

Name: Meet Gandhi

Batch: T8

PRN: 2020BTECS00112

Module 1- Introduction to FOSS

1. Differentiate in between free software, open-source software and proprietary software with respect to its properties.
 - a. **Free Software:** “Free software” means software that respects users’ freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. The term “free software” is sometimes misunderstood - it has nothing to do with price. It is about freedom.
 - b. **Open-source software:** Open-source software is computer software whose source code is available openly on the internet and programmers can modify it to add new features and capabilities without any cost. Here the software is developed and tested through open collaboration. This software is managed by an open-source community of developers. It provides community support, as well as commercial support, which is available for maintenance. We can get it for free of cost. This software also sometimes comes with a license and sometimes does not. This license provides some rights to users.
 - The software can be used for any purpose
 - Allows to study how the software works
 - Freedom to modify and improve the program
 - No restrictions on redistribution
 - Some examples of Open-source software include Android, Ubuntu, Firefox, Open Office, etc.
 - c. **Proprietary Software:** Proprietary software is computer software where the source codes are publicly not available only the company that has created them can modify it. Here the software is developed and tested by the individual or organization by which it is owned not by the public. This software is managed by a closed team of individuals or groups that developed it. We have to pay to get this software and its commercial support is available for maintenance. The company gives a valid and authenticated license to the users to use this software. But this license puts some restrictions on users also like.
 - Number of installations of this software into computers
 - Restrictions on sharing of software illegally
 - Time period up to which software will operate
 - Number of features allowed to use
 - Some examples of Proprietary software include Windows, macOS, Internet Explorer, Google Earth, Microsoft Office, etc.
2. Enlist some examples along with its purpose and properties (at least 10) of FOSS and proprietary software with respect to database.
 - a. **MYSQL:**

Purpose – it is a database management system and uses SQL (sequential query language) for operations. It is a solid, quick, and dependable jack-of-all-trades.

Properties:

 - It is open source.
 - It is free.
 - Highly Scalable
 - Elastic Replication
 - Secure

b. MariaDB:

Purpose - MariaDB is a backward compatible, drop-in replacement of the MySQL® Database Server. It includes all major open-source storage engines. The MariaDB database is used for various purposes such as data warehousing, ecommerce, enterprise-level features, and logging applications.

Properties:

- Modern SQL and JSON support
- Oracle Database compatibility, continuous availability and comprehensive security.
- MariaDB's versatility stems from its pluggable storage engine architecture, with different storage engines optimized for different workloads.

c. PostgreSQL:

Purpose - PostgreSQL is a powerful, open-source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

Properties:

- Data Types
 - Primitives: Integer, Numeric, String, Boolean
 - Structured: Date/Time, Array, Range / Multirange, UUID
 - Document: JSON/JSONB, XML, Key-value (Hstore)
 - Geometry: Point, Line, Circle, Polygon
 - Customizations: Composite, Custom Types
- Data Integrity
 - UNIQUE, NOT NULL
 - Primary Keys
 - Foreign Keys
 - Exclusion Constraints
 - Explicit Locks, Advisory Locks
- Concurrency, Performance
 - Indexing: B-tree, Multicolumn, Expressions, Partial
 - Advanced Indexing: GiST, SP-Gist, KNN Gist, GIN, BRIN, Covering indexes, Bloom filters
 - Sophisticated query planner / optimizer, index-only scans, multicolumn statistics
 - Transactions, Nested Transactions (via savepoints)
 - Multi-Version concurrency Control (MVCC)
 - Parallelization of read queries and building B-tree indexes
 - Table partitioning
 - All transaction isolation levels defined in the SQL standard, including Serializable
 - Just-in-time (JIT) compilation of expressions
- Reliability, Disaster Recovery
 - Write-ahead Logging (WAL)
 - Replication: Asynchronous, Synchronous, Logical
 - Point-in-time-recovery (PITR), active standbys
 - Tablespaces
- Security
 - Authentication: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificate, and more
 - Robust access-control system
 - Column and row-level security
 - Multi-factor authentication with certificates and an additional method
- Extensibility
 - Stored functions and procedures
 - Procedural Languages: PL/pgSQL, Perl, Python, and Tcl. There are other languages available through extensions, e.g. Java, JavaScript (V8), R, Lua, and Rust
 - SQL/JSON path expressions
 - Foreign data wrappers: connect to other databases or streams with a standard SQL interface
 - Customizable storage interface for tables
 - Many extensions that provide additional functionality, including PostGIS

- Internationalisation, Text Search
 - Support for international character sets, e.g. through ICU collations
 - Case-insensitive and accent-insensitive collations
 - Full-text search

d. Redis (Remote Dictionary Server)

Purpose: Redis is a data structure server. At its core, Redis provides a collection of native data types that help you solve a wide variety of problems, from caching to queuing to event processing.

Properties:

- In-memory data store to ingest, process, and analyze real-time data with submillisecond latency.
- Redis delivers sub-millisecond response times, enabling millions of requests per second for real-time applications in industries like gaming, ad-tech, financial services, healthcare, and IoT.
- Redis employs a primary-replica architecture and supports asynchronous replication where data can be replicated to multiple replica servers.

e. SQLite:

Purpose: SQLite is an embedded, server-less relational database management system. It is an in-memory open-source library with zero configuration and does not require any installation. Also, it is very convenient as it's less than 500kb in size, which is significantly lesser than other database management systems.

Properties:

- SQLite is serverless as it doesn't need a different server process or system to operate.
- SQLite facilitates you to work on multiple databases on the same session simultaneously, thus making it flexible.
- SQLite is a cross-platform DBMS that can run on all platforms, including macOS, Windows, etc.
- SQLite doesn't require any configuration. It needs no setup or administration.

f. Neo4j:

Purpose: Neo4j is a graph database. A graph database, instead of having rows and columns has nodes edges and properties. It is more suitable for certain big data and analytics applications than row and column databases or free-form JSON document databases for many use cases. A graph database is used to represent relationships.

Properties:

- It is ACID (atomicity, consistency, isolation, and durability) compliant.
- It encourages versatility.
- Replicates information with quality and security.
- It works with Web applications for recovering chart information.
- It bolsters enquiry information sent out to JSON and XLS design.

g. OrientDB:

Purpose: OrientDB is an Open Source NoSQL Database Management System. NoSQL Database provides a mechanism for storing and retrieving NO-relation or NON-relational data that refers to data other than tabular data such as document data or graph data.

Properties:

- It is written in JAVA
- Zero Configuration Multi-master Architecture: This provides horizontal scalability and reliability.
- Multi-language support
- Multi-model support
- Zero delay start

h. CouchDB:

Purpose: CouchDB enables applications to store collected data locally on mobile devices and browsers, then synchronizes that data once it is back online.

Properties:

- An open source NoSQL document database that collects and stores data in JSON based document formats
- Replication: One of CouchDB's defining features is bi-directional replication, which enables synchronization of data across multiple servers and devices via bidirectional replication.
- Views: CouchDB uses views as the primary tool for running queries and creating reports from stored document files. Views allow you to filter documents to find information relevant to a particular database process.
- HTTP API: CouchDB uses a REST API to access the database from anywhere, with full CRUD (create, read, update, delete) operations flexibility.
- Built for offline: When you are scaling your database usability and accessibility, being able to build applications that work as well offline as they do online is essential.

i. FirebirdSQL:

Purpose: Firebird is a relational database offering many ANSI SQL standard features that runs on Linux, Windows, and a variety of UNIX platforms. Firebird offers excellent concurrency, high performance, and powerful language support for stored procedures and triggers.

Properties:

- It offers a small footprint, which is handy where space is premium.
- You can use FirebirdSQL as the database for a desktop app that needs to scale, such as LibreOffice.
- The database is cross-platform and has low needs in terms of hardware. This makes it a solid solution when you need to run large databases.

j. BigchainDB:

Purpose: BigchainDB is a blockchain database offering decentralization, immutability and native assets. BigchainDB allows for the deployment of large-scale applications in a variety of use cases and industries from intellectual property and identity to supply chain, and Internet-of-Things.

Properties:

- Scale (throughput, capacity, low latency)
- Queryability.
- Decentralized (no single entity owns or controls it)
- Immutable (tamper-resistance)
- Assets (you own the asset if you own the private key, aka blockchain-style permissioning).

3. Enlist some examples of free open-source exam software for online assessment.

- [TCExam](#)
- [VirtualX](#)
- [Moodle](#)
- [TAO](#)
- [Kaldin](#)
- [Papershala](#)
- [Edbase](#)
- [Mettl](#)
- [FlexiQuiz](#)
- [Eklavya](#)
- [Think Exam](#)

4. Demonstrate any one exam software which is open source and freely available.

FlexQuiz - Editor

flexquiz.com/Create/Edit/5e092d9-09a7-41a3-9f9c-414b3ecb0469

Google Walkhand College... Student login WCE Moodle ADQL Submission List All TV Series... yts-movie: YIFY Mo... Pep Font Awesome Course Downloader... Django server Strivers A2Z DSA C...

Dashboard Users Groups Reports Settings Upgrade Now

Create Configure Publish Analyze Print Preview

Quiz: Quiz 1 Status: In Design

Page: 1

Add Item -

Which batsman has highest ODI score in cricket?

☐ Chris Gayle

☐ Martin Gupit

☒ Rohit Sharma

☐ Pranav Hingankar

Which is a Group-7 element?

☐ Chlorine

☐ Beryllium

☐ Boron

☐ Oxygen

FlexQuiz - Khelo Dimaag Se

flexquiz.com/SC/5?ip=8563cf64-4843-4768-a563-db124f283f94&rg=t

Google DriveTCS Research & Inn...

Khelo Dimaag Se

Time remaining
0:04:51

1. Which language doesn't support pointers?

☐ Java

☐ C

☐ C++

☐ Pascal

2. Which is a Group-7 element?

☐ Chlorine

☐ Oxygen

☐ Beryllium

☐ Boron

3. Which batsman has highest ODI score in cricket?

☐ Rohit Sharma

Answered 0 of 3 (0%)

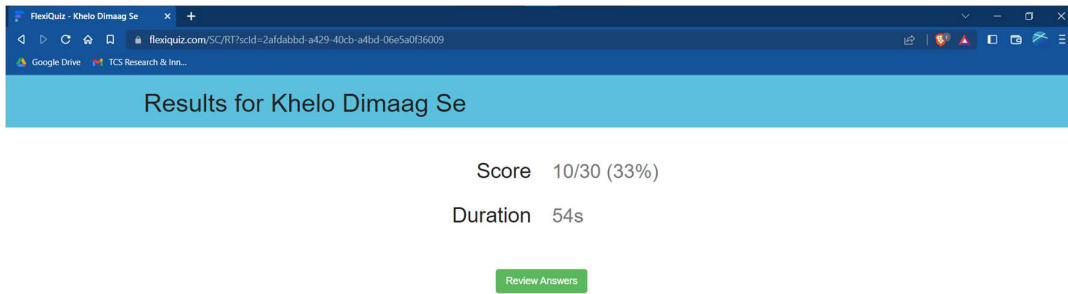
Praveen Bhoopalkar

Chris Gayle

Powered by FlexQuiz

Submit

Save and Continue



5. Demonstrate FOSS software related to database.

An open-source database is any database application with a codebase that is free to view, download, modify, distribute, and reuse. Open-source licenses give developers the freedom to build new applications using existing database technologies.

Opensource database management systems provide a layer of abstraction developers can use to store information for organizations and their applications.

Databases are typically categorized into two groups:

- Relational databases: The traditional data storage approach in which key-value pairs are used to store structured data into tables consisting of columns and rows.
- NoSQL (non-relational) databases: Data stored using alternative data storage architectures, including document data store, column-oriented database, key-value store, and graph databases. Non-relational databases are the preferred choice for handling unstructured data.

Database management systems give you the software layer you need to control and manage your data for a multitude of purposes. For example, you can store business intelligence in a relational database for fast SQL queries or save unstructured image files in a graph database for an AI-powered analytics app.

Closed source vs open-source databases

Closed source databases are proprietary software. The source code cannot be accessed, modified, distributed, or reused. You may have to pay a subscription or licensing fees to use the database within your applications. The company that wrote the code maintains the codebase. That means you'll have to wait for the company to add new features or address any bugs in the database management system. In contrast, anyone can view and access the source code for open-source databases. There are no licensing fees so the total cost of ownership (TCO) is lower for open-source databases than for commercial databases. You can download and modify source code to power your apps free of charge and without vendor lock-in. On the flip side, though, you're responsible for maintaining and securing your implementation of the open-source database.

Example of open-source softwares

Whether your goal is to store structured data for SQL queries or unstructured data via JSON objects, there are plenty of open-source database solutions to choose from on the web.

Examples of commonly used open-source relational databases include:

- MySQL
- PostgreSQL
- MariaDB

Examples of commonly used open-source NoSQL databases include:

- MongoDB
- CouchDB
- Cassandra

Best free and open-source database management systems and their type.

S. No.	Database Name	Type
1	PostgreSQL	Object-Relational Database Management System
2	MariaDB	Relational Database Management System
3	CockroachDB	Relational Database Management System
4	Neo4j	Graph Database Management System
5	CouchDB	NoSQL Database Management System
6	RethinkDB	Distributed Document-oriented Database Management
7	Redis	NoSQL Database Management System
8	SQLite	Relational Database Management System
9	Cassandra	NoSQL Database Management System
10	Timescale	NoSQL Database Management System

6. How does the Exam software work?

Popularly used by recruiters and educational institutions, exam software is used for setting up online exams. The best online examination software helps with the following procedures:

- Students' Registration** - Online exam software helps with the registration process of students and generates unique IDs for them.
- Test Paper Creation** - You can create a subjective, objective, multiple-choice, and other types of questions online and ensure zero spam.
- Take Tests Anytime, Anywhere** - Students can take tests from anywhere with a stable internet connection and a system. Similarly, teachers can invigilate directly through the system.
- Automated Evaluation** - Teachers don't need to evaluate answers manually, as the exam software helps analyze students' performance digitally.
- Track Students' Progress** - YouTube broadcast software enables users to list their live streams as videos on their channels. This way the live stream can be seen even after it ended.
- Data analysis** - The performance reports include detailed info about the strengths and weaknesses of every student. Accordingly, teachers can make the improvement plan.