DUMPSARENA

Oracle Database Security Administration

Oracle 1z0-116

Version Demo

Total Demo Questions: 12

Total Premium Questions: 129

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

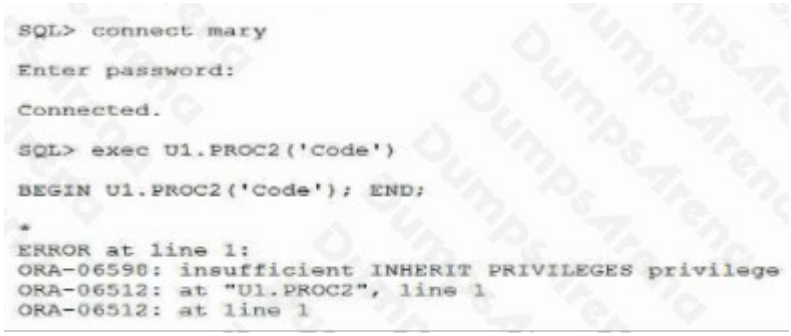
sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Oracle Database Security	31
Topic 2, User Authentication	24
Topic 3, Privilege Management	14
Topic 4, Data Encryption	23
Topic 5, Database Auditing	11
Topic 6, Oracle Database Vault	16
Topic 7, Oracle Label Security	4
Topic 8, Mix Questions	6
Total	129

QUESTION NO: 1

Examine these commands and responses:



Which object privilege must be granted to allow execution of the stored procedure?

A. grant EXECUTE ON MARY.PBOC2 Co U1;

B. grant INHERIT PRIVILEGES ON USER U1 TO MARY;

C. grant INHERIT PRIVILEGES ON USER MARY TO UI ;

D. grant EXECUTE ON UI.PROC2 TO MARY;

ANSWER: B

Explanation:

`grant INHERIT PRIVILEGES ON USER U1 TO MARY;` is required when MARY owns an invoker's-rights stored procedure that U1 executes. An invoker's-rights unit, created with `AUTHID CURRENT_USER`, runs with the privileges of its invoker rather than exclusively with the privileges of its owner. Before Oracle permits the program-unit owner to use the invoker's privileges in this way, the owner must be explicitly authorized to inherit those privileges. This authorization is granted by U1 to MARY using `INHERIT PRIVILEGES ON USER U1`. Once that grant exists, Oracle can execute MARY's invoker's-rights procedure for U1 using U1's applicable privileges, avoiding the insufficient INHERIT PRIVILEGES condition. This control prevents a program-unit owner from unintentionally gaining access through another user's privileges merely because that user invokes the owner's code. Oracle documents this requirement as part of the security model for invoker's-rights program units and the `INHERIT PRIVILEGES` privilege. See the Oracle Database Security Guide: Definer's Rights and Invoker's Rights and the Oracle SQL Language Reference: GRANT.

QUESTION NO: 2

Which profile resource parameter specifies the number of consecutive failed login attempts that are permitted before an Oracle Database account is locked?

A. FAILED_LOGIN_ATTEMPTS

B. PASSWORD_LIFE_TIME

C. PASSWORD_REUSE_MAX

D. PASSWORD_LOCK_TIME

E. SESSIONS_PER_USER

ANSWER: A

Explanation:

FAILED_LOGIN_ATTEMPTS is correct. This password-profile parameter defines the maximum number of consecutive unsuccessful authentication attempts allowed before Oracle locks the account. A successful login resets the failed-login count.

The account remains locked until it is unlocked by an administrator or until the configured PASSWORD_LOCK_TIME interval expires. Oracle documents password resource limits and account-locking behavior in the CREATE PROFILE SQL reference and the Oracle Database Security Guide.

QUESTION NO: 3

You must restrict execution of the alter system checkpoint command to certain conditions, specified in a rule set used by a command rule.

Which two parameters must be specified In the dbms_macadm.create_command_rule procedure to do this?

A. PARAMETER_NAM£=>'CHECKPOINT'

B. OBJECT_OWNER=>'SYS'

C. CLAUSE_NAME=>'CHECKPOINT'

D. CLAUSE_NAME=>'SYSTEM'

E. COMMAND=>'ALTER SYSTEM'

F. OBJECT_NAME=>'CHECKPOINT'

G. COMMAND=>'ALTER'

ANSWER: C E

Explanation:

DBMS_MACADM.CREATE_COMMAND_RULE identifies the SQL command being protected by separating the command text from an optional clause. For the statement ALTER SYSTEM CHECKPOINT, COMMAND=>'ALTER SYSTEM' identifies the base administrative statement, and CLAUSE_NAME=>'CHECKPOINT' identifies the specific CHECKPOINT clause/action to which the command rule applies. Together, these values allow Database Vault to associate the designated rule set with precisely this command form, so execution is permitted only when the rule set evaluates successfully. This is the appropriate level of granularity when the requirement is to control checkpoint execution without necessarily governing every possible ALTER SYSTEM operation. Oracle documents command rules as Database Vault controls that can enforce rule-set authorization for SQL statements, and the CREATE_COMMAND_RULE procedure accepts command and clause_name arguments for this purpose. See the Oracle Database Vault Administrator's Guide: Command Rules and the DBMS_MACADM package reference.

QUESTION NO: 4

You use DBMS_RLS.ADD_POLICY to create Virtual Private Database policies.

Which three are valid values for the policy_type parameter?

A. DBMS_RLS.STATIC

B. DBMS_RLS.CONTEXT_SENSITIVE

C. DBMS_RLS.SHARED_CONTEXT_SENSITIVE

D. DBMS_RLS.SESSION_STATIC

E. DBMS_RLS.USER_SENSITIVE

F. DBMS_RLS.TRANSACTION_DYNAMIC

ANSWER: A B C

Explanation:

STATIC, **CONTEXT_SENSITIVE**, and **SHARED_CONTEXT_SENSITIVE** are valid VPD policy types. A static policy generates its predicate once and can reuse it. A context-sensitive policy reevaluates the predicate when a relevant application-context value changes. A shared context-sensitive policy enables predicate sharing among objects that use the same policy function while still responding to relevant context changes.

Oracle also supports other documented policy types, including dynamic and shared static policies. The policy type controls predicate caching and reevaluation behavior; it does not define the SQL statements to which the policy applies. See the DBMS_RLS package reference and the Oracle Virtual Private Database documentation.

QUESTION NO: 5

A database user must invoke `UTL_HTTP` to access an HTTPS service that requires either an Oracle wallet password credential or a client certificate.

Which two wallet ACL privileges can be granted by using `DBMS_NETWORK_ACL_ADMIN`?

A. use_password_credentials

B. use_client_certificates

C. use_encryption_keys

D. use_keystore_password

E. use_network_credentials

ANSWER: A B

Explanation:

use_password_credentials and **use_client_certificates** are the wallet ACL privileges used for this purpose. The `use_password_credentials` privilege authorizes a database principal to use password credentials stored in an Oracle wallet. The `use_client_certificates` privilege authorizes use of client certificates from the wallet, such as for mutual TLS authentication.

These wallet privileges are separate from host ACL privileges such as `connect` and `resolve`. A user can require both the appropriate host authorization and wallet authorization before an outbound authenticated HTTPS request can succeed. See the DBMS_NETWORK_ACL_ADMIN package reference and the Network ACL administration documentation.

QUESTION NO: 6

You must rekey encrypted sensitive credential data In your database.

You run the command alter database dictionary rekey credentials.

Which three options ate true about the bkkey process?

A. Credential Data Is automatically encrypted using aes2S6.

B. The credential data encryption process does not de-obfuscate the obfuscated passwords before re-encrypts begin.

C. Both sys. links and sys . SCHEDULER_CREDENTIAL tables are rekeyed.

D. The rekey process prompts the user to provide a new key algorithm If needed.

E. The process of rekeylng does not automatically open the keystore.

F. The rekey process only applies to the sys.ltnks CREDENTUIALS table.

G. The rekey process only applies to the SYS.SCHEDULES$ credential table.

ANSWER: A B C

Explanation:

`ALTER DATABASE DICTIONARY REKEY CREDENTIALS` rotates the encryption key used to protect sensitive credential values held in the data dictionary. Credential encryption uses the AES256 algorithm, providing strong symmetric encryption for these protected values. During a rekey operation, Oracle re-encrypts the protected credential representation under a newly generated credential-encryption key; it does not need to de-obfuscate passwords as a preliminary step. This design avoids unnecessarily restoring the legacy obfuscated representation while performing the key rotation.

The command covers both classes of dictionary credentials protected by this feature: database-link credentials stored in `SYS.LINK$` and Scheduler credentials stored in the Scheduler credential dictionary table (commonly documented internally as `SYS.SCHEDULER$_CREDENTIAL`). Consequently, rekeying credentials is not limited to only database links or only Scheduler credentials. The operation is specifically intended to rotate the credential-encryption key while preserving the ability of the database components that use these credentials to continue accessing their protected values. Oracle documents dictionary credential encryption and its rekeying procedure in the Oracle Database Security Guide; the command syntax is also covered in the ALTER DATABASE SQL Reference.

QUESTION NO: 7

Which two statements are true about Database Vault factors?

A. A factor can reference a function In another schema to compute its value, provided execute privilege Is granted to the sys user.

B. Changing a factor type can change how factors are evaluated.

C. You can configure a factor to be evaluated only once per session.

D. You get an error at the time of factor creation If the retrieval method function you have specified does not exist.

E. You can use a factor to enforce conditions for a command rule.

ANSWER: C E

Explanation:

You can configure a factor to be evaluated only once per session. A Database Vault factor has an evaluation option that controls when its retrieval expression is run. Selecting session-based evaluation causes Database Vault to obtain and retain the factor value for the duration of the session, rather than reevaluating it at every access. This is useful when the value is stable for a session and repeated execution of the retrieval method is unnecessary. Oracle documents factor evaluation behavior and the factor configuration attributes in the Oracle Database Vault Administrator's Guide.

You can use a factor to enforce conditions for a command rule. Command rules enforce authorization for protected SQL commands through an associated rule set. A rule in that rule set can obtain a contextual value from a Database Vault factor, such as a session, client, or environment attribute, and use that value in its rule expression to decide whether the command is authorized. This makes factors a reusable source of trusted context for command-rule enforcement. Oracle's Database Vault administration documentation describes using factors in rule expressions and rule sets that are associated with Database Vault controls.

QUESTION NO: 8

Examine these commands that execute successfully:

```
CREATE TABLE employee (
first_name VARCHAR2(128) ENCRYPT ,
last_name VARCHAR2(128) ENCRYPT NO SALT,
empID NUMBER,
salary NUMBER(6) ENCRYPT );

CREATE INDEX ix_employee ON employee(last_name);
```

Examine the execution plan generated for this query:

```
SELECT * FROM employee WHERE last_name='King'

-------------------------------------
| Id  | Operation         | Name     |
-------------------------------------
|   0 | SELECT STATEMENT  |          |
|*  1 |  TABLE ACCESS FULL| EMPLOYEE |
-------------------------------------
```

What must be done to allow the index to be used?

A. Use tublespace encryption instead of column encryption.

B. Add the first_name column to the ix_employee Index to Improve its selectivity.

C. Create a SQL baseline to preserve the execution plan from before the encryption.

D. Enable encryption hardware acceleration on the CPUs of the machine.

E. Encrypt the indexed column using NO SALT encryption.

ANSWER: E

Explanation:

The encrypted indexed column must use deterministic, unsalted Transparent Data Encryption by specifying `NO SALT`. Oracle column encryption uses salting by default. Salting introduces random data into each encrypted value, so two identical plaintext values produce different ciphertext values. This protects against frequency analysis, but it also prevents Oracle from performing the equality comparisons needed to use a conventional B-tree index on the encrypted column. Re-encrypting the column with `NO SALT`, for example through an appropriate `ALTER TABLE ... MODIFY (... ENCRYPT NO SALT)` operation, makes equal plaintext values encrypt consistently. Oracle can then maintain and use an index for predicates against that encrypted value. The security trade-off is important: unsalted encryption is less resistant to frequency analysis, so it should be used only where indexed searching is required and the associated risk is acceptable. Oracle documents that encrypted columns that are indexed must be encrypted without salt and describes the corresponding restrictions and considerations for TDE column encryption. See the Oracle Database Advanced Security Guide: Transparent Data Encryption and the CREATE TABLE SQL Reference.

QUESTION NO: 9

Which type of attack attempts to find data by repeatedly trying similar SQL with a modified predicate?

A. timing attack

B. Inference attack

C. data remanence attack

D. cache attack

E. side-channel attack

F. known-plaintext attack

ANSWER: B

Explanation:

Inference attack is correct. An inference attack obtains sensitive information indirectly by issuing a sequence of closely related queries and varying predicates, then comparing the returned rows, aggregate values, error conditions, or presence and absence of results. Even where direct access to a protected value is restricted, carefully changing conditions in a SQL `WHERE` clause can allow an attacker to infer facts about individual records from the differences between query outcomes. For example, a user might repeatedly refine a predicate involving salary, medical status, or another confidential attribute until the result set reveals whether a particular condition is true. This is an important database-security concern because disclosure can occur through permitted query behavior rather than by directly reading a protected column. Oracle security guidance recognizes inference as a risk when information can be deduced from accessible data and query results, particularly in environments that expose summarized, filtered, or otherwise limited information. Appropriate controls include least-privilege authorization, careful use of views and data redaction, auditing, and policies that prevent users from combining query results to derive restricted facts. Oracle discusses protecting sensitive information and applying access controls in its Database Security Guide; related Oracle Advanced Security documentation is available in the Advanced Security Administrator's Guide.

QUESTION NO: 10

Examine the statement:

CREATE BOLE hr_admin IDENTIFIED USING pac_mgr.hr_admin_rola_ch9ck;

Which three are true about the sec_mgr. hr_admin_role_check procedure?

A. It must use only one security check to validate the user.

B. It must use the invokcr's rights to enable the role,

C. It must use the deflner's rights to enable the role.

D. It can Include one or more security checks to validate the user.

E. It must contain a SET ROLE statement or a DBMS_SESSION.SET_ROLE call.

F. It can use only the DBMS_SESSION. SET_ROLE procedure.

G. Its owner SEC_MGR must be granted the execute any procedure role.

ANSWER: B D F

Explanation:

A secure application role created with `IDENTIFIED USING` is enabled through its associated PL/SQL role-validation routine. “It must use the invokcr's rights to enable the role” is correct because the routine for a secure application role must be an invoker’s-rights program unit, allowing role activation to be evaluated in the context of the session requesting it. “It can Include one or more security checks to validate the user” is also correct: the routine can implement whatever combination of authorization controls the application requires, such as verifying application context values, client identity, network attributes, time restrictions, or other business-specific conditions before enabling the role.

“It can use only the DBMS_SESSION. SET_ROLE procedure” is correct because the authorized PL/SQL role-validation routine enables the secure application role by calling `DBMS_SESSION.SET_ROLE`. This ties role activation to execution of the procedure specified in the role definition and permits the procedure to perform its validation before enabling the role. Oracle documents secure application roles and the `IDENTIFIED USING` clause in the CREATE ROLE SQL Reference; implementation guidance for authorization and secure application roles is provided in the Oracle Database Security Guide.

QUESTION NO: 11

Examine this command:

```
emcli list_masking_definitions -definition_name=credit -target_name=test%
```

What masking definitions does it list?

A. All with the name credit and the commands to deploy them on all databases with names starting with test.

B. All with names starting with credit and created on databases with names starting with test.

C. All with the name credit and created on databases with name starting with teat.

D. All with names starting with credit and created on any database.

E. All with the name credit and the commands to deploy them on a database with the name test.

ANSWER: C

Explanation:

All with the name credit and created on databases with name starting with teat. The `list_masking_definitions` EM CLI verb lists masking definitions stored in Oracle Enterprise Manager. Its definition-name filter is supplied as `credit`, without a wildcard character, so the command is restricted to definitions whose name is exactly `credit`. The database-target filter includes the trailing wildcard pattern `teat%`. In this context, the percent sign matches any remaining characters, limiting the results to definitions associated with database targets whose names begin with `teat`. The command is a listing operation: it returns the metadata for matching masking definitions and their associated database context; it does not generate deployment commands. This distinction is important when automating Data Masking and Subsetting administration with EM CLI, because definition discovery and deployment are handled by separate operations. Oracle's Enterprise Manager documentation describes EM CLI support for Data Masking operations and the management of masking definitions. See the Oracle Enterprise Manager Data Masking documentation and the Oracle Enterprise Manager Command Line Interface documentation.

QUESTION NO: 12

You develop an HR application that must allow multiple sessions to share application attributes. Which statement is executed while implementing the requirement?

A. CREATE CONTEXT global_hr USING hr_pkg;

B. CREATE CONTEXT global_hr USING hr_pkg ACCESSED GLOBALLY;

C. CREATE CONTEXT global_hr USING hr_pkg INITIALIZED GLOBALLY;

D. CREATE CONTEXT global_hr USING hr_pkg INITIALIZED EXTERNALLY;

ANSWER: B

Explanation:

`CREATE CONTEXT global_hr USING hr_pkg ACCESSED GLOBALLY;` is correct because an application context created with the `ACCESSED GLOBALLY` clause is a globally accessed application context. Its attribute values are stored centrally by Oracle and can be shared by multiple database sessions, rather than being isolated in the user global area of one session. This is appropriate when an HR application needs common application attributes to be available across sessions, typically with values associated with a client identifier. The trusted package named in the `USING hr_pkg` clause is the package authorized to set context values through `DBMS_SESSION.SET_CONTEXT`; this preserves controlled management of the shared attributes. Oracle documents that globally accessed contexts are intended for situations where multiple sessions must access the same context values, and identifies `ACCESSED GLOBALLY` as the required `CREATE CONTEXT` clause for this context type. See Oracle's Application Contexts documentation and the CREATE CONTEXT SQL reference.