ODBC, Concurrency, and Transactions

Introduction

This article will focus on transactions and the affects that factors, such as concurrency, can have on transactions. I will start by introducing you to an interface that can be used for managing databases. Open Database Connectivity (ODBC) is an interface that provides tools for accessing and managing databases, including databases which involve interactions with a server as well as other applications. As stated on the Microsoft support page (2014), "Open Database Connectivity (ODBC) is Microsoft's strategic interface for accessing data in a heterogeneous environment of relational and non-relational database management systems."("ODBC--Open").

Transactions

When I worked in tech support as our team had supported software being used in a retail environment, I often troubleshooted issues that occurred with transactions. When an error occurred during a transaction, then a transaction would sometimes "fail". There had been many variables that affected which information had been saved for a transaction, where the file for a transaction had been sent, and the cause of an error that occurred during a transaction. For instance; if a transaction had been started and "background processes" had involved a lookup table or other database table being used to retrieve information for a transaction (e.g. item information or delivery information), and the data in the table had not been found or had been deemed as being invalid, then this could cause an interruption in the transaction. If a transaction had been interrupted, then it would fail. There had been cases in which a transaction could be recovered and reprocessed after the transaction had failed.

The behavior of transactions that I had troubleshooted demonstrates some of the general characteristics of transactions. Transactions involve processes which are processed in succession and if a process, such as a process included within a function for a transaction, does not complete successfully then this could result in the overall transaction not being able to complete. As Kroenke, D.M. & Auer, D.J. (2012) state, “A transaction (or LUW) is a series of actions to be taken on the database so that either all of them are performed successfully or none of them are performed at all, in which case the database remains unchanged.”(p.342).

Effect of Concurrency

For example and demonstration purposes, suppose I had come across a case of a transaction failing and found a copy of the transaction that had been saved and automatically sent to a directory that had been used for incomplete or failed transactions, then suppose that I send the transaction file to a user’s directory and a copy of the transaction file to a directory from which it can be “manually” reprocessed - would this cause issues with retention of data accuracy and originality of a transaction or other issues? It would be possible for the instance of a transaction being processed by someone “manually” and being processed by a user at a different computer to result in issues such as data discrepancies. Would this mean that the transactions would have been processed at the same time? Actually, if both copies or instances of the transaction would have been processed when the clock on both computers being used by myself and the other user processing the transaction had displayed the same time - along with the factor of the transactions retrieving the same data from the same database tables - the transactions would not have been processed within the same second. In regards to concurrent transactions, Kroenke, D.M. & Auer, D.J. (2012) make the following statement: “Although it may appear to the users that concurrent transactions are being processed simultaneously, this cannot be true because the CPU of the machine processing the database can execute only one instruction at a time.”(p.342).

Corrective and Preventative Options

As expressed earlier, possible issues that may result from concurrent transactions include discrepancies in data. Kroenke, D.M. & Auer, D.J. (2012) make the following statements, in regards to a case involving the same data being processed by more than one user: “Both users obtained data that were correct at the time they obtained them. But when User B read the record, User A already had a copy that it was about to update. This situation is called the lost update problem, or the concurrent update problem.”(p.345). In regards to remedies for some issues caused by concurrent transaction, Kroenke, D.M. & Auer, D.J. (2012) include the following remedy: “This remedy is called resource locking.”(p.345).

There are additional methods that can be used to help minimalize negative impacts from concurrent transactions and to help prevent concurrent transactions; some of the methods available which can be found within the ODBC include the Cursor operations. As stated on the Microsoft Developer Network page, “Cursor operations, like cursor types, are affected by the concurrency options set by the application. Concurrency options are set using the SQL_ATTR_CONCURRENCY option of SQLSetStmtAttr.”(“Cursor Concurrency”). Many options used with preventing concurrency related issues involve the use of locks placed for a database.

Conclusion

Many variables can affect processing of data and the possibility of an interruption in a process for a transaction. Due to the ways in which transactions should be successfully processes and the way in which data manipulation can affect consistency and accuracy in data, it is important to implement concurrency control. Not only does ODBC provides many tools which can help with managing and accessing databases, ODBC provides tools which assist with concurrency management.

Resources

Kroenke, D.M. & Auer, D.J. (2012). Database Processing Fundamentals, Design, and Implementation (12E). Pearson Education, Inc. : Upper Saddle River, NJ

Microsoft Developer Network. (2014). Cursor Concurrency (ODBC). Microsoft. Retrieved from http://msdn.microsoft.com/en-us/library/ms131308.aspx

Microsoft Support. (2014). ODBC--Open Database Connectivity Overview. Microsoft. Retrieved from http://support.microsoft.com/kb/110093

Stacey Pegues