Use of Outlier Detection in Database Availability Analysis

Mario Žgela
Croatian National Bank,
Trg hrvatskih velikana 3, 10000 Zagreb, Croatia
Mario.Zgela@hnb.hr

Abstract. Usually, database is available and accessible by authorized users. However, there are abnormal situations resulting in database unavailability. Since irregularity may be related to the notion of outliers, in this paper it is checked if outlier detection may be helpful in finding out abnormality in database usage and so improving the process of database availability analysis.

Keywords. outlier detection, database availability, quartile range, database login

1 Introduction

Applications and database platforms are usually in the centre of each information technology (IT) organizational unit efforts to improve efficiency, quality and functionality of information system. Application system platforms cover application programs which correspond to business processes. Database platform stores and manages data which is unique to business entity and correspond to information created, changed and used within the enterprise. Consequently, availability of application and database platforms is of a crucial importance for every IT and business system. Furthermore, even if the application system programs and runtime environments are fully operational, database system availability and operability is at least of the same level of importance. Unavailability of database system impacts users (employees, managers, owners, customers and other external parties) who can not operate and query the business data, execute transactions and seek information stored within database. That usually means important part of business activities can not be registered, controlled, checked against errors and omissions, reports can not be generated. Managers are prevented from decision making, employees and other users from execution of regular business processes. As more and more business entities are increasingly dependent and base their existence on their IT systems, particularly databases, unavailability of that IT component means business will underperform or even not perform at all. Those are the reasons why database availability analysis is of utmost importance.

2 Description of the Problem

2.1 Definitions of major terms

There are a number of various definitions of outlier data. For example, in [10, 1] it is said that outliers are values that lie very far from the middle of the distribution in either direction. Also, slightly different definition is given in the same paper which states an outlier is a single, or very low frequency, occurrence of the value of a variable that is far away from the bulk of the values of the variable. Furthermore, in [11, 1] it is noted outliers are set of observations whose values deviate from the expected range.

In [1] it is explained that an outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism. Outliers are also considered as abnormalities, discordants, deviants or anomalies in the data mining and statistics literature. In [7, 1] outlier is defined as an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data. However, it is stressed that identification of outlier data is not clear since the suspicious observations may be outcome of low probability values from the same distribution or perfectly valid extreme values.

In [8, 544] it is presented outlier is a data object that deviates significantly from the rest of the objects, as if it were generated by a different mechanism. It is said that outliers are abnormal, while data objects that are not outliers may be referred as normal or expected data. Outliers are different from noisy data as noise is random error or variance in a measured variable. In general, noise is not interesting in data analysis, including outlier detection, so it has to be removed before outlier detection. Otherwise, treating each noisy data may cause too many false alarms and thus heavy time consumption and costs.

Database availability is desired state of database in which it is accessible by all authorized users and applications, permits querying, updating, inserting and deleting data. Database availability management includes procedures for collection of specific database events, defining and enforcing rules of database

availability and notifying that database is not available.

2.2 Importance of database availability analysis

It is of a particular importance that database availability is ensured as much as business requests and as much managers and owners are willing to invest in database availability. Although assurance of requested database high availability may be extremely complicated and costly, it is possible to create reasonable solutions for database availability analysis with quite small investments. For the purpose of this paper, database availability analysis covers procedures for collection of specific database events data, defining and enforcing rules of database availability and notifying that database is not available.

In order to improve database availability, it is firstly necessary to understand:
a) what is the current level of database availability - what is the percentage of data unavailability/availability in certain time period
b) what is the requested and agreed level of database availability

When assessing the current level of database availability, it is of a major importance to define what is database availability and when database has to be available. Database availability is the state of database management system in which it can accept and execute requests for logins, business application queries, data definition requests and perform data transactions (updates, deletes and inserts).

In the past, period of time in which database had to be available was defined by regular business entity operating hours. Today, however, it is usual that database has to be available around the clock, 24x5 or even 24x7. That is mostly the consequence of accepting external parties' requests and managing their transactions over the internet. Outcome is that importance of database availability improvement becomes in the focus of IT teams and unavoidable area of disaster recovery and business continuity management.

In this paper, it will be explained how the level of database availability may be measured by assessing current database status in comparison to historical activity data. The prerequisite for such measurement may be established with the help of database audit trail and by calculation related to data outliers. As it is noted in [13], an audit trail (or audit log) is a security-relevant chronological record, set of records, or destination and source of records that provide documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure, or event. Audit records typically result from activities such as financial transactions, scientific research and health care data transactions, or communications by individual people, systems, accounts, or other entities. The process that creates an audit trail is typically required to always run in a privileged mode, so it can access and supervise all actions from all users; a normal user should not be allowed to stop/change it. Furthermore, for the same reason, trail file or database table with a trail should not be accessible to normal users. Another way of handling this issue is through the use of a role-based security model in the software [13].

Database audit trail must at least store data on successful logins. In this research successful login event will be the basis for assurance that database is active and available. If users can connect to a database, it is a principal proof that database is up and running. If users can not even login, then it is obvious database is unavailable for them and no data transaction or queries can not be performed, at least for those users who are not logged in. Of course, there can be situations in which:
a) users can login, but can not perform any data manipulation
b) users that are currently logged in can perform data manipulation, but no additional users can login to database

In the a) case, database may be considered unavailable although logins can be performed, while in the b) case database is partially unavailable because some users (those currently logged in) can use database features and manipulate data. As an ultimate case, if there are no users that are logged in and no users can login it is clear that database is completely unavailable to business users. It is reasonably easy to discover such case since the situation is obviously outlying if no user logged in during regular working or operating hours. The more complex case is when some users can and some users can not execute DB login. Such situations may arise:
- when client computer database access software for specific number of users failed to properly execute
- if some specific switch crashes, thus leaving dependent group of users without network services
- if database can not fulfil some of new login requests because of problems with database internal processes
- if application server services are incapable of handling all users trying to connect to database through business application.

Abovementioned situations are much harder to detect than obviously outlying incidences because of complexity of various IT platforms on which problems may arise (e.g. client computer software, network, database system itself and application server).

Whatever, the idea of this paper is to focus on number of successfully performed login procedures during certain period of time. If number of successful logins is significantly smaller during observed time period then it may be concluded database is unavailable to significant number of users. Prerequisite is that statistical data is continually collected in order to define what is usual number of

logins executed during certain time periods. This data is particularly important since number of regular logins may vary during the day. For example, it may be usual that number of logins is much greater during morning than during late evening. This statistical data serves as a starting point for defining regular and outlying situations concerning successful database logins. The notion of outlier can fit in as a quite reliable measurement of significant deviation from regular situations.

2.3 Importance of outliers

Some authors [10, 1] insist that the main reason for finding outliers is associated with data quality assurance. Removing and replacing outliers can improve the quality of data and thus positive impact on the results of data analysis. Simple statistical estimates, like sample mean and standard deviation can be significantly biased by outliers, i.e. data which is far away from the middle of the distribution.

As it is noted in existing literature [11, 1], detection of outlier data is an important task in data analysis which involves identifying a set of observations whose values deviate from expected range. Those values can excessively influence the results of the analysis and certainly sometimes lead to incorrect conclusions. Therefore, it is very important to give proper attention to the outliers before making any specific analysis and especially before making any decisions based on deviated data.Certainly, there are different needs for outlier detection. Outlier data can significantly influence the results of statistical analysis and thus may lead to incorrect conclusions. Very often it is necessary to isolate outliers in order to preserve data quality assurance. Sometimes, outliers are the result of erroneous data creation, e.g. inappropriate measurement method or instruments. In these cases, it may be advisable to exempt outlier data before analysis.

In [4, 1] it is said that when a large number of variables are sampled, one of the first steps should be detection of outlying observations. Although outliers are often considered as an error or noise, they may carry important information. It is therefore important to identify them prior to modelling and analysis.

So there are cases in which outliers are of a special interest, and other, so to say "regular" data is exempted from analysis and further observations. As it is noted in [1, 1], when the generating process behaves in an unusual way, it results in the creation of outliers. Therefore, there are occasions when an outlier contains useful information about abnormal characteristics of the systems and entities, which impact the data generation process. The recognition of such unusual characteristics provides useful application-specific insights. Some examples noted in [1, 1] include network intrusion detection systems, credit card fraud, interesting sensor events, medical diagnosis, unusual patterns in law enforcement processes, weather and climate change patterns in earth science. In all these examples the data has a "normal" model, and anomalies are recognized as deviations from that normal model. In many cases such as intrusion or fraud detection, the outliers can only be discovered as a sequence of multiple data points, rather than as an individual data point. For example, a fraud event may often reflect the actions committed in a particular sequence. The specificity of the sequence is relevant to identifying the anomalous event which is in fact the outlier.

As it is indicated in [5, 1], the importance of outlier detection is due to the fact that outliers in data denote significant information in a number of application domains. An anomalous traffic pattern in a computer network could mean that a hacked computer sends out sensitive data to unauthorised destination. Outlier detection is widely used to detect anomalous patterns in patient medical records which could be symptoms of a new disease. Similarly, outliers in credit card transactions data may indicate credit card theft or misuse. Outliers can also translate to critical entities such as in military surveillance, where the presence of an unusual region in a satellite image of enemy area could indicate enemy troop movement. Or anomalous readings from a space craft would signify a fault in some component of the craft.

In this paper it will be shown how finding outlier data within specific type of database system events may improve database availability analysis. It will be explained how identification and analysis of outlier database login frequency can be used in database availability analysis and even as a model for early warning that a database system is partially or not accessible at all.

3 Data and Research Method

3.1 Description of Collected Data

Technically, data about database events may be gathered from at least two sources. Firstly, data may be collected from database audit trails which are set up by database administrators and may register various events including user logins, queries, data manipulation instructions, internal database processes, data definition commands, user logouts etc. Database audit trails store all events that database administrator configured, no matter which runtime application environment or database management tool is being used. Usually, database activities executed by typical user will be registered in database audit trail, i.e. those activities can not be circumvented by the user no matter what kind of software he uses. Database audit trails are usually configured in order to register numerous activities initiated by database processes. Also, this type of events collection does not include application specific events like application window opening and closing, starting and closing reports,

clicking application objects and thus performing certain business process.

Application specific events may be collected by application audit trail which can be programmed simultaneously during application logic programming. Along with that, information about window name, client operating system, username, application system name and numerous other application properties can be included in the data collection. Application audit trail is focused on events initiated within application and not only within database. So, the richness of application audit trail may be significantly greater and more interesting for business purposes than those of database audit trail generated by database system alone.

For the purpose of this paper, application events stored in Croatian National Bank application audit trail were analyzed.

As it is already mentioned, in the focus of research are events related to database login process to specific business application in the period between October 1st 2012 and May 31st 2013. By carefully avoiding seasonal influences (e.g. summer time when number of logins and application use is considerably less than during the rest of the year), seasonal fluctuation is excluded from the sample. Furthermore, just logins denoting regular working hours (7:00 - 17:00, Monday to Friday) are analyzed in order to avoid login events related to testing, administration and maintenance. The goal was to observe only regular user logins related to business activity, excluding logins performed by IT experts for the technical purpose. Of course, if the focus of research was intrusion detection or fraud prevention, then analysis of logins out of regular business hours would have remarkable sense.

However, total of 46,426 logins are registered within application in focus between October 1st 2012 and May 31st 2013 during daily time period from 7:00 to 17:00. Data is collected from production transactional database of Croatian National Bank. Row structure and description of data is as follows:

Column name	Data type	Description
ID	NUMBER	Ordinal number of event
TIMEST	DATETIME	Timestamp of event (date + time in format dd.mm.yyyy hh24:mi:ss)
APPLICATION	CHARACTER	Application name from which event is initiated
USER	CHARACTER	Name of the user who initiated the event

Since all data denote login activity and this research does not analyze particular user activity, but only number of logins in specific time periods, TIMEST column is in the focus of interest.

3.2 Some Outlier Detection Methods

There are different methods for outlier detection i.e. suspicious observations that would require further analysis.

One of the most widely used methods is so called boxplot [7, 2], which will be used in this research. The main building elements of a boxplot are the median, lower quartile (Q_1) and upper quartile (Q_3). Cut-off points lie $k(Q_3-Q_1)$ above the upper and bellow lower quartile. Observations beyond cut-off values are considered potential outliers. So, lower cut-off point is $Q_1-k(Q_3-Q_1)$ and $Q_3+k(Q_3-Q_1)$ is upper cut-off points. It means that outliers (O) lie in the following intervals:

$$Q_1-k(Q_3-Q_1)>O>Q_3+k(Q_3-Q_1) \quad (1)$$

This method is based on the graphical technique of constructing a boxplot which represents the median of all the observations and two hinges, or medians of each half of the data set [10, 2]. Most values are expected in the interquartile range (H) located between the two hinges. Values lying outside the ±1.5H range are termed "mid outliers" and values outside the boundaries of ±3H are termed "extreme outliers".

Although k is very often set as 1.5, in this research, boxplot method with k=1 is used for outliers detection. Lower outlier limit is marked H_L, while upper outlier limit is marked H_U.

$$H_L= Q_1-k(Q_3-Q_1) \quad (2)$$

$$H_U= Q_3+k(Q_3-Q_1) \quad (3)$$

$$Q_1-Q_3+Q_1>O>Q_3+Q_3-Q_1 \Rightarrow 2Q_1-Q_3>O>2Q_3-Q_1 \quad (4)$$

The reason for such setup in this research lies in the login events distribution. During certain time periods, number of logins is quite small and if k is set to 1.5 then the lower outlier cut-off point would be negative which is absurd and impossible. However, if the number of login events during observed time period is 0 it is meaningful and indicates possible database unavailability.

Z-scores are also very popular method for outlier detection and has been implemented in different flavours and packages [7, 1-2]. Z-scores are defined as:

$$Z_{score}(i)=\frac{x_i-\bar{x}}{s}, \text{ where}$$

$$s=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\overline{x})^2} \quad (5)$$

A common rule considers observations with Z-scores greater than 3 as outliers, though the criteria

may change depending on the data set and the criterion of the decision maker. However, this criterion also has its problems since the maximum absolute value of Z-scores is (n-1)/$\sqrt{n}$ and it can be possible that none of the outliers Z-scores would be greater than the threshold, especially in small data sets [7, 2].

The problem with the previous Z-score is that $\overline{x}$ and *s* can be greatly affected by outliers, and one alternative is to replace them with robust estimators. Thus, $\overline{x}$ may be replaced by the sample median ($\overline{x}$), and s by the MAD (Median of Absoulte Deviations about the median) [7, 2]:

$$MAD=median\{|xi-\overline{x}|\} \quad (6)$$

Now the modified Z-scores are defined as:

$$M_i=\frac{0.6745(x_i-\overline{x})}{MAD} \quad (7)$$

Observations will be labelled outliers when |Mi|>D. Some authors suggest using D=3.5 relying on a simulation study that calculated the values of D identifying the tabulated proportion of random normal observations as potential outliers.

4 Analysis and Results

Among other data, database audit trail collects data about logins to the database system. These data may be used as a secondary means of identification of database availability. Although business entity may have other IT solutions for identification of database connectivity and availability problems, it will be shown how database login data collected by audit trail can be used as identification of database failures related to user connectivity. Surely, if user's can not connect (login) to a database and thus perform their regular business activity during working time, it is a principle indicator of database failure in the eyes of the average user. Of course, actual reason for inability to connect to a database may not be of database nature, but of problems arising out of network, operating system, client software and application server software. On the other hand, security event and information management software may not appropriately classify or even identify some particular database problems like unplanned database shutdown, frozen listener software, off-line tablespaces, corrupted data files, disk space full etc. If abovementioned groups of problems occur, they may be identified by analysis of database login events. If there are no successful database logins during some period of working time, when users are usually initiating connections, then obviously something unusual is happening in the business entity. There may be justified business reason for such behaviour (e.g. planned database shutdown), but more usually it is the consequence of serious IT failure. When users can not use application transactions based on database, it is obvious they are prevented from performing their everyday business tasks and activities.

Analysis can be performed both in real-time (on-line) as well as in reporting mode (off-line). In real-time mode, analysis is performed in parallel with data collection. With each login, data should be statistically processed and checked against limits defined for outliers. In reporting mode, analysis is performed after bulk of data is collected. At certain periods of time (end of working day, end of week, end of month) data are statistically processed and checked against limits designated for outliers.

Real-time mode should be used for on-line database monitoring, when immediate reaction to deviated events that fall bellow or above outlier limits is needed.

Reporting mode should be used for statistical purposes and post analysis of database availability, when immediate reaction to outlier events is not necessary.

For the purposes of database availability analysis, logins are grouped into 30 minutes intervals during regular working hours and 30 minutes intervals start at 7:00 and finish at 17:00.

Average number of logins per day is 285. Probably, average number of logins would be greater if inactivity timeout period would have been shortened. On the contrary, if inactivity timeout period would be longer, the average number of logins would be even less. Median of logins per day is 284. Some statistical data about number of logins during 30 minutes periods is shown in Table 1. The column 'period' denotes start of 30 min period, thus value 7:00 refers to time period from 7:00 to 7:30, while value 16:30 relates to 16:30 till 17:00 time period. The value Q_1 denotes lower quartile (median of lower or first half of observations), Q_3 denotes upper quartile (median of upper or second half of observations), H_L stands for lower outlier limit (2), while H_U denotes upper outlier limit (3).

Table 1. Statistical data about database logins (Q_1, median, Q_3, H_L, H_U)

PERIOD	Q_1	MEDIAN	Q_3	H_L	H_U
07:00	4	5	6	2	8
07:30	8	10	13	3	18
08:00	13	16	19	7	25
08:30	21	26	31	11	41
09:00	21	25,5	30	12	39
09:30	18	23	29	7	40
10:00	14	19	25	3	36

10:30	14	18	24	4	34
11:00	13	17,5	23	3	33
11:30	12	17	22	2	32
12:00	6	9	12	0	18
12:30	5	7	9	1	13
13:00	8	11	15	1	22
13:30	11	14	20	2	29
14:00	11	15	20	2	29
14:30	9	14	18	0	27
15:00	9	13	17	1	25
15:30	5	7	11	-1	17
16:00	2	3	4	0	6
16:30	1	1	2	0	3

Fig. 1 shows median, lower and upper outlier limits. Median values clearly show variations of logins which is outcome of business activity and working hours of business entity employees. Obligatory working hours start at 7:00 only for one department within the business entity dealing with real time gross settlement payment system. For other departments, working hours start between 7:30 and 9:00. The outcome is that number of logins continually rises from 7:00 till 9:00, then stays stable until 10:00, while continually but slowly drops until 12:00. It should not be forgotten that number of logins represents only new logins and not employees already logged on in previous time period. So, number of logins represent only new authentication events executed during observed timeslot and do not include total number of currently logged users or users logged in previous time period. It is clear that users are performing login procedure slightly after coming to work, and that is why number of logins rise until 9:00 and stays large until 10:00. It shows some employees are quite dependent on application system and database, because they login shortly after they come to work. Although this paper analyzes only login frequencies and not intensity of logoffs, it may be assumed employees usually stay logged on throughout longer period of time. Simply, after they execute login, employees does not perform logouts before lunch break. The effect of such situation is that number of logins decreases from 10:00 to 12:00.

From 12:00 to 13:00 number of new logins decrease and is relatively low, then increases from 13:00 to 14:00 and stays stable until 15:30. Then it drops until 17:00. During lunch break, it is logical that relatively low number of employees starts login procedure, while it is understandable that after they come from the break they re-execute login procedure and thus resulting in increased number of logins. As working hours approach to the end, less and less employees perform login procedure and thus start using application system.

Since in this research focus is set on database availability by the analysis of number of logins, the most interesting observations are lower outlier limits in observed time periods. Whenever number of logins is lower than H_L it is labelled as outlier. However, it should not be concluded that each outlier value is outcome of some especially problematic behaviour or situation. For some outliers there may be logical explanation which is related to decreased business activity or some other business reason.

Since number of database logins in certain time period falls into set of natural numbers including zero (N_0), H_L having negative values should not be taken into consideration. There is only one time period with negative H_L in this research (time period 15:30 to 16:00) and it may be concluded in that time period can not be outlier values. Even if the number of logins is 0, it is not an outlier value. IT may be concluded that even if there are no logins, it does not mean database is not available.

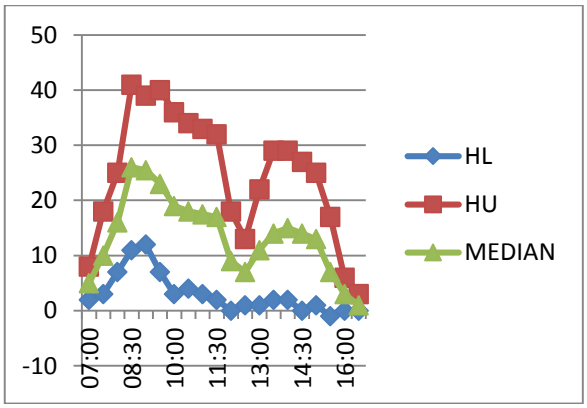

Figure 1. Statistical data about database logins (median, H_L, H_U)

Because lower outlier values must be less than H_L, it may be concluded the same must hold for all H_L=0. So, whenever H_L=0, lower outlier values do not exist. Even if there are no logins in time periods with H_L=0, it does not mean it is the proof of outlying behaviour. To conclude, for $H_L \in <-\infty,0]$ there are no outlier values in this research. So, specific time periods may be excluded from outlier analysis: 12:00 to 12:30, 14:30 to 15:00 and 15:30 to 17:00.

The next research step is to find number of logins less than lower outlier limits (NO_L) by time periods (Table 2.).

Table 2. Number of logins (NO_L) less than H_L per time period

PERIOD	NO_L
7:00	5
7:30	1
8:00	1
9:00	3

9:30	1
10:00	2
10:30	1
13:30	3
14:00	2

The analysis of timeslots with one or two observations less than lower outlier limit (7:30-8:30, 9:30-11:00, 14:00-14:30) shows there were no database availability problems. Database was available, as well as other IT components (application, operating systems, network), while significantly lower number of new logins was result of less business activity in specific time periods.

The second greatest median value (25.5) is noted for the period 9:00-9:30 (Table 1.), while there are 3 occasions (working days) when number of actual logins was less than lower outlier limit value (H_L=12).

Table 3. Number of actual logins (A_L) less than lower outlier limit (H_L) in 9:00-9:30 timeslot

DATE	PERIOD	A_L	H_L
7.2.2013	09:00	10	12
1.3.2013	09:00	8	12
23.5.2013	09:00	10	12

On March 1st 2013 number of actual logins in timeslot 9:00-9:30 was 30% less than lower outlier limit value (A_L=8 vs. H_L=12). Since deviation is the most significant, it is first in focus. IT is obvious database was not fully unavailable for all users, since eight new logins were executed in fairly even distribution throughout the period. Also, database monitoring software did not register any interruption within database management system. However, further investigation of IT components for March 1st 2013 revealed that specific network infrastructure problems occurred. Network switch used by all application users started to malfunction at 9:05 which was discovered by network traffic analyzer. Some of switch ports started to drop packets thus preventing some processes to execute requests and receive network response. That resulted in unavailability of network resources for all users connected to those ports which ultimately led to inability to login to database and prevented the use of application. Approximately half of total number of application users was affected by switch failure which resulted in outlying number of database logins. Since switch was quickly repaired during the 9:00-9:30 timeslot, outlying number of database logins was not identified during the next timeslot starting at 9:30.

On February 7th 2013, update of a few client computer operating systems related to observed application was scheduled. It started immediately after user logged on to workstation operating systems, before database logon procedure. That is the explanation why number of logins was significantly less than usual. After the update finished, users normally started login procedures and use database on regular basis. On May 23rd 2013 the same operating system update was scheduled and executed for second group of observed application users. Since update procedure again started immediately after operating system login, thus preventing users from executing database login, fewer logins were executed resulting in actual number of logins less than lower outlier limit. Both of these cases were not caused by database system fault but effectively decreased availability of database as users were not able to perform their everyday activities. These situations were afterwards discussed with business application owner and conclusion was made that number of client computers for which operating systems will be simultaneously updated should be decreased. Ultimately, fewer users are to be affected by update during the same timeslot and, consequently, more users will be able to login to database and so use business application. That is why client computer operating system update did not cause significant reduction in number of database logins anymore.

There are 5 observations, i.e. working days, with number of database logins less than lower outlier limit for the 7:00 to 7:30 time period. Lower outlier limit for that period is 2 (Table 1.) which means that there are 5 working days with no or just 1 login during that time period. Since during 7:00 to 7:30 time period critical activity for business entity should start with daily activities, this deserves additional attention and investigation.

Additional search through log data shows there are no days without logins in period 7:00 to 7:30. At least one new login was executed each day and there are exactly 5 days with only 1 new login in abovementioned timeslot. Further checks revealed that in these 5 cases business went as usual, and there were no availability issues. Application administrator was the only user who started business day and logged in, started application processes and performed application status checks while other users promptly started to login during timeslot 7:30 to 8:00 when the number of logins was significantly greater than usual.

There are 3 occasions (working days) when number of actual logins during 13:30-14:00 timeslot (A_L=1) was less than lower outlier limit value (H_L=2).

Table 4. Number of actual logins (A_L) less than lower outlier limit (H_L) in 13:30-14:00 timeslot

DATE	PERIOD	A_L	H_L
24.12.2012	13:30	1	2
31.12.2012	13:30	1	2
29.3.2013	13:30	1	2

Infrastructure status checks show no failures on information technology components during timeslots shown in Table 4. Database was up and running as well as network components, application servers, operating systems and client computers. Interestingly, all of the days with outlying number of database login events during 13:30-14:00 timeslot are eves of major holidays. Further investigation shows that business activity was significantly reduced during those three afternoons. Consequently, application activity was reduced, as well as number of database login events. The conclusion is that although number of logins is labelled as outlier data, there is no relation to information technology failure or database unavailability issue.

As it is explained, there were two types of events which resulted in significantly less database logins: partial network switch failure and client computer operating system updates. Both events prevented significant number of users from executing database login and using business application. So, it may be concluded that database system was partially unavailable, i.e. unavailable for some group of users, although database management system itself was fully functional and available. That is why database system monitoring software did not register unavailability - all database processes were up, running and functional. However, it may be noted that if users can not login, no matter what is the reason, then database does not fulfil its basic function: to ensure data retrieval, updates, inserts and deletes. Outlier analysis of database login events can surely help in discovery of database unavailability and enable early warning system. Quick reaction to outlying events may significantly shorten the database unavailability time and speed up the repair process.

Rest of the database login shortages in timeslots were not caused by information technology failure but because of business and organizational reasons. For example, and not surprisingly, on the eve of the major holidays (Christmas, New Year's Day and Easter) there was less database login events thus causing outlier data designation.

In order to assure prerequisites for greater database availability it is necessary to promptly react on outlying database login frequencies. This can be solved with development of active rule that would be triggered after the end of each timeslot. Active rule must check if number of database logins in observed and just finished timeslot is less than lower outlier limit value calculated for historical data. If it is, then the warning message should be delivered to technical staff and possibly to certain level of management because it is possible database is not fully accessible and available. Abovementioned cases shown that even if database login frequencies are significantly less than regular, it is not necessarily the proof that database is unavailable. Sometimes, low login frequencies may be result of business or organizational reasons and are not related to any technical malfunction. So, additional investigation must be carried out by technical staff in order to check functionality of IT infrastructure and applications. Sometimes management should be included in order to check if decrease in database logins is outcome of business and organizational issues.

5 Conclusion

The prerequisite for usage of outlier detection in database availability analysis is application or database audit trail which collects data events of interest. In this research it is shown how it is possible to make valid conclusions on database availability by use of just one set of database events: database login frequency. Total number of 46,426 database logins is collected for specific business application within Croatian National Bank and data is grouped into half hour timeslots. For each timeslot median, lower and upper outlier limit values are calculated. All login frequencies less than lower outlier limit value are additionally investigated. Three cases of partial database availability are discovered by the means of outlier detection application. It may be stated that it is possible to develop solution for promptly reaction on abnormally low database login frequency during specific timeslot which may be sign of database unavailability for some or all database users. As a conclusion, it is demonstrated that outlier detection may be efficient in discovery of partial or complete database inaccessibility.

References

1. Aggarwal, C. *Outlier Analysis*, http://www.charuaggarwal.net/outlierbook.pdf, accessed July 11th 2013.

2. Baisden, J. *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, http://www.redbooks.ibm.com/redbooks/pdfs/sg245690.pdf, accessed July 12th 2013.

3. Barnett, V.; Lewis, T. *Outliers in Statistical Data*, John Wiley & Sons, Chichecter, USA 1994.

4. Ben-Gal, I. *Outlier Detection*, http://www.eng.tau.ac.il/~bengal/outlier.pdf, accessed July 11th 2013.

5. Chandola, V.; Banerjee, A. et al. *Outlier Detection: A Survey*, http://www.bradblock.com.s3-website-us-west-1.amazonaws.com/Outlier_Detection_A_Survey.pdf, accessed July 10th 2013.

6. Cimbala, J. *Outliers*, http://www.mne.psu.edu/me345/Lectures/Outliers.pdf, accessed July 11th 2013.

7. Garcia, F. *Tests to Identify Outliers in Data Series*, http://habcam.whoi.edu/HabCamData/HAB/processed/Outlier%20Methods_external.pdf, accessed July 10th 2013.

8. Han, J. et. al. *Data Mining - concepts and Techniques*, Morgan Kaufmann Publishers, Waltham, 2012.

9. Hodge, V; Austin, J. *A Survey of Outlier Detection Methodologies*, http://eprints.whiterose.ac.uk/767/1/hodgevj4.pdf, accessed June 15th 2013.

10. Last, M.; Kandel, A. *Automated Detection of Outliers in Real-World Data*, www.researchgate.net, accessed May 28th 2013.
11. Tiwari, K.; Mehta, K. et al. *Selecting the Appropriate Outlier Treatment for Common Industry*, http://www.nesug.org/proceedings/nesug07/sa/sa16.pdf, accessed June 29th 2013.

12. Walfish, S. *A Review of Statistical Outlier Methods*, http://statisticaloutsourcingservices.com/Outlier2.pdf, accessed July 10th 2013.

13. ... *Audit Trail*, http://en.wikipedia.org/wiki/Audit_trail, accessed July 10th 2013.

14. ... *Detection of Outliers*, http://www.itl.nist.gov/div898/handbook/eda/section3/eda35h.htm, accessed June 3rd 2013.

15. ...*Outliers... How to detect them and when it's dishonest to remove them*, http://lon03.wordpress.com/2011/10/14/outliers%E2%80%A6-how-to-detect-them-and-when-it%E2%80%99s-dishonest-to-remove-them/, accessed June 10th 2013.