ORACLE DBMS

Lab sheet 3
Database access privileges

Oracle allows multiple users to securely work on the same database. Two commands are particularly important: GRANT and REVOKE, which allow defining the rights of each user on the objects in the database. Every user accesses the database with his username and password. It is the username that determines the access rights to the objects in the database. In the previous tutorials, you worked alone in a schema with a name equal to your username. We are going to check that the DBMS manages the competition of access to objects in the database between several different users. Any user who creates objects is the owner of these objects. The creator of an object can decide to grant or revoke certain access rights to any user.

GRANT privilege [ON table/view] TO username [WITH GRANT OPTION]
This command allows giving the concerned privilege on the table or view to the user.
A user who has granted a privilege can take it back at any time by using the revoke command:
REVOKE privilege ON [table/view] FROM username

The privileges that can be given are the following:
***SELECT** right to read, **INSERT** right to insert rows, **UPDATE** right to update rows, **DELETE** right to delete rows,*
***ALTER** right to modify table definition, **INDEX** right to create index, **ALL** all the above rights.*
*A user who has received a privilege with the option **WITH GRANT OPTION** can pass it on to another user (User3).*

Questions:

1. Create another user: Admin.
2. Log in with this user. What do you notice?
3. Give the right to create a session for this user (Create Session).
4. Give the following privileges to Admin: create tables, users. Log in with Admin and check.
5. Run the query Q1: Select * from DBAINTERVENTION.EMPLOYEE. What do you notice?
6. Give this user read permission for the EMPLOYEE table. Run query Q1 now.
7. The intervention center increases by 3000DA the salaries of employees whose total number of interventions is greater than 5. What should be done? What do you notice?
8. Try again after giving this user update rights for the EMPLOYEE table and read rights for the PARTICIPANTS table.
9. Create a LASTNAMEEMP_IX index on the LASTNAMEEMP attribute of the EMPLOYEE table. What do you notice?
10. Give the index creation rights to Admin for the EMPLOYEE table and try to create the index again. What happens?
11. Revoke the previously granted privileges.
12. Verify that the privileges have been revoked.
13. Create a profile "Interv_Profile" which is characterized by: 3 simultaneous sessions allowed, a system call cannot consume more than 35 seconds of CPU, each session cannot exceed 90 minutes, a system call cannot read more than 1200 blocks of data in memory and on the disk. Each session cannot allocate more than 25 KB of memory in SGA. For each session, a maximum of 30 minutes of inactivity is allowed. 5 login attempts before the account is blocked. The password is valid for 50 days, and it will take 40 days before it can be used again. Only one day of access ban after the 5 login attempts have been reached. The grace period that extends the use of the password before changing it is 5 days.
14. Assign this profile to the Admin user.
15. Create the role: "INTERVENTION_MANAGER" who can see the tables EMPLOYEE, VEHICLE, CUSTOMER and can modify the tables INTERVENTIONS and PARTICIPANTS.
16. Assign this role to Admin. Check that the authorizations assigned to the role INTERVENTION_MANAGER have been transferred to the Admin user.

The SQL syntax for creating a user

CREATE USER username
IDENTIFIED {BY password | EXTERNALLY | GLOBALLY AS 'externalname'}
[DEFAULT TABLESPACE tablespaceName [QUOTA {integer [K | M] | UNLIMITED} ON tablespaceName]]
[TEMPORARY TABLESPACE tablespaceName [QUOTA {integer [K | M] | UNLIMITED} ON tablespaceName]] [PROFILE profileName] [PASSWORD EXPIRE] [ACCOUNT {LOCK |UNLOCK}];

- **IDENTIFIED BY password** allows assigning a password to a local user (most common and simplest case).
- **IDENTIFIED BY EXTERNALLY** allows using the authenticity of the operating system to identify Oracle (case of OPS$ accounts for Unix).
- **IDENTIFIED BY GLOBALLY** allows using the authenticity of a directory system.
- **DEFAULT TABLESPACE nameTablespace** assigns a working disk space (called tablespace) to the user.
- **TEMPORARY TABLESPACE nameTablespace** assigns a temporary disk space to the user.
- **QUOTA** allows limiting or not each allocated space.
- **PROFILE nameProfile** assigns a profile (system characteristics related to CPU and connections) to the user.
- **PASSWORD EXPIRE** to force the user to change his password at the first connection (by default, he is free not to do so). The DBA can also change this password.
- **ACCOUNT** to lock or release access to the database (by default UNLOCK).

The SQL syntax for creating a profile

CREATE PROFILE profileName
LIMIT {Resource Setting | Password Setting} [Resource Setting | Password Setting];

Resource Setting: {{SESSIONS_PER_USER|CPU_PER_SESSION|CPU_PER_CALL|CONNECT_TIME|
IDLE_TIME|LOGICAL_READS_PER_SESSION|LOGICAL_READS_PER_CALL | COMPOSITE_LIMIT}{integer| UNLIMITED | DEFAULT} | PRIVATE_SGA {integer[K|M] | UNLIMITED | DEFAULT}}
Password Setting: {FAILED_LOGIN_ATTEMPTS | PASSWORD_LIFE_TIME | PASSWORD_REUSE_TIME | PASSWORD_REUSE_MAX | PASSWORD_LOCK_TIME | PASSWORD_GRACE_TIME} {expression | UNLIMITED | DEFAULT}}

The main options are as follows:

- SESSIONS_PER_USER: number of concurrent sessions allowed.
- CPU_PER_SESSION: maximum CPU time for a session in hundredths of seconds.
- CPU_PER_CALL: CPU time allowed for a kernel call in hundredths of seconds.
- CONNECT_TIME: total time allowed for a session in minutes (useful for timed TP exams).
- LOGICAL_READS_PER_SESSION: define the maximum number of blocks read during a session. We'll talk here about blocks read on disk and in memory.
- LOGICAL_READS_PER_CALL: define the maximum number of blocks read during a "server call". We'll talk here about blocks read on disk and in memory
- IDLE_TIME: allowed inactivity time, in minutes, within the same session (for students who never close their sessions).
- PRIVATE_SGA: private memory space allocated in the SGA (System Global Area).
- FAILED_LOGIN_ATTEMPTS: number of logins attempt before blocking the user.
- PASSWORD_LIFE_TIME: number of days of validity of the password (it expires if not changed during this period).
- PASSWORD_REUSE_TIME: number of days before the password can be used again. If this parameter is set to an integer, the PASSWORD_REUSE_MAX parameter must be set to UNLIMITED.
- PASSWORD_REUSE_MAX: number of password changes before the current password can be used again. If this parameter is set to an integer, the PASSWORD_REUSE_TIME parameter must be set to UNLIMITED.
- PASSWORD_LOCK_TIME: number of days for which an account will be locked out after the number of login attempts has been reached.
- PASSWORD_GRACE_TIME: number of days of a grace period that extends the use of the password before it is changed (a warning message is displayed when logging in). After this period, the password expires.

Syntax for creating a role

CREATE ROLE rolename [NOT IDENTIFIED | IDENTIFIED {BY password | USING [schema.]package | EXTERNALLY | GLOBALLY}]

Important: Fix the problem: "no privileges on the tablespace"

When you launch the creation of a table, Oracle tells you that your user does not have the rights on the tablespace INTERVENTION_TBS. To solve this problem, you have to allocate a space quota on INTERVENTION _TBS to Admin before giving him the right to create tables. The command is the following:

Alter user Admin quota unlimited on INTERVENTION _TBS;