International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

ADMIN PANEL FOR HETEROGENEOUS DATABASES

Tanvi Shetty, Mithil Poojary, Rohan Sawant and Bushra Shaikh

Department of Information Technology, SIES Graduate School Of Technology, Nerul 400706, India

Abstract

Information systems are integral parts of an organization that plays a vital role in the management of its activities. They help the organization to meet its goals and objectives by delivering output efficiently. Raw material information is collected in a database to be used for processing and storage. The Database Management system is software designed to manage the data effectively and efficiently. End-users can use a DBMS to create, protect, read, update, and delete data in a database. It provides a centralised view of data that can be accessed in a controlled manner by multiple users from multiple locations. It presents a new and simple method for integrating different databases on browser-based support that is easier to manage. The proposed Database Management System will also include backup capabilities to help in database recovery in the event of a failure.

Keywords: Database Management System, Database, SQL, Python, Flask.

1. Introduction

Database is one of the vital features of developing an application. Database Management System refers to an interface between application and data files. DBMS relieves the end-user from the task of understanding where and how the data is stored. Almost every software application makes use of a Database System. But to make use of the database application one needs to set it up by installing which hogs up memory causing slow boot times. Also, it targets a single platform for Database Management Systems. It is not only difficult to set up but also a bit slow. To overcome these problems, the proposed system is hosted on a website, made available as software as a service that would be accessible via a web browser. Since the database is browser-based, it doesn't need to install an application to run. Hence would save up memory. The system is very easy to use and can build databases from scratch. It features a clean and simple UI. It would allow us to easily manage and display all the app's features like menus, icons, tabs, and popups in a natural way. This comprehensive package provides professionals with the necessary tools to manage and maintain databases. It includes a variety of features that allow users to develop and maintain multiple databases like MySQL, PostgreSQL, MariaDB, etc.

2. Literature Review

The below research papers are thoroughly read to have a deeper understanding of the implementation of the project:

B.KiranKumar et al. [1] described how Database Management System simplifies the work of database administrators (DBAs) by allowing them to focus on more important tasks. Due to the increasing complexity of searching for text, the need for better performance and more flexible capabilities has prompted the development of search engines that can handle large amounts of data. This paper shows a new way to compare the performance of different systems when storing the full-text index of a given file system.

Fankar Armash Aslam et al. [2] talks about the advantages of using Python in web development and how python is a great choice for web development as it simplifies the work of structuring data transactions. Also, it provides a powerful framework to implement web apps with ease.

Azhar Susanto et al. [3] explained the database as an integral part of a system. The paper also discusses functions that a DBMS can perform to help maintain and improve the integrity and efficiency of a database. These functions help in ensuring that the various components of the system are working seamlessly and providing the necessary features and capabilities to support the various needs of the users.

Antonio Badia [4] focussed on the role of Entity-Relationship model. The E-R model provides us with the structural information necessary to identify normal forms and the correctness of the schema database. A detailed study on relation constraints, attribute constraints, and characteristics of ER models through examples.

Abhijit Banubakode et al. [5] talked about exploring a scheme that allows optimizing queries over object-oriented databases with encapsulated behavior. Objects can reveal the structural access paths that are used by various types of procedures and cost structures for performing query optimization.The main feature of their approach is that the object-oriented user interface language can perform general computation and to preserve the encapsulation envelope around classes and types.

Ronald Fagin, Benny Kimelfeld et al.[6] threw light on the difficulty of translating abstract notions like concept and interpretation into a concrete search algorithm that operates over the auxiliary database. This paper proposes a formal framework for approaching the complexity of search algorithms, which can be easily implemented in a search database system.

Saurabh Gupta et al.[7] discussed today's relational database management system, the query processor and optimizer are key components. This section is in charge of converting a user query – usually written in a non-procedural language such as SQL – into an effective query evaluation program that can be run against the database. This paper provided a thorough examination of query optimization techniques.

Cornelia A. Gyorödi et al. [8] explained different techniques of optimization like reducing the length of the field, changing the datatype and using indexes. Also, CRUD operations were performed to compare the performances before and after applying optimization techniques.

Vangala Rama Vyshnavi et al. [9] emphasized how python has the potential to be more efficient for backend development. With the help of the flask Template Engine, it is easier to retrieve data from WWW. powerful.

3. Proposed system

A database administration user interface is responsible for allowing the users to create, update, delete & manage the databases. The objective is to support the most commonly used databases out there. The App was able to support Postgre, MySQL, MariaDB, and SQLLite databases. The App is also scalable as it uses the distributed architecture. Some of the features of the system are listed below:

- DBMS can create a database by providing an existing database name or creating a new database. It can also connect to the remote database..
- It also provides the users with the possibility to update the database. To update the database, the user should provide the database name, the table, and the values.
- One would be able to delete the database by providing the name of the database to be deleted.
- It will also provide the users with the ability to create a backup of the database by providing the database name, table, and credentials.
- A unique feature of the system would be the ability to draw Entity-Relationship Model.

4. Architectural Overview

The high level overview of the architecture for the Advanced Database Management System explained in this paper is explained here.

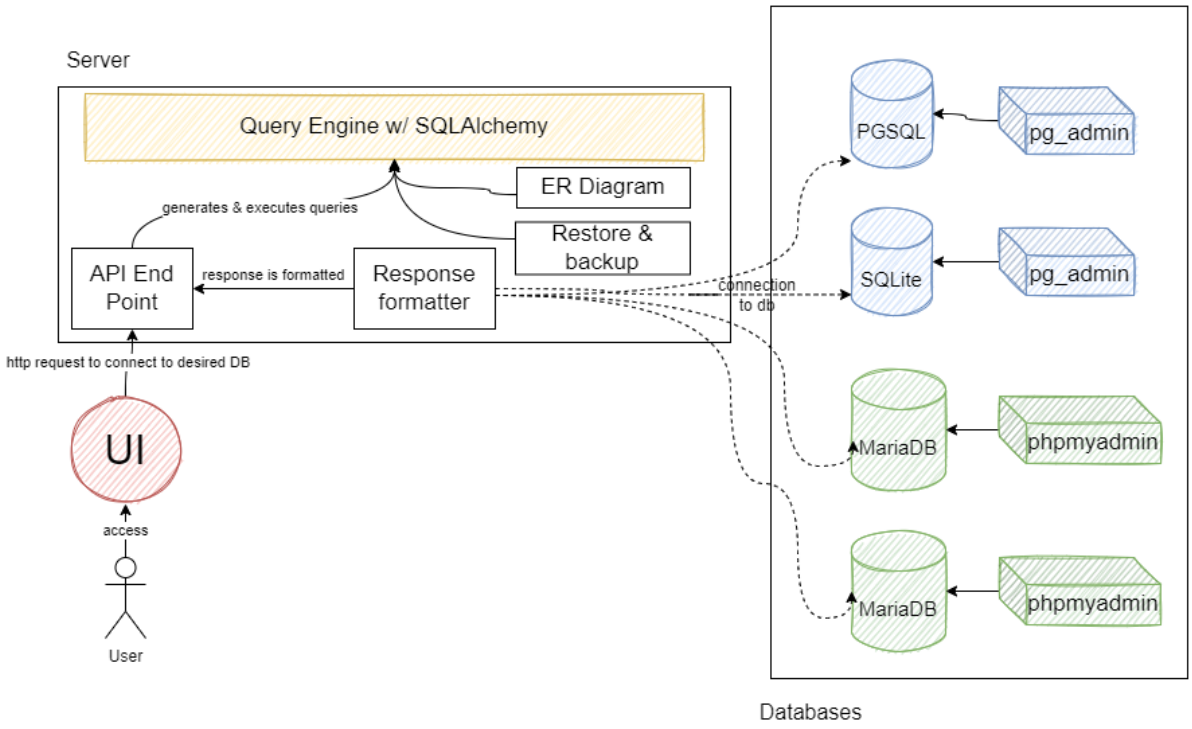


Figure 1. architectural overview of the system.

Before the ADBMS system explained in this paper, the user would have needed to use 3-4 different systems & UI to browse through all the different data sources which means, there was a lack of unified functionality through the different UIs.

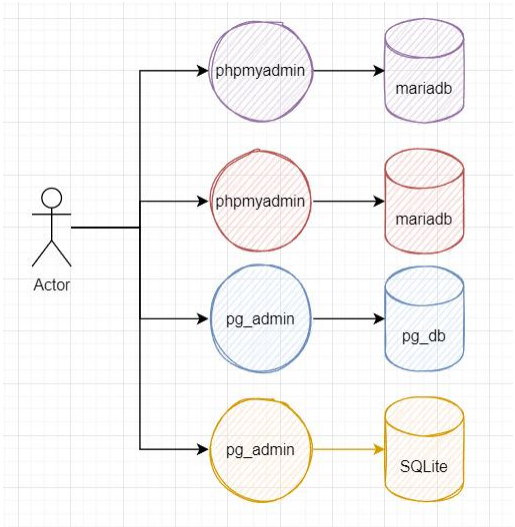
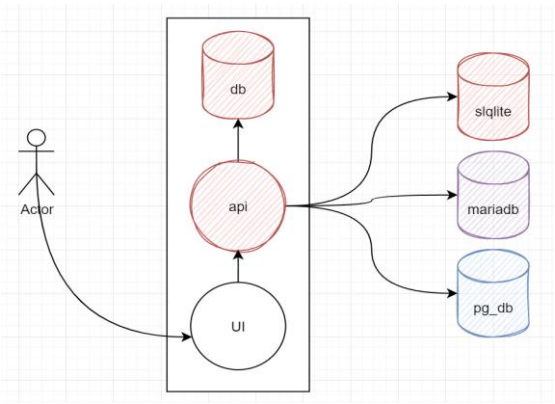


Fig 2. Technical Architecture before the deployment of ADBMS

The UIs, commands & feature inconsistencies meant that 3-4 separate bits of software were necessary to do simple CRUD tasks. The ADBMS

fixes this. Through it's unified UI & Application programming interface(API) it allows the user to not only access just a single database but also access several other relational databases which makes the job of the user easier.

Fig 3.3. Technical Architecture after the deployment of ADBMS



After the successful deployment of the ADBMS the users will be able to access all the different databases from a singular UI. These different frontend features are also explained in the future sections.

4.1 Implementation

The technical implementation consists of a few components

- The UI

  This is the user interface which will be used to explore and edit the different databases. The UI also has a variety of features including the support to create UML diagrams, show outputs of queries and allowing the user to run actual database queries on the different databases. The UI has been built with Javascript, BootStrap and HTML

- The API

  API is the acronym for Application Programming Interface, the brain of the operation. API is an interface to communicate with applications or computers. This is the component which does a lot of the heavy lifting and where the majority of the work has been done.

  The API is responsible for

  - allowing the user to browse through the different databases
  - listing of indices
  - listing of tables
  - conversion of ER diagrams
  - display of results

  The API has been built with Flask. Flask is a simple web framework for Python. The app is deployed with Docker & Docker compose which allows the different databases to be spun up during the development and testing of the entire project.

- The Database

  The database is the datasource where all of the information regarding how to connect to the slave databases is stored. This is also different from the database which the ADBMS system is actually accessing. This database also allows user authentication to be added to the system so that the access to the different features and the data can be

limited. The PostgreSQL, MySQL, SQLite database are available. Native support will be used for backup restore solutions like pgdump, MySQL Workbench and file based backup for sqlite.

- Query Engine
  Query engine is the component which generates and executes the queries. And then, the query is sent to the desired database. Done with the help of SQLAlchemy.

- ER diagram
  ER Diagram stands for Entity Relationship Diagram, is a diagram that displays the relationship of entity sets stored in a database. Analysis will be done based on the attributes and relationships with the help of foreign keys. The graph can then be drawn with the help of Python plotting libraries.

4.2 Frontend Features

Here are some of the features of the frontend for the ADBMS

- Create, view and edit on all objects. ADBMS has a syntax-highlighting sql editor and a live SQL Query Tool.
- Supportive error messages and helpful hints are present. Responsive, context-sensitive behavior means that the process of editing the data using the ADBMS is easy. Online help and information about using tools, tons of docs would be available online on how to use the tool.
- For the UI - a Browser tree control in the left pane and a tabbed browser in the right pane. Easy to use UI means that the user can focus on the task at hand rather than wasting time trying to learn how the system works. There are multiple data views
- It would be possible to generate ER diagrams. Perform queries online without knowledge of SQL - Aggregations, easy sorting automatic joins, editing data within tables

4.3 Backend Features

Not just the frontend features there are several backend features as well, which will be covered in this section. It allows us to connect multiple databases. It would also be able to backup and restore the data. SQL execution is possible. Listing schema and other db details are supported by the backend. The results could be exported as images and PDFs. Logs and history of all the calls is maintained. API calls & Server activity can also be monitored.

5. Results and Discussion

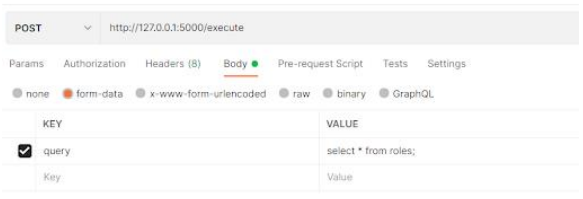


Fig 5.1 API Execution

This is a system where multiple database systems can be handled at the same place. This is achieved with the help of Python and its ORM libraries like SQLAlchemy[12] and the JDBC driver. The application is tested with Postman[13]. Postman offers numerous endpoint interaction methods : GET to obtain information, POST to add information, PUT to Replace information and DELETE to delete information.

Fig 5.2 Rows printed after connecting to the database

This is the terminal output where one receives an API request and can see the list of indices printed. Indices is a structure that holds the key values for a table or view. It is used to quickly and efficiently retrieve rows from a database. Below is the output for a select query.

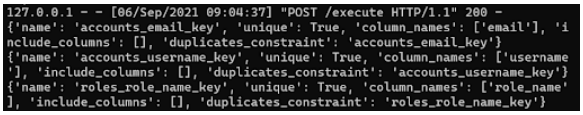


Fig 5.3 Indices listed

The above image has indices printed to the terminal after a Post request. It is a Python dictionary with several attributes describing the indices.

Fig 5.4 Tables listed

Above figure 5.4 has tables printed to the terminal after a Get request. It has a list with 2 strings indicating the tables in the database.

Fig 5.5 Connecting to the database with url

The UI is built with the help of flask which allows it to run in a web browser. The UI is divided into several segments- navbar, side navbar and sql editor. The above figure has an input field for db_url and a connect button.

“postgresql://YourUserName:YourPassword@YourHostname:5432/YourDatabaseName" is the specific format that db_url accepts.

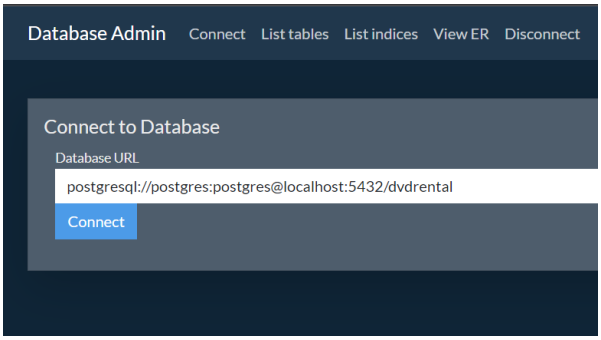


Fig 5.6 Connect to database in browser

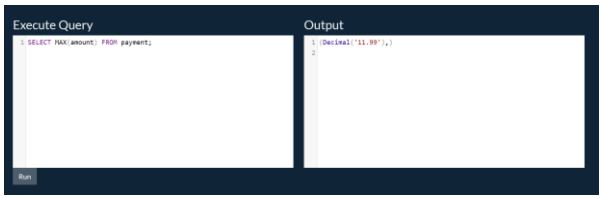


Fig 5.7 Execute Query

As seen in the fig. 5.7, UI includes a sql editor which supports sql syntax highlighting and indentation features. The sql editor is built using a JavaScript component called CodeMirror[14].

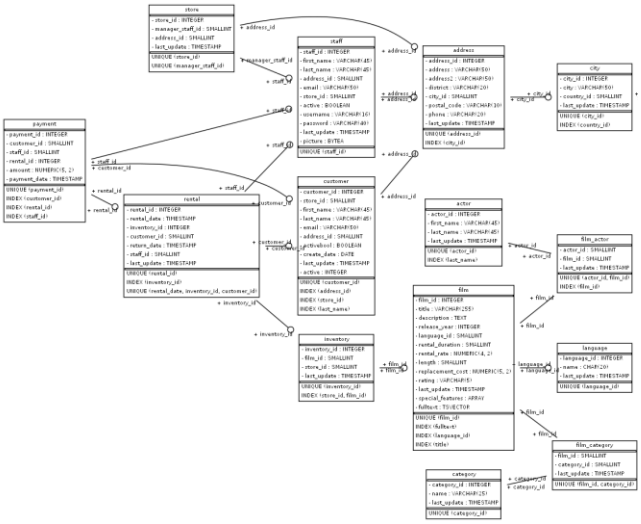


Fig 5.8 Generated ER diagram

We can also see in fig. 5.8, the generated ER diagram image for the selected database.

5. Conclusion

The field of information technology has been growing at a fast pace. There has been an increasing number of new types of requirements and techniques being developed in the area of database processing. The advantages of DBMS include reducing data duplication, maintaining data integrity, and improving the efficiency of data usage. The main aim of this project is to simplify database management, and with this system. This helps in understanding database systems and makes the process easier. The purpose of this project is that it should be able to help use databases more easily and efficiently. It should help in visualizing databases. The current user interface allows us to connect to the database and run queries.

Acknowledgement

This work is supported by the Department of Information Technology, SIES Graduate School of Technology. This Project is also supported by the Head of the Department Dr. K. Lakshmisudha and Project Guide Prof. Bushra Shaikh.

References

1. B.KiranKumar S.Durga Prasad, P M Manohar, KVVS SatyaPrakash, M.Chiranjeevi and K.Venkat Kiran, "Database Management System and Information Retrieval," International Journal of Computer Science and Information Technologies (IJCSIT).
2. Fankar Armash Aslam,Hawa Nabeel Mohammed Jummal Musab Mohd. Munir Murade Aaraf Gulamgaus, "Efficient Way Of Web Development Using Python And Flask", 2015 International Journal of Advanced Research in Computer Science.
3. Azhar Susanto, Meiryani , "Database Management System" 2019 International Journal of Scientific & Technology Research.
4. Antonio Badia, "Entity-Relationship Modeling Revisited", March 2004 Article in ACM SIGMOD Record.
5. Abhijit Banubakode,Haridasa Acharya, "Query Optimization in Object-Oriented Database Management Systems: A short review ", International Journal of Computer Science & Engineering Technology (IJCSET).
6. Ronald Fagin, Benny Kimelfeld, Yunyao Li, Sriram Raghavan, Shivakumar Vaithyanathanit, "Understanding Queries in a Search Database System", 2010 Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems,, Indianapolis, Indiana, USA.
7. Saurabh Gupta, Gopal Singh Tandel, Umashankar Pandey, "A survey on query processing and optimization in relational database management system", International Journal of Latest Trends in Engineering and Technology (IJLTET).
8. Cornelia A. Gyorödi , Diana V. Dumse-Burescu , Robert S. Gyorödi, Doina R. Zmaranda, Livia Bandici and Daniela E. Popescu , "Performance Impact of Optimization Methods on MySQL Document-Based and Relational Databases", Applied Science 2021.
9. SQLAlchemy : (https://www.sqlalchemy.org/library.html)
10. Postman : (https://learning.postman.com/docs/getting-started/introduction)
11. CodeMirror : (https://codemirror.net)