

Base de Datos II (1)

Resumen: Clases Emilson Acosta

Sintaxis a Aprender : Emilson

- Crear Usuario y Asignar Privilegios

```
-- CREANDO USUARIO
/*Se recomienda usar una contraseña sencilla, para el usuario.*/
CREATE USER NOMBRE_USUARIO IDENTIFIED BY CONTRASENIA
/*Es donde guardamos los datos del usuario,
users ya viene por defecto en oracle.*/
DEFAULT TABLESPACE users
/*Aqui almacenaremos los objetos que use o cree, en una sesión
con el usuario creado previamente.*/
TEMPORARY TABLESPACE temp
/*Es la capacidad de objetos que el usuario puede crear.*/
QUOTA UNLIMITED ON users;

-- CREANDO ROL
/*El rol consiste en una agrupacion de varios privilegios, luego se le
puede asignar a varios usuarios*/
CREATE ROLE NOMBRE_DEL_ROL;

-- ASIGNANDO PRIVILEGIOS
/**/
GRANT CREATE SESSION, CREATE ANY TABLE, CREATE ANY PROCEDURE,
ALTER ANY TABLE, ALTER ANY PROCEDURE, DROP ANY TABLE,
DROP ANY PROCEDURE TO NOMBRE_DEL_ROL;

-- ASIGNANDO ROL A USUARIO
GRANT NOMBRE_DEL_ROL TO NOMBRE_USUARIO;

-- CONSULTAR ROLES QUE SE LE ASIGNARON A UN USUARIO
/*Se debe de hacer desde la conexión de SYSTEM*/
SELECT * FROM DBA_ROLES;

-- CONSULTA PARA VERIFICAR LOS PRIVILEGIOS ASIGNADOS A UN ROL
/*Se debe de hacer desde la conexión de SYSTEM*/
SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE='NOMBRE_DEL_ROL';
/* Consulta para verificar los privilegios de insert, update y delete, que tiene un rol*/
SELECT * FROM DBA_TAB_PRIVS WHERE GRANTEE='NOMBRE_DEL_ROL';

-- ASIGNAR PRIVILEGIOS A ROLES
GRANT SELECT ON NOMBRE_CONEXION.NOMBRE_TABLA TO NOMBRE_DEL_ROL;
GRANT INSERT ON NOMBRE_CONEXION.NOMBRE_TABLA TO NOMBRE_DEL_ROL;
GRANT DELETE ON NOMBRE_CONEXION.NOMBRE_TABLA TO NOMBRE_DEL_ROL;
GRANT UPDATE ON NOMBRE_CONEXION.NOMBRE_TABLA TO NOMBRE_DEL_ROL;
-- ó
GRANT INSERT, UPDATE, DELETE ON NOMBRE_CONEXION.NOMBRE_TABLA TO NOMBRE_DEL_ROL;

-- PARA REVOCAR PRIVILEGIOS
/*Se debe hacer desde el usuario de SYSTEM*/
/*Eliminando privilegio sobre un rol*/
REVOKE NOMBRE_PRIVILEGIO[CREATE ANY PROCEDURE|ALTER ANY PROCEDURE|DROP ANY PROCEDURE] FROM NOMBRE_DEL_ROL;
```

```
/*Eliminando privilegio select sobre un rol*/  
REVOKE SELECT ON NOMBRE_CONEXION.NOMBRE_TABLA FROM NOMBRE_DEL_ROL;
```

- Bloques Anónimos

```
-- Estructura Básica  
/* Para imprimir en pantalla usamos*/  
SET SERVEROUTPUT ON;  
DECLARE  
  /*Aquí se declararán las variables a utilizar.  
   Este parte es opcional*/  
  NOMBRE_VARIABLE TIPO_VARIABLE;  
  -- para asignar valor a una variable se utiliza :=  
BEGIN  
  /*Instrucciones*/  
  DBMS_OUTPUT.PUT_LINE('Lo que queremos imprimir en pantalla'||NOMBRE_VARIABLE)  
END;
```

- IF | CASE

IF:

```
IF (CONDICION) THEN  
  INSTRUCCIONES;  
ELSIF (CONDICION) THEN  
  INSTRUCCIONES;  
ELSE  
  INSTRUCCIONES;  
END IF;
```

SWITHC-CASE:

```
CASE  
  WHEN (CONDICION) THEN  
    INSTRUCCIONES;  
  WHEN (CONDICION) THEN  
    INSTRUCCIONES;  
  WHEN (CONDICION) THEN  
    INSTRUCCIONES;  
  ELSE  
    INSTRUCCIONES;  
END CASE;
```

- CICLOS

LOOP:

```
i NUMBER:=0;  
  
LOOP  
  EXIT WHEN (i>3);  
  INSTRUCCIONES;
```

```
i:=i+1;  
END LOOP;
```

WHILE:

```
WHILE (CONDICION) LOOP  
  INSTRUCCIONES;  
END LOOP;
```

FOR:

```
FOR i IN [LISTA] LOOP  
  INSTRUCCIONES  
END LOOP;
```

- **SELECT INTO**

Almacenamos en una variable los valores de los campos que queremos obtener en la consulta. Este

```
DECLARE  
  NOMBRE_VAR NOMBRE_TABLA.NOMBRE_CAMPO%TYPE; --%TYPE Toma el tipo de dato del campo.  
BEGIN  
  SELECT NOMBRE_COLUMNNA INTO NOMBRE_VAR FROM NOMBRE_TABLA WHEN (CONDICION);  
END;
```

- **CURSORES**

- **Cursores Implícitos:** se llaman SQL y tienen diferentes Propiedades

- **SQL%ISOPEN:** True si el cursor está abierto.

- **SQL%FOUND:** True si devuelve una fila

- **SQL%NOTFOUND:** True si no se devuelve ninguna fila

- **SQL%ROWCOUNT:** Proporciona el número total de filas devueltas hasta ese momento

- **FUNCTION**

- **EXCEPCIONES**

- **PROCEDURE**