ASSIGNMENT 1

TITLE - Study of Open Source Relational Databases

Problem Statement - Study of Open Source Relational Databases : MySQL

Introduction

Definition:

A database management system (DBMS) is a general purpose system software facilitating each of the following (with respect to a database): defining: specifying data types, data organization, and constraints to which the data must conform constructing: the process of storing the data on some medium (e.g., magnetic disk) that is controlled by the DBMS manipulating: querying, updating, report generation Sharing: a database allows multiple users and programs to database concurrently. Protection & Security:protection from hardware and software crashes and security from unauthorized access

Importance:

Database systems have become an essential component of life in modern society, in that many frequently occurring events trigger the accessing of at least one database: bibliographic library searches, bank transactions, hotel/airline reservations, grocery store purchases, etc.

Capabilities of DBMS:

These are additional characteristics of dbms.

1. Controlling Redundancy: Data redundancy (such as tends to occur in the "file processing" approach) leads to wasted storage space, duplication of effort (when multiple copies of a datum need to be updated), and a higher likelihood of the introduction of inconsistency. a. On the other hand, redundancy can be used to improve performance of queries. Indexes, for example, are entirely redundant, but help the DBMS in processing queries more quickly. b. A DBMS should provide the capability to automatically enforce the rule that no inconsistencies are introduced when data is updated.

2. Restricting Unauthorized Access: A DBMS should provide a security and authorization subsystem, which is used for specifying restrictions on user accounts. Common kinds of restrictions are to allow read-only access (no updating), or access only to a subset of the data

3. Providing Persistent Storage for Program Objects: Object-oriented database systems make it easier for complex runtime objects (e.g., lists, trees) to be saved in secondary storage so as to survive beyond program termination and to be retrievable at a later time

4. Providing Storage Structures for Efficient Query Processing: The DBMS maintains indexes (typically in the form of trees and/or hash tables) that are utilized to improve the execution time of queries and updates. (The choice of which indexes to create and maintain is part of physical database design and tuning and is the responsibility of the DBA. a. The query processing and optimization module is responsible for choosing an efficient query execution plan for each query submitted to the system.

5. Providing Backup and Recovery: The subsystem having this responsibility ensures that recovery is possible in the case of a system crash during execution of one or more transactions.

6. Providing Multiple User Interfaces: For example, query languages for casual users, programming language interfaces for application programmers,

forms and/or command codes for parametric users, menu-driven interfaces for standalone users.

7. Representing Complex Relationships Among Data: A DBMS should have the capability to represent such relationships and to retrieve related data quickly.

8. Enforcing Integrity Constraints: Most database applications are such that the semantics (i.e., meaning) of the data require that it satisfy certain restrictions in order to make sense. Perhaps the most fundamental constraint on a data item is its data type, which specifies the universe of values from which its value may be drawn. (E.g., a Grade field could be defined to be of type Grade_Type, which, say, we have defined as including precisely the values in the set { "A", "A-", "B+", ..., "F" }. a. Another kind of constraint is referential integrity, which says that if the database includes an entity that refers to another one, the latter entity must exist in the database.

9. Permitting Inference and Actions Via Rules: In a deductive database system, one may specify declarative rules that allow the database to infer new data! E.g., Figure out which students are on academic probation. Such capabilities would take the place of application programs that would be used to ascertain such information otherwise. a. Active database systems go one step further by allowing "active rules" that can be used to initiate actions automatically.

10. Potential for enforcing standards: this is very crucial for the success of database applications in large organizations Standards refer to data item names, display formats, screens, report structures, meta-data (description of data) etc.

11. Reduced application development time: incremental time to add each new application is reduced.

12. Flexibility to change data structures: database structure may evolve as new requirements are defined.

13. Availability of up-to-date information – very important for on-line transaction systems such as airline, hotel, car reservations.

14. Economies of scale: by consolidating data and applications across departments wasteful overlap of resources and personnel can be avoided.

Traditional vs. more recent applications of databases:

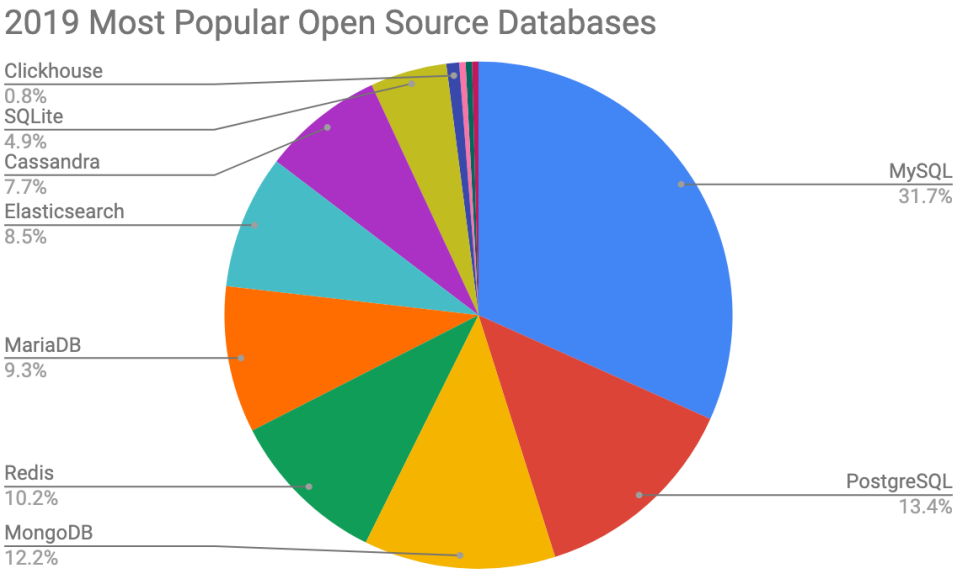
The applications mentioned above are all "traditional" ones for which the use of rigidly structured textual and numeric data suffices. Recent advances have led to the application of database technology to a wider class of data. Examples include multimedia databases (involving pictures, video clips, and sound messages) and geographic databases (involving maps, satellite

images). Also, database search techniques are applied by some WWW search engines.

MySQL-

MySQL is a popular open source relational database management system (RDBMS) choice for web-based applications. Developers, database administrators and DevOps teams use MySQL to build and manage next-generation web- and cloud-based applications.

The current version of MySQL, 8.0, offers a multitude of security improvements, including the removal of nonempty column values in the mysql.user table, policy-controlled automatic password expiration, user account locking and improved OpenSSL RSA certificate management. Other capabilities added in this version include SQL mode changes, online ALTER TABLE, enhanced and monitoring capabilities, expanded geographic information system for mobile applications, and many improvements to the InnoDB database engine.



As with most open source RDBMS options, MySQL is available in

several different editions and runs on Windows, OS X, Solaris, FreeBSD and other variants of Linux and Unix:

- **MySQL Classic Edition**, available to only independent software vendors, OEMs and value-added resellers, is designed to be an embeddable database for read-intensive applications.
- **MySQL Community Edition** is the free downloadable version of MySQL available under the GNU General Public License (GPL).
- **MySQL Standard Edition** is the entry-level RDBMS offering for online transaction processing applications.
- **MySQL Enterprise Edition** adds advanced features, management tools (including OEM for MySQL) and technical support.
- **MySQL Cluster Carrier Grade Edition** is designed for Web and cloud development.

MySQL Cluster is an open source, adaptable DBMS designed to deliver high performance and availability for many types of business applications. For example, MySQL Cluster powers the subscriber databases of major communications service providers and is used in global fraud detection for financial transactions.

The current version of MySQL Cluster, 7.4, offers features to automatically scale database services while maintaining five nines of availability. MySQL Cluster enables shared-nothing clustering of in- databases. It can be a viable

solution when you need ACID (atomicity, consistency, isolation and durability) transactions with high availability or very rapid data modification or insertion rates on inexpensive hardware.

MySQL uses a pluggable storage engine architecture that enables storage engines to be loaded into and unloaded from a running MySQL server. The storage engine handles the SQL operations for different types of MySQL tables. By swapping MySQL storage engines, you can change the behavior of MySQL.

InnoDB, a transaction-safe, ACID-compliant storage engine, is the default MySQL storage engine. Other storage engines include MyISAM (for Web, data warehousing), (for in-), CSV (for text), and Archive.