CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE &ENGINNERING

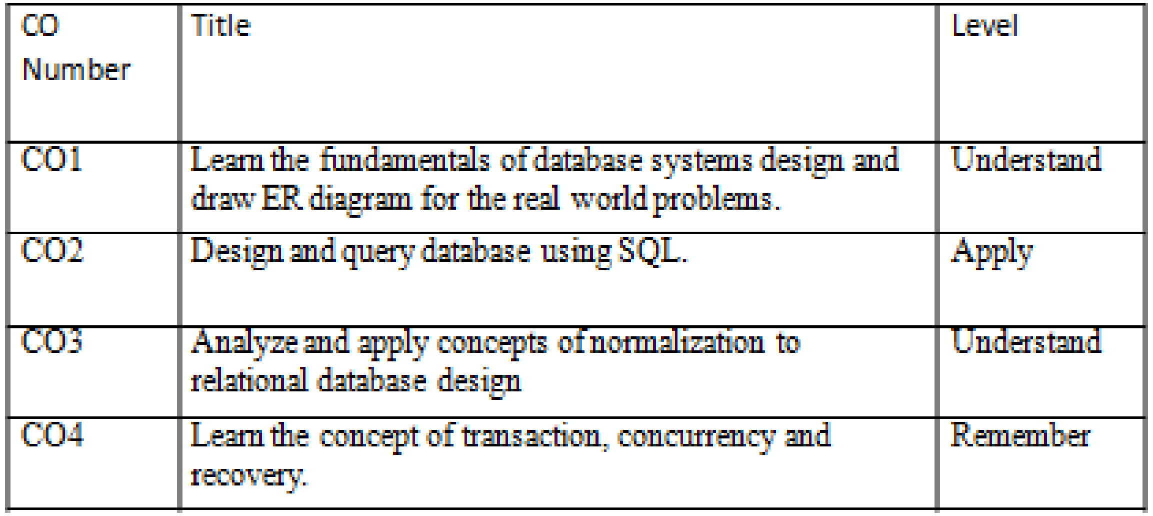
BE: CSE

Database Management System and CST-227

Prepared By: Ms. Heena Arora

Database Management System

Course Outcome



CO Number	Title	Level
CO1	Learn the fundamentals of database systems design and draw ER diagram for the real world problems.	Understand
CO2	Design and query database using SQL.	Apply
CO3	Analyze and apply concepts of normalization to relational database design	Understand
CO4	Learn the concept of transaction, concurrency and recovery.	Remember

Database Management System

Data Models

Different types of Models

ER Model

Types of relationship

Comparison among different models

Data Model

Definition: precise description of the data content in a system

Categories of data models:

1. Conceptual(high-level, semantic): describes WHAT the system contains
2. Logical(low-level, internal): describes HOW the system will be implemented, regardless of the DBMS
3. Physical(representational): describes HOW the system will be implemented using a specific DBMS

Categories of Data Models

- Object Based Data Models: Object based data models use concepts such as entities, attributes, and relationships.
    - Entity Relationship
    - Object Oriented
    - Semantic
    - Functional
- Physical Data Models
    - Unifying Model
- Record Based Data Models: Record based logical models are used in describing data at the logical and view levels
    - Hierarchical
    - Network
    - Relational

Need of Data Models

To aid in the development of a sound database design that does not allow anomalies or inconsistencies

Goal: to create database tables that do not contain duplicate data values that can become inconsistent

Schema Vs Instances

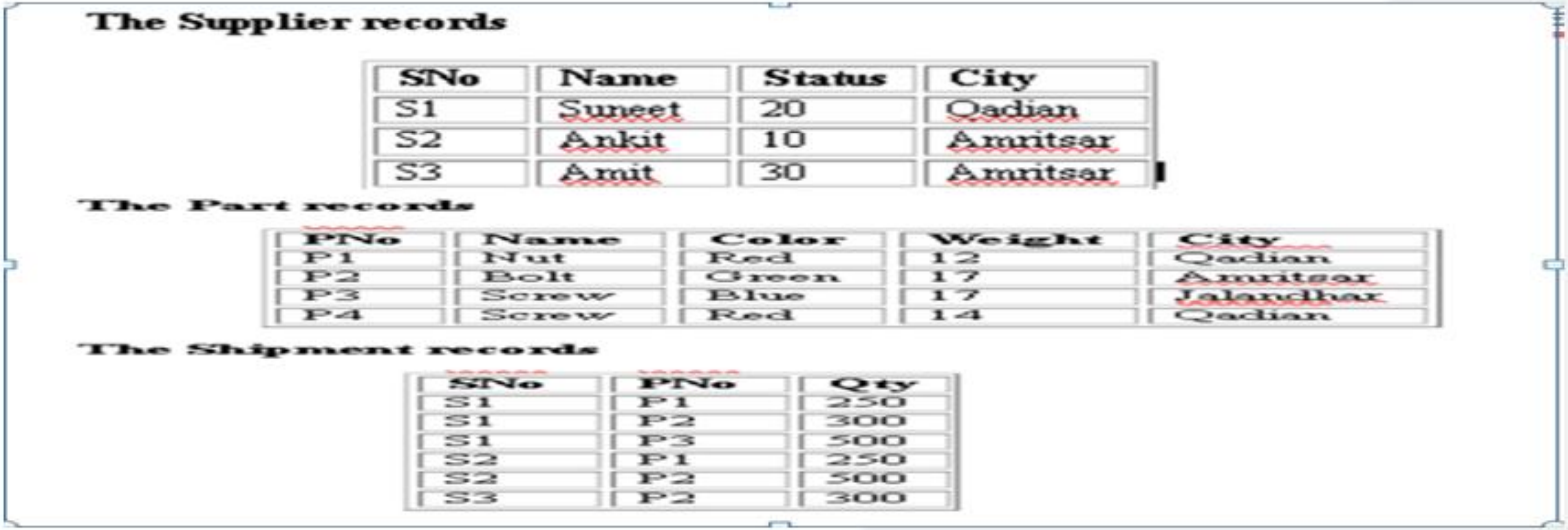
- **Database Schema**: The *description* of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
- **Schema Diagram:** A diagrammatic display of (some aspects of) a database schema.
- **Database Instance:** The actual data stored in a database at a *particular moment in time*. Also called **database state** (or **occurrence**).

Schema Vs State

- **Database State:** Refers to the content of a database at a moment in time.
- **Initial Database State:** Refers to the database when it is loaded
- **Valid State:** A state that satisfies the structure and constraints of the database.

- **Distinction**
  - The **database schema** changes *very infrequently*. The **database state** changes *every time the database is updated.*
  - **Schema** is also called **intension**, whereas **state** is called **extension**.

Sample Database



The Supplier records

SNo	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	30	Amritsar

The Part records

PNo	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

SNo	PNo	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

Figure: 1:Sample Database

Hierarchical Model

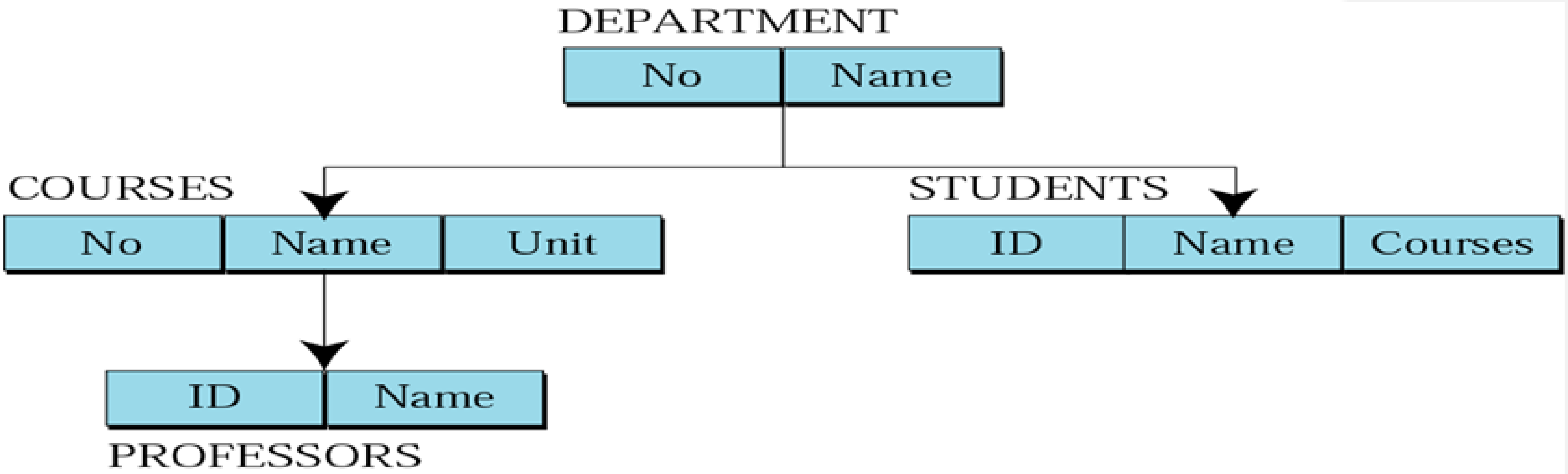
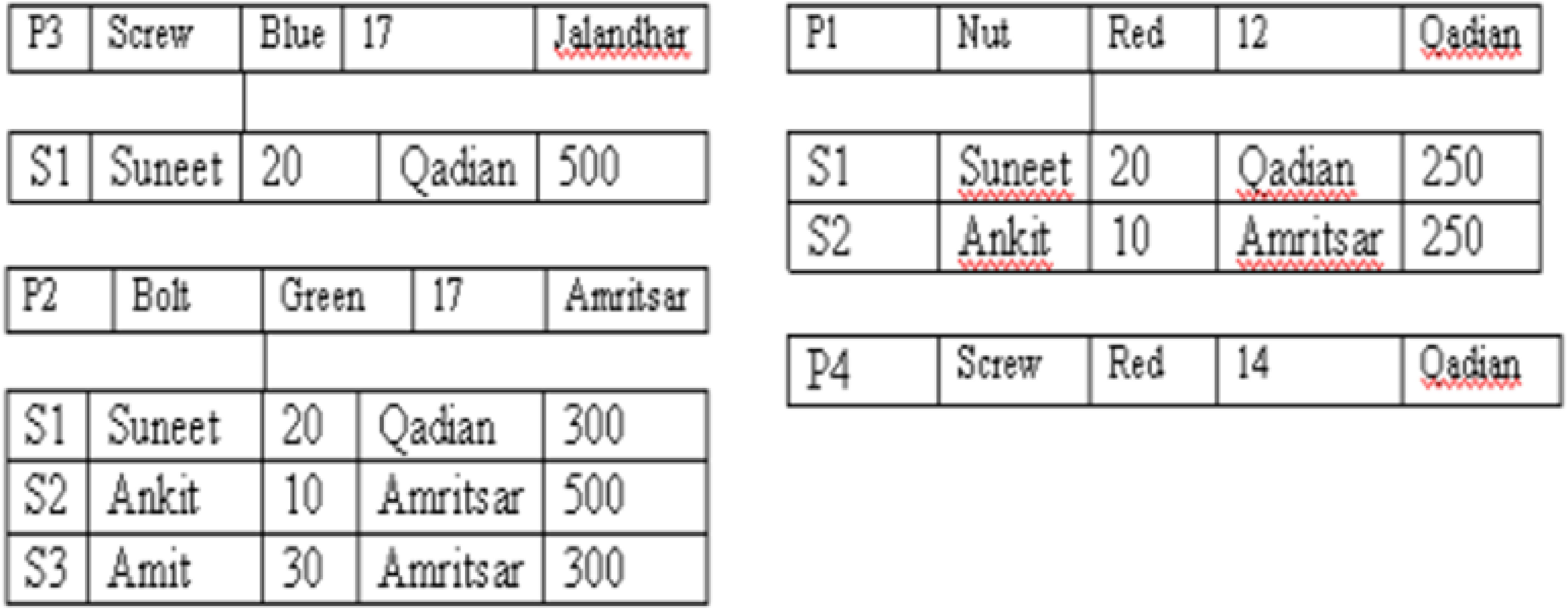


Figure: 2:Hierarchial Model

Equivalent Hierarchical Model



P3	Screw	Blue	17	Jalandhar

S1	Suneet	20	Qadian	500

P2	Bolt	Green	17	Amritsar

S1	Suneet	20	Qadian	300
S2	Ankit	10	Amritsar	500
S3	Amit	30	Amritsar	300

P1	Nut	Red	12	Qadian

S1	Suneet	20	Qadian	250
S2	Ankit	10	Amritsar	250

P4	Screw	Red	14	Qadian

Figure3:Equivalent Hierarchical Model

Advantages or Disadvantages

- Advantages
  - Many of the hierarchical data model's features formed the foundation for current data models
  - Its database application advantages are replicated, implemented in a different form, in current database environments
  - Generated a large installed (mainframe) base, created a pool of programmers who developed numerous tried-and-true business applications
- Disadvantages
  - Complex to implement
  - Difficult to manage
  - Lacks structural independence
  - Implementation limitations
  - Lack of standards

Network Model

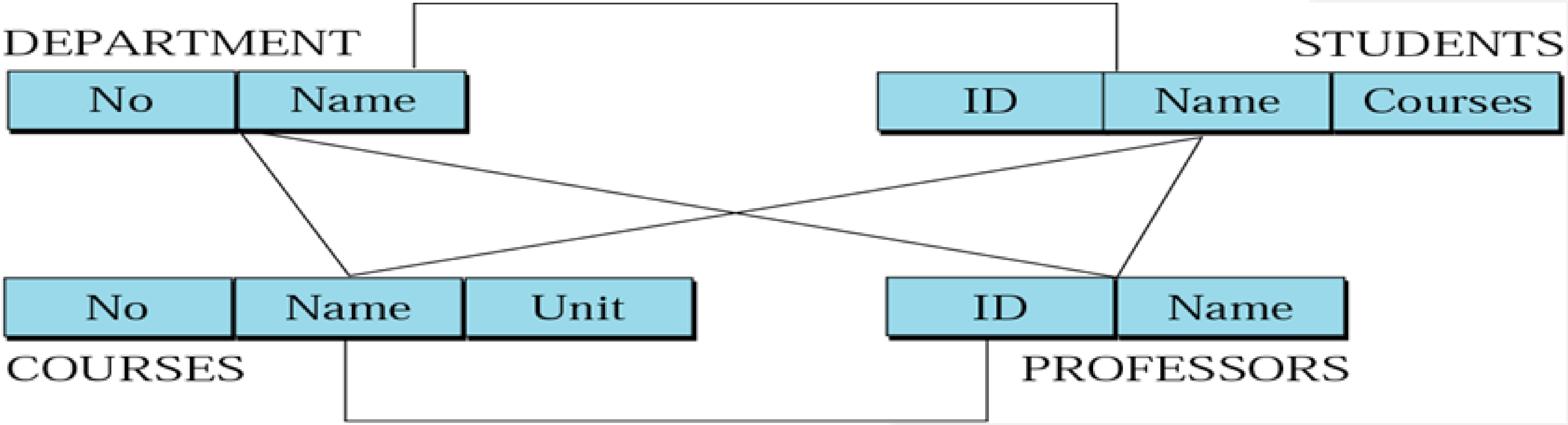


Figure: 4:Network Model

Equivalent Network Model

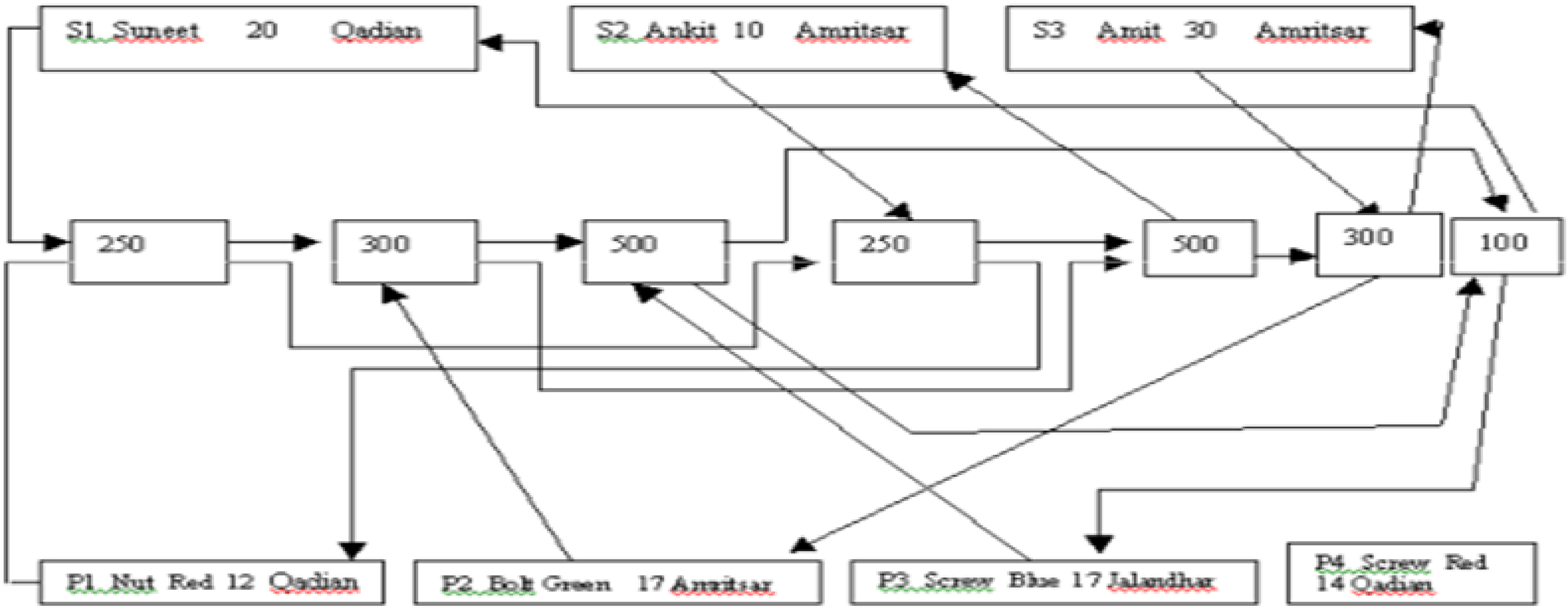
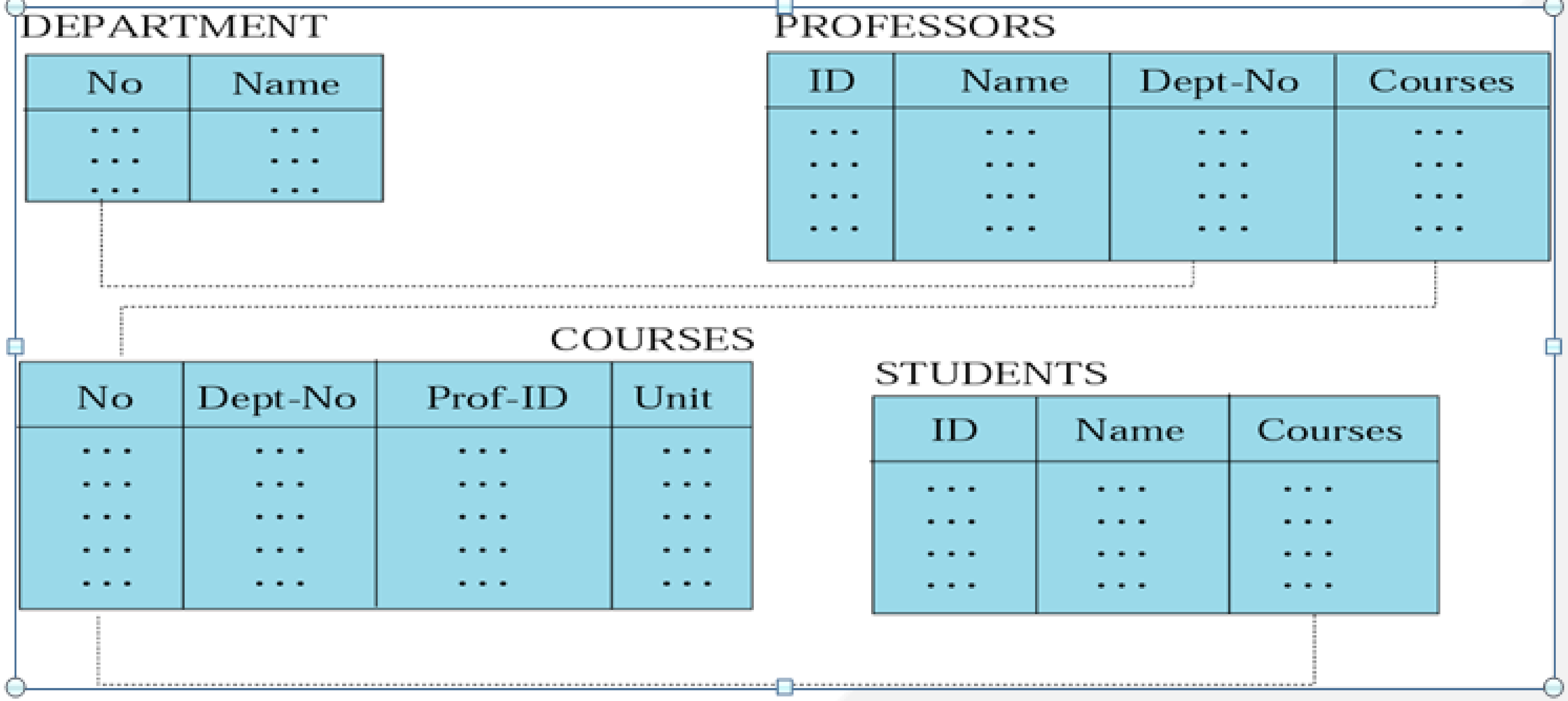


Figure: 5: Equivalent Network Model

Advantages or Disadvantages

- Advantages
  - It helps to make relationship between data efficiently.
  - It helps to make database performance better.
- Disadvantages
  - Very difficult to implement it.
  - Take lots of time to implement it.

Relational Model



DEPARTMENT

No	Name
...	...
...	...
...	...

PROFESSORS

ID	Name	Dept-No	Courses
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

COURSES

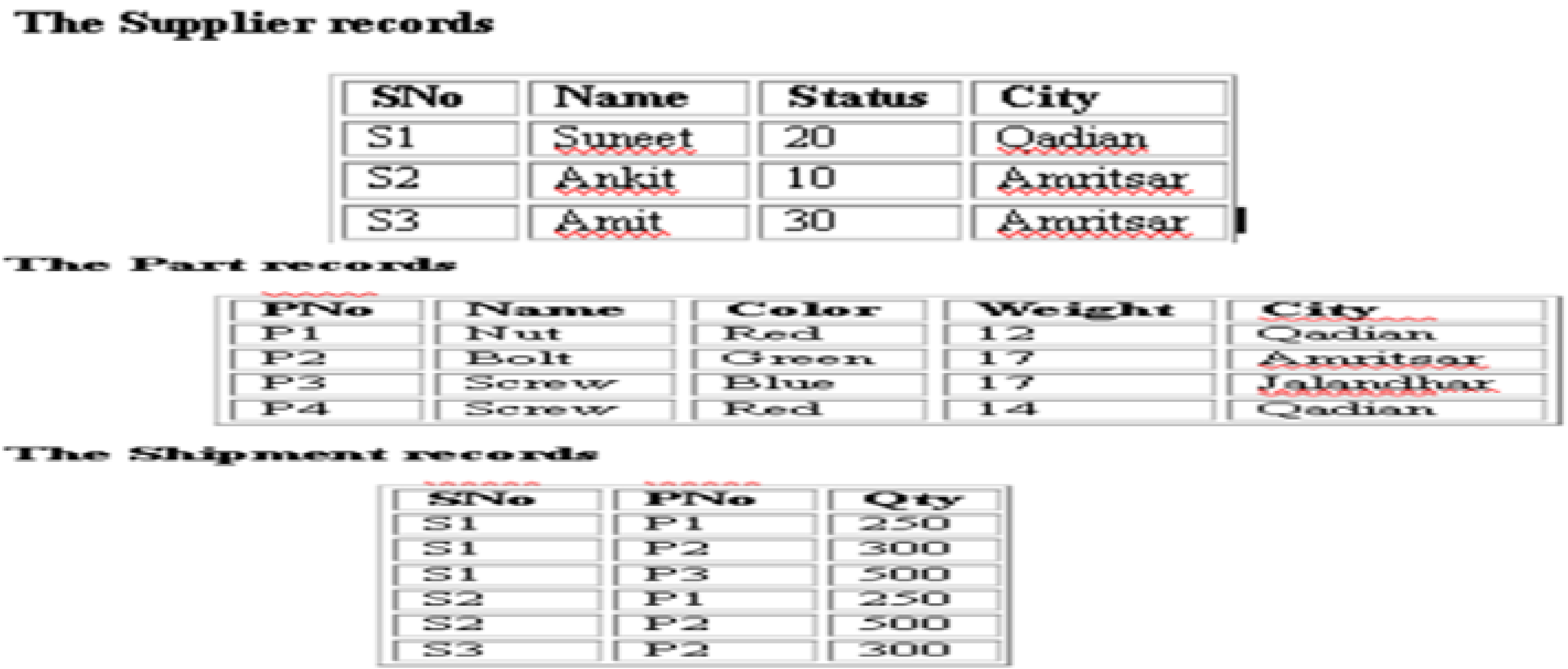
No	Dept-No	Prof-ID	Unit
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

STUDENTS

ID	Name	Courses
...	...	...
...	...	...
...	...	...
...	...	...

Figure: 6:Relational Model

Equivalent Relational Model



The Supplier records

SNo	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	30	Amritsar

The Part records

PNo	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

SNo	PNo	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

Figure: 7: Equivalent Relational Model

Relational Database

- A database whose logical organization is based on relational data model is a **Relational Database**

Relational Model

The main highlights of this model are −

- Data is stored in tables called **relations**.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from a same domain.

E-R Model

- Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.
- ER Model is best used for the conceptual design of a database.

Entities and Relationships

ER Model is based on −

Entities and their *attributes.*

Relationships among entities.

- **Entity** − An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
- **Relationship** − The logical association among entities is called ***relationship***. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.
- An **entity set** is a set of entities of the same type that share the same properties.

Example: set of all persons, companies, trees, holidays

Types of Attributes

Types of Attributes

- **Simple attribute** – Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- **Composite attribute** – Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- **Derived attribute** – Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department should not be saved directly in thedatabase, instead it can be derived. For another example, age can be derived from data_of_birth etc.
- **Single-value attribute** – Single-value attributes contain single value. For example – Social_Security_Number.
- **Multi-value attribute** – Multi-value attributes may contain more than one values. For example, a person can have more than one phone

Continued

Types of Attributes

- **Single-value attribute** − Single-value attributes contain single value. For example − Social Security Number.
- **Multi-value attribute** − Multi-value attributes may contain more than one values. For example, a person can have more than one phone

Mapping Cardinalities

Relationships: Link between different entities of the database is called relationship.

Mapping Cardinalities

Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.

Types of Relationships

One - One Relationship:- (1 – 1)

Each value in the first table may relate with only one record in the second table.

One – Many Relationship:- (1 - ∞)

Every value in the first table may relate with many records in the second table.

Many – Many Relationship (∞ - ∞)

Each value in the first table could relate with many records in the second table and each value of the second table could relate with many records in the first table.

Many – One Relationship(∞-1)

More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

E-R Diagrams

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
  - Double ellipses represent multivalued attributes.
  - Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes

Example of ER Diagram

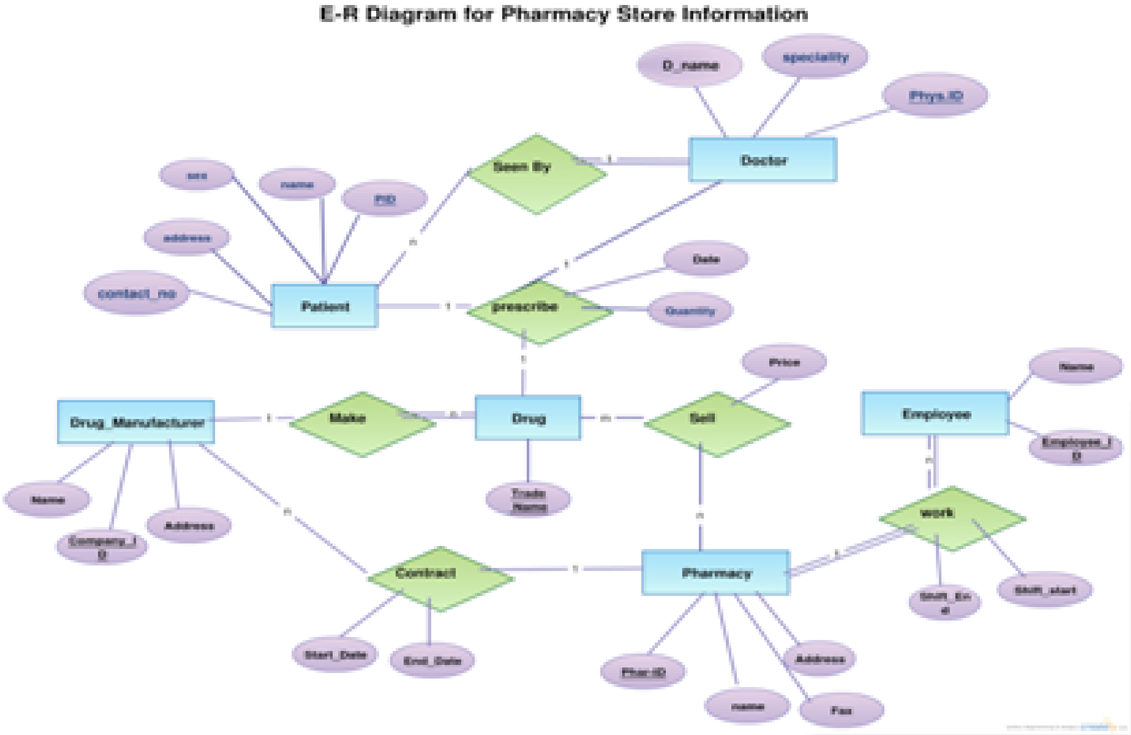


Figure-8:Example of ER Diagram

(https://in.images.search.yahoo.com/search/images;_ylt=AwrwJUW4qxVd1hkAbuq9HAx.;_ylu=X3oDMTBsZ
29xY3ZzBHNlYwNzZWFyY2gEc2xrA2J1dHRvbg-)

Design Issues

- **Use of entity sets vs. attributes**
  Choice mainly depends on the structure of the enterprise being modeled, and on the semantics associated with the attribute in question.
- **Use of entity sets vs. relationship sets**
  Possible guideline is to designate a relationship set to describe an action that occurs between entities
- **Binary versus n-ary relationship sets**
  Although it is possible to replace any nonbinary (*n*-ary, for $n > 2$) relationship set by a number of distinct binary relationship sets, a *n*-ary relationship set shows more clearly that several entities participate in a single relationship.

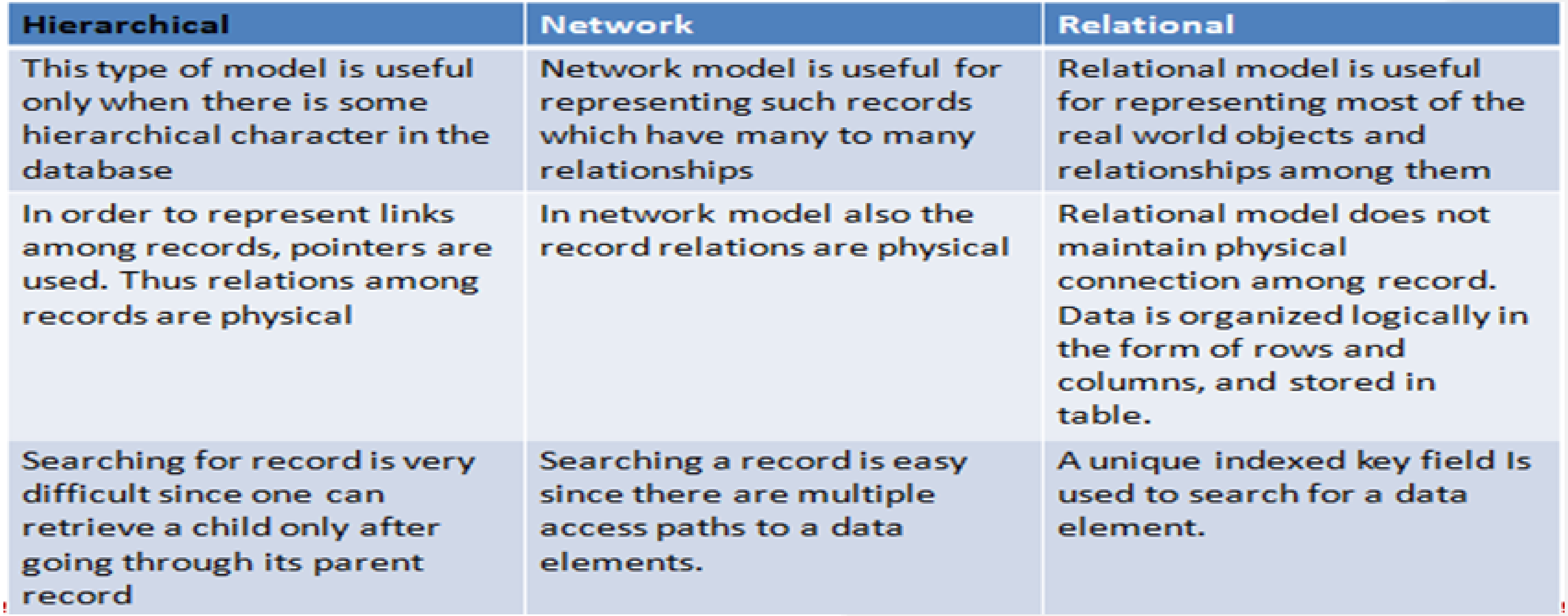
Techniques Used For Data Abstraction

- Specialization- In specialization, an entity is divided into sub-entities based on their characteristics. It is a top-down approach where higher level entity is specialized into two or more lower level entities.
- Generalization- Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it.
- Aggregation-In this a relationship with its corresponding entities is aggregated into a higher level entity

Comparison of Record Based Models

Hierarchical	Network	Relational
This type of model is useful only when there is some hierarchical character in the database	Network model is useful for representing such records which have many to many relationships	Relational model is useful for representing most of the real world objects and relationships among them
In order to represent links among records, pointers are used. Thus relations among records are physical	In network model also the record relations are physical	Relational model does not maintain physical connection among record. Data is organized logically in the form of rows and columns, and stored in table.
Searching for record is very difficult since one can retrieve a child only after going through its parent record	Searching a record is easy since there are multiple access paths to a data elements.	A unique indexed key field Is used to search for a data element.

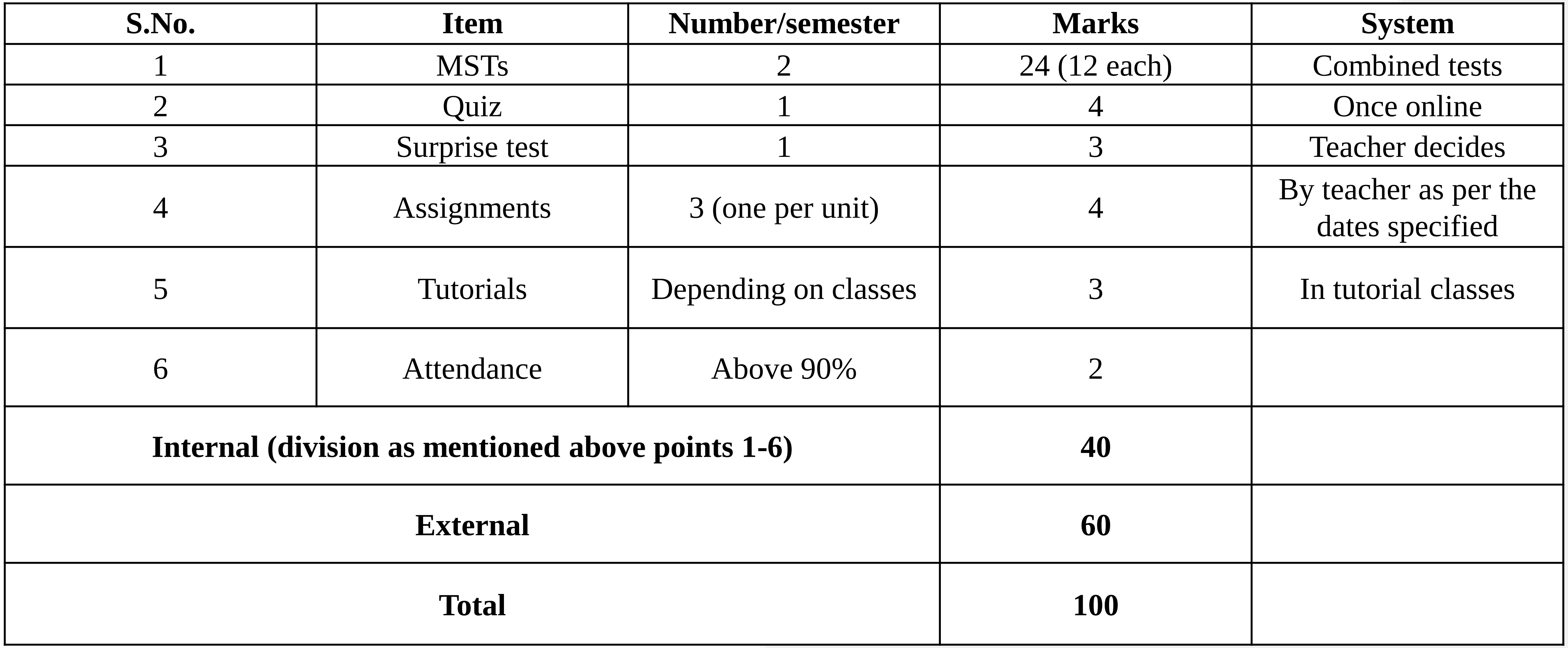
Continued



Hierarchical	Network	Relational
This type of model is useful only when there is some hierarchical character in the database	Network model is useful for representing such records which have many to many relationships	Relational model is useful for representing most of the real world objects and relationships among them
In order to represent links among records, pointers are used. Thus relations among records are physical	In network model also the record relations are physical	Relational model does not maintain physical connection among record. Data is organized logically in the form of rows and columns, and stored in table.
Searching for record is very difficult since one can retrieve a child only after going through its parent record	Searching a record is easy since there are multiple access paths to a data elements.	A unique indexed key field Is used to search for a data element.

Figure:9:Comparison of Record Based Models (https://in.images.search.yahoo.com/search/images;_ylt=AwrPg1DYsxVd4w0ANYK7HAx)

Assessment Pattern



S.No.	Item	Number/semester	Marks	System
1	MSTs	2	24 (12 each)	Combined tests
2	Quiz	1	4	Once online
3	Surprise test	1	3	Teacher decides
4	Assignments	3 (one per unit)	4	By teacher as per the dates specified
5	Tutorials	Depending on classes	3	In tutorial classes
6	Attendance	Above 90%	2	
Internal (division as mentioned above points 1-6)			**40**	
External			**60**	
Total			**100**	

REFERENCES

- Elmasri and Shamkant B. Navathe, “Fundamentals of Database System”, The Benjamin / Cummings Publishing Co.
- Korth and Silberschatz Abraham, “Database System Concepts”, McGraw Hall.
- C.J.Date, “An Introduction to Database Systems”, Addison Wesley.
- Thomas M. Connolly, Carolyn & E. Begg, “Database Systems: A Practical Approach to Design, Implementation and Management”, 5/E, University of Paisley, Addison-Wesley.

THANK YOU

For queries

Email: Heenae7725@cumail.in