ADVANCES IN DATABASE

EMERGING DATABASE

The landscape of databases is evolving to accomodate new data types and complex data types/structures Thus there is emergence of new databases

1. ACTIVE DB : They react automatically to specific condition or events based on predefined rules they are particularly valuable for application requiring real time monitoring and automated responses.
   - Event driven : They automatically perform actions in response to events. These events could be data changes, user actions, or external triggers.
   - Triggers : They are predefined rules that specify actions to be taken when a specific event occurs. They are essential for automating tasks and enforcing business logic in real time
2. DEDUCTIVE DB : These make logical deductions based on rules or facts stored in DB They are suitable for application requiring reason and interference
   - Facts : Basic info stored in DB
   - Rules : Logical statements that define relation between facts.
   - Inference : The process of deriving new info from existing facts and rules.
3. MAIN MEMORY DB : These store all their data in main memory (RAM), offering ~~high~~ fast access time & high throughput well suited for application requiring real time processing.

- Fast Access : Storing data in RAM reduces access time. This allows faster data retrievals
- Real time processing : MMDB are best for real time processing & analysis
- High throughput : ability to handle large volume of data quickly and efficiently is a key advantage in MMDB making them suitable for high performance application.

4. SEMANTIC DB : These store data in a way that preserves relationship between different data points. They use ontologies, Resource Description Framework (RDF) to represent knowledge and facilitate complex queries.

Ontologies : They are formal representation of knowledge that define concepts and relationship within a specific domain. They provide structured way to represent information and facilitate knowledge sharing.

RDF : It is a standard data model in semantic DB for representing knowledge graphs. It allows defining relation between different entities and facilitates complex queries.

COMPLEX DATATYPE :

- Flexible Format
- Schema less or Evolving Schema.
- Self describing
- Heterogenous Structure.

NESTED DATA TYPE

1. XML :

Extensible Markup Language is a structured syntax for representing data in hierarchical tree like format.
It uses tags to define and organize elements within data.

Nesting elements:
It allows nesting, enabling multilayered data structures. This hierarchical organization makes it easy to model and process information.

Example XML.

```
<country>
    <city1>
        <neighborhood1>
        </neighborhood1>
        <neighborhood2>
        </neighborhood2>
    </city1>
    <city2>
        <neighborhood1>
        </neighborhood1>
    </city2>
</country>
```

2. JSON :

Javascript Object Notation, is a lightweight data interchange format that uses a compact syntax to represent structured data.

Flexibility:

JSON allows nesting of data elements, enabling the efficient representation of complex hierarchical information.

example:

```
{ [{"country":"value1",
      "states":["v1","v2",...],        }
      "cities": ["v1","v2",...],       } object 1
      "population": v1                 }
    }

    {"country":"value2",               }
          :                            } object 2.
          :                            }
    }]  - array of obj.
}
```

OBJECT ORIENTATION.

Object Relational Database System (ORDBMS)

It bridges gap between oop & ~~rela~~ RDBMS allowing seamless storage & retrieval of data.

This system allows ADT's to be stored in RDBMS.

This means user can define their own datatype in the obj that is stored in DB.

RDBMS. ORDBMS.

Customer..

Schema:

ID	First name	LastName	DOB

RDBMS:

```
Select InitCap( Firstname)||','||
       InitCap(Surname) ~~from~~.
       from Customer.as C.
       where Month((C.DOB) =
             Month(getdata())
       AND Day((C.DOB) =
           Day(getdata());
```

ORDBMS:

```
Select Formal (C.Id)
   From Customer as C
   where Birthday (C.DOB) =
         TODAY;
```

In ORDBMS we can also write functions like Birthday( ) for our data.

TABLE INHERITANCE

Single Table. (Table per Hierarchy).

It stores all entities super classes subclasses in single table.

Every entry has a unique marker known as discriminator.

This leads to lot of null value & sparse table.

Class Table (Table per type)

It include ~~all~~ single type of entity with the superclass and its subclasses.

each type of entity has seperate table.

Foreign keys are used to for associations. between tables.

Concrete Table

Assigns unique table for each subclass and derieved entity

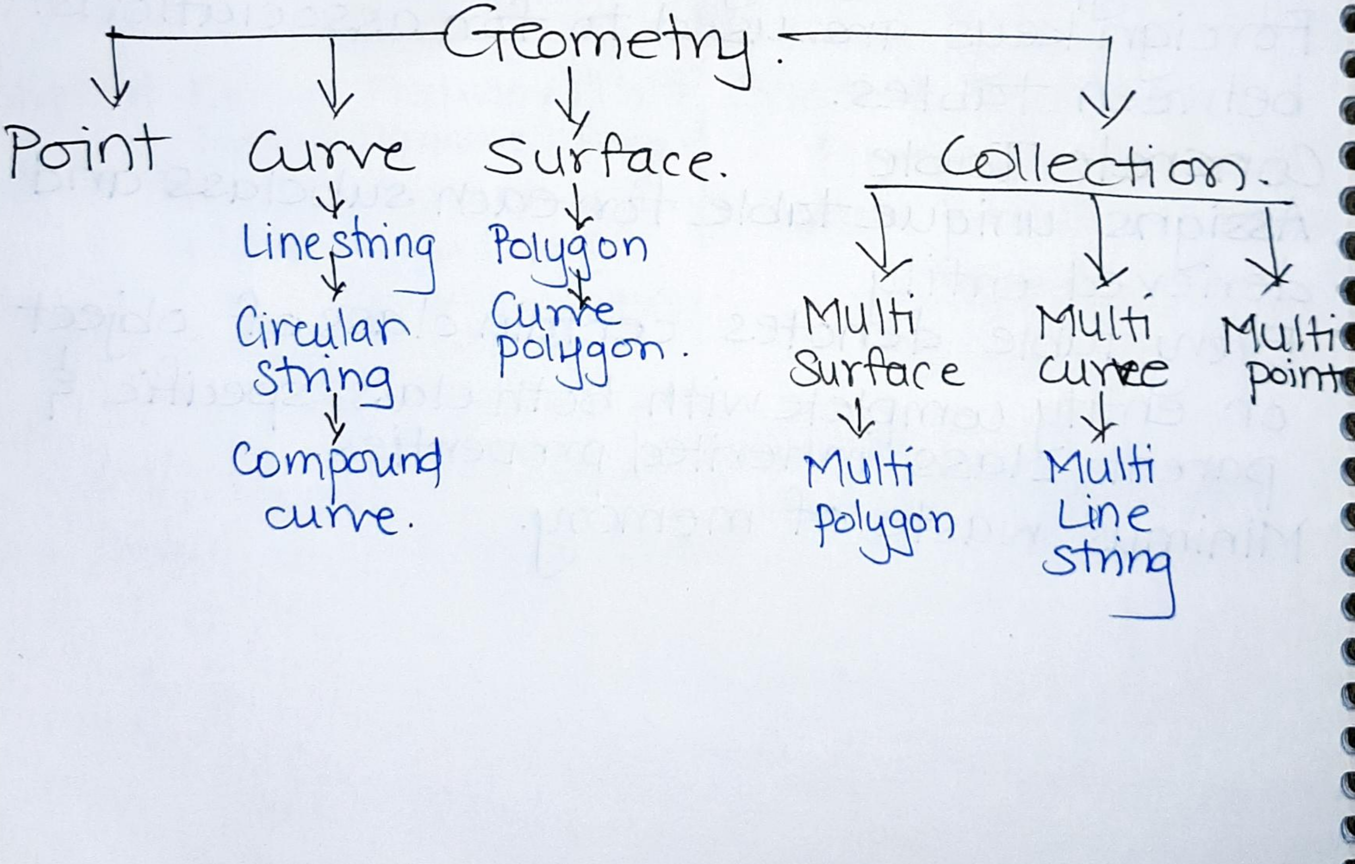
Every table denotes certain class of object or entity complete with both class specific & parent class inherited properties.

Minimus waste of memory.

SPATIAL DATA.

Geographic data:
Represents physical location and characteristics on earths surface.
example :(lat,lang.)

Geo~~geo~~metric data:
captures space, shape, size and orientation of object in spatial co~~r~~ordinates.
example : (x, y) 2D.
(x, y, z) 3D.


Geometry.
Point
Curve
Line string
Circular string
Compound curve.
Surface.
Polygon
Curve polygon.
Collection.
Multi Surface
Multi Polygon
Multi curve
Multi Line string
Multi point