



Administración de bases de datos

D07

M-J 7-9 AM



## **Práctica 12: Seguridad**

Centro Universitario de Ciencias Exactas e Ingenierías

Universidad de Guadalajara

2024<sup>a</sup>

## 1. Tablespaces

Para el primer paso se crean las tablespaces RH y VENTAS con la cantidad de 10MB con los siguientes comandos.

CREATE

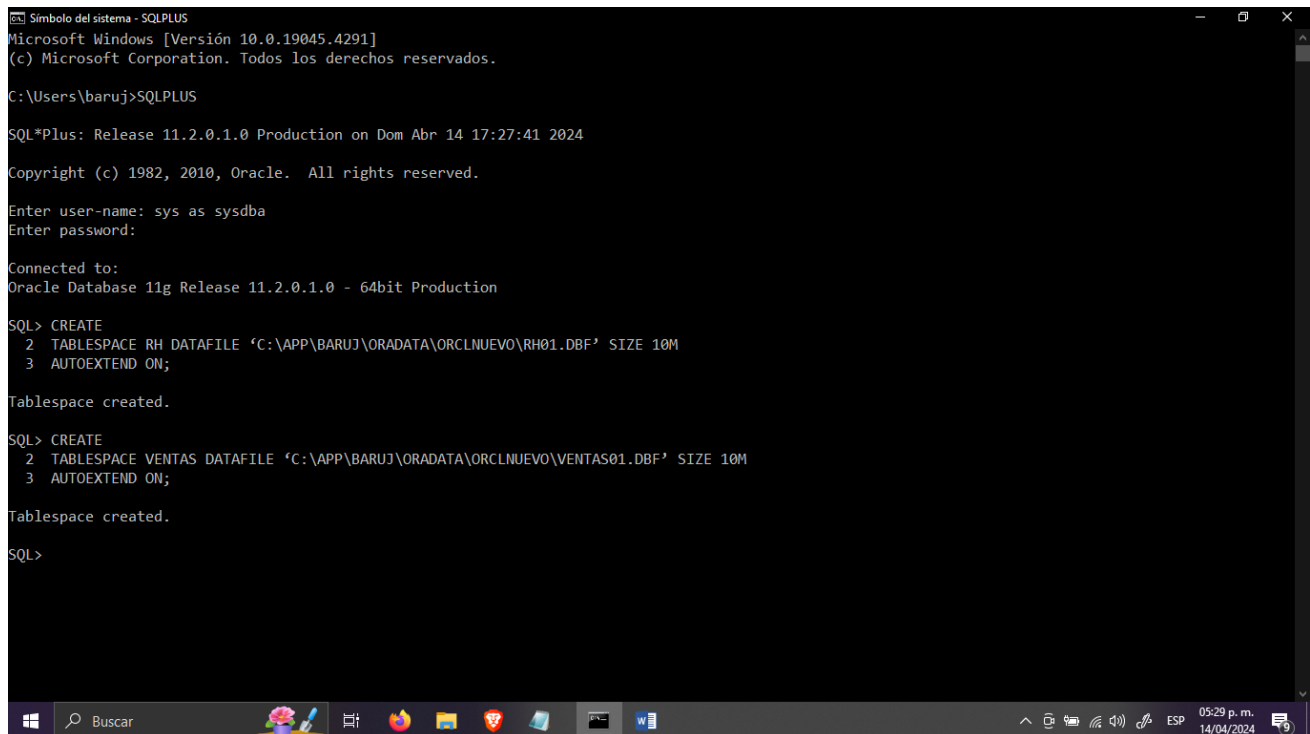
TABLESPACE RH DATAFILE 'C:\APP\BARUJ\ORADATA\ORCLNUEVO\RH01.DBF'  
SIZE 10M

AUTOEXTEND ON;

CREATE

TABLESPACE VENTAS DATAFILE  
'C:\APP\BARUJ\ORADATA\ORCLNUEVO\VENTAS01.DBF' SIZE 10M

AUTOEXTEND ON;



```
Símbolo del sistema - SQLPLUS
Microsoft Windows [Versión 10.0.19045.4291]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\baruj>SQLPLUS

SQL*Plus: Release 11.2.0.1.0 Production on Dom Abr 14 17:27:41 2024

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production

SQL> CREATE
  2  TABLESPACE RH DATAFILE 'C:\APP\BARUJ\ORADATA\ORCLNUEVO\RH01.DBF' SIZE 10M
  3  AUTOEXTEND ON;

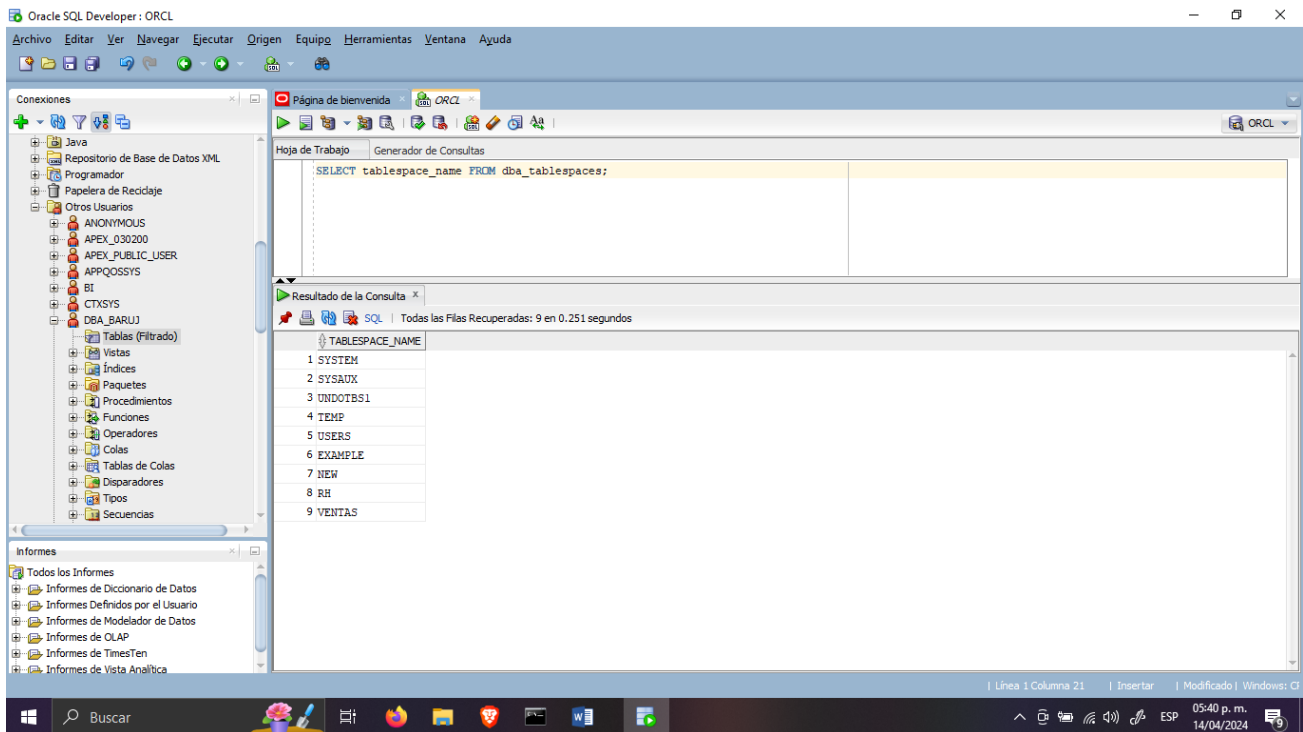
Tablespace created.

SQL> CREATE
  2  TABLESPACE VENTAS DATAFILE 'C:\APP\BARUJ\ORADATA\ORCLNUEVO\VENTAS01.DBF' SIZE 10M
  3  AUTOEXTEND ON;

Tablespace created.

SQL>
```

Así se ven dentro de sql developer las tablas recién creadas.



## 2. Usuarios

Se crean los siguientes usuarios con los permisos mínimos, es decir que solo pueda iniciar sesión.

JUAN

JOSE

JESUS

YO

TATA

SABE

RH\_APP

VENTAS\_APP

CREATE USER JUAN IDENTIFIED BY JUAN;

GRANT CREATE SESSION TO JUAN;

CREATE USER JOSE IDENTIFIED BY JOSE;

GRANT CREATE SESSION TO JOSE;

```
CREATE USER JESUS IDENTIFIED BY JESUS;  
GRANT CREATE SESSION TO JESUS;
```

```
CREATE USER YO IDENTIFIED BY YO;  
GRANT CREATE SESSION TO YO;
```

```
CREATE USER TATA IDENTIFIED BY TATA;  
GRANT CREATE SESSION TO TATA;
```

```
CREATE USER SABE IDENTIFIED BY SABE;  
GRANT CREATE SESSION TO SABE;
```

```
CREATE USER RH_APP IDENTIFIED BY RH_APP;  
GRANT CREATE SESSION TO RH_APP;
```

```
CREATE USER VENTAS_APP IDENTIFIED BY VENTAS_APP;  
GRANT CREATE SESSION TO VENTAS_APP;
```

```
Simbolo del sistema - sqlplus

SQL> CREATE USER JUAN IDENTIFIED BY JUAN;

User created.

SQL> GRANT CREATE SESSION TO JUAN;

Grant succeeded.

SQL>
SQL> CREATE USER JOSE IDENTIFIED BY JOSE;

User created.

SQL> GRANT CREATE SESSION TO JOSE;

Grant succeeded.

SQL>
SQL> CREATE USER JESUS IDENTIFIED BY JESUS;

User created.

SQL> GRANT CREATE SESSION TO JESUS;

Grant succeeded.

SQL> CREATE USER YO IDENTIFIED BY YO;

User created.

SQL> GRANT CREATE SESSION TO YO;

Grant succeeded.
```

```
Seleccionar Símbolo del sistema - sqlplus

Grant succeeded.

SQL> CREATE USER TATA IDENTIFIED BY TATA;

User created.

SQL> GRANT CREATE SESSION TO TATA;

Grant succeeded.

SQL> CREATE USER SABE IDENTIFIED BY SABE;

User created.

SQL> GRANT CREATE SESSION TO SABE;

Grant succeeded.

SQL> CREATE USER RH_APP IDENTIFIED BY RH_APP;

User created.

SQL> GRANT CREATE SESSION TO RH_APP;

Grant succeeded.

SQL> CREATE USER VENTAS_APP IDENTIFIED BY VENTAS_APP;

User created.

SQL> GRANT CREATE SESSION TO VENTAS_APP;

Grant succeeded.

SQL>
```

Para los usuarios RH\_APP y VENTAS\_APP tenemos que asignarles las tablespaces RH y Ventas respectivamente con quota ilimitada, utilizando los siguientes comandos:

VENTAS\_APP

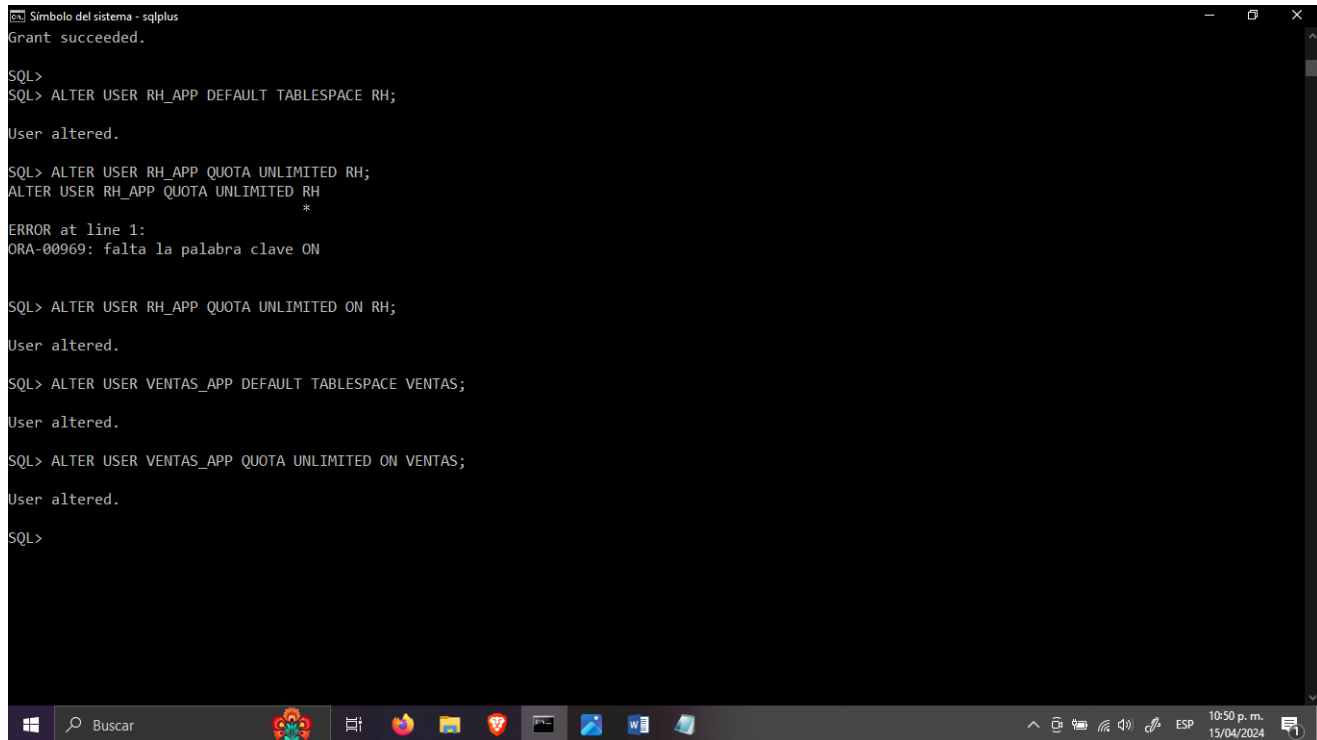
ALTER USER VENTAS\_APP DEFAULT TABLESPACE VENTAS;

```
ALTER USER VENTAS_APP QUOTA UNLIMITED ON VENTAS;
```

RH\_APP

```
ALTER USER RH_APP DEFAULT TABLESPACE RH;
```

```
ALTER USER RH_APP QUOTA UNLIMITED ON RH;
```



```
Simbolo del sistema - sqlplus
Grant succeeded.

SQL>
SQL> ALTER USER RH_APP DEFAULT TABLESPACE RH;

User altered.

SQL> ALTER USER RH_APP QUOTA UNLIMITED RH;
ALTER USER RH_APP QUOTA UNLIMITED RH
*
ERROR at line 1:
ORA-00969: falta la palabra clave ON

SQL> ALTER USER RH_APP QUOTA UNLIMITED ON RH;

User altered.

SQL> ALTER USER VENTAS_APP DEFAULT TABLESPACE VENTAS;

User altered.

SQL> ALTER USER VENTAS_APP QUOTA UNLIMITED ON VENTAS;

User altered.

SQL>
```

Para el usuario YO con el rol DBA:

Se crea el rol dba como DBA\_ROLE para evitar errores con la similitud de nombre, se le conceden todos los privilegios con GRANT ALL PRIVILEGES TO DBA\_ROLE; Se crea el usuario YO y se le da permiso para iniciar sesión, además de concederle el rol con GRANT DBA\_ROLE TO YO;

```
Símbolo del sistema - sqlplus

SQL> CREATE ROLE DBA_ROLE;

Role created.

SQL> GRANT ALL PRIVILEGES TO DBA_ROLE;

Grant succeeded.

SQL> CREATE USER YO IDENTIFIED BY YO;
CREATE USER YO IDENTIFIED BY YO
*
ERROR at line 1:
ORA-01920: conflicto entre el nombre de usuario 'YO' y otro nombre de usuario o rol

SQL> GRANT CREATE SESSION TO YO;

Grant succeeded.

SQL> GRANT DBA_ROLE TO YO;

Grant succeeded.

SQL>
```

### 3. Crear los roles DEV, EXTERNO y DATASC

Se crean los roles con los siguientes comandos.

```
CREATE ROLE DEV;
```

```
CREATE ROLE EXTERNO;
```

```
CREATE ROLE DATASC;
```

```
Símbolo del sistema - sqlplus
SQL>
SQL> CREATE ROLE DEV;

Role created.

SQL> CREATE ROLE EXTERNO;

Role created.

SQL> CREATE ROLE DATASC;

Role created.

SQL>
```

#### 4. Asignar permisos a DEV y asignar DEV a desarrolladores

Otorgamos los permisos DML (Data Manipulation Language): Que incluyen operaciones como INSERT, UPDATE Y DELETE, para otorgar los permisos se usan los siguientes comandos:

```
GRANT SELECT ANY TABLE TO DEV;
```

```
GRANT INSERT ANY TABLE TO DEV;
```

```
GRANT UPDATE ANY TABLE TO DEV;
```

```
GRANT DELETE ANY TABLE TO DEV;
```



```
Símbolo del sistema - sqlplus
SQL> GRANT SELECT ANY TABLE TO DEV;

Grant succeeded.

SQL> GRANT INSERT ANY TABLE TO DEV;

Grant succeeded.

SQL> GRANT UPDATE ANY TABLE TO DEV;

Grant succeeded.

SQL> GRANT DELETE ANY TABLE TO DEV;

Grant succeeded.

SQL>
```

Otorgamos los permisos DDL (Data Definition Language): que incluye operaciones como CREATE, ALTER y DROP sobre los objetos de la base de datos, para otorgar estos permisos usamos:

```
GRANT CREATE ANY TABLE TO DEV;
```

```
GRANT ALTER ANY TABLE TO DEV;
```

```
GRANT DROP ANY TABLE TO DEV;
```

```
Símbolo del sistema - sqlplus

SQL>
SQL> GRANT CREATE ANY TABLE TO DEV;

Grant succeeded.

SQL> GRANT ALTER ANY TABLE TO DEV;

Grant succeeded.

SQL> GRANT DROP ANY TABLE TO DEV;

Grant succeeded.

SQL>
```

Para los usuarios juan, jose y jesus deben ser asignados los roles de DEV, para ello usamos el comando:

GRANT DEV TO JUAN;

GRANT DEV TO JOSE;

GRANT DEV TO JESUS;

```
Símbolo del sistema - sqlplus
Grant succeeded.

SQL> GRANT DEV TO JUAN;

Grant succeeded.

SQL> GRANT DEV TO JOSE;

Grant succeeded.

SQL> GRANT DEV TO JESUS;

Grant succeeded.

SQL>
```

## 5. Crear Tablas con una cuenta de DEV

Nos conectamos con la cuenta de JUAN que tiene el ROL de DEV.

```
Símbolo del sistema - sqlplus

Enter user-name: JUAN
Enter password:

Connected to:
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production

SQL> SELECT SESSION_USER
      2 FROM dual;
SELECT SESSION_USER
      *
ERROR at line 1:
ORA-00904: "SESSION_USER": identificador no válido

SQL> SELECT USER
      2 FROM dual;

USER
-----
JUAN

SQL> SELECT role
      2 FROM session_roles;

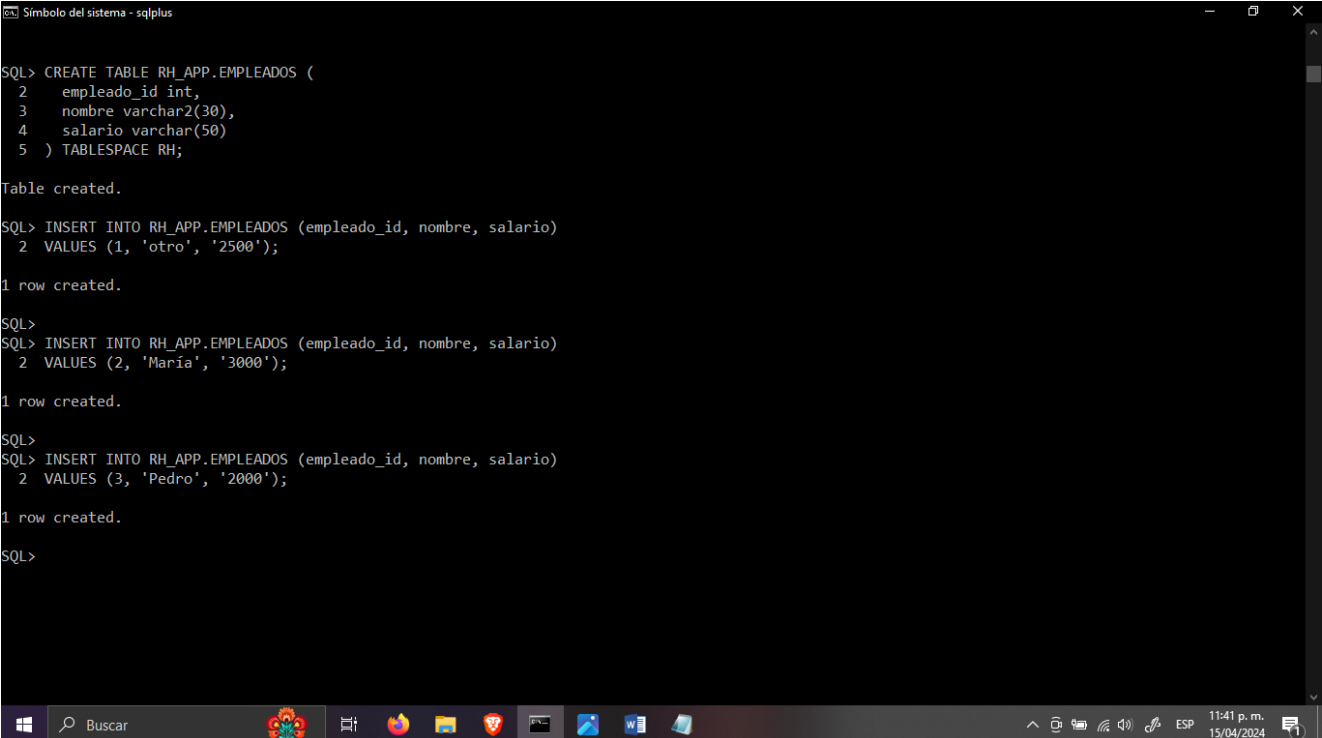
ROLE
-----
DEV

SQL>
```

Con el siguiente comando creamos la tabla de empleados, además de agregar algunos datos

```
CREATE TABLE RH_APP.EMPLEADOS (  
    empleado_id int,  
    nombre varchar2(30),  
    salario varchar  
) TABLESPACE RH;
```

Además se insertan algunos datos dentro de la tabla.



```
Símbolo del sistema - sqlplus  
  
SQL> CREATE TABLE RH_APP.EMPLEADOS (  
2     empleado_id int,  
3     nombre varchar2(30),  
4     salario varchar(50)  
5 ) TABLESPACE RH;  
  
Table created.  
  
SQL> INSERT INTO RH_APP.EMPLEADOS (empleado_id, nombre, salario)  
2 VALUES (1, 'otro', '2500');  
  
1 row created.  
  
SQL>  
SQL> INSERT INTO RH_APP.EMPLEADOS (empleado_id, nombre, salario)  
2 VALUES (2, 'Maria', '3000');  
  
1 row created.  
  
SQL>  
SQL> INSERT INTO RH_APP.EMPLEADOS (empleado_id, nombre, salario)  
2 VALUES (3, 'Pedro', '2000');  
  
1 row created.  
  
SQL>
```

Revisamos la tabla creada con `SELECT * FROM RH_APP.EMPLEADOS;`

```
Símbolo del sistema - sqlplus
ORA-00942: la tabla o vista no existe

SQL> SELECT * FROM RH_APP.EMPLEADOS;

EMPLEADO_ID NOMBRE
-----
SALARIO
-----
1 otro
2500
2 María
3000
3 Pedro
2000

SQL>
```

En la tablespace RH es donde se harán las siguientes operaciones:

Se crea la siguiente tabla PUESTO y se agregan datos con los siguientes comandos:

```
CREATE TABLE RH_APP.PUESTOS (
    puesto_id int,
    puesto varchar2(20)
) TABLESPACE RH;
```

```
INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (1, 'Gerente');
INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (2, 'Secretario');
INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (3, 'Vendedor');
```

```
Símbolo del sistema - sqlplus
2000

SQL> CREATE TABLE RH_APP.PUESTOS (
  2   puesto_id int,
  3   puesto varchar2(20)
  4 ) TABLESPACE RH;

Table created.

SQL>
SQL> INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (1, 'Gerente');

1 row created.

SQL>
SQL> INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (2, 'Secretario');

1 row created.

SQL>
SQL> INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (3, 'Vendedor');

1 row created.

SQL>
```

Se crea la siguiente tabla TURNO:

```
CREATE TABLE RH_APP. TURNO (
    turno_id int,
    turnon varchar2(20)
) TABLESPACE RH;
```

```
Símbolo del sistema - sqlplus
3000
      3 Pedro
2000

SQL> CREATE TABLE RH_APP.PUESTOS (
  2     puesto_id int,
  3     puesto varchar2(20)
  4 ) TABLESPACE RH;

Table created.

SQL>
SQL> INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (1, 'Gerente');

1 row created.

SQL>
SQL> INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (2, 'Secretario');

1 row created.

SQL>
SQL> INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (3, 'Vendedor');

1 row created.

SQL> CREATE TABLE RH_APP.TURNO (
  2     turno_id int,
  3     turnon varchar2(20)
  4 ) TABLESPACE RH;

Table created.
```

Se insertan datos en la tabla RH\_APP.TURNO

```
Símbolo del sistema - sqlplus
*
ERROR at line 1:
ORA-00904: "TURNON": identificador no v8lido

SQL> INSERT INTO RH_APP.TURNO (turno_id, turnon) VALUES (1, 'Matutino');

1 row created.

SQL>
SQL> INSERT INTO RH_APP.TURNO (turno_id, turnon) VALUES (2, 'Vespertino');

1 row created.

SQL>
SQL> INSERT INTO RH_APP.TURNO (turno_id, turnon) VALUES (3, 'Nocturno');

1 row created.

SQL>
```

En la tablespace VENTAS se harán las siguientes operaciones:

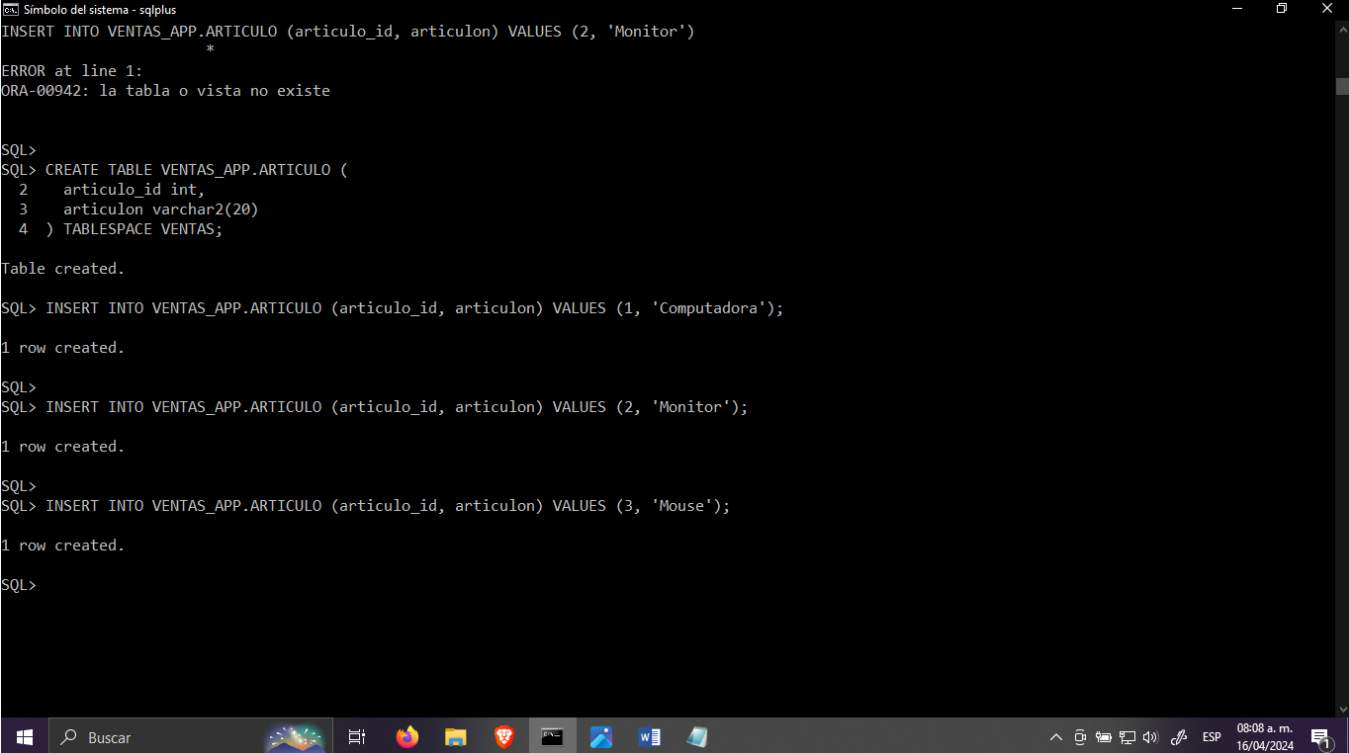
Se crea la tabla ARTICULO y se insertan varios datos:

```
CREATE TABLE VENTAS_APP.ARTICULO (  
    articulo_id int,  
    articulon varchar2(20)  
) TABLESPACE VENTAS;
```

```
INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (1,  
'Computadora');
```

```
INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (2, 'Monitor');
```

```
INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (3, 'Mouse');
```



```
Simbolo del sistema - sqlplus  
INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (2, 'Monitor')  
*  
ERROR at line 1:  
ORA-00942: la tabla o vista no existe  
  
SQL>  
SQL> CREATE TABLE VENTAS_APP.ARTICULO (  
2   articulo_id int,  
3   articulon varchar2(20)  
4 ) TABLESPACE VENTAS;  
  
Table created.  
  
SQL> INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (1, 'Computadora');  
  
1 row created.  
  
SQL>  
SQL> INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (2, 'Monitor');  
  
1 row created.  
  
SQL>  
SQL> INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (3, 'Mouse');  
  
1 row created.  
  
SQL>
```

Se crea la tabla FACTURA y se insertan tres datos con los siguientes comandos:

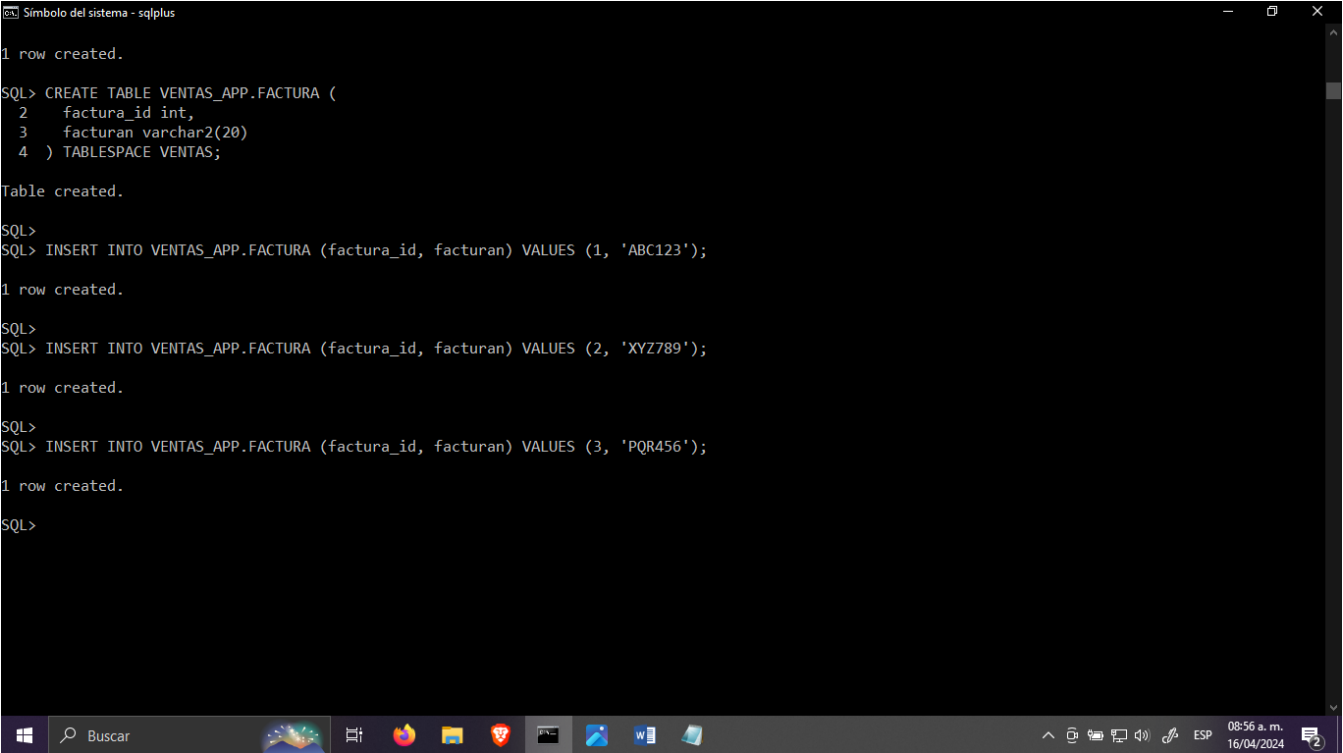
```
CREATE TABLE VENTAS_APP.FACTURA (  
    factura_id int,  
    facturan varchar2(20)  
) TABLESPACE VENTAS;
```

```
INSERT INTO VENTAS_APP.FACTURA (factura_id, facturan) VALUES (1, 'ABC123');
```



```
INSERT INTO VENTAS_APP.FACTURA (factura_id, facturan) VALUES (2, 'XYZ789');
```

```
INSERT INTO VENTAS_APP.FACTURA (factura_id, facturan) VALUES (3, 'PQR456');
```



```
Simbolo del sistema - sqlplus

1 row created.

SQL> CREATE TABLE VENTAS_APP.FACTURA (
  2   factura_id int,
  3   facturan varchar2(20)
  4 ) TABLESPACE VENTAS;

Table created.

SQL>
SQL> INSERT INTO VENTAS_APP.FACTURA (factura_id, facturan) VALUES (1, 'ABC123');

1 row created.

SQL>
SQL> INSERT INTO VENTAS_APP.FACTURA (factura_id, facturan) VALUES (2, 'XYZ789');

1 row created.

SQL>
SQL> INSERT INTO VENTAS_APP.FACTURA (factura_id, facturan) VALUES (3, 'PQR456');

1 row created.

SQL>
```

Se crea la tabla de DETALLE y se insertan tres datos:

```
CREATE TABLE VENTAS_APP.DETALLE (
    detalle_id int,
    detallan varchar2(30)
) TABLESPACE VENTAS;
```

```
INSERT INTO VENTAS_APP.DETALLE (detalle_id, detallan) VALUES (1, 'Computadora
marca acer');
```

```
INSERT INTO VENTAS_APP.DETALLE (detalle_id, detallan) VALUES (2, 'Monitor 27
pulgadas');
```

```
INSERT INTO VENTAS_APP.DETALLE (detalle_id, detallan) VALUES (3, 'Mouse
inalambrico');
```

```
Símbolo del sistema - sqlplus
SQL> INSERT INTO VENTAS_APP.FACTURA (factura_id, facturan) VALUES (3, 'PQR456');

1 row created.

SQL> CREATE TABLE VENTAS_APP.DETALLE (
  2  detalle_id int,
  3  detallan varchar2(30)
  4 ) TABLESPACE VENTAS;

Table created.

SQL> INSERT INTO VENTAS_APP.DETALLE (detalle_id, detallan) VALUES (1, 'Computadora marca acer');

1 row created.

SQL>
SQL> INSERT INTO VENTAS_APP.DETALLE (detalle_id, detallan) VALUES (2, 'Monitor 27 pulgadas');

1 row created.

SQL>
SQL> INSERT INTO VENTAS_APP.DETALLE (detalle_id, detallan) VALUES (3, 'Mouse inalambrico');

1 row created.

SQL>
SQL>
```

## 6. Permisos EXTERNO y DATA y asignar a usuarios

Para empezar se asignan los permisos al rol DATASC, estos permisos solo podrán hacer SELECT a las tablas de RH y VENTAS, ninguna modificación:

```
GRANT SELECT ON RH_APP.EMPLEADOS TO DATASC;
GRANT SELECT ON RH_APP.PUESTOS TO DATASC;
GRANT SELECT ON RH_APP.TURNO TO DATASC;
GRANT SELECT ON VENTAS_APP.ARTICULO TO DATASC;
GRANT SELECT ON VENTAS_APP.FACTURA TO DATASC;
GRANT SELECT ON VENTAS_APP.DETALLE TO DATASC;
```

```
Símbolo del sistema - sqlplus

SQL> GRANT SELECT ON RH_APP.EMPLEADOS TO DATASC;
Grant succeeded.

SQL> GRANT SELECT ON RH_APP.PUESTOS TO DATASC;
Grant succeeded.

SQL> GRANT SELECT ON RH_APP.TURNO TO DATASC;
Grant succeeded.

SQL> GRANT SELECT ON VENTAS_APP.ARTICULO TO DATASC;
Grant succeeded.

SQL> GRANT SELECT ON VENTAS_APP.FACTURA TO DATASC;
Grant succeeded.

SQL> GRANT SELECT ON VENTAS_APP.DETALLE TO DATASC;
Grant succeeded.

SQL>
```

Después se asignan los permisos al rol de EXTERNO, estos permisos le permiten hacer INSERT y SELECT en las tablas de VENTAS, por lo que las tablas de RH no podrán ser modificadas:

```
GRANT SELECT ON VENTAS_APP.ARTICULO TO EXTERNO;
GRANT SELECT ON VENTAS_APP.FACTURA TO EXTERNO;
GRANT SELECT ON VENTAS_APP.DETALLE TO EXTERNO;
```

```
GRANT INSERT ON VENTAS_APP.ARTICULO TO EXTERNO;
GRANT INSERT ON VENTAS_APP.FACTURA TO EXTERNO;
GRANT INSERT ON VENTAS_APP.DETALLE TO EXTERNO;
```

```
Simbolo del sistema - sqlplus
SQL>
SQL>
SQL>
SQL> GRANT SELECT ON VENTAS_APP.ARTICULO TO EXTERNO;

Grant succeeded.

SQL> GRANT SELECT ON VENTAS_APP.FACTURA TO EXTERNO;

Grant succeeded.

SQL> GRANT SELECT ON VENTAS_APP.DETALLE TO EXTERNO;

Grant succeeded.

SQL>
SQL> GRANT INSERT ON VENTAS_APP.ARTICULO TO EXTERNO;

Grant succeeded.

SQL> GRANT INSERT ON VENTAS_APP.FACTURA TO EXTERNO;

Grant succeeded.

SQL> GRANT INSERT ON VENTAS_APP.DETALLE TO EXTERNO;

Grant succeeded.

SQL>
```

Otorgamos a los usuarios TATA y SABE los roles correspondientes:

GRANT EXTERNO TO TATA;

GRANT DATASC TO SABE;

```
Símbolo del sistema - sqlplus

SQL> GRANT EXTERNO TO TATA;

Grant succeeded.

SQL> GRANT DATASC TO SABLE;

Grant succeeded.

SQL>
```

Conectándonos con TATA que tiene rol EXTERNO, podemos hacer SELECT a la tabla de ARTICULO, se hace INSERT y se hace SELECT para verificar el dato.

```
Símbolo del sistema - sqlplus

Grant succeeded.

SQL> CONN TATA/TATA;
Connected.
SQL> SELECT USER
  2 FROM dual;

USER
-----
TATA

SQL> SELECT * FROM VENTAS_APP.ARTICULO;

ARTICULO_ID ARTICULON
-----
1 Computadora
2 Monitor
3 Mouse

SQL> INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulon) VALUES (4, 'Audifonos');

1 row created.

SQL> SELECT * FROM VENTAS_APP.ARTICULO;

ARTICULO_ID ARTICULON
-----
1 Computadora
2 Monitor
3 Mouse
4 Audifonos

SQL>
```

Sin embargo, si se quiere hacer SELECT a otras tablas fuera de los permisos del rol EXTERNO, esto nos dará error.

```
Símbolo del sistema - sqlplus
3 Mouse
4 Audifonos

SQL> SELECT * FROM RH_APP.EMPLEADOS;
SELECT * FROM RH_APP.EMPLEADOS
*
ERROR at line 1:
ORA-00942: la tabla o vista no existe

SQL> SELECT * FROM RH_APP.PUESTOS;
SELECT * FROM RH_APP.PUESTOS
*
ERROR at line 1:
ORA-00942: la tabla o vista no existe

SQL> SELECT * FROM RH_APP.TURNO;
SELECT * FROM RH_APP.TURNO
*
ERROR at line 1:
ORA-00942: la tabla o vista no existe

SQL>
```

Al conectarnos con el usuario SABA que tiene el rol DATASC, podemos verificar ambas tablas EMPLEADOS de RH y ARTICULO DE VENTAS, las cuales pueden revisarse con SELECT.

```
Símbolo del sistema - sqlplus
SQL> CONN SABA/SABA;
Connected.
SQL> SELECT USER FROM DUAL;

USER
-----
SABA

SQL> SELECT * FROM RH_APP.EMPLEADOS;

EMPLEADO_ID NOMBRE
-----
SALARIO
-----
1 otro
2500
2 María
3000
3 Pedro
2000

SQL> SELECT * FROM VENTAS_APP.ARTICULO;

ARTICULO_ID ARTICULON
-----
1 Computadora
2 Monitor
3 Mouse
4 Audifonos

SQL>
```

Sin embargo, debido a los permisos del rol, acciones como el INSERT darán error

```
Símbolo del sistema - sqlplus

1 Computadora
2 Monitor
3 Mouse
4 Audífonos

SQL> SELECT * FROM RH_APP.PUESTOS;

PUESTO_ID PUESTO
-----
1 Gerente
2 Secretario
3 Vendedor

SQL> SELECT * FROM VENTAS_APP.FACTURA;INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (4, 'Repartidor');
SELECT * FROM VENTAS_APP.FACTURA;INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (4, 'Repartidor')
*
ERROR at line 1:
ORA-00911: car cter no v lido

SQL> INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (4, 'Repartidor');
INSERT INTO RH_APP.PUESTOS (puesto_id, puesto) VALUES (4, 'Repartidor')
*
ERROR at line 1:
ORA-01031: privilegios insuficientes

SQL> INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulo) VALUES (5, 'Smarthphone');
INSERT INTO VENTAS_APP.ARTICULO (articulo_id, articulo) VALUES (5, 'Smarthphone')
*
ERROR at line 1:
ORA-01031: privilegios insuficientes

SQL>
```

Otro ejemplo de estos permisos, es que no se puede hacer DROP de una tabla por tener permisos insuficientes.

```
Símbolo del sistema - sqlplus

ORA-01031: privilegios insuficientes

SQL> DROP TABLE RH_APP.EMPLEADOS;
DROP TABLE RH_APP.EMPLEADOS
*
ERROR at line 1:
ORA-01031: privilegios insuficientes

SQL>
```