Running Head: UNIT 1 LAB

Unit 1 Lab: Installing Oracle 11g R2 on Windows Server 2012 via Virtualbox

Jered McClure

Walden University

Unit 1 Lab: Installing Oracle 11g R2 on Windows Server 2012 via Virtualbox

Install Confirmation Screen on Windows 2012 via Virtualbox:



SHUTDOWN and STARTUP:



Data Dictionary Query:



SYSTEM Tablespace Information:



Database Users With number in dropdown box (top right):



Tables owned by user SCOTT:



Initialization Parameters of db\_block\_size



Question 1:

The four different shutdown commands for an Oracle database can each be used for different shutdown scenarios. SHUTDOWN NORMAL or just plain SHUTDOWN prevents any further database connections and rolls back any running transactions. This method of shutting down is the preferred method as it ensures the highest level of data integrity. All situations in which multiple users are expected to be logged into the database performing transactions should evoke a SHUTDOWN NORMAL command when required.

 SHUTDOWN TRANSACTIONAL is similar to SHUTDOWN NORMAL, however, it does not roll back transactions. Instead, it completes and commits all running transactions. Situations where a transaction must be completed for the sake of data integrity, and a rollback to a prior state would cause a dramatic loss of integrity should implement this version of SHUTDOWN. An example of this is in a banking system where online transactions occur all the time. These transactions can have a detrimental effect on a person’s balance if the transactions are not completed as required.

 SHUTDOWN IMMEDIATE forces the disconnection of all users and rolls back all transactions. This should only ever be used when it is certain that no transactions are running which are required for data integrity. An example of this is a development environment where running transactions may be a bug and user accounts are left connected even though no one is transacting. Subsequently, this version of SHUTDOWN should only be used in a production environment under extreme care and knowledge that a database restore may be required if data integrity faults arise.

 SHUTDOWN ABORT forcefully disconnects all users and cancels all running transactions. This version of SHUTDOWN should never be used in a production environment due to the guaranteed data integrity faults which will occur. Additionally, development environments should only use this version of SHUTDOWN when no other version of SHUTDOWN is possible. That is, if the database refuses to release system memory or a transaction cannot be halted through SHUTDOWN IMMEDIATE.

Question 2:

 SQL commands through a GUI can be easier to manage, provide human readable graphical output of data, and allow greater leverage over proprietary SQL language parameters. Additionally, GUIs can allow code to be saved and rerun via \*.sql scripts which can save on code writing time down the track. The downside to this is that these graphic user interfaces tend to not be 100% ANSI SQL compliant. As such, trying to transfer SQL from one database system to another can be a massive undertaking due to translating code from one SQL language type into another.

Question 3:

 Information removal from a redo log file is risky business in a production environment. In a development environment it is really up to the system developers as to whether they want the ability to roll back to a prior database layout. The production environment, on the other hand, needs to maintain a solid state of data integrity. Data should only ever be removed from the redo log if and when it is sure that the expected rollback state will never be required.

 Alternatively, log data can be purged based on a rolling schedule. That is, in a transactional environment, it is expected that after a given time, rollbacks will be unfeasible due to the loss of current data structures. Given this scenario, maintain a redo log which allows a rollback to an infinite period is a waste of resources. That being said, determining the length of redo log data should be brought before a panel of stakeholders to define purge intervals.

Reference

Poweel, G., & McCullough-Dieter, C. (2010). *Oracle 10G: database Administrator: Implementation & Administration.* Boston: Cengage Learning.