**Database Tools**

Database tools are software applications or utilities designed to help users manage, manipulate, and work with databases effectively. Databases are essential for storing, retrieving, and managing large volumes of structured data. Database tools can assist with tasks such as creating, querying, designing, and maintaining databases.

**PhpMyAdmin**

phpMyAdmin is an open-source software tool introduced on September 9, 1998, which is written in PHP. phpMyAdmin supports various type of operations on MariaDB and MySQL. The main purpose of phpMyAdmin is to handle the administration of MySQL over the web.

It is the most popular application for MySQL database management. We can create, update, drop, alter, delete, import, and export MySQL database tables by using this software. phpMyAdmin also supports a wide range of operation like managing databases, relations, tables, columns, indexes, permissions, and users, etc., on MySQL and MariaDB. These operations can be performed via user interface, while we still have the ability to execute any SQL statement.

phpMyAdmin is translated into 72 languages and also supports both RTL and LTR languages so that the wide range of people can easily use this software.

phpMyAdmin is a GUI-based application which is used to manage MySQL database. It provides a web-based interface and can run on any server.

## **Features of phpMyAdmin**

* phpMyAdmin can create, alter, browse, and drop databases, views, tables, columns, and indexes.
* It can display multiple results sets through queries and stored procedures.
* phpMyAdmin use stored procedure and queries to display multiple results sets.
* It supports foreign keys and InnoDB tables.
* phpMyAdmin can track the changes done on databases, views, and tables.
* We can also create PDF graphics of our database layout.
* phpMyAdmin can be exported into various formats such as XML, CSV, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet.
* It supports mysqli, which is the improved MySQL extension.
* phpMyAdmin can interact with 80 different languages.
* phpMyAdmin can edit, execute, and bookmark any SQL-statements and even batch-queries.
* By using a set of pre-defined functions, it can transform stored data into any format. For example - BLOB-data as image or download-link.
* It provides the facility to backup the database into different forms.

## **Advantage of phpMyAdmin**

* phpMyAdmin can run on any server or any OS as it has a web browser.
* We can easily create, delete, and edit the database and can manage all elements using the **graphical interface of phpMyAdmin**, which is much easier than MySQL command-line editor.
* phpMyAdmin helps us to control the user's permission and operate several servers at the same time.
* We can also backup our database and export the data into different formats like XML, CSV, SQL, PDF, OpenDocument Text, Excel, Word, and Spreadsheet, etc.
* We can execute complex SQL statements and queries, create and edit functions, triggers, and events using the graphical interface of phpMyAdmin.

## **Disadvantage of phpMyAdmin**

* phpMyAdmin is a simple interface, but quite tough for a beginner to learn.
* phpMyAdmin is difficult to install as it needs three more software tools before installation, which is- **Apache server, PHP**, and **MySQL**.
* We have to install all these software tools individually, whereas XAMPP already contains them in a single package. XAMPP is the easiest way to get phpMyAdmin.
* It has no schema visualization.
* phpMyAdmin is a web-based software tool which runs only on the browser, so It completely depends on browsers.
* It does not have auto-compilation capability.

## **Data Backup problem with phpMyAdmin**

phpMyAdmin lacks a lot of features in import/export functionality. There are some backup problems with phpMyAdmin that are

* **Scheduling -** There is no way to export the data of the database in phpMyAdmin automatically.
* **Storage media support -** As we have discussed earlier, phpMyAdmin is web-based software, so it runs only on the browser. We can take backups only to local drives of our system.
* **Compression, Encryption, and other option -** The files which are exported with phpMyAdmin are saved as common text files, with any additional processing. Whereas storing these files in the original form usually takes a lot of disk storage.

**Indexing, Backing up and maintenance issues**

Indexing is a very useful technique that helps in optimizing the search time in database queries. The table of database indexing consists of a search key and pointer. There are three types of indexing namely Ordered, Single-level, and multi-level. Single Level Indexing is divided into three types namely Primary(index table is created using primary keys), Secondary(index table is created using candidate keys), and Clustered(index table is created using non-key values).

Backing up refers to the process of creating copies of data, applications, and system configurations to safeguard against data loss, hardware failures, disasters, or other unexpected events. Backups serve as a safety net to recover information and restore systems to a working state in case of problems.

**Data Loss Prevention**: The primary purpose of backups is to prevent data loss. Whether due to hardware failures, human errors, malware attacks, or natural disasters, having up-to-date backups ensures that critical data can be restored.

* **Types of Backups**: There are various backup strategies, including full backups (copying all data), incremental backups (copying only changed data since the last backup), and differential backups (copying all changes since the last full backup). Choosing the right backup strategy depends on factors like data volume, recovery time objectives, and available storage.
* **Backup Storage**: Backup data should be stored in a secure and redundant manner. This often involves offsite or cloud storage to protect against physical damage or theft of on-premises backups.
* **Frequency**: The frequency of backups depends on the data's importance and the acceptable level of data loss. Critical data may require real-time or daily backups, while less critical data may be backed up less frequently.
* **Testing:** Regularly testing backups is essential to ensure they can be successfully restored. Without testing, you may discover that your backups are ineffective when you actually need them.
* **Maintenance Issues:**
* Maintenance encompasses a wide range of activities to keep systems, software, and data in good working condition. It involves routine tasks, updates, and optimizations. Here are some maintenance issues and considerations:
* **Software Updates**: Regularly applying software updates, patches, and security fixes is crucial to address vulnerabilities and improve system stability. However, updates can sometimes introduce compatibility issues or require system downtime.
* **Hardware Maintenance**: Hardware components, such as disks, memory, and processors, require periodic checks and replacements to prevent failures that can lead to downtime.
* **Database Maintenance**: Databases need ongoing maintenance to optimize performance. Tasks include index rebuilding, vacuuming, and archiving old data. Failure to maintain databases can result in slow query performance and data corruption.
* **Security Maintenance**: Security is an ongoing concern. Regularly updating and monitoring security measures, such as firewalls, intrusion detection systems, and antivirus software, is essential to protect against cyber threats.
* **Data Cleanup:** Over time, data can become cluttered with obsolete records, duplicate entries, or outdated information. Regular data cleanup and data quality checks are necessary to maintain data integrity and ensure efficient data processing.
* **Backup Maintenance**: Backups should be regularly tested and verified for integrity. Backup procedures and policies may need updates as systems and data evolve

**Connecting Databases with PHP**

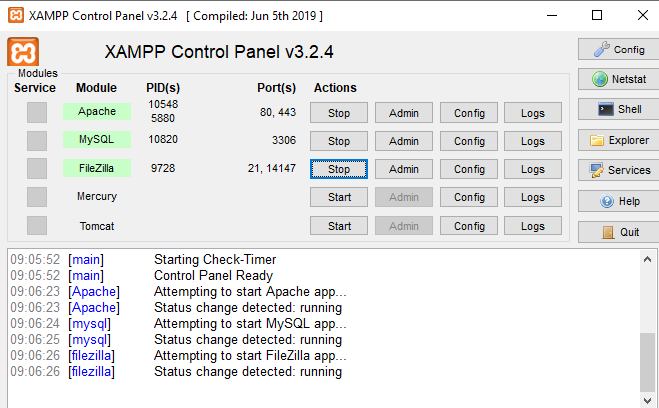
The collection of related data is called a database. XAMPP stands for cross-platform, Apache, MySQL, PHP, and Perl. It is among the simple light-weight local servers for website development.

**PHP Database connection**

* Start XAMPP server by starting Apache and MySQL.
* Write PHP script for connecting to XAMPP.
* Run it in the local browser.
* Database is successfully created which is based on the PHP code.

In PHP, we can connect to the database using XAMPP web server by using the following path.

**"localhost/phpMyAdmin”**



Now open your PHP file and write your PHP code to create database and a table in your database.

PHP code to create a database:

// Server name must be localhost

$servername = "localhost";

// In my case, user name will be root

$username = "root";

// Password is empty

$password = "";

// Creating a connection

$conn = **new** mysqli($servername,

            $username, $password);

// Check connection

**if** ($conn->connect\_error) {

**die**("Connection failure: "

        . $conn->connect\_error);

}

// Creating a database named geekdata

$sql = "CREATE DATABASE geekdata";

**if** ($conn->query($sql) === TRUE) {

    echo "Database with name geekdata";

} **else** {

    echo "Error: " . $conn->error;

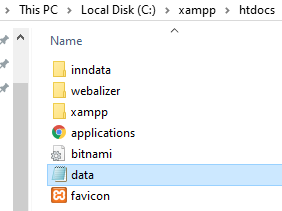
}

// Closing connection

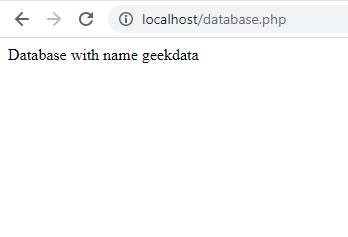
$conn->close();

?>

Save the file as “data.php” in htdocs folder under XAMPP folder.



Then open your web browser and type localhost/data.php



Finally, the database is created and connected to PHP.

If you want to see your database, just type *localhost/phpmyadmin* in the web browser and the database can be found.

