CIT 225 Notes – Database basics

DBMS -application software of a database a database management system (DBMS). Regardless of the database type, databases organize data in a structured format. The structured format requires software to manage its organization and both maintain and query the data.

Another reason for the existence of DBMS software is the need to access different data simultaneously by different users.

* **Hierarchical structures** act like inverted trees, where you have a root node that descends through nodes to leaf nodes. Leaf nodes are nodes without dependent nodes.
* **Networked structures** are like inverted trees but they also have pointers at each ordinary node. Pointers act like traffic cops, and they point requests for matching data to data that may exist on a different branch of the same inverted tree
* **Relational structures [RDMS]** create relations between tables. They do this by designating some set of columns in a table as a unique key. The unique keys let you connect rows from one table to rows from another table { Oracle, MySQL, DB2} { PostgreSQL and Oracle but also support objects, also act as ORDMS (object related database management systems). Common query lnag is SQL {structured query language}.
  + Most provide common query lang. Common language is *application-programming interface* (API).
  + While the API is critical to the success or failure of application software, its maintainability and flexibility is conjoined to the successful organization of data in tables within the database.
* **Object-oriented** structures group data into structures with accessor and mutator methods. An accessor method gets data from an object and is more commonly known as a getter. A mutator method assigns values to an attribute (or field) in an object and is more commonly known as a setter.
  + Object-oriented database management systems use (OODBMS) implement Object-oriented programming languages as primary interfaces. XQuery.
* **JSON** is an acronym that stands for *JavaScript Object Notation*. JSON structures are name and value pairs. The name can point to simple data, like a number or string; or a name can point to a complex data, like a collection or embedded object. { most DBMS support but Cassandra and MongoDB are designed for high concurrency and name-value pair management as well as allowing direct access to JSON objects.
  + Name-value pair databases do not use SQL.
* **MultiVersion Concurrency Control** **(MCC) (MVCC) :** concurrent operations to gain processing benefits unobtainable any other way are achieved through ACID-compliant guaranteed changes.
  + ACID { atomicity, consistency, isolation, and durability}
    - Atomically – everything or nothing happens in transaction
    - Consistency- data will change from one state to anther
    - Isolation – no other can see changes until made permanent
    - Durable – written to disk
* Processes and Thread
  + A process is a program running in a discrete or shared memory segment on a computer. When computer launches process control execution within scope of privileges granted by operating system.
  + Processes read and write information to memory in their *Process Control Block (PCB).*
  + A thread is a lightweight process, a process starts a thread within its memory space. //threads are part of process
    - . Threads may read and write their own information to their Thread Control Block (TCB). They may also read or write to the owning PCB.
* **HOW DOES IT WORK?**
  + - Client-Server Model:
    - Listener processes
    - Optimistic and pessimistic communication paradigms
    - Session management
    - Locking
    - Shared versus discrete memory
    - Repository management
  + Client-Server Model:
    - Interface mechanism. Database that maintains data is the server,
      * RDBMS provide client interfaces that support interactive and batch modes of operation.
    - Interactive mode- user types and sees result of the typing and execution of the commands on the users console.
    - Batch processing: user submits job and sees a log file that docs its execution.
    - Communicate directly with DBMS across network or operation system pipes.
      * Can be through Open Database connectivity (ODBC) or java database
    - Major database products provide command-line interfaces that work in both interactive and batch modes.
* Listener Processes
  + Part of server software, runs in background, listen on server port; virtually or logically addressed though with 1 process that can communicate with another process.
* Optimistic and Pessimistic Communication Paradigms
  + pessimistic connection requires a permanent channel between the client and server software
    - while connected software can
      * instruct server software to do many things
      * confirm if those things are preformed
      * prevent others from seeing changes until finished
      * implemented across server side pipe or network socket
    - IPC – interposes Communication is a server-side pipe. It manages how messages are sent and received.
    - Network sockets: 2 way chennels, rely on TCP/IP (transport control protocol/internet protocol).
  + Optimistic – instant messaging. Rely on UDP – user datagram protocol. Statless communication method
  + JDBC and ODBC may use optimistic but both can also support pessimistic
* Session management: client connects to server through listener process, optimistic may be short and pessimistic may be long
  + Manages #, resource access, and duration of sessions.
    - Session are working environment for users who are connected
* **Locking:** prevents users from changing data before first user completes work.
  + DBMS- lock rows, sets of rows, or table.
  + May prevent another from querying result in middle of change but always prevents another from changing data before finished.
  + COMMIT statement when they want to make their change(s) permanent
  + ROLLBACK statement when they want to undo their change(s).
* **Shared versus Discrete Memory:** 
  + Shared memory describes the process of creating a memory segment that you want to share with other processes.
    - Form of static marshaling: eliminates dynamic marshaling cost.
    - More consistent, contiguous, and manageable.
  + Discrete memory the process of creating a memory segment\ that you want to use exclusively by single program.
    - Seldom implemented, because of cost od dynamic marshaling.
* **Repository Management**
  + Sever-side component with 2 pricniple parts: data repository and set of programs that manage it.
    - Most dbms manage more than one repository.
  + Repository synonymous with database instance
  + Programs ensure acitvies comply with MVCC rules and ACID-compliance of operations.
    - Also ensure integrity of memory segaments, sessions, and resource locking.
    - Write log files allowing dbms step-back to undo operations and provide ability to revover.
  + More than single physical file: manages reading, writing, and controlling these files.