Practice - Schema Privileges

Practice Target

Configure and verify schema-level privilege.

Practice Overview

In this practice you will perform the following:

- Create two test users and a role
- Build sequences, tables, views, and a procedure in the one schema
- Grant schema privileges on the first user to the second test user and role
- Verify those privileges by performing DDL/DML and execution

Demonstrating Fast Ingest Feature

1. Open SQL Developer and connect to the PDB as **SYSTEM** or **SYS** user

2. Create two users and one role.

```
CREATE USER user1 IDENTIFIED BY oracle QUOTA UNLIMITED ON USERS;
GRANT CREATE SESSION, CREATE SEQUENCE, CREATE TABLE, CREATE VIEW, CREATE
PROCEDURE
  TO user1;

CREATE USER user2 IDENTIFIED BY oracle QUOTA UNLIMITED ON USERS; GRANT
CREATE SESSION TO user2;

-- Create a role
CREATE ROLE schema_role;
```

3. Open Putty session (or SQL Developer) and connect to the database as user1.

```
sql user1/oracle@//localhost:1521/freepdb1
```

4. Create objects in the USER1 schema.

```
-- Sequences
CREATE SEQUENCE seq1 START WITH 1 INCREMENT BY 1 NOCACHE;
-- Tables
CREATE TABLE employees_copy (
  employee_id   NUMBER       PRIMARY KEY,
first_name     VARCHAR2(20),    last_name
VARCHAR2(25)
);
INSERT INTO employees_copy (employee_id, first_name, last_name)
VALUES (1001, 'Alice', 'Johnson');

INSERT INTO employees_copy (employee_id, first_name, last_name)
VALUES (1002, 'Brian', 'Lee');

INSERT INTO employees_copy (employee_id, first_name, last_name)
VALUES (1003, 'Carla', 'Martinez');

INSERT INTO employees_copy (employee_id, first_name, last_name)
VALUES (1004, 'David', 'Nguyen');

INSERT INTO employees_copy (employee_id, first_name, last_name)
VALUES (1005, 'Elena', 'Patel');

COMMIT;
```

```
CREATE TABLE departments_copy (
department_id NUMBER PRIMARY KEY,
department_name VARCHAR2(50)
);

INSERT INTO departments_copy
  VALUES (100, 'Practice Dept');
COMMIT;

-- View
CREATE OR REPLACE VIEW emp_view AS
  SELECT employee_id, first_name, last_name
    FROM employees_copy;

-- Procedure
CREATE OR REPLACE PROCEDURE proc1 AS
BEGIN
  NULL;
END;
/
```

5. In the SYSTEM session, grant system privileges on the USER1 schema to the role and then grant the role to USER2.

```
-- Sequences
GRANT SELECT ANY SEQUENCE
  ON SCHEMA user1
  TO schema_role;

-- Tables and Views
GRANT SELECT ANY TABLE
  ON SCHEMA user1
  TO schema_role;
GRANT INSERT ANY TABLE
  ON SCHEMA user1
  TO schema_role;
GRANT UPDATE ANY TABLE
  ON SCHEMA user1
  TO schema_role;
GRANT DELETE ANY TABLE
  ON SCHEMA user1
  TO schema_role;

-- Procedures
GRANT EXECUTE ANY PROCEDURE
  ON SCHEMA user1
  TO schema_role;
```

```
-- grant the role to the user GRANT schema_role to user2;
```

6. Connect as USER2 and verify each granted privilege.

```
conn user2/oracle@//localhost:1521/freepdb1

-- Sequences
SELECT user1.seq1.NEXTVAL FROM dual;

-- Tables
SELECT COUNT(*) FROM user1.employees_copy;
INSERT INTO user1.employees_copy  VALUES (user1.seq1.NEXTVAL, 'Test', 'Row');
UPDATE user1.departments_copy
  SET department_name = 'New Dept';
DELETE FROM user1.departments_copy
  WHERE department_id = 100;
COMMIT;

-- Views
SELECT * FROM user1.emp_view;

-- Procedure execution EXEC user1.proc1;
```

7. Try to execute the granted privilege on a schema other than USER1, like HR schema.

```
SELECT count(*) FROM HR.EMPLOYEES;
```

8. As a cleanup, perform the following
   a. Exit from USER2 session.
   b. In SYSTEM session, drop the created testing users and role.

```
DROP USER IF EXISTS user1 CASCADE;
DROP USER IF EXISTS user2 CASCADE;
DROP ROLE   IF EXISTS schema_role;
```