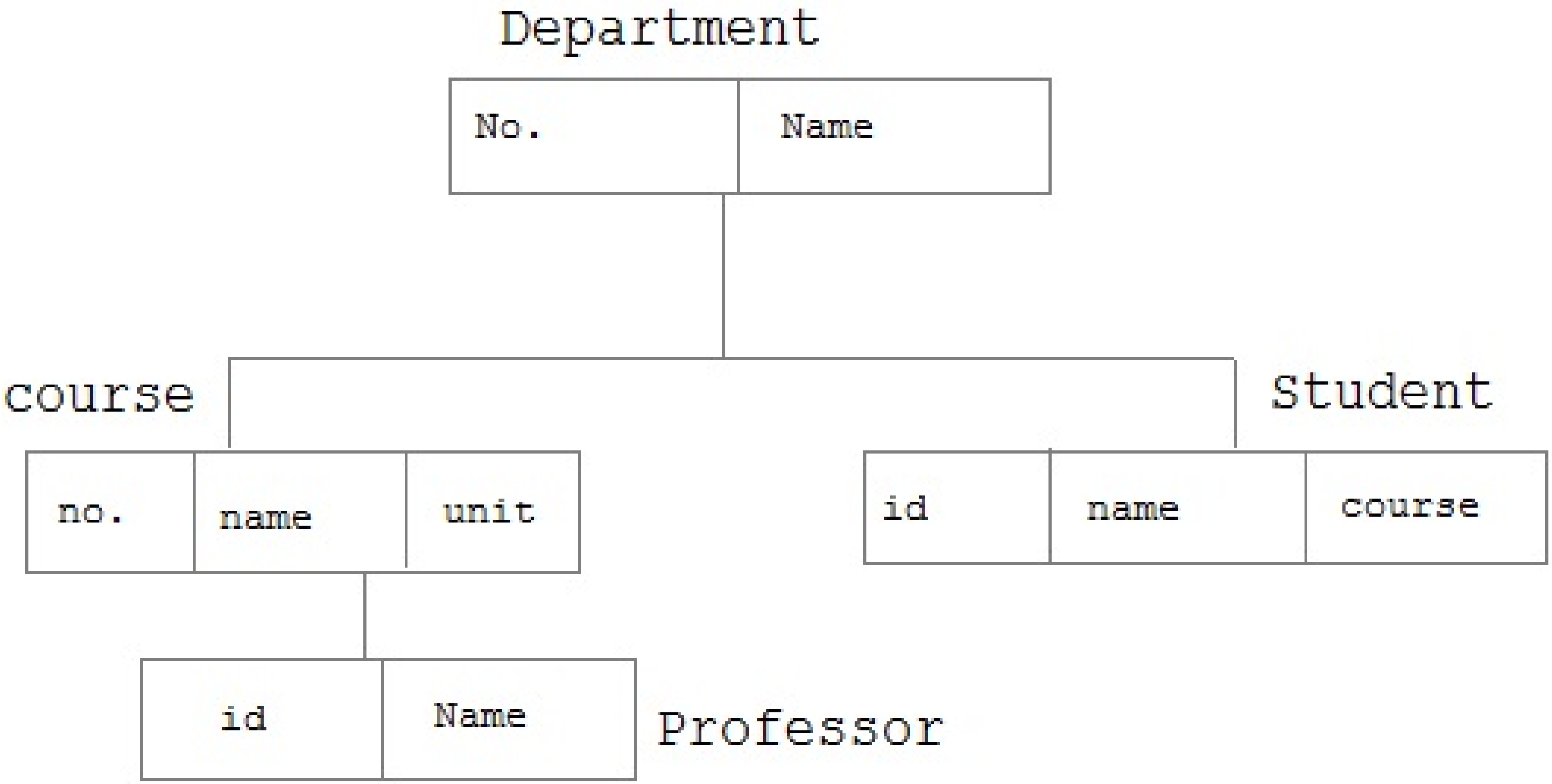
Data models

- Relational data model
- Entity – relationship model
- Hierarchical data model
- Network model
- Object oriented model
- Object relational model
- Semi structured data model
- Flat data model

Hierarchical data model

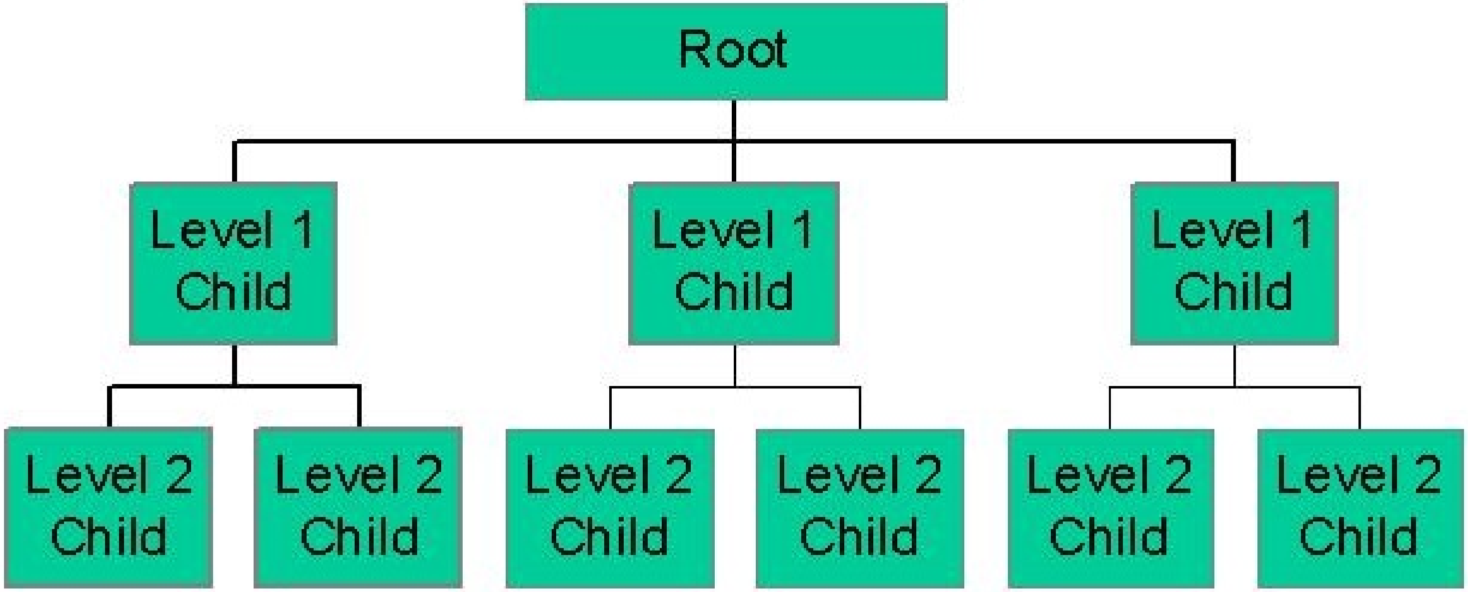
- A **hierarchical database model** is a data model in which the data is organized into a tree-like structure. The data is stored as **records** which are connected to one another through **links**.
- A record is a collection of fields, with each field containing only one value.

Example for hierarchical model

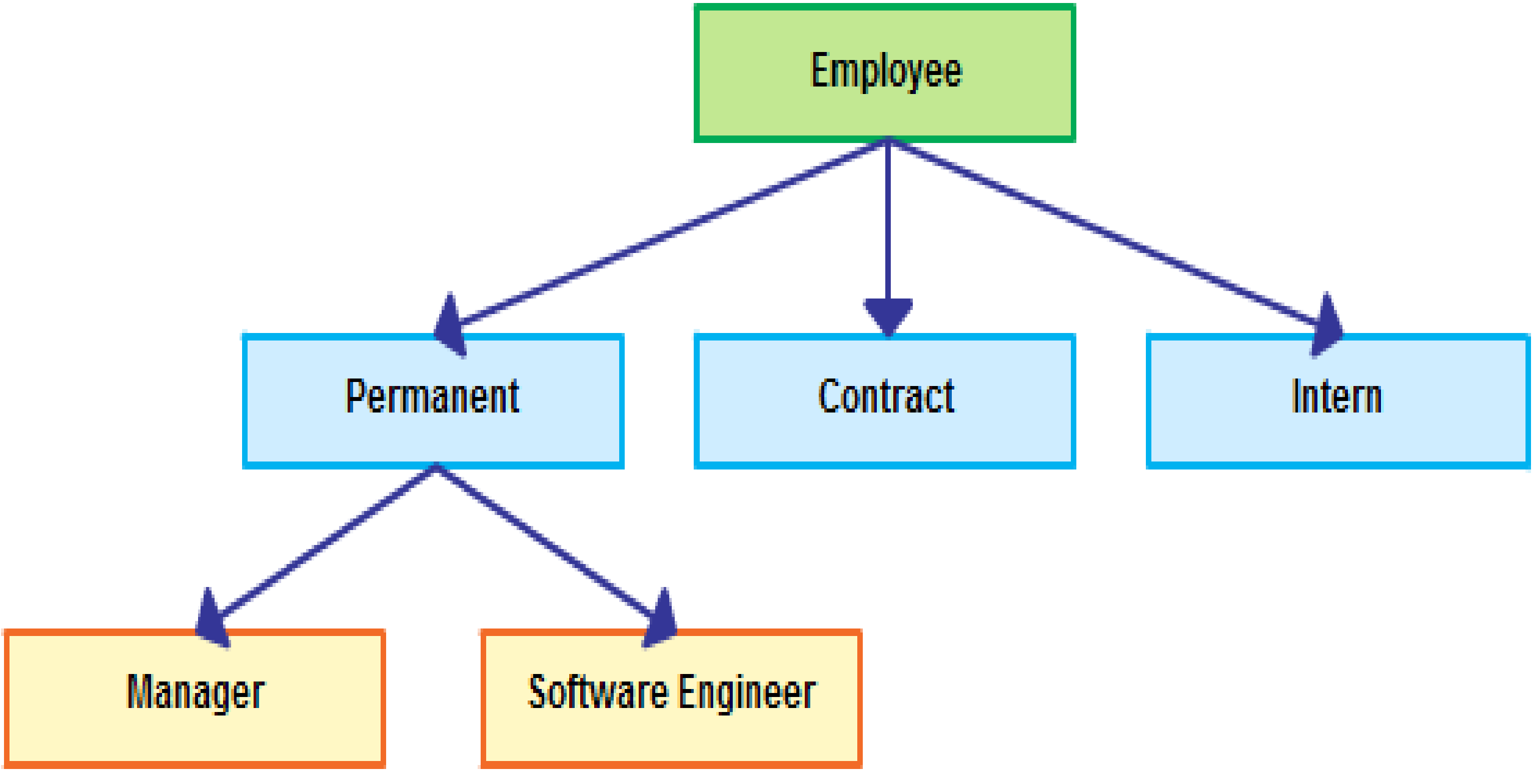


Example of hierarchical model

Hierarchical database model



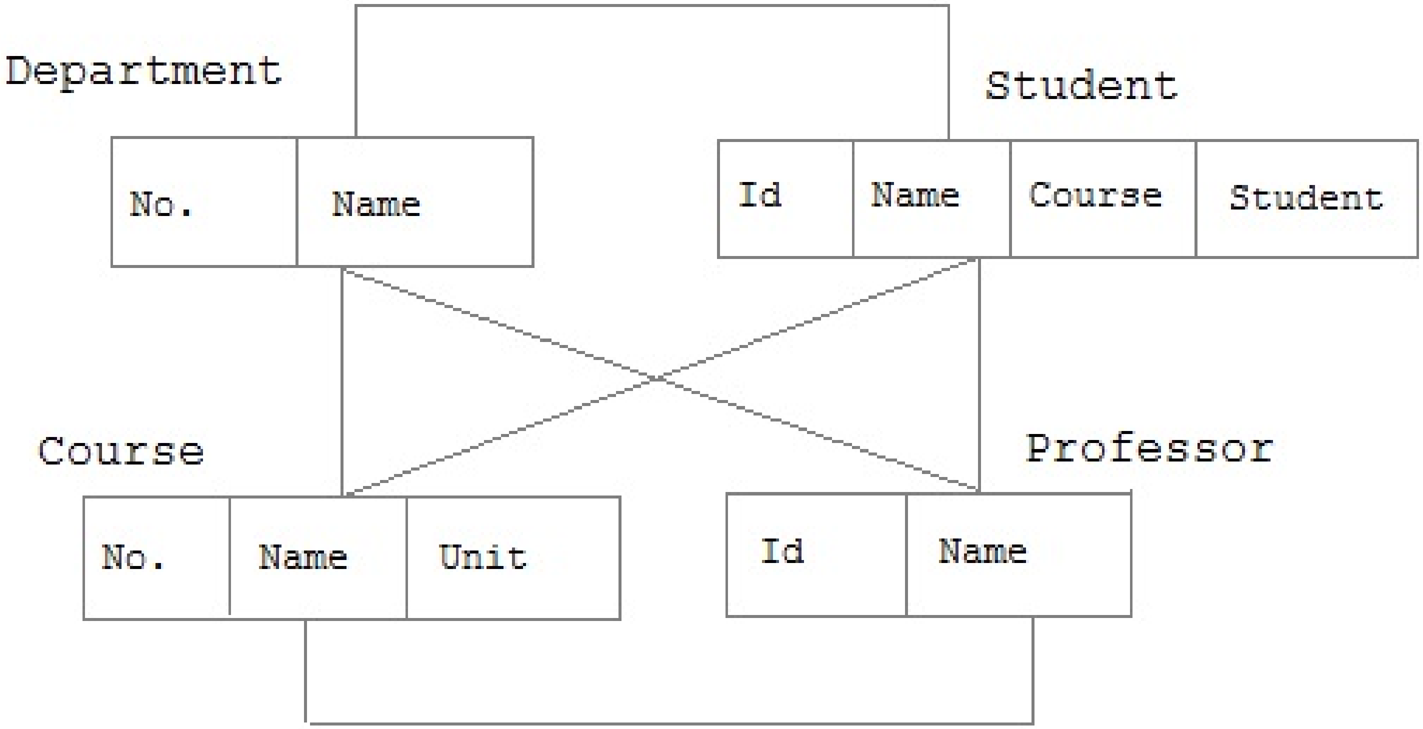
Hierarchical model



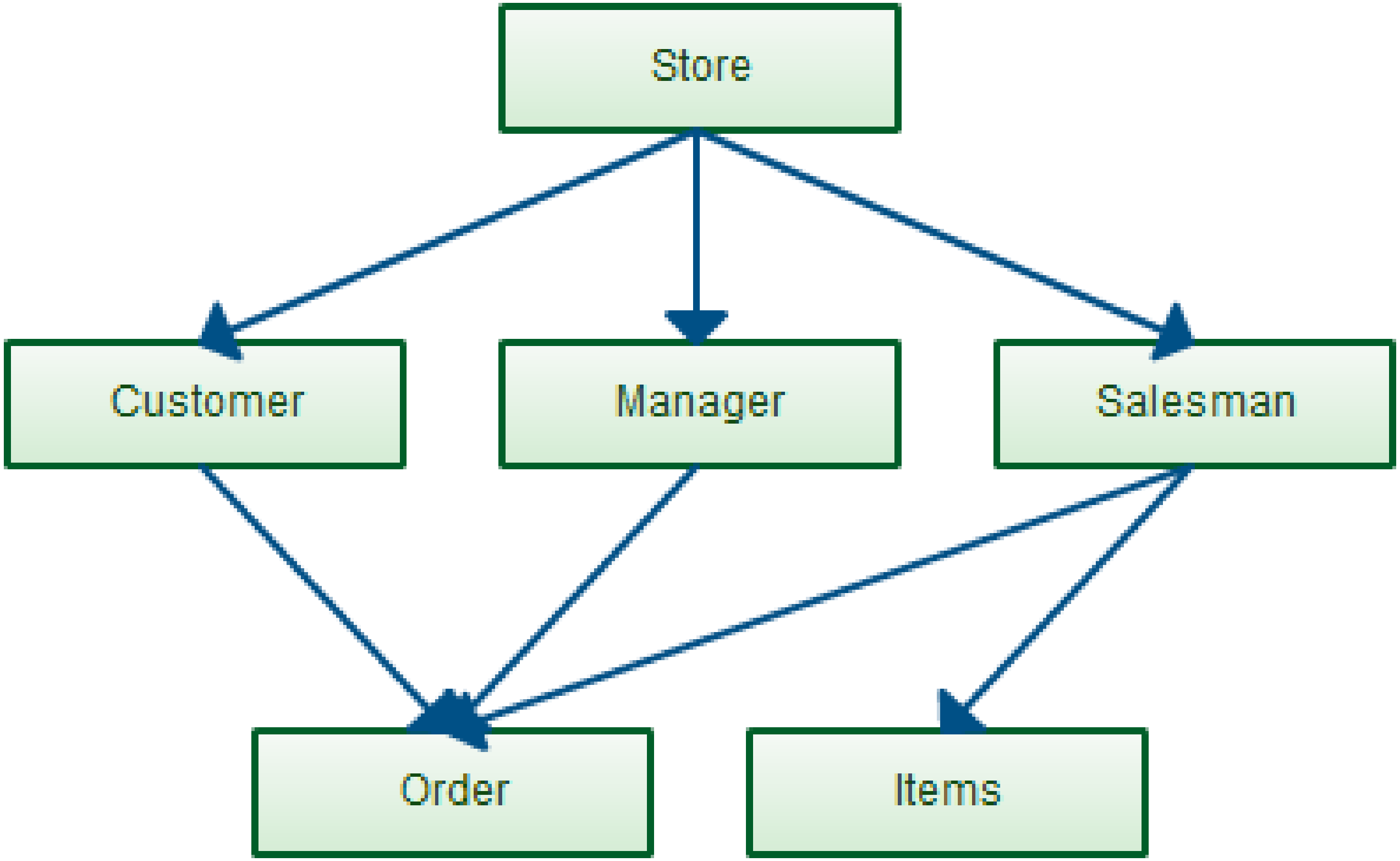
Network model

- In the network model, entities are organized in a graph, in which some entities can be accessed through several paths.

Network data model example



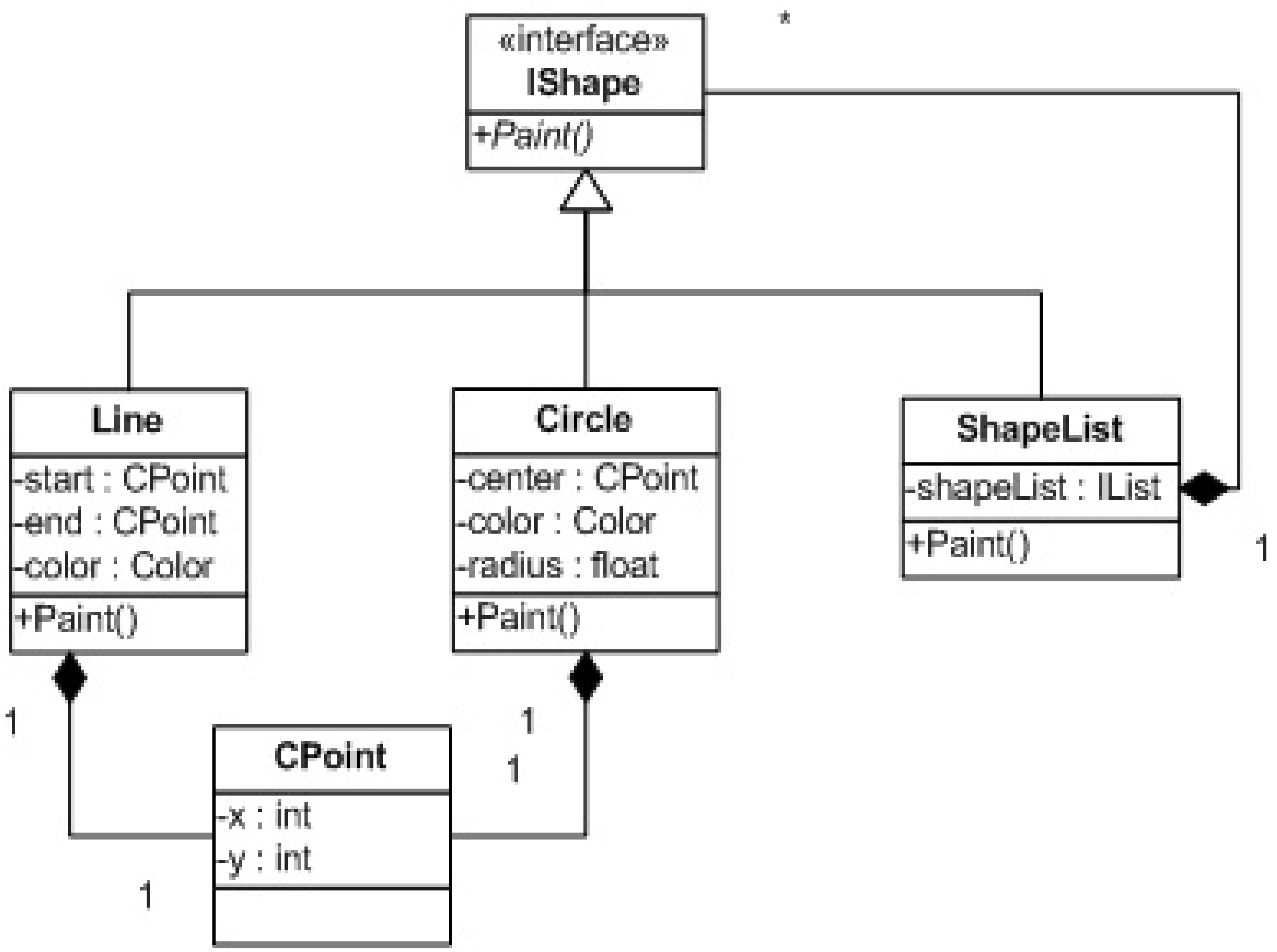
Example for network model



Object oriented data model

- This data model is another method of representing real world objects.
- It considers each object in the world as objects and isolates it from each other.
- It groups its related functionalities together and allows inheriting its functionality to other related sub-groups.

Object oriented model

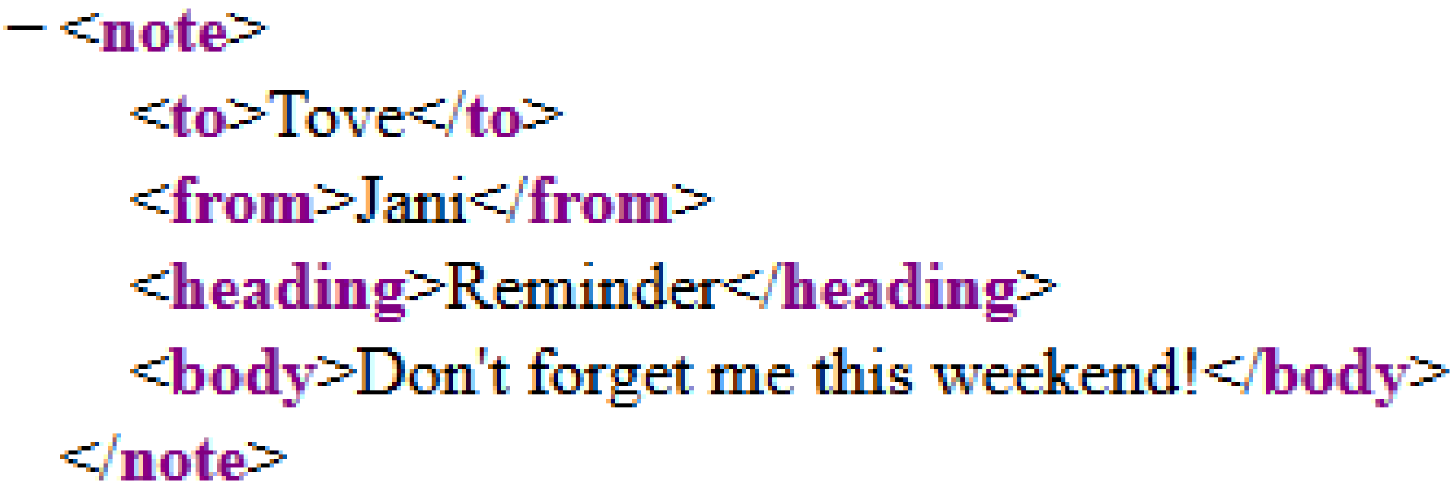


Object relational data model

- An **object-relational database** (**ORD**), or **object-relational database management system** (**ORDBMS**), is a database management system (DBMS) similar to a relational database, but with an object-oriented database model: objects, classes and inheritance are directly supported in database schemas and in the query language

Semi structured data model:

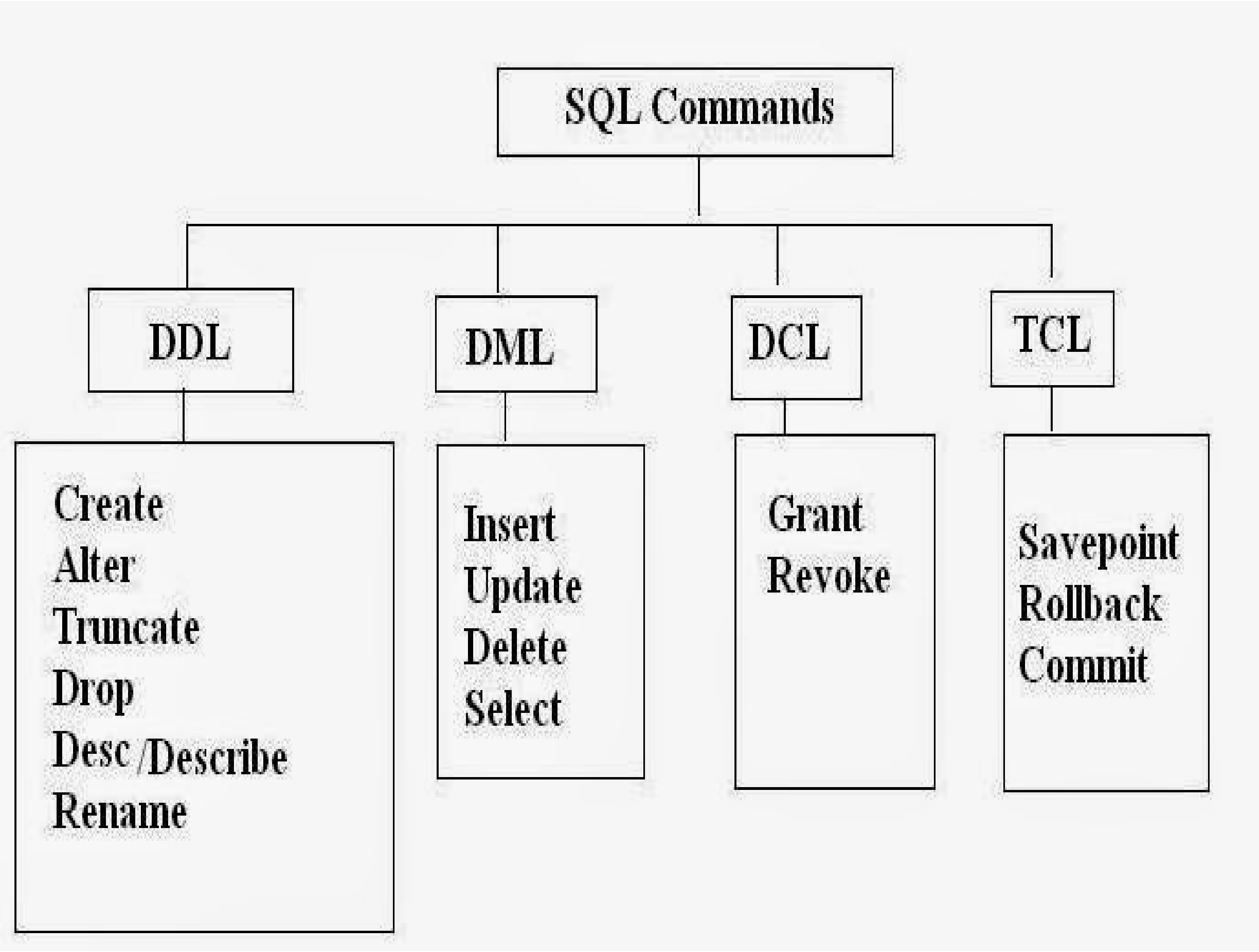
The semi-structured data model is designed as an evolution of the relational data model that allows the representation of data with a flexible structure.
Semi-structured data model is model where schema is part of data



```
- <note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
```

Summary of the evolution of database modeling

File System	**No database model**
Hierarchical	Allowing one-to-many relationships
Network	Allowing for special relationships
Relational	Allowing for individual element access anywhere in the database
Object	Handling high-speed application of small data items within large highly complex data sets
Object-relational	Including the most accountable aspects of object database into the structure of the relational database model



SQL Commands

DDL

Create
Alter
Truncate
Drop
Desc /Describe
Rename

DML

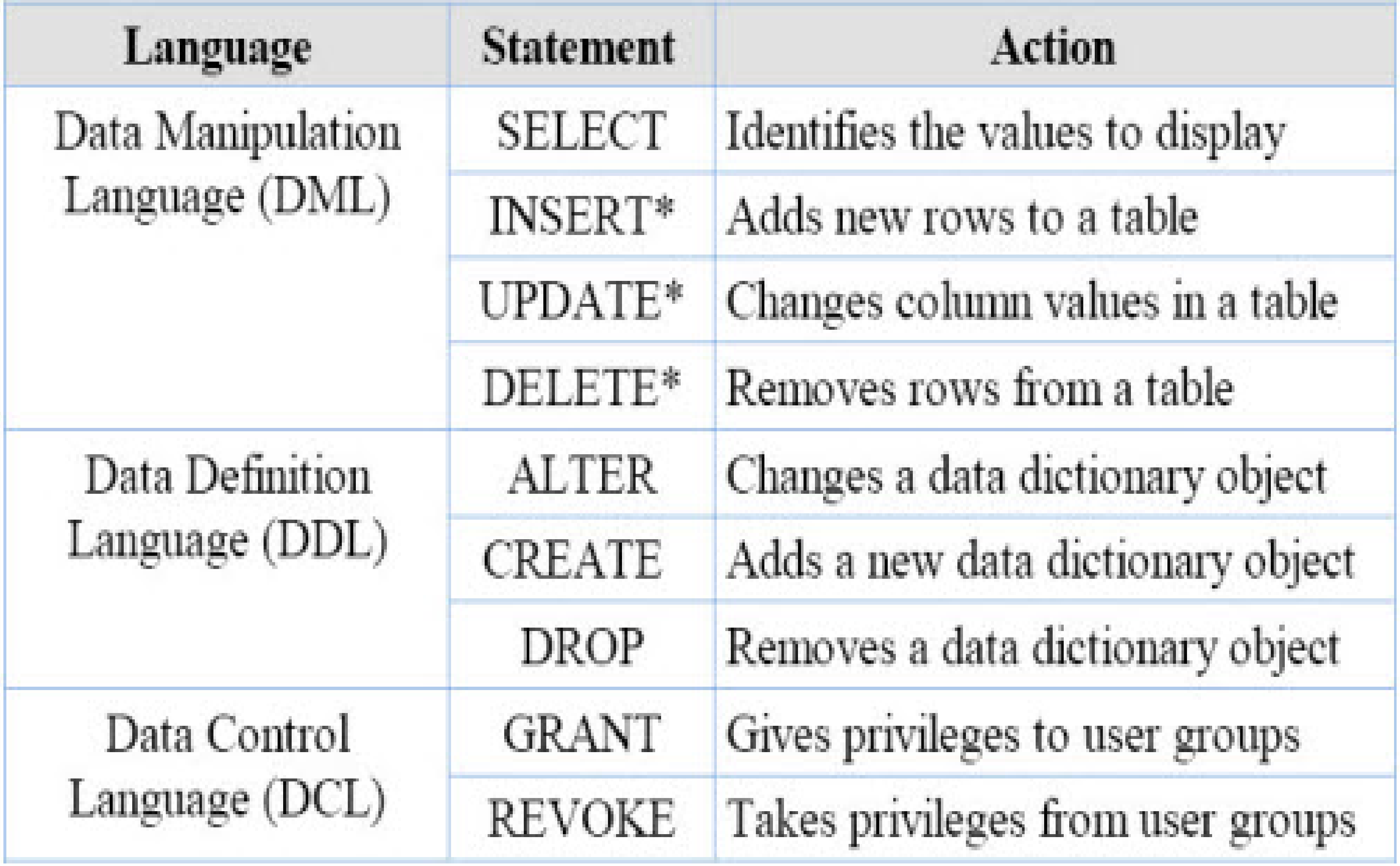
Insert
Update
Delete
Select

DCL

Grant
Revoke

TCL

Savepoint
Rollback
Commit



Language	Statement	Action
Data Manipulation Language (DML)	SELECT	Identifies the values to display
	INSERT*	Adds new rows to a table
	UPDATE*	Changes column values in a table
	DELETE*	Removes rows from a table
Data Definition Language (DDL)	ALTER	Changes a data dictionary object
	CREATE	Adds a new data dictionary object
	DROP	Removes a data dictionary object
Data Control Language (DCL)	GRANT	Gives privileges to user groups
	REVOKE	Takes privileges from user groups

- Select
  - Unique
  - Distinct
  - Count
  - As
  - In
  - Sum
  - IS NULL
  - IS NOT NULL
  - ORDER BY
  - AND, OR

Database users and administration

- Database users
  1. Application Programmers
  2. Sophisticated users
  3. Specialized users
  4. Stand-alone users
  5. Native Users

Database administrators

- Responsibilities of DBA:
  - Installing and upgrading DBMS servers
  - Design and implementation of databases
  - Schema definition , storage structure, access method definition
  - Performance Tuning
  - Migrate database servers
  - Backup and recovery
  - Security
  - Documentation etc

Types of database administrators

- Administrative DBA
- Development DBA
- Database Architect
- Data warehouse DBA
- Application DBA
- OLAP DBA

Transaction management

- Transaction
- Partial transactions
- Concurrent execution of transactions
- Locking protocol
- Locks
- Shared locks
- Exclusive locks
- Incomplete transactions
- System crashes
- Log
- Write Ahead Log (WAL)
- Checkpoint
- Periodic checkpoints

Database system structure

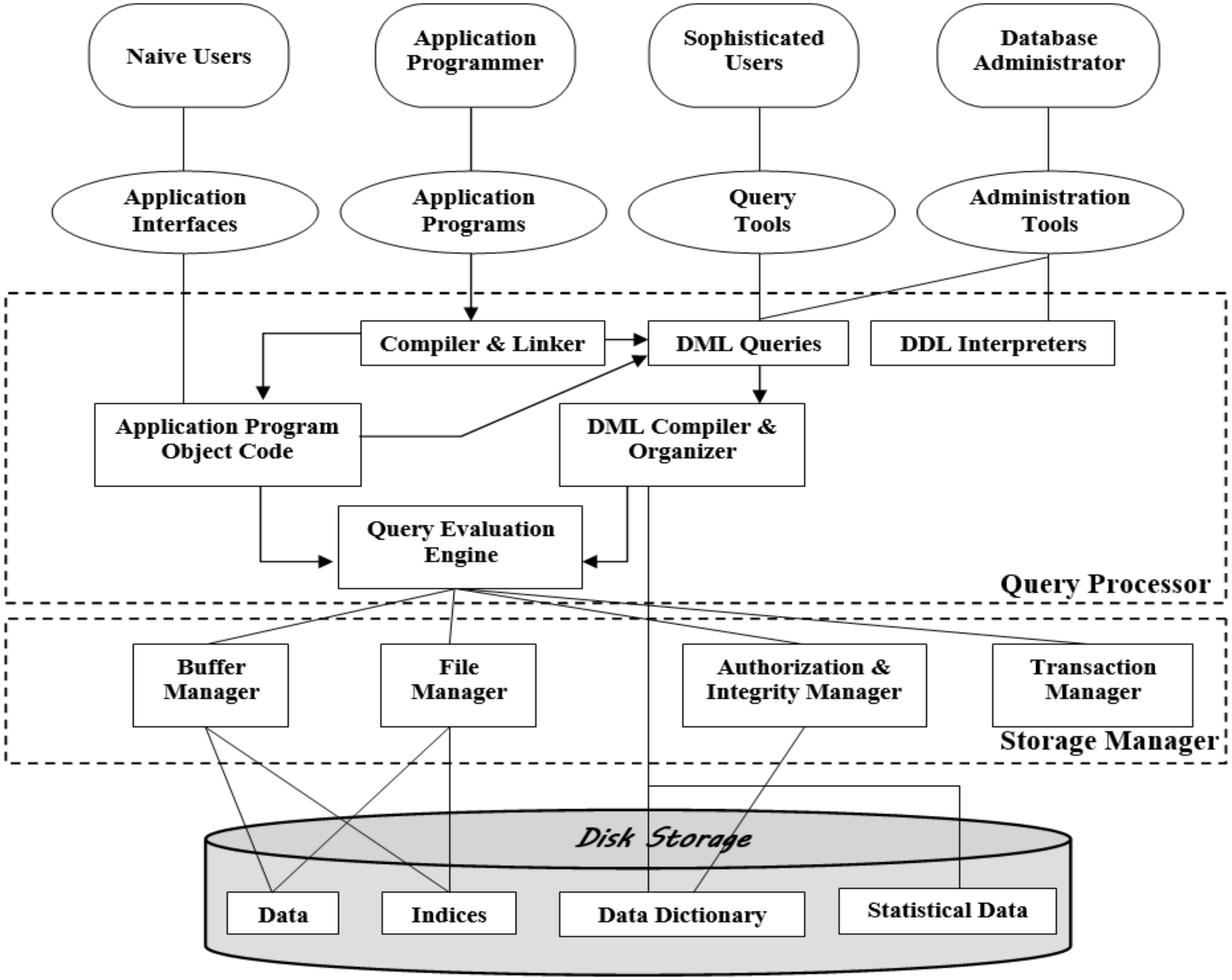
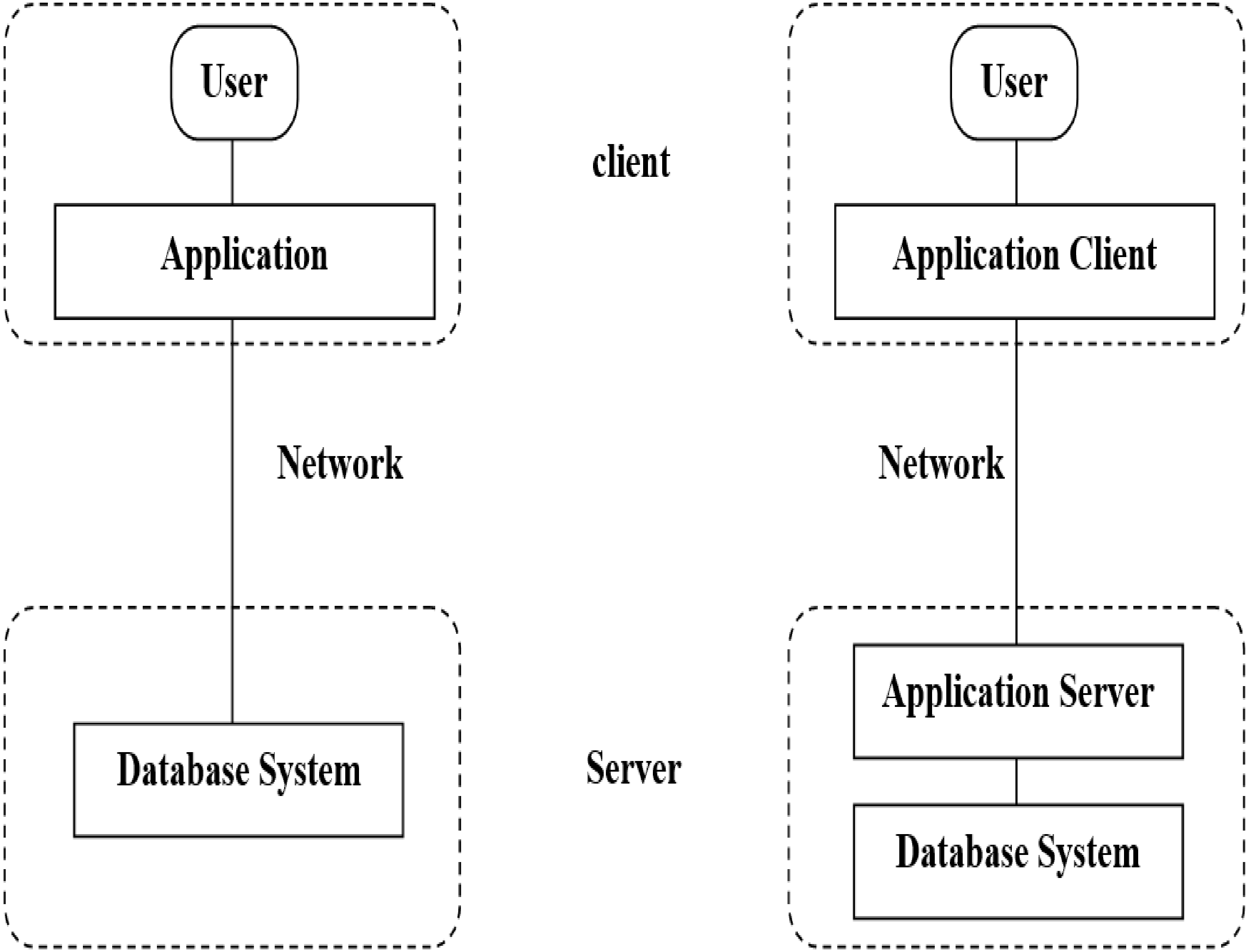


Figure: System Architecture



(a) Two-tier Architecture

(a) Three-tier Architecture

Introduction to database design
Database design process steps

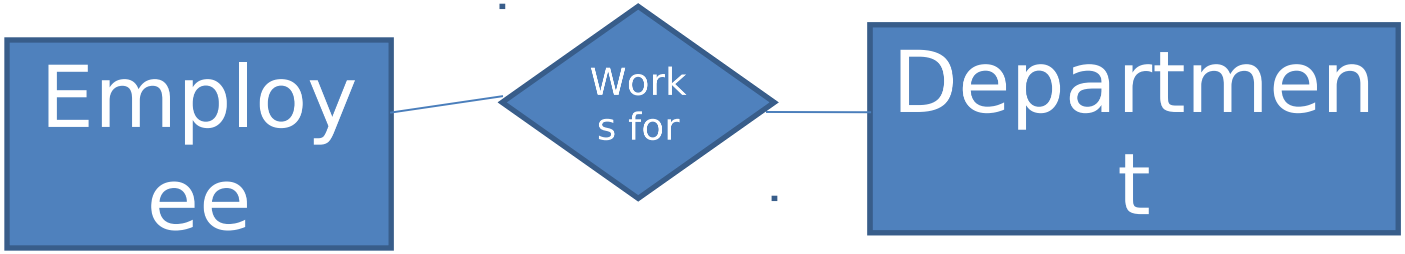
- Requirement Analysis
- Conceptual database design
- Logical database design
- Schema refinement
- Physical database design
- Application and security design

E -R diagrams

- The E-R data model allows us to describe the data involved in a real-world enterprise in terms of **objects and their relationships** and is widely used to develop an initial database design.

Entities

- An entity is an object in the real world that is distinguishable from other objects.
- An **Entity** can be any object, place, person or class.
- an **entity** is represented using rectangles.



Weak entity

Weak entity is an entity that depends on another entity. Weak entity doesn't have key attribute of their own. Double rectangle represents weak entity.

attributes

- An **Attribute** describes a property or characteristic of an entity. For example, Name, Age, Address etc can be attributes of a Student. An attribute is represented using eclipse

Types of attributes

- Key attribute
- Composite attributes
- Derived attributes

Entity sets

relationships

- A Relationship describes relations between **entities**. Relationship is represented using diamonds
- Relationship is an association among 2 or more entities
- There are three types of relationship that exist between Entities.
- Binary Relationship
- Recursive Relationship
- Ternary Relationship

relationship sets

- Collection of set of similar relationships is called a relationship set.

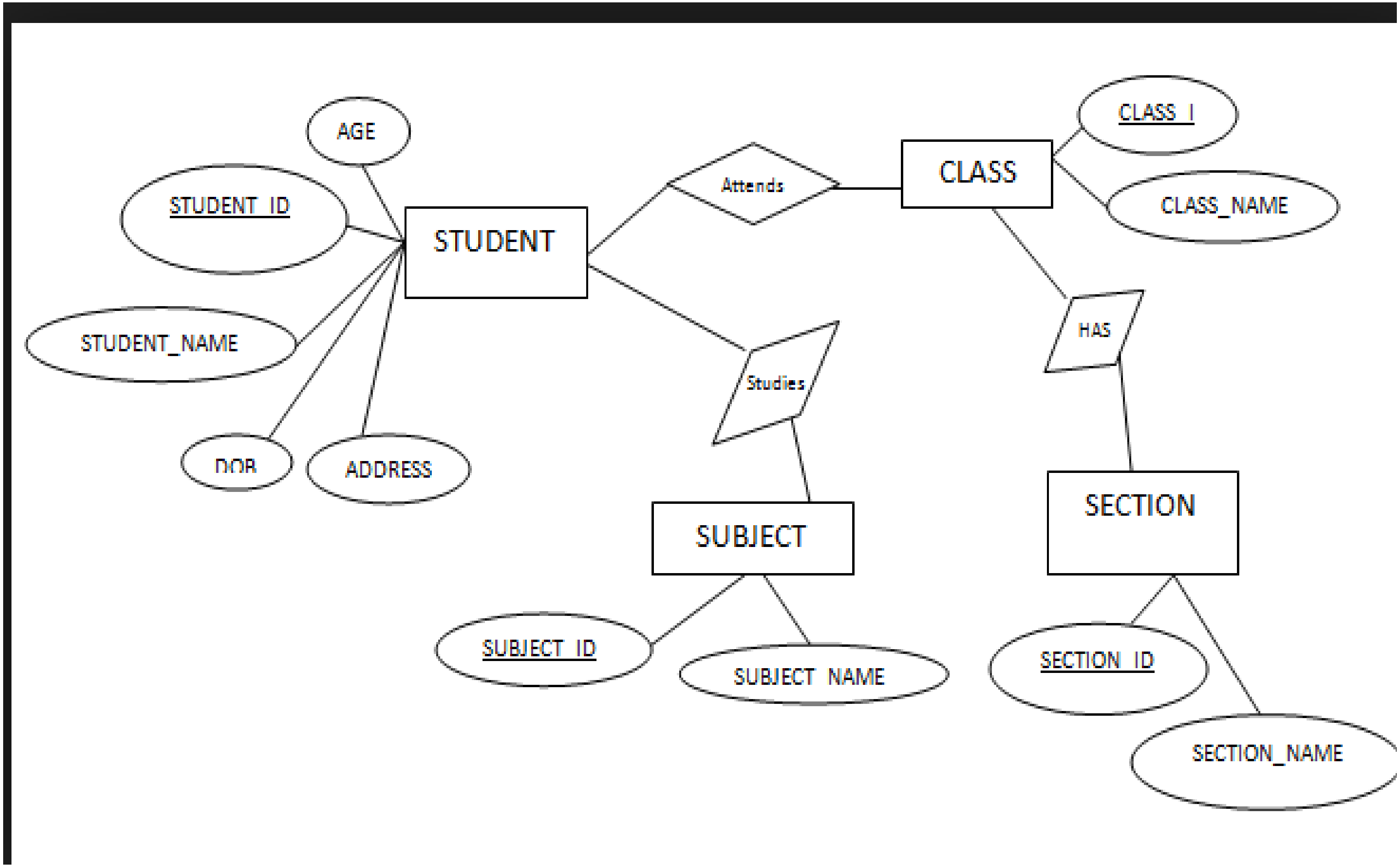
additional features of the E -R model

- Key constraints
- Key constraints for ternary relationships
- Participation constraints(partial and complete)
- Class hierarchies
- Aggregation

conceptual design with the E-R model

- Entity versus attribute
- Entity versus relationship
- Binary versus ternary relationships
- Aggregation versus ternary relationships

conceptual design for large enterprises


AGE
STUDENT_ID
STUDENT_NAME
DOB
ADDRESS
STUDENT
Attends
CLASS
CLASS_I
CLASS_NAME
Studies
HAS
SUBJECT
SECTION
SUBJECT_ID
SUBJECT_NAME
SECTION_ID
SECTION_NAME

- A college contains many departments
- Each department can offer any number of courses
- Many instructors can work in a department
- An instructor can work only in one department
- For each department there is a Head
- An instructor can be head of only one department
- Each instructor can take any number of courses
- A course can be taken by only one instructor
- A student can enroll for any number of

- Step 1 : Identify the Entities
- Stem 2 : Identify the relationships
- Step 3: Identify the key attributes
- Step 4: Identify other relevant attributes
- Step 5: Draw complete ER diagram

- Step 1 : Identify the Entities

What are the entities here?

From the statements given, the entities are

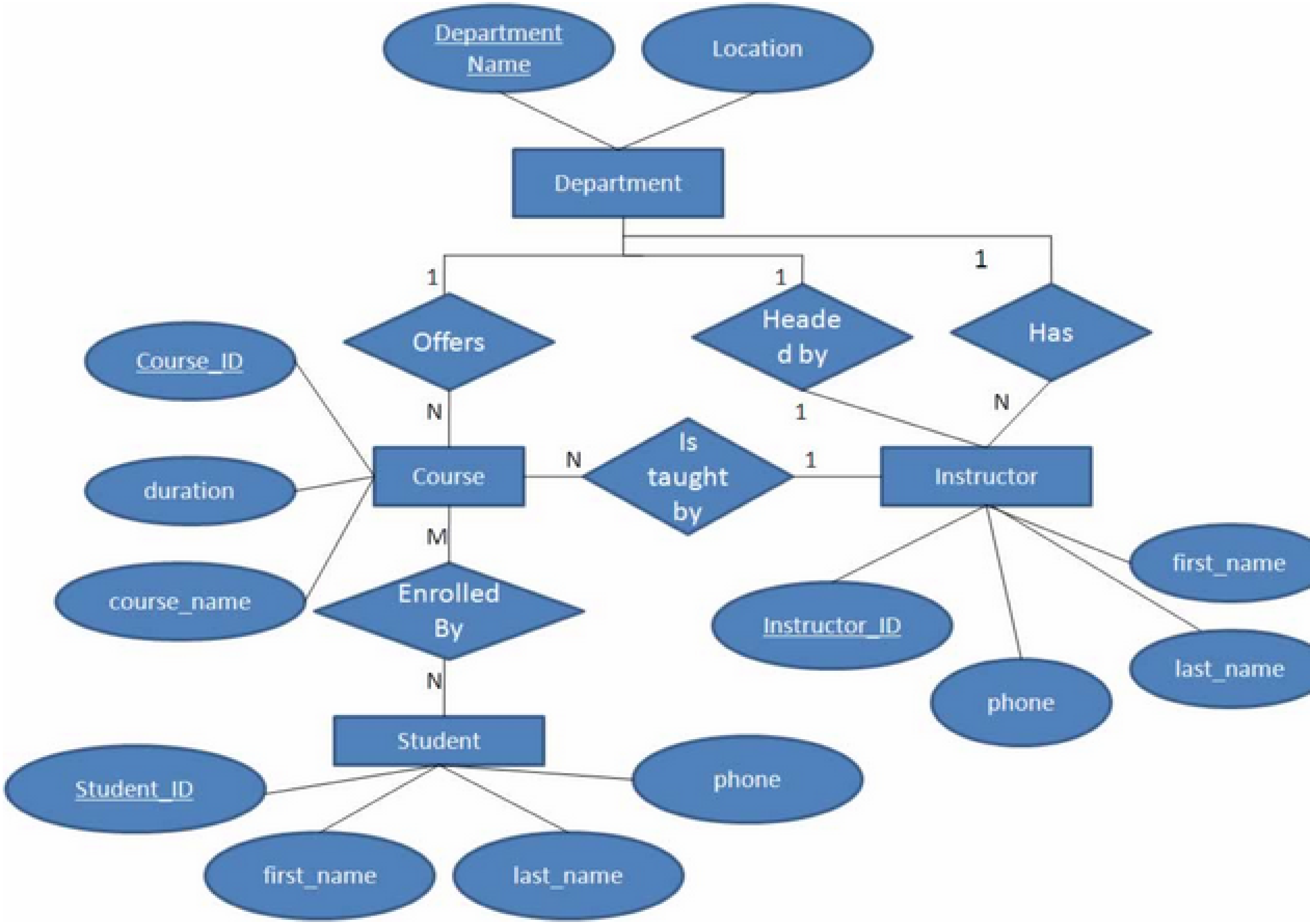
- Department
- Course
- Instructor
- Student

Stem 2 : Identify the relationships

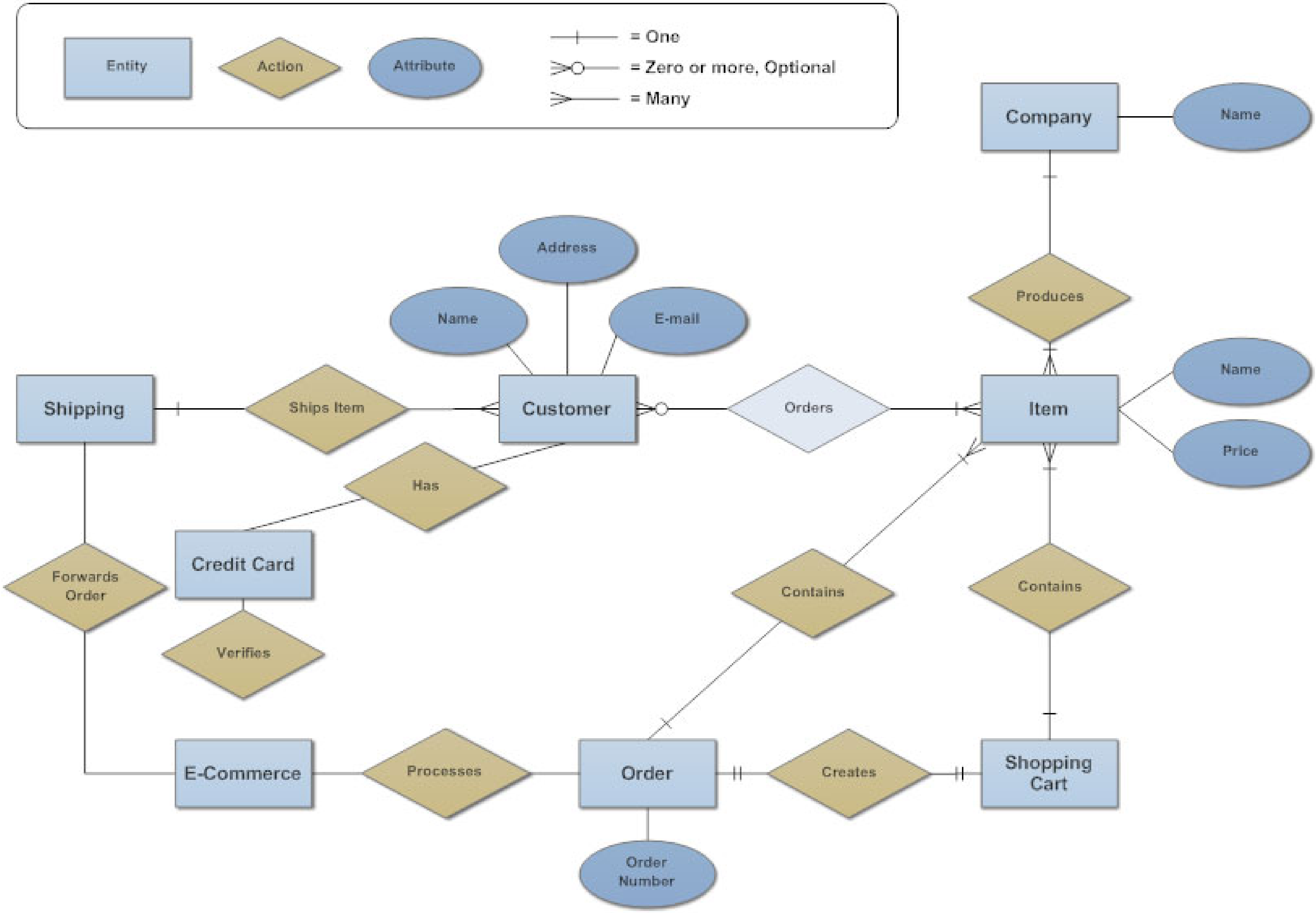
- One department offers many courses. But one particular course can be offered by only one department. hence the cardinality between department and course is **One to Many (1:N)**
- One department has multiple instructors . But instructor belongs to only one department. Hence the cardinality between department and instructor is **One to Many (1:N)**
- One department has only one head and one head can be the head of only one department. Hence the cardinality is one to one. (1:1)
- One course can be enrolled by many students and one student can enroll for many courses. Hence the cardinality between course and student is **Many to Many (M:N)**
- One course is taught by only one instructor. But one instructor teaches many courses. Hence the cardinality between course and instructor is **Many to**

- Step 3: Identify the key attributes
- "Departmen_Name" can identify a department uniquely. Hence Department_Name is the key attribute for the Entity "Department".
- Course_ID is the key attribute for "Course" Entity.
- Student_ID is the key attribute for "Student" Entity.
- Instructor_ID is the key attribute for "Instructor" Entity.

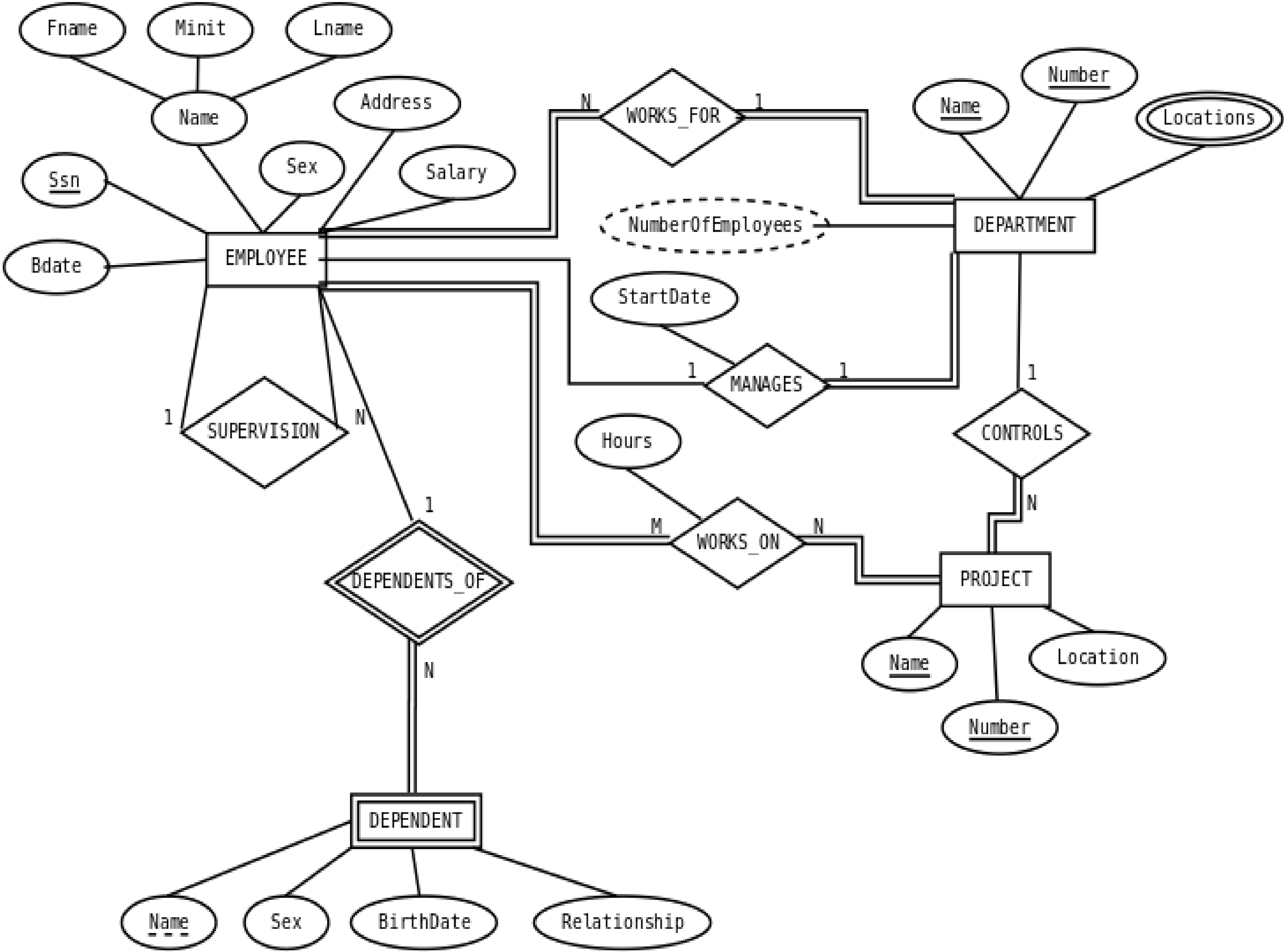
- Step 4: Identify other relevant attributes
- For the department entity, other attributes are location
- For course entity, other attributes are course_name,duration
- For instructor entity, other attributes are first_name, last_name, phone
- For student entity, first_name, last_name, phone

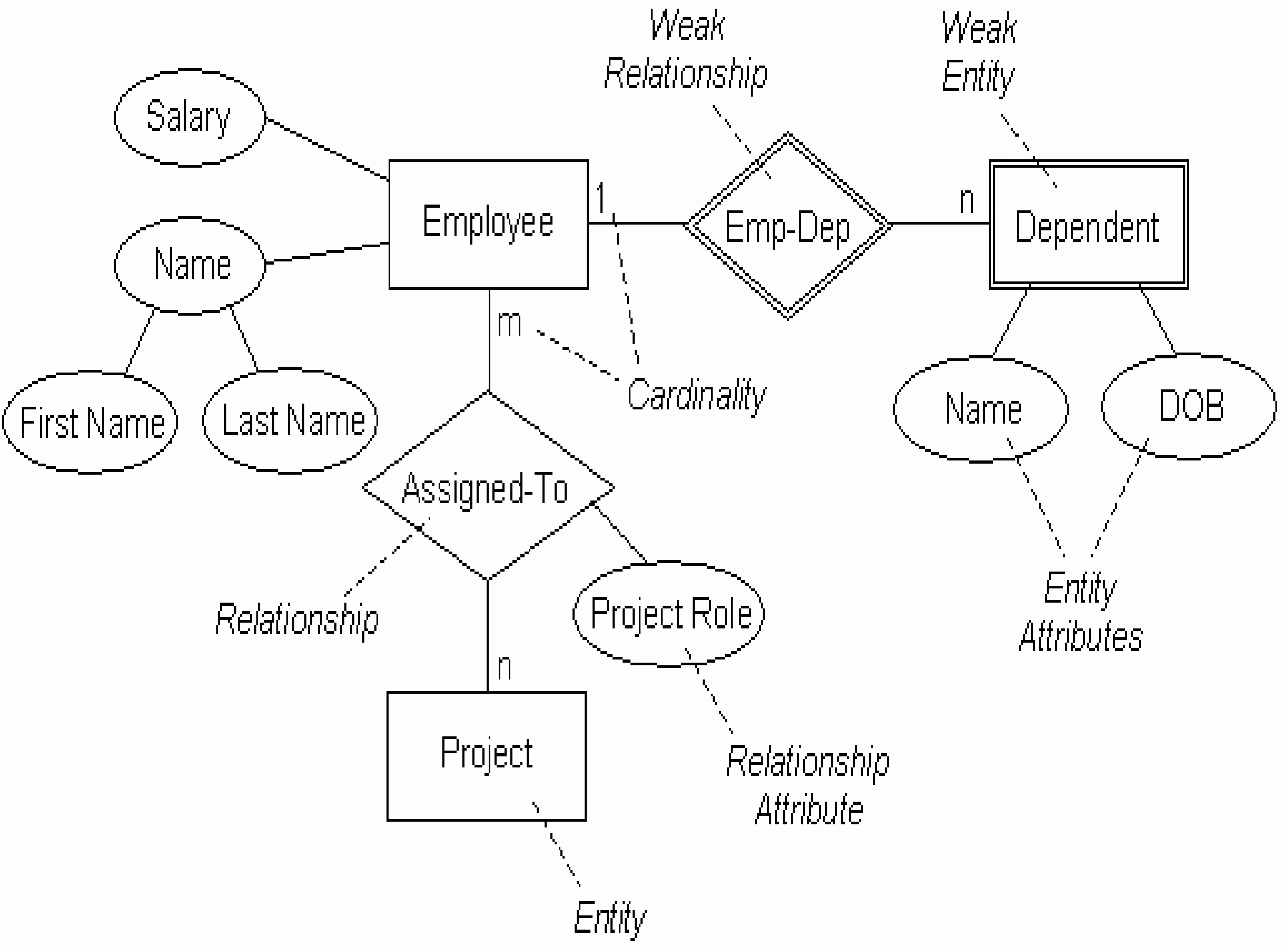

Department Name
Location
Department
1
1
1
Offers
Heade d by
Has
Course_ID
N
1
N
duration
Course
N
Is taught by
1
Instructor
M
course_name
Enrolled By
first_name
Instructor_ID
last_name
phone
N
Student
Student_ID
phone
first_name
last_name

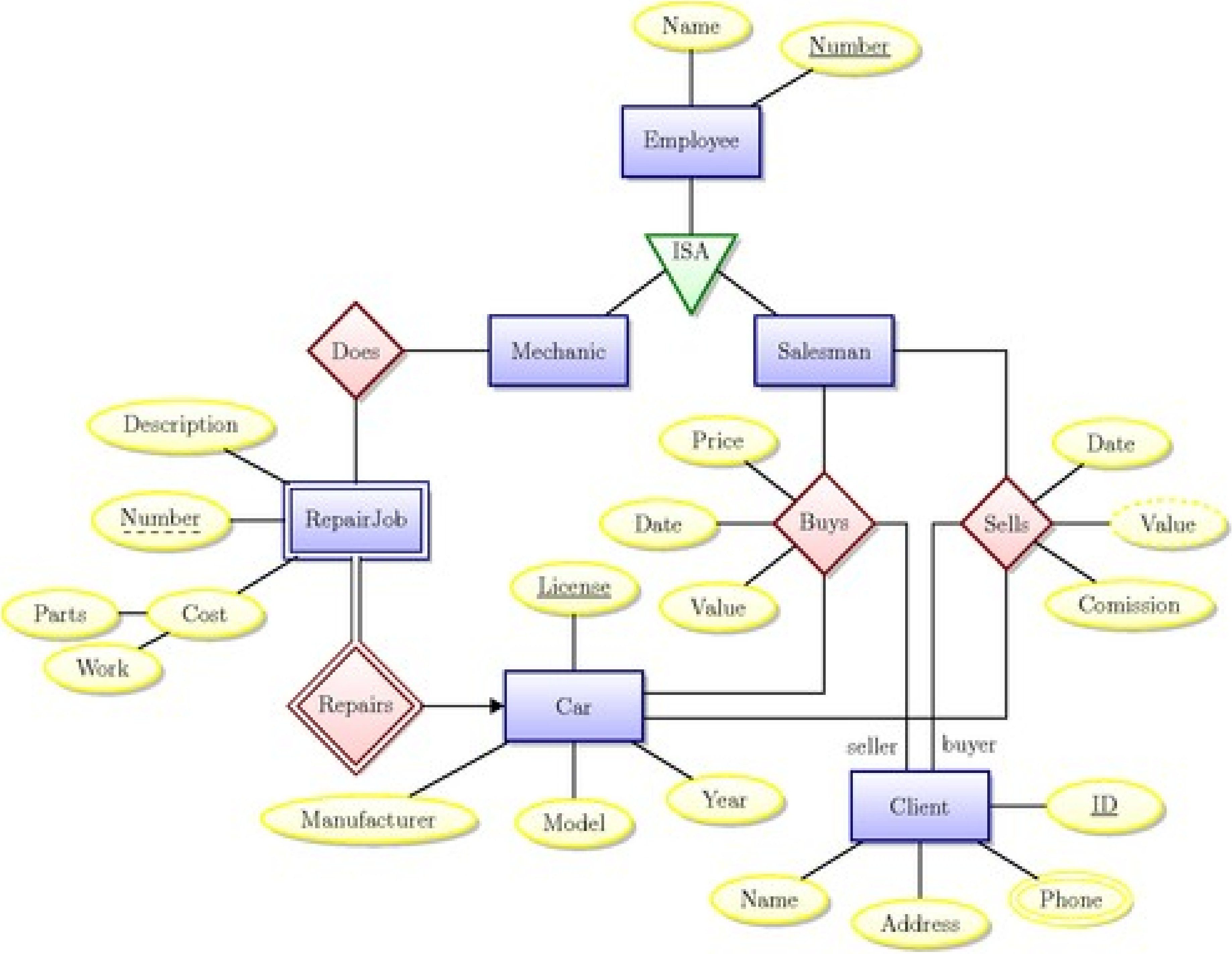
Entity Relationship Diagram - Internet Sales Model

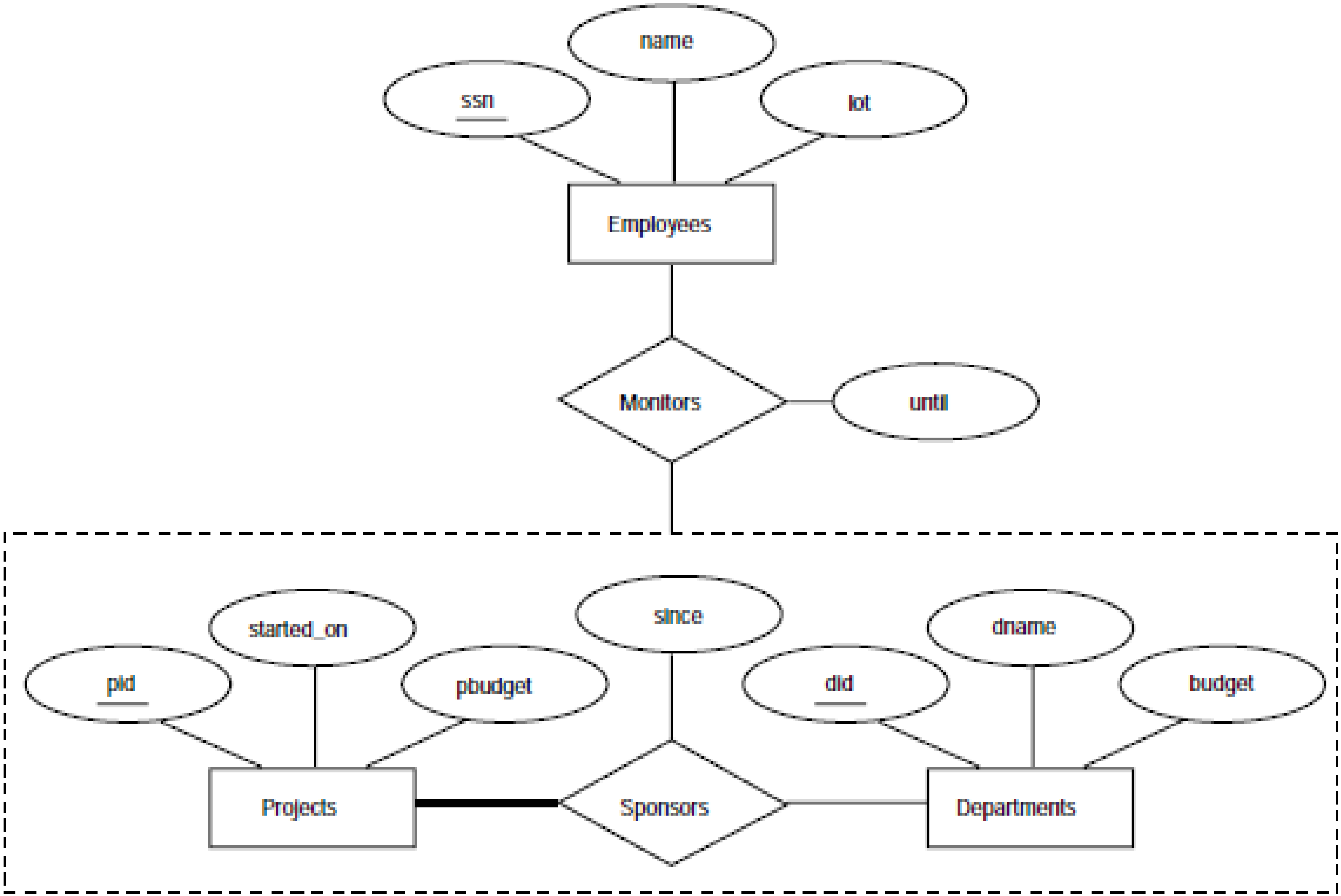


- A typical example could be entities Customer, Order, and Product.
- An instance of the Customer entity is identified by a unique customer number,
- an instance of the Order entity is identified by a unique order number, and
- an instance of the Product entity is identified by a unique product


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Name
Number
Locations
NumberOfEmployees
DEPARTMENT
StartDate
1
MANAGES
1
1
1
SUPERVISION
N
CONTROLS
Hours
1
N
M
WORKS_ON
N
DEPENDENTS_OF
PROJECT
N
Name
Location
Number
DEPENDENT
Name
Sex
BirthDate
Relationship


Weak Relationship
Weak Entity
Salary
Employee
1
Emp-Dep
n
Dependent
Name
m
First Name
Last Name
Cardinality
Name
DOB
Assigned-To
Relationship
Project Role
Entity Attributes
n
Project
Relationship Attribute
Entity


Name
Number
Employee
ISA
Does
Mechanic
Salesman
Description
Number
RepairJob
Parts
Cost
Work
Repairs
License
Car
Manufacturer
Model
Year
Price
Date
Buys
Value
Date
Sells
Value
Comission
seller
buyer
Client
ID
Name
Address
Phone


name
ssn
lot
Employees
Monitors
until
started_on
since
dname
pid
pbudget
did
budget
Projects
Sponsors
Departments