

# Chapter 15 Security

## Security Concepts

- Schema
    - a container for a collection of objects
    - essentially a user account
  - SHOW USER
    - SQL\*Plus command to confirm which schema you're connected to
- ```
SHOW USER
```
- Two Special User Accounts
    - each is created when database is created and granted DBA role
    - SYSTEM
    - SYS
      - contains the data dictionary base tables and views

2

## Security Concepts

- Privileges
  - the right to execute particular SQL statements
  - control what users can do
  - are GRANTed to a grantee by a grantor
  - can be REVOKEd
- Users
  - can use the objects they own and any additional objects they've been granted permissions on
  - PUBLIC
    - a special user group
- Roles
  - a named bundle of related privileges
  - can be granted to users or to other roles
  - simplifies the process of granting and revoking privileges

3

## System Privileges

- Allow a user to perform a particular database operation or class of database operations
- Examples
  - CREATE SESSION
  - CREATE TABLE
  - CREATE SEQUENCE
  - CREATE VIEW
  - CREATE INDEX
  - ALTER TABLE
  - ALTER SEQUENCE
  - DROP TABLE
  - DROP VIEW

4

## Object Privileges

- A right to perform a particular action on a specific object in a specific schema
- Each type of object has a specific set of grantable privileges
- Examples
  - ❑ SELECT
  - ❑ INSERT
  - ❑ UPDATE
  - ❑ DELETE
  - ❑ INDEX
  - ❑ FLASHBACK
  - ❑ REFERENCES
  - ❑ EXECUTE
  - ❑ ALL

5

## Data Control Language (DCL)

- Commands that control what users can do
- Examples
  - ❑ GRANT
  - ❑ REVOKE
  - ❑ CREATE USER
  - ❑ CREATE ROLE
  - ❑ ALTER USER
  - ❑ DROP ROLE
  - ❑ DROP USER

6

## CREATE USER

```
CREATE USER username  
IDENTIFIED BY password;
```

- Must have sufficient privileges
  - The new user does not yet have any privileges
    - ❑ the DBA can then grant system/object privileges to that user
- ```
CREATE USER newbie IDENTIFIED BY artichoke;
```
- ❑ launch a 2<sup>nd</sup> instance of SQL Developer, can you now logon as newbie?

```
GRANT connect, resource TO newbie;
```

7

## GRANTing System Privileges

```
GRANT privilege [, privilege...]  
TO user [, user...];
```

- Generally done by DBAs

```
GRANT CREATE SESSION TO newbie;
```

```
GRANT CREATE TABLE TO scott;
```

8

## GRANTing Object Privileges

```
GRANT object_priv [(columns)]
ON object
TO {user|role|PUBLIC}
[WITH GRANT OPTION];
```

- Granted privileges take effect immediately
- Privileges vary by object type
- Must be the object's owner or have sufficient privileges

```
GRANT SELECT, INSERT
ON ttrollen.TYPE TO newbie WITH GRANT OPTION;
```

- Can newbie now select from my table?
- Can newbie now delete rows from my table?

```
SELECT * FROM ttrollen.type;
```

```
DELETE FROM ttrollen.type;
```

## WITH GRANT OPTION

- Allows the grantee to grant the privilege to others
- Object privileges granted under WITH GRANT OPTION are revoked if the grantor's object privilege is later revoked
  - does not apply to system privileges
- In the session where you're logged on as **newbie**... try the following:

```
GRANT SELECT, INSERT
ON ttrollen.type TO yourusername
```

- Return to your own session, can **you** now select from my table?

```
SELECT * FROM ttrollen.type
```

10

## Granting Column Privileges

- By default, table privileges apply to all columns in a table
- Can grant UPDATE, REFERENCES privs on specified column(s)

```
GRANT UPDATE (lastcontact, phone)
ON ttrollen.writer
TO newbie;
```

```
UPDATE ttrollen.writer
SET phone = '(480) 423-6000'
WHERE writerid = 'J525';
```

```
UPDATE ttrollen.writer
SET amount = 150
WHERE writerid = 'C200';
```

11

## Object Privilege Dictionary Views

- ALL\_TAB\_PRIVS** (nib)
  - lists **object grants** for which the user is the grantor, grantee, object's owner, or the grantee is an enabled role or PUBLIC

```
SELECT *
FROM ALL_TAB_PRIVS
WHERE TABLE_SCHEMA = 'TTROLLEN'
```

ORDER BY PRIVILEGE;

	GRANTOR	GRANTEE	TABLE_SCHEMA	TABLE_NAME	PRIVILEGE	GRANTABLE
1	TTROLLEN	PUBLIC	TTROLLEN	AGE	EXECUTE	NO
2	TTROLLEN	NEWBIE	TTROLLEN	TYPE	INSERT	YES
3	TTROLLEN	GRUNT	TTROLLEN	TYPE	INSERT	NO
4	TTROLLEN	PUBLIC	TTROLLEN	WORKORDER	SELECT	NO
5	TTROLLEN	PUBLIC	TTROLLEN	MAILLIST	SELECT	NO
6	TTROLLEN	NEWBIE	TTROLLEN	TYPE	SELECT	YES
7	TTROLLEN	PUBLIC	TTROLLEN	EVENS_SEQ	SELECT	NO
8	TTROLLEN	PUBLIC	TTROLLEN	RECEIPT	SELECT	NO
9	TTROLLEN	GRUNT	TTROLLEN	TYPE	SELECT	NO
10	TTROLLEN	GRUNT	TTROLLEN	WRITER_ACTIVITY	SELECT	NO
11	TTROLLEN	PUBLIC	TTROLLEN	OLEGE	SELECT	NO

USER\_TAB\_PRIVS\_MADE (pg. 691) USER\_TAB\_PRIVS\_RECD (pg. 671)

## REVOKE

```
REVOKE {obj_priv|ALL} [, {obj_priv|ALL}]  
ON    object  
FROM  {user|role|PUBLIC} [, {user|role|PUBLIC}]..
```

- The specified privileges are immediately revoked from the user and from any other users to whom those privileges may have been granted through the WITH GRANT OPTION clause
- Of course `newbie` can no longer use my TYPE table
- But `newbie` granted you privileges, can you still select from my table even though `newbie` lost their privileges?

```
REVOKE ALL ON writer FROM newbie;  
REVOKE SELECT, INSERT ON type FROM newbie;
```

```
SELECT * FROM ttrollen.type;
```

13

## Managing Users

```
ALTER USER username [IDENTIFIED BY password]  
DROP USER username [CASCADE]
```

- ALTER USER
  - ❑ users can change their own password
  - ❑ a DBA can change another user's password
- DROP USER
  - ❑ will fail if the user is connected
  - ❑ use CASCADE option to drop the user and all objects owned by the user
  - ❑ REVOKEs privileges the dropped user GRANTed to others via WITH GRANT OPTION

```
ALTER USER newbie IDENTIFIED BY playground;
```

```
DROP USER newbie CASCADE;
```

14

## Role Concepts

- Roles simplify the process of granting/revoking privileges
- A role is a named bundle of related privileges
  - ❑ can be a mix of system and object privileges
  - ❑ a user can be granted several roles
  - ❑ several users can be assigned the same role
- Roles are the mechanism that allows a user to acquire the set of privileges that permit them to do their job(s)
  - ❑ when a user logs on, Oracle enables all privileges granted explicitly to the user and the privileges derived from the roles they've been granted
  - ❑ any role granted after you connect does not automatically take effect until your next connection

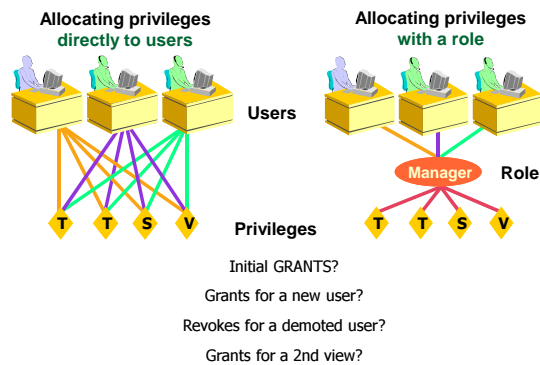
15

## System-Defined Roles: Examples

- CONNECT
  - ❑ confers CREATE SESSION
  - ❑ <10g conveyed several additional system privileges
- RESOURCE
  - ❑ confers CREATE TABLE, CREATE SEQUENCE, CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE TRIGGER, CREATE TYPE
  - ❑ I also granted CREATE VIEW to RESOURCE in our cis119do database
- DBA
  - ❑ confers 160 different system privileges

16

## Using Roles to Manage Privileges



17

## User-Defined Roles

```
CREATE ROLE rolename;
```

- Requires DBA role or CREATE ROLE system privilege
- A three-step process
  - create the role
  - grant permissions to the role
  - grant the role to appropriate users

```
1 CREATE ROLE manager;  
Role created.
```

```
2 GRANT DELETE, UPDATE ON payment TO manager;  
Grant succeeded.
```

```
3 GRANT manager TO blake, clark;  
Grant succeeded.
```

18

## Practice Time: User Defined Role

- An organization has new employees whose work requires them to read from and add new rows to the **emp** and **dept** tables in the **scott** schema, the **type** table, and read from the **writer\_activity** view in the **ttrollen** schema.
- Help the DBA:
  - create users named **jbrown**, **manderson**, and **pdaley** with passwords **alpha**, **beta**, and **gamma**, respectively
  - create a new role named **grunt**
  - assign appropriate permissions to the role
  - assign the role to **jbrown**, **manderson**, and **pdaley**
- Now, launch another instance of SQL Developer and log in as your choice of **jbrown**, **manderson**, **pdaley**, or **scott**

19

## Role-Related Data Dictionary Views

- USER\_ROLE\_PRIVS** (used pg. 688)

- lists the roles granted to the current user

```
SELECT * FROM USER_ROLE_PRIVS;
```

- ROLE\_SYS\_PRIVS** (pg. 670)

- lists **system** privileges granted to roles the user has been granted

```
SELECT * FROM ROLE_SYS_PRIVS WHERE ROLE = 'RESOURCE';
```

- ROLE\_TAB\_PRIVS** (pg. 677)

- lists **object** privileges granted to roles the user has been granted

```
SELECT * FROM ROLE_TAB_PRIVS WHERE ROLE = 'GRUNT';
```

20

## DROP ROLE

```
DROP ROLE rolename;
```

- Oracle revokes the role from all users (and roles) it had been granted and removes it from the database
- You must have been granted the role with the ADMIN OPTION or you must have the DROP ANY ROLE system privilege

```
DROP ROLE grant;
```

21

## Synonym Concepts

- An alias for a table, view, sequence, procedure, function, package, or materialized view (snapshot)
- Commonly used to
  - eliminate the need to qualify the object name with the schema name when referring to another user's object
  - protect/hide the real object's name and schema location
- Private Synonym
  - exists in the schema of a specific user who has control over its availability to others
  - use to create a shorter name for an object in your own schema
- Public Synonym
  - owned by the user group PUBLIC and every user can access it
- If an object in your schema has the same name as a public synonym, the **local object** prevails

22

## Managing Synonyms

```
CREATE [OR REPLACE] [PUBLIC] SYNONYM synonym_name  
FOR [schema].object;
```

### ■ CREATE SYNONYM

```
CREATE PUBLIC SYNONYM author FOR ttrollen.writer;  
Synonym created.  
GRANT SELECT, INSERT ON author TO public;  
Grant succeeded.  
SELECT * FROM author;
```

### ■ DROP SYNONYM

- only a DBA can drop a public synonym

```
DROP PUBLIC SYNONYM author;  
Synonym dropped.
```

23

## Profile

- A named set of **resource** and **password limits**
- For profiles to take effect, resource limiting must be turned on for the database as a whole

```
CREATE PROFILE appuser LIMIT  
FAILED_LOGIN_ATTEMPTS 3  
SESSIONS_PER_USER 2  
CPU_PER_SESSION unlimited  
CONNECT_TIME 60  
IDLE_TIME 15;  
CREATE PROFILE succeeded.
```

### ■ Can assign a profile to each user

- a default profile is used for users not assigned a specific profile

```
ALTER USER newbie PROFILE appuser;  
ALTER USER newbie succeeded.
```

24