**DBMA Notes (Including Old Questions)**

**Create Statements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table | View | Tablespace | Role | Profile |
| Physical  Table | A **virtual table** **constructed from other tables or views.**  A view has no data of its own, but rather depends on the data in tables or other views. | Database is **divided into logical storage units called tablespaces**, which group related logical structures (such as tables, views, and other database objects).  Tablespaces provide a means to physically locate data (data files) on storage as well as a unit of backup and recovery. | Roles are **named groups of related privileges.**  You create roles, grant system and object privileges to the roles.  And then grant roles to users.  You can also grant roles to other roles. | A profile is **a named set of resource limits and password parameters** that restrict database usage and instance resources for a user.  Each user can have only one profile, and creating a new one supersedes an earlier version. |

**1. Create Table**

Q. Create table **locations** in **db\_user44** schema with the structure described in the table below. The table should be stored in the tablespace **EXAMPLE**. **Make sure all constraints have user-defined names.**



**Ans.**

CREATE TABLE **db\_user44.LOCATIONS**

(LOCATION\_ID NUMBER CONSTRAINT LOC\_ID\_PK PRIMARY KEY,

STREET\_ADDRESS VARCHAR2(50),

CITY VARCHAR2(30) CONSTRAINT LOC\_CITY\_NN NOT NULL,

COUNTRY\_ID CHAR(2) CONSTRAINT LOC\_C\_ID\_FK

REFERENCES COUNTRIES (COUNTRY\_ID))

TABLESPACE EXAMPLE;

**Q.** Provide an explanation for every step.

**Ans:** table name “location” in “db\_user44” is created with the statement “CREATE TABLE db\_user44.LOCATIONS”, using schema.table naming database object.

- location\_id is defined by LOC\_ID\_PK user defined primary key constraint and defined on a column level. Datatype is used by NUMBER.

- Datatype of street\_address is VARCHAR2(50). Datatype of city is used by VARCHAR2(30)and it is defined by LOC\_CITY\_NN user defined NOT NULL constraint and it is defined **only on** a column level.

- CONTRY\_ID is a foreign key from country table that belongs to the same user db\_user44. It is defined by (CONSTRAINT LOC\_C\_ID\_FK) user defined column level foreign key constraint written as REFERENCES COUNTRIES (COUNTRY\_ID).

- In order to store the tablespace named EXAMPLE, **“**TABLESPACE EXAMPLE” is written at the end of the create table statement.

// Note: You can write using table level constraint on both primary key and foreign key as follows

CREATE TABLE db\_user44.LOCATIONS

(LOCATION\_ID NUMBER,

STREET\_ADDRESS VARCHAR2(50),

CITY VARCHAR2(30) CONSTRAINT LOC\_CITY\_NN NOT NULL,

COUNTRY\_ID CHAR(2),

**CONSTRAINT LOC\_ID\_PK PRIMARY KEY (LOCATION\_ID),**

**CONSTRAINT LOC\_C\_ D\_FK FOREIGN KEY REFERENCES COUNTRIES** (COUNTRY\_ID))

TABLESPACE EXAMPLE;

// Note: you can write “NOT NULL and CHECK” constraints as column or table level constraints

Table level constraint ဆိုတာ filed အားလံုးေရးျပီးမွ ေအာက္မွာ သတ္သတ္ constraint ကို ေရးတာ။ Conlumn level က field define လုပ္တုန္း တစ္ခါထည္း ေဘးမွာေရးတာ။

Q. Create a table students in tutor1 schema with the structure described in the table below. Make sure all constraints have user-defined names.



**Ans.**

CREATE TABLE **tutor1.students**(

student\_id NUMBER CONSTRAINT st\_id\_PK PRIMARY KEY,

last\_name VARCHAR2(60) CONSTRAINT st\_lname\_NN NOT NULL,

first\_name VARCHAR2(30) CONSTRAINT st\_fname\_NN NOT NULL,

gender CHAR(1) CONSTRAINT st\_gen\_CK CHECK(gender IN (‘F’,’M’))

);

**Q.** Provide an explanation for every step.

**ANS:** Explanation:

- table name “students” in “tutor1” is created with the statement “CREATE TABLE tutor1.students”, using schema.table naming database object.

- student\_id is defined by CONSTRAINT st\_id\_PK user defined primary key constraint and defined on a column level. Datatype is used by NUMBER.

- Datatype of last\_name is VARCHAR2(60). It is defined by st\_lname\_NN user defined NOT NULL constraint and it is defined only on a column level. Datatype of first\_name is VARCHAR2(30). It is defined by st\_fname\_NN user defined NOT NULL constraint and it is defined only on a column level.

- Gender of a student data should only accept ‘M’ for male and ‘F’ for female. So, we use char datatpe as CHAR(1) and defined by CONSTRAINT st\_gen\_CK user defined check constraint CHECK(gender IN (‘F’,’M’)).

Q. Write SQL statement to create a table books in db\_user1 schema with the structure described in the table below. Make sure all constraints have user-defined names.



**ANS:**

CREATE TABLE db\_user1.books (

isdn VARCHAR2(20) CONSTRAINT book\_isdn\_PK PRIMARY KEY,

title VARCHAR2(60) CONSTRAINT book\_title\_NN NOT NULL,

author VARCHAR2(50) CONSTRAINT book\_auth\_NN NOT NULL,

available CHAR(1) CONSTRAINT book\_av\_CK CHECK(available IN (‘Y’,’N’))

);

Explanation: See above examples.

**Q**. Create a table oe\_product in oe\_mgr schema with the structure described in the table below. Make sure all constraints have user- defined names. Provide explanation for every step.

****

**Try yourself.**

**2. Create View**

**Q:**

(i) Describe the concept of a database view.

(ii) Discuss the purposes that database view may serve in a DB application.

(ii) Illustrate your answer with suitable examples and SQL commands.

**ANS:**

**(i)**

**A SQL view is a virtual table constructed from other tables or views. A view has no data of its own, but rather depends on the data in tables or other views.**

**Once the view is created, it can be used in the FROM clause of a SELECT statement as though it is a table.**

(ii)

Views can be used for many purposes:

**Views can be used to hide columns. This is done to simplify results or to prevent the display of sensitive data.**

Views can be used to hide rows by providing a WHERE clause in the view definition.

**Views can be used to show the results of computed columns without requiring the user to enter the computation expression.**

Views can be used to hide complicated SQL syntax.

**Views can be used to layer built-in functions. For example, you can construct a view that computes a variable and then write an SQL statement that uses that view that uses the computed variable in a WHERE clause.**

Views can be used to isolate source data tables from application code.

Views can be used to give different sets of processing permissions to the same table.

Views can be used to enable the definition of multiple sets of triggers on the same source data.

(iii) Example,

CREATE OR REPLACE VIEW dept\_50

AS

SELECT e.employee\_id, e.first\_name, e.last\_name, e. salary,

j.job\_title

FROM employees e, jobs j

WHERE employees.job\_id=jobs.job\_id

AND deparment\_id=50;

View can be used as follow,

SELECT \* FROM dept\_50;



Create a database **view dept\_50** that will **display employee’s id, employee’s first and last names, salary and job title** for all **employees in department 50**. Provide explanation and examples of how the view can be used.

**ANS:** The view can be created by executing the following SQL statement:

CREATE OR REPLACE VIEW dept\_50

AS

SELECT e.employee\_id, e.first\_name, e.last\_name, e.salary,

j.job\_title

FROM employees e JOIN jobs j

ON (e.job\_id=j.job\_id)

WHERE e.department\_id=50;

Or

CREATE OR REPLACE VIEW dept\_50

AS

SELECT e.employee\_id, e.first\_name, e.last\_name, e. salary,

j.job\_title

FROM employees e, jobs j

WHERE employees.job\_id=jobs.job\_id

AND deparment\_id=50;

အေပၚက ၾကိဳက္တဲ႔ပံုစံတစ္ခုေရး

Explanation:

To create view name dept\_50 we use sql statement as CREATE OR REPLACE VIEW dept\_50. To display employee’s id, employee’s first and last names, salary and job title from department id 50, we have used

SELECT e.employee\_id, e.first\_name, e.last\_name, e.salary,

j.job\_title

FROM employees e JOIN jobs j

ON (e.job\_id=j.job\_id)

WHERE e.department\_id=50;

In which job id in employees table and jobs table must be same for each employee’s record. This sql select query is connected with “AS” keyword to create view statement.

View can be used as follow,

SELECT \* FROM dept\_50;

or

SELECT employee\_id

FROM dept\_50;

**Q:** Write SQL statement to create **dept\_stats** view in the **AGREEN Schema** that will select information from hr.departments and hr.employees tables and will display each department.

Department name in department table,

Number of employees (employee id in employees table) in that department,

Average salary (salary in employee table) in that department,

Maximum salary (salary in employee table),

And minimum salary (salary in employee table) in that department.

**ANS:**

CREATE OR REPLACE VIEW AGREEN. **dept\_stats**

AS

SELECT D.DEPARTMENT\_NAME AS DEPARTMENT\_NAME,

COUNT (E.EMPLOYEE\_ID) AS STAFF,

AVG(E.SALARY) AS AVG\_SALARY,

MAX(E.SALARY) AS MAX\_SALARY,

MIN(E.SALARY) AS MIN\_SALARY

FROM HR.DEPARTMENTS D, HR.EMPLOYEES E

WHERE D.DEPARTMENT\_ID=E.DEPARTMENT\_ID

AND D.DEPARTMENT\_NAME IS NOT NULL

GROUP BY D. DEPARTMENT\_ID, D.DEPARTMENT\_NAME

ORDER BY D.DEPARTMENT\_NAME;

**Or**

CREATE OR REPLACE VIEW AGREEN. **dept\_stats**

AS

SELECT D.DEPARTMENT\_NAME AS DEPARTMENT\_NAME,

COUNT (E.EMPLOYEE\_ID) AS STAFF,

AVG(E.SALARY) AS AVG\_SALARY,

MAX(E.SALARY) AS MAX\_SALARY,

MIN(E.SALARY) AS MIN\_SALARY

FROM HR.DEPARTMENTS D RIGHT OUTER JOIN HR.EMPLOYEES E

ON D.DEPARTMENT\_ID=E.DEPARTMENT\_ID

WHERE D.DEPARTMENT\_NAME IS NOT NULL

GROUP BY D. DEPARTMENT\_ID, D.DEPARTMENT\_NAME

ORDER BY D.DEPARTMENT\_NAME;

//see explanation above

**3. Create Tablespace**

**Q:** (i) Explain what a tablespace is in an Oracle database and its purpose?

(ii) Critically evaluate the different tablespace types?

(iii) Which SQL command do you need to use in order to create a tablespace? Provide an example.

**ANS:**

(i) A **database is divided into logical storage units called tablespaces,** **which group related logical structures (such as tables, views, and other database objects)**.

**Tablespaces provide a means to physically locate data on storage as well as a unit of backup and recovery.**

A tablespace is a logical storage unit within an Oracle database. A tablespace consists of one or more segments.

(ii)There are three types of tablespaces in Oracle:

• Permanent tablespaces

• Undo tablespaces

• Temporary tablespaces

(iii)

create tablespace data\_tblsp

datafile '/home/oracle/db/ora11/data.dbf'

size 10M

autoextend on maxsize 200M

extent management local uniform size 64K;

**Q:** Write SQL Statement to create a permanent tablespace INVENTORY, size 70MB with one data file invo1.dbf. Space management within this table space should be l**ocally** managed and automatically extensdable by **10 MB** to a **maximum size unlimited**. Make sure that the database manages extents in this tablespace automatically. Provide explanation.

**ANS:**

CREATE TABLESPACE "INVENTORY"

DATAFILE 'inv01.dbf'

SIZE 70M

AUTOEXTEND ON MAXSIZE 120M

LOGGING EXTENT MANAGEMENT LOCAL

SEGMENT SPACE MANAGEMENT AUTO

**4. Create User**

**Q:**

Using SQL commands create a user account with the following requirements:

1. Username = > **db75**, password => **!first\_passw**.

2. Default tablespace => **USERS**, temporary tablespace => **TEMP**.

3. Set up quotas on default tablespace to **unlimited**.

4. Make sure the user can log on to the database.

5. Make sure the user can create a table

Provide explanation for every step.

**ANS:**

1)

CREATE USER db75 IDENTIFIED BY !first\_passw

DEFAULT TABLESPACE USERS

TEMPORARY TABLESPACE TEMP

QUOTA unlimited ON USERS;

2)

GRANT CREATE SESSION, CREATE TABLE to db75;

Explanation:

Specified requirements can be implemented through two SQL statements.

First the user needs to be created with the CREATE USER command.

For creating default table space name “USERS”, “DEFAULT TABLESPACE USERS” is used. For temporary table space “TEMP”, TEMPORARY TABLESPACE **TEMP** is used. Default tablespace USERS can be unlimited by writing “QUOTA unlimited ON **USERS”** statement.

Then CREATE SESSION and CREATE TABLE system privilege has to be granted to the user to be able to log in and creating tables.

**Q:**

Creating and managing user accounts are the most common responsibilities of the DBA.

(i) Using SQL commands create a user account with the following requirements.

1. Username = > hr\_mgr1, password => !4mgr\_pass.

2. Default tablespace => users, temporary tablespace => temp.

3. Set up quotas on default tablespace to 10MB.

4. Make sure the user can log on to the database.

5. Make sure the user can create tables and views.

Provide explanation.

**ANS:**

1)

CREATE USER hr\_mgr1 IDENTIFIED by !4mgr\_pass

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp

QUOTA 10M ON users;

2)

GRANT CREATE SESSION TO hr\_mgr1;

3)

GRANT CREATE TABLE, CREATE VIEW TO hr\_mgr1;

**Explanation**

Specified requirements can be implemented through three SQL statements.

First the user needs to be created with the CREATE USER command.

Then CREATE SESSION **system** privilege needs to granted to the user, so that he/she can log on to the database.

Finally CREATE TABLE and CREATE VIEW **system** privileges must be granted be granted to the user , so that the user may create a table or a view.

**Q:** Using SQL commands create a user account with the following requirements.

1. Username = > db\_user1, password => !first\_passw.

2. Default tablespace => users, temporary tablespace => temp.

3. Set up quotas on default tablespace to 4MB.

4. Make sure the user can log on to the database.

5. Make sure the user can **select data** from the table ORDERS that belongs to the user oe.

ii. Provide an explanation for every step.

**ANS:**

1)

CREATE USER db\_user1 IDENTIFIED by !first\_passw

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp

QUOTA 4M ON users;

2)

GRANT CREATE SESSION TO db\_user1;

3)

GRANT SELECT ON oe.orders TO db\_user1;

Explanation:

Specified requirements can be implemented through three SQL statements. First the user needs to be created with the CREATE USER command.

Then CREATE SESSION **system privilege** needs to granted to the user, so that he/she can log on to the database.

Finally SELECT ON orders **object privilege** must be granted be granted to the user, so that the user may execute SELECT command against it.

**Q**:

Creating and managing user accounts are the most common responsibilities of the DBA. Using SQL commands create a user account with the following requirements. Provide explanation for every step.

1. Username = > scott, password => Hello123.

2. Default tablespace => users, temporary tablespace => temp.

3. Set up quotas on default tablespace to unlimited

4. Make sure the user can log on to the database.

5. Make sure the user can create a table.

**ANS:**

CREATE USER scott identified by Hello123

default tablespace users

temporary tablespace temp

quota unlimited on users;

GRANT CREATE SESSION to scott;

GRANT CREATE TABLE to scott;

**Explanation:**

Specified requirements can be implemented through two SQL statements. First the user needs to be created with the CREATE USER command and then CREATE SESSION system privilege has to be assigned to the user and CREATE TABLE system privilege.

**Q:** Creating and managing user accounts are the most common responsibilities of the DBA. Using SQL commands create a user account with the following requirements. Provide explanation for every step.

1. Username = > oe\_mgr, password => oe\_change\_me.

2. Default tablespace => users, temporary tablespace => temp.

3. Set up quotas on default tablespace to unlimited.

4. Make sure the user can log on to the database.

5. Make sure the user can create a table.

**ANS: Try yourself**

**5. Creating Role**

**Q:** Security in an Oracle database can be managed in a number of ways including the use of roles.

(i) Describe the process of creating and managing roles including SQL commands.

(ii) Critically discuss the benefits of using roles. Illustrate your answer with suitable examples and SQL commands.

**ANS:**

Creating roles is one of the methods to implement data authorisation in an Oracle database.

(i) Managing privileges is made easier **by using roles, which are named groups of related privileges.**

**You create roles, grant system and object privileges to the roles, and then grant roles to users.** **You can also grant roles to other roles**.

Unlike schema objects, roles are not contained in any schema.

In order to create a role the following SQL command should be used:

**CREATE ROLE role\_name;**

In order to create a secure role the following SQL command should be used:

**CREATE ROLE role\_name IDENTIFIED BY password;**

(ii) Some of the benefits of using roles include: **roles can have sets of privileges grouped by job function.**

Roles can **indirectly assign privileges to individual users, roles can be granted to users or other roles. A user can have multiple roles.**

Roles can be enabled by default, by password or selectively.

Roles provide better and easier administration.

//Note --creating role and assigning role to user example

SQL> CREATE ROLE clerk;

SQL> GRANT CREATE TABLE to clerk;

SQL> GRANT SELECT ON oe.ORDERS to clerk;

SQL> GRANT clerk to agreen;

**6. Creating Profile**

**Q:**

A user account has a number of attributes defined at account creation time. User profile is one of these attributes.

i. Describe the concept of a user profile, its purposes and uses.

ii. Provide an example of a SQL code for creating a user profile and assigning it to a user.

**ANS:**

(i) A profile **is a named set of resource limits and password parameters that restrict database usage and instance resources for a user**.

**Each user can have only one profile, and creating a new one supersedes an earlier version.**

The DBA can enforce a limit on resource utilization using resource limit parameters. Also the DBA can maintain database security by using password management feature.

Profiles **only take effect when resource limits are "turned on"** for the database as a whole.

**Profiles restrict users from performing operations that exceed reasonable resource utilization**. Examples of resources that need to be managed:

• Disk storage space.

• I/O bandwidth to run queries.

• CPU power.

• Connect time.

• CPU time

Profiles also enforce password practices – how user passwords are created, reused, and validated.

(ii)

**CREATE PROFILE** **new\_profile**

**LIMIT**

FAILED\_LOGIN\_ATTEMPTS 3

PASSWORD\_LOCK\_TIME 1

SESSIONS\_PER\_USER 2

IDLE\_TIME 15;

This profile will allow only 2 sessions to be opened concurrently per user and 15 min of idle time.

Also the user with this profile will be able to enter the password incorrectly 3 times consecutively before the account will be locked for 1 day.

In order to assign a profile to a user, the DBA can use the following command:

ALTER USER db\_user1 PROFILE new\_profile;

// NOTE profile usage

SQL> CREATE USER jsmith identified by pass4u

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp

PROFILE **new\_profile**

ACCOUNT UNLOCK;