RBAC – Implementation

3.1 Authorization

The authorization gives permission to user to access database, modify the data or display the information.

It also controls user to access another user object schemas such as tables or rows or resources like connection, idle times.

There is two methods to grant privileges to user one is grant privileges to user explicitly and another method is grant privileges to a role.

And then grant role to one or group of users in the database.

The privileges are classified into two features as System privilege and Object Privilege.

USER PRIVILEGE

SYSTEM PRIVILEGE

Permission to perform any action on any schema objects such as ALTER Or CREATE TABLESPACE, CREATE USER, ALTER USER, DROP USER, ALTER SYSTEM

```
SQL> Grant create table, create view to salesrep;

Grant succeeded.
```

OBJECT PRIVILEGE

Grant permission to a user on a database object. For example update or delete table, Execute procedure, function, views of another user.

```
SQL> Grant select on hr.company to salesrep;

Grant succeeded.
```

3.2 Role

In Oracle roles are created by database administrator.

The user privileges are grant permission to execute specific type of SQL statement or access another user's database object.

The roles are storing group of privileges. The roles are assign to group of user or another roles. Roles can be created with or without password.

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> Create role Salesrep;

Role created.

SQL> Create role salesmanager identified by sm123;

Role created.

SQL> Create role cashier;

Role created.
```

Assign password to a role will protect authorization in the database.

Using SET ROLE statement explicitly to role will enable the role.

For disabling the role, the user assign proper password to the role.

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> SET ROLE salesmanager IDENTIFIED BY sm123;

Role set.
```

Within a database, each **role name must be unique**. we already created role salesrep, again create the role it will give error.

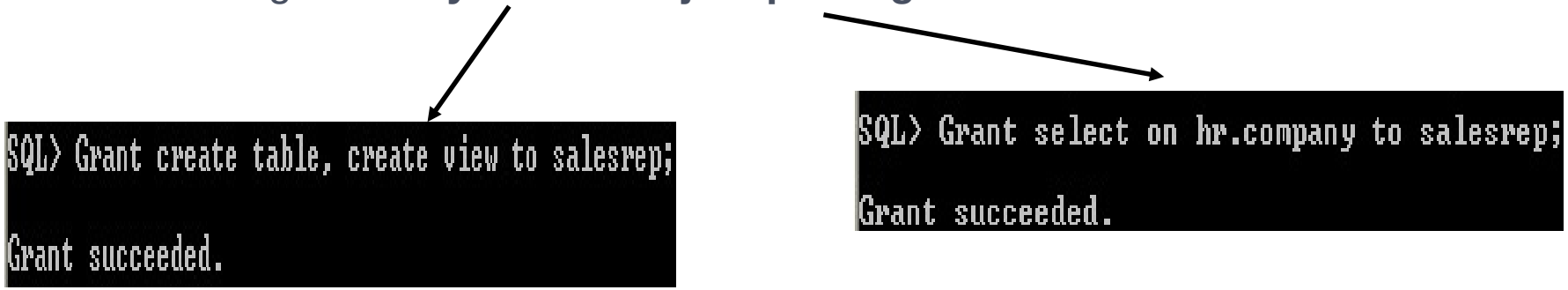
```
SQL> Create role salesrep;
Create role salesrep
            *
ERROR at line 1:
ORA-01921: role name 'SALESREP' conflicts with another user or role name
```

Role and User name should be unique, don't have same name. Username arvind is exist in the database, role should be different than user name.

```
SQL> Create role Arvind;
Create role Arvind
            *
ERROR at line 1:
ORA-01921: role name 'ARVIND' conflicts with another user or role name
```

3.3 Functionality of Roles

•A role can be granted **system** or **object privileges**.



•Any role can be granted to any database user.

```
SQL> Grant salesrep to sinduja;

Grant succeeded.
```

- A role can be granted to other roles.

```
SQL> Grant salesmanager to salesrep;

Grant succeeded.
```

- A role cannot be granted itself

```
SQL> Grant salesrep to salesrep;
Grant salesrep to salesrep
*
ERROR at line 1:
ORA-01934: circular role grant detected
```

- A role cannot be circular

```
SQL> Grant salesmanager to salesrep;

Grant succeeded.

SQL> Grant salesrep to salesmanager;
Grant salesrep to salesmanager
*
ERROR at line 1:
ORA-01934: circular role grant detected
```

•If a role is not password authenticated or a secure application role, then you can grant the role indirectly to the user. User Arvind has direct role salesrep and indirect role cashier.

```
SQL> Grant cashier to salesrep;

Grant succeeded.

SQL> Grant salesrep to Arvind;

Grant succeeded.
```

- Assign directly or indirectly granted role as default role.

Before assign to user as default role, first you should grant role to user.

Here Cashier is indirect role and salesrep is direct role.

But Salesmanager is not grant to user arvind, if try to assign as default role then it will give error

```
SQL> ALTER USER ARVIND DEFAULT ROLE SALESREP;

User altered.

SQL> ALTER USER ARVIND DEFAULT ROLE CASHIER;

User altered.

SQL> ALTER USER ARVIND DEFAULT ROLE SALESMANAGER;
ALTER USER ARVIND DEFAULT ROLE SALESMANAGER
*
ERROR at line 1:
ORA-01955: DEFAULT ROLE 'SALESMANAGER' not granted to user
```

- The data dictionary view DBA_ROLE_PRIVS -Lists roles granted to users and roles . The ROLE_ROLE_PRIVS -This view describes roles granted to other roles. Information is provided only about roles to which the user has access.

```
SQL> describe dba_role_privs;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 GRANTEE                                            VARCHAR2(30)
 GRANTED_ROLE                              NOT NULL VARCHAR2(30)
 ADMIN_OPTION                                       VARCHAR2(3)
 DEFAULT_ROLE                                       VARCHAR2(3)

SQL> describe role_role_privs;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 ROLE                                      NOT NULL VARCHAR2(30)
 GRANTED_ROLE                              NOT NULL VARCHAR2(30)
 ADMIN_OPTION                                       VARCHAR2(3)
```

```
SQL> SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE LIKE 'SALES%';

GRANTEE                        GRANTED_ROLE                   ADM DEF
------------------------------ ------------------------------ --- ---
SALESREP                       CASHIER                        NO  YES
SALESREP                       SALESMANAGER                   NO  YES

SQL> SELECT * FROM ROLE_ROLE_PRIVS WHERE ROLE LIKE 'SALES%';

ROLE                           GRANTED_ROLE                   ADM
------------------------------ ------------------------------ ---
SALESREP                       CASHIER                        NO
SALESREP                       SALESMANAGER                   NO
```

Type of Role Authorization

Role by using Database

Role by using Database is protected by password and SET ROLE statement will enable the password.

```
SQL> Create role salesmanager identified by sm123;

Role created.
```

Role by using Application

Instead of providing password to the role the application developers secure the role by mentioning PL/SQL package name will authorized to enable the role.

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL>
SQL> CREATE ROLE emp_role IDENTIFIED USING scott.emp_actions;

Role created.
```

Role by using Operating system

Role is authenticated by operating system will grant the operating system privilege to the user. In this case it will dynamically link operating system privileges with applications. It enables whenever application is open and will revoke whenever application is closed.

```
SQL> create role cashier identified externally;

Role created.
```

3.5 Change the authorization method

It is possible to shift from one type of authorization to another type of authorization.

For changing authorization the user has ALTER ANY ROLE system privilege or the user has ADMIN privilege.

```
SQL> ALTER ROLE salesmanager IDENTIFIED EXTERNALLY;

Role altered.

SQL> alter role cashier identified by cash123;

Role altered.
```

3.6 Drop Role

For dropping role the user should have DROP ANY ROLE system privilege or ADMIN option.

The dropping role has indirectly granted role, all the granted roles also be removed from dropping role.

Only privileges are removed, but indirect role is available in the database.

```
SQL> grant cashier to salesmanager;
Grant succeeded.
SQL> drop role salesmanager;
Role dropped.
SQL> grant cashier to arvind;
Grant succeeded.
```

3.7 Authorization of Object privilege

The schema object are table, views, indexes, triggers and database links, dimension, materialized view etc.,

For accessing another user schema object the user need object privilege it will granted through role for user or group of users.

Every user by default has all object privileges for his/her own Schema containing objects.

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> GRANT SELECT ON hr.company TO arvind;

Grant succeeded.

SQL> GRANT SELECT ON hr.company TO public;

Grant succeeded.
```

The object privileges can be grant specifically or all the object privilege assign to user using ALL privilege, it will assign privilege **select, update, debug, flashback, query rewrite, on commit refresh** and **delete**, **references** etc.,

```
SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON hr.company TO arvind;

Grant succeeded.

SQL> GRANT ALL ON hr.company TO sinduja;

Grant succeeded.
```

If user has GRANT ANY OBJECT PRIVILEGE , then the user can grant any specified object privilege to another user.

Using GRANT statement with or without GRANT OPTION the privilege to assign to user.

If user has GRANT ANY OBJECT PRIVILEGE by default the user has to revoke any object privilege that was granted by the owner of the object.

```
SQL> grant GRANT ANY OBJECT PRIVILEGE to Arvind;

Grant succeeded.

SQL> conn arvind/arvind9999
Connected.
SQL> Grant select on hr.company to sinduja;

Grant succeeded.

SQL> Revoke select on hr.company from sinduja;

Revoke succeeded.
```

Without GRANT option means the user is having GRANT ANY OBJECT PRIVILEGE can't pass any grant object privilege to other user, otherwise it will show ORA-01031 error.

```
SQL> conn sinduja/sinduja1234
Connected.
SQL> grant select on hr.company to lakshmi;
grant select on hr.company to lakshmi
                              *
ERROR at line 1:
ORA-01031: insufficient privileges
```

```
SQL> conn arvind
Enter password:
Connected.
SQL> grant select on hr.company to sinduja with grant option;

Grant succeeded.
```

```
SQL> conn sinduja/sinduja1234
Connected.
SQL> grant select on hr.company to lakshmi;

Grant succeeded.
```

```
SQL> revoke select on hr.company from lakshmi;

Revoke succeeded.
```

SQL> grant insert on hr.company to lakshmi;
grant insert on hr.company to lakshmi
*

ERROR at line 1:
ORA-01031: insufficient privileges

3.8 With GRANT option Vs. with ADMIN option

With..Option

With Grant option

It is for object privileges not for system privilege. Grantor of the privilege can revoke the privilege

```
SQL> grant select on hr.company to sinduja with grant option;

Grant succeeded.

SQL> grant select on hr.company to sinduja with admin option;
grant select on hr.company to sinduja with admin option
                                           *
ERROR at line 1:
ORA-00993: missing GRANT keyword
```

With Admin Option

This option used only for system privileges not object privileges.

This option allow system level activities of the users to grant, such as create session, alter session, create table, create user etc..

```
SQL> grant create session, create table to arvind with admin option;

Grant succeeded.

SQL> conn arvind/arvind9999
Connected.
SQL> grant create table to sinduja;

Grant succeeded.
```

3.9 Data Dictionary view of DBA_TAB_PRIVS

DBA_TAB_PRIVS describes all object grants in the database. Object means tables, view, index, procedures and packages etc.,

```
SQL> describe DBA_TAB_PRIVS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GRANTEE                                   NOT NULL VARCHAR2(30)
 OWNER                                     NOT NULL VARCHAR2(30)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 GRANTOR                                   NOT NULL VARCHAR2(30)
 PRIVILEGE                                 NOT NULL VARCHAR2(40)
 GRANTABLE                                          VARCHAR2(3)
 HIERARCHY                                          VARCHAR2(3)

SQL> set linesize 300
SQL> select grantee,grantor,privilege from dba_tab_privs where owner='HR';

GRANTEE                        GRANTOR                        PRIVILEGE
------------------------------ ------------------------------ ----------------
PUBLIC                         HR                             EXECUTE
SINDUJA                        HR                             FLASHBACK
SINDUJA                        HR                             DEBUG
SINDUJA                        HR                             QUERY REWRITE
SINDUJA                        HR                             ON COMMIT REFRESH
SINDUJA                        HR                             REFERENCES
SINDUJA                        HR                             UPDATE
SINDUJA                        HR                             INSERT
SINDUJA                        HR                             INDEX
SINDUJA                        HR                             DELETE
SINDUJA                        HR                             ALTER

GRANTEE                        GRANTOR                        PRIVILEGE
------------------------------ ------------------------------ ----------------
ARVIND                         HR                             UPDATE
ARVIND                         HR                             INSERT
ARVIND                         HR                             DELETE
ARVIND                         HR                             SELECT
SINDUJA                        HR                             SELECT
PUBLIC                         HR                             SELECT
```

3.9.1 Grantor, Grantee and Owner

Owner - The owner of the object, in HR schema the table object COMPANY is created.

Grantor - The user who granted the privilege , the grantor is not owner of the object. The user Sinduja has Grant option privilege of HR.COMPANY.

Grantee- The Privilege is grantable by the grantee, Using grant option Sinduja grant privilege to Lakshmi then Lakshmi is Grantee and Sinduja is Grantor.

```
SQL> grant create session to sinduja identified by sinduja123;

Grant succeeded.

SQL> grant select on hr.company to sinduja with grant option;

Grant succeeded.

SQL> conn sinduja/sinduja123
Connected.
SQL> grant select on hr.company to lakshmi;

Grant succeeded.

SQL> conn sys as sysdba
Enter password:
Connected.
SQL> set linesize 300
SQL> select * from dba_tab_privs where table_name='COMPANY';

GRANTEE                        OWNER                          TABLE_NAME                     GRANTOR                        PRIVILEGE
------------------------------ ------------------------------ ------------------------------ ------------------------------ ----------
SINDUJA                        HR                             COMPANY                        HR                             FLASHBACK
SINDUJA                        HR                             COMPANY                        HR                             DEBUG
SINDUJA                        HR                             COMPANY                        HR                             QUERY REWRITE
SINDUJA                        HR                             COMPANY                        HR                             ON COMMIT REFRESH
SINDUJA                        HR                             COMPANY                        HR                             REFERENCES
SINDUJA                        HR                             COMPANY                        HR                             UPDATE
PUBLIC                         HR                             COMPANY                        HR                             SELECT
SINDUJA                        HR                             COMPANY                        HR                             SELECT
ARVIND                         HR                             COMPANY                        HR                             SELECT
SALESREP                       HR                             COMPANY                        HR                             SELECT
LAKSHMI                        HR                             COMPANY                        SINDUJA                        SELECT
```

3.10 Granting Column Privileges

Using GRANT statement the user can apply INSERT, UPDATE and DELETE column privileges.

The ORA-00969 error will occur while grant SELECT privilege for column to the user.

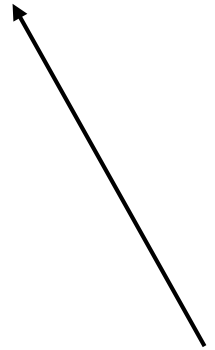
Instead of SELECT privilege column, the user create VIEW for specific column of the schema object and then grant the view to the user.

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> Grant insert(company_name,package_no,year) on hr.company to Arvind;

Grant succeeded.
```

```
SQL> Grant select(company_name,package_no,year) on hr.company to Arvind;
Grant select(company_name,package_no,year) on hr.company to Arvind
             *
ERROR at line 1:
ORA-00969: missing ON keyword
```

```
SQL> Create view compview as select company_name,package_no,year from hr.company;

View created.

SQL> Grant select on compview to arvind;

Grant succeeded.
```

```
SQL> conn arvind/arvind9999
Connected.
SQL> select * from sys.compview;

COMPANY_NAME          PACKAGE_NO       YEAR
-------------------- ---------- ----------
Leisure_time                  1       2015
Leisure_time                  2       2015
Leisure_time                  3       2015
Leisure_time                  1       2016
Leisure_time                  2       2016
Leisure_time                  3       2016
Stay_holiday                  1       2014
Stay_holiday                  2       2014
Stay_holiday                  3       2014
Stay_holiday                  1       2015
Stay_holiday                  2       2015

COMPANY_NAME          PACKAGE_NO       YEAR
-------------------- ---------- ----------
Stay_holiday                  3       2015
Stay_holiday                  1       2016
Stay_holiday                  2       2016
Stay_holiday                  3       2016
Stay_holiday                  1       2016
Stay_holiday                  2       2016
Stay_holiday                  3       2016

18 rows selected.
```



Compview is created under "

Using REVOKE individually column privilege can't be revoked, revoke used only for object privilege not for column object privilege.

It display error ORA-01750. In this case revoke entire object and then grant only specific column.

```
SQL> GRANT UPDATE(YEAR,PACKAGE_NO) ON HR.COMPANY TO ARVIND;

Grant succeeded.

SQL> REVOKE UPDATE(YEAR) ON HR.COMPANY FROM ARVIND;
REVOKE UPDATE(YEAR) ON HR.COMPANY FROM ARVIND
             *
ERROR at line 1:
ORA-01750: UPDATE/REFERENCES may only be REVOKEd from the whole table, not by column


SQL> REVOKE UPDATE ON HR.COMPANY FROM ARVIND;

Revoke succeeded.

SQL> GRANT UPDATE(PACKAGE_NO) ON HR.COMPANY TO ARVIND;

Grant succeeded.
```

The DBA_COL_PRIVS data dictionary view display the granted table column privilege information

```
SQL> DESCRIBE DBA_COL_PRIVS;
 Name
 -----------------------------------------------------------------------------------------------------------------------------
 GRANTEE
 OWNER
 TABLE_NAME
 COLUMN_NAME
 GRANTOR
 PRIVILEGE
 GRANTABLE

SQL> SET LINESIZE 300
SQL> SELECT * FROM DBA_COL_PRIVS WHERE GRANTEE='ARVIND';

GRANTEE                        OWNER                          TABLE_NAME                     COLUMN_NAME                    GRANTOR
------------------------------ ------------------------------ ------------------------------ ------------------------------ --------
ARVIND                         HR                             COMPANY                        COMPANY_NAME                   HR
ARVIND                         HR                             COMPANY                        PACKAGE_NO                     HR
```

The ROLE_ROLE_PRIVS data dictionary view display how many role granted to another role.

```
SQL> DESCRIBE ROLE_ROLE_PRIVS;
 Name
 -----------------------------------------------------------------
 ROLE
 GRANTED_ROLE
 ADMIN_OPTION

SQL> SELECT * FROM ROLE_ROLE_PRIVS WHERE ROLE='SALESREP';

ROLE                           GRANTED_ROLE                   ADM
------------------------------ ------------------------------ ---
SALESREP                       CASHIER                        NO
```

```
SQL> Grant cashier to salesrep;

Grant succeeded.

SQL> Grant salesrep to Arvind;

Grant succeeded.
```

3.11 ALL Shortcut

"ALL" is a shortcut method to use single grant or revoke statement to assign privilege.

```
SQL> conn sys as sysdba
Enter password:
Connected.
SQL> grant all on hr.company to arvind;

Grant succeeded.
```

```
SQL> SET LINESIZE 300
SQL> GRANT ALL ON HR.COMPANY TO ARVIND;

Grant succeeded.

SQL> SET LINESIZE 300
SQL> SELECT * FROM DBA_tab_PRIVS WHERE GRANTEE='ARVIND';

GRANTEE                        OWNER                          TABLE_NAME                     GRANTOR                        PRIVILEGE
------------------------------ ------------------------------ ------------------------------ ------------------------------ ----------
ARVIND                         SYS                            COMPVIEW                       SYS                            SELECT
ARVIND                         HR                             COMPANY                        HR                             FLASHBACK
ARVIND                         HR                             COMPANY                        HR                             DEBUG
ARVIND                         HR                             COMPANY                        HR                             QUERY REWRITE
ARVIND                         HR                             COMPANY                        HR                             ON COMMIT REFRESH
ARVIND                         HR                             COMPANY                        HR                             REFERENCES
ARVIND                         HR                             COMPANY                        HR                             UPDATE
ARVIND                         HR                             COMPANY                        HR                             INSERT
ARVIND                         HR                             COMPANY                        HR                             INDEX
ARVIND                         HR                             COMPANY                        HR                             DELETE
ARVIND                         HR                             COMPANY                        HR                             SELECT

GRANTEE                        OWNER                          TABLE_NAME                     GRANTOR                        PRIVILEGE
------------------------------ ------------------------------ ------------------------------ ------------------------------ ----------
ARVIND                         HR                             COMPANY                        HR                             ALTER

12 rows selected.
```

The REVOKE ALL privilege revoke only what all are the privilege given using GRANT ALL object privilege, it doesn't revoke already given specific privileges . still **compview** object privilege is available,

```
SQL> revoke all on hr.company from arvind;

Revoke succeeded.
```

```
SQL> SELECT GRANTEE,OWNER,PRIVILEGE,TABLE_NAME FROM DBA_TAB_PRIVS WHERE GRANTEE='ARVIND';

GRANTEE                        OWNER                          PRIVILEGE                                TABLE_NAME
------------------------------ ------------------------------ ---------------------------------------- ----------
ARVIND                         SYS                            SELECT                                   COMPVIEW
```

3.12 SYNONYM Object Privilege

Crete synonym statement will create alternative name for schema objects like view, tables, sequences, procedures and packages etc.,

Like granting table object the synonym to be grant to user.

Here synonym is created in HR schema, even though synonym is granted to user ARVIND while selecting mention schema.object name HR.COMP_SYN

```
SQL> conn hr/hr
Connected.
SQL> CREATE SYNONYM COMP_SYN FOR COMPANY;

Synonym created.

SQL> GRANT SELECT ON COMP_SYN TO ARVIND;

Grant succeeded.
```

```
SQL> conn arvind/arvind9999
Connected.
SQL> select * from hr.comp_syn;

COMPANY_NAME          PACKAGE_NO       YEAR NO_OF_TOURIST NO_OF_DAYS
-------------------- ---------- ---------- -------------- ----------
Leisure_time                  1       2015             45          9
Leisure_time                  2       2015             78         11
Leisure_time                  3       2015             80         15
Leisure_time                  1       2016             45          8
Leisure_time                  2       2016             37         11
Leisure_time                  3       2016             42         15
Stay_holiday                  1       2014             80          9
Stay_holiday                  2       2014             55         12
Stay_holiday                  3       2014             22         14
Stay_holiday                  1       2015             35          9
Stay_holiday                  2       2015             65         12

COMPANY_NAME          PACKAGE_NO       YEAR NO_OF_TOURIST NO_OF_DAYS
-------------------- ---------- ---------- -------------- ----------
Stay_holiday                  3       2015             78         14
Stay_holiday                  1       2016             90          9
Stay_holiday                  2       2016            102         12
Stay_holiday                  3       2016            110         14
Stay_holiday                  1       2016             90          9
Stay_holiday                  2       2016            102         12
Stay_holiday                  3       2016            110         14

18 rows selected.
```

Extra Slides

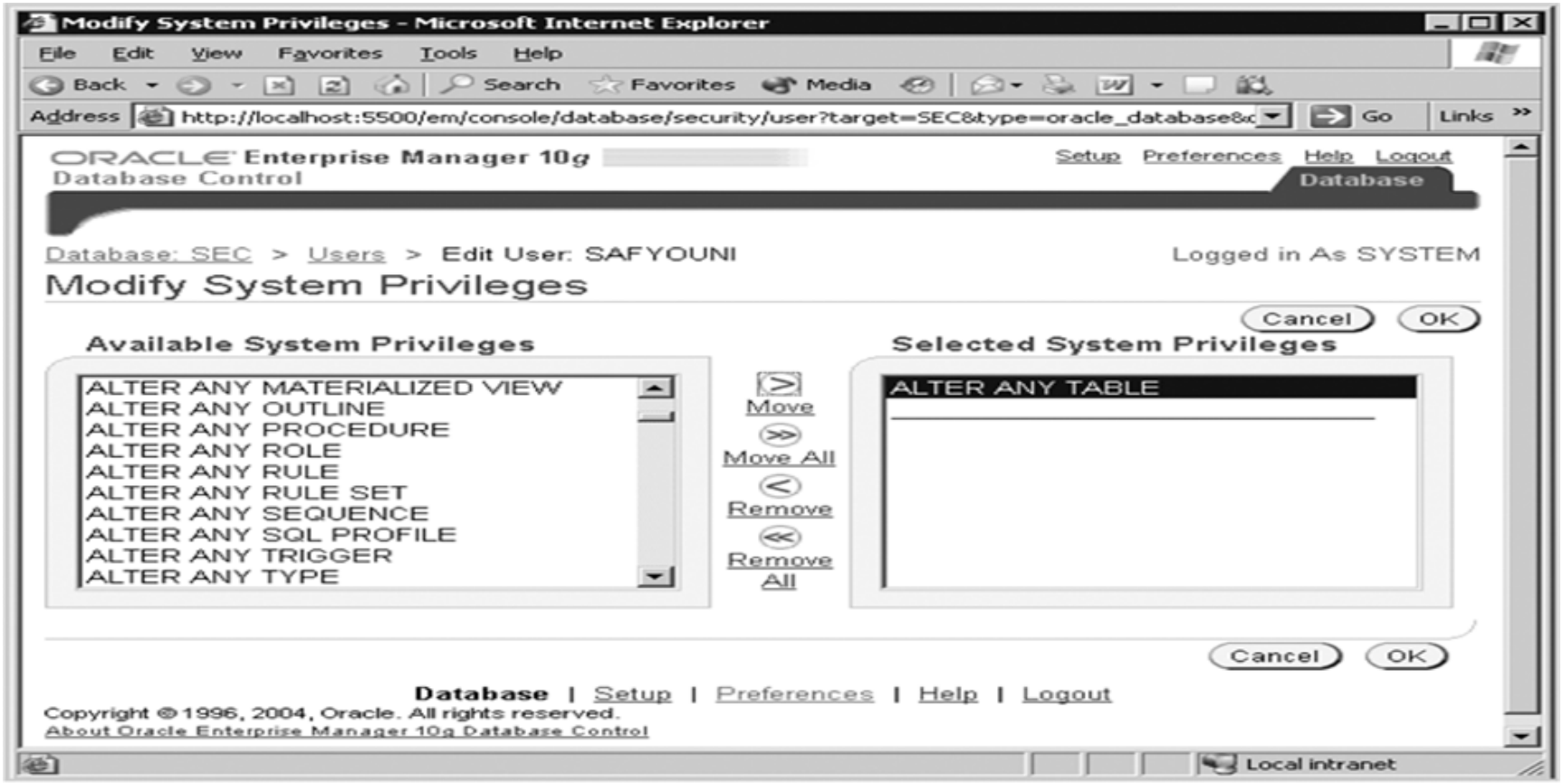
Granting and Revoking User Privileges

- Permit or deny access to data or to perform database operations
- In Oracle:
  - System privileges:
    - Granted only by a database administrator
    - Granted by a user with administration privileges
  - Object privileges:
    - Granted to a user by the schema owner
    - Granted by a user with GRANT privileges

Granting and Revoking User Privileges (continued)

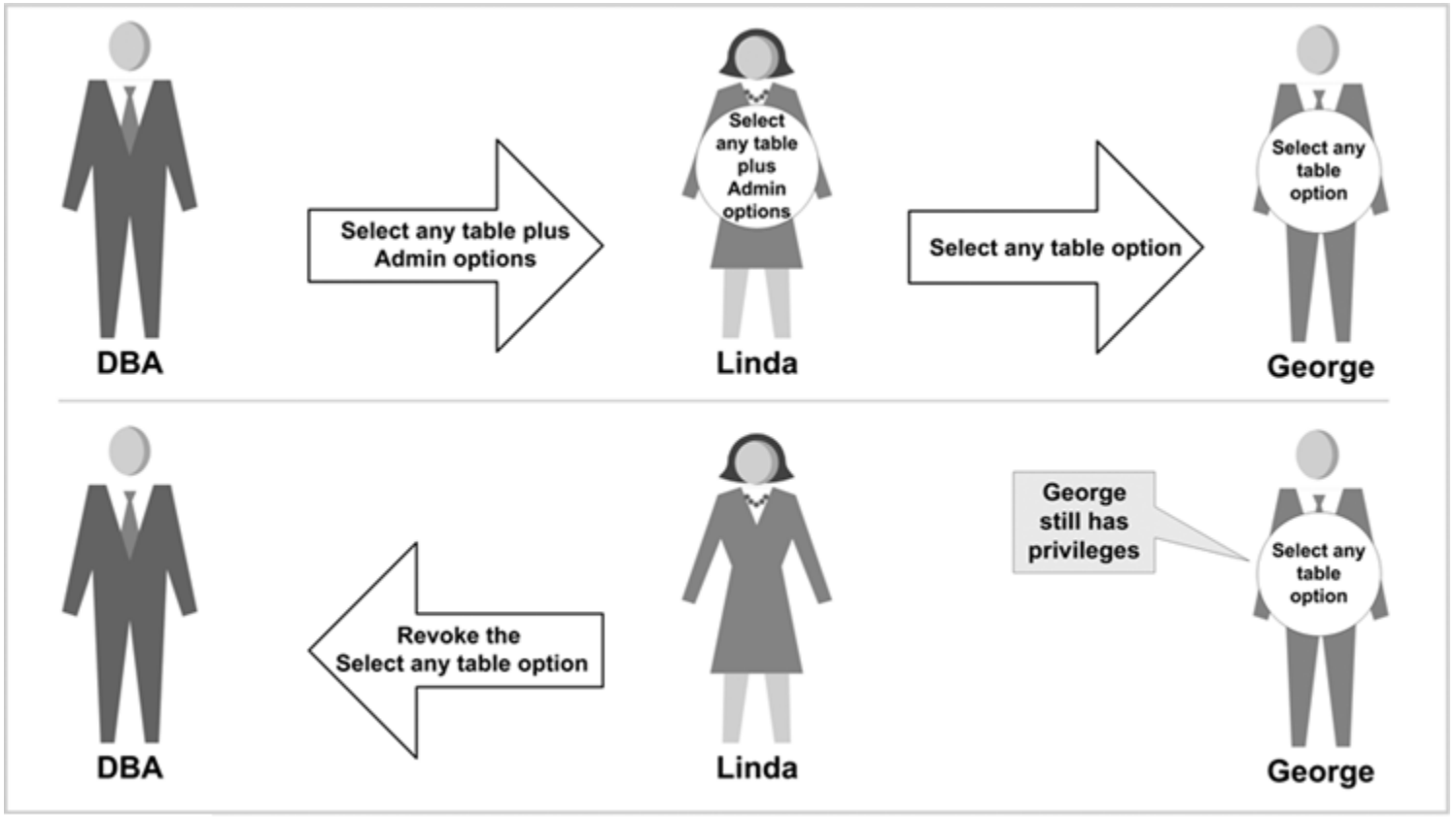
- In Oracle (continued):
  - Grant a privilege using the DCL GRANT statement
  - Revoke a privilege using the DCL REVOKE statement:
    - ADMIN option
    - GRANT option
  - Oracle Enterprise Manager Security

Granting and Revoking User Privileges (continued)



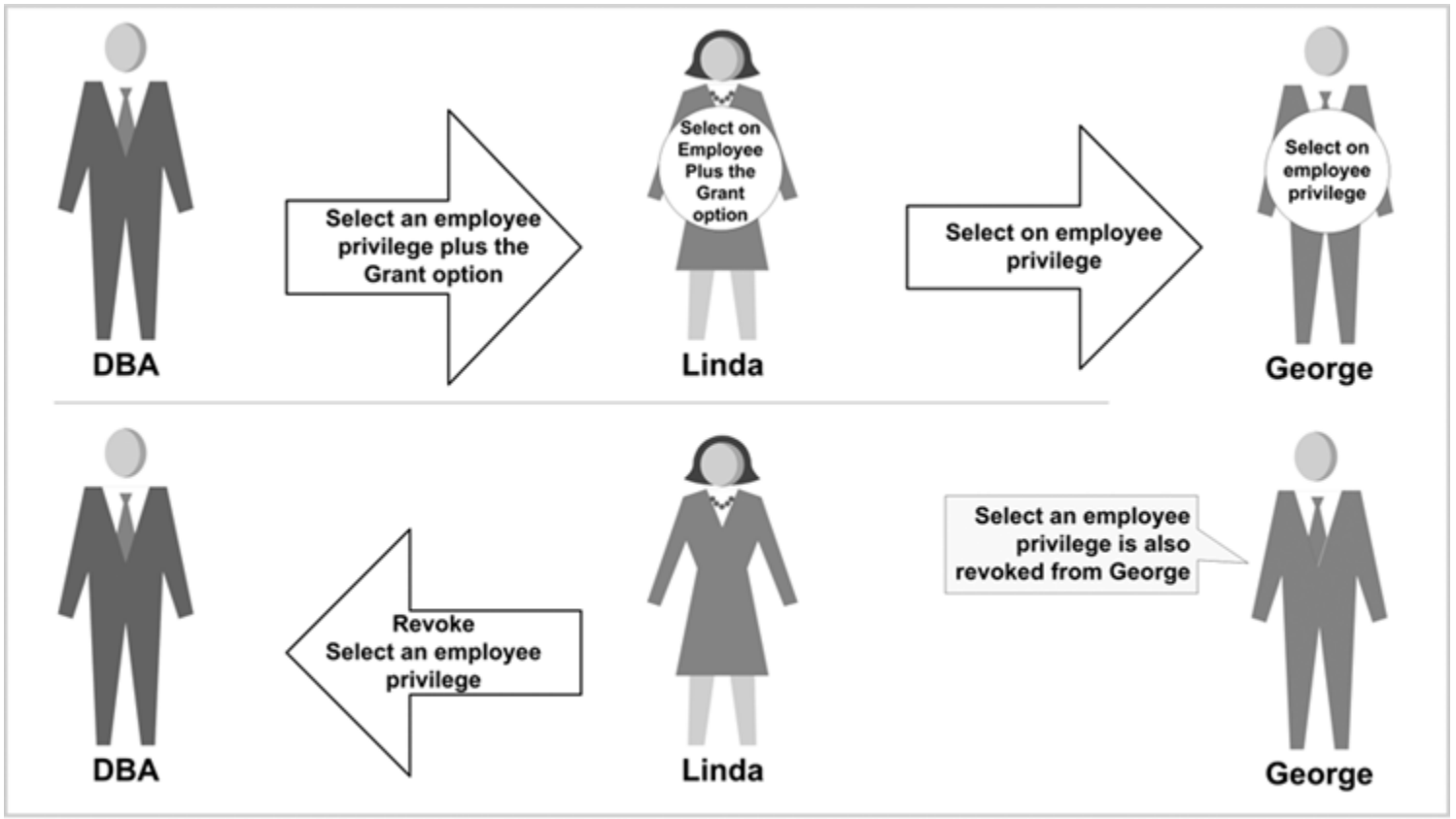
Granting a system privilege to a user

Granting and Revoking User Privileges (continued)



Revoking a system privilege with the ADMIN option

Granting and Revoking User Privileges (continued)



Revoking an object privilege with the GRANT option

Granting and Revoking User Privileges (continued)

- In SQL Server (4 levels); system/server privileges:
  - Sysadmin
  - Serveradmin
  - Setupadmin
  - Securityadmin
  - Processadmin
  - Dbcreator
  - Diskadmin
  - Bulkadmin

Granting and Revoking User Privileges (continued)

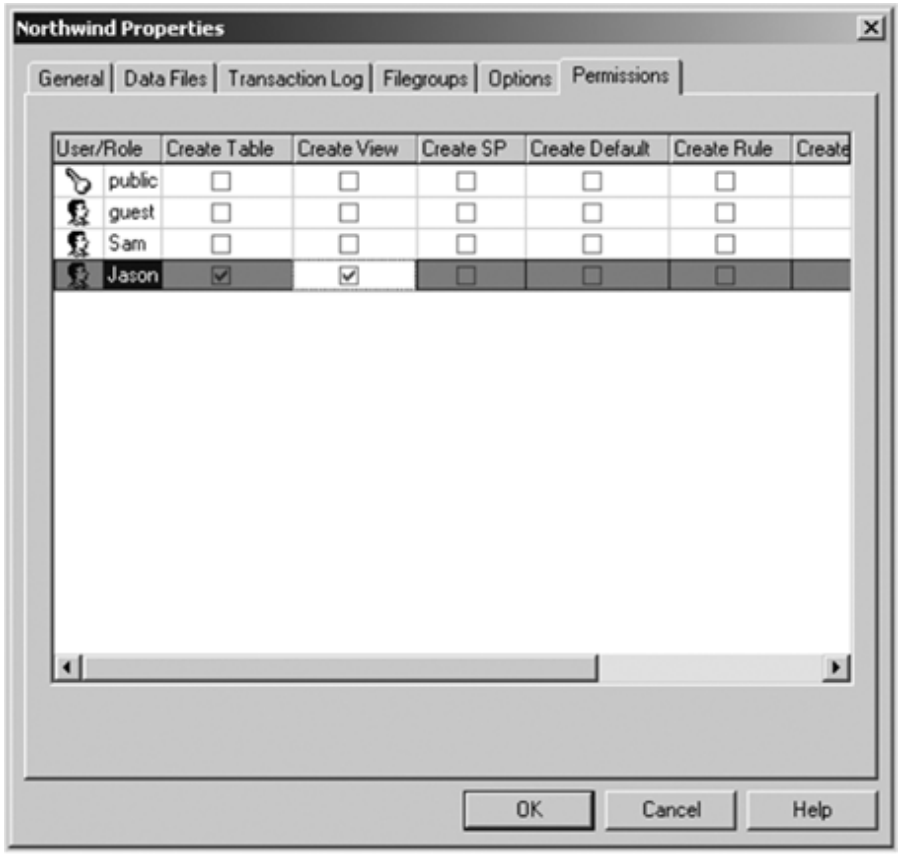
- In SQL Server (continued):
  - Database privileges:
    - Fixed database roles
    - Statement permissions
  - Grant permission using the GRANT statement
  - Revoke permission using the REVOKE statement
  - Enterprise Manager
  - Deny permission using the DENY statement

Granting and Revoking User Privileges (continued)

SQL Server statement permissions

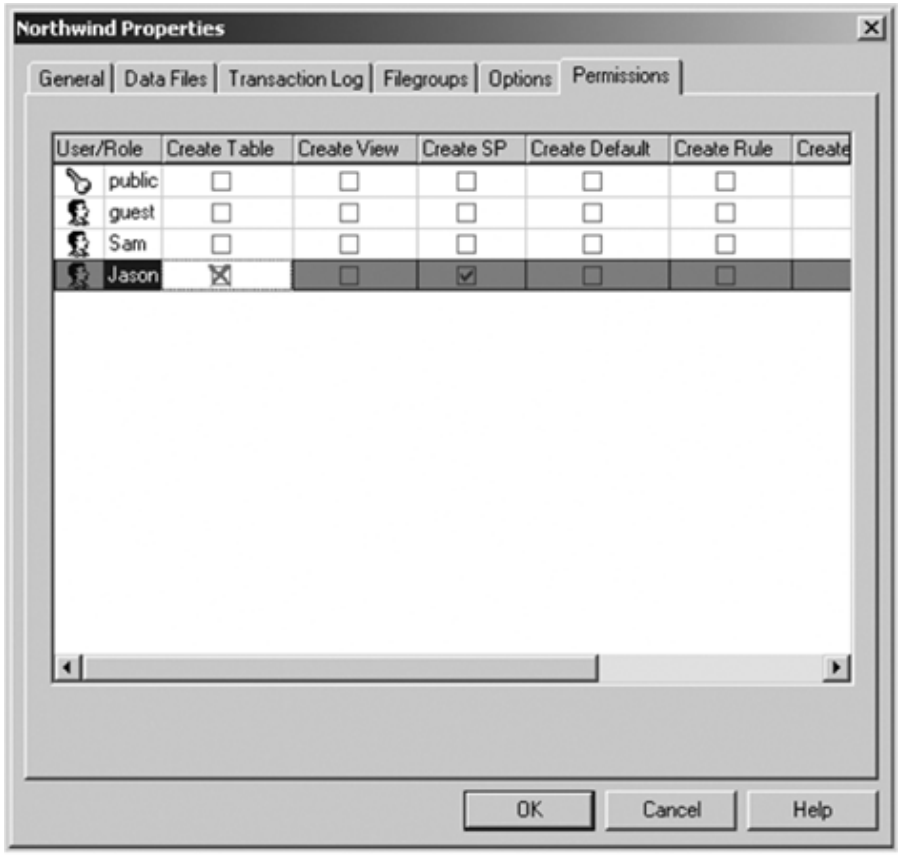
Statement	Permits the User to
CREATE TABLE	Create tables in the database
CREATE VIEW	Create views in the database
CREATE PROCEDURE	Create stored procedures in the database
CREATE FUNCTION	Create functions in the database
CREATE DEFAULT	Create defaults in the database
CREATE RULE	Create rules in the database
BACKUP DATABASE	Back up the database
BACKUP LOG	Back up the database transaction log(s)

Granting and Revoking User Privileges (continued)



Granting user permissions

Granting and Revoking User Privileges (continued)



Revoking database permissions with SQL Server

Granting and Revoking User Privileges (continued)

- In SQL Server:
  - Table and database objects privileges:
    - GRANT, REVOKE, and DENY
    - EXECUTE permission
    - Enterprise Manager (3 methods)
  - Column privileges:
    - GRANT, REVOKE, and DENY
    - Enterprise Manager (2 methods)

Creating, Assigning, and Revoking User Roles

- Role:
  - Used to organize and administer privileges
  - It is like a user, except it cannot own object
  - Can be assigned privileges
  - Can be assigned to users

Creating, Assigning, and Revoking User Roles (continued)

- In Oracle:
  - Create a role using CREATE ROLE statement
  - Assign a role using GRANT statement
  - Oracle Enterprise Manager Roles tool
  - Revoke a role using REVOKE statement
  - Drop a role using DROP statement

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server; user-defined roles:
  - Standard and application
  - Create roles using SP_ADDROLE system-stored procedure
  - Add members to a role using SP_ADDROLEMEMBER stored procedure
  - Drop members from a role using SP_DROPROLEMEMBER stored procedure

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server (continued):
  - User-defined roles (continued):
    - Drop roles using SP_DROPROLE stored procedure
    - Use Enterprise Manager
  - Fixed server roles:
    - Cannot be modified or created
    - Add member to a role using SP_ADDSRVROLEMEMBER stored procedure

Creating, Assigning, and Revoking User Roles (continued)

Description of fixed server roles

SQL Server Fixed Role	Role Description
sysadmin	A super role that allows assigned user to perform any task within SQL Server
serveradmin	Allows assigned user to modify SQL Server configuration.
setupadmin	Allows assigned user to perform specific SQL Server setup, such as linking servers and execution of system stored procedures
securityadmin	Allows assigned user to administer SQL Server logins
processadmin	Allows assigned user to administer SQL Server instance processes
dbcreator	Allows assigned user to create and modify SQL Server database
diskadmin	Allows assigned user to administer database data files
bulkadmin	Allows assigned user to perform BULK INSERT statement

* Information presented in this table is adapted from Microsoft SQL Server 2000 documentation

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server (continued):
  - Fixed server roles (continued):
    - Drop members from a role using SP_DROPSRVROLEMEMBER stored procedure
    - Use Enterprise Manager
  - Fixed database roles:
    - Cannot be modified
    - Give access to database administrative tasks
    - Add members to a role using SP_ADDROLEMEMBER stored procedure

Creating, Assigning, and Revoking User Roles (continued)

SQL Server built-in roles

Fixed Database Role	Description
db_owner	Has all permissions in the database
db_accessadmin	Can add or remove user IDs
db_securityadmin	Can manage all permissions, object ownerships, roles, and role memberships
db_ddladmin	Can issue all DDL, but cannot issue GRANT, REVOKE, or DENY statements
db_backupoperator	Can issue DBCC, CHECKPOINT, and BACKUP statements
db_datareader	Can select all data from any user table in the database
db_datawriter	Can modify any data in any user table in the database
db_denydatareader	Cannot select any data from any user table in the database
db_denydatawriter	Cannot modify any data in any user table in the database

* From Microsoft SQL Server Books Online

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server (continued):
  - Fixed database roles (continued):
    - Drop members from a role using SP_DROPROLEMEMBER stored procedure
    - Use Enterprise Manager
  - Public database role:
    - Cannot be dropped
    - Users automatically belong to this role
    - Users cannot be dropped