AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING

DEPARTMENT OF COMUTER SCIENCE AND ENGINEERING

R2017 - SEMESTER V

OPEN ELECTIVE I: OCE552 - GEOGRAPHICAL INFORMATION SYSTEM

UNIT II - SPATIAL DATA MODELS

SYLLABUS

UNIT II SPATIAL DATA MODELS 9

Database Structures – Relational, Object Oriented – ER diagram - spatial data models – Raster Data Structures – Raster Data Compression - Vector Data Structures - Raster vs Vector Models- TIN and GRID data models - OGC standards - Data Quality.

Table of Contents

2.1 INTRODUCTION TO DATA STRUCTURES

2.1.1 Introduction to Data Structure

- A data structure is a way of storing data in a computer so that it can be used efficiently and it will allow the most efficient algorithm to be used.

- A data structure should be seen as a logical concept that must address two fundamental concerns.

  I. First, how the data will be stored, and
  II. Second, what operations will be performed on it.

What are Databases?

- To find out what database is, we have to start from data, which is the basic building block of any DBMS.
- Data: Facts, figures, statistics etc. having no particular meaning (e.g. 1, ABC, 19 etc).
- Record: Collection of related data items,
- e.g. in the above example the three data items had no meaning.
- But if we organize them in the following way, then they collectively represent meaningful information.

Table or Relation: Collection of related records.

Roll	Name	Age
1	ABC	19

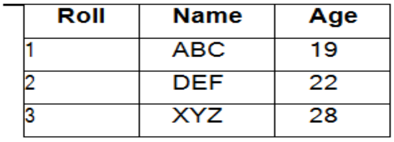
Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

The columns of this relation are called Fields, Attributes or Domains. The rows are called Tuples or Records.

Database: Collection of related relations.

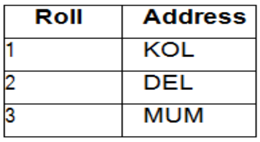
Consider the following collection of tables:

T1



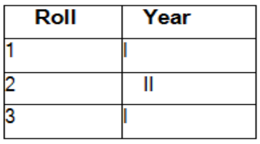
Roll	Name	Age
1	ABC	19
2	DEF	22
3	XYZ	28

T2



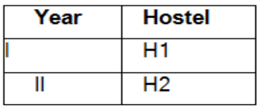
Roll	Address
1	KOL
2	DEL
3	MUM

T3



Roll	Year
1	I
2	II
3	I

T4



Year	Hostel
I	H1
II	H2

- We now have a collection of 4 tables.
- They can be called a "related collection" because we can clearly find out that there are some common attributes existing in a selected pair of tables.
- Because of these common attributes we may combine the data of two or more tables together to find out the complete details of a student.

Questions like "Which hostel does the youngest student live in?" can be answered now, although Age and Hostel attributes are in different tables.

What are Database Management Systems?

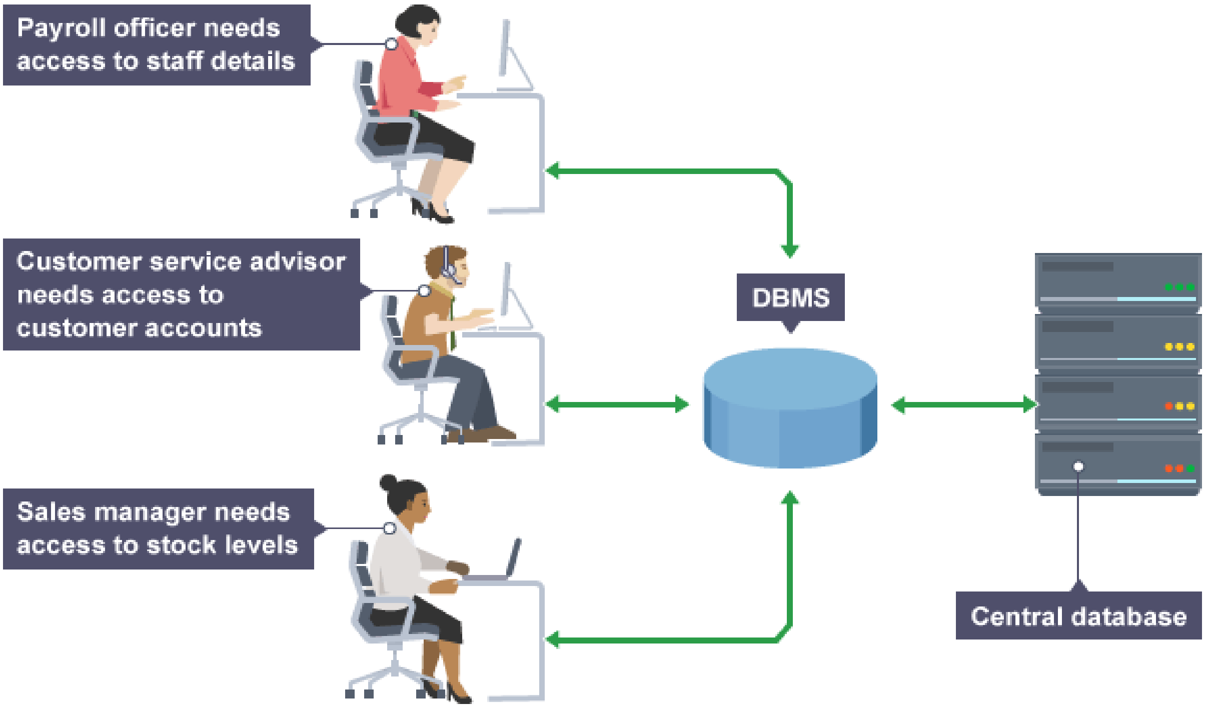
- Collection of interrelated data
- Set of programs to access the data
- DBMS contains information about a particular enterprise
- DBMS provides an environment that is both convenient and efficient to use.

Database Applications:

Banking: all transactions
Airlines: reservations, schedules
Universities: registration, grades
Sales: customers, products, purchases
Manufacturing: production, inventory, orders, supply chain
Human resources: employee records, salaries, tax deductions
Databases touch all aspects of our lives

- As the name suggests, the database management system consists of two parts. They are:
    1. Database and
    2. Management System
- A database in a DBMS could be viewed by lots of different people with different responsibilities.

Employees are accessing Data through DBMS


Payroll officer needs access to staff details
Customer service advisor needs access to customer accounts
Sales manager needs access to stock levels
DBMS
Central database

2.1.2 Purpose of Database System

- For example, within a company there are different departments, as well as customers, who each need to see different kinds of data.
- Each employee in the company will have different levels of access to the database with their own customized front-end application.
- In a database, data is organized strictly in row and column format. The rows are called Tuple or Record.
- The data items within one row may belong to different data types.
- On the other hand, the columns are often called Domain or Attribute.
- All the data items within a single attribute are of the same data type.

In the early days, database applications were built on top of file systems

Drawbacks of using file systems to store data:

a) **Data redundancy and inconsistency**
   Multiple file formats, duplication of information in different files

b) **Difficulty in accessing data**
   Need to write a new program to carry out each new task

c) **Data isolation — multiple files and formats**

d) **Integrity problems**

Integrity constraints (e.g. account balance > 0) become part of program code
Hard to add new constraints or change existing ones

e) **Atomicity of updates**

- Failures may leave database in an inconsistent state with partial updates carried out
- E.g. transfer of funds from one account to another should either complete or not happen at all

f) **Concurrent access by multiple users**

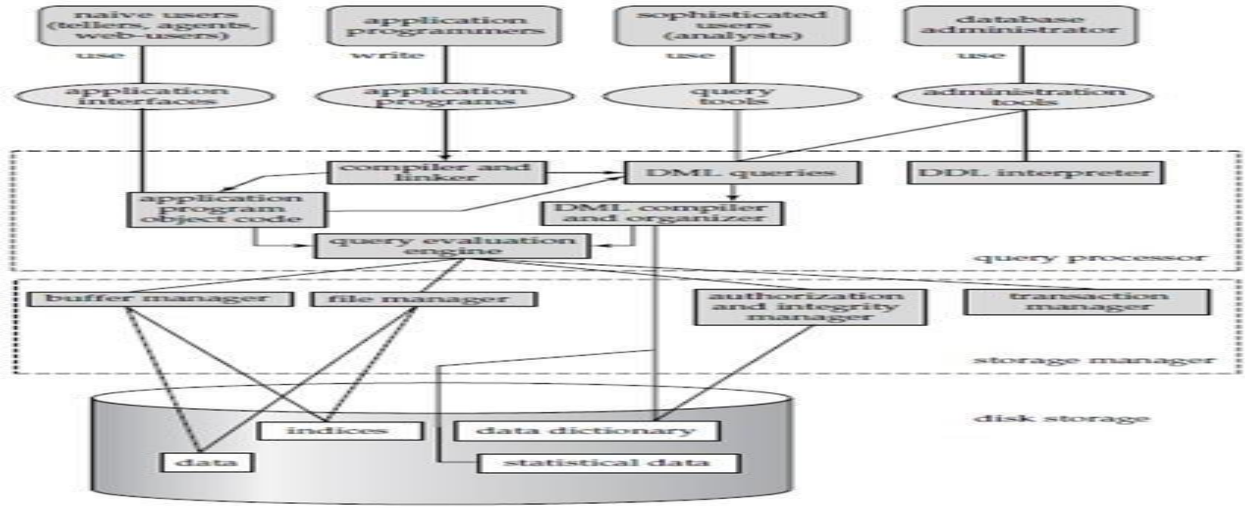
- Concurrent accessed needed for performance
- Uncontrolled concurrent accesses can lead to inconsistencies
- E.g. two people reading a balance and updating it at the same time

g) **Security problems**

Database systems offer solutions to all the above problems

2.1.3 Database Architecture

- Architecture provide a single picture of the various components of a database system and the connections among them.
- The architecture of a database system is greatly influenced by the underlying computer system on which the database system runs.
- Database systems can be centralized, or client-server, where one server machine executes work on behalf of multiple client machines.
- Database systems can also be designed to exploit parallel computer architectures.
- Distributed databases span multiple geographically separated machines.



- A database system is partitioned into modules that deal with each of the responsibilities of the overall system.
- The functional components of a database system can be broadly divided into the storage manager and the query processor components.
- The storage manager is important because databases typically require a large amount of storage space.

- The query processor is important because it helps the database system simplify and facilitate access to data.
- It is the job of the database system to translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level.

****

2.2. RELATIONAL OBJECT ORIENTED (DATA MODELS)

2.2.1 Introduction to Data Models

- Underlying the structure of a database is the data model: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
- A data model provides a way to describe the design of a database at the physical, logical, and view levels.

What is Data Model?

- Data model defines the logical structure of a database. Data Models are fundamental entities to introduce abstraction in a DBMS.
- Data models define how data is connected to each other and how they are processed and stored inside the system. There are a number of different database data models.
- Underlying the structure of a database is the data model: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

- A data model provides a way to describe the design of a database at the physical, logical, and view levels.
- Historically, the network data model and the hierarchical data model preceded the relational data model.
- Models were tied closely to the underlying implementation, and complicated the task of modeling data.
- As a result they are used little now, except in old database code that is still in service

in some places.

- The relational model uses a collection of tables to represent both data and the relationships among those data.
- Each table has multiple columns, and each column has a unique name. Tables are also known as relations.
  - Amongst those that have been used for attribute data in GIS are the hierarchical, network, relational, object-relational and object-oriented data models.
  - Of these the relational data model has become the most widely used model.

Types of Data Models

The different types of data models are:

a) Hierarchical data model
b) Network data model
c) Relational Model
d) Entity-Relationship Model
e) Object-Based Data Model
f) Semi-structured Data Model

Models were tied closely to the underlying implementation, and complicated the task of modeling data.

1. Hierarchical Data Model

- Hierarchical data model is the oldest type of the data model.
- It was developed by IBM in 1968.
- It organizes data in the tree-like structure.

Hierarchical model consists of the the following :

- It contains nodes which are connected by branches.
- The topmost node is called the root node.
- If there are multiple nodes appear at the top level, then these can be called as root segments.
- Each node has exactly one parent.
- One parent may have many child.

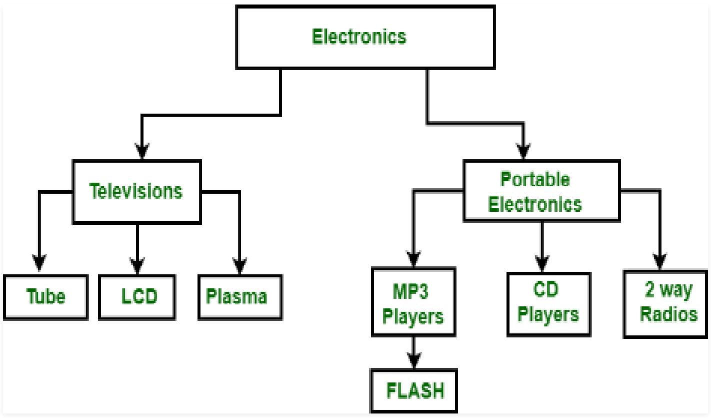


Figure – Hierarchical Data Model

- Electronics is the root node which has two children i.e. Televisions and Portable Electronics.
- These two has further children for which they act as parent.

For example:

- Television has children as Tube, LCD and Plasma, for these three Television act as parent. It follows one to many relationships.

2. Network Data Model

- It is the advance version of the hierarchical data model.
- To organize data it uses directed graphs instead of the tree-structure.
- In this child can have more than one parent.
- It uses the concept of the two data structures i.e. Records and Sets.

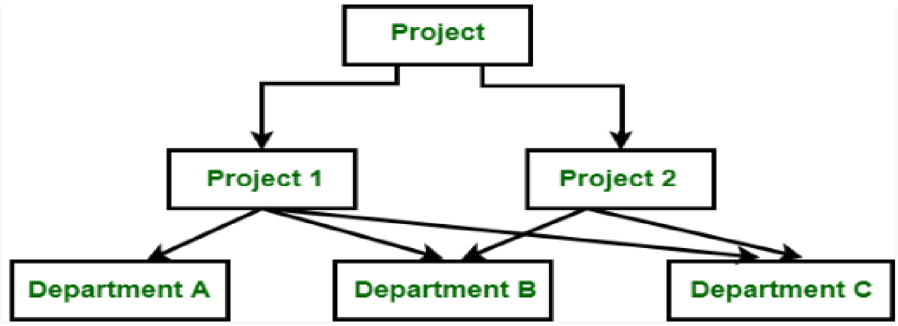


Figure – Network Data Model

- In the above figure, Project is the root node which has two children i.e. Project

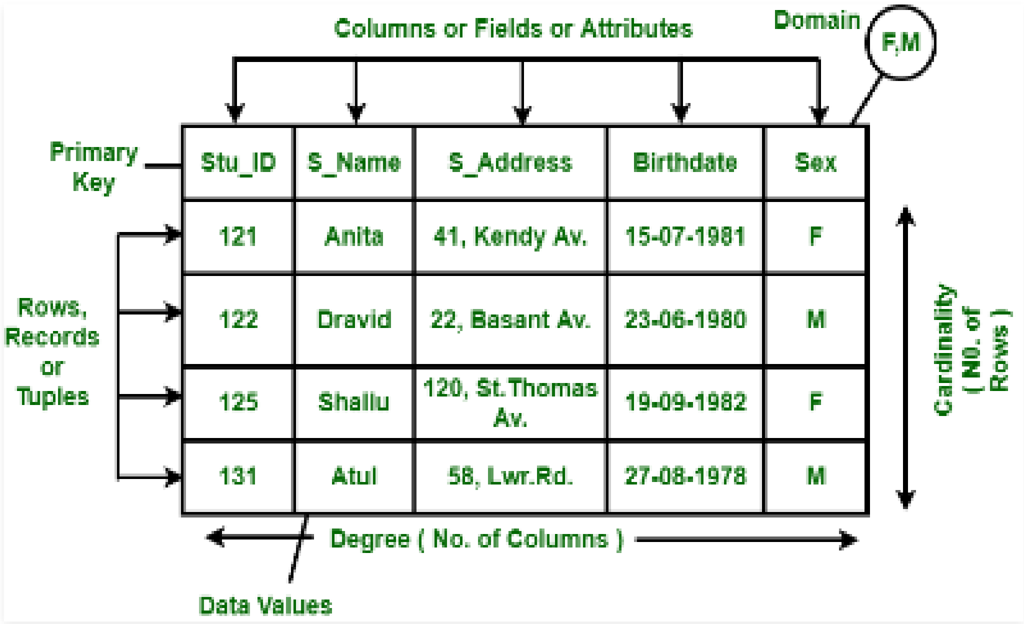
1 and Project 2.

- Project 1 has 3 children and Project 2 has 2 children.
- Total there are 5 children i.e Department A, Department B and Department C, they are network related children as we said that this model can have more than one parent.
- So, for the Department B and Department C have two parents i.e. Project 1 and Project 2.

3. Relational Data Model :

- The relational data model was developed by E.F. Codd in 1970.
- There are no physical links as they are in the hierarchical data model.
- Following are the properties of the relational data model :
    - Data is represented in the form of table only.
    - It deals only with the data not with the physical structure.
    - It provides information regarding metadata.
    - At the intersection of row and column there will be only one value for the tuple.
    - It provides a way to handle the queries with ease.
    - The relational model is today the primary data model for commercial data processing applications.
    - It attained its primary position because of its simplicity, which eases the job of the programmer, compared to earlier data models such as the network model or the hierarchical model.
    - The relational model uses a collection of tables to represent both data and the relationships among those data.
    - Each table has multiple columns, and each column has a unique name. Tables are also known as relations.
    - The relational model is an example of a record-based model.
    - The relational data model is the most widely used data model, and a vast majority of current database systems are based on the relational model.
    - Record-based models are so named because the database is structured in fixed-format records of several types.
    - Each table contains records of a particular type.
    - Each record type defines a fixed number of fields, or attributes.
    - A relational database consists of a collection of tables, each of which is assigned a unique name.
    - The columns of the table correspond to the attributes of the record type.

Relational Model Example

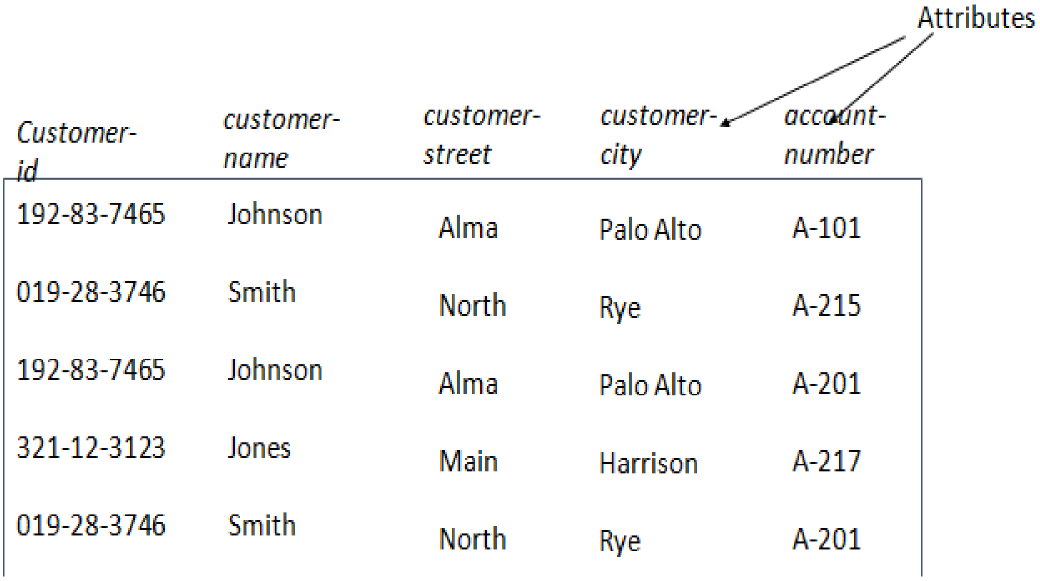


Columns or Fields or Attributes
Domain
F,M
Primary Key
Rows, Records or Tuples
Cardinality (N0. of Rows)
Degree (No. of Columns)
Data Values

Stu_ID	S_Name	S_Address	Birthdate	Sex
121	Anita	41, Kendy Av.	15-07-1981	F
122	Dravid	22, Basant Av.	23-06-1980	M
125	Shallu	120, St.Thomas Av.	19-09-1982	F
131	Atul	58, Lwr.Rd.	27-08-1978	M

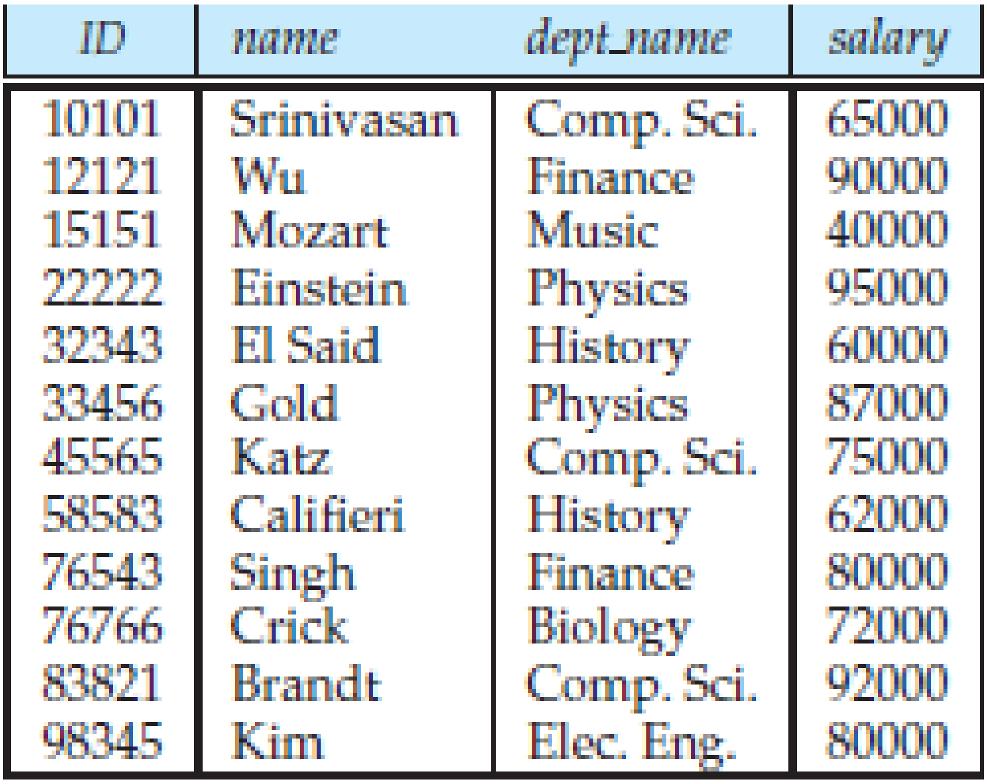
Figure – Relational Data Model

Historically, the network data model and the hierarchical data model preceded the relational data model.



Attributes

Customer-id	customer-name	customer-street	customer-city	account-number
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201



ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

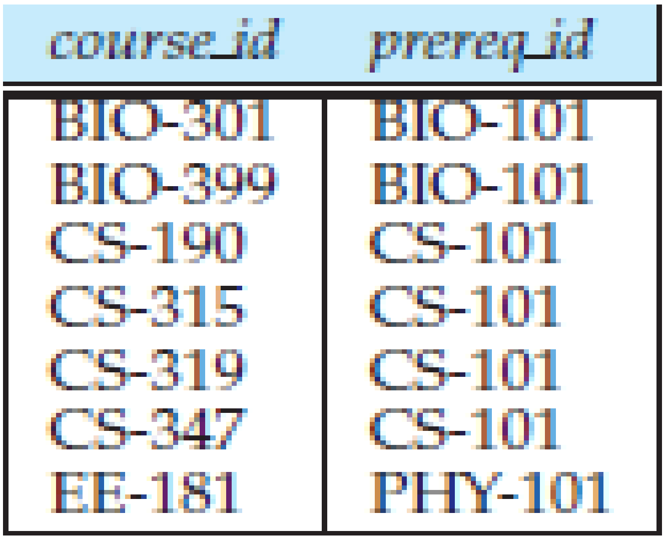
Figure 1: INSTRUCTOR TABLE

- For example, consider the instructor table of Figure:1, which stores information about instructors.
- The table has four column headers:
    - ID,
    - name,
    - dept name, and
    - salary.
- Each row of this table records information about an instructor, consisting of the instructor’s ID, name, dept name, and salary.

- Figure 3 shows a third table, prereq, which stores the prerequisite courses for each course.
- The table has two columns, course id and prereq id.
- Each row consists of a pair of course identifiers such that the second course is a prerequisite for the first course.
- Thus, a row in the prereq table indicates that two courses are related in the sense that one course is a prerequisite for the othe**r.**

Example 2:

- we consider the table instructor, a row in the table can be thought of as representing the relationship between a specified ID and the corresponding values for name,dept name, and salary values.



course_id	prereq_id
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

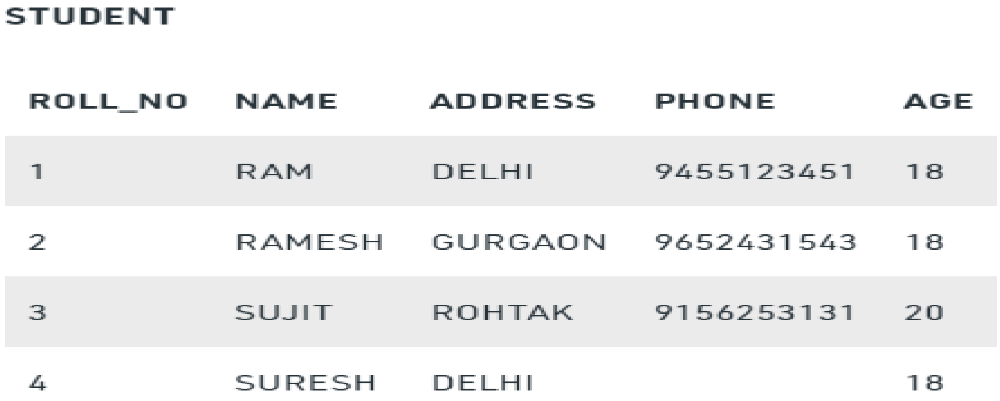
Figure 3: prereq TABLE

- Thus, in the relational model the term relation is used to refer to a table, while the term tuple is used to refer to a row. Similarly, the term attribute refers to a column of a table.
- Examining Figure 1 , we can see that the relation instructor has four attributes:
  - ID, name, dept name, and salary.
- We use the term relation instance to refer to a specific instance of a relation, i.e., containing a specific set of rows.
- The instance of instructor shown in Figure 1 has 12 tuples, corresponding to 12 instructors.

Components of Relational Data Model

- Data are organized in a series of two-dimensional tables, each of which contains records for one entity.
- These tables are linked by common data known as keys. Queries are possible on individual tables or on groups of tables.
- A relational database stores data in the form of relations (tables).

Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE and AGE shown in Table 1.



STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

Attribute: Attributes are the properties that define a relation. e.g.; ROLL_NO, NAME

Relation Schema: A relation schema represents name of the relation with its attributes.

e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE and AGE) is relation schema for STUDENT.

If a schema has more than 1 relation, it is called Relational Schema.

Tuple: Each row in the relation is known as tuple.

The above relation contains 4 tuples, one of which is shown as:

4	RAM	DELHI	9455123451	18

Relation Instance: The set of tuples of a relation at a particular instance of time is called as relation instance.

Table 1 shows the relation instance of STUDENT at a particular time. It can change whenever there is insertion, deletion or updation in the database.

Degree: The number of attributes in the relation is known as degree of the relation.

The STUDENT relation defined above has degree 5.

Cardinality: The number of tuples in a relation is known as cardinality.

The STUDENT relation defined above has cardinality 4.

Column: Column represents the set of values for a particular attribute.

The column ROLL_NO is extracted from relation STUDENT

ROLL_NO

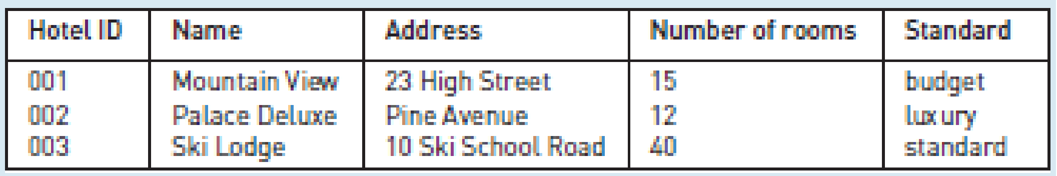
1

2

3

4

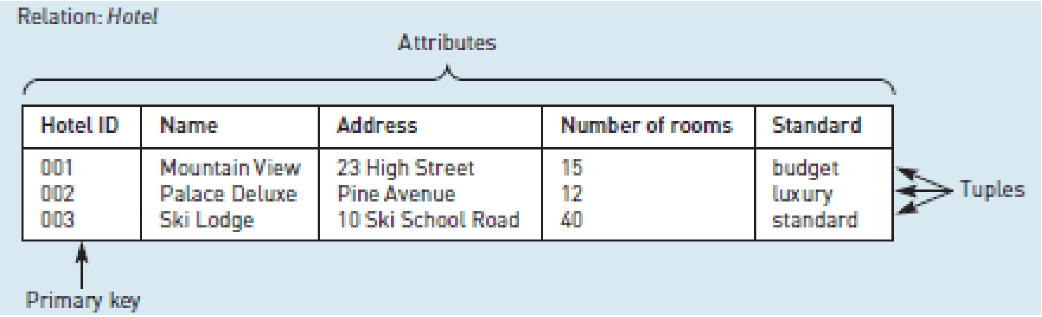
For the Happy Valley data, the below figure illustrates an example of one such table.



Hotel ID	Name	Address	Number of rooms	Standard
001	Mountain View	23 High Street	15	budget
002	Palace Deluxe	Pine Avenue	12	luxury
003	Ski Lodge	10 Ski School Road	40	standard

Relational database table data for Happy Valley

- The data in a relational database are stored as a set of base tables with the characteristics described above.
- Other tables are created as the database is queried and these represent virtual views.
- The table structure is extremely flexible and allows a wide variety of queries on the data.
- Queries are possible on one table at a time (for example, you might ask 'which hotels have more than 14 rooms?' or 'which hotels are luxury standard?'), or on more than one table by linking through key fields (for instance, 'which passengers originating fromthe UK are staying in luxury hotels?' or 'which ski lessons have pupils who are over 50 years of age?').
- Queries generate further tables, but these new tables are not usually stored. There are few restrictions on the types of query possible.



Hotel ID	Name	Address	Number of rooms	Standard
001	Mountain View	23 High Street	15	budget
002	Palace Deluxe	Pine Avenue	12	luxury
003	Ski Lodge	10 Ski School Road	40	standard

Database terminology applied to Happy Valley table

- With many relational databases querying is facilitated by menu systems and icons, or ‘query by example’ systems.
- Frequently, queries are built up of expressions based on relational algebra, using commands such as SELECT (to select a subset of rows), PROJECT (to select a subset of columns) or JOIN (to join tables based on key fields).
- SQL (standard query language) has been developed to facilitate the querying of relational databases.
- The advantages of SQL for database users are its completeness, simplicity, pseudo English- language style and wide application.
- However, SQL has not really developed to handle geographical concepts such as ‘near to’, ‘far from’ or‘connected to’.

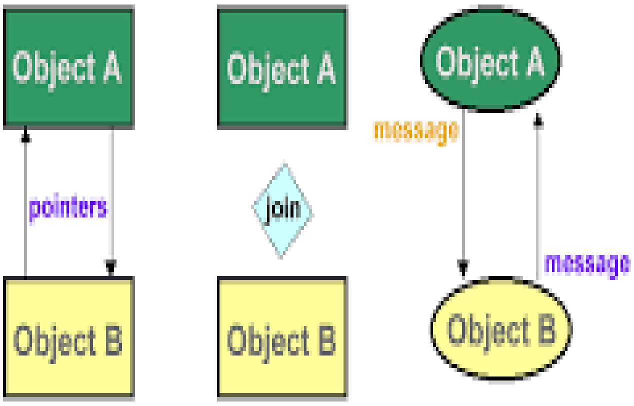
Semi-structured Data Model

- The semi-structured data model permits the specification of data where individual data items of the same type may have different sets of attributes.
- This is in contrast to the data models mentioned earlier, where every data item of a particular type must have the same set of attributes.
- The Extensible Markup Language (XML) is widely used to represent semi-structured data.

OBJECT –ORIENTAL DATA MODEL

What is Object-Oriental Data Model?

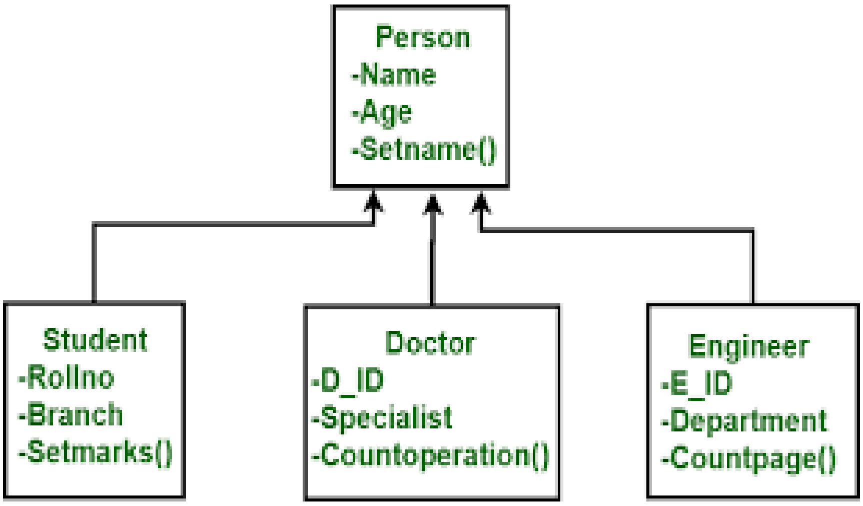
- An object data model is a data model based on object-oriented programming, associating methods (procedures) with objects that can benefit from class hierarchies.
- An object-oriented data model is one that extends the individual program space into the world of persistent object management and shareability.
- O-O Data model simplified, less constrained interface between objects.
- Relationship between entites is via messages not pointers or joins fields.



- Object Oriented Data Model represents the real world problems easily.
- An object is an abstraction of a real world entity or we can say it is an instance of class.

For example:

- Instances of student, doctor, engineer in the below figure:



- In Object Oriented Data Model, data and their relationships are contained in a single structure which is referred as object in this data model.
- In this, real world problems are represented as objects with different attributes.
- All objects have multiple relationships between them.
- Basically, it is combination of Object Oriented programming and Relational Database Model as it is clear from the following figure :

Object Oriented Data Model = Combination of Object Oriented Programming + Relational database model

Major components of Object-Oriental Data Model

- **Objects –**
- An object is an abstraction of a real world entity or we can say it is an instance of class.
- Objects encapsulates data and code into a single unit which provide data abstraction by hiding the implementation details from the user.

For example:

Instances of student, doctor, engineer in above figure.

- **Attribute –**

  An attribute describes the properties of object.

For example: Object is STUDENT and its attribute are Roll no, Branch, Semester in the Student class.

- **Methods –**

  Method represents the behavior of an object. Basically, it represents the real-world action.

For example:

Finding a STUDENT marks in above figure as Setmarks().

- **Class –**

  A class is a collection of similar objects with shared structure i.e. attributes and behavior i.e. methods.
- An object is an instance of class.

For example:

Person, Student, Doctor, Engineer in above figure.

- **Inheritance –**

  By using inheritance, new class can inherit the attributes and methods of the old class i.e. base class.

For example:

as classes Student, Doctor and Engineer are inherited from the base class Person

```
class student
{
char Name[20];
```

```
 int roll_no;
 -- --
public: void search();
 void update();
}
```

In this example, students refers to class and S1, S2 are the objects of class which can be created in main function.

Need of Object Oriented Data Model :

- To represent the complex real world problems there was a need for a data model that is closely related to real world.
- Object Oriented Data Model represents the real world problems easily.

Advantages and disadvantages of Object Oriented Data Model

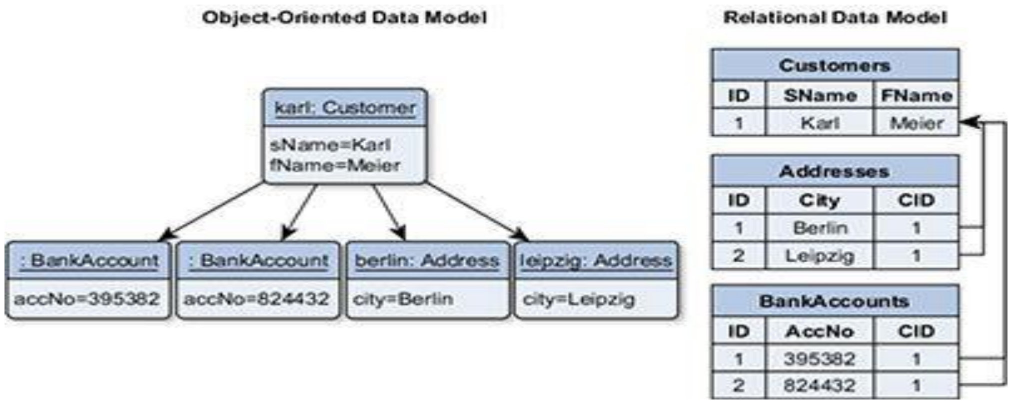
Advantages :

a) Codes can ne reused due to inheritance.
b) Easily understandable.
c) Cost of maintenance can reduced due to reusability of attributes and functions because of inheritance.

Disadvantages :

a) It is not properly developed so not accepted by users easily

Difference between OOM Vs RM



Customers

ID	SName	FName
1	Karl	Meier

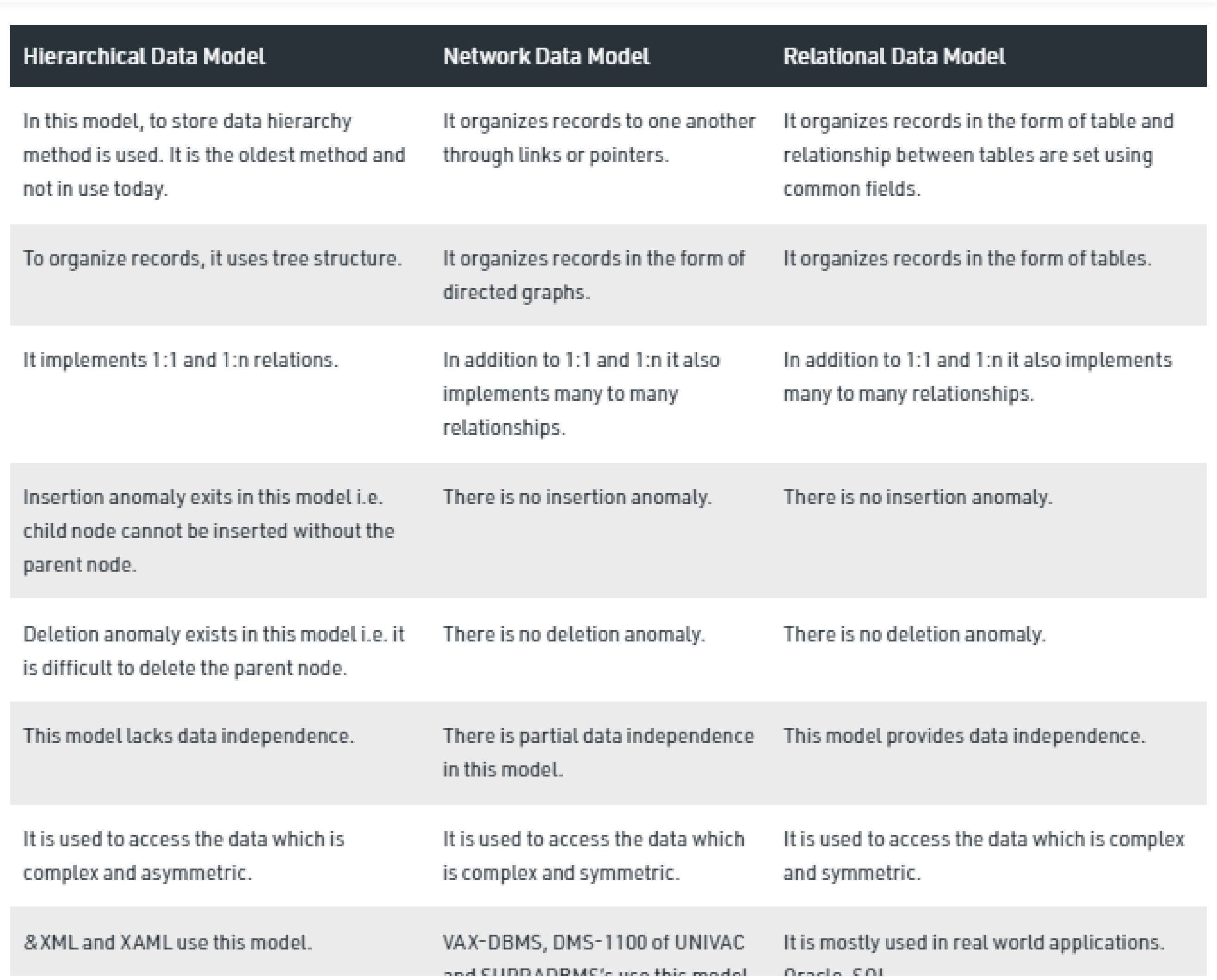
Addresses

ID	City	CID
1	Berlin	1
2	Leipzig	1

BankAccounts

ID	AccNo	CID
1	395382	1
2	824432	1

Difference between Hierarchical, Network and Relational Data Model



Hierarchical Data Model	Network Data Model	Relational Data Model
In this model, to store data hierarchy method is used. It is the oldest method and not in use today.	It organizes records to one another through links or pointers.	It organizes records in the form of table and relationship between tables are set using common fields.
To organize records, it uses tree structure.	It organizes records in the form of directed graphs.	It organizes records in the form of tables.
It implements 1:1 and 1:n relations.	In addition to 1:1 and 1:n it also implements many to many relationships.	In addition to 1:1 and 1:n it also implements many to many relationships.
Insertion anomaly exits in this model i.e. child node cannot be inserted without the parent node.	There is no insertion anomaly.	There is no insertion anomaly.
Deletion anomaly exists in this model i.e. it is difficult to delete the parent node.	There is no deletion anomaly.	There is no deletion anomaly.
This model lacks data independence.	There is partial data independence in this model.	This model provides data independence.
It is used to access the data which is complex and asymmetric.	It is used to access the data which is complex and symmetric.	It is used to access the data which is complex and symmetric.
&XML and XAML use this model.	VAX-DBMS, DMS-1100 of UNIVAC and SUPRADBMS's use this model	It is mostly used in real world applications. Oracle, SQL

****

2.3 E-R DIAGRAM

2.3.1 Overview of Database Design

- An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).
- ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.
- ERD is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

- Conceptual design follows requirements analysis, Yields a high-level description of data to be stored

- ER model popular for conceptual design
- Constructs are expressive, close to the way people think about their applications.
- Basic constructs: entities, relationships, and attributes (of entities and relationships).
- Some additional constructs: weak entities, ISA hierarchies, and aggregation.
- Note: There are many variations on ER model.

Conceptual design: (ER Model is used at this stage.)

- What are the entities and relationships in the enterprise?
- What information about these entities and relationships should we store in the database?
- What are the integrity constraints or business rules that hold?

A database `schema' in the ER Model can be represented pictorially (ER diagrams).
Can map an ER diagram into a relational schema.

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge.
A basic ER model is composed of entity types and specifies relationships that can exist between entities. Wikipedia

- Entity-relationship diagrams or ER diagrams are used to showcase the relationships developed between objects or entities in a system.
- Also known as the entity-relationship model, this type of flowchart is used in various fields such as research, education, business information system, or software engineering.
- Several kinds of integrity constraints can be expressed in the ER model: key constraints, participation constraints, and overlap/covering constraints for ISA hierarchies.
- Some foreign key constraints are also implicit in the definition of a relationship set.
- Some constraints (notably, functional dependencies) cannot be expressed in the ER model.
- Constraints play an important role in determining the best database design for an enterprise.

What is Entity-Relationship Model ?

- E-R model stands for Entity-Relationship model.
- ER Model is used to model the logical view of the system from data perspective which consists of these components:

  - Entity,
  - Entity Type,
  - Entity Set.

- The entity-relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects.
- **An Entity may be an object with a physical existence –**
  a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.
- An Entity is an object of Entity Type and set of all entities is called an entity set.
- e.g.; E1 is an entity having Entity Type Student and set of all students is called Entity Set.

- **An Entity Type** defines a collection of similar entities and set of all entities is called an entity set.
- An entity is a "thing" or "object" in the real world that is distinguishable from other objects.
- The entity- relationship model is widely used in database design.
- An ER model is a design or blueprint of a database that can later be implemented as a database. ER Model is best used for the conceptual design of a database.

Entity: Real-world object distinguishable from other objects. An entity is described (in DB) using a set of attributes.

Entity Set: A collection of similar entities.

E.g., all employees.

All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies, anyway!)

Each entity set has a key.

Each attribute has a domain.

2.3.2 Main components of E-R model

The main components of E-R model are:

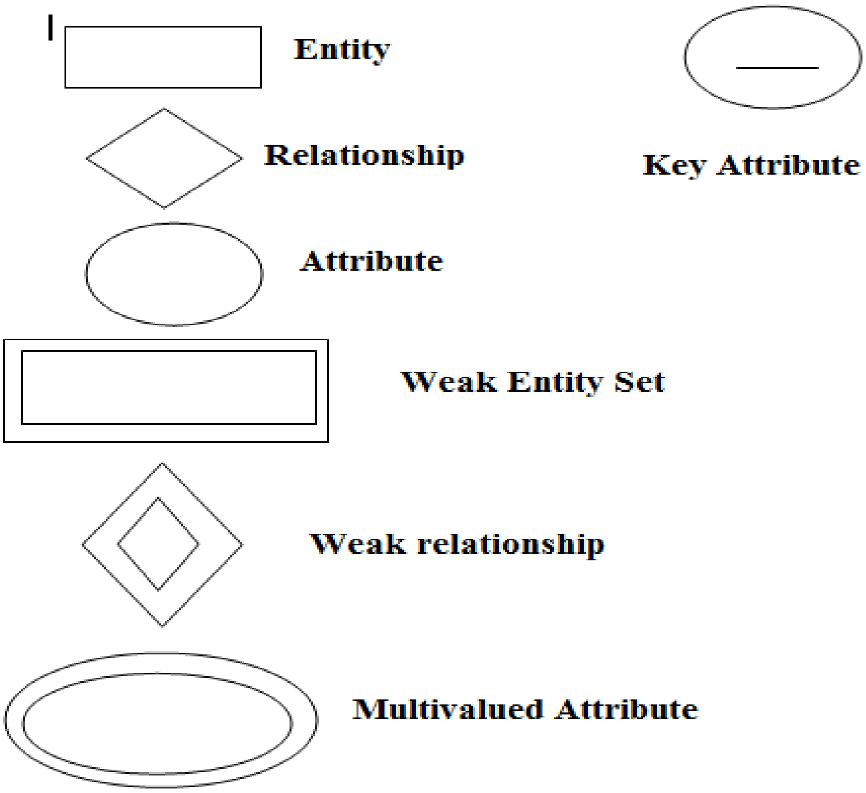
a) **Entity –** An entity in an ER Model is a real-world entity having properties called attributes. Every attribute is defined by its set of values called domain. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.

b) **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. The following are the Mapping cardinalities - one to one, one to many, many to one & many to many.

c) **Attribute(s):**

Attributes are the properties which define the entity type.

For example, Roll_No, Name, DOB, Age, Address, Mobile_No are the attributes which defines entity type Student.

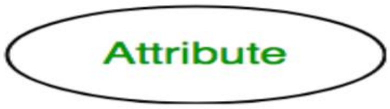


ER diagrams or ERD's are composed of three main elements:

- entities,
- attributes, and
- relationships.

- Entities - typically displayed in a rectangle, entities can be represented by objects, persons, concepts, or events that contain data.

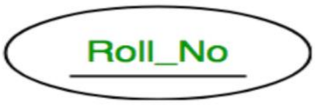
- Attributes - displayed in a circle or an oval, the attributes refer to the characteristics of an entity. They can be categorized as simple, composite, or derived, and an object can have one or multiple attributes.



<u>1. Key Attribute</u> –

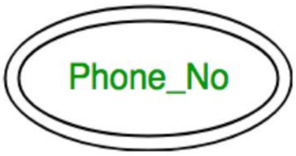
The attribute which uniquely identifies each entity in the entity set is called key attribute.

For example, Roll_No will be unique for each student. In ER diagram, key attribute is represented by an oval with underlying lines.



3. Multivalued Attribute –

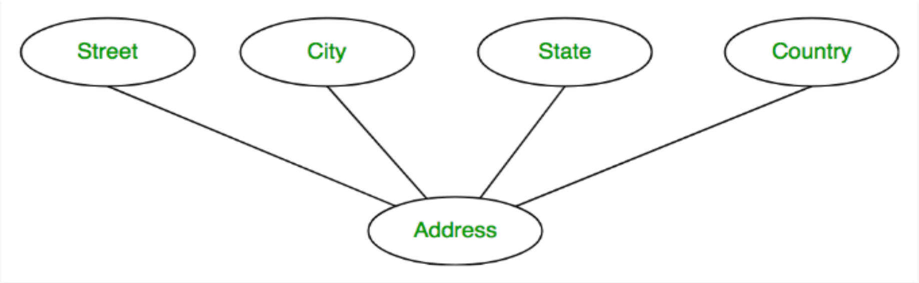
An attribute consisting more than one value for a given entity.



<u>4. Composite Attribute</u> –

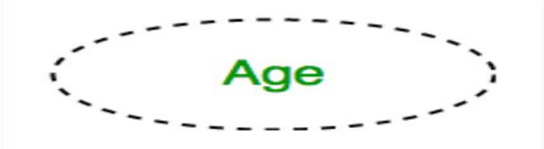
An attribute composed of many other attribute is called as composite attribute.

For example, Address attribute of student Entity type consists of Street, City, State, and Country. In ER diagram, composite attribute is represented by an oval comprising of ovals.
For example, Phone_No (can be more than one for a given student). In ER diagram, multivalued attribute is represented by double oval.



5. Derived Attribute –

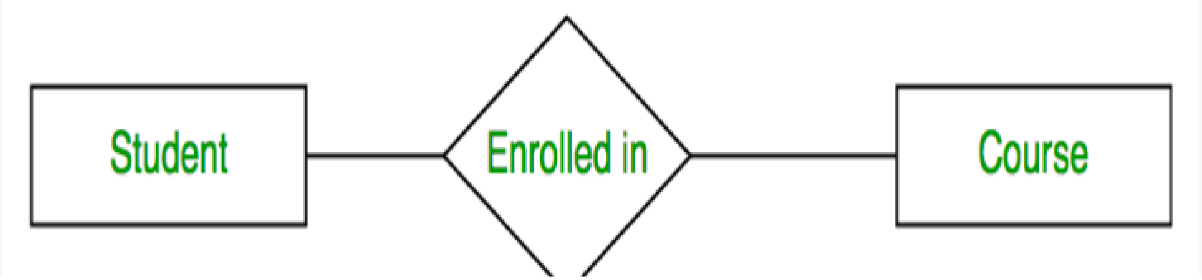
An attribute which can be derived from other attributes of the entity type is known as derived attribute. e.g.; Age (can be derived from DOB).
In ER diagram, derived attribute is represented by dashed oval.



- Relationships - illustrate how two or more entities interact with each other. They are displayed as labels placed on the lines connecting the objects.

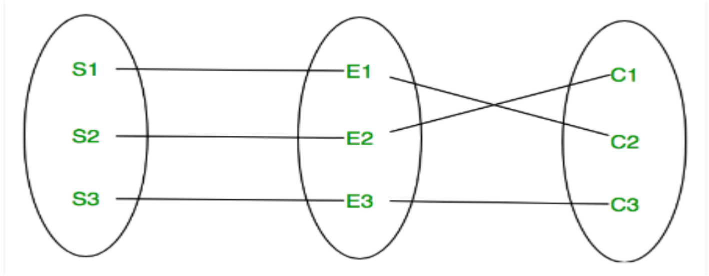
Relationship Type and Relationship Set:

A relationship type represents the association between entity types.
For example,‘Enrolled in’ is a relationship type that exists between entity type Student and Course. In ER diagram, relationship type is represented by a diamond and connecting the entities with lines.



A set of relationships of same type is known as relationship set.

The following relationship set depicts S1 is enrolled in C2, S2 is enrolled in C1 and S3 is enrolled in C3.

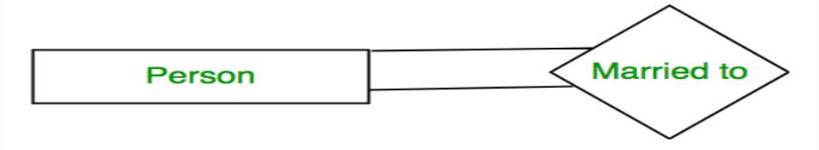


Degree of a relationship set:

The number of different entity sets participating in a relationship set is called as degree of a relationship set.

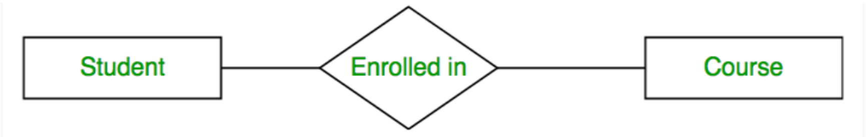
1. Unary Relationship –

When there is only ONE entity set participating in a relation, the relationship is called as unary relationship. For example, one person is married to only one person.



2. Binary Relationship –

When there are TWO entities set participating in a relation, the relationship is called as binary relationship.For example, Student is enrolled in Course.



3. n-ary Relationship –

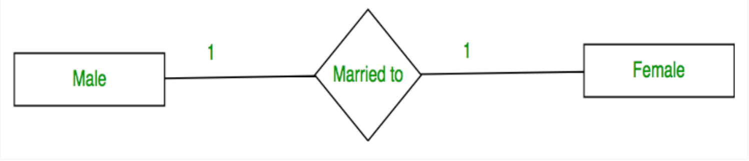
When there are n entities set participating in a relation, the relationship is called as n-ary relationship.

Cardinality:

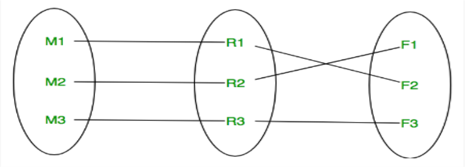
The number of times an entity of an entity set participates in a relationship set is known as cardinality.

Cardinality can be of different types:

1. One to one – When each entity in each entity set can take part only once in the relationship, the cardinality is one to one. Let us assume that a male can marry to one female and a female can marry to one male. So the relationship will be one to one.



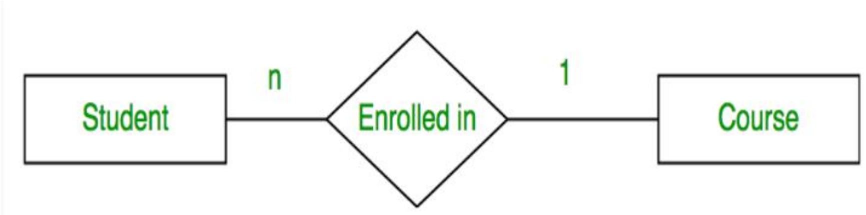
Using Sets, it can be represented as:



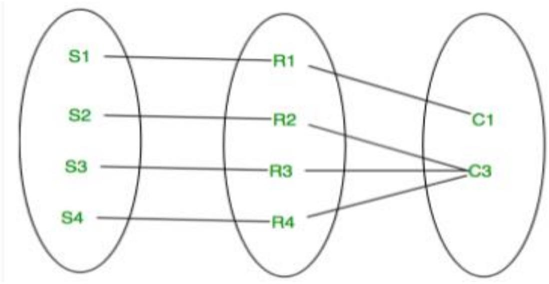
2. Many to one –

When entities in one entity set can take part only once in the relationship set and entities in other entity set can take part more than once in the relationship set, cardinality is many to one.

Let us assume that a student can take only one course but one course can be taken by many students.
So the cardinality will be n to 1. It means that for one course there can be n students but for one student, there will be only one course.



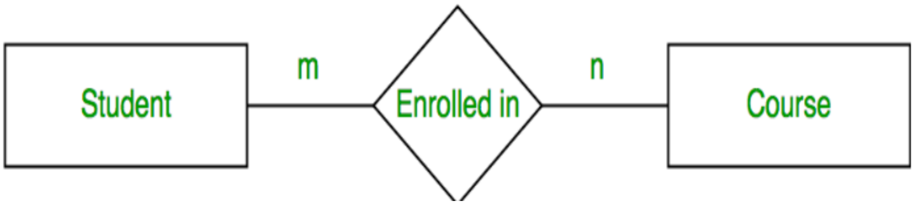
Using Sets, it can be represented as:



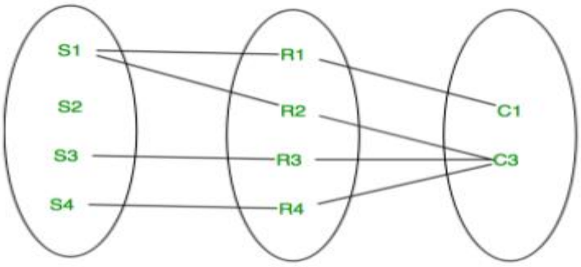
3. Many to many –

When entities in all entity sets can take part more than once in the relationship cardinality is many to many.
Let us assume that a student can take more than one course and one course can be taken by many students.

So the relationship will be many to many.

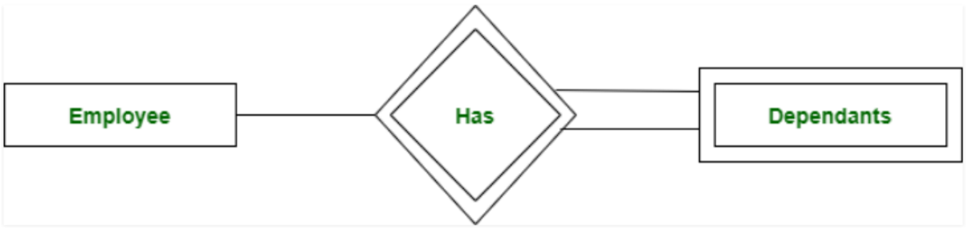


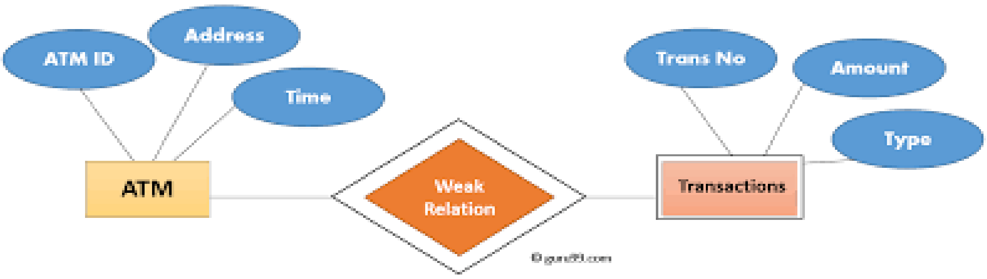
Using sets, it can be represented as:



Weak Entity Type and Identifying Relationship:

- An entity type has a key attribute which uniquely identifies each entity in the entity set. But there exists some entity type for which key attribute can’t be defined. These are called Weak Entity type.
- For example, A company may store the information of dependents (Parents, Children, Spouse) of an Employee. But the dependents don’t have existence without the employee. So Dependent will be weak entity type and Employee will be Identifying Entity type for Dependent.



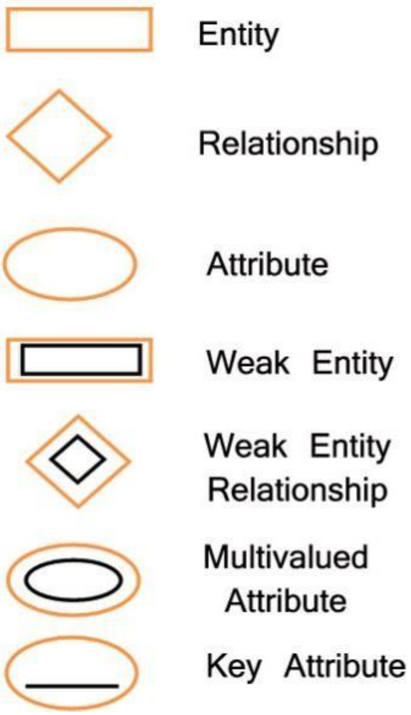


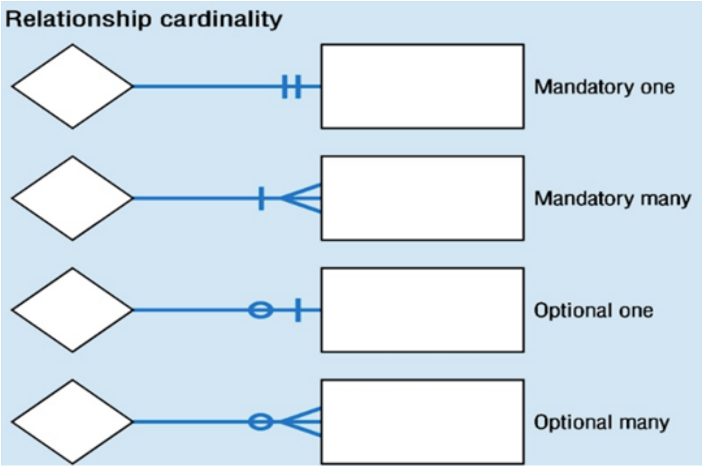
Modes of entity-relationship Models

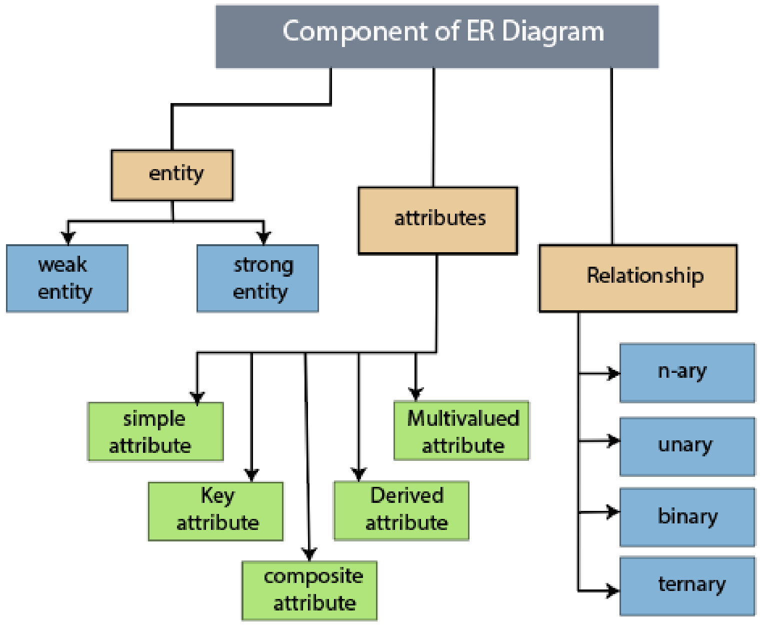
- ERDs are generally depicted in one or more of the following models:
- A conceptual data model, which lacks specific detail but provides an overview of the scope of the project and how data sets relate to one another.
- A logical data model, which is more detailed than a conceptual data model, illustrating specific attributes and relationships among data points. While a conceptual data model does not need to be designed before a logical data model, a physical data model is based on a logical data model.

- A physical data model, which provides the blueprint for a physical manifestation -- such as a relational database -- of the logical data model. One or more physical data models can be developed based on a logical data model.

The following are the various symbols used in ER diagram:





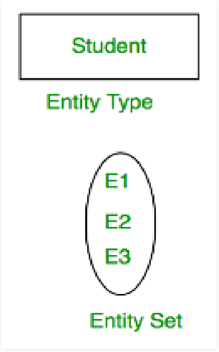


How do you draw an entity-relationship diagram?
When building your ER diagram, there are a few steps to keep in mind:
Identify the purpose - what is the purpose of the ERD template you are creating?
Identify entities - once you have identified these entities, add them in rectangles.
Identify relationships - how are these entities related?

Identify attributes - what are the key attributes of

Example

EXAMPLE:
E1 is an entity having Entity Type Student and set of all students is called Entity Set.
In ER diagram, Entity Type is represented as:



- ER Model is used to model the logical view of the system from data perspective which consists of these components:

- Entity, Entity Type, Entity Set –
- An Entity may be an object with a physical existence – a particular person, car, house, or employee – or it may be an object with a conceptual existence – a company, a job, or a university course.

- An Entity is an object of Entity Type and set of all entities is called as entity set.

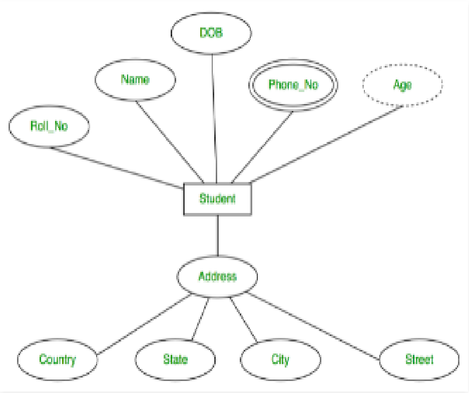
In ER diagram, attribute is represented by an oval.

- Relationship: Association among two or more entities. E.g., Attishoo works in Pharmacy department.
- Relationship Set: Collection of similar relationships.

  An n-ary relationship set R relates n entity sets E1 ... En;
  each relationship in R involves entities e1 E1, ..., en En

  - Same entity set could participate in different relationship sets, or in different "roles" in same set.

The complete entity type Student with its attributes can be represented as:



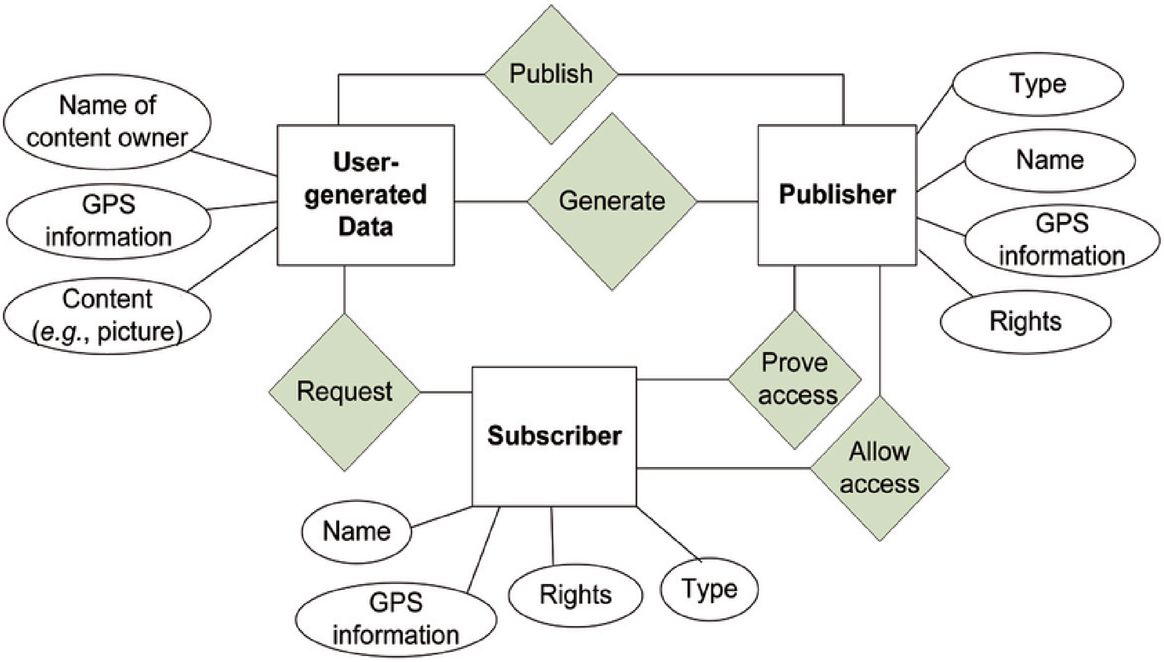
Uses of ER Model

- Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization.
- An ERD can still serve as a reference point, should any debugging or business process re-engineering be needed later.
- While an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured data.
- It's also unlikely to be helpful on its own in integrating data into a pre-existing

information system.

The figure shows the ER diagram for the GPS tracking system.

The design has three entities namely User-generated Data, Publisher and Subscriber.



ER Diagram of GPS System

****

2.4 SPATIAL DATA MODELS

2.4.1 Representation of Spatial Features:

- The vector data model uses the geometric objects of point, line, and polygon to represent spatial features.
- A point has zero dimension and has only the property of location.
- A point feature is made of a point or a set of points.
- Wells, benchmarks, and gravel pits on a topographic map are examples of point features.
- A line is one-dimensional and has the property of length, in addition to location.
- A line has two end points and may have additional points in between to mark the shape of the line. polygon is two-dimensional and has the properties of area (size) and perimeter, in addition to location.

- Made of connected, closed, nonintersecting lines, the perimeter or the boundary defines the area of a polygon.

2.4.2 SPATIAL DATA OR GIS DATA

- The three types of GIS Data are:
- Spatial
- Attribute &
- Metadata

- Traditionally spatial data are stored in the form of digital databases and presented them in the form of maps.
- Two basic types of spatial data models have been evolved for storing data geographically.

Real world
?
map

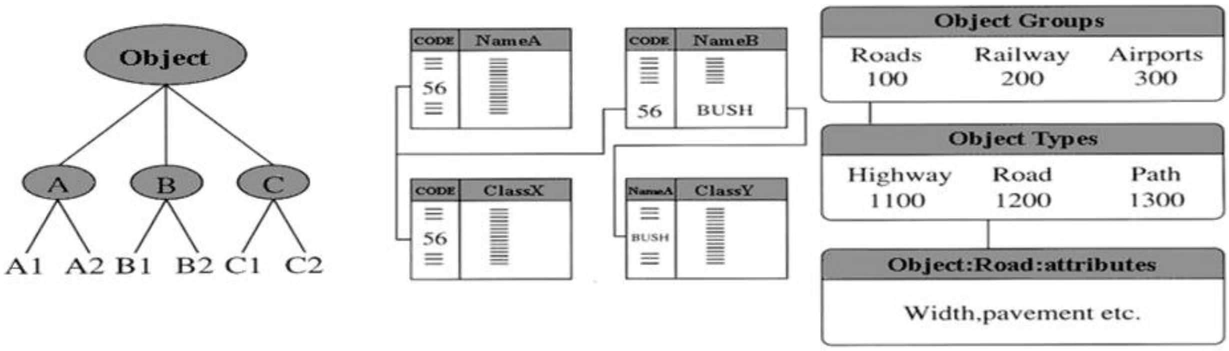
- GIS Spatialdata Model allows geographic features in real world location to be digitally represented and stored in a database.
- So that they can be abstractly presented in map (analog) form, and can also be worked with and manipulated to address some problem.

2.4.3 Types of Basic Data Models

a) Hierarchical Model

b) Network Model

c) Object-oriented



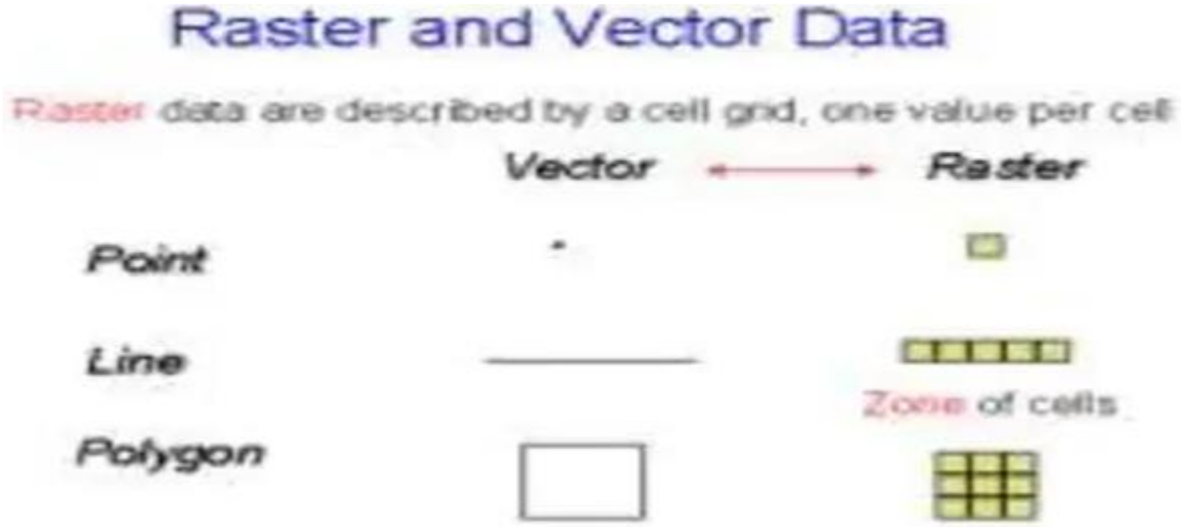
(a) Hierarchical model (c) Relation of model (d) Object oriented model

Spatial Data Model Basics

The spatial data models are evolved from two major spatial data types:

i) Raster data

ii) Vector data



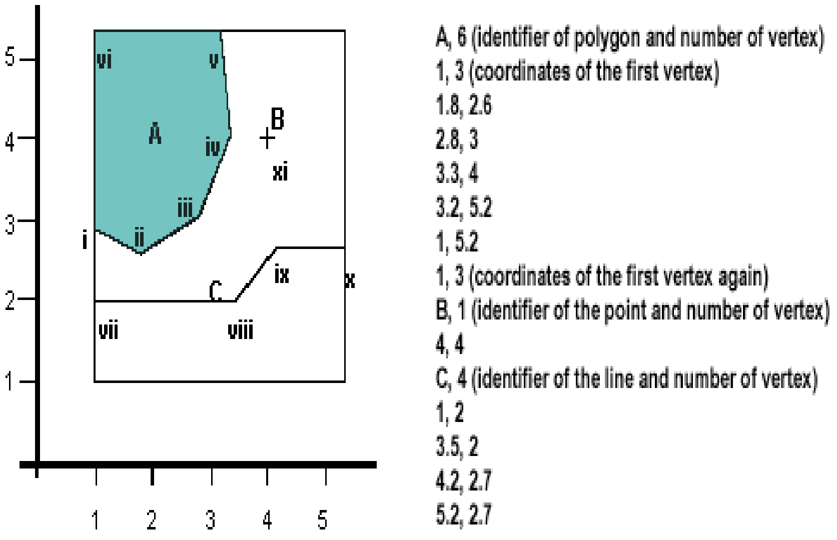
2.4.4 SPATIAL DATA OR GIS DATA: VECTOR

Point Data — layers containing by points (or "events") described by x,y (lat,long; easting, northing)

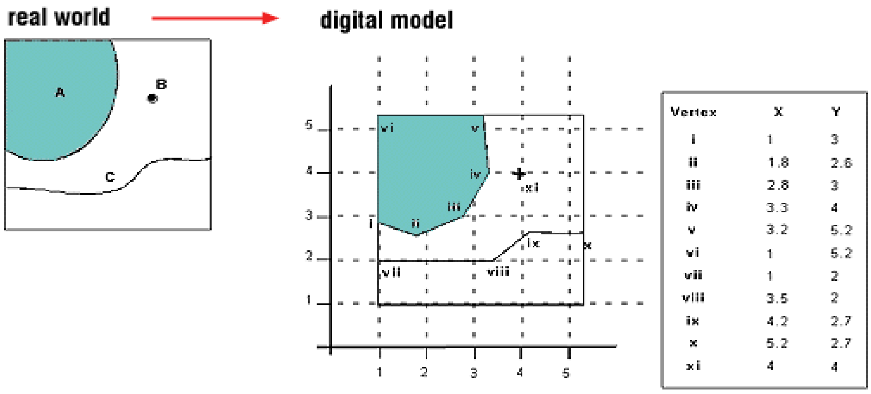
Line/Polyline Data — layers that are described by x,y points (nodes, events) and lines (arcs) between points (line segments and polylines)

Polygon Data — layers of closed line segments enclosing areas that are described by attributes.

Polygon data can be "multipart" like the islands of the state of Hawaii.



A, 6 (identifier of polygon and number of vertex)
1, 3 (coordinates of the first vertex)
1.8, 2.6
2.8, 3
3.3, 4
3.2, 5.2
1, 5.2
1, 3 (coordinates of the first vertex again)
B, 1 (identifier of the point and number of vertex)
4, 4
C, 4 (identifier of the line and number of vertex)
1, 2
3.5, 2
4.2, 2.7
5.2, 2.7



Vertex	X	Y
i	1	3
ii	1.8	2.6
iii	2.8	3
iv	3.3	4
v	3.2	5.2
vi	1	5.2
vii	1	2
viii	3.5	2
ix	4.2	2.7
x	5.2	2.7
xi	4	4

2.4.5 SPATIAL DATA OR GIS DATA: RASTER

- Raster or grid data (matrices of numbers describing e.g., elevation, population, herbicide use, etc.
- images or pictures such as remote sensing data or scans of maps or other photos.
- This is special "grid" where the number in each cell describes what color to paint or the spectral character of the image in that cell. (to be used, the "picture" must be placed on a coordinate system, or "rectified" or "georeferenced")

2.4.6 Types of GIS data models

The two basic data models of GIS are

a) Raster data model and
b) Vector data model

Other important data models are :

c) TIN (Triangulated Irregular Network) and
d) DEM (Digital Elevation Model).

Raster consists of matrix of cells organized into rows and coloumns where as vector represents data using points, lines and polygons

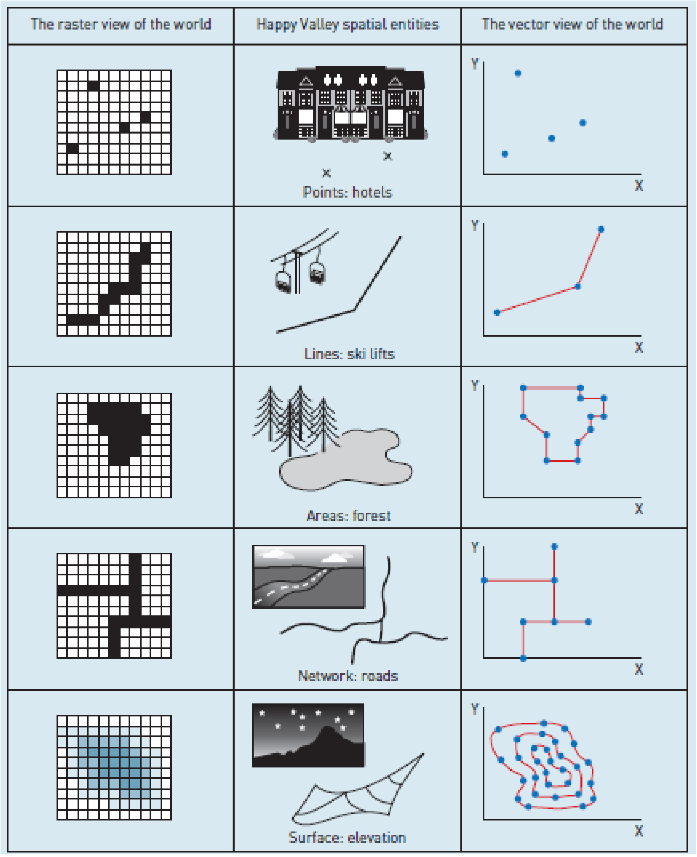
Raster Data Model:

- The raster spatial data model is one of a family of spatial data models described as tessellations.
- In the raster world individual cells are used as the building blocks for creating images of point, line, area, network and surface entities.
- In the raster world the basic building block is the individual grid cell, and the shape and character of an entity is created by the grouping of cells.
- The size of the grid cell is very important as it influences how an entity appears.

Vector Data Model

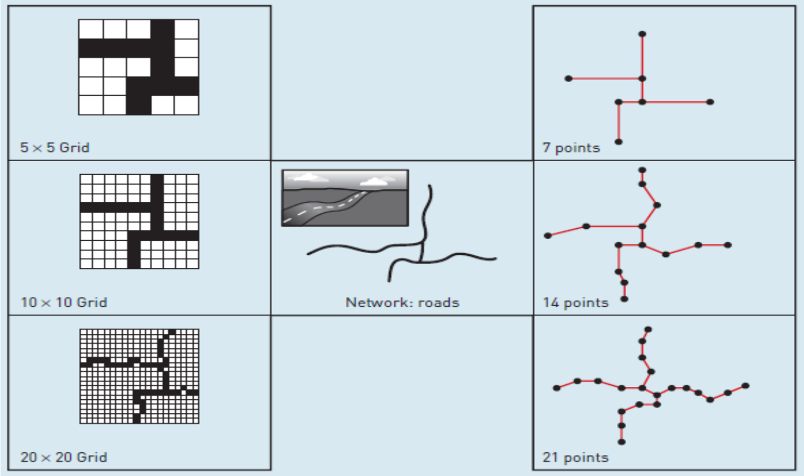
A vector spatial data model uses two-dimensional Cartesian (x,y) co-ordinates to storethe shape of a spatial entity.

- In the vector world the point is the basic building block from which all spatial entities are constructed.
- The simplest spatial entity, the point, is represented by a single (x,y) co-ordinate pair. Line and area entities are constructed by connecting aseries of points into chains and polygons.
- The more complex the shape of a line or area feature the greater the number of points required to represent it. Selecting the appropriate number of points to construct an entity is one of the major dilemmas when using the vector approach.


The raster view of the world
Happy Valley spatial entities
The vector view of the world
Points: hotels
Lines: ski lifts
Areas: forest
Network: roads
Surface: elevation
Y
X

Raster and vector spatial data

- If too few points are chosen the character, shape and spatial properties of the entity (for example, area, length, perimeter) will be compromised.
- If too many points are used, unnecessary duplicate information will be stored and this will be costly in terms of data capture and computer storage.



Effect of changing resolution in the vector and raster worlds

****

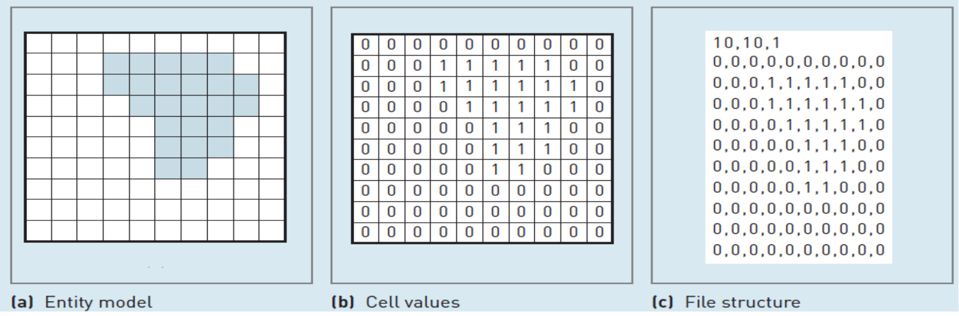
2.5 RASTER DATA STRUCTURES

2.5.1 Introduction to Spatial Data Structure

- Data structures provide the information that the computer requires to reconstruct the spatial data model in digital form.
- There are many different data structures in use in GIS.
- This diversity is one of the reasons why exchanging spatial data between different GIS software can be problematic.
- However, despite this diversity data structures can be classified according to whether they are used to structure raster or vector data.

Raster data structures

- In the raster world a range of different methods is used to encode a spatial entity for storage and representation in the computer. The below figure shows the most straightforward method of coding raster data.
- The cells in each line of the image (Figure: a) are mirrored by an equivalent row of numbers in the file structure (Figure: c).
- The first line of the file tells the computer that the image consists of 10 rows and 10 columns and that the maximum cell value is 1.
- In this example, a value of 0 has been used to record cells where the entity is not present and a value of 1 for cells where the entity is present (Figure: b).



(a) Entity model

(b) Cell values

0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

(c) File structure

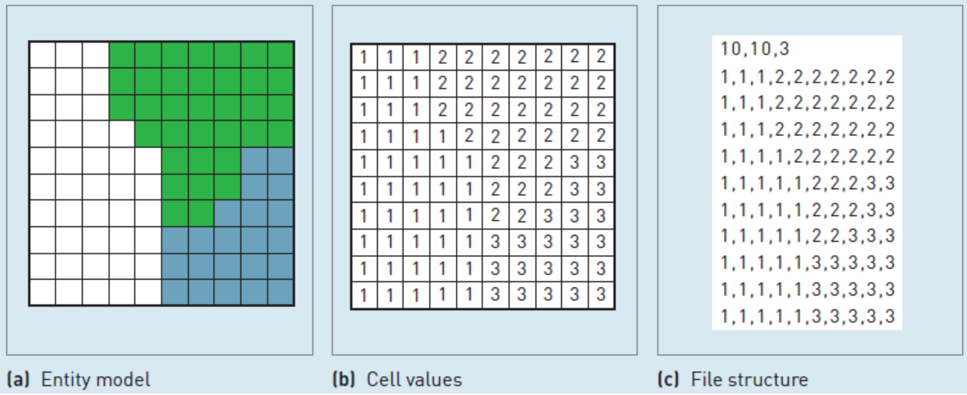
```
10,10,1
0,0,0,0,0,0,0,0,0,0
0,0,0,1,1,1,1,1,0,0
0,0,0,1,1,1,1,1,1,0
0,0,0,0,1,1,1,1,1,0
0,0,0,0,0,1,1,1,0,0
0,0,0,0,0,1,1,1,0,0
0,0,0,0,0,1,1,0,0,0
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0
```

A simple raster data structure

- In a simple raster data structure, such as illustrated in the above figure, different spatial features must be stored as separate data layers.
- Thus, to store more raster entities, separate data files would be required, each

representing a different layer of spatial data.

- However, if the entities do not occupy the same geographic location (or cells in the raster model), then it is possible to store them all in a single layer, with an entity code given to each cell.
- This code informs the user which entity is present in which cell.



(a) Entity model

1	1	1	2	2	2	2	2	2	2
1	1	1	2	2	2	2	2	2	2
1	1	1	2	2	2	2	2	2	2
1	1	1	1	2	2	2	2	2	2
1	1	1	1	1	2	2	2	3	3
1	1	1	1	1	2	2	2	3	3
1	1	1	1	1	2	2	3	3	3
1	1	1	1	1	3	3	3	3	3
1	1	1	1	1	3	3	3	3	3
1	1	1	1	1	3	3	3	3	3

(b) Cell values

```
10,10,3
1,1,1,2,2,2,2,2,2,2
1,1,1,2,2,2,2,2,2,2
1,1,1,2,2,2,2,2,2,2
1,1,1,1,2,2,2,2,2,2
1,1,1,1,1,2,2,2,3,3
1,1,1,1,1,2,2,2,3,3
1,1,1,1,1,2,2,3,3,3
1,1,1,1,1,3,3,3,3,3
1,1,1,1,1,3,3,3,3,3
1,1,1,1,1,3,3,3,3,3
```

(c) File structure

Feature coding of cells in the raster world

- Above figure shows how different land uses can be coded in a single raster layer.
- The values 1, 2 and 3 have been used to classify the raster cells according to the land use present at a given location.
- The value 1 represents residential area; 2, forest; and 3, farmland.

- One of the major problems with raster data sets is their size, because a value must be recorded and stored for each cell in an image.
- Thus, a complex image made up of a mosaic of different features (such as a soil map with 20 distinct classes) requires the same amount of storage space as a similar raster map showing the location of a single forest.
- To address this problem a range of data compaction methods have been developed.

****

2.6 RASTER DATA COMPRESSION

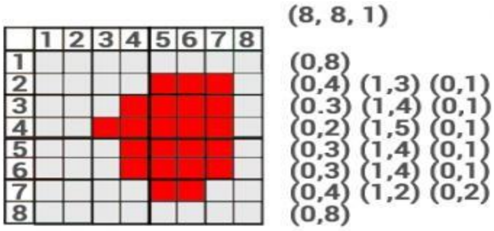
- Data compression refers to the reduction of data volume, a topic particularly important for data delivery and Web mapping. Data compression is related to how raster data are encoded.
- Quadtree and RLE, because of their efficiency in data encoding, can also be considered as data compression methods.
- A variety of techniques are available for data compression.
- They can be lossless or lossy. A lossless compression preserves the cell or pixel values and allows the original raster or image to be precisely reconstructed.
- Therefore, lossless compression is desirable for raster data that are used for analysis or deriving new data. RLE is an example of lossless compression.
- Other methods include LZW (Lempel—Ziv-Welch) and its variations (e.g., LZ77,LZMA).

2.6.1 Lossy compression

- A lossy compression cannot reconstruct fully the original image but can achieve higher compression ratios than a lossless compression.
- Lossy compression is therefore useful for raster data that are used as background images rather than for analysis. Image degradation through lossy compression can affect GIS-related tasks such as extracting ground control points from aerial photographs or satellite images for the purpose of georeferencing.

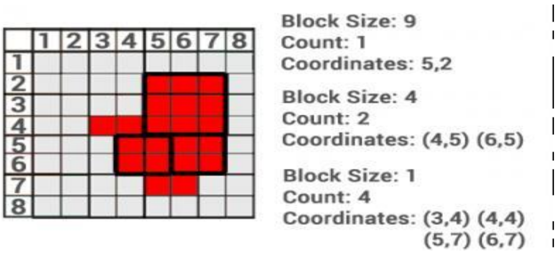
2.6.2 Run length encoding:

Run length encoding stores cells on a row-by-row basis. Instead of recording each individual cell's values, run length encoding groups cell values by row.



2.6.3 Block coding:

The block coding raster storage technique assigns areas that are blocks to reduce redundancy. The block coding raster image compression method subdivides an entire raster image into hierarchical blocks. It's an extension of the run length encoding technique, but extends it to two dimensions.



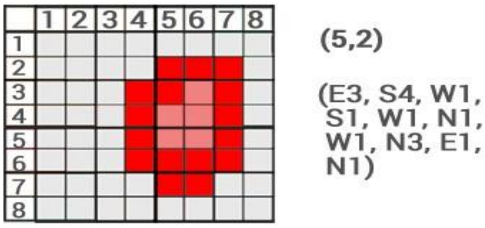
Block Size: 9
Count: 1
Coordinates: 5,2

Block Size: 4
Count: 2
Coordinates: (4,5) (6,5)

Block Size: 1
Count: 4
Coordinates: (3,4) (4,4)
(5,7) (6,7)

2.6.4 Chain Coding:

- Chain coding defines the outer boundary using relative positions from a start point. The sequence of the exterior is stored where the endpoint finishes at the start point.
- During the encoding, the direction is stored as an integer.
- However, in this example we use cardinal directions for simplicity.
- For example, the value 0 is north and 1 is east.

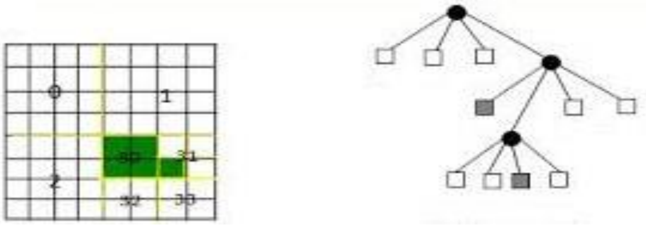


(5,2)

(E3, S4, W1, S1, W1, N1, W1, N3, E1, N1)

2.6.5 Quadtree encoding:

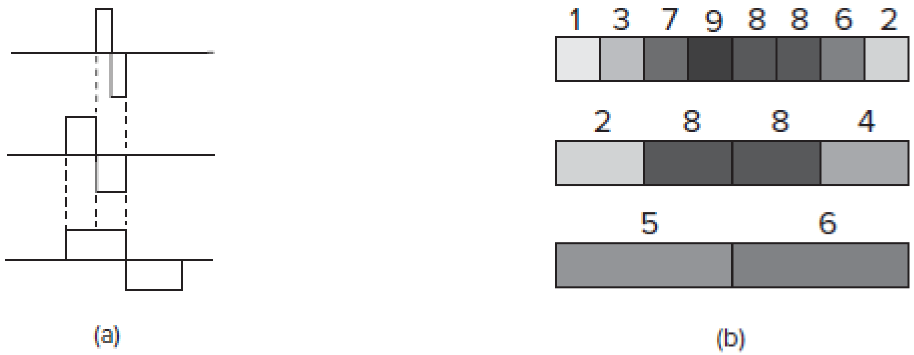
- Quadtrees are raster data structures based on the successive reduction of homogeneous cells.
- It recursively subdivides a raster image into quarters.
- The subdivision process continues until each cell is classed.



- MrSID uses the wavelet transform for data compression.
- The wavelet-based compression is also used by JPEG 2000 and ECW (Enhanced Compressed Wavelet).
- The wavelet transform treats an image as a wave and progressively decomposes the wave into simpler wavelets (Addison 2002).
- Using a wavelet (mathematical) function, the transform repetitively averages

groups of adjacent pixels (e.g., 2, 4, 6, 8, or more) and, at the same time, records the differences between the original pixel values and the average.

- The differences, also called wavelet coefficients, can be 0, greater than 0, or less than 0. In parts of an image that have few significant variations, most pixels will have coefficients of 0 or very close to 0.
- To save data storage, these parts of the image can be stored at lower resolutions by rounding off low coefficients to 0, but storage at higher resolutions is required for parts of the same image that have significant variations (i.e., more details). Box 4.4 shows a simple example of using the Haar function for the wavelet 3transform.



The Haar wavelet and the wavelet transform. (a) Three Haar wavelets at three scales (resolutions).

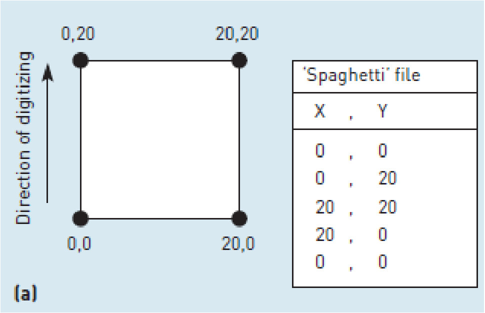
(b) A simple example of the wavelettransform.

****

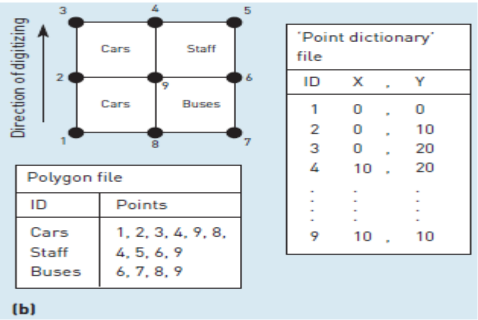
2.7 VECTOR DATA STRUCTURE

2.7.2 Introduction to Vector Data Structure

- There are many potential vector data structures that can be used to store the geometricrepresentation of entities in the computer.
- The simplest vector data structure that can be used to reproduce a geographical image in the computer is a file containing (*x*,*y*) co-ordinate pairs that represent the location of individual point features (or the points used to construct lines or areas).



'Spaghetti' file	
X ,	Y
0 ,	0
0 ,	20
20 ,	20
20 ,	0
0 ,	0



Polygon file

ID	Points
Cars	1, 2, 3, 4, 9, 8,
Staff	4, 5, 6, 9
Buses	6, 7, 8, 9

'Point dictionary' file

ID	X ,	Y
1	0 ,	0
2	0 ,	10
3	0 ,	20
4	10 ,	20
:	:	:
9	10 ,	10

Data structures in the vector world: simple

data structure

point dictionary

- The above figure shows such a vector data structure for the Happy Valley car park. Note how a closed ring of co-ordinate pairs defines the boundary of the polygon.
- The limitations of simple vector data structures start to emerge when more complex spatial entities are considered.
- For example, consider the Happy Valley car park divided into different parking zones (Figure: b).
- The car park consists of a number of adjacent polygons. If the simple data structure, illustrated in Figure: a, were used to capture this entity then the boundary line shared between adjacent polygons would be stored twice.
- This may not appear too much of a problem in the case of this example, but consider the implications for a map of the 50 states in the USA.
- The amount of duplicate data would be considerable.
- This method can be improved by adjacent polygons sharing common co-ordinate pairs (points).
- To do this all points in the data structure must be numbered sequentially and contain an explicit reference which records which points are associated with which polygon. This is known as a point dictionary.
- The data structure in Figure: b, shows how such an approach has been used to store data for thedifferent zones in the Happy Valley car park.
- There is a considerable range of topological data structures in use by GIS.

All the structures available try to ensure that:

δ) no node or line segment is duplicated;

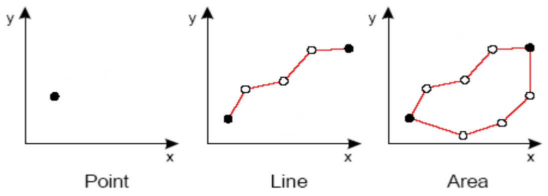
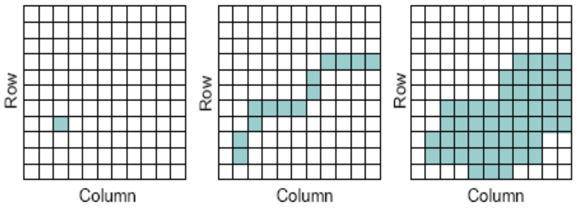
ε) line segments and nodes can be referenced to more than one polygon;

φ) all polygons have unique identifiers; and island and hole polygons can be adequately represented.

****

2.8 VECTOR AND RASTER MODELS

2.8.1 VECTOR vs RASTER:

Vector	Raster
Usually Complex.	Usually Simple.
Difficult for overlay operation.	Efficient for overlay operation.
High spatial variability is inefficiently represented.	High spatial variability is efficiently represented.
Small file size.	Large file size.
Vector data model is often used for representing discrete features with definable boundaries.	Raster data model is widely used for representing continuous spatial features.
Example: 	Example: 

****

2.9 TIN AND GRID MODELS

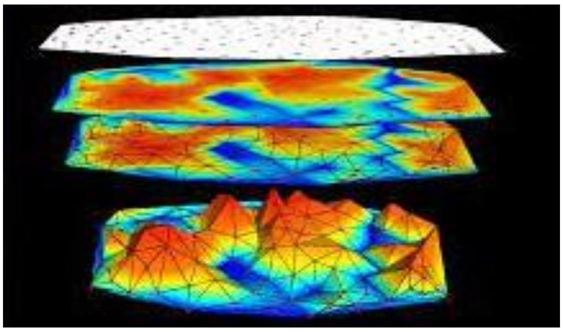
2.9.1 TINs MODEL – Triangular Irregular Networks Model

TINs – Triangular Irregular Networks – used to discrete continuous data.

- A Triangulated Irregular Network (TIN) (triangulate ['traıæ ŋgjʊleıt] verbo transitivo triangular) is a
- digital data structure used in a geographic information system (GIS) for the

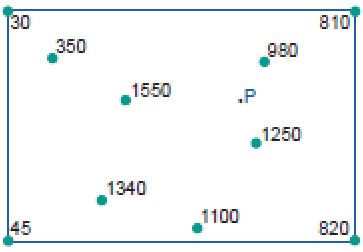
representation of a surface.

- TINs are often derived from the elevation data of a rasterized digital elevation model (DEM).
- Triangular irregular networks (TIN) have been used by the GIS community for many years and are a digital means to represent surface morphology.
- TINs are a form of vector-based digital geographic data and are constructed by triangulating a set of vertices (points).



Triangulated Irregular Networks (TIN) Model

- A commonly used data structure in GIS software is the triangulated irregular network (TIN). It is on the standard implementation techniques for digital terrain models, but it can beused to represent any continuous field.
- The principles behind a TIN are simple.
- It is built from a set of locations for which we have a measurement for instance an elevation.
- The locations can be arbitrarily scattered in space and are usually not on a nice regular grid. Any location together with its elevation value can be viewed as a point in three dimensional space.This is illustrated in below figure.
- From these 3D points, we can construct an irregular tessellation made of triangles.

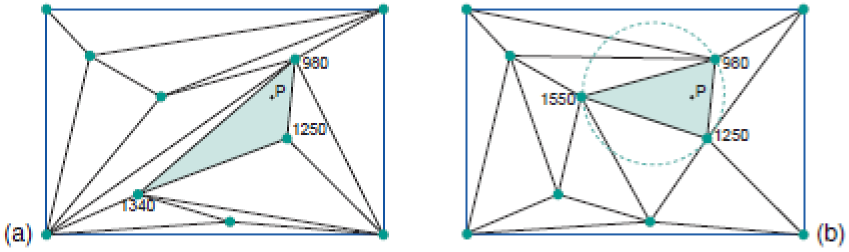


Input locations and their (elevation) values for a TIN construction.

- In three-dimensional space, three points uniquely determine a plane, as long as they not collinear, i.e. they must not be positioned on the same line.
- A plane fitted through these points has a fixed aspect and gradient and can be

used to compute an approximation f elevation of other locations.

- Since we pick many triples of points, we can construct many such planes and therefore we can have many elevation approximations for a single location such as `*P*`.
- So, it is wise to restrict the use of a plane to the triangular area between the three points.
- If we restrict the use of a plane to the area between its three anchor points, we obtain a triangular tessellation of the complete study space.
- Unfortunately, there are many different tessellations for a given input set of anchor points.
- Some tessellations are better than others, in the sense that they make smaller errors of elevation approximation.
- For instance, it we base our elevation computation for location `*P*` on the left hand shaded triangle, we will get another value than from the right hand shaded triangle.
- The second will provide a better approximation because the average distance from `*P*` to the three triangle anchors is smaller. The triangulation shown in below figure happens to be a Delaunay triangulation, which in a sense is an optimal triangulation.
- There are multiple ways of defining what such atriangulation is, but we suffice here to state two important properties.
- The first is that the triangles are as equilateral (‘equal-sided’) as they can be, given the set of anchor points.
- The second property is that for each triangle, the circumcircle through its three anchor points doesnot contain any other anchor point. One such circumcircle is depicted on the right of Figure(b).



Two triangulations based on the input locations (a) one with many ‘stretched’ triangles;

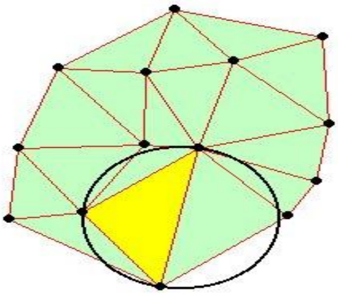
(a) the triangles are more equilateral – Delaunay triangulation.

- A TIN clearly is a vector representation: each anchor point has a stored georeference.
- Yet, we might also call it an irregular tessellation, as the chosen triangulation provides a partitioning of the entire study space.
- However, in this case, the cells do not have an associated stored value as is typical of tessellations, but rather a simple interpolation function that uses the elevation values

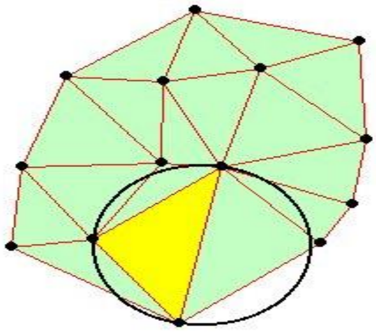
of its three anchor points.

Storing TINs

- There are basically two ways of storing triangulated networks:
- Triangle by triangle
- Points and their neighbors
- The first method is better for storing attributes (slope, aspect ..) for each triangle, but uses more storage space.
- The second one is better for generating contours and uses less storage space, but slope, aspect , etc. must be calculated and stored separately.

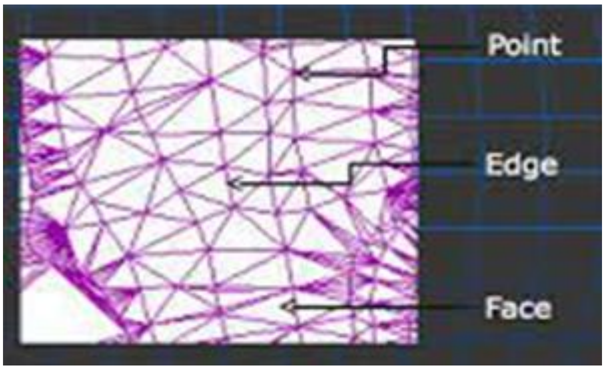


- The TIN model represents a surface as a set of contiguous, non-overlapping triangles.
- Within each triangle the surface is represented by a plane.
- The triangles are made from a set of points called mass points.
- Mass points can occur at any location, the more carefully selected, the more accurate the model of the surface.
- Well-placed mass points occur where there is a major change in the shape of the surface,
- for example, at the peak of a mountain, the floor of a valley, or at the edge (top and bottom) of cliffs.
- The TIN model is attractive because of its
- simplicity and economy and is a significant alternative to the regular raster of the GRID model.



Anatomy of TIN

- TIN = Triangulated irregular Network is connected three soundings to make a triangular ‘face’.
- The faces can then be used to represent a surface.



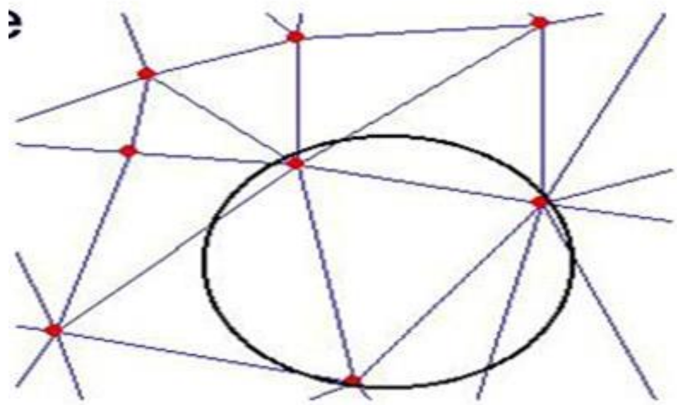
TIN structures is defined by two elements:

a) a set of input points with x,y and z values.

- Each input point becomes the node of a triangle in the TIN structure

b) a set of output points with x,y and z values

- The output is a continuous faceted surface of triangles.



Advantages :

- **ability to describe the surface at different level of resolution**
- **efficiency in storing data.**

Disadvantages :

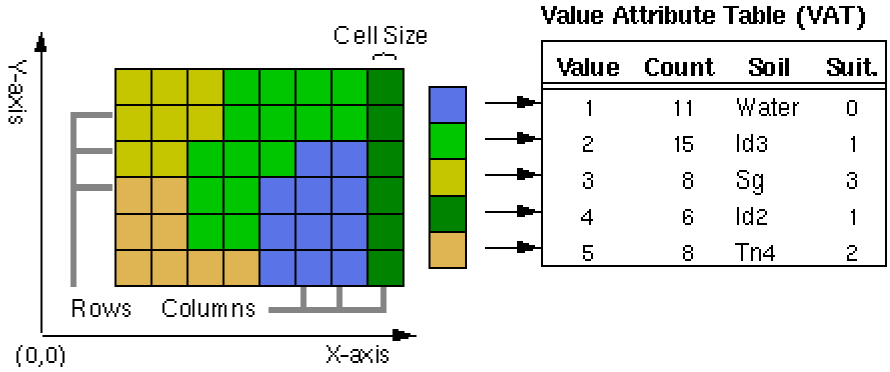
In many cases require visual inspection and manual control of the network.

2.9.2 GRID MODEL

What is grid model in GIS?

- GIS Dictionary. grid. [cartography] In cartography, **any network of parallel and perpendicular lines superimposed on a map and used for reference**.
- These grids are usually referred to by the map projection or coordinate system they represent, such as universal transverse Mercator grid. [data models] See raster.

- A grid is a raster data storage format native to Esri.
- There are two types of grids: integer and floating point.
- Use integer grids to represent discrete data and floating-point grids to represent continuous data.
- The cells in this type of grid do not fall neatly into discrete categories.

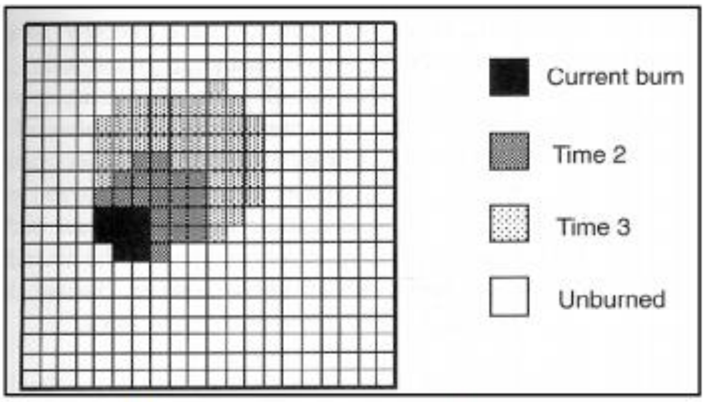


Value Attribute Table (VAT)

Value	Count	Soil	Suit.
1	11	Water	0
2	15	Id3	1
3	8	Sg	3
4	6	Id2	1
5	8	Tn4	2

Grid Analysis

- **Grid analysis**: involves the processing of spatial data in a special, regularly spaced form. The following illustration (figure 9) shows a grid-based model of fire progression.
- The darkest cells in the grid represent the area where a fire is currently underway.
- A fire probability model which incorporates fire behaviour in response to environmental conditions such as wind and topography delineates areas that are most likely to burn in the next two stages.
- These areas are represented by lighter shaded cells.
- Fire probability models are especially useful to fire fighting agencies for developing quick-response, effective suppression strategies.



A fire behaviour model delineates areas of fire progression based on a grid analysis.

****

2.10 GIS DATA STANDARDS

2.10.1 Introduction to GIS data Standards

- The number of formats available for GIS data is almost as large as the number of GIS packages on the market.
- This makes the sharing of data difficult and means that data created on one system is not always easily read by another system.
- This problem has been addressed in the past by including data conversion functions in GIS software.
- These conversion functions adopt commonly used exchange formats such as DXF and E00.

2.10.2 Open Geospatial Consortium (OGC)

- There is still no universally accepted GIS data standard, although the
- Open Geospatial Consortium (OGC), formed in 1994 by a group of leading GIS software and data vendors, is working to deliver spatial interface specifications that are available for global use (OGC, 2001).
- The OGC has proposed the Geography Markup Language (GML) as a newGIS data standard.
- The Geography Markup Language (GML) is a non-proprietary computer language designed specifically for the transfer of spatial data over the Internet.
- GML is based on XML (eXtensible Markup Language), the standard language of the Internet, and allows the exchange of spatial information and the construction of distributed spatial relationships.
- GML has been proposed by the Open Geospatial Consortium as a universal spatial data standard. GML is likely to become very widely used because it is:
  - Internet friendly;
  - not tied to any proprietary GIS;
  - specifically designed for feature-based spatial data;
  - open to use by anyone;
  - compatible with industry-wide IT standards.

It is also likely to set the standard for the delivery of spatial information content to PDA and WAP devices, and so form an important component of mobile and location-based (LBS) GIS technologies.

The collection of geoportals and various other compliemntary services, create a Spatial Data Infrastructure (SDI).

2.10.3 Spatial Data Infrastructure (SDI)

- An SDI is used to represent all the components that enable access to spatial data including relevant technologies, policies and institutional arrangements.
- Using electronic media, SDIs connect nationally distributed repositories of geospatial information and make them available on a device through a single entry point often referred to as a 'geoportal'.
- They facilitate data providers and users to participate in the digital spatial community at a national scale and provide a basis for spatial data discovery, evaluation and application for users within government, commercial and non-profit sectors, and academia and by citizens in general.
- The Global Spatial Data Infrastructure (GSDI) Association links national SDIs to establish a connection for all users in the world to share and reuse the available datasets.

Data Accuracy:

- In GIS, *data quality* is used to give an indication of how good data are. It describes the overall fitness or suitability of data for a specific purpose or is used to indicate data free from errors and other problems.
- Examining issues such as *error*, *accuracy*, *precision* and *bias* can help to assess the quality of individual data sets.
- In addition, the *resolution* and *generalization* of source data, and the data model used, may influence the portrayal of features of interest. Data sets used for analysis need to be *complete*, *compatible* and *consistent*, and *applicable* for the analysis being performed.
- Accuracy is the extent to which an estimated data value approaches its true value (Aronoff, 1989). If a GIS database is accurate, it is a true representation of reality. It is impossible for a GIS database to be 100 per cent accurate, though it is possible to have data that are accurate to within specified tolerances. For example, a ski lift station co-ordinate maybe accurate to within plus or minus 10 metres.
- Several types of error can arise when accuracy and/or precision requirements are not met during data capture and creation. The five types of error in a geospatial dataset are related to -

Positional Accuracy:

- The identification of positional accuracy is important. This includes consideration of inherent error (source error) and operational error (introduced error).

Attribute Accuracy:

- Consideration osf the accuracy of attributes also helps to define the quality of the data.
- This quality component concerns the identification of the reliability, or level of purity (homogeneity), in a data set.

Logical Consistency:

- This component is concerned with determining the faithfulness of the data structure for a data set.
- This typically involves spatial data inconsistencies such as incorrect line intersections, duplicate lines or boundaries, or gaps in lines. These are referred to as spatial or topological errors.

Completeness:

- The final quality component involves a statement about the completeness of the data set.
- This includes consideration of holes in the data, unclassified areas, and any compilation procedures that may have caused data to be eliminated.

****

2.11 DATA QUALITY

2.11.1 Introduction to Data Quality

What defines data quality?
Data quality is the measure of how well suited a data set is to serve its specific purpose.

Measures of data quality are based on data quality characteristics :

- accuracy,
- completeness,
- consistency,
- validity,

- uniqueness, and
- timeliness.

What is Data Quality?

- Data quality refers to the development and implementation of activities that apply quality management techniques to: data in order to ensure the data is fit to serve the specific needs of an organization in a particular context.
- Data that is deemed fit for its intended purpose is considered high quality data.

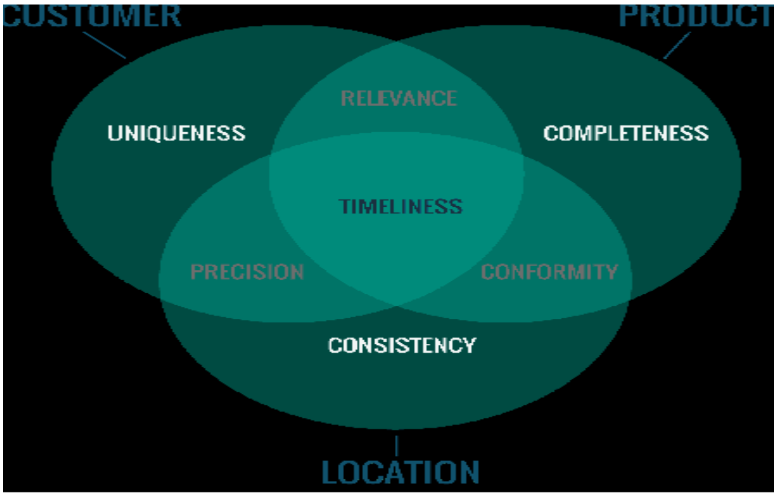
2.11.2 Data Quality Motive

- Data quality is the measure of how well suited a data set is to serve its specific purpose.
- Measures of data quality are based on data quality characteristics such as accuracy, completeness, consistency, validity, uniqueness, and timeliness.

Data Quality Issues

Examples of data quality issues include

- duplicated data
- incomplete data
- inconsistent data
- incorrect data
- poorly defined data
- poorly organized data and
- poor data security.



Why Data Quality is Important to an Organization?

- An increasing number of organizations are using data to inform their decisions regarding
  - marketing,
  - product development,
  - communications strategies and more.
- High quality data can be processed and analyzed quickly, leading to better and faster insights that drive business intelligence efforts and big data analytics.
- Good data quality management helps
  - extract greater value from data sets, and
  - contributes to reduced risks and costs,
  - increased efficiency and productivity,
  - more informed decision-making, better audience targeting,
  - more effective marketing campaigns,
  - better customer relations, and
  - an overall stronger competitive edge.

Poor data quality standards can cloud visibility in operations,

- making it challenging to meet regulatory compliance;
- waste time and labor on manually reprocessing inaccurate data; provide a disaggregated view of data,
- making it difficult to discover valuable customer opportunities;
- damage brand reputation; and even threaten the safety of the public.

2.11.3 Data Quality Improvement

Data quality measures can be accomplished with data quality tools, which typically provide data quality management capabilities :

1. Data profiling -

- The first step in the data quality improvement process is understanding your data.
- Data profiling is the initial assessment of the current state of the data sets.

2. Data Standardization -

- Disparate data sets are conformed to a common data format.

3. Geocoding -

The description of a location is transformed into coordinates that conform to U.S. and worldwide geographic standards.

4. Matching or Linking -

Data matching identifies and merges matching pieces of information in big data sets.

5. Data Quality Monitoring -

Frequent data quality checks are essential.

Data quality software in combination with machine learning can automatically detect, report, and correct data variations based on predefined business rules and parameters.

6. Batch and Real time -

Once the data is initially cleansed, an effective data quality framework should be able to deploy the same rules and processes across all applications and data types at scale.

2.11.4 Data Quality Processes and Evaluations

- Data quality assessments are executed by data quality analysts, who: assess and interpret each individual data quality metric,
- aggregate a score for the overall quality of the data, and provide organizations with a percentage to represent the accuracy of their data.
- A low data quality scorecard indicates:
- poor data quality, which is of low value, is misleading, and can lead to poor decision making that may harm the organization.
- Data quality rules are an integral component of data governance, which is:
- the process of developing and establishing a defined, agreed-upon set of rules and standards by which all data across an organization is governed.
- Effective data governance should harmonize data from various data sources, create and monitor data usage policies, and eliminate inconsistencies and inaccuracies that would otherwise negatively impact data analytics accuracy and regulatory compliance.

2.11.6 Data Quality vs Data Integrity

Data quality oversight is just one component of data integrity.

Data integrity refers to the process of making data useful to the organization.

The four main components of data integrity include:

1. **Data Integration:** data from disparate sources must be seamlessly integrated.

2. **Data Quality:** Data must be complete, unique, valid, timely, consistent, and accurate.

3. **Location Intelligence**: Location insights adds a layer of richness to data and makes it more actionable.

4. **Data Enrichment:** Data enrichment adds a more complete, contextualized view of data by adding data from external sources, such as customer data, business data, location data, etc

2.11.7 Data Quality Dimensions

By which metrics do we measure data quality?

There are six main dimensions of data quality:

a) accuracy

b) completeness

c) consistency

d) validity

e) uniqueness and

f) timeliness.

1. Accuracy:

The data should reflect actual, real-world scenarios; the measure of accuracy can be confirmed with a verifiable source.

2. Completeness:

Completeness is a measure of the data's ability to effectively deliver all the required values that are available.

3. Consistency:

Data values stored in difference locations should not : conflict with one another.

4. Validity:

- Data should be collected according to : defined business rules and parameters, and should conform to the right format and fall within the right range.

5. Uniqueness:

- Uniqueness ensures there are no duplications or overlapping of values across all data sets.
- Data cleansing and deduplication can help remedy a low uniqueness score.

6. Timeliness:

Timely data is data that is available when it is required.

Data may be updated in real time to ensure that it is readily available and accessible.

****