Module 2 - Object-Based, Spatial and NoSQL databases.

* Object-Based DBMS

→ In traditional DBMS, we has common data types which are compatible with the relational data model, for example, {int, text, float etc}

→ Object-Based Databases have each entity treated as an object and represented in the table.

→ Similar objects are classified into classes and sub-classes and relationship b/w objects is maintained using inverse reference

* Features of object-Based DBMS.

① Object-oriented Data Model

They use an object-oriented data model which allows more natural and intuitive representation of data, objects can contain attributes, methods, relationships, making it easy to store complex data.

② Inheritance

Similar to OOPS, subclasses can inherit properties from the parent class, simplifying complex designs. Also allowing efficient retrival of data.

③ Polymorphism - OBDs allow the support of polymorphism, which means that an object can have multiple forms or types. This allows greater flexibility in representation of data and can handle data which changes frequently.

④ Extensible

Object-Based Databases can also be customized to meet specific needs of an application. This allows developers to add new features and functionality.

⑤ ~~Re~~ Encapsulation

OBD allows the data hiding concept in OOPL which binds the data and functions together and can manipulate data.

---

* Advantages of Object-Based Databases.

① They can provide a more natural way of representing data.

② Better performance than traditional DBMS as objects run as a single unit rather than seperate attributes.

* Disadvantages of object-Based Databases

① More complex to set up than regular DBMS.

② Not supported by some dev. tools and frameworks.

* Some Examples of Object-Based Databases

Suppose you are building a web-applica. for a small retail-store that sells clothing. You want to store information of Inventory, customers and staff.

→ You can create an Item class which represent any item from your store. It can have attributes like name, desc., price, quantity.

→ You can have a Customer class with properti like name, email, phone number and address

→ You can have an Order class with attributes like order number, cust. ID, total price etc.

* Database Design concepts of ORDBMS

→ ORDBMS stands for object-relational database management systems. It combines the concepts of regular DBMS with the concepts of OOPS (Object-oriented programming).

→ An object oriented data Model has 3 main components.

① Object structure - The structure of an object refers to the properties it possesses. These properties are called attributes. Also an object encapsulates multiple attribute and hides methods of implementation from user.

Structure is further divided into 3 types :-

a) Object Messages - A message provides an interface or acts as a communication medium blw object and outside world.

→ Read-Only Message
→ Update Message

b) Object Methods - When a message is passed then the body of code that is executed is known as a method.

Method has two types :-

① Read-Only Method.
② Update Method

③ Object Variables - Stores the data of an object. The data stored in the object makes them distinguishable and unique.

---

② Object classes :-

An object which is a real world entity is an instance of a class. A class needs to be defined before defining an object since class is a blueprint of an object

Example :-

```
class clerk
{
    char name;
    string address;
    int id;
    int salary;

    char getname();
    string get_address();
    int annual_salary;
};
```

* Nested Relations and Collections of ORDBMS.

* Nested Relations in ORDBMS

→ In ORDBMS, nested relations refer to the representation of complex data structures as nested tables or arrays within a single table

→ In nested relation, one or more columns of a table are table themselves, they have their own data, key, attributes and relationships.

→ For example, consider a database for a library. Main table can be 'books' and each row in books represents a single book in a library. One of the columns in the 'books' table can have a nested table of 'authors' which can have more columns like 'name', 'contact', 'age' etc.

→ Nested relations can be a useful way to represent complex relations b/w data and can reduce the need for complex joins and queries

→ Example → Nested Relations (SQL)

Assume table 'books' that contains a nested table called 'authors'

```
SELECT books.title, authors.name
FROM books, UNNEST(authors) AS authors.
```

→ This function allows us to elaborate the nested table into a set of rows and columns.

---

* Collections in ORDBMS

→ A collection in ORDBMS is used to store multiple values in a single variable.

There are four types of collections

① Arrays :- An array is a collection of same data type can be declare by keyword 'TYPE' and the 'IS TABLE OF'

```
TYPE int_array IS TABLE OF INTEGER;
```

2. Associative Arrays :- An associative array is a collection of key-value pairs, where each key maps to a single value. They can be declared using the keyword 'TYPE'.

```
TYPE int_string_arr IS TABLE OF INDEX BY
                    VARCHAR(200)
```

③ Nested Tables - A nested table is a collection of values of same data type similar to array, but it can be extended or shrunk dynamically.

```
TYPE nested_table IS TABLE OF INTEGER
```

④ VARRAYS (Variable ARRAYS) of max size during declaration. They can be declare

```
TYPE int_varray IS VARRAY(100) of Integer
```

→ Collections can be manipulate using PL/SQL syntax for FOR, WHILE Loops and IF statements.

* Spatial Databases.

→ Spatial databases are specialised databases designed to store and manage spatial data.

→ Spatial Data refers to data that is associate with specific locations, geographical coordinate maps, satellite imagery etc.

→ Spatial database holds complex data types like point, line, polygons and raster image

→ S.D also use complex algorithms which allows spatial indexing, allowing efficient spatial queries.

---

* Spatial Database Components

→ These are software tools or libraries that are designed to within a database managem system (DBMS).

→ Spatial DB components typically provide a set of functions which allow the enabling the data to be efficiently stored and processed.

Some major components are :-

① Spatial Indexing - Organizes the spatial data in a way that allows for efficient retrieval of subsets of data. Some techniques include R-trees, quad-trees and grid-based indexing.

② Spatial data types - These are specialized data types used to represent spatial data, for example point, line, polygon and multipoint data types.

③ Spatial query language - This is a special language used to query spatial data within the DBMS. Eg. SQL/MM Spatial and PostGIS

④ Spatial analysis functions :- Spatial analysis function allow for the manipulation of spatial data such as bufferring, clipping and spatial join operation.

⑤ Spatial data modelling - This component allows users to create and manage spatial data models that represent real world objects.

* Spatial Objects

→ These are specialized data types that are used to represent spatial data within a database.

→ These mainly include a combination of geometri primitives, such as points, lines, polygons as well as associated attributes like name, description, etc.

Some common spatial objects are:-

① Points - A spatial object that represents a single location in space, defined by x, y, z (x, y, z) coordinates.

② Lines - A spatial object that represents a connected sequence of points, such as road or river

③ Polygon - A spatial object that represents a closed shape by a series of connected lines such as a country or a building footprint.

④ Multi-attributes:- These are spatial objects that represent collection of points, lines and polygon.

* Spatial Dimensions

→ Spatial Dimensions in a database management System (DBMS) refers to the spatial properties or attributes associated with spatial data.

→ These are mostly methods of the attributes we saw in Spatial Objects.

→ Some common spatial dimensions are :-

① Coordinates - This is a set of numerical values that represent the location of spatial data, such as latitude and longitude. or UTM (Universal Transverse Mercator)

② Distance - This refers to the distance between two points and stored in meters or kilometres.

③ Area - This refers to the size of two-dimensional shape or a polygon spatial object in spatial data. Usually measured in m^2 or Km^2.

④ Volume - This refers to the size of a three-dimensional object like cube, cuboid, with a stack of multiple 2-D polygons. Measured in m^3 or Km^3.

→ Spatial Dimensions are critical components of Spatial Databases and used in a variety of location-based services. Some common Spatial DB are - Oracle Spatial, Postg and Microsoft Server SQL Spatial.

---

* Spatial Relations.

→ Spatial Relations in DBMS refers to the ~~relationship~~ relationships between Spatial objects.

→ Spatial Relations are based the following criterias :-

① Proximity :- This refers to the spatial relation between two objects based on their distance from each other. Example - Spatial Queries can be used to find all points within a certain distance of the object.

② Containment - This refers to the spatial relation when one object completely overlaps another or is placed inside another object. Example. A spatial query can be used to find all polygons within a big polygon.

③ Intersection - This refers to the spatial relation when two spatial objects share a same boundary or overlap in some way.

Example - A spatial query used to find all polygons which intersect a given line.

④ Adjacency - This refers to the spatial relationship between two objects based on whether they share a common boundary or at adjacent level with each other.

---

* Spatial queries.

→ Spatial Queries are used to retrieve and analyze spatial data stored in a spatial database

Here are some examples of spatial queries

① Retrieve all points in a given radius from a specific location

```
SELECT * FROM points
WHERE ST_Distance(geom, ST_MakePoint(lon, lat) < radius
```

② Get all points that intersect a given boundar box.

```
SELECT * FROM polygons
WHERE geom && ST_MakeEnvelope (minlon, maxlon,
                               maxlat, 4326);
```

③ All points located within a certain radius

```
SELECT* FROM points
WHERE ST_Within ( SELECT geom from polygons
                  WHERE id = polygon_id)
```

→ Some commonly used operators in Spatial Queries

① Named Spatial Relationships.

a) ST_ Contains()
b) ST_ Crosses()
c) ST_ Disjoint()
d) ST_ Equals()
e) ST_ Intersects()

② General Spatial relationships.

a) ST_ GeometryType()
b) ST_ containsProperly()
c) ST_ NumGeometries()

* NoSQL Databases

→ 'NoSQL' database technology stores information in JSON documents instead of columns and rows used by relational databases. This means that data can be analyzed and retrieved using 'no SQL'.

→ NoSQL provides more flexibility and scalability in terms of data analyzing and retrieval.

→ Most popular types of NoSQL are:-

① Document Databases :- are primarily built for for storing information as documents, but not limited to JSON documents. These systems can also be used for storing XML documents.

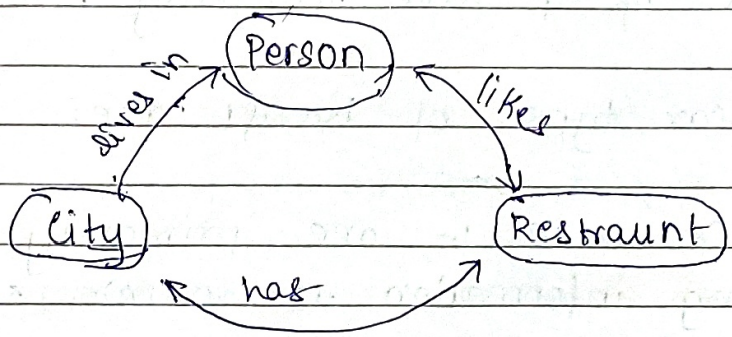
② Key : Value Mapping - Groups associated data in collections with records that are identified with unique keys for easy retrieval. They have a structure which mirrors the value of relational databases while preserving the NoSQL database structure.

③ Wide - column databases - Use the tabular format of relational databases yet allow a wide variance is how data is named and formatted in each row, even in the same table.

④ Graph Database - Uses graph structures to define relation b/w stored data points.

→ Graph databases are useful for identifying patterns in unstructured and semi-structured information.

→ Data is stored as nodes and relations as edges.



* Advantages of NoSQL

→ NoSQL has BigData capabilities

→ There is no single point of failure

→ Simple and easy to implement.

→ Easy Replication.

* Disadvantages of NoSQL

→ No rules for standardization.

→ Query capabilities are limited.

→ Lack of maturity

→ Doesnt work well with relational model.

* NoSQL Buisness Drivers.

→ There are 4 main drivers in NoSQL

① Volume - The key factor pushing alternatives is need for Big Data using clustering and commodity processor.

→ The need to scale out (horizontal scaling) rather than scale up (faster processor). moved organisations from serial to parallel processing.

② Velocity - The ability of a single processor system to rapidly read and write data is also key. Many single processors in RDBMS cannot keep up with demands of real time inserts and online queries.

(3) Variability - Companies that want to capture and report on exception data struggle when attempting to use rigid database schema structures.

Adding new columns requires database to shut down and ALTER TABLE commands.

(4) Agility - The most complex part of building applications is putting data in and retrieving data out.

O.R mapping layer has a major responsib. to select the correct DDL and DML command combination to move through the RDBMS persistence layer.