Bases de Datos Avanzadas

Práctica no. 1:

Instalación y uso de un sistema de bases de datos relacional (SBDR)

**Número de Práctica: 1**

**Nombre de la Práctica: Instalación y uso de un sistema de bases de datos (SBDR)**

REQUISITOS PREVIOS

Para la realización de la práctica se deberá tener instalado en la computadora el software:

* Oracle XE (Express Edition), proporcionado por el profesor.

MARCO TEÓRICO

La documentación se puede encontrar en la dirección [Oracle Database Express Edition](https://www.oracle.com/mx/database/technologies/appdev/xe.html)

DESARROLLO DE LA PRÁCTICA:

1.- Para poder realizar la práctica es necesario descargar el archivo [OracleXE11g.zip](https://drive.google.com/file/d/1P9EVa3eU5Fp3v80ojSSzDFpSzoH6nMaH/view?usp=drive_web&authuser=0) y guardarlo en una carpeta temporal. Descomprimir el archivo descargado y ejecutar el instalador (**setup.exe**) localizado en la carpeta *DISK1*, tal y como se muestra en la figura 1.

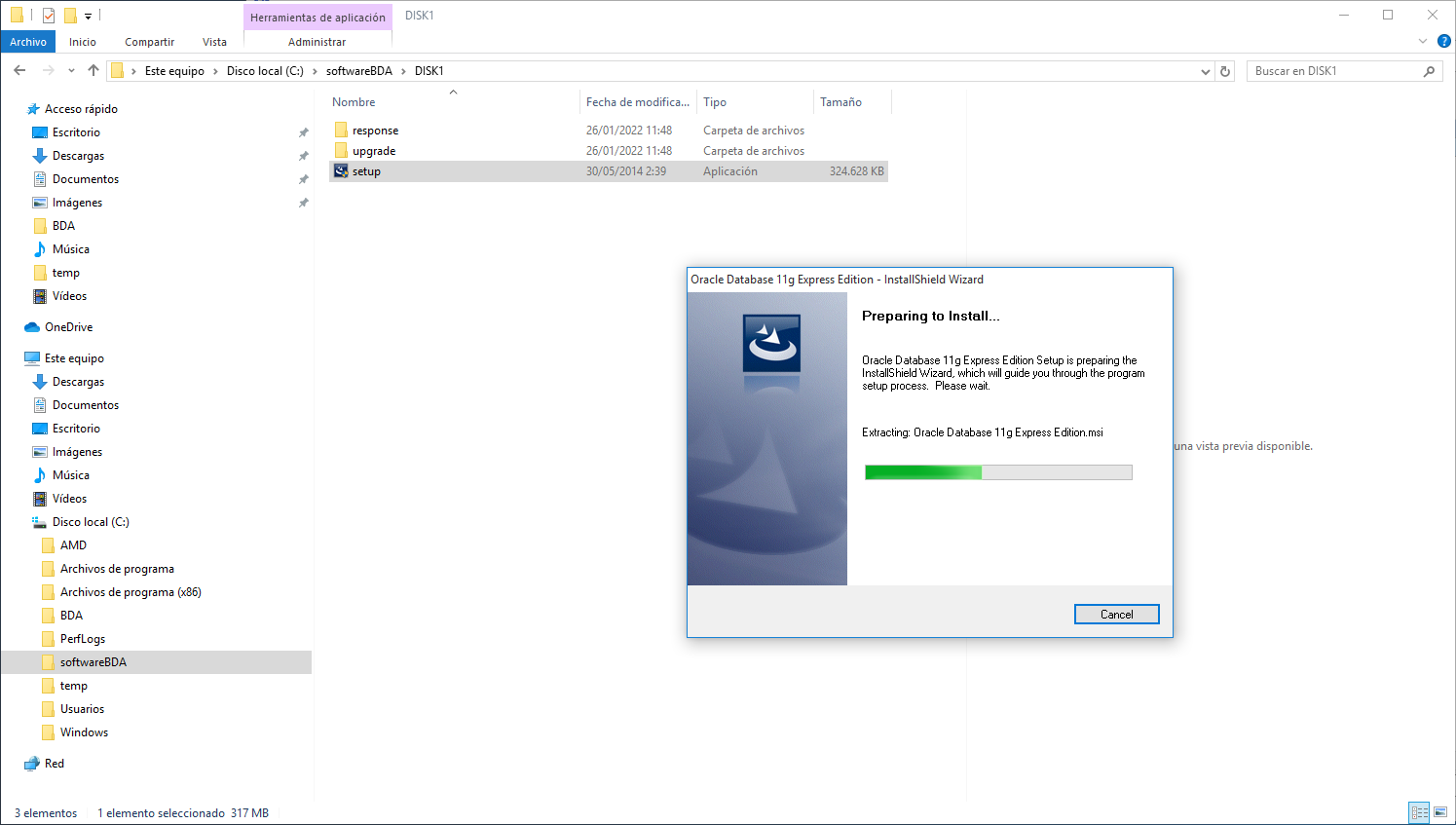


Figura 1. Ejecución del instalador de Oracle XE

2.- La siguiente pantalla muestra el acuerdo de licencia del software. Es necesario aprobar para poder continuar (figura 2).

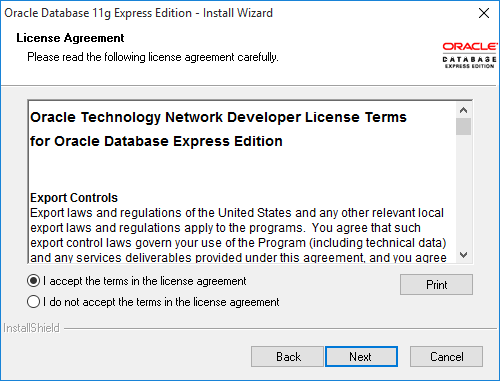


Figura 2. Acuerdo de licencia del software Oracle XE.

3.- La pantalla de la figura 3 muestra la ruta de instalación del software. Se recomienda no modificar y continuar mediante el botón **Next** (figura 3).

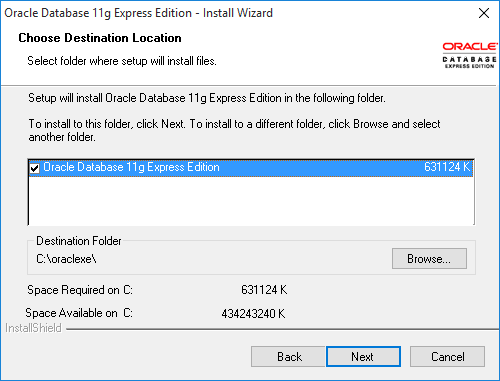


Figura 3. Ruta de instalación por omisión del software Oracle XE.

4.- En la siguiente pantalla (figura 4), se solicita la escritura de una contraseña para el usuario *system* (administrador del sistema). Escribala, considerando que debe recordarla para interactuar posteriormente con el sistema, y vuelva a escribirla para confirmar la correcta correspondencia. Continúe con el botón **Next**.

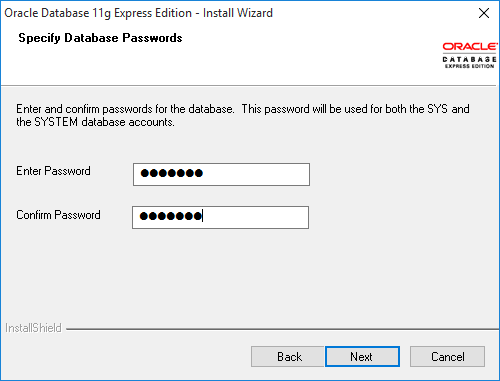


Figura 4. Escritura de la contraseña del usuario *system*.

5.- Para concluir, se muestran en pantalla los parámetros de instalación. Trate de guardarlos para referencias futuras. Continúe con el botón **Install**.

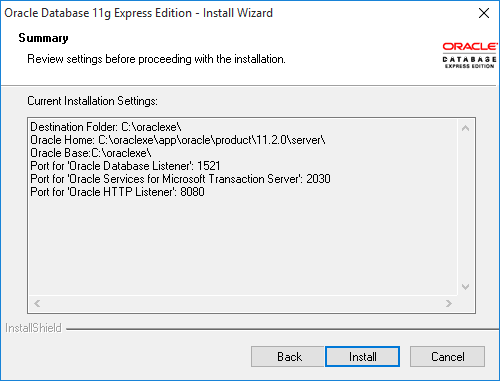


Figura 5. Finalización del instalador de Oracle XE.

6.- Una vez concluida la instalación, se crea un grupo de programas en el menú del sistema de Windows, con varios íconos descriptivos de las actividades que se pueden realizar con la base de datos. Los que se van a utilizar durante el curso son: **Start Database**, **Stop Database** y **Run SQL Command Line**. Ejecute el ícono de **Start Database,** y observe que se abre una nueva ventana de la terminal de comandos de Windows, mandando mensajes de inicialización del sistema. Al indicar que la base de datos ya está levantada en memoria, no cierre la ventana, solo minimicela.

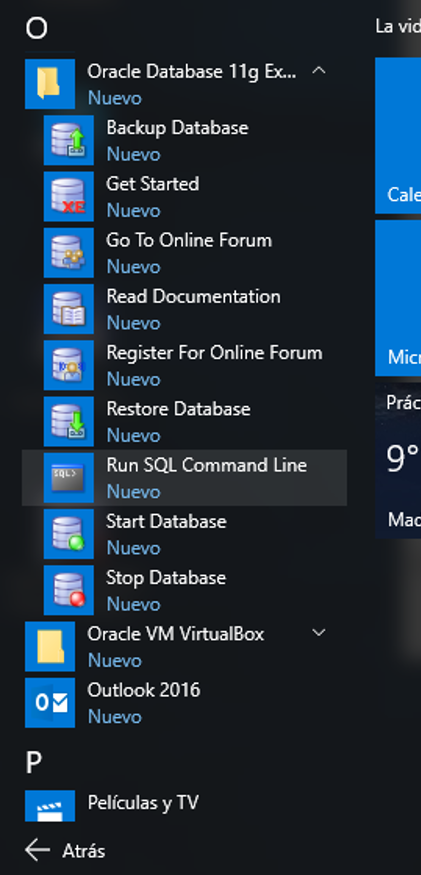


Figura 6. Íconos del grupo de programas Oracle Database 11g Express Edition

7.- También es posible inicializar el sistema a través de la ventana de *Servicios de Windows*. Mediante la ejecución del *administrador de tareas* de Windows, en la pestaña de *Servicios*, busque los procesos nombrados **OracleServiceXE** y **OracleTNSListener**, y si es que están detenidos, pongálos en ejecución. Vea la figura 7 como referencia.

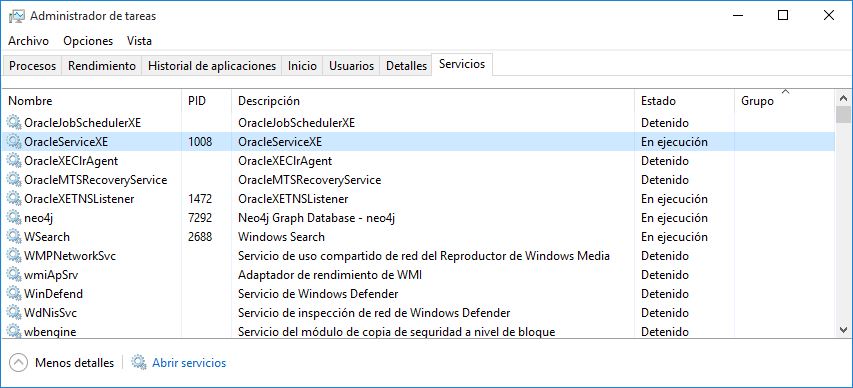


Figura 7. Inicialización de los servicios de Oracle XE mediante el administrador de tareas.

8.- Ahora abra una ventana de línea de comandos SQL (como se muestra en la figura 6), y escriba las siguientes sentencias (ignore el prompt *SQL>* de la terminal):

SQL> CONNECT / AS SYSDBA;

SQL> CREATE USER BD\_REL IDENTIFIED BY BD\_REL;

SQL> GRANT RESOURCE, CONNECT TO BD\_REL;

SQL> DISCONNECT;

SQL> CONNECT BD\_REL/BD\_REL;

SQL> ALTER SESSION SET NLS\_TERRITORY = 'Mexico';

SQL> CREATE TABLE EMPLEADO (

NSS CHAR(10) NOT NULL ,

NOMBRE VARCHAR2(30) NOT NULL ,

APELLIDO VARCHAR2(45),

FECHA\_NAC DATE NOT NULL ,

DIRECCION VARCHAR2(120),

GENERO CHAR(1) NOT NULL ,

SALARIO NUMBER(7,2),

NSS\_JEFE CHAR(10),

NODEP NUMBER(\*,0),

TITULO VARCHAR2(20) NOT NULL ,

FECHA\_INGRESO DATE,

CONSTRAINT EMP\_GENERO\_CHK CHECK (GENERO ='M' OR GENERO ='F') ,

CONSTRAINT EMP\_SAL\_CHK CHECK (SALARIO BETWEEN 2500.0 AND 50000.0) ,

CONSTRAINT EMP\_NSS\_PK PRIMARY KEY (NSS)

);

SQL> CREATE TABLE DEPARTAMENTO (

NODEP NUMBER(3,0),

NOMBRE VARCHAR2(30) NOT NULL ,

NSS\_EMP CHAR(10) NOT NULL ,

CONSTRAINT DEP\_NODEP\_PK PRIMARY KEY (NODEP),

CONSTRAINT DEP\_NOMBRE\_PK UNIQUE (NOMBRE)

);

SQL> CREATE TABLE LOCALIZACION\_DPTO (

NODEP NUMBER NOT NULL,

LOC\_DEP VARCHAR2(40),

CONSTRAINT LOC\_NODEP\_LOC\_PK PRIMARY KEY (NODEP, LOC\_DEP)

);

SQL> CREATE TABLE DEPENDIENTE (

NOMBRE VARCHAR2(40),

APELLIDO VARCHAR2(45),

FECHA\_NAC DATE NOT NULL ,

PARENTESCO VARCHAR2(20) NOT NULL ,

NSS\_EMP CHAR(10) NOT NULL ,

GENERO CHAR(1) NOT NULL ,

CONSTRAINT DEP\_NOMBRE\_NSS\_PK PRIMARY KEY (NOMBRE,NSS\_EMP),

CONSTRAINT DEP\_GENERO\_CHK CHECK (UPPER(GENERO) = 'M' OR UPPER(GENERO) = 'F') ,

CHECK (UPPER(PARENTESCO) IN ('HIJO', 'HIJA', 'PADRE', 'MADRE', 'NIETA', 'NIETO'))

);

SQL> CREATE TABLE PROYECTO (

NOP NUMBER NOT NULL ,

NOMBRE VARCHAR2(80),

FECHA\_INICIO DATE,

FECHA\_FIN DATE,

NODEP NUMBER,

CONSTRAINT PROY\_PK PRIMARY KEY (NOP),

CONSTRAINT PROY\_UK UNIQUE (NOMBRE)

);

SQL> CREATE TABLE TRABAJA (

NSS CHAR(10) NOT NULL ,

NOP NUMBER NOT NULL ,

HORAS NUMBER NOT NULL ,

CONSTRAINT TRAB\_PK PRIMARY KEY (NSS, NOP)

);

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (777888999,'Jesús','López', TO\_DATE('04/04/1973','dd/mm/yyyy'),'Calle Venus 45, col. Irrigación, Ecatepec, 89765','M',50000,NULL,3,'Ingeniero', TO\_DATE('05/04/2000','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (222333444,'Guadalupe','Oñate', TO\_DATE('24/11/1969','dd/mm/yyyy'),'Av.Revolución 348, col. Fuentes, San Ignacio, 67656','F',25000,777888999,3,'Licenciado', TO\_DATE('07/04/2000','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (444555666,'Julia','Regalado', TO\_DATE('30/07/1975','dd/mm/yyyy'),'Calle cielo 76, col. Independencia, Tepeji, 34256','F',28000,111222333,1,'Ingeniero', TO\_DATE('28/06/2001','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (555666777,'Mario','Medina', TO\_DATE('01/10/1977','dd/mm/yyyy'),'Av.Politécnico 650, col. Profesiones, CDMX, 09765','M',20000,777888999,3,'Licenciado', TO\_DATE('30/05/2000','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (333444555,'Rogelio','Calzada', TO\_DATE('25/03/1965','dd/mm/yyyy'),'Av.Independencia 123, col. Fuentes, San Ignacio, 56565','M',39000,NULL,2,'Licenciado', TO\_DATE('05/04/2000','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (666777888,'Bruce','Bolaños', TO\_DATE('07/08/1963','dd/mm/yyyy'),'Mar Mediterráneo 56, col. Lomas Lindas, CDMX, 09879','M',24000,111222333,1,'Licenciado', TO\_DATE('17/12/2000','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (999000111,'Laura','Méndez', TO\_DATE('20/02/1982','dd/mm/yyyy'),'Av.Montevideo 98, col. Ampliación Hidalgo, Texcoco, 45679','F',18000,333444555,2,'Ingeniero', TO\_DATE('20/12/2000','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (111222333,'Sandra','Guzmán', TO\_DATE('27/11/1970','dd/mm/yyyy'),'Av.Siempreviva 444, col. San Mateo, Ecatepec, 89740','F',45000,NULL,1,'Ingeniero', TO\_DATE('20/12/2000','dd/mm/yyyy'));

SQL> INSERT INTO EMPLEADO (NSS,NOMBRE,APELLIDO,FECHA\_NAC,DIRECCION,GENERO,SALARIO,NSS\_JEFE,NODEP,TITULO,FECHA\_INGRESO) VALUES (888999000,'Guadalupe','Hidalgo', TO\_DATE('19/03/1985','dd/mm/yyyy'),'Av.Miguel Hidalgo 56, col. Independencia, Tepeji, 87981','F',27000,333444555,2,'Ingeniero', TO\_DATE('11/10/2001','dd/mm/yyyy'));

SQL> ALTER TABLE EMPLEADO ADD CONSTRAINT EMP\_EMP\_FK FOREIGN KEY (NSS\_JEFE)

REFERENCES EMPLEADO (NSS);

SQL> INSERT INTO departamento (NODEP,NOMBRE,NSS\_EMP) VALUES (1,'Sistemas', 111222333);

SQL> INSERT INTO departamento (NODEP,NOMBRE,NSS\_EMP) VALUES (2,'Ventas', 333444555);

SQL> INSERT INTO departamento (NODEP,NOMBRE,NSS\_EMP) VALUES (3,'Administración', 777888999);

SQL> ALTER TABLE EMPLEADO ADD CONSTRAINT EMP\_DEPT\_FK FOREIGN KEY (NODEP)

REFERENCES DEPARTAMENTO (NODEP);

SQL> ALTER TABLE DEPARTAMENTO ADD CONSTRAINT DPTO\_EMP\_FK FOREIGN KEY (NSS\_EMP) REFERENCES EMPLEADO (NSS);

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Gustavo','Guzmán',TO\_DATE('07/12/1999','dd/mm/yyyy'),'hijo',111222333,'M');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Karen','Oñate',TO\_DATE('09/11/1995','dd/mm/yyyy'),'hija',222333444,'F');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Julieta','Castillo',TO\_DATE('09/11/1951','dd/mm/yyyy'),'madre',333444555,'F');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Lucia','Calzada',TO\_DATE('09/11/1995','dd/mm/yyyy'),'hija',333444555,'F');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Ernesto','Medina',TO\_DATE('28/10/1945','dd/mm/yyyy'),'padre',555666777,'M');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Carlos','López',TO\_DATE('12/04/2005','dd/mm/yyyy'),'hijo',777888999,'M');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Karla','Hidalgo',TO\_DATE('26/08/2009','dd/mm/yyyy'),'hija',888999000,'F');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Sergio','Méndez',TO\_DATE('17/12/1992','dd/mm/yyyy'),'hijo',999000111,'M');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Felipe','Bolaños',TO\_DATE('26/08/2009','dd/mm/yyyy'),'nieto',666777888,'M');

SQL> INSERT INTO dependiente(NOMBRE,APELLIDO,FECHA\_NAC,PARENTESCO,NSS\_EMP,GENERO)

VALUES ('Rodrigo','Calzada',TO\_DATE('21/07/1997','dd/mm/yyyy'),'hijo',333444555,'M');

SQL> ALTER TABLE DEPENDIENTE ADD CONSTRAINT DEP\_EMP\_FK FOREIGN KEY (NSS\_EMP)

REFERENCES EMPLEADO (NSS);

SQL> INSERT INTO proyecto(NOMBRE,NOP,NODEP,FECHA\_INICIO,FECHA\_FIN)

VALUES ('Sitio Web',2,1,TO\_DATE('18/02/2010','dd/mm/YYYY'), TO\_DATE('12/05/2011','dd/mm/YYYY'));

SQL> INSERT INTO proyecto(NOMBRE,NOP,NODEP,FECHA\_INICIO,FECHA\_FIN)

VALUES ('Mejoramiento de Calidad',1,2,TO\_DATE('21/06/2011','dd/mm/YYYY'), TO\_DATE('10/01/2012','dd/mm/YYYY'));

SQL> INSERT INTO proyecto(NOMBRE,NOP,NODEP,FECHA\_INICIO,FECHA\_FIN)

VALUES ('Publicidad',3,3,TO\_DATE('06/09/2011','dd/mm/YYYY'), TO\_DATE('29/08/2012','dd/mm/YYYY'));

SQL> ALTER TABLE PROYECTO ADD CONSTRAINT PROY\_DEPT\_FK FOREIGN KEY (NODEP)

REFERENCES DEPARTAMENTO (NODEP);

SQL> INSERT INTO trabaja(NSS,NOP,HORAS) VALUES (222333444,3,8);

SQL> INSERT INTO trabaja(NSS,NOP,HORAS) VALUES (555666777,2,16);

SQL> INSERT INTO trabaja(NSS,NOP,HORAS) VALUES (888999000,2,8);

SQL> INSERT INTO trabaja(NSS,NOP,HORAS) VALUES (555666777,1,10);

SQL> INSERT INTO trabaja(NSS,NOP,HORAS) VALUES (666777888,1,2);

SQL> INSERT INTO trabaja(NSS,NOP,HORAS) VALUES (888999000,1,8);

SQL> ALTER TABLE TRABAJA ADD CONSTRAINT TRAB\_PROY\_FK FOREIGN KEY (NOP)

REFERENCES PROYECTO (NOP);

SQL> ALTER TABLE TRABAJA ADD CONSTRAINT TRAB\_EMP\_FK FOREIGN KEY (NSS)

REFERENCES EMPLEADO (NSS);

SQL> INSERT INTO localizacion\_dpto(LOC\_DEP,NODEP) VALUES ('Monterrey',3);

SQL> INSERT INTO localizacion\_dpto(LOC\_DEP,NODEP) VALUES ('Guadalupe',3);

SQL> INSERT INTO localizacion\_dpto(LOC\_DEP,NODEP) VALUES ('Juárez',3);

SQL> INSERT INTO localizacion\_dpto(LOC\_DEP,NODEP) VALUES ('DF',1);

SQL> INSERT INTO localizacion\_dpto(LOC\_DEP,NODEP) VALUES ('DF',2);

SQL> INSERT INTO localizacion\_dpto(LOC\_DEP,NODEP) VALUES ('Pachuca',2);

SQL> ALTER TABLE LOCALIZACION\_DPTO ADD CONSTRAINT LOC\_DPTO\_FK FOREIGN KEY (NODEP) REFERENCES DEPARTAMENTO (NODEP);

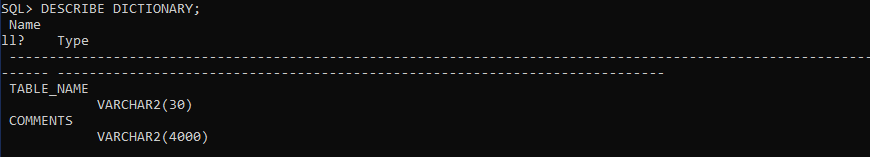
Verifique que la ejecución de las sentencias anteriores no tuvieron errores.

2.- Ejecute las siguientes sentencias. Para las que estén resaltadas (color amarillo), realice una captura de pantalla e insertela como imagen a continuación de la sentencia, cuidando que la captura sea visible y legible, y que los resultados sean consistentes con las tablas y restricciones creadas:

SQL> SET PAGESIZE 200;

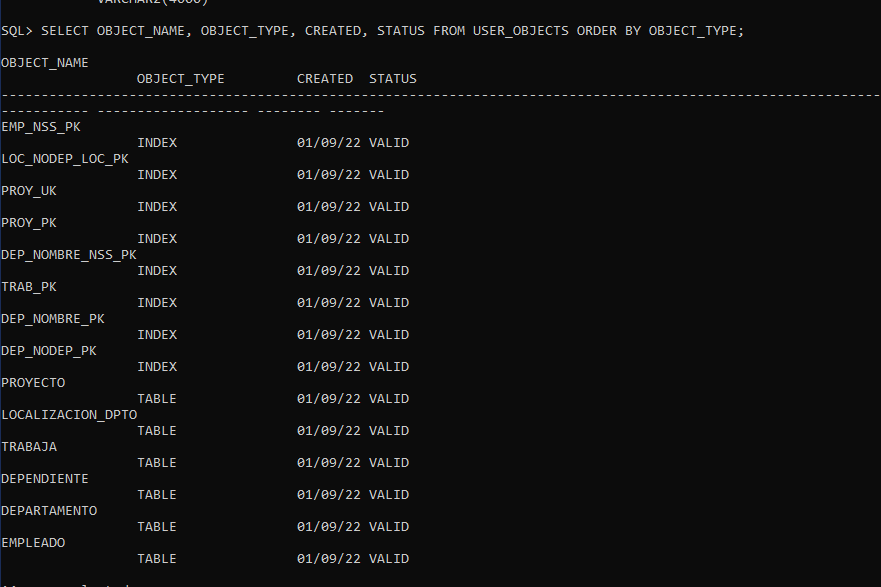
SQL> SET LINESIZE 200;

SQL> DESCRIBE DICTIONARY;

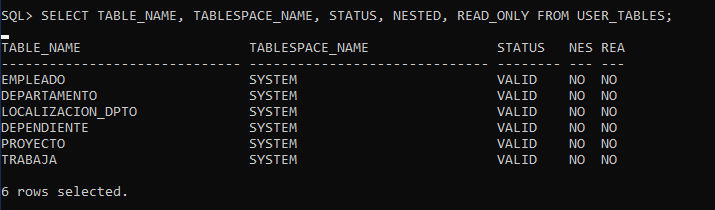


SQL> SELECT TABLE\_NAME FROM DICTIONARY;

SQL> SELECT OBJECT\_NAME, OBJECT\_TYPE, CREATED, STATUS FROM USER\_OBJECTS ORDER BY OBJECT\_TYPE;

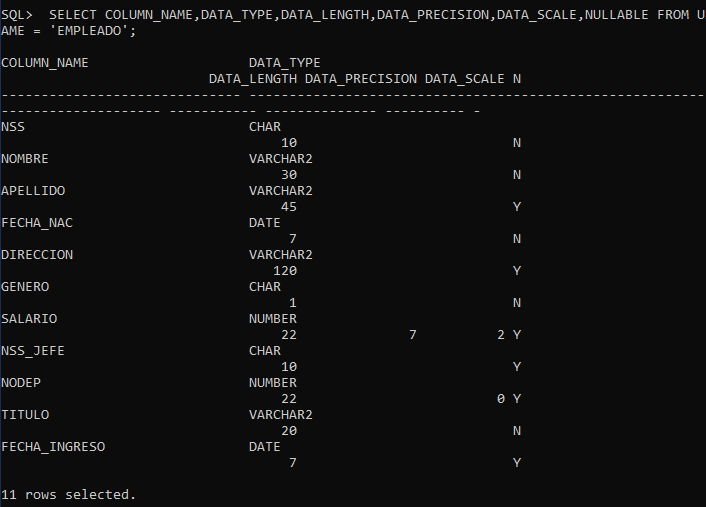


SQL> SELECT TABLE\_NAME, TABLESPACE\_NAME, STATUS, NESTED, READ\_ONLY FROM USER\_TABLES;



SQL> COLUMN DATA\_TYPE FORMAT A20;

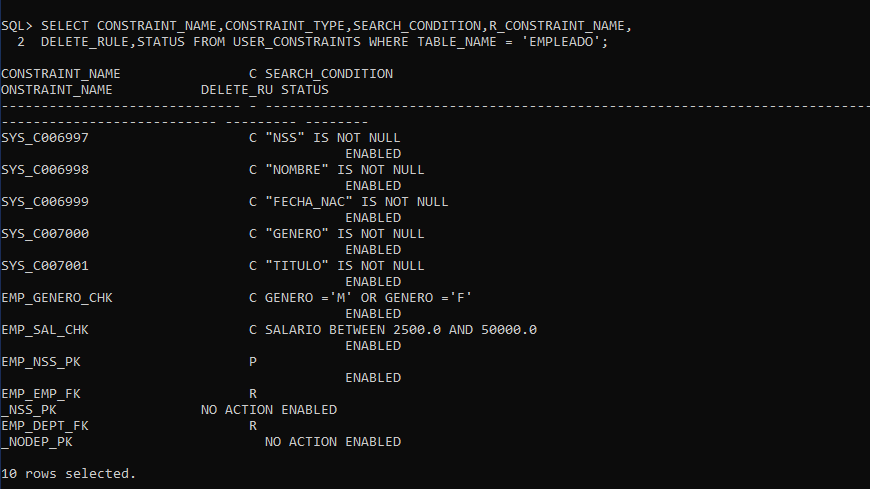
SQL> SELECT COLUMN\_NAME,DATA\_TYPE,DATA\_LENGTH,DATA\_PRECISION,DATA\_SCALE,NULLABLE FROM USER\_TAB\_COLUMNS WHERE TABLE\_NAME = 'EMPLEADO';



SQL> COLUMN SEARCH\_CONDITION FORMAT A35;

SQL> SELECT CONSTRAINT\_NAME,CONSTRAINT\_TYPE,SEARCH\_CONDITION,R\_CONSTRAINT\_NAME,

DELETE\_RULE,STATUS FROM USER\_CONSTRAINTS WHERE TABLE\_NAME = 'EMPLEADO';



3.- Ejecute las siguientes sentencias. Para las que estén resaltadas (color amarillo), realice una captura de pantalla e insertela como imagen a continuación de la sentencia, cuidando que la captura sea visible y legible:

SQL> SET DEFINE OFF;

SQL> CREATE OR REPLACE FUNCTION ultimo\_dep RETURN NUMBER IS

no\_dep departamento.nodep%TYPE := 0;

BEGIN

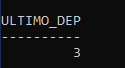
SELECT max(nodep) INTO no\_dep FROM departamento;

RETURN no\_dep;

END;

/

SQL> SELECT ultimo\_dep FROM dual;



SQL> CREATE OR REPLACE PROCEDURE inserta\_departamento(nombre VARCHAR, jefe CHAR) IS

num\_dep departamento.nodep%TYPE := ultimo\_dep;

BEGIN

IF nombre IS NULL OR jefe IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Los datos del departamento y del jefe son obligatorios.');

ELSIF num\_dep IS NOT NULL THEN

INSERT INTO departamento (NODEP,NOMBRE,NSS\_EMP) VALUES (num\_dep + 1,nombre,jefe);

IF SQL%ROWCOUNT > 0 THEN

COMMIT;

ELSE

RAISE\_APPLICATION\_ERROR(-20001,'No fue posible insertar el registro');

END IF;

END IF;

END;

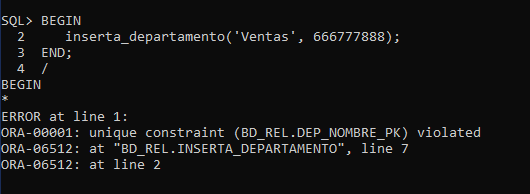
/

SQL> BEGIN

inserta\_departamento('Ventas', 666777888);

END;

/



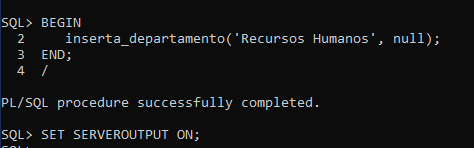
SQL> BEGIN

inserta\_departamento('Recursos Humanos', null);

END;

/

SET SERVEROUTPUT ON;

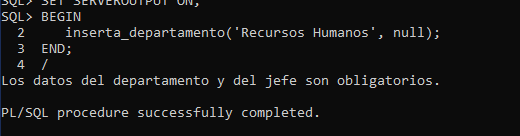


SQL> BEGIN

inserta\_departamento('Recursos Humanos', null);

END;

/



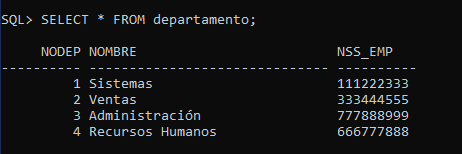
SQL> BEGIN

inserta\_departamento('Recursos Humanos', 666777888);

END;

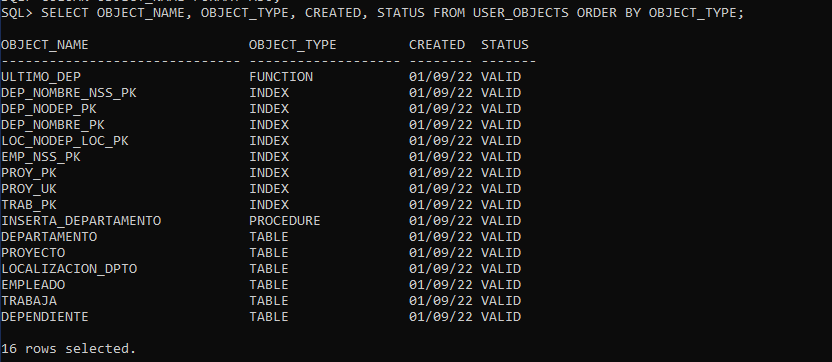
/

SQL> SELECT \* FROM departamento;

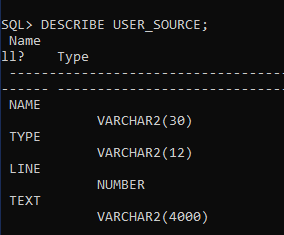


SQL> COLUMN OBJECT\_NAME FORMAT A30;

SQL> SELECT OBJECT\_NAME, OBJECT\_TYPE, CREATED, STATUS FROM USER\_OBJECTS ORDER BY OBJECT\_TYPE;

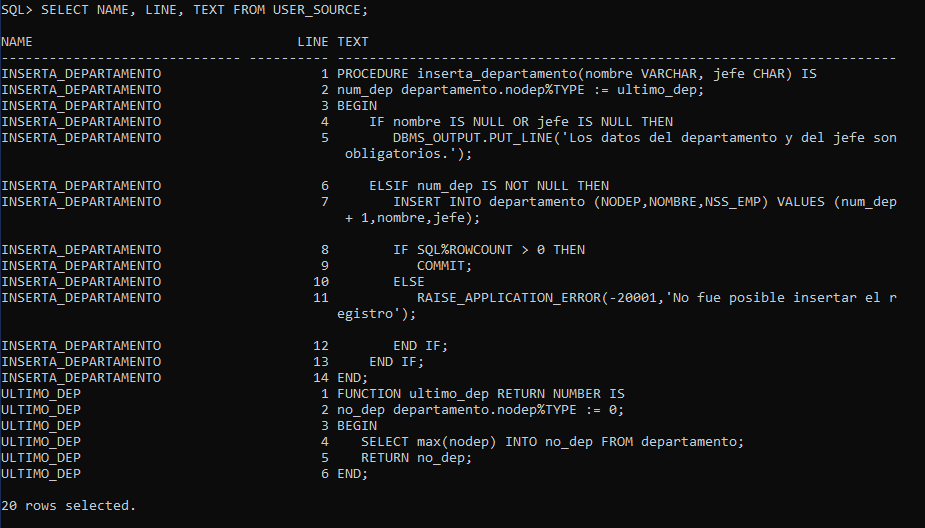


SQL> DESCRIBE USER\_SOURCE;



SQL> COLUMN TEXT FORMAT A70;

SQL> SELECT NAME, LINE, TEXT FROM USER\_SOURCE;



SQL> CREATE OR REPLACE FUNCTION antiguedad(fecha\_cont DATE) RETURN NUMBER IS

res NUMBER := 0;

BEGIN

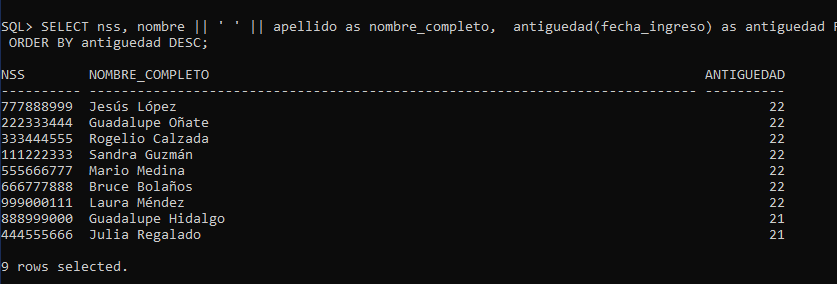
res := EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM fecha\_cont);

RETURN res;

END;

/

SQL> SELECT nss, nombre || ' ' || apellido as nombre\_completo, antiguedad(fecha\_ingreso) as antiguedad FROM empleado ORDER BY antiguedad DESC;



SQL> CREATE OR REPLACE FUNCTION genera\_fecha(anio INTEGER, mes INTEGER, dia INTEGER, sep VARCHAR) RETURN DATE IS

patron CHAR(12) := '''' || 'YYYY' || sep || 'mm' || sep || 'dd' || '''';

fecha CHAR(12):= '' || anio || sep || mes || sep || dia || '';

day DATE := null;

BEGIN

day := TO\_DATE(fecha, patron);

RETURN day;

END;

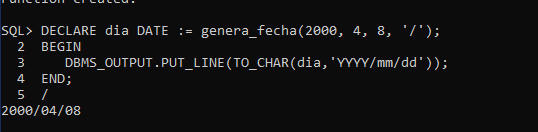
/

SQL> DECLARE dia DATE := genera\_fecha(2000, 4, 8, '/');

BEGIN

DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(dia,'YYYY/mm/dd'));

END;



SQL> CREATE OR REPLACE FUNCTION edad(fecha\_nac DATE) RETURN NUMBER IS

res NUMBER := 0;

BEGIN

res := MONTHS\_BETWEEN(SYSDATE, fecha\_nac);

RETURN ROUND(res/12,0);

END;

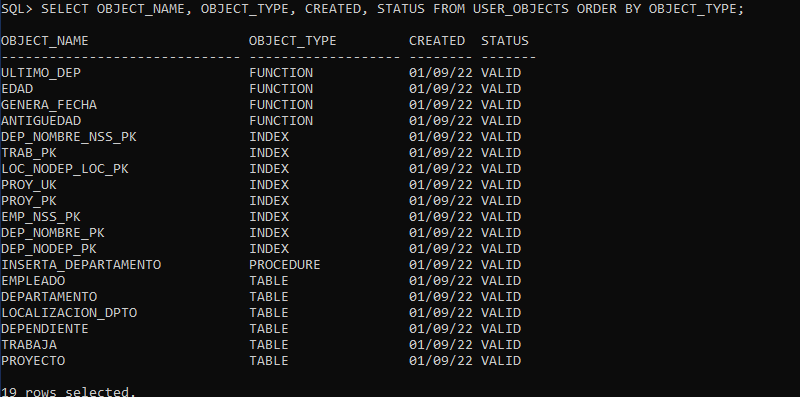
/

SQL> COLUMN NOMBRE\_COMPLETO FORMAT A30;

SQL> SELECT nombre || ' ' || apellido as nombre\_completo, edad(fecha\_nac) as edad, parentesco FROM dependiente ORDER BY edad;

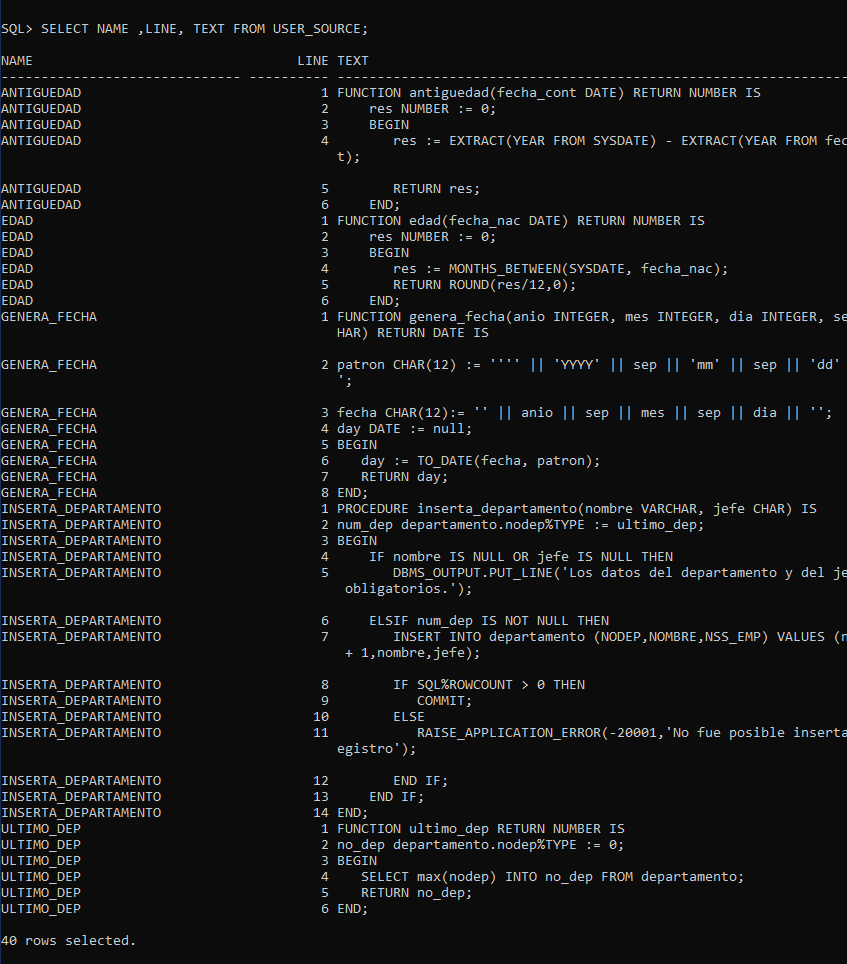
SQL> COLUMN OBJECT\_NAME FORMAT A30;

SQL> SELECT OBJECT\_NAME, OBJECT\_TYPE, CREATED, STATUS FROM USER\_OBJECTS ORDER BY OBJECT\_TYPE;



SQL> COLUMN TEXT FORMAT A70;

SQL> SELECT NAME, LINE, TEXT FROM USER SOURCE;

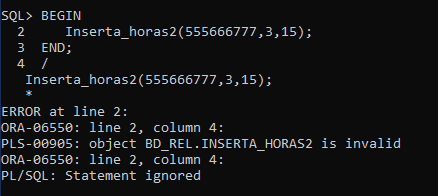


4.-Escriba el código PL/SQL de definición del procedimiento **inserta\_horas(noemp CHAR, noproy NUMBER, horas INTEGER)** que permita insertar un registro de un empleado que labore en un proyecto (en la tabla *trabaja*) y que valide que el total de las horas trabajadas por ese empleado en cada uno de los proyectos que trabaja, no supere las 40 horas laborales por semana. Incluya mensajes informativos al usuario en caso de errores.

5.- Compruebe el funcionamiento correcto del procedimiento anterior ejecutando las siguientes inserciones de datos:

NSS:555666777, NoProyecto: 3, horas:15;

NSS:555666777, NoProyecto: 3, horas:14;



6.- Termine la sesión en la terminal, y apague el proceso del servidor.

CUESTIONARIO

1.- ¿Qué ventajas tiene el uso de procedimientos o funciones en la validación de datos? Mencione al menos dos.

1. Ventaja 1 Los procedimientos nos permiten mantener almacenada una declaracion para formas repetitvas de ejecucion.
2. Ventaja 2 los procedimientos nos ayudan a poder realizar con myor rapidez cambios en las tablas sin tener que escribir la sentencia nuevamente.

2.- ¿Qué diferencias significativas se tienen en la definición de una función con respecto a un procedimiento?

CONCLUSIONES

Escriba sus conclusiones (no personales) con respecto al desarrollo de esta práctica.

El desarrollo de esta practica , nos ayudo a poder ejecutar la instalacion del programa para poder realizar las consultas necesarias y de igual forma para poder crear las tablas.

El desarrolo de procedimientos y funciones para su entendimiento , fue cable vital para poder desarrollar esta practica por que pone en ejecucion los conceptos