

List of tasks for OCM 11g exam course

www.ocmguide.com

Dear OCM practitioner! I have a great news for you! After clearing my second OCM exam, I decided to provide an online course and help you to prepare and pass the exam from the first attempt.

In this course I will provide you with the step by step installation and configuration guides and pre-built Virtual Machines to help you test all exam scenarios practically. Moreover, I will go through the entire topics of the exam and will explain every subject in more detail and show practical scenarios and assign different tasks for you.

The overall course duration is 2 weeks (10 days) / 30 hours! We are going to have limited seats for every class, so make sure you've booked your seat for the next class at www.ocmguide.com!

If you want to pass the exam from the first attempt and become a member of elite group of Oracle Masters and boost up your career, I'm here for you! Being 2xOCM and author of "OCM 11g exam Study Guide" book, I'm ready to help you in person.

If you have any questions related with the exam and the course itself, please use the chat that appears in the bottom of the web page to contact me directly.

Good Luck!

Sincerely,

Kamran Aghayev A.

Oracle Certified Master (10g/11g)

Oracle ACE Director, Author, Speaker, Blogger

President of Azerbaijan Oracle User Group (AzerOUG)

<http://www.kamranagayev.com>

<http://www.ocmguide.com>

Day 1/2 - Server Configuration

Practice	Time
Introduction	
Create a .bash_profile file	3-5 min
Install Oracle in a silent mode (configure response file, run the installation and track it)	10 min.
Create a database in a silent mode (using DBCA) Get the list of all parameters from dbca –help output Run dbca –silent command to create a database Check alert.log file	10 min
Drop database using “DROP DATABASE” command	5 min
Create a database using CREATE DATABASE command using OMF feature	15 min
Set different parameter files	10 min
Set MEMORY_TARGET and check the free memory	10 min
Set MEMORY_TARGET, SGA_TARGET and PGA_AGGREGATE_TARGET	15 min
Create locally managed, compressed, encrypted tablespace	10 min
Create tablespace with all available parameters	5 min
Create tablespace with different block size	3 min
Make tablespace READ ONLY/READ WRITE, OFFLINE, ONLINE, SHRINK, RENAME, ADD DATAFILE, RESIZE, DROP	10 min
Managing UNDO tablespace (create, switch, drop, set undo retention, enable and disable retention guarantee)	10 min
Create temporary tablespace, tablespace group, switch, drop, resize and shrink	10 min
Create, add, drop redo log group, member	5 min
Relocate datafiles of all database	10 min
Relocate controlfiles	8 min
Relocate one tablespace with 3 datafiles	8 min
Relocate datafile of system and sysaux tablespaces	8 min
Create a text parameter file, create server parameter file from text file. Change the value of any parameter and reset it back	6 min

Create a bigfile tablespace. Resize and change autoextend	2 min
Create a database with a default bigfile tablespace using CREATE DATABASE command	8 min
Configure DNFS	10 min
Deconfigure DNFS	5 min
Switch to documentation and create a simple listener.ora file without GUI. Make sure it works	3 min
Create a listener with different port	5 min
Open the documentation and create a simple tnsnames.ora file without GUI. Make sure it works	3 min
Create both listener.ora and tnsnames.ora file using netca utility	5 min
Enable shared server using all parameters, configure tnsnames.ora and connect to the db. Check if shared server connection works	7 min
Use TRACE_LEVEL and TRACE_DIRECTORY parameters, enable tracing and check if it works	5 min
Change TRACE_FILELEN and TRACE_FILENO parameters and check if it works	5 min
Use the rest trace parameters and make sure all of them works	10 min
Create a new database (second) with CREATE DATABASE command	10 min
Edit listener.ora and tnsnames.ora files and add new db information. Connect to the second database via TNS	5 min
Use easy connect method to connect to the database	1 min
Apply the patch	5 min
Rollback the patch	2 min
Install Grid Software	15 min
Register listener and database on Oracle Restart	5 min
Stop and Start Oracle Restart	5 min
Try all available srvctl and crsctl commands. Check the following Reference: Oracle® Database Administrator's Guide 11g Release 2 (11.2) 4 Configuring Automatic Restart of an Oracle Database SRVCTL Command Reference for Oracle Restart CRSCTL Command Reference Oracle® Real Application Clusters Administration and Deployment Guide 11g Release 2 (11.2) A Server Control Utility Reference	
Configure database control using emca utility	5 min

Day 3 - Enterprise Manager Grid Control

Practice	Time
Download necessary softwares and install OEM Grid Control. Apply the patch and check the status of the inventory	No time limit
Practice	Time
Download necessary softwares and install OEM Grid Control. Apply the patch and check the status of the inventory	No time limit
Use Memory Advisors to size the total memory	5 minutes
Use Undo Advisor to size the undo size	5 minutes
Change initialization parameters from OEM	5 minutes
Use OEM to startup and shutdown the database	5 minutes
Create a job to shutdown the database	5 minutes
Create a multi-task job that creates a single file. If it succeeded, then the second job will create the second file in the OS	10 minutes
Create a user managed metric on a database level that get the count of connected sessions and generate alerts on specified warning and critical thresholds	10 min
Create a user-defined metric to alert exceeded count of connected users. Check if it works.	10 min
Edit thresholds and schedules of pre-defined metrics	5 min
Create a job that creates a table once it runs	5 minutes
Create a job that inserts one row to the table. Create a scheduler that repeats the insert process every 5 minutes	10 minute
Implement above mentioned scenario	10 minutes
Create a program that get the size of the /home directory and writes the output to the OS file. Create a simple job and use this program	15 minutes

Day 4 - Managing Database Availability

Practice	Time
Create recovery catalog and register the database Create Tablespace, owner and grant necessary privileges Create catalog, register it and run LIST INCARNATION command	10 min
After creating the second database, create virtual private database and register the second database in it. Run LIST INCARNATION command	10 min
Create script to backup controlfile, run all script commands you're familiar with	10 min
After creating a new database and dropping all users and catalogs, create a catalog in each database and merge them	10 min
Use Backup retention policies	6 min
Change backup destination and use formatting	7 min
Configure controlfile autobackup. Make structural database changes and check how it works	5 min
Take compressed and encrypted backup and restore them	8 min
Configure MAXSETSIZE and MAXPIECESIZE	3 min
Configure backups for duplexing	5 min
Configure RMAN to skip 2 tablespaces (SYSTEM, SYSAUX) and take backup. Use NOEXCLUDE to override this parameter. Create two tablespaces. Make the first one READ ONLY, and OFFLINE the second one. Use SKIP READONLY and SKIP OFFLINE parameters	8 min
Enable controlfile autobackup, take the full backup to the specific directory	5 min
Allocate 4 channels for different folders and take the full backup splitted between 4 different folders	8 min
Backup system and sysaux tablespaces	1 min
Backup any datafile with controlfile included	1 min
Backup control file and parameter file	1 min
Enable block change tracking, take full backup and then incremental backup	5 min
Perform incrementally updated backup scenario	12 min
Take multisectional backup	min
Restore from SPFILE	2 min
Remove controlfiles, restore it from backup and open the database	8 min
Create a table on a specific tablespace, remove datafile, restore/recover it. Check the table	10 min

Create table on a new tablespace, remove datafile of the tablespace, restore/recover it without having any backup	10 min
Use SETNEWNAME command to restore datafile to the new destination. Restore a tablespace, a database to the new folder	20 min
Perform block media recovery	15 min
Remove all physical files of the database. Perform a disaster recovery	20 min
Take backup of the database, create a table, check SCN value. Truncate the table, perform incomplete recovery using SCN, TIME and SEQUENCE options	15 min
Try to repeat the previous incomplete recovery. Recover the database to the previous incarnation	5 min
Repeat prerequisites of incomplete database recovery scenario. Perform tablespace point in time recovery	15 min
Perform Flashback Database scenario	15 min
Change flash recovery area destination and make sure it works	5 min
Monitor flash recovery area usage by querying v\$flash_recovery_area_usage view.	10 min
Move redo log files, controlfile and all backups to the new flash recovery area	15 min
Flashback Query - Create a table. Insert some rows. Get current scn value (or timestamp). Delete all rows. Commit the transaction. Query deleted rows using Flashback Query	5 min
Flashback Version Query - Add supplemental log data. Create a table. Insert some rows. Get current scn value (or timestamp). Delete one row. Update two rows. Commit the transaction. Use pseudo columns to get versions of the rows	10 min
Flashback Transaction Query - Use FLASHBACK_TRANSACTION_QUERY view to see version of the rows in the transaction	5 min
Flashback Transaction Backout - Use this feature to rollback the first update transaction	10 min
Flashback Table - Create a table. Insert some rows. Get the current SCN value. Delete all rows and commit the transaction. Rollback all deleted rows	5 min
Flashback Drop - Create a new table. Insert some rows. Drop it. Check recycle bin and recover it back using flashback drop command	5 min
Flashback Database - Perform the scenario that's is provided in Flashback Database section of the chapter	15 min
Flashback Data Archive - Perform the scenario that's is provided in Flashback Database section of the chapter	15 min
Use all flashback archive parameters (retention, quota, purge historical data, drop, disable the flashback data archive and etc)	10 min
Query flashback archive views	5 min

Day 5 - Data Management

Practice	Time
Create a simple materialized view and DBA_* view to display it.	5 min
Create a materialized view, make sure all rows are populated after the view is created and make sure the view is truncated every time it is refreshed.	8 min
Create a materialized view and make sure it pull changes made to the master table after last refresh	8 min
Create a materialized view log and make sure to purge it every 3 hours	6 min
Create a materialized view and refresh it manually with parallelism degree 6	6 min
Create a materialized view and make sure it is updated every time the master table changes	7 min
Create a materialized view and make sure it is refreshed every 7 minutes. Check the status of the refresh job. Disable auto refresh job.	10 min
Create two tables based on DBA_OBJECTS and DBA_EXTENTS views. Create a query to join them and get the execution plan and the time that is used to run the query. Next, create a materialized view based on the same query and make sure the query uses the materialized view next time it is executed. Get the execution plan and timing.	12 min
Use DBMS_VIEW package to make sure if the query rewrite will be enabled for the specific query.	10 min
Create an encrypted password using AES192 encryption algorithm	10 min
Create a tablespace on the database of first virtual machine and create a table on this tablespace. Move this tablespace to the database of the second virtual machine and query the table.	15 min
Create a tablespace on the database running on Windows OS and create a table on this tablespace. Convert the tablespace to the Linux platform, move the tablespace to the database running on Linux OS and plug it.	20 min
Create two tablespaces on the database running on Linux OS and create a table on this tablespace. Move it to the virtual machine where the Windows OS is running, convert it and plug it to the database.	20 min
Create a comma delimited file with 5 columns, 3 rows as follows: 1,oca,cert,oracle,associate 2,ocp,cert,oracle,professional 3,ocm,cert,oracle,master 5,oce,certoracle,expert	13 min
Create an external table using this flat file and query it. Get the rows that are rejected. Make sure to receive an error if you have any inconsistent row	

User ORACLE_DATAPUMP access driver to create a compressed dump file of the join of DBA_OBJECTS and DBA_SEGMENTS in parallel degree 3. Create an external table and use this dmp file.	10 min
Create 2 tables under USR1 user, export only one of them and import it to the second database under same user	8 min
Create a list partitioned table with 4 partitions under USR1 user based on DBA_OBJECTS view and import only one partition to the second database	10 min
Perform full database export of the GRIDDB, create new database on the second virtual machine and perform full database import. Make sure all objects imported successfully.	14 min
Create a table, procedure and a trigger under USR1 user, export only tables and procedures of this user, import only procedures to the database on the second virtual machine.	10 min
Create 3 views under USR1 table (MV_VIEW1, MV_VIEW2, MYVIEW3) and export views which name start with MV (using INCLUDE parameter)	8 min
Take compressed and uncompressed dump file of user USR1. Compress only data and compare the size of dump files	8 min
Take metadata dump export of user USR1 and import it to the second database under USR2 user.	6 min
Take encrypted dump export of user USR1 using encryption algorithm AES256 and import it to the second database under USR3 user.	6 min
Estimate the size of the dump file of user USR1 and compare it with the result of DBA_EXTENTS view	5 min
Take full dump export of user USR1 and make sure not to have dump file with size greater than 10MB.	6 min
Create a table under USR1 user based on DBA_OBJECTS view, get the current SCN number, remove all rows and commit the transaction. Then export the same table in a parallel mode (degree 2) having all deleted rows in it and import to the second database.	12 min
Create 2 tables under USR1 user based on DBA_SEGMENTS view and export the table using the following conditions: 1) Count of extents greater than 20. 2) Segment type is TABLE. Import the dump file to the second database and check if the export was correct.	12 min
Create a table under USR1 user based on DBA_OBJECTS view, import the dump file to the second database and make sure to update the values of the column OBJECT_NAME to lower case upon the import process.	14 min
Create a table under USR1 user based on DBA_EXTENTS view and import only 30% of the data.	6 min
Create a new tablespace on the GRIDDB and create a new table on this tablespace based on DBA_OBJECTS view. Import the dump file to the second database under different tablespace.	10 min
Import the table that is created in the previous task and make sure to append the rows if the object is already exists in the database. Check the result.	6 min
Create a table under USR1 user. Connect to the second database and create a dump file of the table of USR1 user of the first database.	10 min
Connect to the second database and import the object of USR1 of the first database over the network under USR5 user.	10 min

Enable the trace on the session level and query a table using 4 parallel processes	5 min
Create a table under USR1 user with default 6 parallelism degree. Then query the table and check the explain plan. Then use a hint to disable the parallel query on this table.	8 min
Create an empty table based on DBA_OBJECTS view. Then query DBA_OBJECTS view with parallel degree 4 and insert the result to the table with parallel degree 6.	6 min
Use /tmp/loadfile.dat that was created in the previous task, create a table on the same structure and load the data to the table using SQL*Loader	15 min
Use the data from /tmp/loadfile.dat and load the data to the same table using SQL*Loader without providing a flat file by appending new rows	12 min
Use the /tmp/loadfile.dat and load the data to the new table with ID<3 condition.	10 min
Create a text file with 100 words, create a table with CLOB column type and load it to the table using SQL*Loader	15 min

Day 6 - Data Warehouse Management

Practice	Time
Create following partitioned tables from CLI	
Range partitioned table	10 min
List partitioned table	10 min
Hash partitioned table	10 min
Interval partitioned table (daily, monthly)	15 min
Reference partitioned table	10 min
Create following composite partitioned table from CLI	
Range-hash partitioning	10 min
Range-list partitioning	10 min
Range-range partitioning	10 min
List-range partitioning	10 min
List-hash partitioning	10 min
List-list partitioning	10 min
Hash-hash partitioning	10 min
Interval-list partitioning	10 min
Interval-range partitioning	10 min
Interval-hash partitioning	10 min
Virtual Column based partitioning	10 min
Practice	Duration
Perform following partition maintenance operations	
Adding a new partition to range, hash, list and interval partitioned table	7 min
Add a new partition and subpartition to range-list partitioned table	6 min
Dropping a partition and subpartition from the partitioned table	4 min
Coalescing partition of hash-partitioned table	4 min
Create a list partitioned table and convert it to non-partitioned table	8 min
Use online redefinition feature to convert non-partitioned table to partitioned table	8 min

Create a range partition with 4 partitions and merge 2 of them	6 min
Rename partitions	2 min
Truncate a partition	1 min
Create a new tablespace and move a partition of any table to this tablespace. Check the result	4 min
Create a range-partitioned table with MAXVALUE and split it into 2 different partitions	6 min
Practice	Duration
Create a local partitioned index	2 min
Create a global partitioned index	2 min
Practice	Duration
Create a BasicFile LOB	4 min
Create a SecureFile LOB both from CLI and OEM	8 min
Create 2 tables with compressed and noncompressed LOB segment and show the size difference	10 min
Encrypt a LOB segment	7 min
Create 2 tables with SecureFile LOB segments. Keep duplicates for the first LOB, use deduplicate for the second LOB. Show the size difference	10 min
Create a LOB segment with CACHE and LOGGING options	8 min
Create a BasicFile LOB and convert it to SecureFile LOB	10 min
Practice	Duration
Create a table with two lob segments with the following storage parameters:	8 min
CHUNK 4096	
PCTVERSION 20	
FREEPOOLS 4	
STORAGE INITIAL 50K NEXT 50K PCTINCREASE 10	
NOCACHE	
Load a text file with 10 lines to the table (use example under examples folder at \$ORACLE_HOME)	10 min
Use the following procedures of DBMS_LOB package against the table	
Use DBMS_LOB.INSTR function to find a character inside a LOB segment	8 min
Use DBMS_LOB.SUBSTR function to get a substring from the LOB segment	8 min
Use DBMS_LOB.COMPARE function to compare 2 LOB segments	8 min
Use DBMS_LOB.GETLENGTH function to get the length of LOB	6 min
Practice	Duration
Create a table based on DBA_SEGMENTS view and create a policy that allows user USR_VPD_TEST to select only objects owned by SYSMAN	10 min

Create a policy that doesn't allow user USR_VPD_TEST to update the rows that violates WHERE condition of the policy	10 min
Create a new user USR_VPD_TEST2 and create a column-level VPD and apply the policy to SEGMENT_TYPE column.	12 min
Practice	Duration
Create a table based on DBA_SEGMENTS view and create a policy that allows user USR_VPD_TEST to select only objects owner by SYSMAN	10 min
Create a policy that doesn't allow user USR_VPD_TEST to update the rows that violates WHERE condition of the policy	10 min
Create a new user USR_VPD_TEST2 and create a column-level VPD and apply the policy to SEGMENT_TYPE column.	10 min
Create a user and audit all DML commands of that user. Generate audit records. Get the logged SQL commands.	5 min
Create an object and audit all DML commands for it. Generate audit records	5 min
Audit failed connections to the database and generate audit records	5 min
Create a table based on DBA_SEGMENTS view, use fine-grained auditing feature to audit SELECT and UPDATE commands, TABLESPACE_NAME and EXTENTS columns and OWNER=SYSMAN condition. Run commands to generate different audit logs	10 min
Practice	Duration
Create two contexts and use SYS_CONTEXT function to get the hostname and instance name and set them to an attribute of both contexts	5 min
<p>Implement the following fine-grained auditing task:</p> <p> Create three users USR1, USR2 and USR_MAIN</p> <p> Create two tables under USR_MAIN</p> <p> USR_LISTS with 2 columns (ID and NAME)</p> <p> USR_COURSE with 3 columns (ID, USER_ID and NAME)</p> <p> Insert two rows into USR_LISTS table</p> <p> 1 USR1</p> <p> 2 USR2</p> <p> Insert 2 rows into USR_COURSE table</p> <p> 1 1 Oracle</p> <p> 2 2 PL/SQL</p> <p> Create a context using a package with a procedure which gets the username of connected session, finds the id of this user from USR_LISTS table and set the returned value to the attribute of the context</p> <p> Create a package (function) for the fine-grained auditing policy to generate a predicate on USR_COURSE table to return rows where the value of ID column is equal to the value of the context</p> <p> Create a fine-grained auditing policy on USR_COURSE table for statement type 'SELECT' and use the function created above.</p>	15 min

Connect with USR1 user and query the USR_COURSE table. Check if you got 'Oracle' as an output. Connect with USR2 user and query the same table. Make sure you got PL/SQL as an output.	
Flashback Data Archive - Perform the scenario that's is provided in Flashback Database section of the chapter	15 min
Use all flashback archive parameters (retention, quota, purge historical data, drop, disable the flashback data archive and etc)	10 min
Query flashback archive views	5 min

Day 7 - Performance Management

Practice	Time
<p>Create 2 plans, 2 consumer groups and 4 users. Make sure the first plan is active on the midday (7am-12am) and the second plan is active during the night (12a-7am). Assign the first two users to the first consumer group, and the remaining two users to the second consumer group.</p> <p>Edit the first plan as follows:</p> <ul style="list-style-type: none"> - Make sure the first consumer group has 80% of CPU allocation when the first plan is active, and second consumer group has 10% of CPU allocation. - Set parallelism degree to 8 for the first consumer group, 2 for the second one. - Set maximum number of active sessions to 20 for the first consumer group, 5 for the second consumer group. - Make the first consumer group to use unlimited undo space and limit the undo usage of the second consumer group to 10mb. - Set execution time limit to 600 seconds for the first consumer group, 5 seconds for the second consumer group and make sure the session will be killed when the limit exceeds for the second consumer group. - Set idle time 600 seconds for the first consumer group, 60 seconds for the second consumer group. <p>Then activate the first plan, connect with the user of the second consumer group and check the following cases:</p> <ul style="list-style-type: none"> - Run a query with parallel degree greater than 2 and check the result - Open 5 sessions and see what happens when you create 6th session. - Create a table with 1mln rows (based on DBA_OBJECTS view) and update OBJECT_NAME column and see what happens when the session generates more undo - Query the previous table with "ORDER BY 1,2,3,4,5,6" limit and make it to execute more than 5 seconds and see what happens. - Create a session and stay idle for 2 minutes and see what will happen. 	30 minutes
Enable the result cache, check its status	2 min
Create a table, enable the result cache on the session level, query the table and put the result to the memory. Display the cached objects and attributes. Run the same statement without getting data from the result cache (use hint to disable the result cache).	10 min
Create a table with result cache option enabled, query it and check result cache performance views to see if the results come from the memory.	10 min
Create a function with RESULT_CACHE option to store its results to the memory. Make sure it works	15 min

Create a table based on DBA_OBJECTS view, gather the statistics, query the table and filter the row using two columns (OBJECT_NAME and OBJECT_TYPE), get the execution plan and check the estimated row count. Next, create extended statistics for both columns and check the estimated row count.	15 min
Flush the shared pool, create a new table based on DBA_OBJECTS and gather table statistics. Query the table twice and get count of executions and invalidations from V\$SQL view. Then gather statistics again and invalidate the cached cursors and query V\$SQL view again.	15 min
Create a table and run a query to have a Full Table Scan access method	7 min
Create an index on the previous table and run a query to have an Index Unique Scan access method	7 min
Use the same index and run a query to have an Index Range Scan access method	3 min
Create a composite index on the previous table and run a query to have an Index Skip Scan	8 min
Create an index on the previous table and run a query to have an Index Fast Full Scan	3 min
Generate an explain plan of any statement using EXPLAIN PLAN FOR command	4 min
Generate an explain plan of any statement by having only SQL_ID	6 min
Format the execution plan and make sure to have cost and rows information in the output using DBMS_XPLAN package	7 min
Use SET AUTOTRACE command and display <ul style="list-style-type: none"> - only execution plan - only statistics - both execution plan and statistics - execution plan with output of query 	6 min
Modify AWR Snapshot Retention to 14 days, modify AWR Snapshot interval to 30 minutes	4 min
Choose two different snapshot and create an AWR report. Investigate the Top wait events	5 min
Generate the same report from CLI.	4 min
Compare two different periods and generate comparison report	6 min
Run Automatic Database Diagnostic Management and check the findings.	5 min
Create an ADDM task for two snapshots (using DBMS_ADDM package) and investigate the report	8 min
Run an ASH report from both OEM and CLI and investigate the output	8 min
Get the memory size advice with OEM and performance views for <ul style="list-style-type: none"> - MEMORY_TARGET parameter - SGA and PGA 	15 min

- shared pool - buffer cache	
Create a new session, query DBA_OBJECTS view. Trace this query from another session	8 min
Create SQL Tuning Set and store three different SQLs from the cursor. Use SQL Tuning Advisor to tune SQL statements of the STS	10 min
Load any SQL statement from the cursor to SQL Tuning Advisor and get recommendations	8 min
Load SQL statements from AWR snapshot to SQL Tuning Advisor and get recommendations	8 min
Get the list of top active sessions in OEM and schedule SQL Tuning Advisor Job	8 min
Export SQL Tuning Set to another database and import it	15 min
Create two tables based on DBA_OBJECTS and DBA_EXTENTS, run SQL to get count of extents for each object. Create an SQL Tuning Set using this SQL statement and schedule an SQL Access Advisor using SQLACCESS_WAREHOUSE template and get recommendations for Indexes, Materialized Views and Partitioning access structures for the SQL Tuning Set that was created. Review the recommendation and implement it.	20 min
Create a table based on DBA_OBJECTS view, run two different SQL statement. In the first statement, retrieve rows WHERE OBJECT_NAME='TEST'. In the second query, retrieve rows WHERE UPPER(OWNER)='TEST'. Next, create SQL Tuning Set based on this statements and see what performance benefits you get if you create an index for OBJECT_NAME column and function-based index on OWNER column using SQL Performance Analyzer. Investigate the comparison report.	15 min
Create a baseline and compare it with any pair of snapshots	8 min
Enable automatic capture of SQL plans, create a table based on DBA_OBJECTS view and query the table with condition "WHERE OBJECT_NAME='TEST'". Print the execution plan, check if baseline was created both from OEM and performance view. Next, create an index on OBJECT_NAME column of the same table, run the same query again and print the execution plan. Check if the new baseline is created, evolve it and accept the plan, run the same query again and check the execution plan again. Make sure it is changed.	20 min

Day 8 - Grid Infrastructure and ASM

Practice	Time
Create two virtual machines and install Oracle Grid Infrastructure	There is no timing for this task
Configure grid and database environment variables	10 min
Create external, normal and high redundancy disk groups	10 min
Add new disk to the diskgroup	3 min
Drop a disk from the diskgroup, undrop it	5 min
Run the manual rebalance operation and check V\$ASM_OPERATION view	5 min
Dismount and mount diskgroup, check the status	3 min
Add and drop a disk from the diskgroup using single command	4 min
Create a diskgroup using ASMCA utility	5 min
Create an ACFS file system using ASMCA utility	10 min
Create an ACFS file system manually	15 min
Deregister and Dismount an ACFS file system	7 min
Use crsctl to start, stop the clusterware, disable and enable OHA	5 min
Use srvctl to start and stop database, stop the first instance, disable the second instance, stop the ASM instance of the first node and stop the scan listener	10 min
Create a new diskgroup and use it to add a mirror copy of OCR file	5 min
Get backup location of OCR file, delete current OCR file and restore it from backup	15 min
Remove mirror copy of OCR file, move the OCR file from main storage to the new diskgroup	10 min
Change default location of auto backup of OCR file	5 min
Run the Cluster Verify Utility before and post database installation	6 min
Run the Cluster Verify Utility to make connectivity, user, permission, ASM, OCR and Votedisk verifications	10 min

Day 9 - Real Application Clusters

Practice	Time
Install a RAC database on 2 node cluster using DBCA	15 min
Create and configure new diskgroup using asmca utility in a silent mode	10 min
Install a RAC database in a silent mode	10 min
Enable archivelog mode of the RAC database	5 min
Create a new service, add a new TNS entry that uses the new service, connect to the database with using a new TNS entry, check the instance name, shutdown the connected instance from the different terminal and check the connection again. The session should failover to the second instance.	15 min
Create a new pool using MIN_SIZE, MAX_SIZE and IMPORTANCE parameters. Create a child pool based on the first pool	8 min

Day 10 - Data Guard

Practice	Time
Remove all standby configurations and create standby database using SQL*Plus	25 min
Remove all standby configurations and create standby database using RMAN	25 min
Remove all standby configurations and create standby database using OEM	25 min
Change the data protection mode to maximum availability using SQL*Plus	5 min
Change the data protection mode to maximum protection using OEM	5 min
Stop the archive log apply process and remove archived log files from the standby database that were not applied. Create a table on the primary database, take an incremental backup and recover it on standby database. Query the table on the standby database.	15 min
Create a table on the primary database, perform a switchover using SQL*Plus, create another table on the new primary database, switch database back and check both tables.	20 min
Create a table on the primary database, perform a switchover using DGMGRL, create another table on the new primary database, switch database back and check both tables.	20 min
Create a table on the primary database, perform a switchover using OEM, create another table on the new primary database, switch database back and check both tables.	20 min
Perform failover using SQL*Plus	5 min
Perform failover using DGMGRL	5 min
Perform failover using OEM	5 min
Enable fast start failover, create a database service and create a trigger to start it when the database acts like a primary database. Then connect to the primary db using the database service and run a session. When the session is running, shutdown the primary database and make sure the automatic switchover is performed. Check the status of the first session. At the end, perform a switchover back to the previous primary database.	25 min
Convert physical standby database to snapshot standby database using DGMGRL	15 min
Convert physical standby database to snapshot standby database using SQL*Plus	15 min
Configure archivelog deletion policy and make sure they are eligible for the deletion when they are applied to the standby database. Stop the apply process on the standby database, switch log files on the primary database and make sure they are not applied to the standby database. Try to delete all archived log files on the primary database and make sure they are not deleted (because they are not applied)	10 min