Study Guide for Oracle Certified Master 11g Exam

Kamran Aghayev A.
Oracle Certified Master
ACE Director
Dedication

First of all, I would like to thank God, the Almighty, for giving me the knowledge, vision, and ability to proceed and for making everything possible by giving me strength and courage to do this work. It is only through His grace that achievement can truly be accomplished.

I would also like to thank to my friend Garib Mehdiyev, CTO at wetravel.com for helping and motivating me to finish this book.

I dedicate this book to my beloved country Azerbaijan, to my lovely family and especially to my father for his unconditional love and support throughout my life and my career. All the support he has provided me over the years was the greatest gift anyone has ever given me. It is through him that I first grasped the importance of learning.
Kamran Aghayev A. is an Oracle Certified Master, Oracle Clusterware Certified Expert and RAC Implement Specialist, Oracle Certified Professional (9i/10g/11g) and Oracle ACE Director working as a DBA team head at AzerCell Telecom LLC.

He's author of the books “Oracle Backup and Recovery: Expert secrets for using RMAN and Data Pump” and “Oracle Certified Master 11G Exam Guide”

He runs a popular blog kamranagayev.com where he shares his experience and contributes fairly regularly to newsgroups, forums, and user group meetings and events around the world.

He is a frequent speaker and presented in many countries, most recently at the USA, Japan, Thailand, China, India, Argentina, Uruguay, Panama, Costa Rica, Mexico, Guatemala, Finland and Turkey.

He is a President of Azerbaijan Oracle User Group (AzerOUG).
About the Technical Reviewers

Syed Jaffer Hussain, Technical Director at eProseed KSA is an Oracle ACE Director in the area of Database Management & Performance since 2011. With over 20+ years of experience as a principal consultant, architect and trainer, Jaffer is recognized globally as one of the authorities in his area of expertise. For his immense technical skills and contribution to the Oracle community, he was named as the ‘Best DBA of the year 2011’ by Oracle Head Quarters. He was the first Oracle DBA to achieve Oracle Certified Master (OCM) credentials in Saudi Arabia and only Oracle ACE Director in the region.

Over the past 18 years or so, he has been involved in many local and international banks/customers where he successfully implemented and managed highly complex cluster and non-cluster setups with hundreds of business-critical databases. Involved and successfully implemented Exadata, Super Cluster and Oracle Data Guard technologies for business critical databases. His broad knowledge and high level skills in the areas of Performance Tuning, Architecture, High Availability, Disaster Recovery, Backup and recovery strategies and Engineered systems indeed helped to achieve the great success to the customers and the companies he associated.

He also write articles for magazines, web sites and also manages an Oracle blog (jaffardba.blogspot.com) where he shares his knowledge and problem solving tips. Jaffer is a frequent presenter at international conferences such as All India Oracle User Group (AIOUG), Oracle Open World, Saudi Oracle User Group, Turkey Oracle User Group etc. Jaffar has co-authored four (04) Oracle books, Oracle 11g R1/R2 Real Application Cluster Handbook, Expert Oracle RAC 12c, Oracle Exadata Expert’s Handbook and Oracle Problem Solving and Troubleshooting Handbook.

Joel is an Expert DBA (Oracle ACE Director, Maximum Availability OCM, OCM Cloud Admin. & OCM12c/11g) with over 17 years of Real World Experience in Oracle Technology, specialized in design and implement solutions of: High Availability, Disaster Recovery, Upgrades, Replication, Cloud and all area related to Oracle Databases. International consultant with duties, conferences & activities in more than 50 countries in 5 continents and countless clients around the world. Habitual and one of leading writers of Technical Oracle articles for: OTN Spanish & OTN Portuguese. Regular Speaker in worldwide Oracle events like: OTN LAD (Latin America), OTN MENA (Middle East & Africa), OTN APAC (Asian Pacific), DTCC China and more.

Joel has always been known for being a pioneer in Oracle technology since the early days of his career being the first Latin American awarded as “OTN Expert” at year 2003 by Oracle Corp., one of the first “Oracle ACE” globally in the Oracle ACE Program at year 2004 and as one of the biggest professional achievement in his career he was honoured as one of the first “OCM Database Cloud Administrator” & Maximum Availability OCM in the world. Currently Joel works as Chief Technologist in “Yunhe Enmo (Beijing) Technology Co.,Ltd”., company located in Beijing, China.
Contents

Introduction ................................................................. xvii
  About the exam ....................................................... xvii
  Techniques used by me to prepare for the OCM exam .......... xviii
  Create OCM environment on your own laptop ............... xx
  Install and configure virtual machine .......................... xxi
  Install Oracle Enterprise Linux ................................ xxii
  Installing Oracle Enterprise Linux 5.4 .......................... xxii
  Installing Oracle Enterprise Linux 6.4 .......................... xxvii

Chapter 1. Server Configuration ........................................ 1
  Install Oracle Software ............................................. 4
    Installing Oracle Software with GUI ......................... 4
    Installing Oracle Software in a silent mode .............. 10
  1.1. Create the database ........................................... 11
    Create database using DBCA .................................. 12
    Create database manually using CREATE DATABASE
      command ......................................................... 27
    Create a database using OMF (Oracle Managed Files) ... 29
  1.2. Determine and set sizing parameters for database
      structures ......................................................... 31
  1.3. Create and manage temporary, permanent, and undo
      tablespaces ....................................................... 34
      Create Permanent tablespace ............................... 34
Create Compressed Tablespace .......................... 36
Create Encrypted Tablespace .......................... 36
Create Temporary Tablespace .......................... 38
Create temporary tablespace group ...................... 39
Specify Nonstandard block size for tablespace ........ 40
Modify Tablespace (OFFLINE, ONLINE, READ-ONLY,
ALTER TABLESPACE, SHRINK, RENAME, DROP) ..... 40
Managing UNDO tablespace ............................. 41
1.4. Stripe data files across multiple physical devices
and locations ............................................. 43
    Stripe redo log files across multiple physical devices .... 43
    Relocate controlfile, redo log file and datafiles to the
different folder after database created ................ 44
    Moving redo log files ................................ 44
    Moving controlfile .................................... 46
    Moving datafiles ..................................... 47
1.5. Configure the database environment to support
optimal data access performance ........................ 48
1.6. Create and manage database configuration files ........ 49
1.7. Create and manage bigfile tablespaces ................ 49
1.8. Create and Manage a tablespace that uses
    NFS mounted file system file ........................ 51
1.9. Create and manage multiple network configuration files ........ 54
    Create network configuration files with netca (GUI) .... 54
    Create listener with different port .................... 56
    Create tnsnames.ora file .............................. 57
1.10. Create and configure a listener ...................... 61
1.11. Configure the database instance to support shared
server connections ..................................... 62
1.12. Set up network tracing ............................... 63
1.13. Manage Oracle network processes ................... 64
1.14. Configure the network environment to allow connections
to multiple databases ................................... 65
# Contents

1.15. Use configurationless connections ........................................... 66  
1.16. Use OPatch to install a patch .............................................. 67  
1.17. Use Grid Infrastructure to manage oracle databases and other resources ............................................. 70  
1.18. Use Enterprise Manager Configuration Assistant (EMCA) utility ......................................................... 79  
Conclusion ......................................................................................... 84

## Chapter 2. Enterprise Manager Grid Control ................................. 85  
2.1. Install and Patch Enterprise Manager Grid Control software ........ 88  
   Downloading necessary softwares .............................................. 88  
   Installing Oracle Database 11g Software .................................. 90  
   Installing WebLogic 10.3.2 .......................................................... 90  
   Installing Oracle Enterprise Manager Grid Control (11.0.1.0.1) .......... 97  
   Applying patch to OEM Grid Control ........................................ 116  
2.2. Configure the Enterprise Manager repository ............................ 121  
2.3. Create Enterprise Manager Grid Control users ......................... 122  
2.4. Use Enterprise Manager to modify a database configuration ........ 129  
   Memory Advisors ................................................................. 134  
   Automatic Undo Management ................................................. 135  
   Initialization Parameters .......................................................... 136  
   View Database Feature Usage ................................................. 136  
2.5. Configure Enterprise Manager to modify database availability ..... 137  
2.6. Create and manage jobs ............................................................. 139  
2.7. Create and monitor alerts .......................................................... 145  
2.8. Create notifications ................................................................ 149  
2.9. Implement Grid Control and Database Control ....................... 152  
2.10. Choose the appropriate tablespace type for the intended use .... 153  
2.11. Create Scheduler jobs ............................................................. 153  
2.12. Create schedules ................................................................. 158  
2.13. Assign jobs to windows .......................................................... 160  
2.14. Create programs ................................................................. 163
Chapter 3. Managing Database Availability ............................................. 181
  3.1. Maintain recovery catalogs ......................................................... 184
      Creating a Recovery Catalog ....................................................... 184
      Creating and managing Virtual Private Catalogs .......................... 185
      Managing stored scripts ............................................................ 188
      Resynchronizing recovery catalog .............................................. 189
      Upgrading the Recovery Catalog: .............................................. 190
      Import and Move the recovery catalog ...................................... 190
  3.2. Configure Recovery Manager .................................................... 192
      Changing retention policy ......................................................... 193
      Using archival backups .......................................................... 194
      Change backup destination ....................................................... 195
      Configure autobackup controlfile ............................................ 197
      Configure backup encryption .................................................... 197
      Configuring Backup Compression .............................................. 198
      Configure archive log deletion policy ...................................... 199
      Configuring maximum size of backup ...................................... 199
      Configure backup duplexing .................................................... 200
      Exclude tablespace ............................................................... 200
  3.3. Use Recovery Manager to perform database backups ...................... 201
      Backup the whole database ..................................................... 202
      Backing up tablespace and datafiles ....................................... 203
      Backing up controlfile and spfile ........................................... 203
      Backing up ARCHIVELOG files ................................................ 203
Taking incremental backups ........................................... 204
Incrementally updated backups ....................................... 204
Taking Multisectional Backups ........................................ 206

3.4. Use Recover Manager to perform complete database restore and recovery operations .......................... 207
    Restore parameter file ............................................. 208
    Recover from loss of controlfile without RMAN ............. 209
    Restore controlfile with RMAN .................................. 212
    Restore datafile using RMAN ....................................... 213
    Recovering datafile without having any backup ............. 214
    Using SET NEWNAME command to change location of the restore datafile ........................................ 216
    Restore/recover a tablespace ..................................... 217
    Restoring archived redo log files ............................... 218
    Perform Block media recovery .................................... 218
    Performing disaster recovery .................................... 220
    Perform incomplete recovery (Time, SCN and Change based) ........ 224
    Recover database to the previous incarnation .............. 226
    Performing tablespace point in time recovery (TSPITR) .... 228

3.5. Configure RMAN ..................................................... 235

3.6. Create different types of RMAN backups to cater for different performance and retention requirements ........................ 235
    Monitor RMAN backup/recover progress ........................ 235
    Identify source of backup or restore bottlenecks ............ 237

3.7. Set Flashback Database parameters ............................. 239
    Creating restore point ............................................. 239
    Enable Flashback database ....................................... 240

3.8. Configure a Fast Recovery Area .................................. 247

3.9. Perform various recovery operations using Flashback technology ......................................................... 250
    Flashback Query ................................................... 250
    Flashback Version Query .......................................... 252
    Flashback Transaction Query .................................... 254
Chapter 4. Data Management

4.1. Manage Materialized Views to improve rewrite and refresh performance
- Materialized view logs
- Materialized view query rewrite
4.2. Configure and manage distributed materialized views
4.3. Create and Manage encrypted tablespaces
4.4. Manage Transport of tablespaces across platforms
4.5. Configure a schema to support a star transformation query
4.6. Administer external tables
4.7. Implement Data Pump export and import jobs for data transfer
   - Performing a table mode export (TABLES parameter)
   - Performing a schema mode export (SCHEMAS parameter)
   - Filtering export of database objects
     - (EXCLUDE and INCLUDE parameters)
   - Performing compressed export (COMPRESSION)
   - Specify type of content to export (CONTENT)
   - Perform encrypted export (ENCRYPTION, ENCRYPTION_ALGORITHM, ENCRYPTION_MODE, ENCRYPTION_PASSWORD)
   - Estimate the disk space of the export process
     - (ESTIMATE, ESTIMATE_ONLY)
   - Specify the maximum file size of the dump file (FILESIZE)
   - Perform an export to the specific time or SCN
     - (FLASHBACK_TIME, FLASHBACK_SCN)
   - Performing parallel export (PARALLEL)
   - Specify a query to filter the data that is being exported (QUERY)
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply specific function to each row that is being exported</td>
<td>336</td>
</tr>
<tr>
<td>(REMAP_DATA)</td>
<td></td>
</tr>
<tr>
<td>Specify if the dump file should be reused if the same command is run</td>
<td>339</td>
</tr>
<tr>
<td>again (REUSE_DUMPFILES)</td>
<td></td>
</tr>
<tr>
<td>Export specific percentage of data (SAMPLE)</td>
<td>339</td>
</tr>
<tr>
<td>Load objects under different user (REMAP_SCHEMA)</td>
<td>340</td>
</tr>
<tr>
<td>Rename table during import (REMAP_TABLE)</td>
<td>340</td>
</tr>
<tr>
<td>Import objects to the different tablespace</td>
<td>341</td>
</tr>
<tr>
<td>(REMAP_TABLESPACE)</td>
<td></td>
</tr>
<tr>
<td>Skipping indexes with unusable state</td>
<td>341</td>
</tr>
<tr>
<td>(SKIP_UNUSABLE_INDEXES)</td>
<td></td>
</tr>
<tr>
<td>Write all DDL commands to the file (SQLFILE)</td>
<td>342</td>
</tr>
<tr>
<td>Specify what should happen if the table exists during the export</td>
<td>342</td>
</tr>
<tr>
<td>process (TABLE_EXISTS_ACTION)</td>
<td></td>
</tr>
<tr>
<td>Data Pump interactive-command mode</td>
<td>346</td>
</tr>
<tr>
<td>4.8. Implement Data Pump to and from remote databases</td>
<td>347</td>
</tr>
<tr>
<td>4.9. Configure and use parallel execution for queries</td>
<td>350</td>
</tr>
<tr>
<td>Using hints related with parallelism</td>
<td>350</td>
</tr>
<tr>
<td>Enabling parallelism during creation of an object</td>
<td>352</td>
</tr>
<tr>
<td>4.10. Use SQL*Loader</td>
<td>355</td>
</tr>
<tr>
<td>Load a data without providing a flat file</td>
<td>358</td>
</tr>
<tr>
<td>Load a data to the non-empty table</td>
<td>359</td>
</tr>
<tr>
<td>Truncate table before the load</td>
<td>360</td>
</tr>
<tr>
<td>Load data with fixed format</td>
<td>360</td>
</tr>
<tr>
<td>Load data from multiple files</td>
<td>363</td>
</tr>
<tr>
<td>Load data to multiple tables</td>
<td>363</td>
</tr>
<tr>
<td>Using WHERE clause with SQL*Loader</td>
<td>365</td>
</tr>
<tr>
<td>Methods of data loading</td>
<td>366</td>
</tr>
<tr>
<td>Loading LOB data</td>
<td>367</td>
</tr>
<tr>
<td>4.11. Administer, manage and tune parallel execution</td>
<td>368</td>
</tr>
<tr>
<td>Managing parallelism degree</td>
<td>368</td>
</tr>
<tr>
<td>Monitoring parallel execution</td>
<td>370</td>
</tr>
<tr>
<td>Conclusion</td>
<td>378</td>
</tr>
</tbody>
</table>
Chapter 5. Data Warehouse Management ............................................. 379

5.1. Administer partitioned tables and indexes using appropriate methods and keys ......................................................... 382
  Creating Range partitioned table ............................................. 382
  Creating multi column range partition ...................................... 387
  Creating List partitioned table .................................................. 388
  Creating Hash partitioned table .............................................. 391
  Creating Interval partitioned table .......................................... 394
  Creating Reference partitioned table ....................................... 399
  Creating System partitioned table ......................................... 400
  Creating Composite partitioned table ..................................... 402
    Range-Hash partitioning ..................................................... 403
    Range-list partitioning ..................................................... 405
    Range-range partitioning .................................................. 406
    List-range partitioning .................................................... 407
    List-hash partitioning ..................................................... 408
    List-list partitioning ....................................................... 409
    Interval-list partitioning .................................................. 410
    Interval-range partitioning ............................................... 412
    Interval-hash partitioning ............................................... 412
  Creating Virtual Column based partitioning ............................ 413

5.2. Perform partition maintenance operations ............................ 415
  Adding a new partition .......................................................... 416
    Adding new partition to range-partitioned table ..................... 416
    Adding new partition to hash-partitioned table ...................... 416
    Adding new partition to list-partitioned table ...................... 417
    Adding new partition to Interval-partitioned table ................. 417
    Adding partition and subpartition to composite-*hash partitioned tables ......................................................... 418
    Adding partition and subpartition to composite-*list partitioned tables ......................................................... 418
    Adding partition and subpartition to composite-*range partitioned tables ......................................................... 419
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropping a partition</td>
<td>420</td>
</tr>
<tr>
<td>Coalescing partitions</td>
<td>420</td>
</tr>
<tr>
<td>Exchange partitions</td>
<td>422</td>
</tr>
<tr>
<td>Convert a partition to a non-partitioned table</td>
<td>423</td>
</tr>
<tr>
<td>Convert a subpartition to a non-partitioned table</td>
<td>424</td>
</tr>
<tr>
<td>Convert non-partitioned table to a partition (subpartition) of</td>
<td>425</td>
</tr>
<tr>
<td>partitioned-table</td>
<td></td>
</tr>
<tr>
<td>Exchange partition (subpartition) of partitioned table with</td>
<td>428</td>
</tr>
<tr>
<td>partition (subpartition) of composite-partitioned table and vice</td>
<td></td>
</tr>
<tr>
<td>versa</td>
<td></td>
</tr>
<tr>
<td>Merging partitions</td>
<td>430</td>
</tr>
<tr>
<td>Renaming partitions</td>
<td>432</td>
</tr>
<tr>
<td>Moving partitions</td>
<td>432</td>
</tr>
<tr>
<td>Splitting partitions</td>
<td>433</td>
</tr>
<tr>
<td>Truncating partitions</td>
<td>435</td>
</tr>
<tr>
<td>5.3. Maintain indexes on a partitioned table</td>
<td>436</td>
</tr>
<tr>
<td>Local partitioned index</td>
<td>437</td>
</tr>
<tr>
<td>Global partitioned index</td>
<td>438</td>
</tr>
<tr>
<td>Rebuilding Index Partitions</td>
<td>439</td>
</tr>
<tr>
<td>5.4. Implement SecureFile LOB</td>
<td>440</td>
</tr>
<tr>
<td>Compression</td>
<td>442</td>
</tr>
<tr>
<td>Encryption</td>
<td>444</td>
</tr>
<tr>
<td>Deduplication</td>
<td>445</td>
</tr>
<tr>
<td>Caching</td>
<td>447</td>
</tr>
<tr>
<td>Logging</td>
<td>447</td>
</tr>
<tr>
<td>Migration of LOBs to SecureFiles</td>
<td>448</td>
</tr>
<tr>
<td>Create SecureFiles LOB using OEM</td>
<td>451</td>
</tr>
<tr>
<td>5.5. Create and manage LOB segments</td>
<td>453</td>
</tr>
<tr>
<td>Modify LOB parameters</td>
<td>456</td>
</tr>
<tr>
<td>Inserting data into LOB column</td>
<td>457</td>
</tr>
<tr>
<td>Using DBMS_LOB package subprograms</td>
<td>458</td>
</tr>
<tr>
<td>DBMS_LOB.INSTR</td>
<td>461</td>
</tr>
<tr>
<td>DBMS_LOB.SUBSTR</td>
<td>462</td>
</tr>
</tbody>
</table>
Study Guide for Oracle Certified Master 11g Exam

DBMS_LOB.COMPARE .................................................. 463
DBMS_LOB.GETLENGTH ............................................. 464
DBMS_LOB.LOADFROMFILE ......................................... 464
DBMS_LOB.OPEN and DBMS_LOB.FILECLOSEALL procedures .......................... 464
DBMS_LOB.WRITE and WRITEAPPEND procedures ......................... 464
DBMS_LOB.CREATETEMPORARY, FREETEMPORARY, ISTEMPORARY .................. 464

5.6. Implement fine-grained access control .................................. 466
    update_check parameter ........................................ 470
    Enable column-level VPD ......................................... 472

5.7. Oracle Database Auditing ............................................. 475
    Audit actions performed by a user ............................... 476
    Audit SQL statements on specific table .................................. 477
    Audit Statement executions ........................................ 478
    Fine-Grained Auditing ............................................. 479

5.8. Create and manage contexts ........................................... 486

5.9. Administer flashback data archive and schema evolution ............. 496
    Managing Flashback Data Archive .................................... 499

Conclusion ................................................................. 506

Chapter 6. Performance Management ....................................... 506

6.1. Administer Resource Manager ........................................ 510
6.2. Use Result Cache ................................................... 520
6.3. Use multi column statistics .......................................... 537
6.4. Gather statistics on a specific table without invalidating cursors .................. 545
6.5. Use partitioned indexes ............................................. 547
6.6. Administer and tune schema object to support various access methods ........... 548
6.7. Interpret execution plan ............................................. 555
6.8. Use SQL tuning tools and features ................................... 564
6.9. Use SQL Tuning Advisor ........................................... 596
6.10. Use SQL Access Advisor ........................................ 608
6.11. Use SQL Performance Analyzer ................................... 618
## Contents

6.12. Configure baseline templates ........................................ 625
6.13. Use SQL Plan Management feature .................................. 630
Conclusion ................................................................. 652

### Chapter 7. Grid Infrastructure and ASM .............................. 653

7.1. Install Oracle Grid Infrastructure ..................................... 656
7.2. Create ASM Disk Groups .............................................. 670
  - Configure environment variables for Database and Grid connections ........................................ 671
  - ASM Diskgroup Redundancy ........................................... 672
  - Add new disk to the diskgroup ....................................... 677
7.3. Create and manage as ASM instance ................................ 682
7.4. Implement ASM failure groups ....................................... 682
7.5. Creating ACFS File System ........................................... 682
  - Resize ACFS file system ............................................... 688
  - Creating ACFS Snapshots ............................................ 688
  - Deregistering and Dismounting ACFS file system .............. 689
7.6. Start, stop, configure and administer Oracle Grid Infrastructure .............................................................................. 690
  - Oracle Interface Configuration Tool (OIFCFG) .................. 694
  - OCR Configuration Tool (OCRCONFIG) .............................. 696
  - Cluster Verification Utility (clufvy) ................................. 697
Conclusion ........................................................................... 702

### Chapter 8. Real Application Clusters .................................. 703

8.1. Install the Oracle Database 11gR2 software ....................... 706
8.2. Configure ASM for the shared disks and create a clustered database .......................................................... 708
  - Creating RAC database using Database Configuration Assistant (GUI) ........................................... 709
  - Creating a RAC database in a silent mode .......................... 712
    - Configure ASM in a silent mode ................................. 713
    - Create a RAC database in silent mode with dbca utility ... 715
8.3. Configure archiving ................................................. 717
8.4. Configure services using both Manual and Policy
    Managed methods ................................................. 719
    Configure service using manual method ....................... 719
    Create a service for policy-managed database .................. 727
Conclusion ................................................................. 730

Chapter 9. Data Guard .................................................. 731
9.1. Create Physical Standby Database with real-time apply .......... 734
    Creating Standby database manually using SQL*Plus ........... 735
    Creating Standby database using RMAN .......................... 743
    Creating Standby database using Oracle Enterprise Manager .... 747
    Data Guard Protection Modes ..................................... 754
9.2. Configure the data guard environment to reduce overheads
    of fast incremental backups on the primary database .......... 758
9.3. Configure the Observer ............................................. 763
9.4. Switchover and switch back ....................................... 766
    Perform role transition using DGMGRL utility .................. 772
    Performing role transitions using OEM .......................... 773
9.5. Configure connect time failover ................................... 775
9.6. Convert the standby to a snapshot standby ...................... 787
    Converting physical standby database to snapshot
    standby database using DGMGRL utility ....................... 787
    Converting physical standby database to snapshot
    standby database using SQL*Plus .............................. 793
9.7. Configure archivelog deletion policy for the Dataguard
    configuration ......................................................... 794
Conclusion ................................................................. 798
To become an Oracle Certified Master (OCM) has been my dream for years. It is very challenging and exciting to participate in such a practical exam and try your skills. In addition, it is much more exciting to successfully pass the exam and become an Oracle Certified Master.

Once I suffered from the lack of OCM related articles, records or books and it was hard for me to collect all materials from different sources and create my own practical cases. Nevertheless, I coped with this problem and by making a good progress in both exams I made sure that all my preparation plan was on track. Therefore, I decided to author a book and share it with you. I do believe that this book will contribute to your exam preparation and encourage you to take the exam.

If you are an Oracle DBA and have a good experience in a production environment, you are encouraged to take this exam.

About the exam

Oracle Certified Master exam is believed to be one of the most difficult practical exams among IT community. Despite the fact that every year tens of thousands of DBAs pass Oracle Certified Professional (OCP) exam, only hundreds of DBA will take OCM exam.
Unlike OCP exam, during an OCM exam a candidate must successfully finish two-day hands-on tasks. Therefore, based on recommendations made by Instructors at an Oracle University, a candidate is required to have minimum 3-4 years experience before taking an OCM exam. Being an OCP and taking two Oracle University classes are also pre-requisites of this exam. No access to internet is allowed during the exam (no google, no forums and etc.). The only thing you are provided is a documentaition without a search functionality. You should also make any type of configuration both with Graphical User Interface (GUI) and Command Line Interface (CLI).

Each candidate who will attend the exam should sign an NDA contract to not give more information about the exam, and share an exam questions. In this book the information about the exam is limited to those posted at the web page of the OCM exam. You can check the following link to get more details about the exam from Oracle:


Alternatively, please visit oracle.com and go to Training menu. Select Certification menu and get the list of all exams from Exams menu. Click on “Oracle Database 11g Certified Master Exam | 11GOCM” link and open the exam page.

Moreover, all the tasks covered by me in this book are not those that I got during the exam. The former are the tasks that I created on my own during the preparation for the exam.

Techniques used by me to prepare for the OCM exam

As I had better experience with Oracle 10g, rather than Oracle 11g, I decided to take 10g OCM exam, and then upgrade it to 11g OCM later. It took me 9 months to prepare for 10g OCM, and 4 months for 11g OCM upgrade exam. Due to my excessive experience in this special domain just one month per section was enough for me to get prepared. I made a detailed study of all subjects related to the exam, and meanwhile tried to get some extra time during work hours and launch, and sometimes would stay for additional hours after work and over the weekends.

If you have a few years' professional experience with Oracle, you can easily succeed in OCM exam using preparation tasks that I have covered in this book.

I followed the following tips before taking both OCM exams:

- First, I created a Word file and wrote down all OCM subjects by sections and started studying each topic individually. For each topic, I went through the documentation, obtained all available blog posts, checked all metalink notes, made a research only on that specific topic by turning the pages of available books I have got, and took notes of everything in that file. Finally I came up with a very long list of different sources related to each subject of the exam. You can find those sources in the Reference part under each topic. Before attending the exam, you should be able to find any subject
from the documentation among different books without using search option. During the preparation, you recommended not to use search option of the documentation at ALL! Moreover, I would strongly suggest you not to use any third part utility to manage the database. Make sure you use only SQL*Plus and Enterprise Manager (EM) during the preparation.

- Then I prepared diverse tasks related to almost all subjects of the exam. I knew that I would be struggling with time as it is stated in OCM exam page: “All skillsets have time limitations that require participants to employ best practices in order to successfully complete assigned task” So I allowed specific time limit for each task and tried to make a good and efficient use of time. Well, I remember when I was very quick during the exam and even imagined myself practicing in my own lab setting. At the end of each subject and section I have provided a wide range of tasks that need to be completed within a specified time frame.

- Next, I printed out all subjects of the exam and set a deadline for each section. One month per section was enough for me, so each month I was busy with just only one specific section (however, I was also practicing tasks of already completed sections during the weekends) By reading everything related with each subject and practicing almost everything that came to my mind, I stroke out that subject from the list and at the end, got the following view in front of me:
It meant that I was ready for the exam and I had not missed anything.

As I said before, I supposed that I would be struggling against the time during the exam, so I decided to improve my typing skills further. I had already good and rapid typing skills using all 10 fingers (with 100 word per minute), but by making further training and effort, I just wanted to make sure that I would make a few errors. You can find various typing applications to improve your typing. I used the one at www.typeracer.com to see how I can improve and ensure that I do less typos. And it helped me a LOT.

After 9 months I completed all my researches and made hundreds of different type of practices. I was able to perform most of the tasks with my eyes closed, perform anything using both graphic user interface (GUI) and command line interface (CLI). Two weeks before the exam I started practicing **ALL** tasks on **every single day without stop**.

The key contributing factor in my success has been both my experience and the training I went on doing for the preparation. In this book, I have shared all my practices with you. If you do all the practices given in this book within specified time limit, that means you are ready for the exam.

### Create OCM environment on your own laptop

You are not required to have special hardware to prepare for the OCM exam. You can do all kind of tests on your laptop including RAC and Data Guard. In this book you can find detailed and step-by-step installation and configuration of OCM environment on your own laptop. To do this you need the following softwares:

- Oracle Linux Release 5.4 (32-bit)
- Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 (32-bit)
- Oracle Enterprise Manager 11g Grid Control Release 1 (11.1.0.1.0) (32-bit)

Check the following links to download necessary software:

- To download Oracle Enterprise Linux, check the following link:
  [https://edelivery.oracle.com/linux](https://edelivery.oracle.com/linux)
- To download VirtualBox, check the following link:
- To download Oracle Database 11g (11.2.0.3) search for Patchset 10404530 from “Patchsets and Updates” tab from metalink.oracle.com or use the following direct link:
- Oracle Enterprise Manager 11g Grid Control 11.1.0.1.0 is not available at oracle.com.
You should follow metalink note 1071023.1 to get it from Oracle Support

- To download Enterprise Manager Agent version 11.1.0.1.0 check the following link:


  Once you download all necessary software, please install VirtualBox, create a virtual machine and install Oracle Enterprise Linux (OEL).

## Install and configure virtual machine

Download and install Oracle VirtualBox and create a new virtual machine. Then download Oracle Enterprise Linux 5.4 (if you plan to use Oracle 11gR1) or 6.4 (if you plan to use Oracle 11gR4). The next section will provide more detailed information about OEL setup.

VirtualBox configuration for the RAC installation is different and is described in details in Chapter 7. For now, just create a Virtual Machine with capacity of 30GB and provide 4GB RAM and make sure to choose “Host-only” for the Adapter 1 in the Network configuration Tab and enable it as follows:
Install Oracle Enterprise Linux

I have performed all my tests on OEL 6.4 running Oracle 11gR2 (11.2.0.4) and suggest you to follow the suit. However, if you rely on environment settings that are described in OCM exam page, you need to download and install Oracle Enterprise Linux 5.4 and Oracle 11gR2 (11.2.0.3). Installing Oracle 11gR2 on 6.4 is much easier. In the next sections, I have shown installation steps for both OEL 5.4 and OEL 6.4

Installing Oracle Enterprise Linux 5.4

Mount the .iso image of the Linux installation to the Virtual Machine and get it started. Click Next buttons until the “Package Configuration” page and choose Customize. Make sure you choose the following packages (and click on “Optional Packages” button to get access to the detailed package list)

**Desktop Environments**
- GNOME Desktop Environment

**Development**
- Development Libraries
- Development Tools
- GNOME Software Development
- Java Development
- Legacy Software Development
- X Software Development (Select libxpdevel and openmotif)

**Base System**
- Administration Tools
- Base
- Java
- Legacy Software Support (select compat-db)
- System Tools (select sysstat)
- X Window System
After selecting all necessary packages you can start the installation. When the installation is completed, virtual machine will be restarted and another configuration page will appear. Make sure to disable Firewall and SELinux (don’t create any user) and login to the system with a root user.

The first thing that you need to do after OS installation is to install Guest Additions. This is not a part of OCM exam and you will not be asked to perform OS or VirtualBox installation. We are just configuring our testing environment.

Select “Install Guest Additions” from “Devices” menu. Double click on CD-ROM device and run `VBoxLinuxAdditions.run` file (chose run in Terminal). When the installation is completed, go to the menu ‘Devices-Shared Clipboard” and check “Bidirectional”. Then go to System menu in Linux and Logout from root user. Now login to the OS and you will notice that you can move mouse outside of the machine without clicking on right Ctrl button.

After installing Guest Additions successfully, you can create a shared folder to move the Oracle installation files. Go to Devices menu and click on Shared Folders. Click “Add shared Folder” button and check the folder where installation files are located. Add it and mark “Read-Only” and “Auto-mount” options as shown in the box:

Now create an installation folder in Linux and mount the shared folder as follows:

```bash
[root@ocm11g ] mkdir /tmp/orasoft
[root@ocm11g tmp]# mount -t vboxsf Oracle_installation /tmp/orasoft
[root@ocm11g tmp]# cd /tmp/orasoft
[root@ocm11g tmp]# ls -ltr
```

We will see the installation of Oracle Software on the next chapter.
Before starting the installation, you need to perform some pre-requisite actions to prepare the OS for the Oracle installation:

- Create a user and an OS groups:

  ```
  [root@ocm11g ] groupadd  oinstall
  [root@ocm11g ] groupadd  dba
  [root@ocm11g ] useradd -m -g oinstall -G dba -d /home/oracle -s /bin/bash oracle
  [root@ocm11g ] passwd oracle
  ```

- Change `/etc/sysctl.conf` file as below and run `/sbin/sysctl -p` command:

  ```
  net.ipv4.ip_forward = 0
  net.ipv4.conf.default.rp_filter = 1
  net.ipv4.conf.default.accept_source_route = 0
  kernel.sysrq = 0
  kernel.core_uses_pid = 1
  net.ipv4.tcp_syncookies = 1
  kernel.msgmnb = 65536
  kernel.msgmax = 65536
  kernel.shmmx = 2147483648
  kernel.shmall = 2097152
  kernel.shmni = 4096
  kernel.sem = 250 32000 100 128
  fs.file-max = 6815744
  fs.aio-max-nr = 1048576
  net.ipv4.ip_local_port_range = 9000 65500
  net.core.rmem_default = 1048576
  net.core.rmem_max = 4194304
  net.core.wmem_default = 262144
  net.core.wmem_max = 1048576
  ```

- Change `/etc/pam.d/login` file and add the following line:

  ```
  session required pam_limits.so
  ```
Add the following lines to /etc/security/limits.conf file to change open file limits of OS:

```
oracle    soft    nproc   2047
oracle    hard    nproc   16384
oracle    soft    nofile  1024
oracle    hard    nofile  65536
```

- Create an installation folder:

```
[root@ocm11g oracle]# mkdir -p /u01/home/oracle/product/11.2.0/db_1
[root@ocm11g oracle]# chown -R oracle:oinstall /u01
```

- Create a folder to move the installation files from the shared folder mount point:

```
[root@ocm11g ~] su – oracle
[oracle@ocm11g ~] mkdir /home/oracle/orasoft
[oracle@ocm11g ~] cp /tmp/orasoft/*.zip /home/oracle/orasoft
```

- Change .bash_profile file and add following lines:

```
[root@ocm11g ~] su – oracle
[oracle@ocm11g ~] vi .bash_profile
export ORACLE_HOME=/u01/home/oracle/product/11.2.0/db_1
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
export PATH=$ORACLE_HOME/bin:$PATH
```

**Exam tip:** Before starting any task in the exam, make sure you set the environment variable and create an alias to connect to SQL*Plus easily as follows:

```
alias sql='"sqlplus / as sysdba"'
```

- Now mount the Linux installation .iso file (Devices->CD/DVD Devices) and install the missing packages:

```
[root@ocm11g ~] cd /media/(Linux_Installation_Folder)#/Server
[root@ocm11g ~] rpm -Uvh libaio-devel*
[root@ocm11g ~] rpm -Uvh unixODBC*
```
In order to have external access to the virtual machine from outside (from your laptop), go to the System->Administration->Network, click Edit button and provide Static IP address (make sure you provide a “Host-only” network adapter in the virtual machine) as follows:

![Ethernet Device](image)

Click OK and Activate the network adapter. Run /sbin/ifconfig command to check if the network is up and ping it from outside of virtual machine (from your laptop).

Now you can easily connect to the virtual machine using any SSH client (Putty, SecureCRT and etc.) directly from your laptop.

Next, copy the installation files from the Shared Folder to any folder in the OS and unzip the zip files. In the next chapter, I have presented the installation of an Oracle Software with both Graphic User and Command Line interfaces.
Installing Oracle Enterprise Linux 6.4

Installation of OEL 6.4 is identical to the rest OEL installations, but the main advantage of using version 6.4 is simplifying post installation configuration by using the new rpm package that is introduced in OEL 6 named `oracle-rdbms-server-11gR2-preinstall`. By running this package, the following actions are performed automatically:

- All necessary packages are installed
- User `oracle` with both `oinstall` and `dba` groups are created
- Kernel parameters are modified at `/etc/sysctl.conf` file
- Hard limits are modified at `/etc/security/limits.conf` file

After installing OS and running this package you can directly go to the installation of Oracle Software!

Start creating a virtual machine and install OEL 6.4. All steps are approximately same with OEL 5.4 installation that you have seen in the previous section.

After installation is completed, use `yum` to install Oracle 11gR2 pre-install rpm. If you are already connected to the internet, no configuration is needed before the installation. If you are not connected to the internet and want to install the rpm package using installation DVD that is mounted to the VirtualBox, the `yum` repository is needed to be configured.

For this purpose, create a new folder, mount the cdrom to it, create a new yum repository configuration file under `/etc/yum.repos.d` folder and run the installation of the package:

```
[root@ocm11g ~] mkdir /media/cdrom
[root@ocm11g ~] mount /dev/cdrom /media/cdrom/
[root@ocm11g ~] cd /etc/yum.repos.d
[root@ocm11g ~] rm -rf *
[root@ocm11g ~] cat <<EOF > ol64.repo
[OL64]
name=Oracle Linux 6.4 x86_64
baseurl=file:///media/cdrom
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1
EOF
[root@ocm11g ~] yum install -y oracle-rdbms-server-11gR2-preinstall
```
All pre-requisites are done and the OS is ready for Oracle installation! Now it is time to move to the next chapter where you will see different ways of Oracle Database installation.
CHAPTER 1

Server Configuration
iven your professional experience as a DBA, I assume you have gained sufficient experience to create a database, manage tablespaces, and configure network parameters and etc. I wonder if you ever tried to install Oracle Database Software in silent mode. Have you ever created a production database using CREATE DATABASE command or command line interface of DBCA? Have you ever configured network configuration files without Graphic User Interface (GUI)?

This chapter provides a detailed step-by-step guide on Oracle 11g Database Software setup and new database creation through GUI and silent mode likewise. Once a database is created, you will learn how to create and manage physical database files and tablespaces, and how to configure network configuration files. You will be able to appreciate the steps of new patch installation via OPatch utility and EMCA (Enterprise Manager Configuration Assistant) usage.

By the end of this chapter, you will learn how to install and use Grid Infrastructure to manage databases and different resources.
Here’s the list of OCM exam topics under “Server Configuration” section that we are going to review:

1.1. **Create the database**
1.2. **Determine and set sizing parameters for database structures**
1.3. **Create and manage temporary, permanent, and undo tablespaces**
1.4. **Stripe data files across multiple physical devices and locations**
1.5. **Configure the database environment to support optimal data access performance**
1.6. **Create and manage database configuration files**
1.7. **Create and manage bigfile tablespaces**
1.8. **Create and Manage a tablespace that uses NFS mounted file system file**
1.9. **Create and manage multiple network configuration files**
1.10. **Create and configure a listener**
1.11. **Configure the database instance to support shared server connections**
1.12. **Set up network tracing**
1.13. **Manage Oracle network processes**
1.14. **Configure the network environment to allow connections to multiple databases**
1.15. **Use configurationless connections**
1.16. **Use OPatch to install a patch**
1.17. **Use Grid Infrastructure to manage oracle databases and other resources**
1.18. **Use Enterprise Manager Configuration Assistant(EMCA) utility**
Install Oracle Software

There are different methods to create a database in Oracle and you will get familiar with some of them during exam. Prior to that, you need to install Oracle Database Software. You are not required to install Oracle during the exam, as it had to be installed already, but the test takers shall be ready for other possible unexpected circumstances anyhow. I’m quite certain you will not be asked to perform pre-requisite configurations, but capability to install Oracle Database is a must.

Oracle RDBMS software can be installed either by GUI or in silent mode. The simplest way is to use GUI, and if not available, then the silent mode is the only remaining option.

Installing Oracle Software with GUI

Once all pre-requisites are completed, extract the Oracle Database installation file and execute ./runInstaller.command.

Step-by-step Oracle Database 11gR2 installation is provided below.

Uncheck the box if you do not wish to receive security updates and click Next to proceed to the Installation option page. Figure 1-1.
Oracle Software installation and new database created can be completed in one go, or separately. I suggest to install the software only for now. Check the second option and click Next. Figure 1-2.
A RAC database is not to be created now, therefore please choose "Single instance database installation" in the "Node Selection" and click Next. Figure 1-3.
Choose “Enterprise Edition” in the “Select Database Edition” page and click Next. Figure 1-4.

FIGURE 1-3. Selecting installation type

FIGURE 1-4. Choosing database edition
Create database in a silent mode

Reference:
Oracle® Database Administrator’s Guide 11g Release 2 (11.2) -> Creating a Database with Noninteractive/Silent DBCA

As mentioned earlier, the GUI might be unavailable in some cases or you might be directly asked to create a database in silent mode using dbca during the exam. In this case, all you need is to run DBCA with “-silent” parameter and provide necessary parameters. After creating a database in a silent mode a couple of times, you will easily remember all necessary parameters by getting list of all parameter from the output of dbca -help command. Run the following command to create a database in a silent mode:

```
./dbca -silent -createdatabase -gdbname mydb -templatename /home/oracle/product/11.2.0/db_1/assistants/dbca/templates/General_Purpose.dbc -sid mydb -syspassword oracle -systempassword oracle -emConfiguration none -datafileDestination /home/oracle/oradata -memoryPercentage 40
```

Check the alert.log file during the database creation:

```
[oracle@ocm11g ~] cd $ORACLE_HOME/diag/rdbms/{sid}/{sid}/trace
[oracle@ocm11g ~] tail -f alertmydb.log
```

Once the installation completed successfully connect to the database and get the status of the instance:

```
[oracle@ocm11g ~] export ORACLE_SID=mydb
[oracle@ocm11g ~] sqlplus / as sysdba
SQL> SELECT status FROM V$INSTANCE;
```

The database is ready to use. Get the output of all parameters using dbca -silent command and try to use as many parameters as you can. To get used to the silent database creation procedure, drop the database by using following steps and recreate it again:

```
[oracle@ocm11g ~] export ORACLE_SID=mydb
[oracle@ocm11g ~] sqlplus / as sysdba
SQL> STARTUP FORCE MOUNT EXCLUSIVE RESTRICT;
SQL> DROP DATABASE;
```
**Practice:**

<table>
<thead>
<tr>
<th>Practice</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Create a database in a silent mode (using DBCA)</td>
<td>▶10 min.</td>
</tr>
<tr>
<td>- Get the list of all parameters from dbca – help output</td>
<td></td>
</tr>
<tr>
<td>- Run dbca –silent command to create a database</td>
<td></td>
</tr>
<tr>
<td>- Check alert.log file</td>
<td></td>
</tr>
<tr>
<td>▶ Drop database using “DROP DATABASE” command</td>
<td>▶5 min.</td>
</tr>
</tbody>
</table>

Create database manually using CREATE DATABASE command

**Reference:**

Oracle® Database Administrator’s Guide 11g Release 2 (11.2) -> Creating a Database with the CREATE DATABASE Statement

You may be required to create a database using ‘CREATE DATABASE’ command during OCM exam. In this case, the first thing to do is to create a parameter file.

Create the parameter file (pfile.ora) under /home/oracle folder and provide the necessary parameters. Only a few essential parameters should be kept in mind. Below you can find necessary parameters to create a database:

```
CONTROL_FILES='/home/oracle/oradata/control01.ctl','/home/oracle/oradata/control02.ctl'
DB_NAME=ocm
DB_UNIQUE_NAME=ocm
DB_RECOVERY_FILE_DEST='/home/oracle/flash_recovery_area'
DB_RECOVERY_FILE_DEST_SIZE=10g
DIAGNOSTIC_DEST='/home/oracle/diag'
MEMORY_TARGET=1024m
PGA_AGGREGATE_TARGET=130m
UNDO_TABLESPACE=undotbs01
```

Set ORACLE_SID environment variable, login to the SQL*Plus and start the instance in NOMOUNT mode to check if the parameter file is ok.

```
SQL> STARTUP NOMOUNT PFILE='/home/oracle/pfile.ora';
```
If the instance is started, then create the server parameter file (spfile) and start the instance again:

```sql
SQL> CREATE SPFILE FROM PFILE='/home/oracle/pfile.ora';
SQL> STARTUP NOMOUNT FORCE;
```

Now open the documentation and go to the book/chapter mentioned in the reference page of this topic, copy the CREATE DATABASE command to any editor (run `gedit` in Linux to open Graphical Editor), paste the command and make the necessary changes.

**Note:** Make sure UNDO tablespace name matches the one provided in the parameter file.

After getting the text of CREATE DATABASE command, edit it based exam requirements and run as follows:

```sql
CREATE DATABASE ocm
    USER SYS IDENTIFIED BY oracle
    USER SYSTEM IDENTIFIED BY oracle
    LOGFILE GROUP 1 ('/home/oracle/oradata/redo01a.log','/home/oracle/oradata/redo01b.log') SIZE 100M BLOCKSIZE 512,
        GROUP 2 ('/home/oracle/oradata/redo02a.log','/home/oracle/oradata/redo02b.log') SIZE 100M BLOCKSIZE 512,
        GROUP 3 ('/home/oracle/oradata/redo03a.log','/home/oracle/oradata/redo03b.log') SIZE 100M BLOCKSIZE 512
    MAXLOGFILES 5
    MAXLOGMEMBERS 5
    MAXLOGHISTORY 1
    MAXDATAFILES 100
    CHARACTER SET AL32UTF8
    NATIONAL CHARACTER SET AL16UTF16
    EXTENT MANAGEMENT LOCAL
    DATAFILE '/home/oracle/oradata/system01.dbf' SIZE 325M REUSE
    SYSAUX DATAFILE '/home/oracle/oradata/sysaux01.dbf' SIZE 325M REUSE
    DEFAULT TABLESPACE users DATAFILE '/home/oracle/oradata/users01.dbf'
        SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
```
DEFAULT TEMPORARY TABLESPACE tempts1 TEMPFILE '/home/oracle/oradata/temp01.dbf'
  SIZE 20M REUSE
UNDO TABLESPACE undotbs01 DATAFILE '/home/oracle/oradata/undotbs01.dbf'
  SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

Make sure to check alert.log file during database creation.

After the command is successfully finished, restart the instance and run catalog.sql and catproc.sql files from $ORACLE_HOME/rdbms/admin folder

```
SQL> SHUT IMMEDIATE
SQL> STARTUP
SQL> EXIT
[oracle@ocm11g ~] cd $ORACLE_HOME/rdbms/admin
[oracle@ocm11g ~] sqlplus / as sysdba
SQL> @catalog
SQL> @catproc
```

Create a database using OMF (Oracle Managed Files)

⚠️ **Reference:**

Oracle® Database Administrator’s Guide 11g Release 2 (11.2) -> Creating Oracle Managed Files at Database Creation

It’s possible to create a database using Oracle Managed Files (OMF) feature also. There are two mainly used parameters for this purpose, namely: DB_CREATE_FILE_DEST and DB_CREATE_ONLINE_LOG_DEST_n

- **DB_CREATE_ONLINE_LOG_DEST_n** parameter can be used several times to specify multiplexed copy of the online redo log and control file.
- **DB_CREATE_FILE_DEST** parameter specifies the default location for Oracle-managed data files. This location is also used as the default location for Oracle-managed control files and online redo logs if none of the **DB_CREATE_ONLINE_LOG_DEST_n** initialization parameters are specified. Create a new parameter file and edit it as follows:
1.8. Create and Manage a tablespace that uses NFS mounted file system file

I would strongly suggest checking step-by-step document attached in the above-mentioned MOS note. (DNFS Workshop)

With Oracle Database 11g, instead of using the operating system kernel NFS client, you can configure Oracle Database to access NFS V3 servers directly by using an Oracle internal Direct NFS Client. Here is how to configure NFS client:

To set NFS server, we need to know UID of the oracle user. Run the following command to get UID of the user:

```
[oracle@ocm11g ~] id oracle
uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),54322(dba)
```

As a root user, create a directory on the NFS server:

```
[oracle@ocm11g ~] mkdir /nfs_folder
[oracle@ocm11g ~] chown 54321:54321 /nfs_folder
```

Next, add this directory to /etc/exports file:

```
/nfs_folder *(rw,sync,all_squash,insecure,anonuid=54321,anongid=54321)
```

Make sure the NFS server gets started during boot of this server and start NFS service:

```
[oracle@ocm11g ~] chkconfig --level 345 nfs on
[oracle@ocm11g ~] service nfs start
```

Change /etc/fstab file and make sure NFS export is mounted on the nodes during boot time and mount the folder:
ocm11g:/nfs_folder /u01/nfs_mount/oradata nfs
rw,bg,hard,nointr,rsize=32768,wsize=32768,tcp,actimeo=0,vers=3,timeo=600 0 0
mount /u01/nfs_mount/oradata

Now, configure Direct NFS client and provide necessary information in the
$ORACLE_HOME/dbs/oranfstab file. Below you can see the definition of these parameters:

Server – The NFS server name
Path - Up to four network paths to the NFS server, specified either by IP address, or by name
Local - Up to four local paths on the database host, specified by IP address or by name
Export: The exported path from the NFS server.
Mount: The corresponding local mount point for the exported volume.

Make sure oranfstab file contains the following information:

server: ocm11g
path: 192.168.78.11
local: 192.168.78.12
path: 192.168.78.11
local: 192.168.78.12
export: /nfs_folder mount: /u01/nfs_mount/oradata

Oracle Database is not shipped with Direct NFS Client enabled by default. To enable Direct
NFS Client, change the directory to $ORACLE_HOME/rdbms/lib, run make command and
restart the database:

[oracle@ocm11g lib] make -f ins_rdbms.mk dnfs_on

[oracle@ocm11g ~] sqlplus / as sysdba
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP

Check the alert.log file to see the following message:

Oracle instance running with ODM: Oracle Direct NFS ODM Library Version 2.0

Now create the tablespace on the DNFS directory:

SQL> CREATE TABLESPACE tbs_nfs datafile ‘/nfs_oradata/tbs_nfs.dbf’ size 100m;
Tablespace created.
1.17. **Use Grid Infrastructure to manage oracle databases and other resources**

The Oracle Grid Infrastructure for a stand-alone server is the Oracle software that provides system support for an Oracle database including volume management, file system, and automatic restart capabilities. If you plan to use Oracle Restart or Oracle Automatic Storage Management (Oracle ASM), then you must install Oracle Grid Infrastructure before you install and create the database. Oracle Grid Infrastructure for a stand-alone server is the software that includes Oracle Restart and Oracle ASM. Oracle combines the two infrastructure products into a single set of binaries that is installed as the Oracle Grid Infrastructure home.

**Reference:**

Oracle® Database Administrator’s Guide 11g Release 2 (11.2) -> 4 Configuring Automatic Restart of an Oracle Database

Go to www.oracle.com -> Downloads -> Oracle Database -> 11.2.0.1 -> See all -> Oracle Grid Infrastructure 11g Release 2 (11.2.0.1.0) for Linux x86-64

Download the file, copy it to the temp directory, make sure you make necessary pre-requisites for the Grid Software installation as it is described in the first Reference documentations, unzip the file and start the installation

```
./runInstaller
```

From the “Installation Option” page, choose “Install Grid Infrastructure Software Only” and click Next.
Select English as a Language, and specify OSDBA, OSOPER and OSASM groups. You can create separate OS user called grid as an ASM administration, or you can use dba group. Click Next. Figure 1-33
Next, specify the Oracle Base and Oracle Software locations to store all Oracle software and configuration files. Figure 1-34.
We have already seen how to install required packages in the previous chapter. The *yum* repository should already be configured, so try to install the missing package with *yum*. Before using *yum*, make sure you mounted the Oracle Enterprise Linux Software already and mounted cd-rom (or DVD image) using mount /dev/cdrom /media/cdrom/ command.

If you still get some missing packages and fail to install them with *yum*, switch to the Packages folder in the mounted cd-rom, and run rpm –Uvh command to install the missing packages. In most cases, x86_64 version of the package is installed and the installation requires i386 version of the same package. If you get any dependencies issues while running rpm –Uvh command, run yum install command and specify the whole package name as follows:

```
[oracle@ocm11g ~] yum install unixODBC-2.2.14-12.el6_3.i686
```

Let's assume everything went fine and the software is installed successfully. Run the root.sh script with a root user. Figure 1-35.
List of tasks for the section Chapter 1 – Server Configuration:

<table>
<thead>
<tr>
<th>Practice</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td></td>
</tr>
<tr>
<td>► Create a .bash_profile file</td>
<td>3-5 min.</td>
</tr>
<tr>
<td>► Install Oracle in a silent mode (configure response file, run the installation and track it)</td>
<td>10 min.</td>
</tr>
<tr>
<td><strong>Create the Database</strong></td>
<td></td>
</tr>
<tr>
<td>► Create a database in a silent mode (using DBCA)</td>
<td>10 min.</td>
</tr>
<tr>
<td>► Get the list of all parameters from dbca –help output</td>
<td></td>
</tr>
<tr>
<td>► Run dbca –silent command to create a database</td>
<td></td>
</tr>
<tr>
<td>► Check alert.log file</td>
<td></td>
</tr>
<tr>
<td>► Drop database using “DROP DATABASE” command</td>
<td>5 min.</td>
</tr>
<tr>
<td>► Create a database using CREATE DATABASE command using OMF feature</td>
<td>15 min.</td>
</tr>
<tr>
<td><strong>Determine and set sizing parameters for database structures</strong></td>
<td></td>
</tr>
<tr>
<td>► Set different parameter files</td>
<td>10 min.</td>
</tr>
<tr>
<td>► Set MEMORY_TARGET and check the free memory</td>
<td>10 min.</td>
</tr>
<tr>
<td>► Set MEMORY_TARGET, SGA_TARGET and PGA_AGGREGATE_TARGET</td>
<td>15 min.</td>
</tr>
<tr>
<td><strong>Create and manage temporary, permanent, and undo tablespaces</strong></td>
<td></td>
</tr>
<tr>
<td>► Create locally managed, compressed, encrypted tablespace</td>
<td>10 min.</td>
</tr>
<tr>
<td>► Create tablespace with all available parameters</td>
<td>5 min</td>
</tr>
<tr>
<td>► Create tablespace with different block size</td>
<td>3 min</td>
</tr>
<tr>
<td>► Make tablespace READ ONLY/READ WRITE, OFFLINE, ONLINE, SHRINK, RENAME, ADD DATAFILE, RESIZE, DROP</td>
<td>10 min.</td>
</tr>
<tr>
<td>► Manage UNDO tablespace (create, switch, drop, set undo retention, enable and disable retention guarantee)</td>
<td>10 min.</td>
</tr>
<tr>
<td>► Create temporary tablespace, tablespace group, switch, drop, resize and shrink</td>
<td>10 min.</td>
</tr>
</tbody>
</table>
CHAPTER 2

Enterprise Manager Grid Control
Oracle Enterprise Manager (OEM) is not only used for the tasks of this session, but for the most of the tasks of different sections as well. Most probably, there will not be a lot of time for you to create a composite partitioned table or configure resource manager with myriad of different parameters using command line.

This chapter starts by illustrating a detailed step by step installation of OEM Grid Control starting from the download methods of all OEM components till successful installation.

In this chapter we will also see how to create Programs, automate Jobs, create Schedulers, Alerts and Notifications.

The main two methods of agent deployment are described and practically demonstrated likewise.
Here’s the list of OCM exam topics under “Enterprise Manager Grid Control” section that we are going to review:

2.1. Install and Patch Enterprise Manager Grid Control software
2.2. Configure the Enterprise Manager repository
2.3. Create Enterprise Manager Grid Control users
2.4. Use Enterprise Manager to modify a database configuration
2.5. Configure Enterprise Manager to modify database availability
2.6. Create and manage jobs
2.7. Create and monitor alerts
2.8. Create notifications
2.9. Implement Grid Control and Database Control
2.10. Choose the appropriate tablespace type for the intended use
2.11. Create Scheduler jobs
2.12. Create schedules
2.13. Assign jobs to windows
2.14. Create programs
2.15. Create job classes
2.16. Install the Enterprise Manager Grid Control infrastructure
2.17. Deploy Enterprise Manager Grid Control agents
2.18. Configure Grid Control for business requirements
2.1. Install and Patch Enterprise Manager Grid Control software

There is an entire book (documentation) written on how to install Oracle Enterprise Manager Grid Control (OEM) on Linux. Move to the documentation home page and check the following book:

Reference:
Oracle® Enterprise Manager Grid Control Advanced Installation and Configuration Guide 11g Release 1 (11.1.0.1.0)

You can also check the following step by step OEM 11g installation guide from oracle.com and a couple of MOS notes:

Reference:
http://www.oracle.com/webfolder/technetwork/tutorials/obe/em/emgc1110/installemgc11101 linux/emgc11101install.htm?print=preview&imgs=visible
http://download.oracle.com/technology/products/oem/master_note_11g_grid_control_install.htm
Master Note for Enterprise Manager 11G Grid Control 11.1.0.1.0 Installation and Upgrade (Doc ID 1067438.1)

In this section, I will provide you with detailed step by step installation guide of OEM 11gR1 software. Before installing OEM, you should create a new Virtual Machine and install Oracle Linux 5.4. Based on OCM exam requirements, you should install Oracle Enterprise Manager 11g Grid Control Release 1 (11.1.0.1.0) software.

Prior Grid Control 11gR1 installation, you must first prepare the Weblogic Server and Database installations that will support the Management Server and Management Repository. Comprehensive information about necessary software is provided in the following section.

Downloading necessary softwares

Download and install the following softwares to successfully install and implement Oracle Enterprise Grid Control:
- Oracle Linux Release 5.4 (32-bit)
- Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 (32 bit)
- Oracle Enterprise Manager 11g Grid Control Release 1 (11.1.0.1.0) (32 bit)
- JDK 1.6 (32 bit)
- WebLogic 10.2.3 (This is the only support version for OEM 11g Grid Control)

The OEM Grid Control installation will fail if different version of JDK or Weblogic software is used.
Chapter 2: Enterprise Manager Grid Control

FIGURE 2-27. Login to Oracle Enterprise Manager

FIGURE 2-28. OEM Grid Control home page
It is strongly recommended to install flash player to see performance tuning charts in the browser. Go to the following link, download required flash player, install it and restart the browser: https://get.adobe.com/flashplayer/

The following error may appear while accessing the OEM web page during preparation:

Backend WLS or EM application seems to be down

Check the OEM log file for more information as there are two major causes of this error. First, it might happen if the database or listener is down. Ensure the database is open and registered in the listener. Secondly, it might happen after password change of the SYSMAN user. If you want to change the password, make sure to follow the steps described at Metalink Note: 270516.1 (How to Change the Password of SYSMAN User in 10g and 11g Grid Control?) The key point here is to change the password in the emoms.properties file by adding the following lines:

oracle.sysman.eml.mntr.emdRepPwd=new_password
oracle.sysman.eml.mntr.emdRepPwdEncrypted=FALSE

Restart the OMS to access the web page.

Applying patch to OEM Grid Control

One of the requirements of OCM exam is to know how to apply patches to the Grid Control. There are few bugs that should be resolved by applying patches to OEM regarding the following metalink note:

Reference:
Required Patches for Grid Control 11g (11.1.0.1.0) (Doc ID 1101208.1)

Patch 9738008 is one of the required patches that should be applied after fresh OEM installation. Download it from metalink and move it to the OEM host:

[oracle@oemgrid installation]$ ls -ltr p9738008_111010_LINUX.zip -rw-rw-rw- 1 oracle oinstall 1452092 May 11 12:03 p9738008_111010_LINUX.zip
[oracle@oemgrid installation]$
Unzip it, set environment variables and install it as follows:

```
[oracle@oemgrid installation]$ unzip p9738008_111010_LINUX.zip
Archive:  p9738008_111010_LINUX.zip
  creating: 9738008/
  creating: 9738008/etc/
  creating: 9738008/etc/config/
  inflating: 9738008/etc/config/actions.xml
  inflating: 9738008/etc/config/deploy.xml
<< --- output trimmed -- >>
```

Before starting installation of any patch, make sure to read the README.txt file carefully. Set environment variables and run `opatch lsinventory` command to check the list of patches already applied:

```
[oracle@oemgrid installation]$ export ORACLE_HOME=/home/oracle/Oracle/Middleware/oms11g

[oracle@oemgrid installation]$ export PATH=$ORACLE_HOME/bin:/home/oracle/Oracle/Middleware/oms11g/OPatch:$PATH

[oracle@oemgrid installation]$ opatch lsinventory
Invoking OPatch 11.1.0.8.0
Oracle Interim Patch Installer version 11.1.0.8.0
Copyright (c) 2009, Oracle Corporation. All rights reserved.

Oracle Home : /home/oracle/Oracle/Middleware/oms11g
Central Inventory : /u01/app/oraInventory
  from : /etc/oraInst.loc
OPatch version : 11.1.0.8.0
OUI version : 11.1.0.8.0
OUI location : /home/oracle/Oracle/Middleware/oms11g/oui
Log file location : /home/oracle/Oracle/Middleware/oms11g/cfgtoollogs/opatch/opatch2015-05-11_12-01-22PM.log
```

```
------------------------------------------------------------------------
Installed Top-level Products (1):

Oracle Enterprise Manager Grid Console 11.1.0.1.0
There are 1 products installed in this Oracle Home.
```
2.8. Create notifications

After creating metrics, configuring necessary thresholds and getting some alerts, you might be asked to create a notification mechanism to get emails regarding alerts during OCM exam. There are four notification types in OEM:

- SMTP – is used to send an email by providing smtp server information.
- OS Command – is used to execute an OS command or a script.
- SNMP trap – is used to send out an SNMP trap with notification details.
- PL/SQL procedure – is used to execute a PL/SQL procedure from the repository database.

The easiest and most common way to configure the notification is to provide an smtp server and get direct emails about OEM alerts when problem occurs. And most probably, you might be asked to configure the notification by configuring SMTP e-mail method.

You can get two types of emails:

- Long messages – is used with regular email accounts and detailed information is sent in a body part of the message.
- Short messages – is supposed to be used for SMS text messages and have limited length.

To configure SMTP notification, click on Setup link on the right top of the OEM page, go to the Notification Method link and fill the form. Figure 2.69.
You may get the notification once or repeatedly. To configure it, scroll down, check “Send Repeat Notification” checkbox and provide repeat frequency and maximum repeat notifications. 2.70.

**FIGURE 2-70. Configure Repeat Notification**

After configuring email notification method, click on Preferences link and go to the Notification Rules link to adjust rules for notifications.

**FIGURE 2-71. Notification Rules**

If you click on “Database Availability and Critical States” rule, you will see a list of assigned metrics. Figure 2.72.
CHAPTER 3
Managing Database Availability
This chapter covers an introduction to backup and recovery techniques of an Oracle database. Prior to taking an OCM exam, you must ensure you can take any kind of backup and can recover from any kind of failure.

Within this chapter we have looked at the steps to create a recovery catalog by using recovery catalog 11g new features such as creating a virtual private catalog and merging catalogs with each other.

We also introduced a number of ways to create a backup of a database and provided detailed step by step explanation of database recovery from different failure cases through Recovery Manager (RMAN).

Finally, we provided information about flashback database and configuration of flash recovery area. This chapter is concluded by a detailed information on various flashback technologies.
Here’s the list of OCM exam topics under “Managing Database Availability” section that we are going to review:

3.1. **Maintain recovery catalog**
3.2. **Configure Recovery Manager**
3.3. **Use Recovery Manager to perform database backups**
3.4. **Use Recover Manager to perform complete database restore and recovery operations**
3.5. **Configure RMAN**
3.6. **Create different types of RMAN backups to cater for different performance and retention requirements**
3.7. **Set Flashback Database parameters**
3.8. **Configure a Fast Recovery Area**
3.9. **Perform various recovery operations using Flashback technology**
3.1. Maintain recovery catalogs

Reference:
- Oracle® Database Backup and Recovery User’s Guide 11g Release 2 (11.2) -> 13 Managing a Recovery Catalog
- Recovery catalog for RMAN backup (Doc ID 452529.1)
- How To Configure RMAN Recovery Catalog Using Enterprise Manager DB Console. (Doc ID 467969.1)

☑ Note: A golden rule before starting this section, is – “Make sure all your databases are running in ARCHIVELOG mode”. After creating every database during the exam, make sure you have fresh backup of the database (even if you are not asked to do so)

In fact, a recovery catalog is a schema that is created to store backup metadata and used by RMAN to track one or more database backups. The recovery catalog contains information about RMAN operations, including:

- Datafile and archived redo log backup sets and backup pieces
- Datafile copies
- Archived redo logs and their copies
- Tablespaces and datafiles on the target database
- Stored scripts, which are named user-created sequences of RMAN commands
- Persistent RMAN configuration settings

Creating a Recovery Catalog

First of all, you need to create a tablespace, recovery catalog owner and grant necessary privileges. Generally, recovery catalogs are created in a different database or it can be a schema in the same database.

Create a new database (RCDB), configure TNS configurations and connect to the database:

```
[oracle@ocm11g ~] sqlplus system/oracle@RCDB
SQL> CREATE TABLESPACE rc_catalog DATAFILE '/home/oracle/oradata/rc_catalog.dbf'
SIZE 100M REUSE AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
SQL> CREATE USER rc_owner IDENTIFIED BY rc_owner;
SQL> GRANT RECOVERY_CATALOG_OWNER TO rc_owner;
SQL> ALTER USER rc_owner DEFAULT TABLESPACE rc_catalog;
SQL> ALTER USER rc_owner QUOTA UNLIMITED on rc_catalog;
```
RMAN> RUN
{ 
    ALLOCATE CHANNEL ch1 DEVICE TYPE DISK ;
    ALLOCATE CHANNEL ch2 DEVICE TYPE DISK ;
    BACKUP SECTION SIZE 200M DATAFILE 2;
}

In this example two channels backs up 200Mb file section of datafile 2.

Practice:

<table>
<thead>
<tr>
<th>Practice</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>►Enable controlfile autobackup, take the full backup to the specific directory</td>
<td>►5 min.</td>
</tr>
<tr>
<td>►Allocate 4 channels for different folders and take the full backup splitted between 4 different folders</td>
<td>►8 min.</td>
</tr>
<tr>
<td>►Backup system and sysaux tablespaces</td>
<td>►1 min.</td>
</tr>
<tr>
<td>►Backup any datafile with controlfile included</td>
<td>►1 min.</td>
</tr>
<tr>
<td>►Backup control file and parameter file</td>
<td>►1 min.</td>
</tr>
<tr>
<td>►Enable block change tracking, take full backup and then incremental backup</td>
<td>►5 min.</td>
</tr>
<tr>
<td>►Perform incrementally updated backup scenario</td>
<td>►12 min.</td>
</tr>
<tr>
<td>►Take multisectional backup</td>
<td>►5 min.</td>
</tr>
</tbody>
</table>

3.4. Use Recover Manager to perform complete database restore and recovery operations

As it was mentioned before, you should expect any kind of failure during the exam and must be ready for any kind of crash case of the database during the OCM exam. In this section, I provide a detailed overview of restore/recovery procedures that are available in Oracle and provide different kind of test cases and scenarios.

During the exam (and also you daily job) you must be ready for the following recovery cases:
- Restore parameter file
- Recover from loss of control file
- Restore/recover a datafile (Recover datafile that is not backed up)
- Restore/recover a tablespace
- Restore archivelog files
- Perform Block media recovery
- Performing disaster recovery
- Perform incomplete recovery (Time, SCN and Change based)
- Recover database to the previous incarnation
- Performing tablespace point in time recovery (TSPITR)

Refer to the following documentation to get more information about RMAN recovery.

**Reference:**
Database Backup and Recovery User’s Guide
17 Performing Complete Database Recovery
19 Performing Block Media Recovery
20 Performing RMAN Recovery: Advanced Scenarios
21 Performing RMAN Tablespace Point-in-Time Recovery (TSPITR)

**Restore parameter file**

Oracle database cannot be started without parameter file. This occurs since parameter file is read by Oracle in first place during the startup.

If you lost the parameter file, restore it from RMAN. If you don’t have RMAN backup of spfile, refer to alert.log file to get list of all non-default parameters that are logged during startup of the database:

Starting up:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options.
Using parameter settings in server-side spfile /home/oracle/product/11.2.0/db_1/dbs/spfileocm.ora
System parameters with non-default values:
- processes = 150
- memory_target = 1568M

Add all non-default parameters to the new parameter file, make sure to add mandatory parameters such as DB_NAME, CONTROL_FILES, UNDO_TABLESPACE and etc. You can check mandatory parameters from the “Chapter 1 – Create Database – Create database manually using CREATE DATABASE command” section.
3.7. Set Flashback Database parameters

Oracle Flashback Database feature is used to revert the database back to the specific time. When enabled, it creates flashback log files and stores them in the flash recovery area by applying flashback log files.

During an OCM exam, you might be asked to configure Oracle Flashback Database and bring the database back to the specific time without restoring and recovering the database with RMAN (which might take a long time), but by applying flashback log files and rewinding the database back.

You can use flashback database option both from RMAN and SQL*Plus.

Creating restore point

Restore point is a name that is associated with the SCN number of the database or specific point in time and is used to flash a table or a database back. Run the following command to create a restore point:

SQL> CREATE RESTORE POINT rest_01;

To assure to flash the database back to the specific time or SCN value by keeping flashback log files regardless the value of DB_FLASHBACK_RETENTION_TARGE parameter. Run the following command to create a guarantee restore point:

SQL> CREATE RESTORE POINT rest_guar_01 GUARANTEE FLASHBACK DATABASE;

You can query the list of all restore points either using V$RESTORE_POINT view, or running LIST RESTORE POINT ALL command from RMAN.

To drop restore point, use DROP RESTORE POINT restore_point_name command as follows:

SQL> DROP RESTORE POINT rest_01;
Enable Flashback database

To enable flashback database feature, set DB_FLASHBACK_RETENTION_TARGET parameter (in minutes) and run the following command:

```sql
SQL> ALTER DATABASE FLASHBACK ON;
```

To disable flashback option for a specific tablespace, use following command:

```sql
SQL> ALTER TABLESPACE my_tablespace FLASHBACK OFF;
```

Run the following command to disable the flashback database option completely:

```sql
SQL> ALTER DATABASE FLASHBACK OFF;
```

To get a detailed information about the flashback logs, query V$FLASHBACK_DATABASE_STAT view.

Flashback Database scenario

In the following scenario we will enable flashback database option and create a new table. First, we will create a restore point and revert the database back to this restore point. Next, we will truncate the table, create a second table and will try to flashback the database back to rollback the truncated data, open it in read only mode and extract the data. Then will roll the database forward, open its last state and import the data of the truncated table. Below, you can find the plan of the scenario and detailed explanation:

- Check the status of the flashback database and enable it.
- Change DB_FLASHBACK_RETENTION_TARGET parameter.
- Create a new table based of DBA_OBJECTS view. Update all rows.
- Check flash recovery area for flashback logs. Query V$FLASHBACK_DATABASE_STAT view.
- Get count of the rows the table.
  - Create restore point.
  - Get current SCN.
  - Check restore point from V$RESTORE_POINT view.
  - Delete all rows and commit the transaction.
- Revert the database back to the restore point. Open the database and check the count of the rows.
- Get the current SCN and truncate the table.
Flashback Transaction Query

To get information about each version of the row in a transaction level, use FLASHBACK_TRANSACTION_QUERY view. To query this view, you need to be granted SELECT ANY TRANSACTION privilege. Run the following SQL command and specify the table name that was used in the previous scenario:

```sql
SQL> col logon_user format a5
SQL> col operation format a10
SQL> col table_name format a25
SQL> col undo_sql format a50
SQL> set linesize 150
SQL> set pagesize 1000
SQL> SELECT xid, start_scn,
         logon_user,
         operation,
         table_name,
         undo_sql
FROM flashback_transaction_query
WHERE table_name = 'TBL_FL_VERSIONS_QUERY';
```
### Flashback Transaction Backout

Flashback Transaction Backout feature allows you to backout a single transaction with all dependent transactions by applying compensating statements for the affected transaction.

Now use the previous scenario and roll back all dependent transactions. In this example, we rollback transactions up to the second update. So get the XID of that transaction and provide it to `DBMS_FLASHBACK.TRANSACTION_BACKOUT` procedure as follows:

```sql
SQL> SELECT * FROM TBL_FL_VERSIONS_QUERY;
no rows selected

SQL> DECLARE
    v_txid XID_ARRAY;
BEGIN
```

### Table: Flashback Transaction Backout

<table>
<thead>
<tr>
<th>XID</th>
<th>START_SCN</th>
<th>LOGON</th>
<th>OPERATION</th>
<th>TABLE_NAME</th>
<th>UNDO_SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1100200055000000</td>
<td>2910888</td>
<td>SYS</td>
<td>UPDATE</td>
<td>TBL_FL_VERSIONS_QUERY</td>
<td>update &quot;SYS&quot;.&quot;TBL_FL_VERSIONS_QUERY&quot; set &quot;NAME&quot; = 'The first Update' where ROWID = 'AAASc1AABAAAYo5A</td>
</tr>
<tr>
<td>1200160056000000</td>
<td>2910880</td>
<td>SYS</td>
<td>INSERT</td>
<td>TBL_FL_VERSIONS_QUERY</td>
<td>delete from &quot;SYS&quot;.&quot;TBL_FL_VERSIONS_QUERY&quot; where ROWID = 'AAASc1AABAAAYo5AAA';</td>
</tr>
<tr>
<td>130007005B000000</td>
<td>2910885</td>
<td>SYS</td>
<td>UPDATE</td>
<td>TBL_FL_VERSIONS_QUERY</td>
<td>update &quot;SYS&quot;.&quot;TBL_FL_VERSIONS_QUERY&quot; set &quot;NAME&quot; = 'Oracle Certified Master 11g' where ROWID = 'AAASc1AABAAAYo5AAA';</td>
</tr>
<tr>
<td>1400000055000000</td>
<td>2910891</td>
<td>SYS</td>
<td>DELETE</td>
<td>TBL_FL_VERSIONS_QUERY</td>
<td>insert into &quot;SYS&quot;.&quot;TBL_FL_VERSIONS_QUERY&quot;(&quot;ID&quot;,&quot;NAME&quot;) values ('1','The second Update');</td>
</tr>
</tbody>
</table>
```
CHAPTER 4

Data Management
This chapter provides an introduction to main data management tasks that you might face during OCM exam. In the beginning of the chapter, having talked about materialized views and materialized view logs, we show how query rewrite feature is used with different examples.

Then we show how to create and manage encrypted tablespaces and show a step by step example on how to transport a tablespace from one database to another.

Next we provide a very detailed information on creating and administering external tables in Oracle.

Using a Data Pump for moving data from one database to another is one of the daily jobs of a DBA and requirement of an OCM exam. In this chapter we cover almost all useful Data Pump parameters and provide a lot of different scenarios on how to use them to import and export data.

Towards the end of the chapter, we talk about Oracle parallel query and show ways to administer and manage parallel executions.
Here's the list of OCM exam topics under “Data Management” section that we are going to review:

4.1. Manage Materialized Views to improve rewrite and refresh performance
4.2. Configure and manage distributed materialized views
4.3. Create and Manage encrypted tablespaces
4.4. Manage Transport of tablespaces across platforms
4.5. Configure a schema to support a star transformation query
4.6. Administer external tables
4.7. Implement Data Pump export and import jobs for data transfer
4.8. Implement Data Pump to and from remote databases
4.9. Configure and use parallel execution for queries
4.10. Use SQL*Loader
4.11. Administer, manage and tune parallel execution
4.1. Manage Materialized Views to improve rewrite and refresh performance

A materialized view is an object that contains precomputed result of a query against local or remote table, view or materialized views which are also called master tables.

There are three types of materialized views: read only, updatable and writable. Read only materialized views can't be updated and no DML statements can be performed against them. To make materialized view updatable, `FOR UPDATE` clause is used. Changes made to the materialized view are pushed to the master table when the refresh is occurred. However, when creating writeable materialized view, all DML statements that performed against the view are not pushed back to the master table when the view is refreshed and all changes are lost.

`CREATE MATERIALIZED VIEW` command is used to create a materialized view. If you need to get syntax of this command in detail, switch to the “Database SQL Language Reference” documentation and jump to “CREATE MATERIALIZED VIEW” syntax.

When creating a materialized view you can define storage parameters, the refresh mode and option, whether to use query rewrite or not and a select statement which defines the content of the view.

Let’s create a simple materialized view based on the object locates on the local database:

```sql
SQL> CREATE MATERIALIZED VIEW mv_dba_objects AS SELECT * FROM dba_objects;
Materialized view created.

SQL> SELECT COUNT (1) FROM mv_dba_objects;
COUNT(1)
---------
75605

SQL>
```
Materialized view query rewrite

When running a query with multiple joins and aggregate functions, it may take a lot of time to get a result. But if you have a materialized view with precomputed data that matches the same query you are running, it might be used instead. This feature is called query rewrite. In order to use this feature, a materialized view must be enabled for the query rewrite by having ENABLE QUERY REWRITE clause while creating the view and QUERY_REWRITE_ENABLED parameter must be enabled (either on database, or session level) which is true by default.

In the following example, we will see how this feature is used automatically. For this, create 2 tables based on DBA_OBJECTS and DBA_EXTENTS as a system user as follows:

```
SQL> CREATE TABLE tbl_mv_objects AS
SELECT *
FROM dba_objects
WHERE object_id IS NOT NULL;
Table created.

SQL> CREATE TABLE tbl_mv_extents AS
SELECT * FROM dba_extents;
Table created.

SQL>
```

Now join both tables and get the execution plan as follows:

```
SQL> SET AUTOTRACE TRACEONLY EXP
SQL> SET LINESIZE 150
SQL> SELECT a.object_name, COUNT (extent_id)
FROM tbl_mv_objects a, tbl_mv_extents b
WHERE a.object_name = b.segment_name
```

GROUP BY a.object_name;

Execution Plan
----------------------------------------------------------
Plan hash value: 3294975642

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost (%CPU)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td>222K</td>
<td>25M</td>
<td>343 (3)</td>
<td>00:00:05</td>
</tr>
<tr>
<td>1</td>
<td>HASH GROUP BY</td>
<td></td>
<td>222K</td>
<td>25M</td>
<td>343 (3)</td>
<td>00:00:05</td>
</tr>
<tr>
<td>* 2</td>
<td>HASH JOIN</td>
<td></td>
<td>222K</td>
<td>25M</td>
<td>336 (1)</td>
<td>00:00:05</td>
</tr>
<tr>
<td>3</td>
<td>TABLE ACCESS FULL</td>
<td>TBL_MV_EXTENTS</td>
<td>14025</td>
<td>753K</td>
<td>38 (0)</td>
<td>00:00:01</td>
</tr>
<tr>
<td>4</td>
<td>TABLE ACCESS FULL</td>
<td>TBL_MV_OBJECTS</td>
<td>72058</td>
<td>4644K</td>
<td>297 (1)</td>
<td>00:00:04</td>
</tr>
</tbody>
</table>

Predicate Information (identified by operation id):
---------------------------------------------------
2 - access("A"."OBJECT_NAME"="B"."SEGMENT_NAME")

Note
-----
- dynamic sampling used for this statement (level=2)
SQL>

Now create materialized view logs and a materialized view with the same query as follows:

SQL> CREATE MATERIALIZED VIEW LOG ON tbl_mv_objects WITH ROWID, SEQUENCE(object_name) INCLUDING NEW VALUES;

Materialized view log created.

SQL> CREATE MATERIALIZED VIEW LOG ON tbl_mv_extents WITH ROWID, SEQUENCE(extent_id, segment_name) INCLUDING NEW VALUES;

Materialized view log created.
Practice:

<table>
<thead>
<tr>
<th>Practice</th>
<th>Time</th>
</tr>
</thead>
</table>
| 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, cert, oracle, expert  
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. | ▶ 13 min. |
| 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. |

4.7. Implement Data Pump export and import jobs for data transfer

Reference:
Data Warehousing and Business Intelligence -> Database Utilities -> 2 Data Pump Export
Data Warehousing and Business Intelligence -> Database Utilities -> 3 Data Pump Import
Master Note for Data Pump (Doc ID 1264715.1)
How to use the Data Pump API: DBMS_DATAPUMP (Doc ID 1985310.1)
Import Data Pump: How to Import Table Data into a Table that has Different Name? (Doc ID 342314.1)
Export/Import Data Pump Parameter QUERY - How to Specify a Query (Doc ID 277010.1)
How To Use The REMAP_DATA Command Line Parameter With Datapump (Doc ID 1271820.1)
How To Use The Data Pump Import (IMPDP) Parameters? (Doc ID 1949193.1)
Export/Import Data Pump Parameter TRACE – How to Diagnose Oracle Data Pump [ID 286496.1]
Oracle Data Pump Encrypted Dump File (White Paper)
Data Pump in Oracle® Database 11g Release 2: Foundation for Ultra High-Speed Data Movement Utilities (White Paper)
Data Transformations with Oracle Data Pump (White Paper)

Using Data Pump a data or metadata can be exported to the file called dump file and loaded to the database. It consists of two programs: EXPDP which is used to unload an object, schema, database to the flat file and IMPDP is used to load the flat file to the database.
During an OCM exam, you might be asked to perform different export and import scenarios using Data Pump utility. In this section we will talk about different parameters of expdp and impdp programs and show an example for each parameter.

First of all, let's see a list of different examples of using EXPDP utility.

**Performing a table mode export (TABLES parameter)**

To export a specific table or partition of the table, use TABLES parameter. The syntax is “TABLE=schema.name.table_name:partition_name”. In the following example we create a table and export it to the dump file:

SQL> CONN system/oracle
Connected.

SQL> CREATE TABLE tbl_dp_tab_export AS SELECT * FROM dba_objects;
Table created.

SQL> COL directory_path FORMAT a30
SQL> SET LINESIZE 150
SQL> SELECT * FROM dba_directories;

<table>
<thead>
<tr>
<th>OWNER</th>
<th>DIRECTORY_NAME</th>
<th>DIRECTORY_PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYS</td>
<td>MYDIR</td>
<td>/tmp</td>
</tr>
</tbody>
</table>

SQL> exit

[oracle@oemgrid ~]$ expdp system/oracle directory=mydir dumpfile=export_table.dmp logfile=export_table.log tables=tbl_dp_tab_export

Starting “SYSTEM”.”SYS_EXPORT_TABLE_01”: system/******** directory=mydir dumpfile=export_table.dmp logfile=export_table.log tables=tbl_dp_tab_export
Estimate in progress using BLOCKS method...
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 9 MB
Processing object type TABLE_EXPORT/TABLE/Table
Processing object type TABLE_EXPORT/TABLE/PRE_TABLE_ACTION
. . exported “SYSTEM”.”TBL_DP_TAB_EXPORT” 7.373 MB 75895 rows
Master table “SYSTEM”.”SYS_EXPORT_TABLE_01” successfully loaded/unloaded
Dump file set for SYSTEM.SYS_EXPORT_TABLE_01 is:
/tmp/export_tab_partition2.dmp
Job “SYSTEM”."SYS_EXPORT_TABLE_01” successfully completed at 01:20:23

Performing a schema mode export (SCHEMAS parameter)

SCHEMAS parameter is used to export objects of specific user or list of users. In the following example we create two new users with two tables and export both users:

```
SQL> CREATE USER usr_exp1 IDENTIFIED BY usr_exp1;
User created.

SQL> CREATE USER usr_exp2 IDENTIFIED BY usr_exp2;
User created.

SQL> GRANT CONNECT, RESOURCE TO usr_exp1,usr_exp2;
Grant succeeded.

SQL> CREATE TABLE usr_exp1.tbl_one AS SELECT * FROM dba_objects;
Table created.

SQL> CREATE TABLE usr_exp1.tbl_two AS SELECT * FROM dba_objects;
Table created.

SQL> CREATE TABLE usr_exp2.tbl_one AS SELECT * FROM dba_objects;
Table created.

SQL> CREATE TABLE usr_exp2.tbl_two AS SELECT * FROM dba_objects;
Table created.

[oracle@oemgrid ~]$ expdp system/oracle directory=mydir dumpfile=schema_export_01.dmp logfile=schema_export_01.log schemas=usr_exp1,usr_exp2
```

Starting “SYSTEM”."SYS_EXPORT_SCHEMA_01": system/******** directory=mydir
dumpfile=schema_export_01.dmp logfile=schema_export_01.log schemas=usr_exp1,usr_exp2
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 36 MB
Processing object type SCHEMA_EXPORT/TABLE/TABLE
  . . exported “USR_EXP1”."TBL_ONE"                      7.364 MB  75824 rows
  . . exported “USR_EXP1”."TBL_TWO"                      7.364 MB  75825 rows
  . . exported “USR_EXP2”."TBL_ONE"                      7.364 MB  75826 rows
  . . exported “USR_EXP2”."TBL_TWO"                      7.364 MB  75827 rows
Master table “SYSTEM”."SYS_EXPORT_SCHEMA_01” successfully loaded/unloaded

**************************************************************************
Dump file set for SYSTEM.SYS_EXPORT_SCHEMA_01 is:
   /tmp/schema_export_01.dmp
Job “SYSTEM”."SYS_EXPORT_SCHEMA_01” successfully completed at 01:42:00

[oracle@oemgrid ~]$}

Performing full database export (FULL parameter)
To perform a full database export, use FULL=YES syntax as follows:

[oracle@oemgrid ~]$ expdp system/oracle directory=mydir dumpfile=full_db_export.dmp logfile=full_db_export.log full=yes

Filtering export of database objects
(EXCLUDE and INCLUDE parameters)
To export specific objects you either use INCLUDE to provide a list of objects that will be exported, or EXCLUDE parameter to exclude some object for being exported. Use TABLE_EXPORT_OBJECTS, SCHEMA_EXPORT_OBJECTS and DATABASE_EXPORT_OBJECTS views to get the list of objects that may be passed as a value for both parameters.

SQL> COL object_path FORMAT a30
SQL> COL comments FORMAT a60
SQL> SET LINESIZE 150
SQL> SELECT * FROM database_export_objects WHERE named='Y';

<table>
<thead>
<tr>
<th>OBJECT_PATH</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>----------</td>
</tr>
</tbody>
</table>


4.9. Configure and use parallel execution for queries

Using parallelism a task can be divided into multiple sub tasks and executed in a parallel speeding up the process. To use parallelism feature you should have multiple CPUs and you need to properly configure database parameters related with parallelism, or it will lead to the poor performance.

In Oracle, you can load data, create a database objects, run DML statements, take RMAN backup in a parallel mode. Below, we will show you the main tasks that can be performed in a parallel mode and might be asked during an OCM exam.

Using hints related with parallelism

To speed up the query and get quick result, you can use PARALLEL hint to serialize the output of the SQL command. The following hint will make the query to run in a parallel mode with 8 parallel process:

```
SQL> SET AUTOTRACE TRACEONLY EXP
SQL> SELECT /*+ PARALLEL (tbl_dimension1,8) */ * FROM tbl_dimension1;
```

```
Execution Plan
----------------------------------------------------------
Plan hash value: 1868886941

<table>
<thead>
<tr>
<th>Id</th>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost (%CPU)</th>
<th>Time</th>
<th>TQ</th>
<th>IN-OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQ Distrib</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plain text</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
<td>10</td>
<td>110</td>
<td>2 (0)</td>
<td>00:00:01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PX COORDINATOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PX SEND QC (RANDOM)</td>
<td>:TQ10000</td>
<td>10</td>
<td>110</td>
<td>2 (0)</td>
<td>00:00:01</td>
<td>Q1</td>
<td>P-&gt;S</td>
</tr>
<tr>
<td>3</td>
<td>PX BLOCK ITERATOR</td>
<td></td>
<td>10</td>
<td>110</td>
<td>2 (0)</td>
<td>00:00:01</td>
<td>Q1</td>
<td>PCWC</td>
</tr>
<tr>
<td>4</td>
<td>TABLE ACCESS FULL</td>
<td>TBL_DIMENSION1</td>
<td>10</td>
<td>110</td>
<td>2 (0)</td>
<td>00:00:01</td>
<td>Q1</td>
<td>PCWP</td>
</tr>
</tbody>
</table>
```

Indexes also might be parallelized using PARALLEL_INDEX hint. Create a new table based on the table above with more rows (50k for example), create index on it and use PARALLEL_INDEX hint as follows:

```
SQL> CREATE TABLE tbl_dimension5 AS SELECT * FROM tbl_dimension1;
Table created.
```
Indexes also might be parallelized using PARALLEL_INDEX hint. Create a new table based on the table above with more rows (50k for example), create index on it and use PARALLEL_INDEX hint as follows:

SQL> CREATE TABLE tbl_dimension5 AS SELECT * FROM tbl_dimension1;
Table created.

Run the following command for 10 times to increase count of rows in the table:

SQL> INSERT INTO tbl_dimension5 SELECT * FROM tbl_dimension5;

Now create an index use PARALLEL_INDEX hint as follows:

SQL> CREATE INDEX ttt ON tbl_dimension5 (r_num);
Index created.

SQL> SELECT /*+ PARALLEL_INDEX(TBL_DIMENSION5,TTT,6) */ r_num FROM tbl_dimension5 WHERE r_num<10;

Enabling parallelism during creation of an object
You can assign fixed parallelism degree for the table or the index during creation. Use PARALLEL (DEGREE int) clause at the end of CREATE command. In the following example, we create a table and set a default degree of parallelism (DOP) to 6. It means that anytime a user accesses the table, will get a DOP of 6:

SQL> CREATE TABLE tbl_dop_test AS SELECT * FROM dba_tables;
Table created.

SQL> ALTER TABLE tbl_dop_test PARALLEL (DEGREE 6);
Table altered.

Note
-----
- dynamic sampling used for this statement (level=2)
CHAPTER 5

Data Warehouse Management
In the beginning of this chapter we provide information about main data warehouse topics such as partitioning and managing large objects. We show an example of each partitioning method and explain different partitioning maintenance operations such as adding new partition, dropping and truncating a partition, coalesce, rename and merge partitions. We also provide detailed steps on how to convert a partitioned table to non-partitioned and vice-verse.

Next, we talk about large objects and show how to use various SecureFile LOB features such as compression, deduplication, encryption, caching and logging. We also talk about main DBMS_LOB functions such as INSTR, SUBSTR, COMPARE, LENGTH and etc.

Then we also learn how to use Virtual Private Database (VPD) to prevent a user to access a data of specific tables. Following the information about basics of application context, we show a combination of application context and fine-grained access control and build up a strong row and column level security.

At the end we review the ways a flashback data archive is used (which is also called Total Recall) to store a transactional changes of a table.
Here’s the list of OCM exam topics under “Data Warehouse Management” section that we are going to review:

5.1. **Administer partitioned tables and indexes using appropriate methods and keys**
5.2. **Perform partition maintenance operations**
5.3. **Maintain indexes on a partitioned table**
5.4. **Implement SecureFile LOB**
5.5. **Create and manage LOB segments**
5.6. **Implement fine-grained access control**
5.7. **Oracle Database Auditing**
5.8. **Create and manage contexts**
5.9. **Administer flashback data archive and schema evolution**
5.1. **Administer partitioned tables and indexes using appropriate methods and keys**

I would strongly suggest you to use Enterprise Manager to create and manage partitioned tables during OCM exam. However, you also have to know how to create a partitioned table using command line interface.

During an OCM exam, if you are asked to create a partitioned table, do it with EM. If EM is not available or you are explicitly asked to create the partitioned table using command line interface, check the following documentation to use the syntax.

**Reference:**

Database VLDB and Partitioning Guide -> 4 Partition Administration -> Creating Partitions  
Examples For Creating Partitioned table With Different Partitioning Strategies (Doc ID 733311.1)  
Master Note for Partitioning (Doc ID 1312352.1)  
Information Center: Using Partitioning in the Oracle Database (Doc ID 1551512.2)  
11g Partitioning Enhancements (Doc ID 452447.1)

In this section, we will cover the following partition creation methods:

- Creating Range partitioned table
- Creating List partitioned table
- Creating Hash partitioned table
- Creating Interval partitioned table
- Creating Reference partitioned table
- Creating Composite partitioned table
- Using multicolumn partitioned keys
- Creating Virtual Column based partitioning

**Creating Range partitioned table**

When creating a range-partitioned table, specify a partitioning column and use VALUES LESS THAN clause to define the partition bound.

To create a range partitioned table, open EM, switch to the GRIDDB database and click on Schema link. From the Database objects section, select Tables link and click Create button. Select Standard (Heap Organized) option and click Continue.
Creating Reference partitioned table

With parent-child relationship, a child table now can inherit partitioning key from the parent table without duplicating the key columns.

To get the syntax of reference partitioned table, go to the documentation that is provided in the beginning of this section and move to the “Creating Reference-Partitioned Tables” part.

To see how it works, we need to create one parent and one child table. Create a parent table as follows:

```sql
SQL> CREATE TABLE TBL_REF_PART
     ( id NUMBER PRIMARY KEY,
       name VARCHAR2 (20),
       surname VARCHAR2 (20) )
PARTITION BY RANGE (id)
     (PARTITION p1 VALUES LESS THAN (10), PARTITION p2 VALUES LESS THAN (20));
Table created.
SQL>
```

Now create a child table with foreign key to the parent table and use PARTITION BY REFERENCE syntax to inherit a partitioning from the parent table as follows:

```sql
SQL> CREATE TABLE tbl_ref_part_child
     ( id NUMBER PRIMARY KEY,
       name VARCHAR2 (20),
       CONSTRAINT fk_id FOREIGN KEY (id) REFERENCES tbl_ref_part (id)
     )
PARTITION BY REFERENCE (fk_id);
Table created.
SQL>
```
If you query `DBA_TAB_PARTITIONS` view, you will see the list of partitions that belongs to the child table are created automatically:

```sql
SQL> SET LINESIZE 150
SQL> SELECT table_name, partition_name
FROM dba_tab_partitions
WHERE table_name LIKE 'TBL_REF%';
```

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>PARTITION_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBL_REF_PART</td>
<td>P1</td>
</tr>
<tr>
<td>TBL_REF_PART</td>
<td>P2</td>
</tr>
<tr>
<td>TBL_REF_PART_CHILD</td>
<td>P1</td>
</tr>
<tr>
<td>TBL_REF_PART_CHILD</td>
<td>P2</td>
</tr>
</tbody>
</table>

Now query `DBA_PART_TABLES` view to see the foreign key constraint name on which the partition was created:

```sql
SQL> SELECT table_name, partitioning_type, ref_ptn_constraint_name
FROM dba_part_tables
WHERE table_name LIKE 'TBL_REF%';
```

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>PARTITION</th>
<th>REF_PTN_CONSTRAINT_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBL_REF_PART</td>
<td>RANGE</td>
<td></td>
</tr>
<tr>
<td>TBL_REF_PART_CHILD</td>
<td>REFERENCE</td>
<td>FK_ID</td>
</tr>
</tbody>
</table>

### Creating System partitioned table

By using system partitioning option, you don’t define a specific partitioning key and create many physical partitions. When you insert a row, however, you need to specify the partition which will
5.4. Implement securefile LOB

In this section we will talk about SecureFile LOB (Large Object) segments. Detailed information about creating and managing LOB segments is provided in the next chapter.

SecureFiles is a LOB storage architecture that is used to store unstructured data and contains numerous features as deduplication, compression, encryption and etc. If you want to store a LOB as a SecureFile, use STORE AS SECUREFILE clause in the CREATE TABLE statement. If you omit this clause, the LOB will be stored in a regular traditional way which is called BasicFile.

A SecureFile LOB can be created only in a tablespace managed with Automatic Segment Space Management (ASSM). The following error will appear if you attempt to create a SecureFile LOB in a non-ASSM tablespace:

ORA-43853: SECUREFILE lobs cannot be used in non-ASSM tablespace “SYSTEM”

DB_SECUREFILE parameter is used to control the action of the database with regards to the LOB storage. There are four values for this parameter:

- NEVER is used to make all LOBs that are created as a SecureFile to be created as a BasicFile LOB.
- PERMITTED is a default value that allows LOBs to be created as a SecureFile using STORE AS SECUREFILE clause.
- ALWAYS is used to make all LOBs in a ASSM tablespace be created as a SecureFile LOBs.
- IGNORE is used to ignore store as securefile clause and creation of SecureFile LOBs are permitted.

Note: Make sure the value of parameter DB_SECUREFILE is not set to NEVER and IGNORE if you are asked to create a SecureFile LOB during an exam.
Let's create a simple table with a SecureFile LOB segment. First, create a table with BasicFile LOB as follows:

```
SQL> CREATE TABLE tbl_basicfile_lob
   2  (id number,
   3  lob_data clob)
   4  LOB (lob_data) STORE AS BASICFILE;
Table created.

SQL> INSERT INTO tbl_basicfile_lob VALUES(1,'This is my long text');
1 row created.

SQL> COMMIT;
Commit complete.
```

Now create a table with SecureFile LOB data as follows:

```
SQL> CREATE TABLE tbl_securefile_lob (
   2  id number, lob_data clob)
   3  LOB (lob_data) STORE AS SECUREFILE (tablespace users);
Table created.

SQL> INSERT INTO tbl_securefile_lob VALUES(1,'This is my long text');
1 row created.

SQL> COMMIT;
Commit complete.
```

Query `DBA_LOBS` view to get more information about the lob segments:

```
SQL> SET LINESIZE 150
SQL> COL column_name FORMAT a10
SQL> SELECT table_name, column_name, segment_name, securefile FROM dba_lobs
   2  where table_name like 'TBL%';

<table>
<thead>
<tr>
<th>TABLE_NAME</th>
<th>COLUMN_NAME</th>
<th>SEGMENT_NAME</th>
<th>SEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBL_BASICFILE_LOB</td>
<td>LOB_DATA</td>
<td>SYS_LOB0000096459C00002$$</td>
<td>NO</td>
</tr>
<tr>
<td>TBL_SECUREFILE_LOB</td>
<td>LOB_DATA</td>
<td>SYS_LOB0000096464C00002$$</td>
<td>YES</td>
</tr>
</tbody>
</table>
```

SQL>
Using DBMS_LOB package subprograms

Refer to the following documentation for more information on DBMS_LOB package:

💡 Reference:
Database PL/SQL Packages and Types Reference -> 82 DBMS_LOB

All examples related with DBMS_LOBS subprograms are located under $ORACLE_HOME/rdbms/demo/lob/plsql folder. The whole text of all LOB related examples are not available at Oracle documentation. During OCM exam, you either will be provided with the demo folder provided above, or the data will be loaded with different utilities such as SQL*Load and Data Pump.

Let’s load some data to the previous table and use some subprograms of DBMS_LOB package. Create a text file under /tmp and load it to the table.

[oracle@oemgrid ~]$ more /tmp/lob.dat
This is the first line
This is the second line
This is the third line
[oracle@oemgrid ~]$ 

Refer to the following documentation:

💡 Reference:
Database SecureFiles and Large Objects Developer’s Guide -> 22 Using LOB APIs -> Loading a CLOB or NCLOB with Data from a BFILE

To load a text file created above, you need to check file lldclobf.sql under the examples folder. Here’s the source of the file:

/* This file is installed in the following path when you install */
/* the database: $ORACLE_HOME/rdbms/demo/lob/plsql/lldclobf.sql */

CREATE OR REPLACE PROCEDURE loadCLOB1_proc (dst_loc IN OUT CLOB) IS
src_loc     bfile := bfilename('MEDIA_DIR','monitor_3060.txt') ;
amt         number := dbms_lob.lobmaxsize;
5.9. Administer flashback data archive and schema evolution

Flashback data archive is a logical container for storing historical information with minimal performance impact and in compressed form to minimize storage requirements. It consists of one or more tablespaces, and while creating them, a retention duration shall be specified. To create a flashback data archive you must have FLASHBACK ARCHIVE ADMINISTER system rights. Here are the steps to create a flashback data archive:

1. Create a user and grant necessary privileges:

   SQL> CREATE USER usr_fda IDENTIFIED BY usr_fda;
   User created.

2. Grant FLASHBACK ARCHIVE ADMINISTER right to the user to create and maintain flashback data archive:

   SQL> GRANT CONNECT, RESOURCE, FLASHBACK ARCHIVE ADMINISTER TO usr_fda;
   Grant succeeded.

   SQL> CREATE TABLESPACE tbs_fda DATAFILE '/home/oracle/oradata/tbs_fda01.dbf' SIZE 20M;
   Tablespace created.

3. Create a flashback data archive

   SQL> CREATE FLASHBACK ARCHIVE f1_archive TABLESPACE tbs_fda RETENTION 1 YEAR;
   Flashback archive created.

   SQL> CONN usr_fda/usr_fda
Next, create a table and insert one row:

```sql
SQL> CREATE TABLE tbl_fl_archive (id NUMBER, name VARCHAR2(20));
Table created.

SQL> INSERT INTO tbl_fl_archive VALUES(1,'Flashback Archive');
1 row created.

SQL> COMMIT;
Commit complete.

SQL> SELECT * FROM tbl_fl_archive;

<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Flashback Archive</td>
</tr>
</tbody>
</table>

Enable historical tracking for the table:

```sql
SQL> ALTER TABLE tbl_fl_archive FLASHBACK ARCHIVE fl_archive;
Table altered.

SQL> SELECT TO_CHAR(SYSDATE,'ddmmyyyy hh24:mi:ss') ddate FROM DUAL;

DDATE
--------
17032015 10:10:01
```

Make some changes to the table, create new undo segment, assign it to the database, shutdown and bring the database up to make sure that you query historical data not from UNDO, but from flashback data archive:

```sql
SQL> DELETE FROM tbl_fl_archive;
1 row deleted.

SQL> COMMIT;
Commit complete.
```
CHAPTER 6

Performance Management
Tuning a database and resolving performance issues is one of the essential duties of DBA. This is one of the challenging sections of the exam. In general, OEM Grid Control simplifies a lot of performance related tasks providing user friendly GUI. There are also certain tasks that should be performed from CLI. It is strongly recommended to perform all tasks from CLI even if it can be done from GUI.

In this chapter, we start talking about Resource Manager and its components such as plans and consumer groups and show how to manage the workload among different users. Next, we see how to enable result cache feature and incorporate results of SQL queries and PL/SQL functions in the cache.

Having talked about execution plans, we show different ways of access methods in Oracle such as Full Table Scan and Index Lookup and provide a lot of practical demonstrations on Index Unique Scan, Index Range Scan, Index Skip Scan, Full Index Scan, Fast Full Index Scan, and Index Join methods.

Beside the manual tuning, it also possible to use Oracle Advisors to get different performance related recommendations from Oracle. In this section we provide a very detailed demonstration of using SQL Tuning Advisor, Access Advisor, Performance Analyzer and different memory advisors.
Chapter 6: Performance Management

Here’s the list of OCM exam topics under “Performance Management” section that we are going to review:

6.1. **Administer Resource Manager**
6.2. **Use Result Cache**
6.3. **Use multi column statistics**
6.4. **Gather statistics on a specific table without invalidating cursors**
6.5. **Use partitioned indexes**
6.6. **Administer and tune schema object to support various access methods**
6.7. **Interpret execution plan**
6.8. **Use SQL tuning tools and features**
6.9. **Use SQL Tuning Advisor**
6.10. **Use SQL Access Advisor**
6.11. **Use SQL Performance Analyzer**
6.12. **Configure baseline templates**
6.13. **Use SQL Plan Management feature**
6.14. **Implement instance caging**
6.1. Administer Resource Manager

Reference:

Database Administration -> Database Administrator’s Guide -> 27 Managing Resources with Oracle Database Resource Manager
Master Note: Overview of Oracle Resource Manager and DBMS_RESOURCE_MANAGER (Doc ID 1484302.1)
Resource Manager Enhancements in Oracle Database 11g (Doc ID 884082.1)
Database Resource Manager samples (Doc ID 106948.1)
Resource Manager Enhancements in Oracle Database 11g (Doc ID 884082.1)
Configuring Resource Manager for Mixed Workloads in a Database (Doc ID 1358709.1)

Resource Manager is used to manage a workload and the CPU usage of specific sessions in the database. It is used to control long running queries and prevents them from excessive CPU consumption, to set the priority for specific sessions to use more resources to minimize the response time during specific time range, to make the sessions automatically switch from one group to another based to use difference resource on pre-defined criteria, to limit degree of parallelism for certain users or group of users etc.

Resource Manager can be managed using both Enterprise Manager and command line interface using DBMS_RESOURCE_MANAGER package. The best way to use the resource manager is Enterprise Manager. However, if you failed to install and configure the Grid Control during the exam, the only choice you will have is to use procedure and functions of DBMS_RESOURCE_MANAGER package.

Before providing a practical scenario on resource manager, check the following terminology of its elements:

- Resource plan is used to specify how the resources are to be distributed among the consumer groups. A number of resource plans might be created, but only one will be active at a time.
- Consumer group is a set of sessions that are grouped based on resource requirements. When a session is created in the database, it is automatically mapped to a consumer group. There are two default consumer groups in the database: SYS_GROUP and OTHER_GROUPS.
- Plan directive is used to associate consumer groups with plans.

In the following example we will see how to use resource manager in a practice. So, here’s the task:
6.4. Gather statistics on a specific table without invalidating cursors

When gathering statistics using DBMS_STATS package, NO_INVALIDATE parameter can be used to not invalidate the cursors in the cache automatically. This parameter accepts 3 values: 1) TRUE doesn't invalidate dependent cursors 2) FALSE invalidate dependent cursors 3) AUTO_INVALIDATE which is the default value that lets Oracle decide when to invalidate the cursors. To get the default value of NO_VALIDATE parameter, use GET_PREFS function as follows:

```
SQL> SELECT DBMS_STATS.GET_PREFS( 'NO_VALIDATE' ) FROM dual;

DBMS_STATS.GET_PREFS('NO_VALIDATE')
---------------------------------
DBMS_STATS.AUTO_INVALIDATE

SQL>
```

In the following example NO_INVALIDATE with FALSE option will be used to show how cursors are invalidated in the cache. Let's flush the shared pool, create a new table and gather the statistics:

```
SQL> ALTER SYSTEM FLUSH SHARED_POOL;
System altered.

SQL> CREATE TABLE tbl_invalidation_test AS SELECT * FROM dba_objects;
Table created.

SQL> EXEC DBMS_STATS.GATHER_TABLE_STATS(NULL,'tbl_invalidation_test');
PL/SQL procedure successfully completed.

Now query the table twice and check V$SQL to see count of executions, status of the cursor and count of invalidations:
```
SQL> SELECT COUNT(1) FROM tbl_invalidation_iest;
COUNT(1)
---------
    71972

SQL> /

COUNT(1)
---------
    71972

SQL> SELECT child_number, parse_calls, executions, object_status, invalidations FROM v$sql WHERE sql_text = 'SELECT COUNT(1) FROM tbl_invalidation_test';

<table>
<thead>
<tr>
<th>CHILD_NUMBER</th>
<th>PARSE_CALLS</th>
<th>EXECUTIONS</th>
<th>FIRST_LOAD_TIME</th>
<th>INVALIDATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>2016-07-19/19:09:06</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Now gather the statistics of the table with NO_INVALIDATE=>FALSE which will invalidate the cached cursors, query the table and check V$SQL view again:

SQL> EXEC DBMS_STATS.GATHER_TABLE_STATS(
    NULL,'TBL_INVALIDATION_TEST',
    NO_INVALIDATE => FALSE);

PL/SQL procedure successfully completed.

SQL> SELECT COUNT(1) FROM tbl_invalidation_test;
COUNT(1)
---------
    71972
6.8. Use SQL tuning tools and features

**Reference:**

Database Administration -> Database Performance Tuning Guide -> 1.2 Introduction to Performance Tuning Features and Tools

Oracle provides a lot of different tools and features to optimize and tune the database. Below you can find the list of performance tuning tools and features which should be mastered before entering the exam. Some of this tools are explained in a detail as a separate topic, and some of them are within this topic:

**SQL Tuning Advisor** is used to optimize SQL statement without modifying it. Refer to “Use SQL Tuning Advisor” topic in this section to get more information on this tool.

**SQL Access Advisor** is used to achieve a performance by recommending indexes, materialized views and partitions for a given workload. Refer to “Use SQL Access Advisor” topic in this section to get more information on this tool.

**SQL Performance Analyzer** is used to compare SQL execution results before and after specific changes and provides a report that help you to evaluate and estimate the impact of the change. Refer to “Use SQL Performance Analyzer” topic in this section to get more information.

**SQL Plan Management** is used to make sure the consistent SQL performance by using only accepted execution plans on SQL statements. Refer to “Use SQL Plan Management Feature” topic in this section to get more information.

**Automatic Workload Repository (AWR)** is used to collect and maintain performance statistics of the database.

**Automatic Database Diagnostic Monitor (ADDM)** is used to analyze the AWR data for the performance problems.

**Active Session History (ASH)** gathers sampled data of active sessions and is used to diagnose performance problems.

**Memory Advisors** are used to get recommendations on specific memory structures and provide information for tuning them.

**V$ Performance Views** are updated continuously and maintained by the database and are used to diagnose performance problems.

**Application tracing tools** are used to diagnose performance problems by debugging and tracing a session or the database.
Automatic Workload Repository (AWR)

AWR is used to collect a performance data such as wait events, object statistics, time model statistics, system and session statistics, SQL statements and etc. and is used to by many components of the database. The statistics are available in both memory and database and can be accessed from both SQL*Plus and OEM. AWR takes snapshot of statistics every hour and retention of this historical data is 8 days by default.

The best way to manage AWR is using OEM. Switch to the Server tab and click on Automatic Workload Repository link under Statistics Management section as show in Figure 6.13.

From this page, check the default parameters as shown in Figure 6.14. As you see, the snapshot retention is 8 days and snapshot interval is 1 hour. We have 48 snapshots so far. To see the list of all snapshots, click on Snapshots link. Click on any snapshot to generate an AWR report.
The Automatic Workload Repository is used for storing database statistics that are used for performance tuning.

To change the snapshot retention or collection period, click on Edit button as shown in Figure 6.15.

To generate a report between two snapshots, click Run AWR Report button. Select begin and end snapshots and click “Generate Report” button as shown in Figure 6.16.
### 6.9. Use SQL Tuning Advisor

Oracle automatically runs the SQL Tuning Advisor (STA) on SQL statements with high load from AWR on every night during a default maintenance window and provides performance tuning recommendations. It checks every query for missing or stale statistics and recommends...
to gather statistics for specific objects. It checks access path and can suggest different access path. It also verifies the structure of SQL statements and existence of another execution plans and reports it.

Query DBA_AUTOTASK_CLIENT view from the database to check the status of the STA:

```
SQL> SET LINESIZE 150
SQL> SELECT client_name, status FROM dba_autotask_client;

CLIENT_NAME                                                      STATUS
---------------------------------------------------------------- --------
auto optimizer stats collection                                  ENABLED
auto space advisor                                               ENABLED
sql tuning advisor                                               ENABLED

SQL>
```

Query DBA_AUTOTASK_JOB_HISTORY view to get the history of STA jobs:

```
SQL> COL job_start_time FORMAT a40
SQL> SET LINESIZE 150
SQL> COL job_duration FORMAT a30
SQL>  SELECT job_start_time, job_status, job_duration
FROM DBA_AUTOTASK_JOB_HISTORY
WHERE client_name = 'sql tuning advisor' AND job_start_time >= SYSDATE - 7
ORDER BY 1 DESC;

JOB_START_TIME                           JOB_STATUS                     JOB_DURATION
---------------------------------------- ------------------------------ --------------
28-AUG-16 05.02.20.957986 PM EST5EDT     SUCCEEDED                      +000 00:00:52
27-AUG-16 07.41.26.329823 AM EST5EDT     SUCCEEDED                      +000 00:00:18
25-AUG-16 10.01.00.206832 PM EST5EDT     SUCCEEDED                      +000 00:00:09
24-AUG-16 10.00.59.588990 PM EST5EDT     SUCCEEDED                      +000 00:00:26
23-AUG-16 10.01.00.142949 PM EST5EDT     SUCCEEDED                      +000 00:00:16
22-AUG-16 10.01.00.496620 PM EST5EDT     SUCCEEDED                      +000 00:00:23

6 rows selected.

SQL>
```
To lunch STA, scroll down in Home page of OEM and click on Advisor Central link. Click on SQL Advisors link. Figure 6.34.

<table>
<thead>
<tr>
<th>SQL Advisors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SQL Access Advisor</strong></td>
</tr>
<tr>
<td><strong>SQL Tuning Advisor</strong></td>
</tr>
<tr>
<td><strong>SQL Repair Advisor</strong></td>
</tr>
<tr>
<td><strong>SQL Incident Analysis</strong></td>
</tr>
<tr>
<td><strong>SQL Failure Analysis</strong></td>
</tr>
</tbody>
</table>

**FIGURE 6-34. SQL Advisors**

Click on “Automatic SQL Tuning Results” link to see results of previous automatic tuning tasks. Figure 6.35.

<table>
<thead>
<tr>
<th>Automatic SQL Tuning Result Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task Status</strong></td>
</tr>
<tr>
<td><strong>Automatic Implementation</strong></td>
</tr>
<tr>
<td><strong>SQL Profile</strong></td>
</tr>
</tbody>
</table>

**Summary Time Period**

Choose a time period to focus the graphs and statistics below on a specific range of tuning results. Drill down to view focused results or see the results for all SQLs by clicking the “View Report” button.

**Time Period**

- Begin Date: Aug 18, 2016 21:01:02 PM GMT-04:00
- End Date: Aug 20, 2016 21:01:15 AM GMT-04:00

**Overall Task Statistics**

- Executions: 7
- Candidate SQL: 375
- Distinct SQL Examined: 34

**Number of 1st-QL**

- SQL Profile: 1
- Index: 1
- Statistics: 5
- Restructure SQL Alternative Plan: 4
- Not implemented: 1
- Implemented: 2

**FIGURE 6-35. Automatic SQL Tuning Result Summary**
CHAPTER 7

Grid Infrastructure and ASM
This chapter covers more than enough material to get you started with step by step installation of Oracle Grid Infrastructure. During OCM exam, you will be asked to install the Oracle Clusterware and perform essential maintenance tasks.

It will be necessary to create ASM diskgroups during Grid Installation, and therefore this chapter will also shed light on general Oracle ASM technology and provide information on how to create ASM diskgroups, how to configure failure groups and how to practice disk crash in ASM environment.

Next, we will introduce ACFS file system and will provide steps to create and manage it.

At the end of the chapter we will look at how to manage Oracle Grid Infrastructure using different clusterware tools.
Here's the list of OCM exam topics under “Grid Infrastructure and ASM” section that we are going to review:

7.1. **Install Oracle Grid Infrastructure**

7.2. **Create ASM Disk Groups**

7.3. **Create and manage as ASM instance**

7.4. **Implement ASM failure groups**

7.5. **Creating ACFS File System**

7.6. **Start, stop, configure and administer Oracle Grid Infrastructure**
7.1. Install Oracle Grid Infrastructure

In this section I will show a step by step guide on how to install Grid Infrastructure on the Virtual Machine (VM). As a scenario, two node cluster will be created. You will be asked to proceed to installation directly as both machines will be already configured during OCM exam. I will show you how to create and clone a VM and how to provide necessary network and storage configurations as well as install the Oracle Grid Infrastructure. Let’s get started.

First of all, create a new Virtual Machine (OCM_Node1) and mount the installation of OEL 5.4. Then switch to the Network tab and set the Adapter 1 as a “Bridget Adapter”. Move to “Adapter 2” tab and select “Internal Network”. Click OK and start installing the operating system. Follow steps of installing OEL on VM which was described in detail in the Introduction of the book.

Once OS installation is finished, install VirtualBox Guest Additions and copy the grid installation software to VM. In this scenario, we will install Oracle Grid Infrastructure 11g Release 2 (11.2.0.1). It is publicly available for download at http://www.oracle.com

Before starting the installation, we need to properly configure both network cards. Open the Network utility from System->Administration menu. Select the first device (eth0) and click Edit. Provide the “ocmnode1” as a hostname. Check “Activate device when computer starts” option, click on “Statically set IP addresses” option and provide the following IP addresses:

<table>
<thead>
<tr>
<th>Address:</th>
<th>192.168.0.10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet mask:</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>Default gateway address:</td>
<td>192.168.0.1</td>
</tr>
</tbody>
</table>

FIGURE 7-1. Configuring Network parameters of the first node
Choose “Typical Installation” and click Next.
Click Add button to add the second node (ocmnode2) and provide the virtual ip name (ocmnode2-vip). If you haven’t configured SSH connectivity before, click on “SSH Connectivity” button. Click on Setup button to configure the SSH.

FIGURE 7-5. Specify Cluster Configuration

If SSH is already configured, the following notification will appear:

FIGURE 7-6. Passwordless SSH connectivity notification
If not configured, click on Setup button to configure the SSH connectivity. After it is configured, click Next to proceed with installation.

In the next page, provide the location for Oracle Base and GI software. Choose “Automatic Storage Management” option to store the OCR file, provide a password to connect to the ASM instance with SYSASM privilege and click Next as shown in Figure 7.7.

![Specify Install Locations](image)

**FIGURE 7-7. Specify Install Locations**

Next, provide the name of ASM diskgroup to store clusterware configuration files (OCR and VOTEDISK). If you don’t see list of disks in the Candidate Disks page, click on Change Discovery Path button, provide the path as “/dev/asmdisk” and get list of candidate ASM disks as it shown in Figure 7.8.
FIGURE 7-8. Create ASM Disk Group

Select all disks and create a diskgroup called DATA with normal redundancy option and click Next. Specify Oracle Inventory as “/u01/app/oraInventory” and click Next.

In the next page, make sure all pre-requisites are done and click Next. At the end, check the Summary page and click Finish to start the installation.

When the installation is completed, you will be asked to run two different shell scripts with a root user on both nodes (orainstRoot.sh and root.sh).

If you were able to run both root.sh scripts successfully, the clusterware should be up and running. Run the following command to check the status of the clusterware after the installation:
7.6. Start, stop, configure and administer Oracle Grid Infrastructure

After installing GI, you need to know how to configure and administer GI using different utilities. Below, you can find the list of the main tools that are used to configure and administer the clusterware:

Cluster Ready Services Control (CRSCTL) is used to manually control the Clusterware. This tool is used to start and stop clusterware resources, enable and disable clusterware daemons, debugging clusterware components and etc.

Reference:
Clusterware Administration and Deployment Guide -> E CRSCTL Utility Reference

To get the list of all available commands, run the crsctl -h command:

```
[oracle@ocmnode1 ~]$ crsctl -h
Usage: crsctl add - add a resource, type or other entity
    crsctl check - check a service, resource or other entity
    crsctl config - output autostart configuration
    crsctl debug - obtain or modify debug state
    crsctl delete - delete a resource, type or other entity
    crsctl disable - disable autostart
    crsctl enable - enable autostart
    crsctl get - get an entity value
    crsctl getperm - get entity permissions
    crsctl lsmodules - list debug modules
    crsctl modify - modify a resource, type or other entity
    crsctl query - query service state
    crsctl pin - Pin the nodes in the nodelist
    crsctl relocate - relocate a resource, server or other entity
    crsctl replace - replaces the location of voting files
    crsctl setperm - set entity permissions
    crsctl set - set an entity value
    crsctl start - start a resource, server or other entity
    crsctl status - get status of a resource or other entity
    crsctl stop - stop a resource, server or other entity
```
crsctl unpin     - unpin the nodes in the nodelist
crsctl unset     - unset a entity value, restoring its default

If you want to get more detailed information regarding any parameter from the above output, use –h parameter again:

[oracle@ocmnode1 ~]$ crsctl check -h
Usage:
crsctl check crs
   Check status of OHAS and CRS stack
crsctl check cluster [[-all]|[-n <server> [...]]]
   Check status of CRS stack
crsctl check ctss
   Check status of Cluster Time Synchronization Services
crsctl check resource {<resName> [...]|-w <filter>} [-n <server>] [-k <cid>] [-d <did>]
   Check status of resource(s)
crsctl check css
   Check status of Cluster Synchronization Services

Below, you can find the most used options to configure and manage the clusterware using crsctl command.

To start, stop and get the status of the clusterware stack, run:

[oracle@ocmnode1 ~]$ crsctl start [stop, check] crs

To start, stop and get the status of the OHAs, run:

[oracle@ocmnode1 ~]$ crsctl start [stop, check] cluster
CHAPTER 8

Real Application
Clusters
In this chapter we will explore steps to successfully create a RAC database on two nodes. With only few additional steps, you will successfully create a RAC database. Then we will see the way how to create and configure ASM with command line interface. Once ASM is configured, we will see the silent RAC database creation steps.

Toward the end of the chapter, we will review information about services and server pools in Oracle.
Here’s the list of OCM exam topics under “Data Management” section that we are going to review:

8.1. Install the Oracle Database 11gR2 software
8.2. Configure ASM for the shared disks and create a clustered database
8.3. Configure archiving
8.4. Configure services using both Manual and Policy Managed methods
8.1. Install the Oracle Database 11gR2 software

The introduction part of the book already provided a detailed step by step installation of Oracle 11gR2 Database software on Linux. Most steps of installing and configuring a RAC database are same with installing it on a single instance, but there are some special cases you need to be aware of regarding RAC database creation which will be explained below.

Let’s start the both nodes, copy the Oracle 11g Database software installation files to the first node and extract zip files.

Before starting the database installation, run cluvfy utility to perform pre-requisite checks. If you don’t know the exact syntax, run the following command to get the help of the cluster verify utility:

```
[oracle@ocmnode1 ~]$ cluvfy stage -pre
```

You will get the list of all available parameters related with pre installation/configuration process. From the output, check the line where the “dbinst” is mentioned:

```
cluvfy stage -pre dbinst -n <node_list> [-r {10gR1|10gR2|11gR1|11gR2}]
    [-osdba <osdba_group>]
    [-fixup [-fixupdir <fixup_dir>]] [-verbose]
```

Now, it will be easy for you to run the verification utility before database installation on the RAC environment:

```
[oracle@ocmnode1 ~]$ cluvfy stage -pre dbinst -n ocmnode1,ocmnode2 -verbose
```

If the pre installation check is completed, copy the installation file under /home/oracle/orasoft folder, unzip both files and start the installation.

Choose “Install database software only” option and click Next. From the “Node Selection” page, select “Real Application Clusters database installation” option and make sure to select both nodes as shown in Figure 8.1, then click Next.
FIGURE 8-1.  Node Selection

Provide the following folder as a software location and click Next:

/u01/app/oracle/product/11.2.0/dbhome_1

The remaining steps are the same as in database installation on single node. Click Next and Finish to start the installation. The database software will be installed on both nodes and you will be asked to run root.sh script as shown in Figure 8.2.
Now click OK and finish the installation. In the next section, you will learn how to create a RAC database.

8.2. Configure ASM for the shared disks and create a clustered database

Once the software is installed, you will be asked to create a RAC database. In this section, we will see 2 ways of creating a database: 1) Using DBCA 2) Silent installation

Before getting into database creation, it’s a good idea to create .dbprofile and .gridprofile files under /home/oracle folder on both nodes and set necessary environment variables to easily switch between database and grid environment as we did in the previous chapter. The source of each file is provided below:

vi .dbprofile
export ORACLE_SID=RACDB1
8.4. Configure services using both Manual and Policy Managed methods

Database service is used to manage a workload in a database. From the client side, users connect to the service, not to the database and the service forwards the connection to the appropriate and least loaded instance. Prior to 11g, administrator managed service were used by defining preferred and available instances for a service. Each service is assigned to the instance that is known as a preferred instance. When the preferred instance fails, Oracle automatically move the service to the surviving instance known as available instance.

Configure service using manual method

Before creating any service, check if there is any available service assigned to the RACDBNEW database:

```
[oracle@ocmnode1 ~]$ srvctl status service -d RACDBNEW
```

If you want to find out the list of all parameters when creating a service, run the following command:

```
[oracle@ocmnode1 ~]$ srvctl add service –h
```

Use `srvctl add service` command to create a service and assign both instances as a preferred instance. Set BASIC as a TAF method and a SESSION as a failover type. Set failover method to BASIC.

```
[oracle@ocmnode1 ~]$ srvctl add service -d RACDBNEW -s srv01_racdbnew -r RACDBNEW1,RACDBNEW2 -P BASIC -e SESSION -m BASIC
```
CHAPTER 9

Data Guard
In this section you will see a step by step installation guide of standby database using various methods like OEM Grid Control, SQL*Plus and RMAN. By performing the steps provided in this section you will successfully install and configure a standby database on your local virtual machine. After having configured the standby database you will see how to perform a switchover and failover using SQL*Plus, DGMGRL and OEM.

Then you will see how to convert a standby database to snapshot standby database for read write operations and then convert it back and continue redo apply.
Here's the list of OCM exam topics under "Data Guard" section that we are going to review:

9.1.  **Create Physical Standby Database with real-time apply**

9.2.  **Configure the data guard environment to reduce overheads of fast incremental backups on the primary database**

9.3.  **Configure the Observer**

9.4.  **Switchover and switch back**

9.5.  **Configure connect time failover**

9.6.  **Convert the standby to a snapshot standby**

9.7.  **Configure archivelog deletion policy for the Dataguard configuration**
9.1. Create Physical Standby Database with real-time apply

Reference:
A Standby database is the identical copy of the primary database on a block-for-block basis and is updated by performing a recovery of the redo transactions that are received from the primary database.
There are three different ways to create a Standby database: 1) Manual step by step creation using SQL*Plus 2) Enterprise Manager 3) RMAN

I would highly favor Enterprise Manager to configure a standby database if OEM will be up and running up to the end of the exam. To control the whole process, the step by step manual configuration using SQL*Plus is preferred. In this section, we will cover all methods and see different ways to create and manage a standby database.

Reference:
11.2 Data Guard Physical Standby Switchover Best Practices using SQL*Plus (Doc ID 1304939.1)
Step by Step Guide to Create Physical Standby Using RMAN DUPLICATE (non ASM) on different / new host (Doc ID 374069.1)

✔ Note: During the last month of my preparation for the exam, I was creating and configuring a couple of standby databases every single day. After so much practice, it took only few minutes to create and activate the standby database in my test environment with no failure.

In all following scenarios, we will use two cluster nodes of the previous section (ocmnnode1 and ocmnode2) to create and configure a standby database.

Create a new single instance database (PRODDB) on the first node (ocmnode1) using DBCA. Do not configure OEM, use /home/oracle/oradata folder to store datafiles and enable ARCHIVELOG mode.

Once the database is created on the first node, make sure to change ORACLE_HOME environment variable at .basrc to easily connect to the new database.
Creating Standby database manually using SQL*Plus

Reference:
Step-By-Step Guide To Create Physical Standby On Normal File System For ASM Primary using RMAN (Doc ID 838828.1)

I don’t rely much on GUI, so in my case it was up to me to create a standby database and I used SQL*Plus. You can check the following documentation and follow step by step guide to configure the standby database during an exam.

Reference:
High Availability -> Data Guard Concepts and Administration -> 3 Creating a Physical Standby Database

Now, let’s start creating a standby database.

- Enable force logging to make sure that there will be no statements running with NOLLOGGING option and bypass the redo logs:

```sql
SQL> ALTER DATABASE FORCE LOGGING;
Database altered.

SQL> SELECT force_logging FROM v$database;
FORCE_LOGGING
---
YES

SQL>
```

- Next, you need to change following initialization parameters of the production database:

  - LOG_ARCHIVE_DEST_1 specifies archived log destination of the production database and should have VALID_FOR attribute to specify whether redo data will be written to a destination. The default value is VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
Run the following command on both nodes to find the most recent archived log file. Make sure the output same at both databases:

```
SQL> SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED_LOG
WHERE RESETLOGS_CHANGE# = (SELECT MAX(RESETLOGS_CHANGE#) FROM V$ARCHIVED_LOG)
GROUP BY THREAD#;
```

```
MAX(SEQUENCE#)    THREAD#
-------------- ----------
14          1
```

SQL>

You can use V$ARCHIVE_GAP view to see if there is any redo gap on a standby database. The following query should return no rows:

```
SQL> SELECT * FROM V$ARCHIVE_GAP;
no rows selected
```

SQL>

### Creating Standby database using RMAN

**Reference:**

- Step By Step Guide To Create Physical Standby Database Using RMAN Backup and Restore (Doc ID 469493.1)
- Creating a Standby using RMAN Duplicate (RAC or Non-RAC) (Doc ID 1617946.1)
Starting from 9i, DUPLICATE ... FOR STANDBY command can also be used to create a standby database. Check the following documentation to see the syntax and usage of this command.

**Reference:**
High Availability -> Data Guard Concepts and Administration -> E Creating a Standby Database with Recovery Manager

You have two options while creating a standby database using DUPLICATE command. You either use FROM ACTIVE DATABASE syntax to make RMAN copy the datafiles from production database to standby (without having a backup), or ignore it to make RMAN use backups to create a standby database.

To create a standby database, remove the database that was created above from the second node as follows:

```sql
SQL> STARTUP FORCE MOUNT EXCLUSIVE RESTRICT;

SQL> DROP DATABASE;
Database dropped.

SQL>
```

First of all, let's create a standby database without backup using DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE command.

If the parameter file was dropped along with the database files, create a new parameter file as in the previous example, start the database in NOMOUNT mode, edit the listener.ora file and use static registration of the standby database:

```sql
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = STBDB)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME = STBDB)
    )
  )
```
LISTENER =
   (DESCRIPTION_LIST =
      (DESCRIPTION =
         (ADDRESS = (PROTOCOL = TCP)(HOST = ocmnode2)(PORT = 1521))
      )
      (DESCRIPTION =
         (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
      )
   )

Start the listener on the second node, switch to the first node and connect to both primary and standby databases with RMAN as follows:

[oracle@ocmnode1 ~]$ rman target sys/oracle@PRODDB auxiliary sys/oracle@STBDB

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
connected to target database: PRODDB (DBID=654484567)
connected to auxiliary database: PRODDB (not mounted)
RMAN> DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE;

This command will run the following set of commands automatically:

contents of Memory Script:
{
  backup as copy reuse
targetfile `/u01/app/oracle/product/11.2.0/dbhome_1/dbs/orapwPRODDB' auxiliary format
`/u01/app/oracle/product/11.2.0/dbhome_1/dbs/orapwSTBDB' ;
backup as copy current controlfile for standby auxiliary format `'/home/oracle/oradata/STBDB/control01.ctl';
sql clone 'alter database mount standby database';
set newname for tempfile 1 to
`/home/oracle/oradata/STBDB/temp01.dbf';
switch clone tempfile all;
set newname for datafile 1 to
Chapter 9: Data Guard

Check alert.log file of the standby database:

```
[oracle@ocmnode2 trace]$ tail -f alert_STBDB.log
Archived Log entry 4 added for thread 1 sequence 19 rlc 914436829 ID 0x27028757 dest 2:
    RFS[3]: Opened log for thread 1 sequence 20 dbid 654484567 branch 914436829
    Wed Jun 15 03:04:14 2016
    Media Recovery Log /home/oracle/arch1_19_914436829.dbf
    Media Recovery Waiting for thread 1 sequence 20 (in transit)
Archived Log entry 5 added for thread 1 sequence 20 rlc 914436829 ID 0x27028757 dest 2:
    Media Recovery Log /home/oracle/arch1_20_914436829.dbf
    Media Recovery Waiting for thread 1 sequence 21
    Wed Jun 15 03:04:24 2016
    RFS[3]: Opened log for thread 1 sequence 21 dbid 654484567 branch 914436829
    Log files are being applied successfully.
```

If you want to use RMAN backup files instead of active duplication, take backup of the database with RMAN as it was explained in the first scenario, backup the controlfile for standby and move all backups to the second node. Then use `DUPLICATE TARGET DATABASE ... FOR STANDBY` command that was used above (without `FROM ACTIVE DATABASE` clause) to create a standby database with RMAN as follows:

```
RMAN> DUPLICATE TARGET DATABASE FOR STANDBY DORECOVER;
```

**Creating Standby database using Oracle Enterprise Manager**

The easiest way to create a standby database is using an Enterprise Manager Grid Control. I would strongly suggest you to use OEM if another method is not asked explicitly.

You have to use OEM Grid Control to perform this task. Close both nodes and open the virtual machine where Grid Infrastructure was installed. We will create a standby database on the same machine.
Now open home page of the Grid Control, choose the database (GRIDDB), switch to Availability tab and from Data Guard section choose “Add Standby Database” link as shown in Figure 9.1.

**FIGURE 9-1. Add Standby Database**

From the next page, select “Create a new physical standby database” option as it is shown in Figure 9.2.

**FIGURE 9-2. Create new physical standby database**
Next, Select “Online Backup” and make RMAN copy backups to the required destination as shown in Figure 9.3 and click Next.

Add Standby Database: Backup Type

Data Guard uses Oracle Recovery Manager (RMAN) to create the standby database from a new or existing backup of the primary database. Select the type of backup to use for the standby database creation.

- **Online Backup**
  - Use Recovery Manager (RMAN) to copy database files
    - Staging areas not required. RMAN will copy files directly to destination locations.
  - Copy database files via staging areas
    - Requires staging areas in both primary and standby hosts.

- **Existing Backup**
  - RMAN Backup
    - A whole database backup performed typically as part of regular backup strategy.
  - Backup from a previous standby database creation
    - A backup performed by the Add Standby Database wizard.

FIGURE 9-3. Select backup type to create a standby database

From the next page, provide a degree of parallelism, provide the host credentials of primary database, and click Next as shown in Figure 9.4.

Add Standby Database: Backup Options

<table>
<thead>
<tr>
<th>Primary Database</th>
<th>Primary Host</th>
<th>Degree of Parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRDC01</td>
<td>orcmgr.localdomain</td>
<td></td>
</tr>
</tbody>
</table>

The primary database files will be copied directly to the standby database Oracle Home. No staging areas are required.

Primary Host Credentials

Enter the credentials of the user owns the primary database Oracle server installation

- Username: orcmgr
- Password: orcmgr

Save as Preferred Credential

Primary Database Standby Redo Log Files

Several Data Guard features require standby redo log files. They will be added to the primary database.

FIGURE 9-4. Provide parallelism degree of backup and host credentials of primary database
9.3. Configure the Observer

Before exploring observer, I would like to introduce Data Guard Broker utility (DGMGRL) which is used to manage the data guard in Oracle. During OCM exam, you might be asked to use only DGMGRL for specific tasks related with Data Guard. This tool uses the data guard configuration file to manage both physical and standby databases.

To start the Data Guard Broker process (DMON), set the following parameter on both databases:

SQL> ALTER SYSTEM SET DG_BROKER_START=TRUE;
System altered.
SQL> exit

Run dgmgrl command to enter the command line interface of the broker and connect to the database:

[oracle@ocmnode2 ~]$ dgmgrl
DGMGRL for Linux: Version 11.2.0.1.0 - Production
Copyright (c) 2000, 2009, Oracle. All rights reserved.

Welcome to DGMGRL, type “help” for information.
DGMGRL> CONNECT sys/oracle
Connected.

You can get a list of all commands that are used in DGMGRL utility with detailed explanation of each command.

DGMGRL> help

The following commands are available:
add          Adds a standby database to the broker configuration
connect      Connects to an Oracle database instance
convert      Converts a database from one type to another
create       Creates a broker configuration
disable      Disables a configuration, a database, or fast-start failover
disable      Disables a configuration, a database, or fast-start failover
disable      Disables a configuration, a database, or fast-start failover
disable      Disables a configuration, a database, or fast-start failover
disable      Disables a configuration, a database, or fast-start failover
disable      Disables a configuration, a database, or fast-start failover
edit         Edits a configuration, database, or instance
enable       Enables a configuration, a database, or fast-start failover
exit         Exits the program
failover     Changes a standby database to be the primary database
help         Displays description and syntax for a command
quit         Exits the program
reinstate    Changes a database marked for reinstatement into a viable standby
rem          Comment to be ignored by DGMGRL
remove       Removes a configuration, database, or instance
show         Displays information about a configuration, database, or instance
shutdown     Shuts down a currently running Oracle database instance
start        Starts the fast-start failover observer
startup      Starts an Oracle database instance
stop         Stops the fast-start failover observer
switchover   Switches roles between a primary and standby database

Use "help <command>" to see syntax for individual commands

DGMGRL>

To get more the usage of any command of the broker, use the help command as follows:

DGMGRL> help create

Creates a broker configuration

Syntax:

    CREATE CONFIGURATION <configuration name> AS
    PRIMARY DATABASE IS <database name>
    CONNECT IDENTIFIER IS <connect identifier>;
To create a broker configuration on the primary database, run `CREATE CONFIGURATION` command as follows:

```
DGMGRL> CREATE CONFIGURATION dg_conf AS PRIMARY DATABASE IS proddb CONNECT IDENTIFIER IS proddb;
```

Configuration “dg_conf” created with primary database “stbdb”

Use `add` command to add the standby database to the configuration. You don’t need to check the syntax in the documentation during an exam. Run the `help` command and you will get the usage of the command as follows:

```
DGMGRL> help add

Adds a standby database to the broker configuration

Syntax:

```
ADD DATABASE <database name>
    [AS CONNECT IDENTIFIER IS <connect identifier>]
    [MAINTAINED AS {PHYSICAL|LOGICAL}];
```
```
DGMGRL> ADD DATABASE stbdb AS CONNECT IDENTIFIER IS stbdb MAINTAINED AS PHYSICAL;
```

Database “stbdb” added

To verify the configuration, run the `SHOW CONFIGURATION` command as follows:

```
DGMGRL> show configuration;

Configuration - dg_conf

Protection Mode: MaxPerformance
Databases:
    stbdb - Primary database
    proddb - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
DISABLED
Welcome to DGMGRL, type “help” for information.
DGMGRL> connect sys/oracle
Connected.
DGMGRL> show configuration;

Configuration - dg_conf

Protection Mode: MaxPerformance
Databases:
  proddb - Primary database
  stbdb  - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL> CONVERT DATABASE stbdb to SNAPSHOT STANDBY;
Converting database “stbdb” to a Snapshot Standby database, please wait...
Database “stbdb” converted successfully
DGMGRL> show configuration;

Configuration - dg_conf

Protection Mode: MaxPerformance
Databases:
  proddb - Primary database
  stbdb  - Snapshot standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>

Check alert.log file of the standby database:

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL