

Chapter 14 Security

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 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

2

System Privileges

- Allows 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

3

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
 - REFERENCES
 - EXECUTE
 - ALL

4

Data Control Language (DCL)

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

5

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

```
SQL> CREATE USER newbie IDENTIFIED BY artichoke
```

- Launch a 2nd SQL*Plus session, can you now logon as newbie?

```
SQL> GRANT connect, resource TO newbie
```

6

GRANTing System Privileges

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

- Generally done by DBAs

```
SQL> GRANT CREATE SESSION TO newbie;
```

```
SQL> GRANT CREATE TABLE TO scott;
```

7

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
- You must be the object's owner or have sufficient privileges

```
SQL> GRANT SELECT, INSERT  
ON TYPE TO newbie WITH GRANT OPTION
```

- Can newbie now select from my table?

```
SQL> SELECT * FROM ttrollen.type
```

- Can newbie now delete rows from my table?

```
SQL> DELETE FROM ttrollen.type
```

8

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
- You're still logged on as newbie... try the following:

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

- Now, logon as yourself

```
SQL> CONNECT yourusername@cis119do  
Password: *****
```

- Can you now select from my table?

```
SQL> SELECT * FROM ttrollen.type
```

9

Granting Column Privileges

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

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

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

```
SQL> UPDATE ttrollen.writer
      SET amount = 150
      WHERE writerid = 'J525'
```

10

Object Security-Related Data Dictionary Views

- ALL_TAB_PRIVS (tab)
 - lists object grants for which
 - the user is the grantor, grantee, owner
 - or an enabled role or PUBLIC is the grantee
 - displays tables, views, sequences

```
SQL> SELECT *
      FROM ALL_TAB_PRIVS
      WHERE GRANTEE = 'PUBLIC'
      AND GRANTOR = 'TTROLLEN'
```

- USER_TAB_PRIVS_MADE (pg. 691)
- USER_TAB_PRIVS_RECD (pg. 671)

11

REVOKE

```
REVOKE {privilege [, privilege...]}|ALL
      ON object
      FROM {user[, 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

```
SQL> REVOKE ALL ON writer FROM newbie
SQL> REVOKE SELECT, INSERT ON type FROM newbie
```

- 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?

```
SQL> SELECT * FROM ttrollen.type
```

12

ALTER USER

```
ALTER USER username [IDENTIFIED BY password]
[DEFAULT TABLESPACE tablespacename]
```

- To change a user's password, profile, or default tablespace
 - users can change their own password
 - a DBA can change another user's password

```
SQL> ALTER USER newbie IDENTIFIED BY playground
```

13

DROP USER

```
DROP USER username [CASCADE]
```

- Will fail if the user is connected
- Will fail if the user owns any objects
 - use the CASCADE option to drop the user and all objects owned by the user
- Also REVOKES privileges the dropped user GRANTED to others via WITH GRANT OPTION

```
SQL> DROP USER newbie CASCADE
```

14

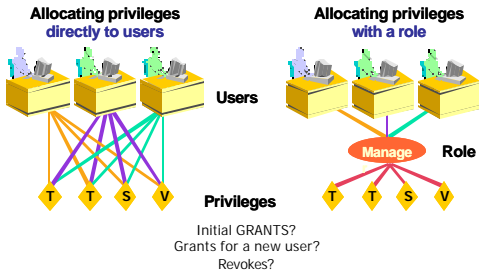
Role Concepts

- A role is a named bundle of privileges
 - can mix system and object privileges
- When a user logs on, Oracle enables all privileges granted explicitly to the user and privileges in the user's default roles
 - any role granted to you after you connect does not take effect until your next connection
- Use the SET ROLE statement to enable and disable roles for your current session

```
SQL> SET ROLE ALL;
Role set.
```

15

Using Roles to Manage Privileges



16

System-Defined Roles: Examples

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

17

User-Defined Roles

```
CREATE ROLE rolename;
```

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

```
SQL> CREATE ROLE manager;
Role created.

SQL> GRANT DELETE, UPDATE ON payment TO manager;
Grant succeeded.

SQL> GRANT manager TO blake, clark;
Grant succeeded.
```

18

Practice Time

- Our organization has new employees whose grunt work will require they be able to `select` from and `insert` into the `emp` and `dept` tables in the `scott` schema, the `type` table, and `select` on 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*Plus and log in as your choice of `jbrown`, `manderson`, `pdaley`, or `scott`

19

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

```
SQL> DROP ROLE manager;
```

20

Role-Related Data Dictionary Views

- `USER_ROLE_PRIVS` (used pg. 685)
 - lists `roles` that have been granted to the current user

```
SQL> SELECT * FROM USER_ROLE_PRIVS
```
- `ROLE_SYS_PRIVS` (nib)
 - shows information about `system` privileges granted to the `roles` to which the user has access

```
SQL> SELECT *
      FROM ROLE_SYS_PRIVS WHERE ROLE = 'RESOURCE'
```
- `ROLE_TAB_PRIVS` (nib)
 - contains information about `object` privileges granted to the `roles` the current user has been granted
 - includes tables, views, sequences

```
SQL> SELECT * FROM ROLE_TAB_PRIVS
```

21



Synonym Concepts

- An **alias** for a table, view, sequence, procedure, function, package, or materialized vies (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 [PUBLIC] SYNONYM synonym
FOR [schema].object;
```

■ CREATE SYNONYM

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

■ DROP SYNONYM

- only a DBA can drop a public synonym

```
SQL> DROP PUBLIC SYNONYM employee;
Synonym dropped.
```

23



Profile

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

```
SQL> CREATE PROFILE appuser LIMIT
      FAILED_LOGIN_ATTEMPTS 5
      SESSIONS_PER_USER 2
      CPU_PER_SESSION unlimited
      IDLE_TIME 15;
```

- Can assign a profile to each user
 - and a default profile for users who've not been assigned a specific profile

```
SQL> ALTER USER newbie PROFILE appuser;
```

24
