

RESEARCH TITLE

Developing a tool for Redo log Management in Oracle Database

Leena Awad Ahmed Elhusain¹

¹ Sudan University of Science and Technology

Email: leenaawad5@gmail.com

HNSJ, 2023, 4(10); <https://doi.org/10.53796/hnsj4102>

Published at 01/10/2023

Accepted at 07/09/2023

Abstract

The log retains the operations that occur in the database. In this paper, a tool was developed to manage database logs, allowing access to information and reviewing the actions of certain individuals. An analysis of the log was conducted using LogMiner, a tool that enables us to view the changes made to the database. A user interface has been designed to interact with the log.

Key Words: LogMiner ; Redo log ; Archived log.

تطوير أداة لإدارة السجل في قاعدة بيانات اوراق

لينة عوض أحمد الحسين¹

¹ جامعة السودان للعلوم والتكنولوجيا، السودان.
بريد الكتروني: leenaawad5@gmail.com

HNSJ, 2023, 4(10); <https://doi.org/10.53796/hnsj4102>

تاريخ القبول: 2023/09/07م

تاريخ النشر: 2023/10/01م

المستخلص

السجل يحتفظ بالعمليات التي تتم في قاعدة البيانات .
في هذه الورقة تم تطوير أداة لإدارة قواعد بيانات السجل بحيث يمكن الوصول الى المعلومات واستعراض تصرفات بعض الأفراد .
تم عمل تحليل للسجل باستخدام Log Miner وهو أداة نستطيع من خلالها الاطلاع على التغييرات التي تمت على قاعدة البيانات .
تم تصميم واجهة للمستخدم للتعامل مع السجل .

الكلمات المفتاحية: لوغمينر ; إعادة السجل ; السجل المؤرشف.

1. Introduction:

There are two oracle memory structures relating to recovery log buffers and data block buffers. the log buffers are the memory buffers that record the changes, or transactions, to data block buffers before they are written to online redo logs or disk .Online redo logs record all changes to the database, whether the transactions are committed or rolled back .The data block buffers are the memory buffers that store all the database information .A data block buffer stores mainly data that needs to be queried, read changed, or modified by users. The oracle file structures relating to recovery include the online redo logs. archived redo logs, control files, data files, and parameter files. The redo logs consist of files that record all the changes to the database [1].

2. Problem definition

The problem at hand pertains to the challenge of accessing the redo log file and archived log file, which collectively store a comprehensive record of all changes and transactions that have taken place within a database over time.

3. Methodology

In this section, the methodology employed for the development and implementation of the proposed tool is presented .

3.1 Design Objectives

- In this section, the key objectives that guide the development and implementation of the proposed tool are defined. The primary objective of this Tool is to enable real-time monitoring and review of modifications made to data. This ensures data integrity and security.
- To provide supplementary information that supports capacity planning and resource allocation, helping optimize system performance.
- The Tool will facilitate the retrieval of critical data required for debugging and troubleshooting complex applications, reducing downtime.
- The tool will incorporate data recovery mechanisms to restore information that has been unintentionally deleted, minimizing data loss risks.

3.2 Tool Description

In this section, an overview is provided of the fundamental tool used in this methodology . LogMiner is a robust Oracle utility utilized in this paper. It empowers users to conduct online queries on the contents of both Redo log files and archived log files. the functionality of LogMiner includes:

- **Online Querying:** LogMiner allows for real-time queries, enabling instant access to transactional data stored in Redo log files.
- **Archived Log Files:** It extends its capabilities to archived log files, ensuring historical data retrieval.
- **Audit and Analysis:** LogMiner serves as a powerful audit tool, facilitating detailed tracking of data modifications.
- **Data Recovery:** Additionally, LogMiner provides data recovery features, enhancing the system's resilience against data loss incidents.

4. Implementation and Results

This section outlines the transition from the theoretical framework and design phase to the practical implementation .

4.1 LogMiner Configuration

- Enable Supplemental Logging
- Set Initialization Parameters
- Prepare the LogMiner Dictionary
- Start LogMiner
- Specify Log Files and Time Range
- Load Log Files
- Query LogMiner Views

4.2 Results

The screenshot shows the 'DataBase Management System For Log' interface. On the left, there are three sections: 'Add Log File' with a dropdown for 'LOG Name' set to 'REDO01' and an 'Add Log File' button; 'Start Logminer Ranged' with 'Start Time' set to '8/1/2012' and 'End Time' set to '10/1/2012', and a 'Start Logminer' button; and 'Query Logminer Contents' with 'User file...' set to 'SCOTT' and a 'Fetch' button. On the right, a table displays the results of the query, with columns for ID, RED SQL, Operation, User file, and TIME(S). The table contains 20 rows of data, including operations like 'set transaction read write;', 'update', 'commit;', 'insert into', and 'create table'.

ID	RED SQL	Operation	User file	TIME(S)
04000C042010000	set transaction read write;	START	SCOTT	2012-01-08
04000C042010000	update 'SYS','OBJ\$' set 'OBJ#' = 1, 'DATAOBJ#'	UPDATE	SCOTT	2012-01-08
04000C042010000	commit;	COMMIT	SCOTT	2012-01-08
0400140042010000	set transaction read write;	START	SCOTT	2012-01-08
0400140042010000	commit;	COMMIT	SCOTT	2012-01-08
04001A0042010000	set transaction read write;	START	SCOTT	2012-01-08
04001A0042010000	commit;	COMMIT	SCOTT	2012-01-08
04000F0042010000	set transaction read write;	START	SCOTT	2012-01-08
04000F0042010000	insert into 'SYS','SEG\$' values 'TS#' = 4, 'FILE#'	INSERT	SCOTT	2012-01-08
04000F0042010000	update 'SYS','TSQ\$' set 'GRAITOR#'	UPDATE	SCOTT	2012-01-08
04000F0042010000	commit;	COMMIT	SCOTT	2012-01-08
0400160042010000	set transaction read write;	START	SCOTT	2012-01-08
0400160042010000	commit;	COMMIT	SCOTT	2012-01-08
0400080042010000	set transaction read write;	START	SCOTT	2012-01-08
0400080042010000	commit;	COMMIT	SCOTT	2012-01-08
0400080042010000	insert into 'SYS','OBJ\$' values 'OBJ#'	INSERT	SCOTT	2012-01-08
0400080042010000	insert into 'SYS','TAB\$' values 'OBJ#'	INSERT	SCOTT	2012-01-08
0400080042010000	insert into 'SYS','COL\$' values 'OBJ#'	INSERT	SCOTT	2012-01-08
0400080042010000	create table st(no number(3));	DCL	SCOTT	2012-01-08
0400080042010000	update 'SYS','SEG\$' set 'TYPE#'	UPDATE	SCOTT	2012-01-08
0400080042010000	commit;	COMMIT	SCOTT	2012-01-08
0500000050010000	set transaction read write;	START	SCOTT	2012-01-08
0500000050010000	insert into 'SCOTT','ST' values 'NO'	INSERT	SCOTT	2012-01-08
0500000050010000	commit;	COMMIT	SCOTT	2012-01-08

Figure 1 illustrates the activation of LogMiner for the time period from 8/1/2012 to 10/1/2012 specifically for the "SCOTT" user . The information is obtained in 13 seconds.

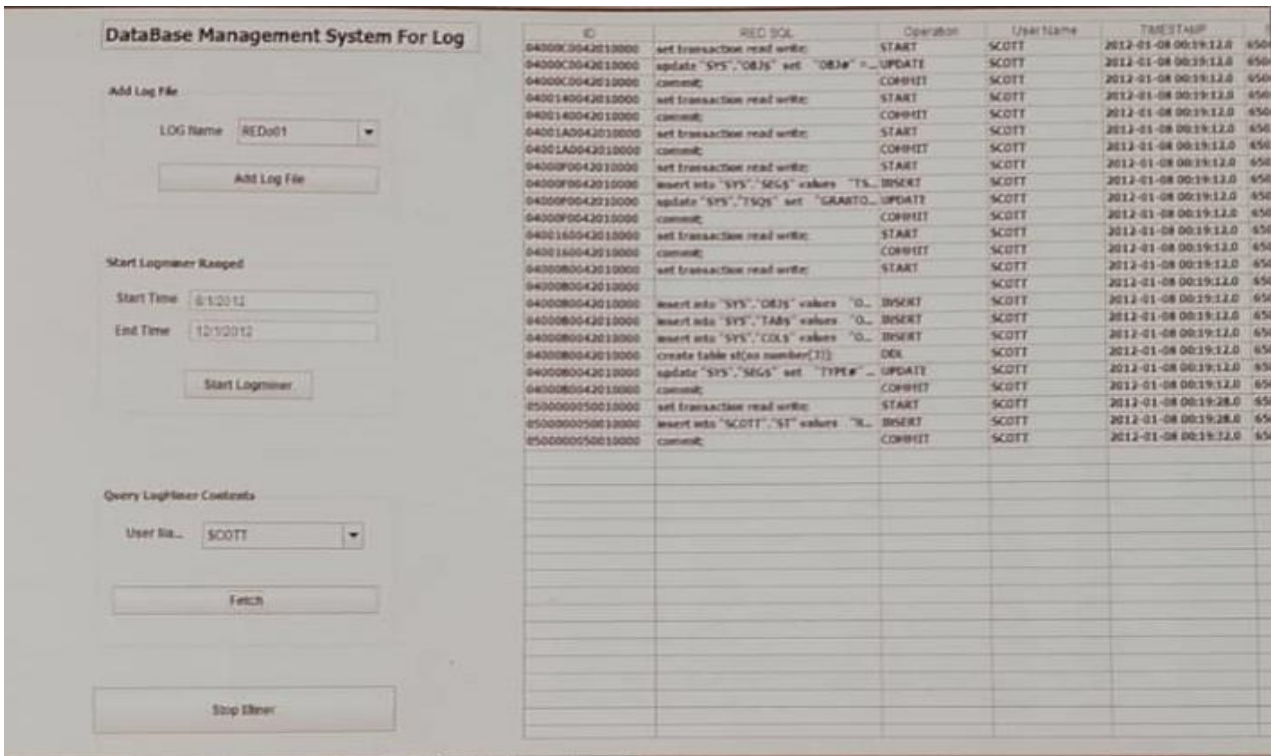


Figure 2 illustrates the activation of LogMiner for the time period from 8/1/ 2012, to 12/1/ 2012, specifically for the "Scott" user .

The information is obtained in 27 seconds.

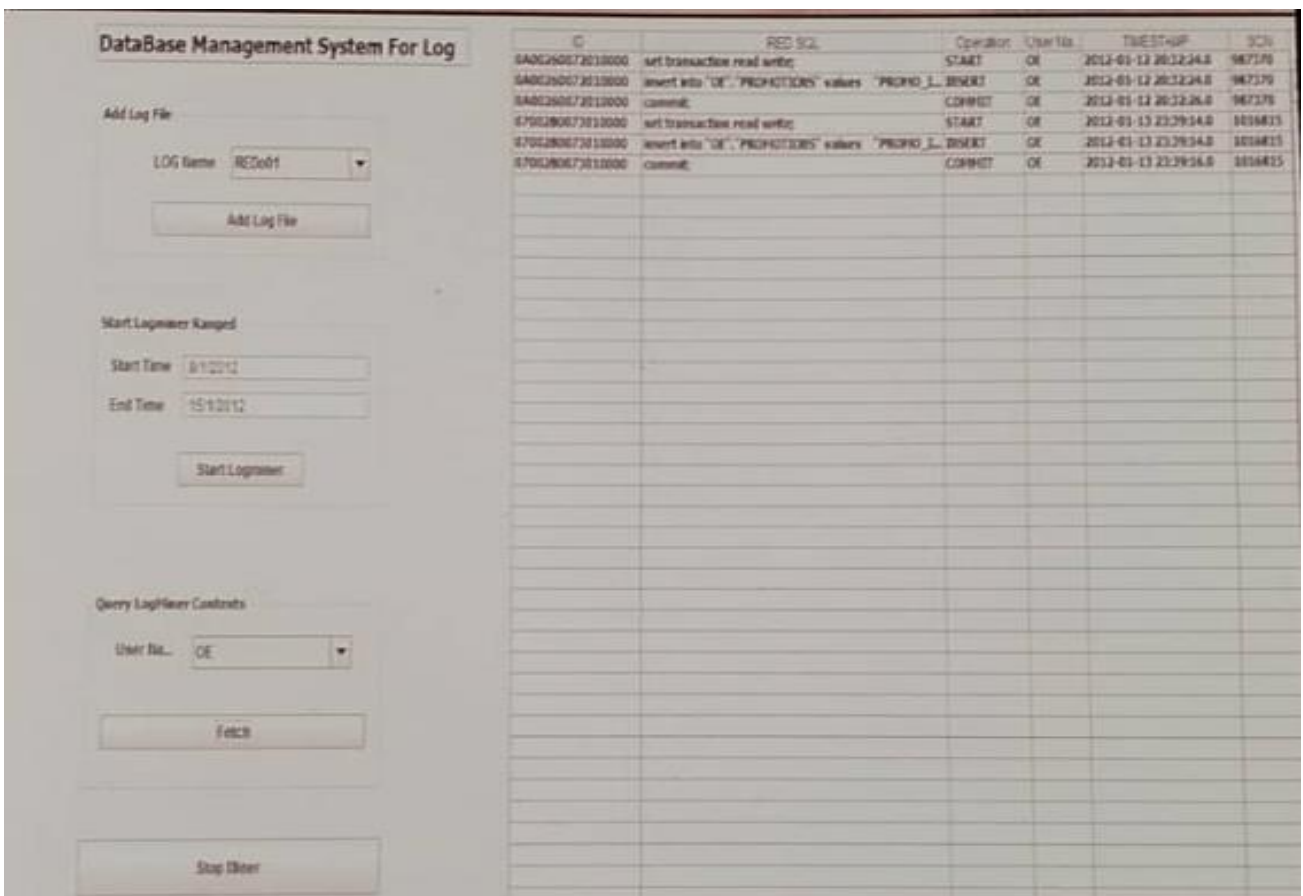


Figure 3 illustrates the activation of LogMiner for the time period from 8/1/2012 to 15/1/2012 specifically for the "OE" user . The information is obtained in 46 seconds.

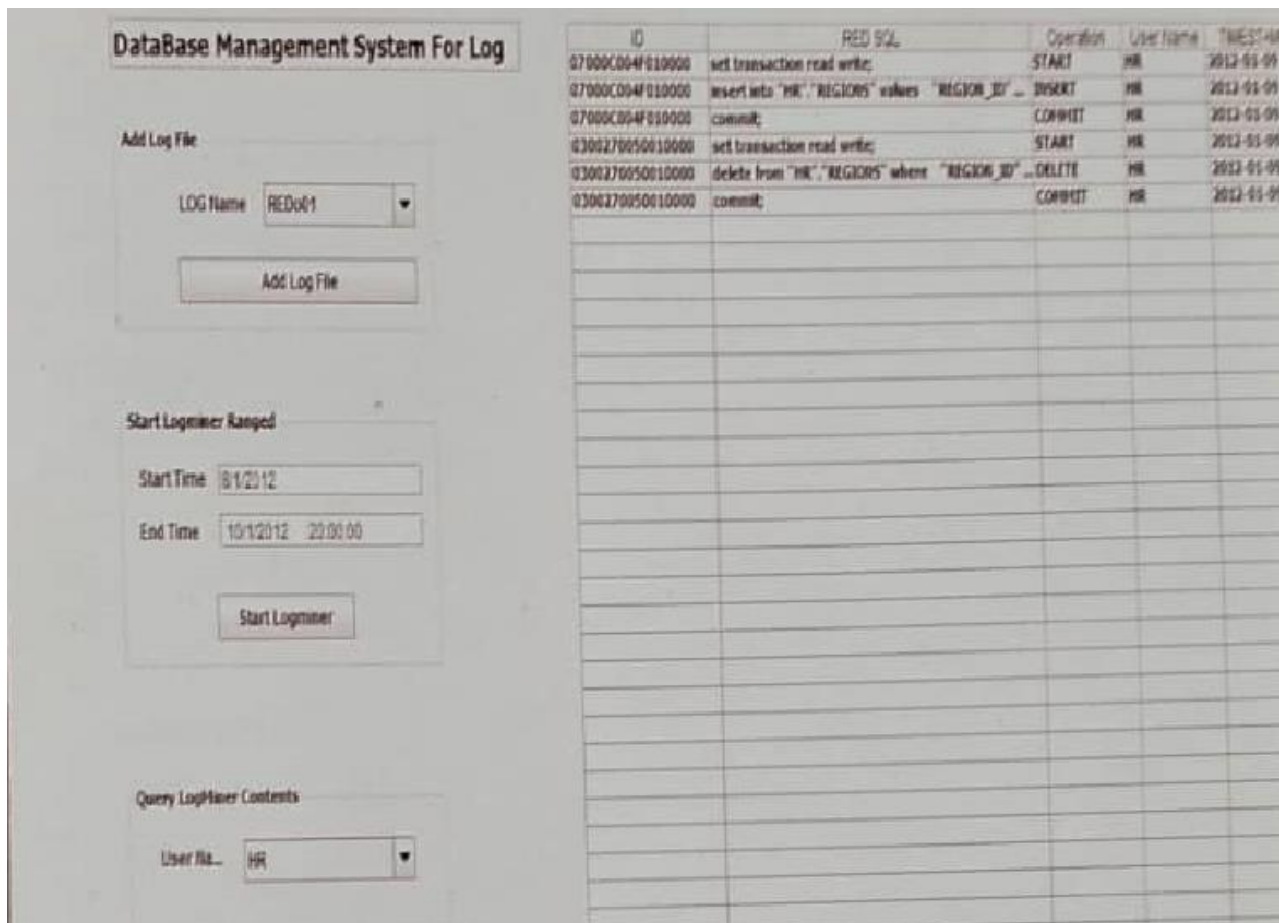


Figure 4 illustrates the activation of LogMiner for the time period from 8/1/2012 to 10/1/2012 specifically for the "HR" user . The information is obtained in 13 seconds.

5. Conclusion

In conclusion, the proposed tool offers a robust solution to address the critical needs of data management and system performance optimization. By enabling real-time monitoring and review of data modifications, it ensures data integrity and security in an ever-evolving digital landscape. Moreover, the supplementary information it provides supports efficient capacity planning and resource allocation, leading to enhanced system performance.

The tool's role in facilitating the retrieval of critical data for debugging and troubleshooting complex applications cannot be understated, as it significantly reduces downtime and accelerates issue resolution. Furthermore, with its integrated data recovery mechanisms, the tool acts as a safeguard against unintended data loss, minimizing risks and ensuring the continuity of data-driven operations.

In summary, the proposed tool stands as a valuable asset in the realm of data management and system optimization, offering a holistic solution that empowers organizations to maintain data integrity, optimize resources, streamline troubleshooting, and mitigate data loss risks effectively.

References

AULT, M., Liu, D., & Tumma, M. (2005) . Oracle database 10g new features.