**DBMA Notes (Including Old Questions)**

**1. Initialisation Parameter Files**

Q: An Oracle instance is defined by the parameters used to build it in memory. These initialisation parameters can be static or dynamic.

(i) Describe the difference between static and dynamic initialisation parameters.

(ii) Provide examples of SQL commands that you would use to change values of static and dynamic parameters.

The parameters used to build the instance come from Initialisation Parameter file.

|  |  |  |
| --- | --- | --- |
| Initialization  parameter | Dynamic | Static |
|  | - Parameters can be changed while the instance is running  - Changes take effect immediately  -Optionally be written out to the spfile.  - If the change is not saved to the spfile, then the change will only persist until the instance is stopped. | - Parameters cannot be changed for the current instance.  - Must change parameters in the text initialization file or sever parameter file and then restart the database.  - Then it will come into effect at the next startup. |

The initialisation parameters can be changed through Database Control or SQL\*Plus using ALTER SYSTEM command.

(ii) Examples:

**changing dynamic parameter:**

memory: effect immediately, spfile: changed in sever parameter file, both: both memory and spfile

alter system set job\_queue\_processes=10 scope=memory;

alter system set job\_queue\_processes=10 scope=spfile;

alter system set job\_queue\_processes=10 scope=both;

**changing static parameter:**

alter system set log\_buffer= 5242880 scope=spfile;

**2. Database Startup and Shutdown (To memorize)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Startup stages** | **1. NoMount** | **2. Mount** | **3. Open** |
| Instance | - reads initialization parameter files  - allocates SGA memory  - starts background processes  -SQL> startup nomount; | - opens the database control file  - does not open the datafiles  -SQL>alter database mount; | - opens datafiles for database  - opens redo log files  - SQL>alter database open; |

|  |  |  |  |
| --- | --- | --- | --- |
| **Shutdown**  **stages** | **1. Closed** | **2. Dismount** | **3. Instance Stopped** |
|  | - Modified data blocks cached in the SGA are written to disk  - contents of redo log buffers are written on disk  - then redo log buffer is cleared  - a checkpoint is performed  - datafiles and log files are closed. | - instance dismount the database  - update relevant entries in control file to record a clean shutdown  - control files are closed  - database is now closed and dismount | - instance stops the background processes  - de-allocates the shared memory used by SGA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shutdown Modes (server)** | **A** | **I** | **T** | **N** |
| Allows new connections | No | No | No | No |
| Waits until current sessions end | No | No | No | Yes |
| Waits until current transactions end | No | No | Yes | Yes |
| Forces checkpoints and closes files | No | Yes | Yes | Yes |

A- Abort, I –Immediate, T-Transactional, N-Normal

Q. For a DBA, starting up and shutting down of a database is a routine and basic operation. You can start an instance in various modes.

(i) Discuss different startup modes and the situations when each of the modes would be required?

(ii) Explain what is happening with the instance and the database during each stage of the startup.

Q. For a DBA, starting up and shutting down of a database is a routine and basic operation.

(i) What SQL command do you need to run in order to accomplish the database startup and what are startup stages?

(ii) Explain what is happening with the database during each stage of the startup.

**ANS:**

(i) You use the SQL\*Plus STARTUP command to start up an Oracle Database instance.

You must first be **logged into an account that has sysdba or sysoper privileges such as the SYS account.**

You can start an instance in various modes:

**NOMOUNT:** Starts the instance without mounting a database. This does not allow access to the database and usually would be done only for database creation or the re-creation of control files. NOMOUNT mode requires a parameter file and usually is needed for some types of recovery operations, like control file recovery.

-SQL> startup nomount;

**MOUNT**: *Starts the instance and mounts the database, but leaves it closed.*

This state allows for certain DBA activities, but does not allow general access to the database.

MOUNT mode requires the control file and usually is needed for datafile recovery operations.

SQL>alter database mount;

**OPEN**: Starts the instance*, and mounts and opens* the database.

This can be done in unrestricted mode, allowing access to all users, or in restricted mode, allowing access for database administrators only. OPEN mode requires the datafiles and online redo log files.

SQL>alter database open;

(ii) **During the NOMOUNT stage**, Oracle first opens and reads the initialization parameter file (init.ora) to see how the database is configured. After the parameter file is accessed, the memory areas associated with the database instance are allocated and the background processes are started.

**When the startup command enters the MOUNT stage**, it opens and reads the control file to determine the location of the datafiles and online redo log files.

**When the startup command enters the OPEN stage**, it accesses all of the datafiles associated with the database and makes database available for users. You can also start the database in restricted mode. Restricted mode will only allow users with special privileges to access the database (typically DBA’s), even though the database is technically open. Use the *startup restricts* command to open the database in restricted mode.

Q: For a DBA, starting up and shutting down of a database is a routine and basic operation.

(i) Explain what is happening with the database and the instance during each stage of the shutdown.

(ii) Discuss the different shutdown modes. Include in your discussion the restrictions that are applied for each mode as well as your choice of shutdown mode for different situations.

**ANS:**

(i) **First stage is to close the database:** modified data blocks cached in the SGA are written to a disk. The contents of the redo log buffer are written to a disk, and then the redo log buffer is cleared. A checkpoint is performed. Datafiles and log files are closed.

**Second stage is to dismount the database**: the instance dismounts the database and updates relevant entries in the control file to record a clean shutdown. The control file is closed. The database is now closed and dismounted.

**Finally the instance has to stop**: the instance stops the background processes and de-allocates the shared memory used by the SGA.

**(ii) Shutdown modes:**

**ABORT**: server doesn’t allow new connections, doesn’t wait until current sessions end, doesn’t wait until current transactions end and doesn’t force a checkpoint.

**IMMEDIATE**: server doesn’t allow new connections, doesn’t wait until current sessions end, doesn’t wait until current transactions end, but it will force a checkpoint and close files.

**TRANSACTIONAL**: server doesn’t allow new connections, doesn’t wait until current sessions end, but it will wait until current transactions end then it will force a checkpoint and close files.

**NORMAL**: server doesn’t allow new connections, but it will wait until current sessions end, it will wait until current transactions end, then it will force a checkpoint and close files.

**3. Privileges (To memorize)**

User privileges are set in the database by

**- System privileges**

**- Object privileges**

**- Roles (granting system and object privileges)**

**System Privilege**: Enables users **to perform particular actions** in the database. Usually corresponds to the permission to **run DDL** **commands**.

(Create, Alter, Drop, Rename, Truncate, Comment)

**Object Privilege**: Enables users **to access and manipulate a content of a specific object**. Usually corresponds to the permission to **run DML** **commands**. (Select, Insert, Update, Delete, Merge)

|  |  |  |
| --- | --- | --- |
| Privilege | Grant | Revoke |
| System | SQL>Grant create table, create sequence, create view to hr;  SQL>Grant create table, alter table, drop table to hr\_dba **with admin option**; | - Only granted the privilege with admin option  SQL> **Revoke** create table, create sequence, create view **FROM** hr; |
| Object | Grant select on hr.employees to user1; | Revoke select on hr.employees From user1; |









**Q**. Privileges in an Oracle database come in two forms: system privileges and object privileges. Investigate the following statements about system privileges. Identify each statement as correct or not and provide an explanation to support your decision:

1. Only SYS user can grant system privileges.

Statement 1 is incorrect:

System privileges can be granted by any user (not just SYS user) who has been granted the privilege WITH ADMIN OPTION.

2. If a system privilege is revoked from a user, it will **not be revoked** from all users to whom he/she granted it.

Statement 2 is correct:

Revocation of a system privilege does not cascade.

3. CREATE TABLE is a system privilege.

Statement 3 is correct:

Any action that updates the data dictionary is a system privilege.

4. SELECT TABLE is a system privilege.

Statement 4 is incorrect:

SELECT TABLE does not exist as a privilege.

5. SELECT ANY TABLE is a system privilege.

Statement 5 is correct:

The ANY privileges, that grant permissions against objects in every user account in the database, are system privileges.

6. **System privileges** can be granted to a role.

Statement 6 is correct:

System privileges can be granted to a role or a user.

Q: Privileges in an Oracle database come in two forms: system privileges and object privileges. Investigate the following statements about object privileges. Identify each statement as correct or not and provide an explanation to support your decision:

1. Only SYS user can grant object privileges.

Statement 1 is incorrect: system privileges can be granted by any user (not just SYS user) who owns the object or has been granted the privilege WITH GRANT OPTION.

2. If an **object privilege is revoked** from a user, **it will be revoked** from all users to whom he/she granted it.

Statement 2 is correct: revocation of an object privilege **does cascade**.

3. CREATE TABLE is an object privilege.

Statement 3 is incorrect: any action that updates the data dictionary is a system privilege and not an object privilege.

4. SELECT ON REGIONS is an object privilege.

Statement 4 is correct: SELECT ON REGIONS is an object privilege, granting permission to select from the table regions.

5. SELECT ANY TABLE is an object privilege.

Statement 5 is incorrect: the ANY privileges, that grant permissions against objects in every user account in the database, are system privileges and not object privileges.

6. **Object privileges** can be granted to a role.

Statement 6 is correct: object privileges can be granted to a role or to a user.

**4. Transaction**

**Q**: Provide your understanding of a database transaction.

(ii) Discuss the meaning of database transaction principles (ACID).

(iii) Provide examples of how COMMIT and ROLLBACK statements are used to enforce these principles.

**ANS**: Database transaction is **a unit of work performed within a database management system against a database**. Transactions in a database environment have two main purposes:

• **To provide reliable units of work** that allows correct recovery from failures and keep a database consistent even in cases of system failure.

• **To provide isolation between programs accessing a database concurrently.** If this isolation is not provided, the program's outcome is possibly erroneous.

ACID principles of a transaction:

**A** – atomic: transaction should be indivisible

**C** – consistent: database must be left in a consistent state

**I** – isolated: effects of each transaction should be isolated

**D** – durable: changes made to the database should be permanent

Commit enforces these by making the end of the transaction(atomicity), shows all steps are completed (consistency) makes result public (isolation) and makes changes permanent (durability).

Rollback enforces these marking end of the transaction(atomicity), returns data to the last consistent state (consistency), prevents results from being seen (isolation) and returns unwanted changes to previous state (durability).

**5. Read Consistency**

**Q**: Two database users, **db\_user1** and **db\_user2** are making transactions against the **ACCOUNTS** table.

In order to enter new account’s information into the database, db\_user1 begins a transaction at 10:00 A.M. that consists of several INSERT and UPDATE statements against the ACCOUNTS table, which are not supposed to end at least until 10:30 A.M.

At 10:15 A.M. db\_user2 issues a SELECT statement against this table. You find that none of the modifications done by db\_user1 are reflected to db\_user2.

i. To which of the database properties will you attribute the above situation?

ii. Explain the mechanism that facilitates this property and its implementation.

**ANS:**

Read consistency is the property in which one user begins making a few changes in a particular table or relation and before this user can complete these changes, another user starts issuing queries against the same table or relation.

**The second user will not see the changes made by first user until the first user’s transaction is committed.**

**Read consistency is an automatic implementation. It keeps a partial copy of the database in undo segments.**

The read-consistent image is constructed from committed data from the table and old data being changed and not yet committed from the undo segment.

**When an insert, update, or delete operation is made to the database, the Oracle server takes a copy of the data before it is changed and writes it to an undo segment.**

All readers, except the one who issued the change, still see the database as it existed before the changes started; they view the undo segment’s “snapshot” of the data.

**Before changes are committed to the database, only the user who is modifying the data sees the database with the alterations**.

Everyone else sees the snapshot in the undo segment. This guarantees that readers of the data read consistent data that is not currently undergoing change.

**When a DML statement is committed, the change made to the database becomes visible to anyone issuing a select statement after the commit is done.** **The space occupied by the old data in the undo segment file is freed for reuse.**

**If the transaction is rolled back, the changes are undone:**

The original, older version of the data in the undo segment is written back to the table.

All users see the database as it existed before the transaction began.

**6. Data Integrity**

**Q:** Data integrity is a fundamental property of the Relational Database Management System (RDBMS).

(i) Discuss what data integrity is and why it is important to maintain data integrity in the RDBMS.

(ii) Critically evaluate the three primary methods to maintain data integrity in the database. Include advantages and disadvantages of each method. Illustrate your answer with suitable examples.

**ANS:**

Data integrity **means that data in a database adheres to business rules.**

Data must be kept free from corruption, modification or unauthorized disclosure to drive any number of mission-critical business processes with accuracy.

Data integrity refers to the **wholeness or completeness of data during operations involving transfer, storage and retrieval.** It also refers to the preservation of data so that whatever process in undergoes through, it will still remain to be what it has been intended for. In other words, data integrity is the **assurance that data will always be correct, consistent and accessible.**

There are three primary ways in which data integrity can be maintained:

• **Application code,**

**• Database triggers,**

**• Declarative integrity constraints.**

Data integrity in relational database can be achieved by having **careful database planning and design.** A database designer or developer **should use integrity constraints** in order to enforce all business rules that are closely associated with the database. **This can ensure that end users cannot enter invalid information or data consumers can alter data without the right privileges**. And when someone **with the appropriate privileges deletes or alters data**, relationships through keys can be maintained so **no record can be left orphaned.**

**7. Flashback Table, Flashback Database**

**Q:** Working with the company database you accidentally dropped **the table** **CLIENTS**. You have to recover it.

(i) Discuss the best functionality to use to solve this problem?

(ii) Write the SQL statement you will execute **to recover the table**.

(iii) Discuss how this recovery functionality is implemented.

**ANS:** **Flashback technology can be used when a logical corruption occurs in an Oracle database and you need to recover data quickly and easily.**

In previous releases of the Oracle database, if you dropped a table by mistake, you had to recover the database to a prior time to recover the dropped table. This procedure was often time consuming and resulted in loss of work of other transactions. **Starting from Oracle 10g Oracle Database includes the Flashback Drop feature, which you can use to undo the effects of a DROP TABLE statement without having to use point-in-time recovery.**

(ii) **FLASHBACK TABLE** **CLIENTS** **TO BEFORE DROP;**

(iii) **Starting from Oracle 10g, the DROP TABLE statement moves a table or object to the recycle bin and gives to a table and its associated objects system-generated names.**

The recycle bin itself is a data dictionary table that maintains the relationships between the original names of dropped objects and their system-generated names.

**You can query the content of the recycle bin** by using the DBA\_RECYCLEBIN view. A user can view only objects that she or he has access to in the recycle bin. (**SQL>show recyclebin**;)

The RECYCLEBIN initialization parameter is used to control whether the Flashback Drop capability is enabled.

**If the parameter is set to OFF, then dropped tables do not go into the Recycle Bin. If this parameter is set to ON, the dropped tables go into the Recycle Bin and can be recovered.** By default, RECYCLEBIN is set to ON.

SQL>alter database FLASHBACK ON;

SQL>alter database FLASHBACK OFF;

**Q:** Starting from the release 10g, Oracle introduced a new feature called Flashback Technologies.

(i) What do you understand about Flashback Technologies?

(ii) Explain the architecture and components of each flashback feature.

(iii) Provide examples of how you would use each flashback feature to solve particular problems in the database.

(i) Flashback Technology provides a set of new features to view and rewind data back and forth in time. The Flashback features offer the capability to query versions of schema objects, query historical data, perform change analysis or perform self-service repair to recover from logical corruptions while the database is online.

(ii)(iii)

**Flashback Database**

Flashback Database quickly rewinds an Oracle database to a previous time; to correct any problems caused by logical data corruptions or user errors. Flashback Database is like a 'rewind button' for your database.

**Flashback Table**

When a human or application error occurs, you want to be able to restore the state of one or more tables to a point in time before the problem occurred. Flashback Table provides the DBA the ability to recover a table or a set of tables to a specified point in time quickly, easily, and online.

**Flashback Drop**

Flashback Drop provides a safety net when dropping objects in Oracle Database. When a user drops a table, Oracle automatically places it into the Recycle Bin. You can view your dropped tables by querying the new RECYCLEBIN view and also restored from the RECYCLEBIN with the help of flashback before drop command.

**Flashback Query**

Flashback Query provides the ability to view the data as it existed in the past. By default, operations on the database use the most recent committed data available. If you want to query the database as it was at some time in the past, you can do so with the Flashback Query feature. It lets you specify either a time or a system change number (SCN) and query using the committed data from the corresponding time.

**(Note: Read also Flashback Database from Assignment and text book)**

The Flashback Database operation:

- Works like a rewind button for the database

- can be used in cases of logical data corruptions made by users

- can bring a database to an earlier point in time by undoing all the changes that have taken place since that time

To configuring Database for the Flashback, before you can issue the command to Flashback Database:

- The database must

1. be configured for archiving,

2. have FLASHBACK enabled

SQL> ALTER DATABASE ARCHIVELOG;

SQL> ALTER DATABASE FLASHBACK ON;

3. be started in MOUNT mode

SQL> SHUTDOWN IMMEDIATE;

SQL> STARTUP MOUNT;

You can use the SQL FLASHBACK DATABASE command to return the database to a past time or SCN.

SQL> FLASHBACK DATABASE TO TIMESTAMP(SYSDATE-1/24);

SQL> FLASHBACK DATABASE TO SCN 53943;

If you use the TO SCN clause, you must provide a number

- If you specify TO TIMESTAMP, you must provide a time stamp value

- To view changes the database can be opened in read-only mode

- When finished the database must be opened in read/write with the RESETLOGS option

SQL> ALTER DATABASE OPEN RESETLOGS;