ADBMS_UNIT-6

1. Temporal Database

- **Definition**: A temporal database manages data with time-related attributes, allowing the storage and querying of data over time. It supports the concept of time directly, tracking data changes and enabling temporal queries.
- **Key Concepts**:
  - **Valid Time**: The period during which data is true in the real world (e.g., a job's duration).
  - **Transaction Time**: The time when data is stored or modified in the database.
  - **Bitemporal Data**: Combines both valid time and transaction time for comprehensive tracking.
- **Features**:
  - Time-based queries (e.g., "What was the price of a product last year?").
  - Supports historical data retention for compliance or analytics.
  - Can handle temporal constraints and joins.
- **Use Cases**:
  - Historical analysis, such as sales trends over time.
  - Legal compliance requiring data retention.
  - Insurance or medical records with time validity.

---

2. Spatial Database

- **Definition**: A spatial database is designed to store, query, and manage data that represents objects defined in a geometric or geographic space. It includes spatial types like points, lines, and polygons.

- **Key Concepts**:
  - **Spatial Data Types**: Geometric (shapes) and geographic (locations on Earth).
  - **Spatial Indexing**: Uses structures like R-trees for efficient querying of spatial relationships.
  - **Spatial Queries**: Includes proximity searches, intersections, and area calculations.
- **Features**:
  - Support for spatial joins and operations (e.g., finding locations within a radius).
  - Integration with mapping and GIS systems.
  - Handles large datasets with spatial indexing for performance optimization.
- **Use Cases**:
  - GIS (Geographic Information Systems) for urban planning or navigation.
  - Location-based services like finding nearby restaurants.
  - Environmental monitoring and disaster management.

---

3. Deductive Database

- **Definition**: A deductive database combines traditional relational database capabilities with rule-based inference to derive new information dynamically.
- **Key Concepts**:
  - **Facts**: Basic data stored in the database (e.g., "John is a doctor").
  - **Rules**: Logical statements that define relationships between facts (e.g., "If X is a doctor, X works in a hospital").
  - **Inference**: The process of deriving new facts using rules.
- **Features**:
  - Uses declarative query languages like **Datalog**.

- Allows reasoning over data, supporting complex queries and derived relationships.
- Ideal for knowledge-based systems requiring dynamic rule evaluation.

- **Use Cases**:
  - Expert systems in medical diagnosis or legal applications.
  - Artificial intelligence for reasoning and decision-making.
  - Logistics and planning systems requiring complex dependency analysis.

Comparison of Temporal, Spatial, and Deductive Databases

Aspect	Temporal Database	Spatial Database	Deductive Database
Primary Focus	Time-related data and changes over time	Geometric and geographic data	Logical reasoning and inference
Key Data Types	Temporal attributes (valid/transaction time)	Spatial types (points, polygons, etc.)	Facts and rules
Core Feature	Tracks historical and real-time data	Handles location-based queries and spatial relationships	Infers new information using rules
Query Language	Extended SQL for temporal queries	SQL with spatial extensions	Declarative logic-based languages (e.g., Datalog)
Applications	Financial, medical, and legal systems	GIS, navigation, urban planning	Expert systems, AI, and knowledge-based systems
Advantages	Enables historical and bitemporal analysis	Optimized for spatial indexing and operations	Supports dynamic reasoning and complex relationships
Challenges	Complexity in managing bitemporal data	High computational cost for spatial queries	Complexity in rule definition and performance

Mobile Database:

A Mobile Database is a type of database that can be accessed by a mobile network and connected to a mobile computing device (or wireless network). Here, there is a wireless connection between the client and the server. In the modern world, **Mobile Cloud Computing** is expanding quickly and has enormous potential for the database industry. It will work with a variety of various devices, including Mobile Databases powered by iOS and Android, among others. Couchbase Lite, Object Box, and other popular databases are examples of databases.

Mobile Database Environment has the Following Components:

- For storing the corporate and providing the corporate applications, a Corporate Database Server and DBMS is used.
- For storing the mobile data and providing the mobile application, a Remote Database and server are used.
- There is always a two-way communication link present between the Mobile DBMS and Corporate DBMS.

Features of Mobile Database:

- There are a lot of features of Mobile Database which are discussed below:
- As more people utilize laptops, smartphones, and PDAs to live on the go.
- To prevent frequent transactions from being missed due to connection failure, a cache is kept.
- Mobile Databases and the main database server are physically independent.
- Mobile gadgets hosted Mobile Databases.
- Mobile Databases can communicate with other mobile clients or a centralized database server from distant locations.
- Due to unreliable or nonexistent connections, mobile users need to be able to operate without a wireless connection with the aid of a Mobile Database (disconnected).
- Information on mobile devices is analyzed and managed using a Mobile Database.

Mobile Database Consists of Three Parties Which are Described Below:

- **Fixed Hosts:** With the aid of database servers, it handles transactions and manages data.
- **Mobile Units:** These are mobile, transportable computers, and the cell tower they utilize to connect to base stations is a part of that geographical area.
- **Base Stations:** These two-way radios, which are installed in fixed places, allow communication between the stationary hosts and the mobile units.

In many instances, a user may utilize a mobile device to log in to any corporate database server and deal with data there, depending on the specific requirements of mobile applications. While in other cases, the user can upload data collected at the remote location to the company database or download it and work with it on a mobile device. The interaction between the corporate and mobile databases is frequently intermittent and only occasionally establishes or establishes a link for a brief period of time.

Backward Skip 10sPlay VideoForward Skip 10s

Additional Functionalities of a Mobile DBMS Consist of the Following Capabilities:

- It should communicate to the centralized and primary database through different modes.
- On mobile devices and centralized DBMS servers, the data should be repeated.
- From the internet, capture the data.
- Mobile devices should be capable of dealing with that data.
- Mobile devices must analyze the data.
- Must create a personalized and customized application.

Limitations:

- There are a lot of limitations or drawbacks available, which are pointed out below:

- Its wireless bandwidth is restricted.
- It is very difficult work to make this database theft-proof.
- To operate this, we need unlimited battery power.
- Wireless communication speed suffers in mobile databases.
- In terms of security, it is less secure.

Multimedia Database

Multimedia database is the collection of interrelated multimedia data that includes text, graphics (sketches, drawings), images, animations, video, audio etc and have vast amounts of multisource multimedia data. The framework that manages different types of multimedia data which can be stored, delivered and utilized in different ways is known as multimedia database management system. There are three classes of the multimedia database which includes static media, dynamic media and dimensional media.

Content of Multimedia Database management system :

1. **Media data –** The actual data representing an object.
2. **Media format data –** Information such as sampling rate, resolution, encoding scheme etc. about the format of the media data after it goes through the acquisition, processing and encoding phase.
3. **Media keyword data –** Keywords description relating to the generation of data. It is also known as content descriptive data. Example: date, time and place of recording.
4. **Media feature data –** Content dependent data such as the distribution of colors, kinds of texture and different shapes present in data.

Types of multimedia applications based on data management characteristic are :

1. **Repository applications –** A Large amount of multimedia data as well as meta-data(Media format date, Media keyword data, Media feature data) that is stored for retrieval purpose, e.g., Repository of satellite images, engineering drawings, radiology scanned pictures.

2. **Presentation applications –** They involve delivery of multimedia data subject to temporal constraint. Optimal viewing or listening requires DBMS to deliver data at certain rate offering the quality of service above a certain threshold. Here data is processed as it is delivered. Example: Annotating of video and audio data, real-time editing analysis.
3. **Collaborative work using multimedia information –** It involves executing a complex task by merging drawings, changing notifications. Example: Intelligent healthcare network.

There are still many challenges to multimedia databases, some of which are :

1. **Modelling –** Working in this area can improve database versus information retrieval techniques thus, documents constitute a specialized area and deserve special consideration.
2. **Design –** The conceptual, logical and physical design of multimedia databases has not yet been addressed fully as performance and tuning issues at each level are far more complex as they consist of a variety of formats like JPEG, GIF, PNG, MPEG which is not easy to convert from one form to another.
3. **Storage –** Storage of multimedia database on any standard disk presents the problem of representation, compression, mapping to device hierarchies, archiving and buffering during input-output operation. In DBMS, a "BLOB"(Binary Large Object) facility allows untyped bitmaps to be stored and retrieved.
4. **Performance –** For an application involving video playback or audio-video synchronization, physical limitations dominate. The use of parallel processing may alleviate some problems but such techniques are not yet fully developed. Apart from this multimedia database consume a lot of processing time as well as bandwidth.
5. **Queries and retrieval –**For multimedia data like images, video, audio accessing data through query opens up many issues like efficient query formulation, query execution and optimization which need to be worked upon.

Areas where multimedia database is applied are :

- **Documents and record management :** Industries and businesses that keep detailed records and variety of documents. Example: Insurance claim record.

- **Knowledge dissemination :** Multimedia database is a very effective tool for knowledge dissemination in terms of providing several resources. Example: Electronic books.
- **Education and training :** Computer-aided learning materials can be designed using multimedia sources which are nowadays very popular sources of learning. Example: Digital libraries.
- Marketing, advertising, retailing, entertainment and travel. Example: a virtual tour of cities.
- **Real-time control and monitoring :** Coupled with active database technology, multimedia presentation of information can be very effective means for monitoring and controlling complex tasks Example: Manufacturing operation control.

Issues in Storing Multimedia Data in Databases

1. **Support for Large Objects**:

   Multimedia data, such as videos, can be several gigabytes in size, which many databases cannot natively handle. Solutions include:

   - Splitting large objects into smaller pieces for storage.
   - Storing multimedia objects in the file system with database pointers using standards like SQL/MED, enabling external data to appear as part of the database.

2. **Isochronous Data Retrieval**:

   Continuous-media data like audio and video require steady data delivery rates to prevent issues like sound gaps or data loss due to buffer overflows.

3. **Similarity-Based Retrieval**:

   Multimedia applications often need to retrieve similar data (e.g., fingerprint matches). Traditional index structures (e.g., B+-trees, R-trees) are insufficient, necessitating specialized indexing techniques.

Geographical Information Systems (GIS) - Detailed Explanation

A **Geographical Information System (GIS)** is a framework that integrates hardware, software, and data for capturing, managing, analyzing, and visualizing spatial or geographic data. It enables users to understand relationships, patterns, and trends in geographical contexts, supporting a wide range of applications.

Key Components of GIS

1. **Hardware**:
   - Physical devices used to collect, store, and process geographic data.
   - Examples:
     - Computers and servers for data storage and processing.
     - GPS devices for collecting location-based data.
     - Remote sensing tools like satellites and drones for capturing spatial data.
2. **Software**:
   - GIS software tools used to perform spatial analysis, mapping, and visualization.
   - Popular software includes:
     - **ArcGIS**: Comprehensive GIS solution with advanced analysis and cartography features.
     - **QGIS**: Open-source alternative for spatial analysis and map creation.
     - **Google Earth**: For 3D visualization of geographic data.
3. **Data**:
   - The core of GIS, consisting of two types:
     - **Spatial Data**: Represents the shape and location of geographic features (e.g., points, lines, polygons).

    - **Attribute Data**: Descriptive data about spatial features (e.g., population density, land use types).
  - Sources: Satellite imagery, surveys, sensors, and existing databases.

4. **People**:
   - GIS professionals like cartographers, data scientists, urban planners, and environmental researchers who analyze and interpret data for decision-making.
5. **Processes**:
   - Well-defined workflows for data collection, storage, analysis, and presentation to ensure accuracy and consistency in outputs.

Key Features of GIS

1. **Data Integration**:
   - Combines spatial data with attribute data for comprehensive insights.
   - Example: Linking rainfall patterns (spatial data) with crop yield statistics (attribute data).
2. **Spatial Analysis**:
   - Tools for analyzing spatial relationships, such as proximity, clustering, and patterns.
   - Example: Identifying high-risk flood zones by overlaying elevation maps with rainfall data.
3. **Visualization**:
   - Produces maps, graphs, and 3D models to represent geographic data.
   - Example: Visualizing urban expansion with time-series maps.
4. **Geocoding**:
   - Converts textual addresses into geographic coordinates for mapping.
   - Example: Locating customer addresses on a map for market analysis.
5. **Remote Sensing Integration**:

- Uses satellite imagery and aerial photography for real-time data updates.

Applications of GIS

1. **Urban Planning**:
    - Planning and designing urban infrastructure by analyzing population density, land use, and transportation networks.
    - Example: Designing metro routes based on population distribution.
2. **Environmental Management**:
    - Monitoring natural resources, tracking deforestation, and managing disasters.
    - Example: Mapping areas affected by forest fires or floods.
3. **Transportation and Logistics**:
    - Optimizing routes, analyzing traffic patterns, and planning new roads or railways.
    - Example: Identifying the shortest delivery routes for logistics companies.
4. **Public Health**:
    - Tracking disease outbreaks and planning healthcare facility locations.
    - Example: Mapping COVID-19 hotspots to allocate resources effectively.
5. **Business and Marketing**:
    - Site selection, market segmentation, and customer targeting based on geographic data.
    - Example: Identifying ideal locations for new retail stores.
6. **Agriculture**:
    - Precision farming, analyzing soil quality, and monitoring crop health.
    - Example: Mapping irrigation needs based on soil moisture data.

Benefits of GIS

1. **Enhanced Decision-Making**:
   - Provides spatial insights for strategic planning and problem-solving.
   - Example: Selecting the best location for a new hospital based on accessibility.
2. **Resource Management**:
   - Optimizes the use of natural resources by analyzing geographical data.
   - Example: Managing water resources in drought-prone areas.
3. **Disaster Management**:
   - Aids in planning and responding to natural disasters like hurricanes and earthquakes.
   - Example: Mapping evacuation routes in disaster-prone regions.
4. **Real-Time Monitoring**:
   - Tracks changes in the environment or infrastructure.
   - Example: Monitoring traffic congestion in real-time.
5. **Cost and Time Efficiency**:
   - Reduces costs and saves time by automating data collection and analysis processes.
   - Example: Using satellite imagery to replace manual field surveys.

Challenges in GIS

1. **Data Accuracy**:
   - Errors in spatial data collection or processing can lead to incorrect analyses.
2. **High Initial Costs**:
   - Setting up GIS infrastructure, including hardware and software, can be expensive.
3. **Complexity**:

- Requires skilled professionals for effective data analysis and interpretation.

4. **Data Integration**:

   - Combining data from various sources can be challenging due to differences in formats or projections.

Genome Data Management

Definition:

Genome data management refers to the processes and systems used to store, organize, analyze, and retrieve vast amounts of genetic information. With advances in genome sequencing technologies, managing genomic data has become essential for research, healthcare, and biotechnology.

Key Concepts in Genome Data Management

1. **Genome Data**:

   - Refers to the complete set of DNA sequences in an organism, including genes, non-coding regions, and regulatory elements.
   - Modern sequencing technologies generate data in terabytes or petabytes, requiring robust storage and analysis solutions.

2. **Data Characteristics**:

   - **Volume**: The massive scale of data due to high-throughput sequencing.
   - **Complexity**: Includes diverse data types, such as sequences, annotations, and metadata.
   - **Variability**: Genetic variations between individuals and species.

3. **Data Life Cycle**:

   - **Acquisition**: Data generation using sequencing technologies (e.g., Illumina, PacBio).
   - **Storage**: Efficient storage solutions like relational databases, NoSQL databases, or distributed file systems.

- **Processing**: Bioinformatics pipelines for cleaning, aligning, and assembling genome sequences.
- **Analysis**: Tools for identifying mutations, gene functions, or evolutionary patterns.

Components of Genome Data Management

1. **Storage Solutions**:
    - **Local Databases**: Relational databases like MySQL and PostgreSQL.
    - **Cloud-Based Systems**: AWS, Google Cloud, and Microsoft Azure for scalable storage and analysis.
    - **Specialized Genomic Repositories**:
        - **GenBank**: A public repository for genetic sequences.
        - **Ensembl**: Provides genome annotations and comparative genomics tools.
        - **UCSC Genome Browser**: For visualizing genomic data.
2. **Data Integration**:
    - Combining genomic data with other biological datasets like proteomics, metabolomics, or clinical data.
    - Tools like BioMart or integration platforms help link different data sources.
3. **Data Analysis Tools**:
    - **Genome Alignment**: Tools like BLAST, Bowtie, and BWA for comparing DNA sequences.
    - **Variant Calling**: GATK for identifying SNPs (Single Nucleotide Polymorphisms) and structural variations.
    - **Functional Annotation**: Tools like ANNOVAR and VEP for interpreting genetic variations.
4. **Privacy and Security**:

- Genomic data often contains sensitive information requiring compliance with regulations like HIPAA or GDPR.
- Data encryption and access controls are critical for protecting privacy.

Applications of Genome Data Management

1. **Personalized Medicine**:
    - Tailoring treatments based on individual genetic profiles.
    - Example: Identifying genetic predispositions to diseases like cancer or diabetes.
2. **Drug Discovery**:
    - Analyzing genome data to identify drug targets and biomarkers.
    - Example: Using CRISPR-based studies to develop gene therapies.
3. **Agriculture**:
    - Improving crop yields and livestock through genetic analysis.
    - Example: Genetically engineering drought-resistant crops.
4. **Evolutionary Studies**:
    - Understanding genetic variations and evolutionary history of species.
5. **Disease Research**:
    - Identifying mutations associated with genetic disorders.

Challenges in Genome Data Management

1. **Data Volume**:
    - Managing the ever-increasing size of genomic datasets.
2. **Data Standardization**:
    - Lack of uniform formats for sharing and processing data.
3. **Computational Requirements**:

- High-performance computing resources needed for processing and analysis.

4. **Privacy and Ethics**:
    - Ensuring secure storage and ethical use of genomic information.
5. **Integration Complexity**:
    - Combining genomic data with clinical and environmental datasets.

Active Database Concepts – Detailed Explanation

An **active database** is a database that not only stores and retrieves data but also has the ability to react automatically to predefined events or changes in the database. This behavior is achieved through the implementation of **triggers**, **rules**, and the **Event-Condition-Action (ECA)** model.

Core Concepts of Active Databases

1. Event-Condition-Action (ECA) Rules

ECA rules form the backbone of active databases. These rules specify the database's reactive behavior and define how it should respond to events.

- **Event**:

    The specific occurrence that activates the rule.

    Examples: Insert, update, delete operations, or external events like a time-based trigger.

- **Condition**:

    A logical test that determines whether the action should execute after the event occurs.

    Example: Check if an updated salary exceeds a certain threshold.

- **Action**:

    The task or operation the database performs when the event occurs, and the condition is satisfied.

Examples: Logging a change, sending a notification, or modifying another table.

ECA Model Workflow:

1. Detect an event.
2. Evaluate the condition.
3. Execute the action if the condition is true.

---

2. Triggers in Active Databases

Triggers are procedural codes or database objects that implement ECA rules. They are executed automatically in response to specific events.

Structure of a Trigger:

1. **Triggering Event**: Specifies what action will fire the trigger (INSERT, UPDATE, DELETE).
2. **Trigger Timing**: Defines when the trigger fires relative to the event:
    - **BEFORE**: Executes before the event.
    - **AFTER**: Executes after the event.
    - **INSTEAD OF**: Substitutes the event processing (commonly used for views).
3. **Condition**: A logical test to determine whether the action executes.
4. **Action**: The operation performed by the trigger, often written in procedural SQL or PL/SQL.

---

3. Rule-Based Systems

Active databases can use rule-based systems that go beyond triggers. These systems allow complex rules to be applied to broader database operations. For example:

- A rule might activate when a certain threshold of records is reached, even if no specific event occurs.

Features of Active Databases

1. **Automation**:
   - Automatically respond to changes, reducing manual intervention.
2. **Real-Time Responses**:
   - Provide immediate reactions to critical events, ensuring timely actions.
3. **Support for Complex Business Rules**:
   - Enforce business policies like auditing and maintaining data integrity.
4. **Continuous Monitoring**:
   - Constantly monitor database states and changes to ensure compliance and consistency.

Applications of Active Databases

1. **Auditing and Logging**:
   - Automatically track changes to sensitive data.
   - Example: Logging every deletion in a financial database.
2. **Notification Systems**:
   - Send alerts based on certain conditions.
   - Example: Notifying a manager when inventory drops below a threshold.
3. **Data Validation and Constraints**:
   - Enforce business rules by validating data during transactions.
   - Example: Prevent negative values in financial records.
4. **Replicating and Synchronizing Data**:
   - Maintain consistency between databases by replicating updates.
   - Example: Synchronizing a backup database in real time.
5. **Automated Maintenance**:
   - Automate database optimization tasks like purging old records.

Examples of Active Database Behavior

1. **Automatic Inventory Management**:
    - Event: Inventory level drops below a threshold.
    - Condition: Quantity < reorder level.
    - Action: Generate a purchase order or notify the procurement team.
2. **Fraud Detection**:
    - Event: A withdrawal transaction is logged.
    - Condition: Withdrawal > $10,000 without prior authorization.
    - Action: Freeze the account and notify the bank's fraud detection team.
3. **Data Consistency in Banking**:
    - Event: Account balance is updated.
    - Condition: Balance becomes negative.
    - Action: Reject the transaction and log an error.

Advantages of Active Databases

1. **Improved Efficiency**:
    - Automatically executes tasks, saving time and effort for users.
2. **Enhanced Data Integrity**:
    - Enforces rules and constraints at the database level.
3. **Real-Time Monitoring and Action**:
    - Responds to critical events immediately, enabling timely actions.
4. **Reduces Human Error**:
    - Automates repetitive tasks, minimizing manual mistakes.
5. **Simplifies Application Logic**:
    - Moves complex business rules from application code to the database.

Challenges of Active Databases

1. **Performance Overhead**:
   - Triggers and rules may slow down the system, especially with frequent events.
2. **Debugging Complexity**:
   - Errors in triggers or rules can be challenging to identify and fix.
3. **Maintenance Burden**:
   - Keeping track of multiple rules and triggers can be difficult in large systems.
4. **Portability Issues**:
   - Triggers and rules are often database-specific, reducing portability across systems.
5. **Unintended Consequences**:
   - Poorly designed triggers can lead to cascading effects or infinite loops.

Comparison of Active and Passive Databases

Aspect	Active Database	Passive Database
Event Handling	Automatically reacts to predefined events.	Requires user or application intervention.
Rules/Triggers	Implements ECA rules for automatic actions.	Does not support triggers or automatic rules.
Automation	Highly automated.	Relies on manual processes.
Applications	Auditing, notifications, enforcing constraints.	Simple storage and retrieval of data.
Performance	Can introduce overhead for frequent triggers.	No additional performance impact.