1. **Creating Roles**

GRANT **CREATE**ROLE TO store;  
GRANT **CREATE**USER TO store WITH ADMIN OPTION;  
**CREATE**ROLE product\_manager;  
**CREATE**ROLE hr\_manager;  
**CREATE**ROLE overall\_manager IDENTIFIED by manager\_password;

1. **Granting Privileges to Roles**

GRANT SELECT, INSERT, UPDATE, **DELETE**ON tableNames TO product\_manager;  
GRANT **CREATE**USER TO hr\_manager;  
GRANT product\_manager, hr\_manager TO overall\_manager;

1. **Granting Roles to a User**

GRANT overall\_manager TO userName;

1. **Assign Role to User**

 GRANT TestRole TO Alice;

1. **Checking Roles Granted to a User**

SQL> desc user\_role\_privs;

**SELECT**\* **FROM**user\_role\_privs;

1. **Checking System Privileges Granted to a Role**

SQL> desc role\_sys\_privs;

**SELECT**\*  
**FROM**role\_sys\_privs;

|  |  |  |
| --- | --- | --- |
| 1. **Checking Object Privileges Granted to a Role**   SQL> desc role\_tab\_privs;  **SELECT**\* **FROM**role\_tab\_privs   1. **Assign CONNECT and RESOURCE Roles**   SQL> SQL> **CREATE**USER Alice IDENTIFIED BY simplepassword;  User created.  SQL> SQL> GRANT CONNECT, RESOURCE to Alice;  Grant succeeded.  SQL> SQL> SQL> drop user alice;  User dropped.  SQL> SQL>   1. **Default Roles**   ALTER USER steve DEFAULT ROLE ALL EXCEPT overall\_manager;  SET ROLE NONE; SET ROLE ALL EXCEPT overall\_manager;   1. **Revoking a Role**   REVOKE overall\_manager **FROM**steve;   1. **Revoking Privileges from a Role**   **CREATE**ROLE product\_manager; GRANT SELECT, INSERT, UPDATE, **DELETE**ON employee TO product\_manager;  REVOKE ALL ON products **FROM**product\_manager;  DROP ROLE product\_manager;   1. **Dropping a Role**   **CREATE**ROLE product\_manager; GRANT SELECT, INSERT, UPDATE, **DELETE**ON employee TO product\_manager;  REVOKE ALL ON products **FROM**product\_manager;  DROP ROLE product\_manager;   1. **Regina and Kristen are manager level users**   SQL> **CREATE**ROLE manager;  Role created.  SQL> SQL> **CREATE**USER regina identified by regina;  User created.  SQL> **CREATE**USER kristen identified by kristen;  User created.  SQL> GRANT CONNECT TO regina, kristen;  Grant succeeded.  SQL> GRANT manager TO regina, kristen;  Grant succeeded.  SQL> SQL> drop user regina;  User dropped.  SQL> drop user kristen;  User dropped.  SQL> SQL> drop role manager;  Role dropped.   1. **Rich and Brad are maintenance level users**   QL> **CREATE**ROLE maintenance;  Role created.  SQL> SQL> **CREATE**USER rich identified by rich;  User created.  SQL> **CREATE**USER brad identified by brad;  User created.  SQL> GRANT CONNECT TO rich, brad;  Grant succeeded.  SQL> GRANT maintenance TO rich, brad;  Grant succeeded.  SQL> SQL> drop user rich;  User dropped.  SQL> drop user brad;  User dropped.  SQL> drop role maintenance;  Role dropped.   |  | | --- | |  | |  | |

1. **Unassign Role**

SQL>  
SQL> **CREATE**USER Alice IDENTIFIED BY simplepassword;  
  
User created.  
  
SQL>  
SQL> GRANT CONNECT, RESOURCE to Alice;  
  
Grant succeeded.  
  
SQL>  
SQL> REVOKE RESOURCE **FROM**Alice;  
  
Revoke succeeded.  
  
SQL>  
SQL> drop user alice;  
  
User dropped.  
  
SQL>  
SQL>  
SQL>

Drop Role

SQL>  
SQL> DROP ROLE TestRole;  
  
Role dropped.

1. **Assign object privileges to roles**

SQL> GRANT SELECT, **INSERT**ON employee TO entry;  
SQL> GRANT SELECT, **INSERT**ON customer TO entry;  
SQL> GRANT SELECT, **UPDATE**ON employee TO maintenance;  
SQL> GRANT SELECT, **UPDATE**ON customer TO maintenance;  
SQL> GRANT EXECUTE ON validate\_salary TO maintenance;  
SQL> GRANT SELECT, INSERT, UPDATE, **DELETE**ON employee TO manager;  
SQL> GRANT SELECT, INSERT, UPDATE, **DELETE**ON customer TO manager;  
SQL> GRANT EXECUTE ON validate\_salary TO manager;  
SQL> GRANT EXECUTE ON adjust\_salary TO manager;

1. **Unassign Role from User**

SQL> REVOKE TestRole **FROM**Alice

Output session roles from procedure

SQL>  
SQL> **create**or replace **procedure**myProcedure  
  2    authid current\_user  
  3    as  
  4    **begin**  
  5      **for**rec in (**select**\* **from**session\_roles)  
  6      loop  
  7        dbms\_output.put\_line(rec.role);  
  8      **end**loop;  
  9    **end**;  
 10    /  
  
Procedure created.  
  
SQL> execute myProcedure  
DBA  
SELECT\_CATALOG\_ROLE  
HS\_ADMIN\_ROLE  
EXECUTE\_CATALOG\_ROLE  
DELETE\_CATALOG\_ROLE  
EXP\_FULL\_DATABASE  
IMP\_FULL\_DATABASE  
GATHER\_SYSTEM\_STATISTICS  
SCHEDULER\_ADMIN  
XDBADMIN  
XDBWEBSERVICES  
  
PL/SQL **procedure**successfully completed.  
  
SQL>  
SQL>  
SQL> drop **procedure**myProcedure;  
  
Procedure dropped.