Analysis on Comparing the Performance of Alpha Miner Algorithm on MYSQL and Oracle

Mekhala

Abstract— **Process mining is a field that analyses the event log to discover, enhance and improve business processes and check performance between run time and design time business processes. The event logs are stored in databases. Many databases are available for handling and saving huge data like Oracle and MYSQL. However there are certain applications for which some databases do not implement. MYSQL has emerged for handling such applications. For this alpha miner algorithm is the most widely used technique. Our objective is to compare the performance of MYSQL and Oracle under process mining. We implement the alpha miner algorithm on row oriented database in database query languages. We present a performance benchmarking and comparison of α miner algorithm on row oriented databases to store massive event logs and analyse it in seconds to discover a process model.**

Index Terms— **MYSQL, Process mining, Oracle, Row Oriented Database, α- miner algorithm**

I. RESEARCH AIM AND OBJECTIVE

Process Mining is new and emerging field which consist of analysing event logs generated from the execution of business process. The insights obtained from event logs help the organizations to improve their business processes. There are three major techniques within Process Mining viz. Process Discovery, Process Conformance and Process Enhancement. One of the most fundamental algorithms under Process Discovery is the alpha -miner algorithm [24] which is used to generate process model from event logs. Before the year 2000, majority of the organizations used traditional Relational Database Management System (RDBMS) to store the data. MYSQL database and Oracle are well suited for such analytical queries. The Oracle RDBMS stores data logically in the form of table-spaces and physically in the form of data files. Our aim is to examine an approach to implement a process discovery alpha miner algorithm on MYSQL and Oracle.[11,14]We aim to implement α-miner algorithm in Structured Query Language (SQL) so that our Process Discovery application is tightly coupled to the database.[13]
Main research aims presented in this paper are:

1. To implement α- miner algorithm in MYSQL and Oracle.
2. To compare mining algorithm performance on row-oriented database.

II. RELATED WORK

In this section, we review closely related work to the study presented in this paper and list the novel contributions of our work in context to existing work. We divide related work into following lines of research:

Manuscript received August 10, 2019

1. Implementation of α- miner algorithm on MYSQL database.
2. To implement α- miner algorithm on Oracle.
3. To compare performance of mining algorithm on row-oriented database.

2.1 IMPLEMENTATION OF α- MINER ALGORITHM ON ROW ORIENTED DATABASE

Ordonez et al. investigate an approach to efficiently implement the EM algorithm in SQL [2].
They perform clustering of large datasets. They effectively handle high dimensional data, a high number of clusters and more importantly, a very large number of data records. Xuequn Shanget al. presents an efficient implementation of frequent pattern mining in relational databases [19]. They propose a concept called Projection Pattern Discovery (Propad). Propad fundamentally differs from Apriori like candidate set generation-and-test approach. This approach successively projects the transaction table into frequent item sets to avoid making multiple passes over the large original transaction table and generating a huge set of candidates. [23]

We present a few segments of our implementation due to limited space in the paper. The entire code and implementation can be downloaded from our website6. Before implementing –miner algorithm, we do pre-processing in JAVA to create two tables viz. causality table (consist of two column event A and event B) and not connected table (consist of two column event A and event B).

1. We create a table event log using create table7 keyword consisting of 5 columns (Case ID, Timestamp, Status, Activity and Actor) each of which are varchar datatype except Timestamp which is of timestamp datatype. The primary key is a composite primary key consisting of Case ID, Timestamp and Status.
2. We load the data into table event log using LOAD DATA INFILE8 command.
3. For Step 1, we create a table total Event that contains a single column (event) which is of varchar datatype. To populate the table we select distinct activities from the table event log.
4. For Step 2, we create a table initial Event that contains a single column (initial) which is of varchar datatype. To populate the table

A. We first select the minimum value of Timestamp from table event log by grouping Case ID.
B. Then we select distinct activities from table event log for every distinct value of Case ID where Timestamp is the minimum Timestamp.

5. For Step 3, we create a table final Event that contains a single column (final) which is of Varchar datatype. To populate the table

A. We first select maximum Timestamp from a table event log by grouping Case ID.
B. Then we select distinct activities from a table event log for every distinct value of Case ID where Timestamp is the maximum Timestamp.

6. For Step 4, we create five tables viz. Safe Event A, Safe Event B, Event A, Event B and XL. All the five tables contain two columns (set A and set B) which are of varchar datatype.

A. In table causality we use group_concat9 to combine the values of column event B of corresponding value of a column event A and insert the results in a table Event A.
B. In table causality we use group concept to combine the values of column event A of corresponding value of a column event B and insert the results in the table Event B.
C. To populate tables Safe Event A and Safe Event B
   I. Select set A and set B from tables Event A and Event B
   II. For every value of set B in table Event A, if value is present in table not connected, insert the corresponding value of set A and set B in table Safe Event A. Repeat the same step for populating table Safe Event B.
D. To populate table XL, we insert all the rows from the three tables Safe Event A, Safe Event B and causality.

7. For Step 5, we create three tables viz. event A Safe, event B Safe and YL. All the three tables contain two columns (set A and set B) which are of varchar datatype.

A. We create a stored procedure to split the values of column set B of table Safe Event A on comma separator. Insert the results in safe A table.
B. We create a stored procedure to split the values of column set A of table Safe Event B on comma separator. Insert the results in safe B table.
C. To populate table event A Safe, insert all the rows from table safe A.
D. To populate table event B Safe, insert all the rows from table safe B.
E. To populate table YL, insert all the rows from tables Safe Event A, Safe Event B, event A Safe, event B Safe and causality.

8. For Step 6, we create two tables viz. terminal Place that contains a single column (event) which is of varchar datatype and PL which also contains a single column (Place) which is of varchar datatype.

A. To populate table terminal Place, insert 'i' and 'o' in the table.
B. To populate table PL, we use concat_ws 10 to combine the values of column set A and column set B of a table YL using & separator and insert the results in table PL. Furthermore, we insert all the rows of table terminal Place into table PL.

9. For Step 7, we create 3 tables viz. Place1 and Place2 which consist of two columns (id and value) which are of varchar datatype and FL which consists of two columns (first place and second place) which are of varchar datatype.

A. To populate table Place1, we use concat_ws to combine the values of column setA and column setB of table YL using & separator. Insert the results in column set B of table Place1. Insert all the values of column set A of table YL into column set A of table Place1.
B. To populate table Place2, we use concept _was to combine the values of column set A and column set B of table YL using & separator. Insert the results in column set A of table Place2. Insert all the values of column set B of table YL in column set B of table Place2.
C. We create a stored procedure to split column set B of table Place1 on comma Separator.In stored procedure we create table temp_place2 to insert the results.
D. We create a stored procedure to split column setA of a table Place2 on comma separator. In stored procedure we create table temp_place2 to insert the results.
E. To populate a table FL, insert all the rows from tables temp_place1 and temp_place2. Insert the results of cross join of two tables viz. terminal Place and intial Event and of table final Event and table terminal Place.

ALPHA – MINER ALGORITHM

The basic ordering relations determined by α -Miner Algorithm are the following:
The two events a and b from the set of activities A belonging to a process M recorded in an event log W. We say sub-string ab occurs in event log W ($ab \in W$) if $N(ab) > 0$. If $ab \in W$, then b directly follows a in at least one trace, which can be write as $a > b$.
The following four possible relations between a and b:
• $a \rightarrow b \Longleftrightarrow N(ab) > 0$ and $N(ba) = 0$;
• $b \rightarrow a$ (which we can also write $a \rightarrow^{-1} b$);
• $a \# b \Longleftrightarrow N(ab) = N(ba) = 0$;
• $a \parallel b \Longleftrightarrow N(ab) > 0$ and $N(ba) > 0$.

Two activities are always related by either $\rightarrow$, $\rightarrow^{-1}$, # or $\parallel$, and these partition the set of activities.
The Alpha algorithm derives an SWF-Net α (W) from an event log W. First, create three sets of activities. TW is the set of transitions in the net, containing all unique activities from A that occur in W. TI and TO contain the 'input' and 'output' transitions in the net, activities that appear first and last in traces in W, respectively (obtained by auxiliary functions first(x), last(x)) (van der et al., 2004).
In step 2 sets $R_>$, $R_\rightarrow$, $R_\parallel$, $R_\#$ will be created. Elements of $R_>$ are pairs of activities related by the > relation, and similarly for the other relations.
Places PW in the net are defined in two stages. First (steps 3–18), XW is created as the set of pairs (A,B) of sets of transitions for which each activity b in B is a successor of each activity a in A, i.e. $a \rightarrow b$, and all activities in A are related by the # relation, as are activities in B.
Thus,
$XW = \{(A,B) \in \{P(TW) \times P(TW)\} \mid$
$\forall a \varepsilon A \forall b \varepsilon B\ a \rightarrow b \wedge \forall a1,a2 \varepsilon A\ a1 \# a2 \wedge \forall b1,b2 \varepsilon B\ b1 \# b2\}$,
Where P(TW) is the power set of TW (the set of all subsets of TW).

XW is created by iterating over all possible pairs of activities (a, b) $\in$ TW $\times$TW. If two
Such activities a, b are related by a $\rightarrow$ b, a pair of sets (A,B) is created; A = {a},B = {b},
and TW is then iterated over again (steps 8–15).
Each activity that correctly relates to

```
Algorithm 1 Alpha Process Mining Algorithm
Input: W, an event log over activities A.
Output: an SWF-Net $\alpha(W) = (PW, TW, FW)$.
1: $TW \leftarrow \{a \in A \mid \exists x \varepsilon W\ a \in x\}$.
TI $\leftarrow \{a \in A \mid \exists x \varepsilon W\ a = first(x)\}$.
TO $\leftarrow \{a \in A \mid \exists x \varepsilon W\ a = last(x)\}$.
(Functions first(x), last(x) return the first and last activities, resp., of trace x.)
2: $R> \leftarrow \{(a, b) \in A \mid a > b\}$, $R\rightarrow \leftarrow \{(a, b) \in A \mid a \rightarrow b\}$,
$Rk \leftarrow \{(a, b) \in A \mid a\ k\ b\}$, $R\# \leftarrow \{(a, b) \in A \mid a\#b\}$.
3: $XW \leftarrow \emptyset$.
4:  for $a \in TW$ do
5:     for $b \in TW$ do
6:       if $(a, b) \in R\rightarrow$ then $A \leftarrow \{a\}, B \leftarrow \{b\}$
7:           $T'W = TW$. Add activities to A,B:
8:              for $c \in T'W$ do
9:              if $\forall d \in A, (c, d) \in R\# \wedge \forall e \in B, (c, e) \in R\rightarrow$ then $A \leftarrow A \cup c$.
10:              end if
11:             if $\forall d \in B, (c, d) \in R\# \wedge \forall e \in A, (e, c) \in R\rightarrow$ then $B \leftarrow B \cup c$.
12:          end if
13:     $T'W = T'W \setminus c$.
14: end for
15: $XW \leftarrow XW \cup (A,B)$.
16: end if
17: end for
18: end for
19: $YW \leftarrow \{(A,B) \in XW \mid \forall (A',B') \in XW\ A \subseteq A' \wedge B \subseteq B' \Longrightarrow (A,B) = (A',B')\}$.
20: $PW \leftarrow \{p(A,B) \mid (A,B) \in YW\} \cup \{iW, oW\}$.
21: $FW \leftarrow \{(a, p(A,B)) \mid (A,B) \in YW \wedge a \in A\} \cup \{(p(A,B), b) \mid (A,B) \in YW \wedge b \in B\}$
$\cup \{(iW, t) \mid t \in TI\} \cup \{(t, oW) \mid t \in TO\}$.
```

MYSQL 5.7

MySQL Database is an open source, multithreaded and multi-user Relational Database Management System which is affiliated with Oracle Corporation. The general availability of MySQL Server 5.7 was announced in October 2015. The top features are mention below [10]. The top features are mention below Replication, InnoDB, Optimizer, Security, Performance Schema, OpenGIS Classes, Trigger, Partitioning, SYS Schema JSON Datatype, ClientPrograms, libmysql Client, Building Changes.[14]

ORACLE 12C

Oracle Database is referring to RDBMS (Relational Database Management System) is an object-relational database management system which is introduced by Oracle Corporation on 22 July 2013 [16]. Oracle also provides a tool for modelling of UML, data warehousing designing etc Oracle database is a most popular database for security purposes. The top features are mention below. Improved Column Defaults, IncreaseSize Limit, Improved Top N-Queries, Temporary UNDO, Invisible Column, Database Archiving, OnlineMigration, Transaction Guard, PGA Aggregate LimitSetting.[17]

2.2 PERFORMANCE COMPARISON OF MINING ALGORITHM ON ROW ORIENTED DATABASE

Hasso conducted common database approach for OLTP and OLAP using an in- memory
Column database [18]. He presented a comparison of OLAP and OLTP considering row-
Oriented database Rana et al. implement Apriori algorithm on MYSQL and Oracle
database and compare their performance in terms of execution time [21].

NOVEL CONTRIBUTION

In context of existing work, this study presented here makes the following novel contributions. The work presented in this paper is extension of the work presented in [21][10]. The study presented in this paper has several more results which are not present in [21][10].

1. While there has been work done on implementing data mining algorithms on row-oriented databases, we are the first to implement Process Mining -miner algorithm on MySQL.
2. While there has been work done on implementing data mining algorithms on row oriented databases, we are the first to implement Process Mining -miner algorithm on Oracle.
3. We present a performance benchmarking and comparison of -miner algorithm on both MYSQL and Oracle. We consider multiple aspects such as -miner stepwise execution, bulk loading across various datasets, write intensive time, read intensive time and disk space of tables.

EXPERIMENTAL DATASET

Business Process Intelligence 2013(BPI 2013) dataset was used to conduct the experiments. The log contains events from an incident and problem management system called VINST. The event log was prepared from Volvo IT Belgium. The dataset was provided in CSV format. The VINST cases incidents CSV file dataset was used, which contained 65534 records to conduct the experiments.

Since α algorithm was concerned only with the sequence of events, and we need to find all transitions that a problem can go through and compute causality between the activities, only three fields in our dataset were considered that have been highlighted in red of table 4.2- SRNumber which was unique for a particular problem, ChangeDatePlusTime which was the timestamp value giving the sequence order of activities for a particular SRNumber and Activity which represented the activities in our data model.
Thus, only three fields in our eventlog table were considered, and corresponding to each SRNumber we obtained the sequence of events as the events were ordered according to the timestamp values[5].

Performance Comparison

For comparison of performance of Oracle and MYSQL we have used 64 bit windows operating system with 4 GB of RAM and 2GB of hard disk.

The α-miner algorithm interacted with the database. The underlying data model for implementing α-miner algorithm consists of 3 columns (SRNumber, ChangeDatePlusTime, Activity) each of which were of datatype varchar except Timestamp which was of timestamp datatype. The primary key was a composite primary key consisting of SRNumber and ChangeDatePlusTime. Each reading was taken five times, and the average of these readings was recorded in the paper.
First experiment consists of investigating the time taken to perform bulk loading in both the databases across various dataset sizes [6].

Below graph shows that the average time taken to load data in MYSQL was lower as compared to Oracle. Therefore, performance of MYSQL was better as compared to Oracle, because the percentage increase of time in MYSQL was more as compared to oracle.[21]

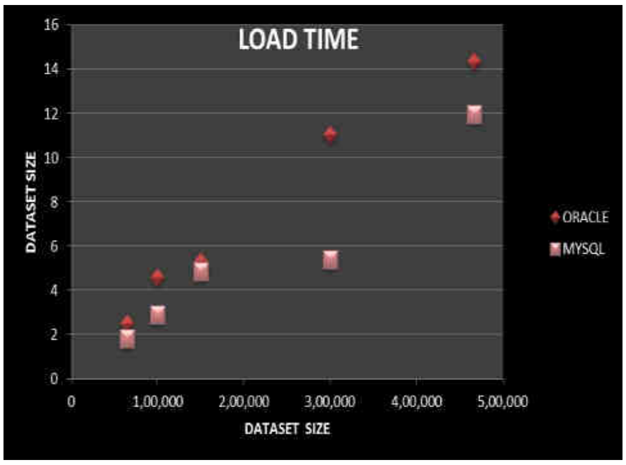

FIGURE1 DATASET LOAD TIME IN SECONDS

In this experiment the α-miner algorithm execution time was computed for each step in both MYSQL and Oracle to examine which database performs better for each step. It revealed that the stepwise time taken in oracle was always lower as compared to MYSQL for all the Steps.

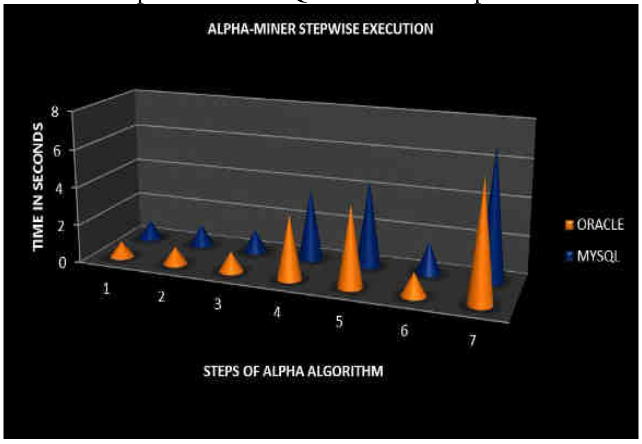

FIGURE 2: ALPHA-MINER STEPWISE EXECUTION

We conducted an experiment to investigate which of the steps of α-miner algorithm are read intensive as well as we compared which of the database performs better for read operations. From the experimental results, it was concluded that Oracle gives better read performance as compared to MYSQL for all the Steps.

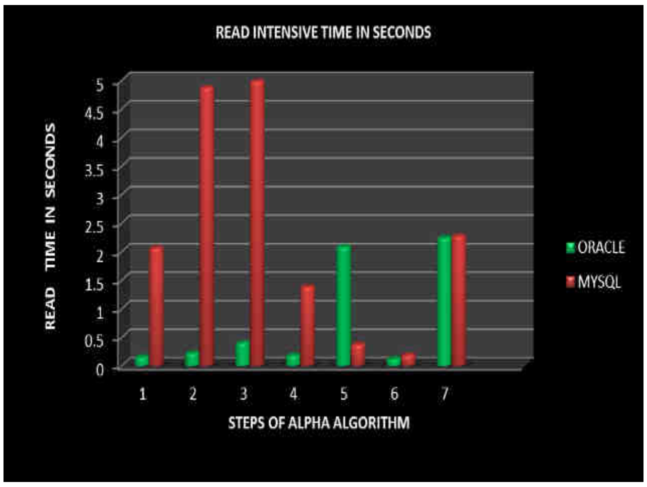

FIGURE 3: READ INTENSIVE TIME IN SECONDS

In this experiment it was investigated that which of the Steps of α-miner algorithm were write intensive as well as comparison was done that which of the database performs better for write operations. It was shown that the write performance of Oracle was better as compared to MYSQL. The reason of oracle giving better write performance can be the difference in the way writes were performed in both the databases

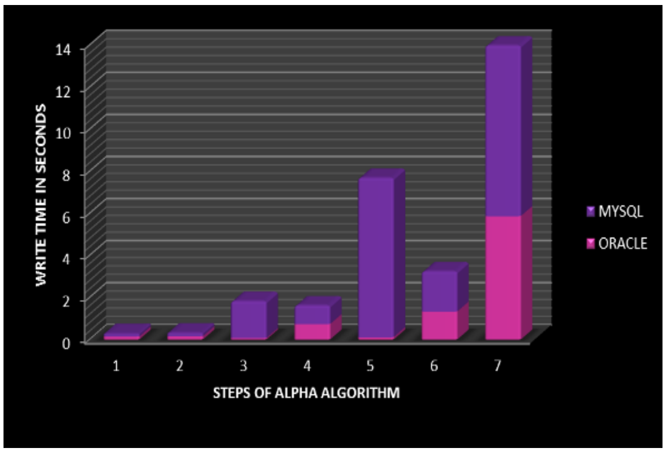

FIGURE 4: WRITE INTENSIVE TIME IN SECONDS

An experiment to investigate which database can efficiently store results of each Step of α-miner algorithm in tables with minimum disk space.
The reason for ORACLE occupying more space is the difference in the way memory is allocated to tables in both the databases.

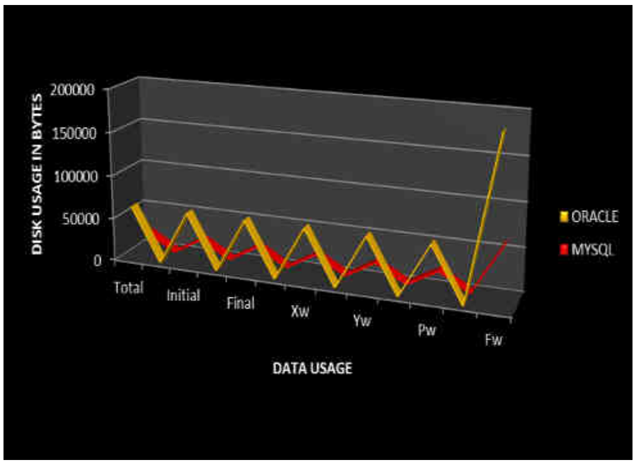

FIGURE 5: DISK USAGE OF TABLES

CONCLUSION

In this paper, we present an implementation of α-miner algorithm in MySQL and Oracle of process mining. Furthermore, we present the performance benchmarking and comparison of α-miner algorithm on MySQL and Oracle. The α -miner implementation in MySQL is a one tier application which uses only standard SQL queries and advanced stored procedures. This work concentrated on defining writing efficient queries and suitable tables.

Generally, for the small and medium size of the management system, a relational database is a good choice but we cannot say that it is best criteria it also depends on the purpose of the company. Selecting MYSQL is the best decision but when database designer creating a database they may take much consideration like load transaction, the different operation applied on a database according to requirement. Thus most of the companies need open source database that complete their needs and Oracle database are available at a high price while MYSQL is open source, which is the big advantage for companies while Oracle database are available at a high price and once installed system can require significant resources, therefore hardware upgrades may be required to even implement Oracle. But it is ideal for the large organizations that handle enormous databases and need a variety of features.

In end, choosing right database is not only depending upon the execution performance but also depend upon particular situation according to their factors that are dependent upon the database.

REFERENCES

[1] Abramova, V., Bernardino,J., and Furtado, P., (2014)" Experimental Evaluation Of Nosql Databases", International Journal of Database Management Systems,6(3).

[2] Arasteh, A. D., Davud, M., Majid, M. (2014), "MapReduce Based Implementation of Aggregate Functions" International Journal of Electronics Communication and Computer Technology 4(3).

[3] Bhadoria, P. S., Chhabra, A., Ojha, S., (2011)," SQL PERFORMANCE TUNING IN ORACLE 10g AND 11g", International Association of Scientific Innovation and Research, 2279-0063.

[4] Commercial and Open Sourced databases [homepage onthe Internet]., Access On 28-Oct-2016, http://dbengines.com/en/ranking_osvsc.

[5] Cook, Jonathan E., and Alexander L. Wolf. "Discovering models of software processes from event-based data." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 7.3 (1998): 215-249.

[6] Gupta, K., Sachdev, A., Sureka, A.: Pragamana: Performance comparison and programming alpha-miner algorithm in relational database query language and nosql column oriented using apache phoenix. In: Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering. pp. 113–118. C3S2E '15 (2008).

[7] Kundra, D., Juneja, P., Sureka, A.: Vidushi: Parallel implementation of alpha miner algorithm and performance analysis on cpu and gpu architecture. In: International Conference on Business Process Management. pp. 230–241. Springer (2015)

[8] Letkowski J. "Doing database design with MySQL", Journalof Technology Research Volume 6, 2014.

[9] L.Suresh and Simha B.Jay., (2008), "Efficiency Analysis of Efficient SQL based Clustering Algorithm", International Journal of Computer Science and Network Security,8(11),293-298.

[10] MS SQL Server [homepage on the Internet]. Database, cited 2016 Oct 28,Available from https://www.microsoft.com/enus/ql-server/sql-server-2016.

[11] MySQL Database [homepage on the Internet]., Access On28-Oct-2016,http://www.mysql.com/products/community/

[12] Microsoft SQL Server, Access On 28-Oct-2016, https://en.wikipedia.org/wiki/Microsoft_SQL_Server.

[13] MS SQL Server 2016 Features, Access On 28-Oct-2016, http://www.databasejournal.com/features/mssql/slideshows/10-new-features-worth-exploring-in-sql-server-2016.html.

[14] MySQL Server 5.7 Features, Access On 28-Oct-2016, http://dev.mysql.com/doc/refman/5.7/en/mysql nutshell.html

[15] Oracle Database [homepage on the Internet]. Databases cited 2016 Oct 28 Available from https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm

[16] Oracle 12c Database History, Access On 28-Oct-2016, https://en.wikipedia.org/wiki/Oracle_Database.

[17] Oracle 12c Database Features, Access On 28-Oct-2016, http://allthingsoracle.com/oracle-database-12c-new-featurespart-i/

[18] Rani, P. , Singh, P. , Sharma, H., (2011) "Self Tuning of Oracle Database Using SQL Scripts", IJCSI International Journal of Computer Science 8 (4(2)), 1694-0814

[19] Sonawane, S. B., Patki, R.. P.,(2015) "Process Mining by using Event Logs", International Journal of Computer Applications, 116 (19), 0975 – 8887

[20] Tiwari, A. and Turner, J. C.,(2008), "A review of business process mining: state-of-the-art and future trends", Process Management Journal,14 (1), 5-22.

[21] Virpura, D., Dr. Swaminarayan, P., (2015), "A Comparative Study of Oracle Application Express with .NET", International Journal of Advance Research in Computer Science and Management Studies, 3(2)

[22] Van der Aalst, Wil, Ton Weijters, and Laura Maruster. "Workflow mining: Discovering process models from event logs." *Knowledge and Data Engineering, IEEE Transactions on* 16.9 (2004): 1128-1142.

[23] W.V.D.Aalst: Process Mining: Overview and Opportunities. ACM (2012)

[24] Zhang, C., Sterck, H.D.: Hbasesi: Multi-Row Distributed Transactions with Global Strong Snapshot Isolation on Clouds. Scientific International Journal for Parallel and Distributed Computing (2011).