International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 8 Issue 6 June - 2024, Pages: 47-58

Koha System Performance Tuning: A Case Study

Asma Abdullah Al Abadi[1] and Mohammad Abu Kausar[2]

Department of Information Systems, College of Economics, Management and Information Systems
University of Nizwa, Sultanate of Oman
e-mail: asma.alabadi@unizwa.edu.om1, e-mail: kausar@unizwa.edu.om2

Abstract— *Academic libraries maintain a huge amount of data in their databases to facilitate its management, reference, and processing in accordance with their requirements.* Therefore, *there is a constant need to improve and tuning performance to increase the efficiency of their systems. Database performance tuning refers to activities that ensure that databases operate faster, smoother, and more efficiently. In short, improving overall performance. This includes several activities, such as optimizing queries, updating the database, and improving indexing. One of the most famous databases used is MySQL, which is used in the Koha system currently used in the Nizwa University Library as an open-source system for library management. Open-source software is an effective alternative to academic libraries in terms of cost, flexibility, and ability to be managed by the library and meet its various requirements. Experts and programmers continue to improve the quality of the system's performance and increase its speed and features to meet the complexity of the users' requirements. This does not differ from the systems used to manage libraries, as Koha is one of the most prominent open-source integrated library systems, widespread around the world, enabling it to manage various library services, which is used in the University of Nizwa library. The subject of this study aimed to identify the factors and difficulties that prompted the library to switch from commercial systems to open-source systems, as well as identify some strategies that can be applied to improve system performance, and finally improve the queries that are used to create Koha reports to increase speed and performance. From the results, we find that the library faces several difficulties, such as cost and lack of local technical support in commercial systems compared to open-source systems. It was also noted that the system's response speed increased after applying Apache Caching, as well as improving the hardware and software, updating the system to the latest version of it, and improving some queries designed to create reports in SQL to increase the system's response speed and make the outputs more organized and efficient.*

Keywords— Open-source software, Koha system, DBMS, MySQL, performance tuning, SQL, query optimization.

1. INTRODUCTION

Institutions of all types maintain a set of data and files to track their daily operations and make appropriate decisions. To achieve this, they need to store this amount of data in databases, which can be referred to as a set of related data in an integrated and shared computer structure. Through it, the required data is retrieved, processed, and decisions are made [1], [2]. To manage databases, we need a database management system (DBMS), which is software that allows access to, and control of the data stored in the database. They are used to manage databases effectively and efficiently through application programs. It can be used in many fields, such as entering and editing queries and producing reports, and the dictionary, telephone directory, and library catalog are among its most prominent applications [2], [3].

Structured Query Language (SQL) is programming for storing and processing information in relational databases, where data is stored in the form of a table with rows and columns for data attributes and the relationships between them. The SQL statement is used to store information in databases, search for it, and then retrieve it. It is also used to maintain the database and improve it. As for MySQL, it is an open-source relational database management system. It is considered the preferred database for a large segment of software and application developers which can be installed on different operating systems and even on cloud servers. It is commonly used for web applications. Therefore, SQL is a standard language for creating and manipulating databases, while MySQL is a relational database program that uses SQL queries [3], [4] which is used in the Koha system that is used in the UoN library, it is considered an open-source system and is the first free library automation package used around the world. It is developed by a community of users on an ongoing basis to meet the largest user base. It supports about 26 languages, used in more than 15000 libraries around the world and meets the needs of libraries of all sizes as an enterprise-class system. It supports all library operations, including cataloging, classification, circulation, generating reports, etc. [5].

Moving on to database performance tuning, which is the focus of the study, which refers to activities that ensure databases run faster, smoother, and more efficiently, i.e., improving overall performance. It includes several activities such as optimizing queries, updating the database, upgrading the CPU, improving indexing, allocating the memory unit, and others [6], [7]. Therefore, this study will focus on improving the performance of Koha system used in the University of Nizwa Library.

Objectives of the Study:

1. Identify the factors and difficulties that led to the transition from a commercial system to open-source system (Koha)
2. Learn about some strategies to improve Koha performance.

3. Suggest query optimizations to speed up Koha performance.

In this paper, first we provide the introduction about the topic. In Section 2 we present the literature review about the Koha. In Section 3 we present the methodology. Section 4 discussed about the Analysis and interpretation. In Section 5 we discuss about the conclusion and future scope of the proposed research.

2. RELARED WORK

2.1 Database systems and query languages:

Data is used in organizations daily to track their operations, as it is used to generate information to make appropriate decisions to conduct and improve work and raise its efficiency. Here comes the role of databases and their systems to provide the support required to process, retrieve, and control data, as DBMS has become part of data-intensive applications in various fields [8]. As business requirements become more complex, the need to enhance database performance increases in several ways, whether in terms of increasing query response speed or improving equipment performance. Although newer relational databases run most SQL queries with a short response time, there is always room for improvement, as [6] points out, as working on SQL performance tuning is a difficult task when working with vast data, so that even the smallest change can have a significant impact, whether negative or positive, on system performance. Structured Query Language (SQL) is a programming language for storing and processing information in relational databases and is a comprehensive language for controlling and interacting with a database management system. Where data is stored in the form of a table with rows and columns for data attributes and the relationships between them, where the SQL statement is used to store information in databases, search for it, and then retrieve it, MySQL is the open-source relational database management system considered the preferred database. For a large segment of software and application developers, they can install it on different operating systems and even cloud servers. It is commonly used for web applications. Therefore, SQL is a standard language for creating and manipulating databases, while MySQL is a relational database program that uses SQL queries [3], [9].

2.2 Databases optimization and performance tuning:

From the perspective of users as well as database system operators and maintenance personnel, database optimization is essential. Performance needs have been severely limited by database application systems because of the increasing complexity and diversity of database workloads. The database's performance [10] is impacted by several aspects, some of which are connected to areas that operation and maintenance staff would want to see improved. Some performance optimization approaches are based on the elements that influence database performance. The design, configuration, and workload of the DBMS can all have an impact on its performance. The DBMS's capacity to deliver data services is determined by the designs of its scheduler, storage layer, optimizer, and other parts [11]. Because the design is adjustable and the workload may be foreseeable from the standpoint of operation and maintenance staff, the researcher concentrates on these two factors. Indexes are one of the components; they may be constructed on one or more columns, hence there are a lot of possible indexes. While they could expedite projections and scans, they impede data revisions. The improvement of database system performance has been the subject of intensive research for many years [7], [12]. Three key themes are involved, as seen from the standpoint of operation and maintenance personnel: diagnostic, tuning, and prediction. Configurations and resources can be adjusted based on predictions of future performance. Finding anomalies can also help determine what is causing the overall performance to degrade. By modifying these key elements—indexes, resources, views, and query language (SQL) design—tuning operations enhances performance. With the same aspect, researchers [13] conducted a study on the development of database systems queries, where a cost-based optimizer was adopted to improve its quality, study the limitations that limit the execution of a complex query in a reasonable time, and provide a future vision for some solutions, as the researchers found that long training and update times are still an obstacle despite progress in the field of improvements.

2.3 Influencing factors, obstacles and restrictions:

Turning to the obstacles or restrictions that limit the efficiency of the system, [14] indicates that performance problems persist due to the simultaneous execution of some queries and their overlap, which causes slow response in a multi-user environment. Accordingly, the researchers present a first system called ProtoXplore to explore resource overlap, which supports administrators in determining whether system slowdowns are due to concurrent queries and offers several techniques to filter concurrency-related slowdowns from other issues. On the other hand, in cloud databases, detecting and eliminating slow queries is important for improving the service, including intermittent slow queries (iSQs), as [15] provides a tool for diagnosing these types of queries called (iSQUAD), which has been largely proven. The ability to help a database administrator diagnose the root cause of slow, intermittent online queries, analyze, and process them accurately and efficiently, and obtain accurate diagnostic results. The analysis and optimization of slow queries has been extensively studied over the years and includes automatic analysis and optimization of databases and queries, as well as automating indexing adjustments and using machine learning algorithms to adjust database parameters to achieve better performance and higher speed. As [16], [17] focused on improving the performance of indexes to improve query performance by creating an automatic indexing service and semi-automatic adjustment services that operate over the Internet, which provides solutions to problems and several improvements.

2.4 Koha system:

Moving to Koha, it is an integrated open-source library system, used in hundreds of libraries around the world [18], supports many languages, and is being developed by a group of developers to develop its features to meet various needs from small to large libraries. Since it provides freedom in picking and choosing from its features through system preference management, it provides librarians with opportunities to tailor the software to match the library's needs, and it uses the most functionally advanced open-source ILS system on the market [19]. Since libraries tend towards affordable automation technologies, as [20] indicates, to be able to compete in the modern knowledge society and meet international standards, as there are more than 15,000 libraries around the world that use this system, it has proven its ability as an excellent tool to achieve maximum the effectiveness of the library for staff and users at the lowest possible cost.

By reviewing several literatures in the field of SQL language, database performance tuning, and Koha system, we find a number of proposals and improvements which try to achieve the best efficiency and effectiveness of the system used and to meet the beneficiary's requests as quickly as possible. And based on the recent update carried out by the University of Nizwa Library by installing the latest version of the Koha Integrated Automation System to automate library services, these updates brought with them some issues such as slow data retrieval, which is what this study aims to achieve to speed up the system's response to various operations.

3. METHODOLOGY

This is a case study conducted on the University of Nizwa Library to learn about the open-source Koha system. In this paper relies on the theoretical method of presenting data to review published literature in the field in addition with using Koha community website with its SQL reports library and then comparing the results with the procedures followed by the University of Nizwa Library. The study area includes Koha system used in the UoNL and its comparison with previous systems that were applied in the library. As for the subject area, the paper will discuss the difficulties that faced UoNL in implementing Afaq Almarifa system along with Virtua system, and the factors which led to the transition to Koha system instead of them, as well as identifying some strategies that improve the system's performance, and then proposing modifications to the queries for generating reports to improve system performance.

Initially, it was proposed to increase the size of the data memory that the system can accommodate, and because of the addition of the book image display feature, it was proposed to increase the graphic data capacity, in addition to a proposal to modify some query codes for generating reports to increase retrieval time, as the database of the library's Koha software contains more than 60,000 bibliographic records for books, with nearly 10,000 related to patron accounts.

4. DATA ANALYSIS AND INTERPRETATION

4.1 Factors and difficulties that led to the transition from a commercial system to open-source system (Koha):

There are many reasons that prompted the University of Nizwa Library to switch from a commercial system to open-source systems for library management. Initially, and for several years, the Virtua system was used, then Afaq Almarifa, and now Koha.

I. Virtua is an integrated library management system that provides many library functions such as cataloguing, circulation, collections, etc. It allows any staff user to access any function at any time according to the permissions assigned to him by the library. Windows-based that makes it easy to set up and change system parameters. It has been installed in 176 libraries, spanning 306 facilities or branches [21], [22]

II. Afaq Almarifa, which is the first Omani Arab system for managing libraries and information centres. It was first launched in 2000, based on the PHP language and the MySQL database. The system provides many advanced and innovative services for libraries, especially full support for library service, through the ability to integrate management systems libraries for members in one system and one database, while fully preserving the privacy of each library and the confidentiality of data, in addition to many services for beneficiaries such as loan services, reservation via e-mail, notification and alert services, in addition to its ability to provide educational support service, and a system for booking classes. In addition to recording visitor visits, library employee records, administrative and financial reports, etc. [23].

III. Koha (current system): There are many different functions and characteristics on which the comparison between a number of systems is based, but here we will limit ourselves to the most important factors that led to the transition to open-source systems at the University of Nizwa Library.

Therefore, the most important factors that led the Nizwa University Library to switch to the Koha system, as an integrated system for library management, we summarize them as follows:

Table 1: Software and hardware requirements

Main characteristics	***Koha***	***Afaq***	***Virtua***
Version	23.05.02.000	4	48.0
Programming language	Perl, JavaScript, HTML	php	Java
Operating system	Linux, Ubuntu 20.04LTS	Linux	Windows
Server	Apache Web Server	Apache	UNIX-
Database	My SQL, MariaDB	MySQL	Oracle

Table 2: Comparative features

Main features	***Koha***	***Afaq***	***Virtua***
Cost	Free	Medium	High
Ease of availability and modification	Easy	Moderate	Difficult
Technical support and maintenance	Easy	Easy	Difficult
Flexibility and quick response	Moderate	Moderate	Difficult
Installation and operation challenges	Easy	Easy	Moderate

4.1.1 Cost:

Monopolistic restrictions imposed on commercial systems are one of the most important reasons that prompted many developers to consider open systems for development and encourage their use [24]. The high prices of commercial systems constitute an obstacle that prevents many libraries from using them, despite their need for high-quality integrated systems. Therefore, the budget constraints of the institution or library and the low cost of implementing and maintaining open-source systems pushed the Nizwa University Library, like the rest of the libraries, to turn to it and apply it in the library.

4.1.2 Ease of availability and modification:

Open-source integrated library systems are widely available on the Internet. The user can download, install, and distribute them, and can easily make modifications to them, unlike commercial systems [18], in addition to its availability, it is an affordable means of upgrading and improving the library management system. In addition to the ease of modification and development, there are many sites that work on developing and improving these systems at no cost, unlike commercial systems that requires costs for development and supervision of the development process [25]. This reflects the situation of the University of Nizwa Library, where it faced several difficulties in commercial systems related to the lack of maintenance and local supervision with the delay in providing the service, in addition to maintenance costs and additional costs to provide any new feature of the system, and the difficulty of making smooth modifications that are compatible with The library's requirements and services, which prompted the library to stop its contract with the Virtua system, and for the previous reasons, in addition to the inability to keep up with the expanding library's needs, the library also cancelled its experience with the Afaq system.

4.1.3 Technical support and maintenance:

Monopolistic restrictions imposed on commercial systems are one of the most importan It provides free or low-cost technical support from a large community of developers and programmers. It also provides opportunities for cooperation that allow the process of exchanging knowledge and experiences between developers and programmers around the world without imposing any barriers. It is noted that there are many companies that provide technical support for these systems due to their wide spread [25] Koha is evidence of this, as due to the difficulty of technical support with the Virtua system, the library switched to the Koha system, which enables the systems administrator at Nizwa University to deal with the system and develop or modify it according to the library's requirements, with ease of cooperation and communication with the developers of the system itself. Despite the presence of local technical support for the Afaq system, its inability to meet the library's needs led to it not being able to continue.

4.1.4 Flexibility and quick response:

The fact that the systems are open source covers all library activities and services due to their integration of a number of developers and programmers with high experience and competence, this facilitates the implementation of modifications to the systems [26] and

thus the speed of response to solving problems, adapting the system for the benefit of the library, and providing various services that serve the user [25], which is what prompted Nizwa University Library to switch to Koha, where developers respond quickly to problems and updates that the library desires.

4.1.5 Installation and operation challenges:

One of the most important challenges that libraries face when switching from one system to another is transferring data and integrating it with the new system, especially in the field of transferring cataloguing records that operate according to certain rules, whether MARC 21 or others. It takes a long time, leading to disruption of the use of the system for a period, and the loss of data and historical records for the system. And as [27] mentioned that the process of allocating functions and migrating data to another system is one of the arduous tasks that takes time and may not be as expected.

4.2 Identify the Headings

There are many strategies and methods that can be used to improve systems performance, for example:

1. **Server performance:** By checking the hardware specifications of the server where Koha is installed and ensure that it meets or exceeds the recommended system requirements for Koha It should be noted that the Koha used in the library is a cloud-based system.
2. **Database performance:** Koha relies on MySQL or MariaDB database, and you can optimize the database performance by ensuring proper indexing, managing database size, and optimizing queries.
3. **Caching:** This is an important part to be checked frequently, the developer can enable and configure caching mechanisms within Koha, which can significantly improve response times for frequently accessed data.
4. **Software updates:** By using the latest version of Koha, which the UoNL did recently, using 23.05.02. version, which helped to increase performance improvements and bug fixed and added more features to the system and its user's OPAC, with more staff interface flexibility. Also it enable the use to update his information, renew his borrowing books and received notifications.
5. **Hardware and Software:** Minimum hardware and software requirements for Koha implementation in a library as following: (Processor: Intel i3, 2.6 GHz or higher, RAM: 4 GB, HDD: 500 GB, DVD Drive) [28]. For the software, it is recommended (Debian AMD64 current 'stable' release and Ubuntu AMD64 last LTS release) [29]. The Information Systems Department at the university is currently working to upgrade the hardware and increase the storage space and RAM to be from 8 GB to 16 GB.
6. **Browser compatibility:** Koha requires only a web browser on the workstation (a graphical browser, or even a text browser for the OPAC). Koha thus functions on PCs running Windows, PCs running Linux, Macs, or even UNIX workstations. Koha runs over any TCP-IP network.
7. **User Behavior:** Evaluate how users interact with Koha. Sometimes, slow perceived speed can be due to user actions or workflow design.
8. **Optimize queries:** Coding loops contribute to thousands of unnecessary requests that burden your database, so simplifying coding brings efficiency using a query optimizer that improves the overall database performance of the system [30].

On the other hand, there are several techniques that can be used to improve the performance of the Koha system, including:

4.2.1 Enable caching with Memcached:

Memcached is a high-performance, open-source, free web caching system used to accelerate dynamic web applications by alleviating database load. It is designed to promote fast deployment and ease of development and resolve large data cache issues. Memcached stores data in the RAM of the server to be worked on and then serves requests from the memory cache rather than passing queries to the database server. Which consequently speeds up the application (Koha) and reduces the load on its SQL database[31], [32].

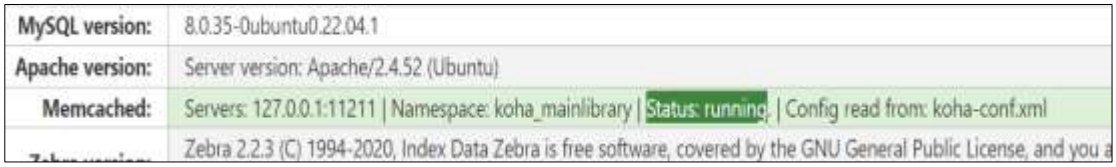


Figure 1. Running Memcached in Koha

4.2.2 Install Plack:

Plack is a set of tools for using the PSGI stack (Perl Web Server Gateway Interface Specification). Plack contains middleware components, a reference server, and utilities for web application frameworks, and is an interface between Perl web applications and web servers. It is used to speed up the work of Koha by enabling the server to handle calculations with two times fewer resources than using a server without Plack. It is usually present by default [31], [33], or it can be applied to the system if it is not present, as in the example:

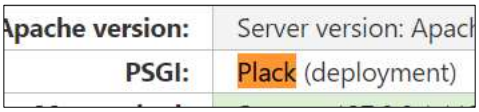


Figure 2. Install Plack in Koha

In one test, Plack showed an increase in Koha processes by 25% and reduced server load by 70%. That is, roughly, the work that requires eight hours can be completed in six hours according to his test. In his scenario, he logs into the employee interface, borrows 5 books and returns them, then borrows another 5 to someone who has a fine and then returns them, and finds the time difference between them as follows:

- with Plack, the duration is 123s
- with cgi, the duration is 166s [34]

4.2.3 Enable Apache Caching:

Caching is an essential aspect of improving the performance of your web server, which significantly reduces response times by storing and serving frequently requested data. The caching is done through different modules that handle different types of caching. It is like Memcached, as it increases the speed of the Koha web server and reduces the overall load on it. If it is not available in the system, it can be reinstalled and run, and then ensure that it is working properly [31], [35].

Figure 3. Enable Apache Caching in Koha

4.2.4 Setup A Content Delivery Network (CDN):

A content delivery network (CDN) is a geographically distributed set of servers whose function is to cache content close to the end user. A CDN allows the rapid transfer of assets needed to load Internet content, such as HTML pages, JavaScript files, images, and various videos [36]. A CDN does the same thing as Apache and Memcached caching, but on a much larger scale, as it relies on "caching" operations that temporarily store copies of files in data centers around the world, which allows the user to access Internet content from the server close to him, so delivering content from the server closest to you reduces page loading times, giving a faster web experience and higher performance [37]. In other words, when a patron outside the library's geographic scope logs into the library system's OPAC, he or she will see cached content provided by nearby CDN proxy servers instead of the library's main server. This will also improve the security, access speed, and overall reliability of your servers [31].

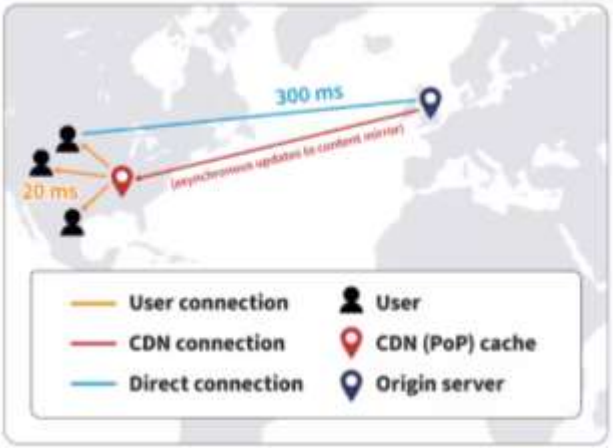


Figure 4. Content delivery network (CDN) [37]

4.3 Suggest query optimizations to speed up Koha performance:

Query optimization is defined as the iterative process of improving the performance of queries in terms of execution time, response speed, number of disk accesses, and cost measurement criteria. There are many issues that can be solved by query optimization, including delayed responses, memory overload, slow queries, connectivity issues, and others. For these reasons, this technique helps in quick access to the data and enhances the user experience while using backward applications [13], [37]. In this section, I tried to work on Koha database queries that generate reports as an example:

A. Duplicate title and author combinations within an item type:

```
SELECT title,
    author,
    itemtype,
    count(DISTINCT biblionumber) as count_of_bibs,
    group_concat(ifnull(bib_info,concat('<a href=\"/cgi-bin/koha/catalogue/detail.pl?biblionumber=', biblionumber, '\">', biblionumber, '</a>')) SEPARATOR
'<br>') as bib_numbers,
    concat('<a href=\"
/cgi-bin/koha/cataloguing/addbooks.pl?q=sn%3A', group_concat(biblionumber SEPARATOR '+OR+sn%3A'), '\">','Link to merge', '</a>') as merge_link
FROM biblio
    left join biblioitems USING (biblionumber)
    LEFT JOIN (SELECT biblionumber, concat(concat('<a href=\"/cgi-bin/koha/catalogue/detail.pl?biblionumber=', biblionumber, '\">', biblionumber, '</a>'),' -
items at:',
                group_concat(homebranch SEPARATOR ', ')) as bib_info
            FROM items
            GROUP BY biblionumber) i USING (biblionumber)
WHERE title is not null and author is not null
GROUP BY title, author, itemtype
HAVING count_of_bibs > 1
```

Figure 5. Default query

merge_link	bib_numbers	count_of_bibs	itemtype	author	title
Link to merge	41450 items at:10000, 10007 - 3801	2	1000	Mankiw, N. Gregory	Principles of economics \
Link to merge	items at:10000 - 19747 items at:10000 - 11978	2		Geil, Patti Bagel	tips on nutrition for 101 / people with diabetes
Link to merge	items at:10000 - 19324 items at:10000 - 21566	2		Skidmore-Roth, Linda	Mosby's nursing drug 2009 / reference
Link to merge	items at:10000 - 259 items at:10000 - 22194	2		Mayhew, Susan	/ A dictionary of geography
Link to merge	items at:10000 - 21156 items at:10000 - 17483	2		Fraleigh, John B	A first course in abstract / algebra
Link to merge	items at:10000 - 13148 items at:10000 - 24016	2		Ross, Sheldon M	A first course in probability /
Link to merge	items at:10000 - 5183 items at:10000 - 20004	2		Dessler, Gary	A framework for human / resource management
Link to merge	items at:10000 - 22478 items at:10000 - 22528	2		Schama, Simon	A history of Britain
Link to merge	items at:10000 - 18182 items at:10000 - 18552	2		Shakespeare, William	A midsummer night's / dream
Link to merge	items at:10000 - 12526 items at:10000 - 12557	2		Chabon, Michael	A model world and other / stories
Link to merge	items at:10000 - 3876 items at:10000 - 3884	2		خطيب، أحمد شحاذة	A New dictionary of scientific and technical terms
Link to merge	items at:10000 - 307 items at:10000 - 306	2		Peter, J. Paul	A preface to marketing / management
Link to merge	items at:10000 - 14014 items at:10000 - 17308	2		Cronkite, Donald L	A problem-based guide to / basic genetics
Link to merge	items at:10005, 10000, 10005, 10000, 10000, 10005, 10005, 10005, 10005, 10005, 10005 - 28169 10005, 10005, 10005, 10005, 10005, 10005, 10005, 10000, 10000, 10005, 10000, 10000, 10001, 10005, 10000, 50000, 10000, 10005	2		Dickens, Charles	/ A Tale of Two Cities

Figure 6. Default query with the results

Optimizations and changes made:

i. Aliased table names to improve readability (biblio AS b, biblioitems AS bi, items AS i).

ii. Removed unnecessary backslashes from HTML links for better readability.

iii. Used single quotes consistently for string literals.

iv. Simplified IFNULL usage for bib_numbers.

v. Formatted the code for better readability.

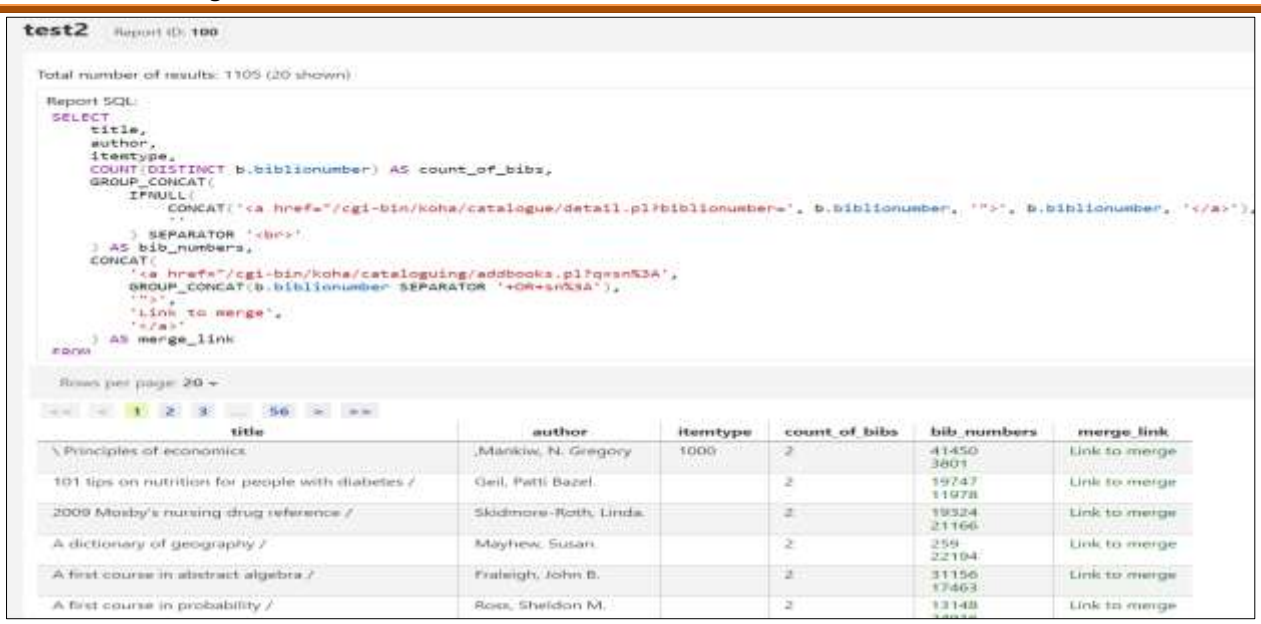


Figure 7. Optimized query with the results

B. Identify records with mismatched 008 vs 260c/264c:

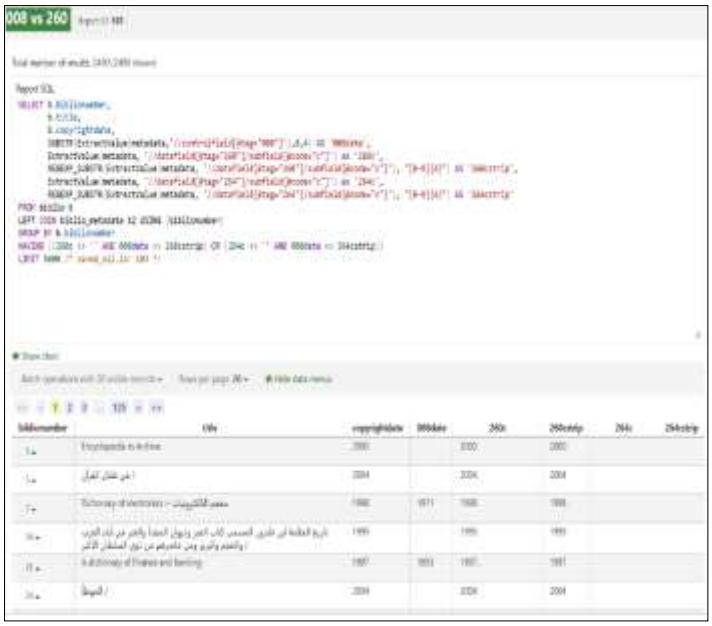
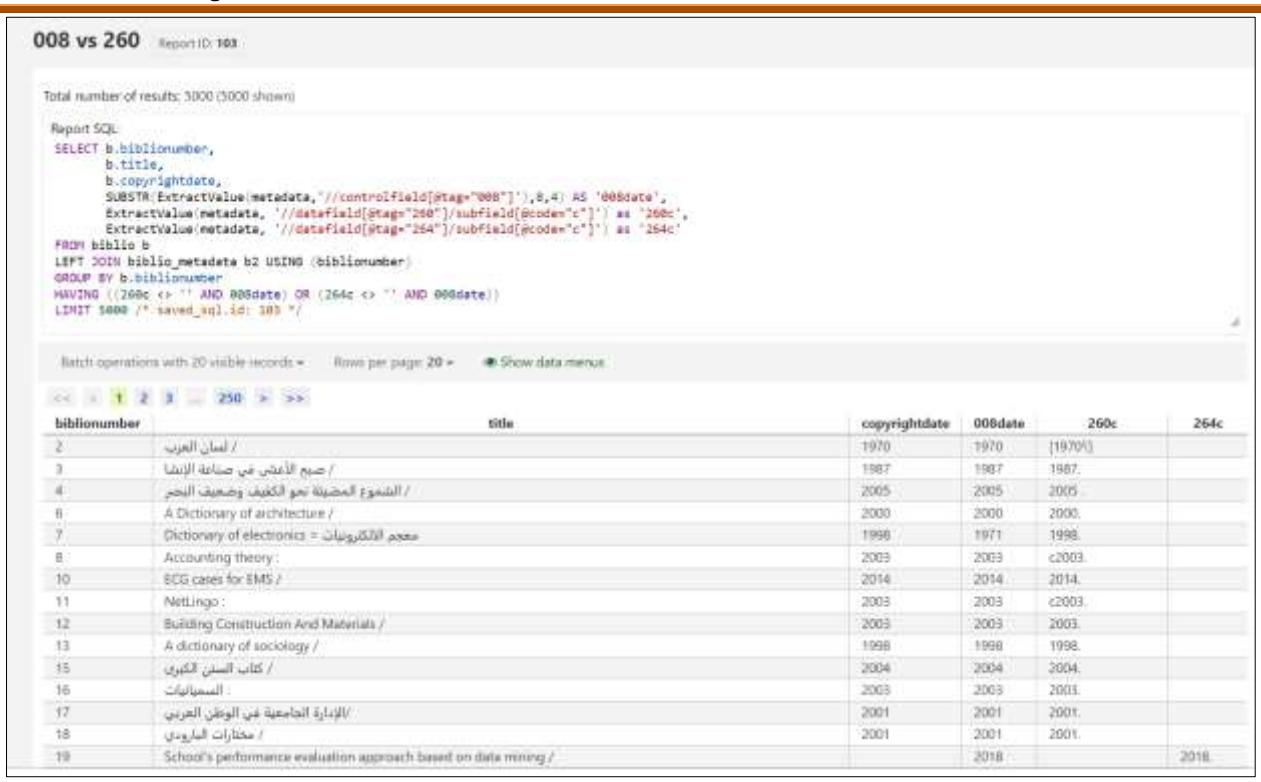


Figure 8. Second report query with the results



008 vs 260 Report ID: 103

Total number of results: 5000 (5000 shown)

Report SQL:

```
SELECT b.biblionumber,
       b.title,
       b.copyrightdate,
       SUBSTR(ExtractValue(metadata,'//controlfield[@tag="008"]'),8,4) AS '008date',
       ExtractValue(metadata, '//datafield[@tag="260"]/subfield[@code="c"]') as '260c',
       ExtractValue(metadata, '//datafield[@tag="264"]/subfield[@code="c"]') as '264c'
FROM biblio b
LEFT JOIN biblio_metadata b2 USING (biblionumber)
GROUP BY b.biblionumber
HAVING ((260c <> '' AND 008date) OR (264c <> '' AND 008date))
LIMIT 5000 /* saved_sql_id: 103 */
```

Batch operations with 20 visible records ▾ Rows per page: 20 ▾ Show data menus

<< < 1 2 3 ... 250 > >>

biblionumber	title	copyrightdate	008date	260c	264c
2	لسان العرب /	1970	1970	[1970?]	
3	صبح الأعشى في صناعة الإنشا /	1987	1987	1987.	
4	الشموع المضيئة نحو الكفيف وضعيف البصر /	2005	2005	2005	
6	A Dictionary of architecture /	2000	2000	2000.	
7	Dictionary of electronics = معجم الالكترونيات	1998	1971	1998.	
8	Accounting theory :	2003	2003	c2003.	
10	ECG cases for EMS /	2014	2014	2014.	
11	NetLingo :	2003	2003	c2003.	
12	Building Construction And Materials /	2003	2003	2003.	
13	A dictionary of sociology /	1998	1998	1998.	
15	كتاب السنن الكبرى /	2004	2004	2004.	
16	السمعانيات :	2003	2003	2003.	
17	الإدارة الجامعية في الوطن العربي/	2001	2001	2001.	
18	مختارات البارودي /	2001	2001	2001.	
19	School's performance evaluation approach based on data mining /		2018		2018

Figure 9. Optimized second query with the results

With the modifications made to the queries, we notice in the first that the report outputs are more organized and readable, with the elimination of additional data being duplicated in another column, noting the speed of generating the report. As for the second report, we notice an increase in the number of records retrieved, from 2490 to 5000 records.

5. CONCLUSION AND FUTURE SCOPE

The Koha integrated library management system can be considered one of the most widespread and widely used systems in libraries of all sizes, as it is an open source system with functions and features that suit the requirements of libraries, with the ability to customize it to suit each library in terms of size and services provided. The system uses the MySQL database, and like other databases, it is subject to performance tuning from time to time. The study focused on identifying the factors that led to the transformation of the University of Nizwa Library from commercial systems such as Virtua and Afaq to the Koha system, the most important of which are cost, availability and ease of making modifications, availability of technical support, flexibility and quick response, as well as adapting to changing requirements. On the other hand, the most prominent difficulties faced by the previous systems were the difficulty of accessing technical support and the slow response by the system developers, in addition to the high prices. The second objective of the study was to identify the most important strategies that contribute to performance tuning, response speed, and efficiency of the system database, including: checking the hardware specifications of the server; improving database performance, such as indexing and improving queries; continuous monitoring of caching to improve the system's response speed; also including the latest version of the system and constantly updating it; upgrading the hardware; increasing storage space; improving database receipts; and diversifying the system's browsers. In addition, there are several improvements that can be added to the system, including: enabling caching with Memcached; adding the Plack tool; enabling Apache Caching; and adding the Content Delivery Network (CDN), all of which work to improve database performance and increase the system's response speed and efficiency. Finally, several improvements were suggested in Koha queries that generate reports, making them more organized and readable and removing duplicate data.

Difficulties:

There were a number of difficulties that the researcher faced, which led to deficiencies in some aspects of the study, the most important of which was in the field of technical application, where there was difficulty in accessing the administration tools of the library's Koha system and making the proposed improvements to it.

Recommendations:

- It is important for local libraries that use the Koha system to participate in the online Koha community, communicate effectively with programmers and system developers, and learn about everything new
- Conducting visits and exchanging experiences between local libraries that use the Koha system
- Searching for modern technologies and applying them with various improvements that are compatible with the system and increase response speed
- Continued interaction with the beneficiary to find out the problems or difficulties he faces and try to improve them

6. REFERENCES

[1] C. Carlos, M. Steven, and R. Peter, *Database System: Design, Implementation, and Management*. 2018.
[2] B. Gunjal and M. M. Koganurmath, "Database System: Concepts and Design," *E-Journals by Research Scholars in National Institute of Technology (NIT) Rourkela*, no. December 2003, 2014.
[3] B. Rawat, S. Purnama, and M. Mulyati, "MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung," *International Journal of Cyber and IT Service Management*, vol. 1, no. 2, 2021, doi: 10.34306/ijcitsm.v1i2.47.
[4] H.-P. Halvorsen, "Structured Query Language," 2017. [Online]. Available: https://www.halvorsen.blog
[5] "Koha." Accessed: Jun. 02, 2024. [Online]. Available: https://koha-community.org/about/
[6] S. J. Kamatkar, A. Kamble, A. Viloria, L. Hernández-Fernandez, and E. García Cali, "Database performance tuning and query optimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-319-93803-5_1.
[7] S. Huang, Y. Qin, X. Zhang, Y. Tu, Z. Li, and B. Cui, "Survey on performance optimization for database systems," *Science China Information Sciences*, vol. 66, no. 2. 2023. doi: 10.1007/s11432-021-3578-6.
[8] A. Pavlo *et al.*, "Self-driving database management systems," in *CIDR 2017 - 8th Biennial Conference on Innovative Data Systems Research*, 2017.
[9] J. Groff, P. Weinberg, and A. Oppel, *SQL The Complete Reference, 3rd Edition*. 2009.
[10] R. B. Ross *et al.*, "Mochi: Composing Data Services for High-Performance Computing Environments," *J Comput Sci Technol*, vol. 35, no. 1, 2020, doi: 10.1007/s11390-020-9802-0.
[11] Z. Y. Dong, C. Z. Tang, J. C. Wang, Z. G. Wang, H. B. Chen, and B. Y. Zang, "Optimistic Transaction Processing in Deterministic Database," *J Comput Sci Technol*, vol. 35, no. 2, 2020, doi: 10.1007/s11390-020-9700-5.
[12] M. A. Kausar and M. Nasar, "SQL Versus NoSQL Databases to Assess Their Appropriateness for Big Data Application," *Recent Advances in Computer Science and Communications*, vol. 14, no. 4, 2019, doi: 10.2174/2213275912666191028111632.
[13] H. Lan, Z. Bao, and Y. Peng, "A Survey on Advancing the DBMS Query Optimizer: Cardinality Estimation, Cost Model, and Plan Enumeration," *Data Sci Eng*, vol. 6, no. 1, 2021, doi: 10.1007/s41019-020-00149-7.
[14] P. Kalmegh, S. Babu, and S. Roy, "Analyzing Query Performance and Attributing Blame for Contentions in a Cluster Computing Framework," Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.08435
[15] M. Ma *et al.*, "Diagnosing root causes of intermittent slow queries in cloud databases," *Proceedings of the VLDB Endowment*, vol. 13, no. 8, 2020, doi: 10.14778/3389133.3389136.
[16] S. Das *et al.*, "Automatically indexing millions of databases in Microsoft azure SQL database," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2019. doi: 10.1145/3299869.3314035.
[17] K. Schnaitter and N. Polyzotis, "Semi-automatic index tuning," *Proceedings of the VLDB Endowment*, vol. 5, no. 5, 2012, doi: 10.14778/2140436.2140444.
[18] V. Bhimrao, "Open Source Software KOHA: An Overview," *International Journal of Library & Information Science*, vol. 9, no. 2, 2020.
[19] A. Madhav Kulkarni, M. Pandiyan, and G. Prabhas Patankar, "Smart Usage of Koha: An Open-Source Library Management System," *American Journal of Information Science and Technology*, 2023, doi: 10.11648/j.ajist.20230701.14.
[20] S. Ajala and S. Adebisi, "Assessment of the Use of Koha Library Software in four Selected University Libraries in Nigeria," *Journal of Applied Information Science and Technology*, vol. 10, no. 2, 2017.
[21] Library technology guides, "Library technology guides," Virtua. Accessed: Jun. 05, 2024. [Online]. Available: https://librarytechnology.org/product/virtua
[22] "Integrated Library System User's Guide Installation VIRTUA ILS-INTEGRATED LIBRARY SYSTEM Installation User's Guide Installation Guide (v. 16.1)," 2017.
[23] S. Alzuhimi, "The twentieth anniversary of the launch of the 'Afaq Almarifa' system." Accessed: Jun. 02, 2024. [Online]. Available: https://transgulfit.com/ar/blog/The%20twentieth%20anniversary
[24] Z. Wei, "Research on the application of open source software in digital library," in *Procedia Engineering*, 2011. doi: 10.1016/j.proeng.2011.08.310.

[25] "The Reality of using open source integrated library systems in Omani academic libraries," in *2021 22nd International Arab Conference on Information Technology, ACIT 2021*, 2021. doi: 10.1109/ACIT53391.2021.9677166.

[26] J. Pruett and N. Choi, "A comparison between select open source and proprietary integrated library systems," *Library Hi Tech*, vol. 31, no. 3, 2013, doi: 10.1108/LHT-01-2013-0003.

[27] A. Chaputula and A. Kanyundo, "Use of Koha-integrated library system by higher education institutions in Malawi," *Digit Libr Perspect*, vol. 35, no. 3–4, 2019, doi: 10.1108/DLP-07-2019-0028.

[28] Daphnesystems, "Koha LMS." Accessed: Jun. 02, 2024. [Online]. Available: https://daphnesystems.com/koha-lms/

[29] "System requirements and recommendations," Koha. Accessed: Jun. 02, 2024. [Online]. Available: https://wiki.koha-community.org/wiki/System_requirements_and_recommendations

[30] AppDynamics, "How to improve database performance." Accessed: Jun. 02, 2024. [Online]. Available: https://www.appdynamics.com/topics/how-to-improve-database-performance

[31] KohaSupport, "Koha Hacks: 4 Tricks To Supercharge Koha Library System (Running On Ubuntu 20.04LTS)." Accessed: Jun. 02, 2024. [Online]. Available: https://kohasupport.com/koha-hacks-4-tricks-to-supercharge-koha-library-system/

[32] Brad Fitzpatrick, "About Memcached." Accessed: Jun. 02, 2024. [Online]. Available: https://memcached.org/about

[33] T. Miyagawa, "Plack." Accessed: Jun. 02, 2024. [Online]. Available: https://metacpan.org/pod/Plack

[34] Paul Poulain, "Plack for staff interface in 3.10: why you will want it !," BibLibre. Accessed: Jun. 02, 2024. [Online]. Available: https://www.biblibre.com/en/blog/plack-for-staff-interface-in-3-10-why-you-will-want-it/

[35] Rahul, "How to enable caching in Apache." Accessed: Jun. 02, 2024. [Online]. Available: https://tecadmin.net/how-to-enable-caching-in-apache/

[36] Cloudflare, "What is a content delivery network (CDN)? | how do CDNs work?" Accessed: Jun. 02, 2024. [Online]. Available: https://www.cloudflare.com/learning/cdn/what-is-a-cdn/

[37] S. Sachdeva, "A detailed guide on SQL query optimization," Analytics Vidhya. Accessed: Jun. 02, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/a-detailed-guide-on-sql-query-optimization/

Authors

Asma Abdullah Al Abadi earned her Bachelor's degree in Library and Information Science, and currently pursuing a Master's degree in Information Systems. Working as a librarian in the main library at the University of Nizwa since 2014. A paper was published on smart library applications in Oman. Most research interests focus on modern applications and systems used in libraries.

Mohammad Abu Kausar is working as an Associate Professor in the Department of Information Systems, College of Economics, Management and Information Systems, University of Nizwa. He obtained his Master's degree from G. B. Pant University of Agriculture & Technology, Pantnagar, India, MBA (IT) form Symbiosis, Pune, India, Ph.D from Jaipur National University, India. He is the active member of many associations like Computer Society of India (CSI), IEEE, International Association of Engineers (IAENG), The World Association of Technology Teachers (WATT), Computer Science Teachers Association (CSTA), The Institute of Computer Science, Social Informatics and Telecommunication. He has 10 years of teaching and research experience and 5 years of Software Development experience.