|  |  |  |
| --- | --- | --- |
| http://4.bp.blogspot.com/-uP_hXyaqILI/ToD3k8H-2kI/AAAAAAAAAzM/1eLxyzPmXJM/s1600/ipn%2Blogo2.jpg | INSTITUTO POLITÉCNICO NACIONALESCUELA SUPERIOR DE CÓMPUTO **ACADEMIA DE CIENCIA DE DATOS** | http://www.sepi.escom.ipn.mx/WPS/WCM/CONNECT/7A176E804EAA2A3E83949B9BEB1E1CB/ESCOM2BB5.JPG?MOD=AJPERES&CACHEID=7a176e804eaa2a3e83949b9beb1e1cba |

Bases de Datos Avanzadas

Práctica no. 3:

Manejo de un modelo orientado a objetos en una base de datos relacional (tipos de datos largos, inserciones y consultas de datos)

**Número de Práctica: 3**

**Nombre de la Práctica: Manejo de un modelo orientado a objetos en una base de datos relacional (tipos de datos largos, inserciones y consultas de datos)**

REQUISITOS PREVIOS

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

* [Oracle Server 11g Express Edition (XE)](https://drive.google.com/file/d/1P9EVa3eU5Fp3v80ojSSzDFpSzoH6nMaH/view?usp=sharing)
* [Oracle SQL Developer](https://www.oracle.com/database/sqldeveloper/technologies/download/)

**Objetos Grandes (Large Objects) LOBs**

Oracle soporta tres tipos de datos para el almacenamiento de grandes volúmenes de datos:

* BLOB representa un objeto grande binario
* CLOB representa un objeto grande de caracteres
* BFILE representa un archivo almacenado fuera de la base de datos.

Hay dos categorías de LOBs:

* LOBs internos (CLOB, BLOB) lo cuales están almacenados dentro del servidor de bases de datos.
* LOBs externos (BFILE) que son almacenados fuera de la base de datos y son manejados por el Sistema Operativo anfitrión.

Los tipos LOBs soportan hasta 4 GB de almacenamiento y pueden existir dentro de la definición de una tabla varias columnas de este tipo. Los tipos BFILE son accedidos con permisos de solo lectura.

Cuando el tamaño de un dato LOB es menor a 4K, se almacena en el mismo espacio de la tabla, pero para tamaños mayores, se crea un *LOB Locator*, (apuntador) que mantiene la dirección de almacenamiento en donde se guardan los valores (*LOB Value*), en el segmento físico designado para datos grandes.

El tipo BFILE permite al servidor de Oracle acceder al sistema de archivo del SO y poder recuperar archivos almacenados en un directorio. Para ello se proporciona la sentencia CREATE DIRECTORY para crear el enlace. De esta manera, el usuario puede agregar archivos a la carpeta del sistema de archivos, y el servidor de Oracle puede tener acceso mediante el objeto DIRECTORY.

El paquete DBMS\_LOB proporciona la funcionalidad para manipular el contenido de los archivos:

COPY Copia todo o parte de la LOB de origen en la LOB de destino

ERASE Borra todo o parte de una LOB

LOADFROMFILE Carga datos BFILE en una LOB interna

TRIM Recorta el valor de LOB a una longitud más corta especificada

WRITE Escribe datos en la LOB desde un offset especificado

GETLENGTH Obtiene la longitud del valor de LOB

INSTR Devuelve la posición coincidente de la enésima aparición del patrón en la LOB

READ Lee datos de la LOB a partir del offset especificado

SUBSTR Devuelve parte del valor de LOB a partir del offset especificado

FILECLOSE Cierra el archivo

FILECLOSEALL Cierra todos los archivos abiertos anteriormente

FILEEXISTS Comprueba si el archivo existe en el servidor

FILEGETNAME Obtiene el alias del directorio y el nombre de archivo

FLEISOPEN Comprueba si el archivo se abrió utilizando los localizadores BFILE de entrada

FILEOPEN Abre un archivo

Ejemplo:

CREATE **DIRECTORY directorio AS** '**/ruta\_archivo**'; -- ejecutar como SYS

CREATE TABLE employee (emp\_id NUMBER, emp\_name VARCHAR2(35), resume **CLOB**, picture **BLOB**);

Para insertar datos es recomendable que se inicialice el espacio de almacenamiento para el LOB con la función EMPTY\_CLOB() o EMPTY\_BLOB():

INSERT INTO employee VALUES (7898, 'James Dean', 'date of birth = 8 february 1931', **EMPTY\_BLOB()**);

Para manejar datos de tipo BFILE:

INSERT INTO employee VALUES (7899, 'Elvis Presley',**EMPTY\_CLOB()**, **BFILE(**'**directorio**'**,** '**pic\_1.jpg**'**)**);

Está práctica es la continuación de la Práctica 2, por lo que ya debe estar disponible la base de datos con la creación de los objetos necesarios para su realización.

DESARROLLO DE LA PRÁCTICA:

a) Para iniciar la práctica debe crear una carpeta en el sistema operativo Windows en la ruta **c:\repositorio** y copiar el archivo *imágenes.zip* allí mismo. Descomprima el contenido del archivo anterior en esa misma carpeta. Abra una ventana de la Línea de Comandos SQL de Oracle, y escriba las siguientes sentencias:

1. CONNECT / AS SYSDBA;
2. GRANT CREATE ANY DIRECTORY TO OBJ;
3. CREATE OR REPLACE DIRECTORY repositorio AS 'C:\repositorio'
4. GRANT READ, WRITE ON DIRECTORY repositorio TO OBJ;
5. ALTER USER OBJ QUOTA UNLIMITED ON USERS;
6. DISCONNECT;
7. ​CONNECT obj/obj;
8. SET PAGESIZE 200;
9. SET LINESIZE 200;
10. ALTER TYPE tEmpleado ADD ATTRIBUTE foto BFILE CASCADE;
11. CREATE TABLE empleado OF tEmpleado(

nombre NOT NULL,

apellido NOT NULL,

fecha\_nac NOT NULL,

genero NOT NULL,

NSS NOT NULL,

salario DEFAULT 3000,

fecha\_ingreso NOT NULL,

CHECK(genero IN ('M','F')),

CHECK(salario BETWEEN 3000 AND 60000));

1. ALTER TABLE empleado ADD CONSTRAINT Pk\_emp PRIMARY KEY(NSS);
2. CREATE TABLE dependiente OF tDependiente(

nombre NOT NULL,

apellido NOT NULL,

fecha\_nac NOT NULL,

genero NOT NULL,

CHECK(genero IN ('M','F')));

1. ALTER TABLE dependiente ADD CONSTRAINT Pk\_nom\_apell PRIMARY KEY(nombre,apellido);
2. CREATE TABLE departamento OF tDepartamento(

nodep NOT NULL,

nombre NOT NULL UNIQUE);

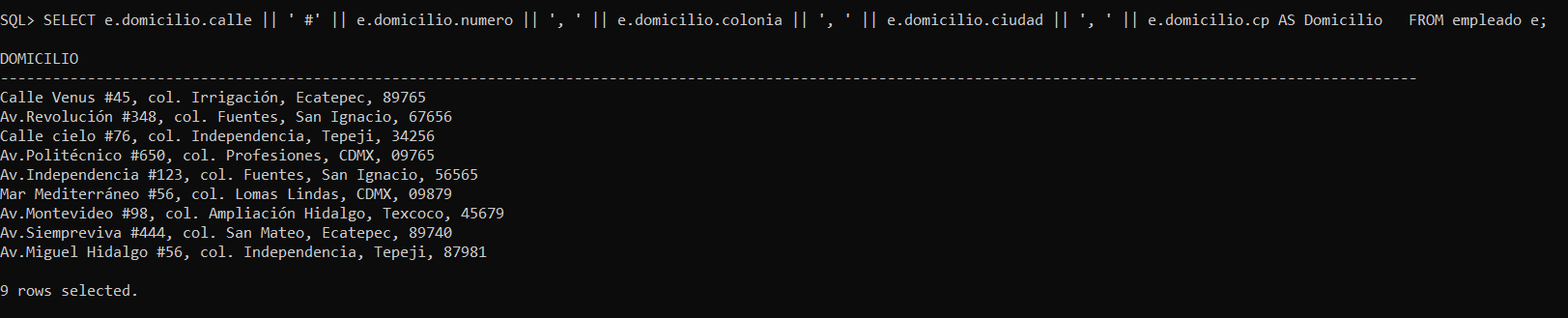
1. ALTER TABLE departamento ADD CONSTRAINT Pk\_nodep PRIMARY KEY(nodep);
2. CREATE TABLE proyecto OF tProyecto(

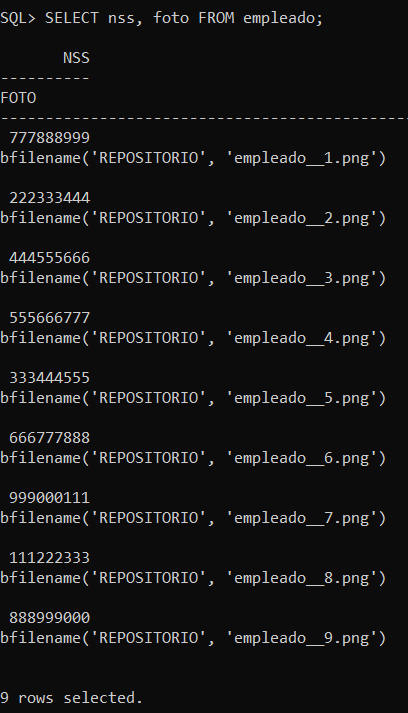
noproy NOT NULL,

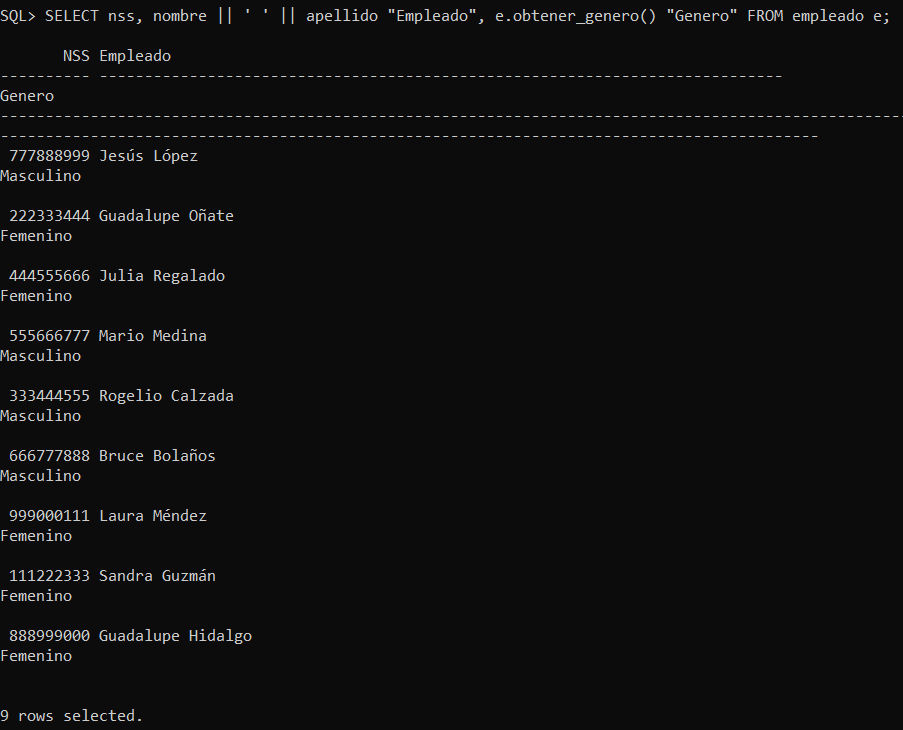
nombre NOT NULL UNIQUE,

fecha\_inicio NOT NULL) NESTED TABLE empleados STORE AS emps;

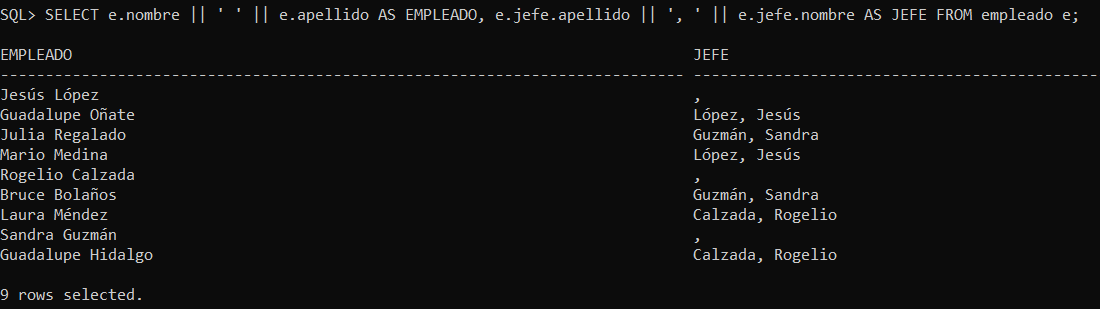
1. ALTER TABLE proyecto ADD CONSTRAINT Pk\_proy PRIMARY KEY(noproy);
2. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (777888999,'Jesús','López',TO\_DATE('04/04/1973','dd/mm/YYYY'),'M',50000,TO\_DATE('05/04/2000','dd/mm/YYYY'),'Ingeniero');
3. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (222333444,'Guadalupe','Oñate',TO\_DATE('24/11/1969','dd/mm/YYYY'),'F',25000,TO\_DATE('07/04/2000','dd/mm/YYYY'),'Licenciado');
4. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (444555666,'Julia','Regalado',TO\_DATE('30/07/1975','dd/mm/YYYY'),'F',28000,TO\_DATE('28/06/2001','dd/mm/YYYY'),'Ingeniero');
5. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (555666777,'Mario','Medina',TO\_DATE('01/10/1977','dd/mm/YYYY'),'M',20000,TO\_DATE('30/05/2000','dd/mm/YYYY'),'Licenciado');
6. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (333444555,'Rogelio','Calzada',TO\_DATE('25/03/1965','dd/mm/YYYY'),'M',39000,TO\_DATE('05/04/2000','dd/mm/YYYY'),'Licenciado');
7. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (666777888,'Bruce','Bolaños',TO\_DATE('07/08/1963','dd/mm/YYYY'),'M',24000,TO\_DATE('17/12/2000','dd/mm/YYYY'),'Licenciado');
8. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (999000111,'Laura','Méndez',TO\_DATE('20/02/1982','dd/mm/YYYY'),'F',24000,TO\_DATE('17/12/2000','dd/mm/YYYY'),'Licenciado');
9. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (111222333,'Sandra','Guzmán',TO\_DATE('27/11/1970','dd/mm/YYYY'),'F',45000,TO\_DATE('05/04/2000','dd/mm/YYYY'),'Ingeniero');
10. INSERT INTO empleado(NSS,nombre,apellido,fecha\_nac,genero,salario,fecha\_ingreso,titulo) VALUES (888999000,'Guadalupe','Hidalgo',TO\_DATE('19/03/1985','dd/mm/YYYY'),'F',27000,TO\_DATE('11/10/2001','dd/mm/YYYY'),'Ingeniero');
11. UPDATE empleado e SET e.domicilio = tdomicilio('Calle Venus', 45, 'col. Irrigación', 'Ecatepec', '89765') WHERE nss = '777888999';
12. UPDATE empleado e SET e.domicilio = tdomicilio('Av.Revolución', 348, 'col. Fuentes', 'San Ignacio', '67656') WHERE nss = '222333444';
13. UPDATE empleado e SET e.domicilio = tdomicilio('Calle cielo', 76, 'col. Independencia', 'Tepeji', '34256') WHERE nss = '444555666';
14. UPDATE empleado e SET e.domicilio = tdomicilio('Av.Politécnico', 650, 'col. Profesiones', 'CDMX', '09765') WHERE nss = '555666777';
15. UPDATE empleado e SET e.domicilio = tdomicilio('Av.Independencia', 123, 'col. Fuentes', 'San Ignacio', '56565') WHERE nss = '333444555';
16. UPDATE empleado e SET e.domicilio = tdomicilio('Mar Mediterráneo', 56, 'col. Lomas Lindas', 'CDMX', '09879') WHERE nss = '666777888';
17. UPDATE empleado e SET e.domicilio = tdomicilio('Av.Montevideo', 98, 'col. Ampliación Hidalgo', 'Texcoco', '45679') WHERE nss = '999000111';
18. UPDATE empleado e SET e.domicilio = tdomicilio('Av.Siempreviva', 444, 'col. San Mateo', 'Ecatepec', '89740') WHERE nss = '111222333';
19. UPDATE empleado e SET e.domicilio = tdomicilio('Av.Miguel Hidalgo', 56, 'col. Independencia', 'Tepeji', '87981') WHERE nss = '888999000';
20. COMMIT;
21. SELECT e.domicilio.calle || ' #' || e.domicilio.numero || ', ' || e.domicilio.colonia || ', ' || e.domicilio.ciudad || ', ' || e.domicilio.cp AS Domicilio FROM empleado e;

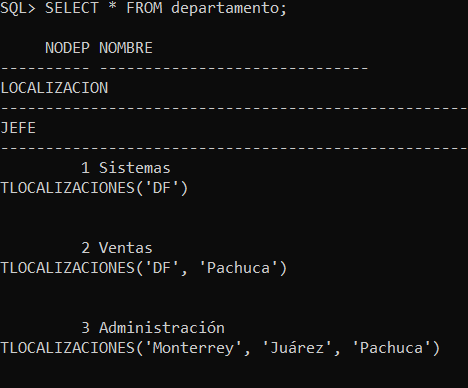
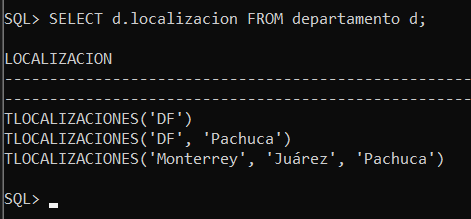
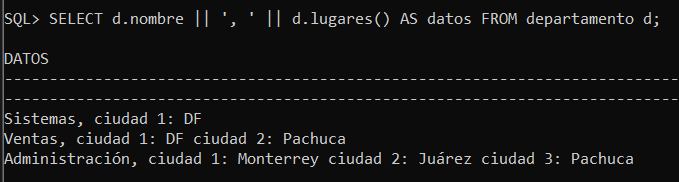
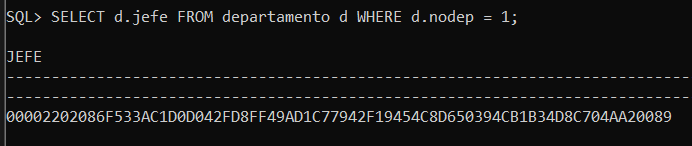


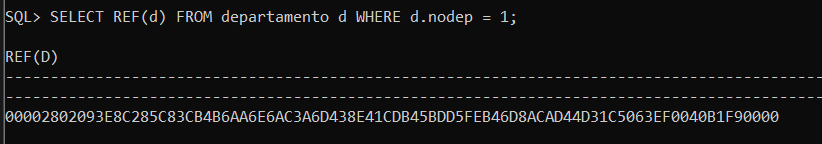
1. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_1.png') WHERE nss = '777888999';
2. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_2.png') WHERE nss = '222333444';
3. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_3.png') WHERE nss = '444555666';
4. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_4.png') WHERE nss = '555666777';
5. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_5.png') WHERE nss = '333444555';
6. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_6.png') WHERE nss = '666777888';
7. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_7.png') WHERE nss = '999000111';
8. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_8.png') WHERE nss = '111222333';
9. UPDATE empleado e SET e.foto = BFILENAME('REPOSITORIO', 'empleado\_\_9.png') WHERE nss = '888999000';
10. COMMIT;
11. SELECT nss, foto FROM empleado;
12. SELECT nss, nombre || ' ' || apellido "Empleado", e.obtener\_genero() "Genero" FROM empleado e;

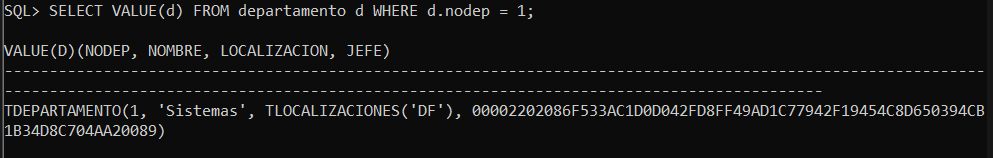


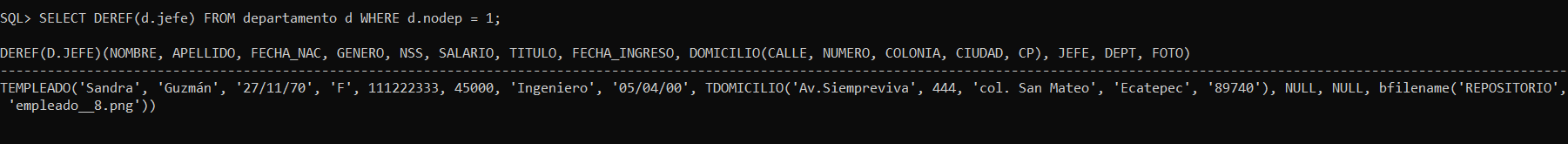
1. UPDATE empleado e SET e.jefe = NULL WHERE nss = '777888999';
2. UPDATE empleado e SET e.jefe = (SELECT REF(e) FROM empleado e WHERE e.nss = '777888999') WHERE nss = '222333444';
3. UPDATE empleado e SET e.jefe = (SELECT REF(e) FROM empleado e WHERE e.nss = '111222333') WHERE nss = '444555666';
4. UPDATE empleado e SET e.jefe = (SELECT REF(e) FROM empleado e WHERE e.nss = '777888999') WHERE nss = '555666777';
5. UPDATE empleado e SET e.jefe = NULL WHERE nss = '333444555';
6. UPDATE empleado e SET e.jefe = (SELECT REF(e) FROM empleado e WHERE e.nss = '111222333') WHERE nss = '666777888';
7. UPDATE empleado e SET e.jefe = (SELECT REF(e) FROM empleado e WHERE e.nss = '333444555') WHERE nss = '999000111';
8. UPDATE empleado e SET e.jefe = NULL WHERE nss = '111222333';
9. UPDATE empleado e SET e.jefe = (SELECT REF(e) FROM empleado e WHERE e.nss = '333444555') WHERE nss = '888999000';
10. COMMIT;
11. SELECT e.nombre || ' ' || e.apellido AS EMPLEADO, e.jefe.apellido || ', ' || e.jefe.nombre AS JEFE FROM empleado e;



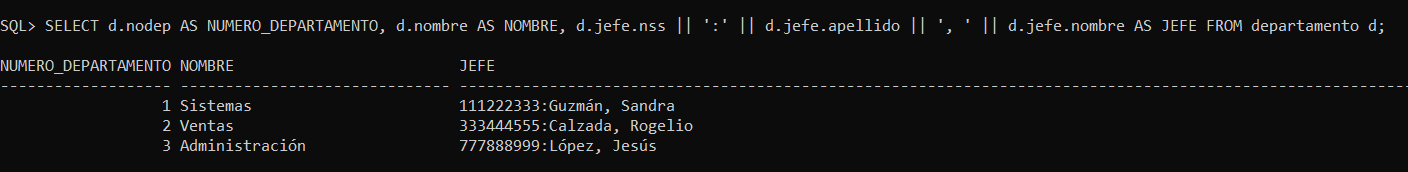
1. INSERT INTO departamento(nodep,nombre,localizacion) VALUES (1,'Sistemas', tlocalizaciones('DF'));
2. INSERT INTO departamento(nodep,nombre,localizacion) VALUES (2,'Ventas', tlocalizaciones('DF','Pachuca'));
3. INSERT INTO departamento(nodep,nombre,localizacion) VALUES (3,'Administración', tlocalizaciones('Monterrey','Juárez','Pachuca'));
4. COMMIT;
5. SELECT \* FROM departamento;
6. SELECT d.localizacion FROM departamento d;
7. SELECT d.nombre || ', ' || d.lugares() AS datos FROM departamento d;
8. UPDATE departamento d SET jefe = (SELECT REF(e) From empleado e WHERE e.nss = '777888999') WHERE d.nodep = 3;
9. UPDATE departamento d SET jefe = (SELECT REF(e) From empleado e WHERE e.nss = '333444555') WHERE d.nodep = 2;
10. UPDATE departamento d SET jefe = (SELECT REF(e) From empleado e WHERE e.nss = '111222333') WHERE d.nodep = 1;
11. COMMIT;
12. SELECT d.jefe FROM departamento d WHERE d.nodep = 1;
13. SELECT REF(d) FROM departamento d WHERE d.nodep = 1;

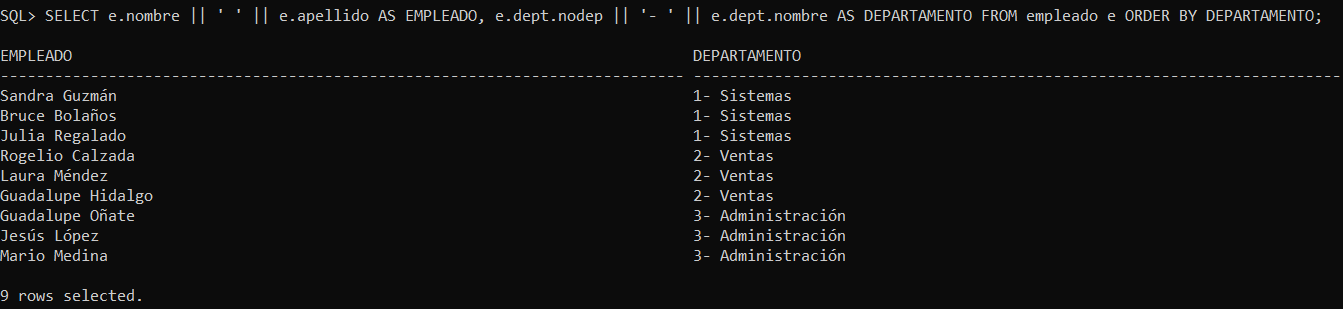


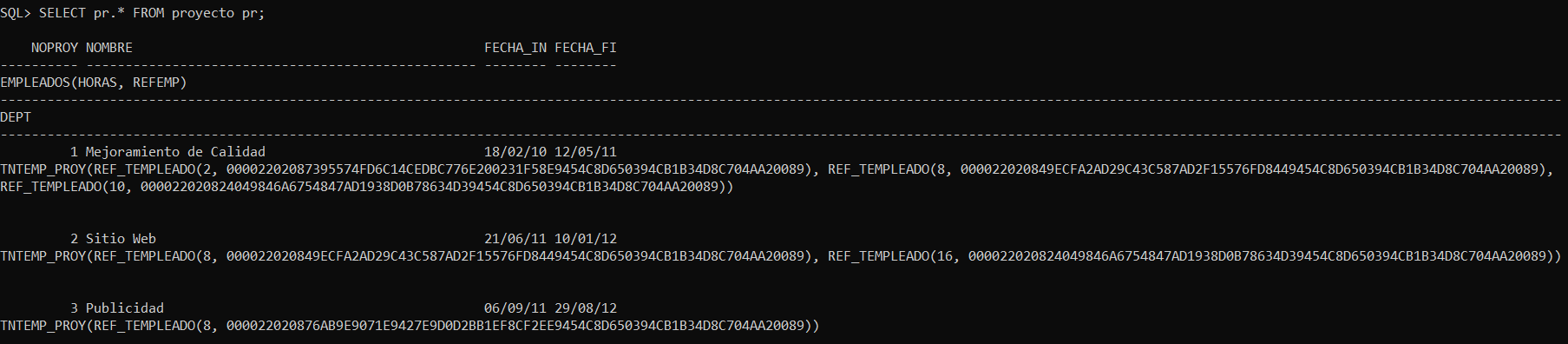
1. SELECT VALUE(d) FROM departamento d WHERE d.nodep = 1;
2. SELECT DEREF(d.jefe) FROM departamento d WHERE d.nodep = 1;

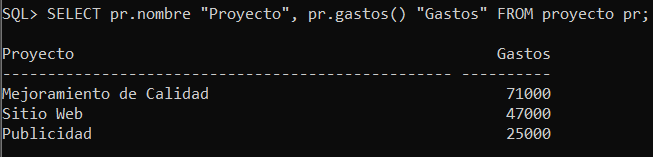


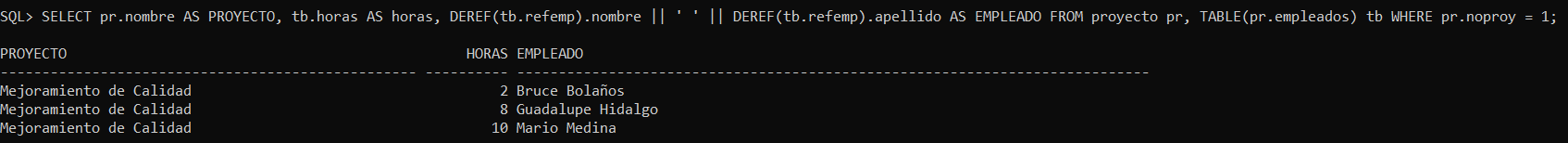
1. SELECT d.nodep AS NUMERO\_DEPARTAMENTO, d.nombre AS NOMBRE, d.jefe.nss || ':' || d.jefe.apellido || ', ' || d.jefe.nombre AS JEFE FROM departamento d;



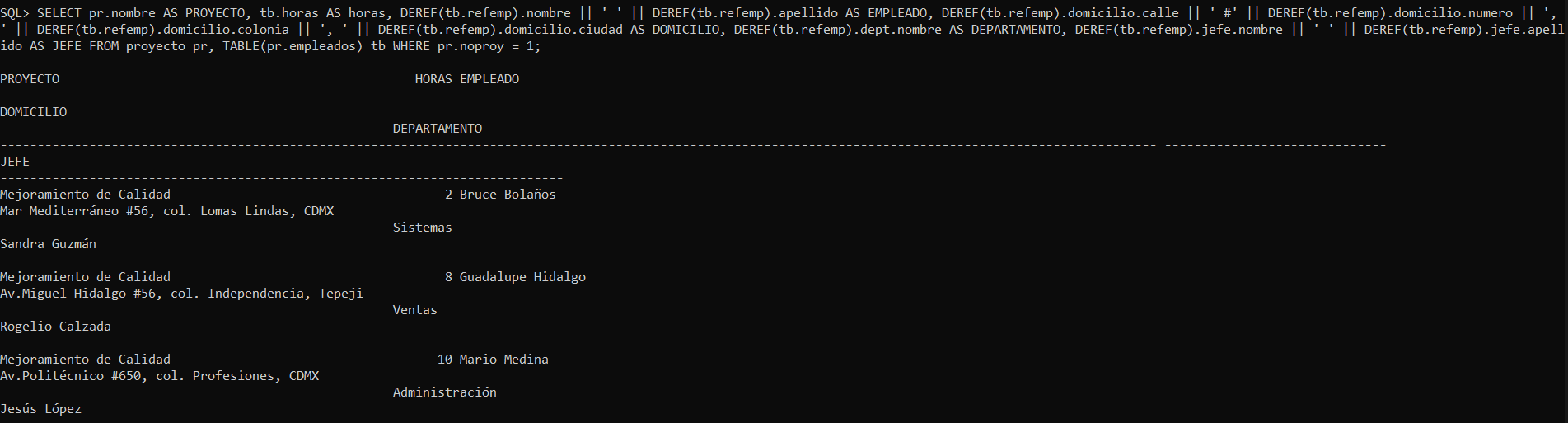
1. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 3) WHERE nss = '777888999';
2. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 3) WHERE nss = '222333444';
3. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 1) WHERE nss = '444555666';
4. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 3) WHERE nss = '555666777';
5. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 2) WHERE nss = '333444555';
6. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 1) WHERE nss = '666777888';
7. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 2) WHERE nss = '999000111';
8. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 1) WHERE nss = '111222333';
9. UPDATE empleado e SET e.dept = (SELECT REF(d) From departamento d WHERE d.nodep = 2) WHERE nss = '888999000';
10. COMMIT;
11. SELECT e.nombre || ' ' || e.apellido AS EMPLEADO, e.dept.nodep || '- ' || e.dept.nombre AS DEPARTAMENTO FROM empleado e ORDER BY DEPARTAMENTO;
12. INSERT INTO proyecto(noproy, nombre, fecha\_inicio, fecha\_fin, empleados) VALUES (1, 'Mejoramiento de Calidad',TO\_DATE('18/02/2010','dd/mm/YYYY'), TO\_DATE('12/05/2011','dd/mm/YYYY'), tNTemp\_proy(ref\_templeado(2,(SELECT REF(e) FROM empleado e WHERE e.nss = '666777888')),ref\_templeado(8,(SELECT REF(e) FROM empleado e WHERE e.nss = '888999000')),ref\_templeado(10,(SELECT REF(e) FROM empleado e WHERE e.nss = '555666777'))));
13. INSERT INTO proyecto(noproy, nombre, fecha\_inicio, fecha\_fin, empleados) VALUES (2, 'Sitio Web',TO\_DATE('21/06/2011','dd/mm/YYYY'), TO\_DATE('10/01/2012','dd/mm/YYYY'), tNTemp\_proy(ref\_templeado(8,(SELECT REF(e) FROM empleado e WHERE e.nss = '888999000')),ref\_templeado(16,(SELECT REF(e) FROM empleado e WHERE e.nss = '555666777'))));
14. INSERT INTO proyecto(noproy, nombre, fecha\_inicio, fecha\_fin, empleados) VALUES (3, 'Publicidad',TO\_DATE('06/09/2011','dd/mm/YYYY'), TO\_DATE('29/08/2012','dd/mm/YYYY'), tNTemp\_proy(ref\_templeado(8,(SELECT REF(e) FROM empleado e WHERE e.nss = '222333444'))));
15. COMMIT;
16. SELECT pr.\* FROM proyecto pr;



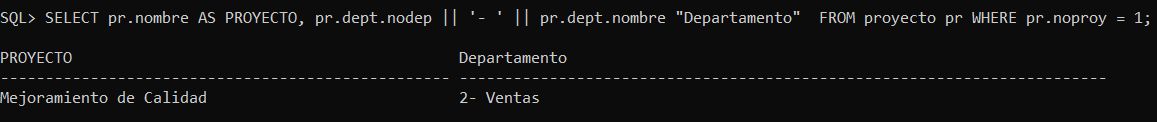
1. SELECT pr.nombre "Proyecto", pr.gastos() "Gastos" FROM proyecto pr;
2. SELECT pr.nombre AS PROYECTO, tb.horas AS horas, DEREF(tb.refemp).nombre || ' ' || DEREF(tb.refemp).apellido AS EMPLEADO FROM proyecto pr, TABLE(pr.empleados) tb WHERE pr.noproy = 1;



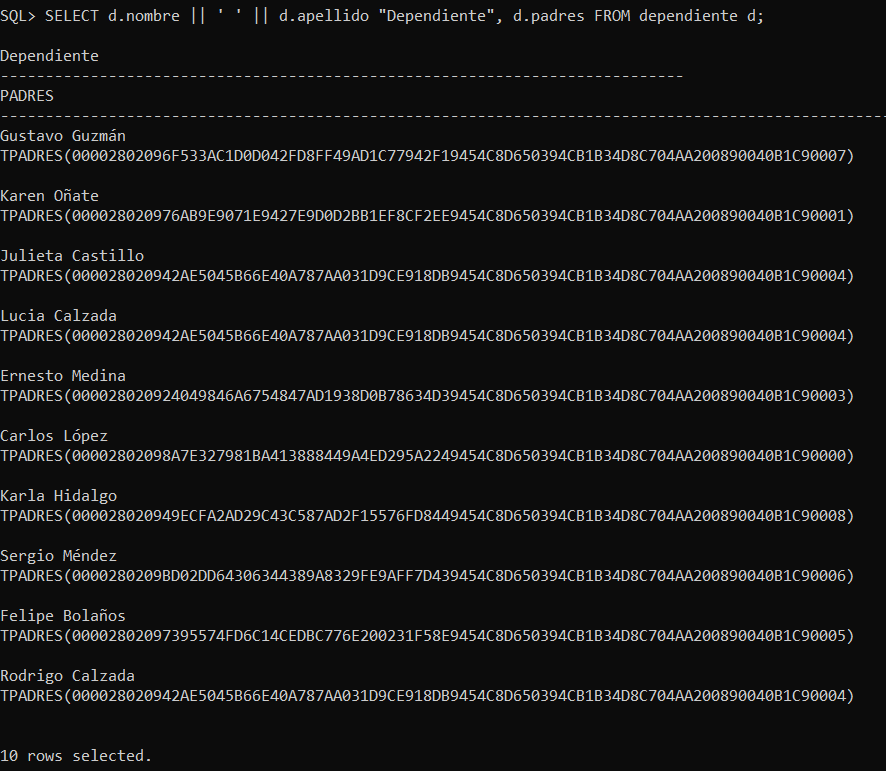
1. SELECT pr.nombre AS PROYECTO, tb.horas AS horas, DEREF(tb.refemp).nombre || ' ' || DEREF(tb.refemp).apellido AS EMPLEADO, DEREF(tb.refemp).domicilio.calle || ' #' || DEREF(tb.refemp).domicilio.numero || ', ' || DEREF(tb.refemp).domicilio.colonia || ', ' || DEREF(tb.refemp).domicilio.ciudad AS DOMICILIO, DEREF(tb.refemp).dept.nombre AS DEPARTAMENTO, DEREF(tb.refemp).jefe.nombre || ' ' || DEREF(tb.refemp).jefe.apellido AS JEFE FROM proyecto pr, TABLE(pr.empleados) tb WHERE pr.noproy = 1;



1. UPDATE proyecto pr SET pr.dept = (SELECT REF(d) FROM departamento d WHERE d.nodep = 1) WHERE pr.noproy = 2;
2. UPDATE proyecto pr SET pr.dept = (SELECT REF(d) FROM departamento d WHERE d.nodep = 2) WHERE pr.noproy = 1;
3. UPDATE proyecto pr SET pr.dept = (SELECT REF(d) FROM departamento d WHERE d.nodep = 3) WHERE pr.noproy = 3;
4. COMMIT;
5. SELECT pr.nombre AS PROYECTO, pr.dept.nodep || '- ' || pr.dept.nombre "Departamento" FROM proyecto pr WHERE pr.noproy = 1;



1. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Gustavo','Guzmán',TO\_DATE('07/12/99','dd/mm/YYYY'),'M','hijo');
2. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Karen','Oñate',TO\_DATE('09/11/95','dd/mm/YYYY'),'F','hija');
3. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Julieta','Castillo',TO\_DATE('09/11/51','dd/mm/YYYY'),'F','madre');
4. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Lucia','Calzada',TO\_DATE('09/11/95','dd/mm/YYYY'),'F','hija');
5. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Ernesto','Medina',TO\_DATE('28/10/45','dd/mm/YYYY'),'M','padre');
6. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Carlos','López',TO\_DATE('12/04/05','dd/mm/YYYY'),'M','hijo');
7. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Karla','Hidalgo',TO\_DATE('26/08/09','dd/mm/YYYY'),'F','hija');
8. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Sergio','Méndez',TO\_DATE('17/12/92','dd/mm/YYYY'),'M','hijo');
9. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Felipe','Bolaños',TO\_DATE('26/08/09','dd/mm/YYYY'),'M','nieto');
10. INSERT INTO dependiente(nombre,apellido,fecha\_nac,genero,parentesco) VALUES('Rodrigo','Calzada',TO\_DATE('21/07/97','dd/mm/YYYY'),'M','hijo');
11. COMMIT;
12. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '111222333')) WHERE d.nombre = 'Gustavo' AND d.apellido = 'Guzmán';
13. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '222333444')) WHERE d.nombre = 'Karen' AND d.apellido = 'Oñate';
14. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '333444555')) WHERE d.nombre = 'Julieta' AND d.apellido = 'Castillo';
15. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '333444555')) WHERE d.nombre = 'Lucia' AND d.apellido = 'Calzada';
16. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '555666777')) WHERE d.nombre = 'Ernesto' AND d.apellido = 'Medina';
17. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '777888999')) WHERE d.nombre = 'Carlos' AND d.apellido = 'López';
18. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '888999000')) WHERE d.nombre = 'Karla' AND d.apellido = 'Hidalgo';
19. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '999000111')) WHERE d.nombre = 'Sergio' AND d.apellido = 'Méndez';
20. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '666777888')) WHERE d.nombre = 'Felipe' AND d.apellido = 'Bolaños';
21. UPDATE dependiente d SET d.padres = tPadres((SELECT REF(e) FROM empleado e WHERE e.nss = '333444555')) WHERE d.nombre = 'Rodrigo' AND d.apellido = 'Calzada';
22. COMMIT;
23. SELECT d.nombre || ' ' || d.apellido "Dependiente", d.padres FROM dependiente d;



1. ALTER TYPE tDependiente ADD MEMBER FUNCTION obtener\_padres RETURN VARCHAR CASCADE;
2. CREATE OR REPLACE TYPE BODY tDependiente IS

CONSTRUCTOR FUNCTION tDependiente (SELF IN OUT NOCOPY tDependiente,nombre VARCHAR,apellido VARCHAR,fecha\_nac VARCHAR,genero CHAR

, parentesco VARCHAR) RETURN SELF AS RESULT

IS

BEGIN

SELF.parentesco := UPPER(parentesco);

SELF.nombre := nombre;

SELF.apellido := apellido;

SELF.fecha\_nac := TO\_DATE(REPLACE(fecha\_nac,'-','/'),'dd/mm/YYYY');

SELF.genero := UPPER(genero);

END;

MEMBER FUNCTION obtener\_padres RETURN VARCHAR IS

dep\_ref templeado;

salida VARCHAR(100) := '';

BEGIN

FOR i IN 1..SELF.padres.COUNT LOOP

UTL\_REF.SELECT\_OBJECT(padres(i), dep\_ref);

salida := salida || 'pariente: ' || dep\_ref.nombre || ' ' || dep\_ref.apellido;

END LOOP;

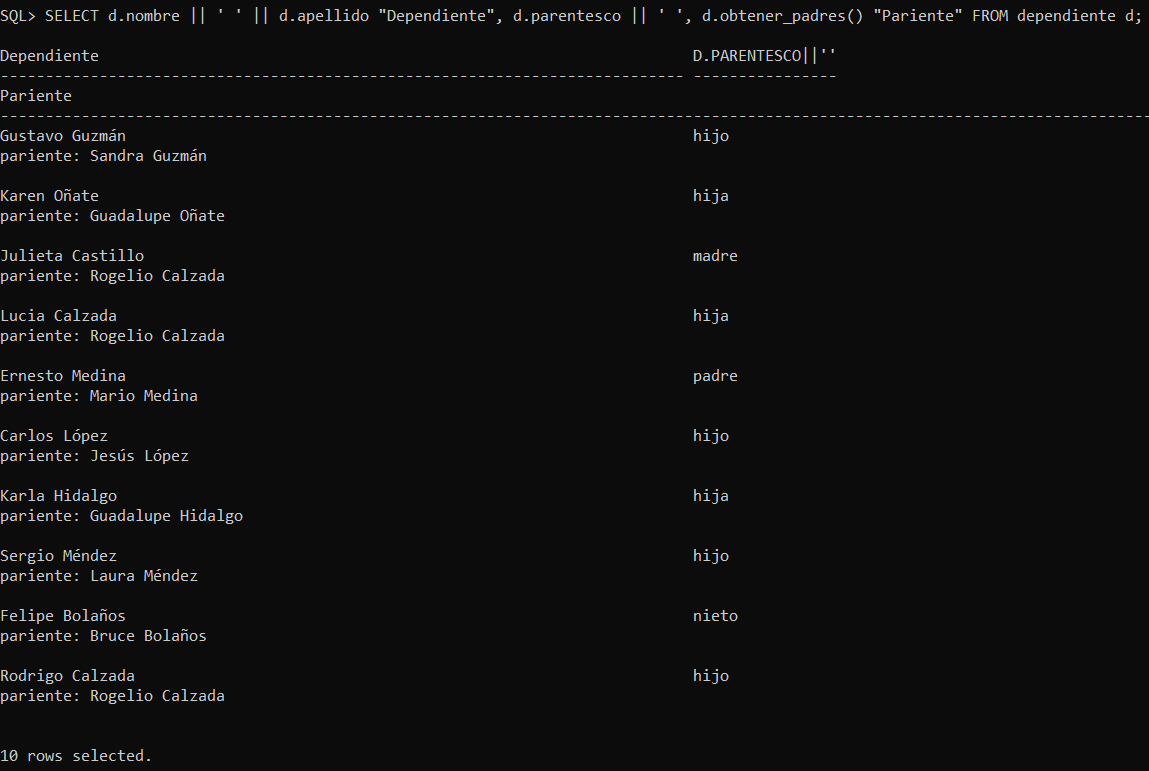
RETURN salida;

END;

END;

/

1. SELECT d.nombre || ' ' || d.apellido "Dependiente", d.parentesco || ' ', d.obtener\_padres() "Pariente" FROM dependiente d;



Incluya las capturas de pantalla (legibles y en buena resolución) con los resultados obtenidos en las líneas de código indicadas en resaltado amarillo y guarde el documento. Resuelva el cuestionario y escriba sus conclusiones. Envíe el documento para su revisión en tiempo y forma. No se aceptan archivos fuera del reporte de la práctica, ni archivos alojados en sitios externos.

CUESTIONARIO

1.- ¿Que mecanismo emplea Oracle en el modelo orientado a objetos para reemplazar a las reuniones (joins) del modelo relacional? Mencione las funciones que se emplean para su consulta.

Los mecanismos utilizados para la realización de estos joins son por pedio de VARRAYS o NESTEAD tables para poder realizar uniones entre las mismas

VARRAY

Elements from t\_names 0

Elements from s\_names 4

NESTEAD

SELECT C1.FirstName, C1.LastName

FROM AdventureWorksModel.Contact as C1

ORDER BY C1.LastName

2.- Al inserta un valor a una columna de tipo de dato compuesto, es necesario invocar a su constructor. ¿Cómo se insertan datos a una columna cuando están establecidos como referencias (REF) hacía otras tablas?

Puede insertarse desde la palabra reservada NEW para instanciar un objeto, ejemplo :

UPDATE empleado e SETe.domicilio = NEWtdireccion('Av.Siempreviva', 444, 'col. San Mateo', 'Ecatepec', '89740') WHEREnss = '111222333';

En el anterior ejemplo el constructor es tdireccion, el cual lo invoca con los parámetros correspondientes.

Mientras que una referencia puede ser de la siguiente manera:

INSERT INTO Empleado VALUES(12345, 'Juan Pérez', TO\_DATE('18/09/1997', 'dd/mm/yyyy'), (SELECTREF(d)FROM departamento d WHEREnombre\_dep = 'Ventas'));Se emplea la notación punto “.” para acceder a los datos:SELECTnombre, e.pertenece\_a.nombre\_depFROMEmpleado eWHEREDNI = 9687452;

CONCLUSIONES

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

La generación de herencias, así como la utilización de otros tipos de herramientas como lo son los VARRAYS Y NESTEAD tables nos dan la posibilidad de generar las relaciones entre las tablas para poder manejar con mayor eficiencia cada una de las mismas.

De igual forma el uso de LOBS se puede determinar como una herramienta útil, en ciertas circunstancias derivado de la necesidad que se tiene de acceder directo al sistema para poder ver las mismas.

Al insertar datos se puede observar la herencia puesto que vemos que si no tienes las tablas y el diagrama bien ejecutado, más adelante al querer hacer una actualización se colapsa todo sin la herencia bien ejecutada o escrita.

REFERENCIAS BIBLIOGRÁFICAS (APA, IEEE, ETC.)

1.- [Database Object-Relational Developer's Guide - Contents (oracle.com)](https://docs.oracle.com/cd/E11882_01/appdev.112/e11822/toc.htm)

2.- Piattini et al. (2006): Tecnología y Diseño de Bases de Datos.

3.- Connolly y Begg (2005): Sistemas de Bases de Datos.

4.- Jeffrey A. Hoffer, Mary B. Prescott, Heikki Topi. Modern Database Management, Ninth Edition.

Pearson/Prentice, Estados Unidos 2009, págs. 690

5.- Raghu Ramakrishnan, Johannes Gehrke. Sistemas de Gestión de Bases de Datos, Tercera Edición. McGrawHill/Interamericana de España, Madrid España 2007, págs. 654

6.- Ramez Elmasri, Shamkant B. Navathe. Fundamentos de Sistemas de Bases de Datos, Quinta Edición. Pearson/Addison Wesley, Madrid España 2007, págs. 988