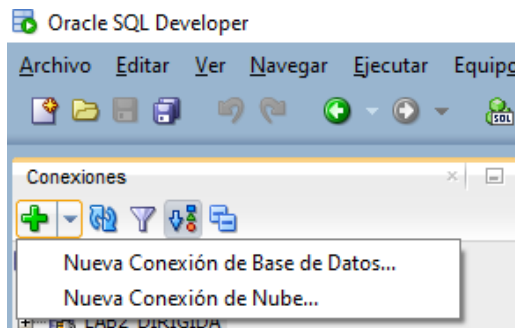


PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
Bases de Datos
4ta. Práctica Dirigida
(Primer Semestre 2022)

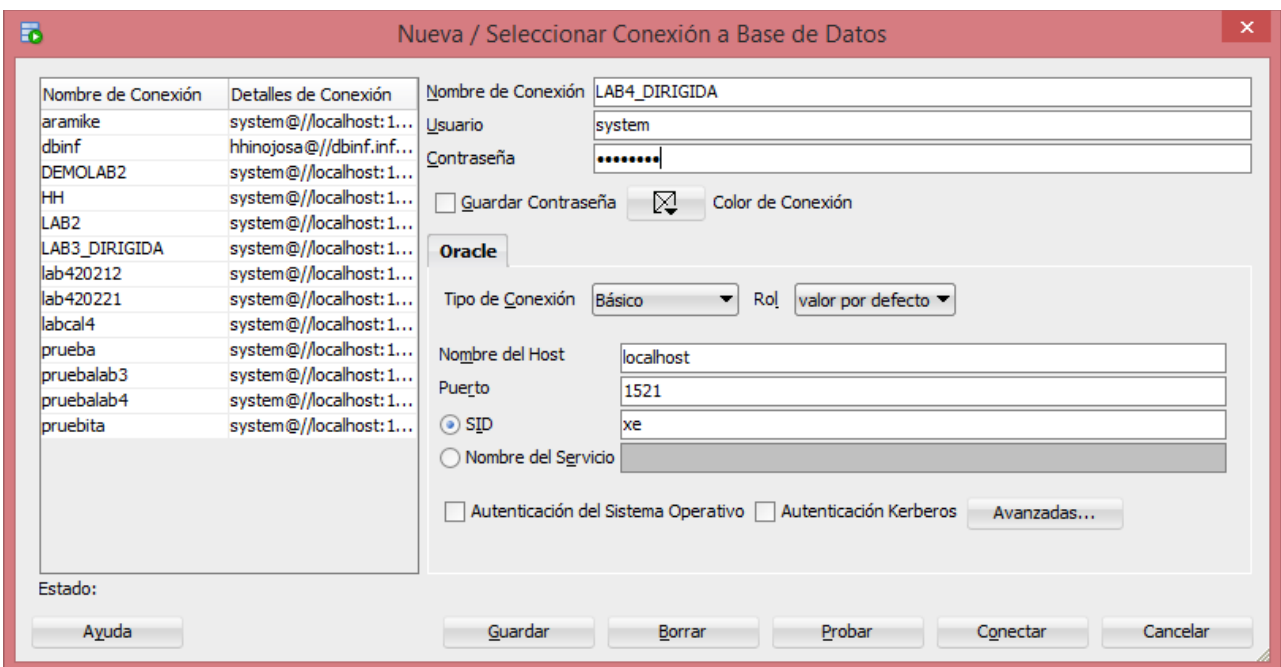
GUIA – SQL DDL – SQL DML

Conectarse a una base de datos

Primero, ejecute **Oracle SQL Developer**, cierre la pestaña Página de bienvenida, y en el panel de **Conexiones**, haga clic en el ícono + para crear una nueva conexión.



Se abrirá la siguiente ventana:



| Nombre de Conexión | Detalles de Conexión |
|--------------------|--------------------------|
| aramike | system@//localhost: 1... |
| dbinf | hhinojosa@//dbinf.inf... |
| DEMOLAB2 | system@//localhost: 1... |
| HH | system@//localhost: 1... |
| LAB2 | system@//localhost: 1... |
| LAB3_DIRIGIDA | system@//localhost: 1... |
| lab420212 | system@//localhost: 1... |
| lab420221 | system@//localhost: 1... |
| labcal4 | system@//localhost: 1... |
| prueba | system@//localhost: 1... |
| pruebalab3 | system@//localhost: 1... |
| pruebalab4 | system@//localhost: 1... |
| pruebita | system@//localhost: 1... |

Nombre de Conexión: LAB4_DIRIGIDA

Usuario: system

Contraseña:

☐ Guardar Contraseña ☐ Color de Conexión

Oracle

Tipo de Conexión: Básico Rol: valor por defecto

Nombre del Host: localhost

Puerto: 1521

☒ SID: xe

☐ Nombre del Servicio

☐ Autenticación del Sistema Operativo ☐ Autenticación Kerberos

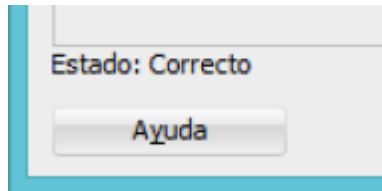
Estado:

Ingresa los siguientes datos para la conexión:

- Nombre de conexión (Name): **LAB4_DIRIGIDA**
- Usuario: **system**
- Contraseña: Debe escribir la contraseña que ingresó al instalar.

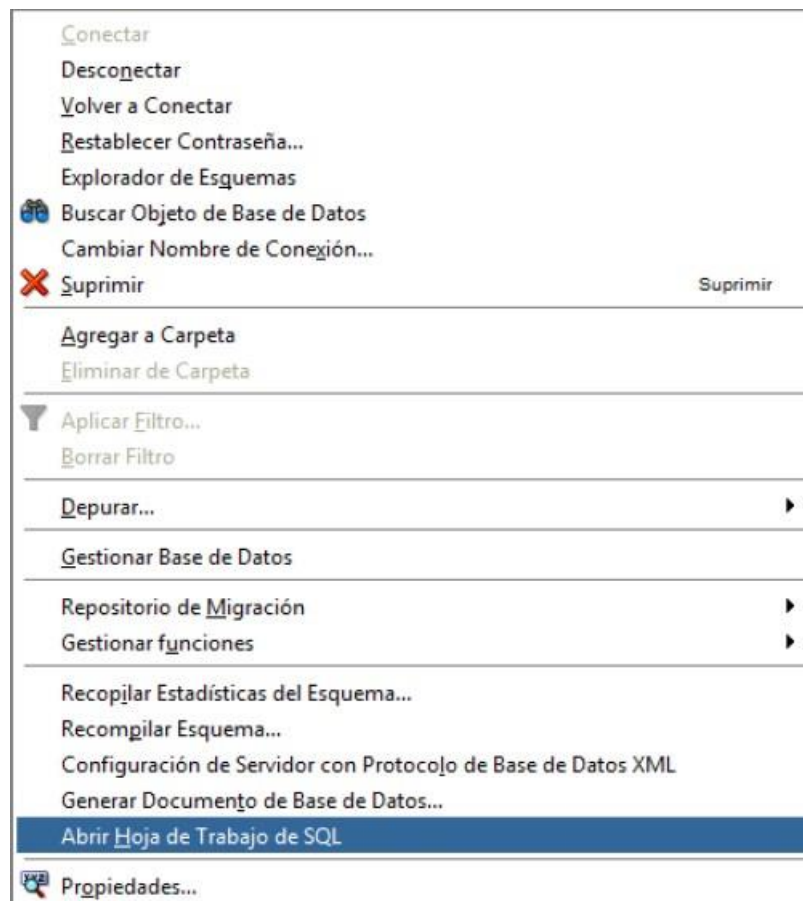
Puede dejar el resto de los parámetros tal y como aparecen por defecto.

Haga clic en el botón **Probar**. Si todo está bien configurado, aparecerá el mensaje “Estado: Correcto” en la parte inferior izquierda.

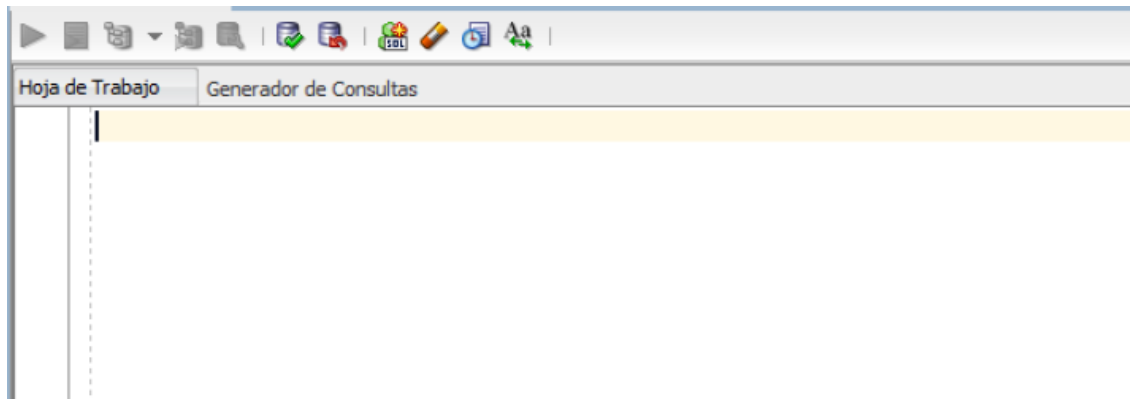


Luego, haga clic en el botón **Guardar**, y luego clic en el botón **Conectar**. El nombre de la conexión creada (LAB4_DIRIGIDA) aparecerá en el panel de Conexiones.

Haga clic derecho sobre el nombre de la conexión *LAB4_DIRIGIDA* y aparecerá el siguiente menú contextual:



Hacemos clic en **Abrir Hoja de Trabajo de SQL**. Nos saldrá una pantalla similar a la siguiente:



Esta es la hoja de trabajo, donde se podrán escribir y ejecutar los scripts SQL.

Para iniciar, ejecute los scripts que se adjuntan en la parte dirigida en el siguiente orden:

- INF246_2022-1_LAB4_01_dirigida_drops.sql
- INF246_2022-1_LAB4_02_dirigida_modelo.sql
- INF246_2022-1_LAB4_03_dirigida_inserts.sql

Ver el diagrama relacional en el **anexo 1**.

Ejercicio 1

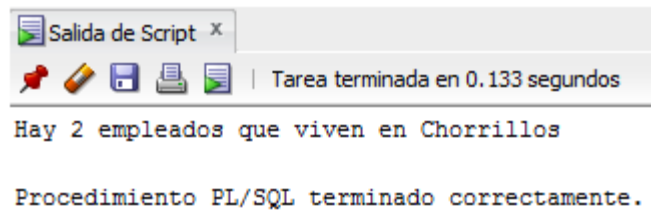
Elaborar una función que reciba como parámetro el nombre de un distrito (municipio) y que devuelva la cantidad de empleados que viven en él.

```
create or replace function f_contar_empleados_x_municipio( v_municipio varchar2)
return number
as
    v_cantidad number;
begin
    select count(*)
    into v_cantidad
    from EMPLEADOS
    where municipio = v_municipio;
    return v_cantidad;
end;
```

```

set SERVEROUTPUT ON
declare
    v_cantidad number;
    v_municipio varchar2(20);
begin
    v_municipio := 'Chorrillos';
    v_cantidad := f_contar_empleados_x_municipio( v_municipio );
    dbms_output.put_line( 'Hay ' || v_cantidad || ' empleados que viven en ' || v_municipio );
end;

```



Ejercicio 2

- Elaborar una función que reciba como parámetro una fecha y que devuelva dicha fecha en forma de frase. Por ejemplo, si la fecha que recibe la función es '16/6/2022', está devolverá la frase '16 de junio de 2022'.
- Elaborar una función que reciba como parámetro una fecha de nacimiento y que devuelva la edad correspondiente.

```

create or replace function f_fecha_frase( v_fecha date )
return varchar2
is
    v_d number;
    v_m number;
    v_a number;
    v_dia varchar2(2);
    v_mes varchar2(20);
    v_año varchar2(4);
    v_frase varchar2(30);
begin
    v_d := extract( day from v_fecha );
    v_m := extract( month from v_fecha );
    v_a := extract( year from v_fecha );
    case v_m
        when 1 then v_mes := 'enero';
        when 2 then v_mes := 'febrero';
        when 3 then v_mes := 'marzo';

```

```





        when 4 then v_mes := 'abril';
        when 5 then v_mes := 'mayo';
        when 6 then v_mes := 'junio';
        when 7 then v_mes := 'julio';
        when 8 then v_mes := 'agosto';
        when 9 then v_mes := 'setiembre';
        when 10 then v_mes := 'octubre';
        when 11 then v_mes := 'noviembre';
        else v_mes := 'diciembre';
    end case;
    v_dia := to_char( v_d );
    v_año := to_char( v_a );
    v_frase := v_dia || ' de ' || v_mes || ' de ' || v_año;
    return v_frase;
end;
```

```

create or replace function f_obtener_edad( v_fecha_nacimiento date )
return number
is
    v_meses number;
    v_edad number;
begin
    v_meses := months_between( sysdate, v_fecha_nacimiento );
    v_edad := trunc( v_meses / 12 );
    return v_edad;
end;
```

```

select apellido1 || ' ' || apellido2 || ' ' || nombre as Empleado,
       f_fecha_frase( fecha_nac ) as Fecha_Nacimiento,
       f_obtener_edad( fecha_nac ) as Edad
from EMPLEADOS;
```

| Resultado de la Consulta x | | | |
|--|--------------------------------|-------------------------|------|
|     SQL Todas las Filas Recuperadas: 7 en 0.007 segundos | | | |
| | EMPLEADO | FECHA_NACIMIENTO | EDAD |
| 1 | Mercal Lopez Jose | 5 de enero de 1974 | 48 |
| 2 | Rosal Cruz Maria | 15 de julio de 1970 | 51 |
| 3 | Perez Rollen Pilar | 2 de agosto de 1978 | 43 |
| 4 | Casas Frias Luis Felipe | 2 de agosto de 1981 | 40 |
| 5 | Segura Chipoco Beatriz Rosario | 25 de noviembre de 1966 | 55 |
| 6 | Palomino Cuya Mercedes | 12 de marzo de 1977 | 45 |
| 7 | Soto Aguilar Harry | 2 de agosto de 1971 | 50 |

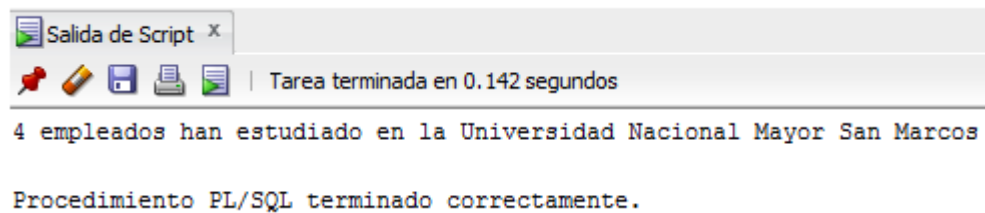
Ejercicio 3

Elaborar un procedimiento que reciba como parámetro el nombre de una universidad y que muestre en pantalla la cantidad de personas que han estudiado en dicha universidad. Si el nombre de la universidad no está registrado en la base de datos, mostrar el mensaje correspondiente.

```
create or replace procedure p_cant_empleados_universidad( v_nom_univ varchar2 )
is
    v_cantidad number;
begin
    select count(*) into v_cantidad
    from UNIVERSIDADES
    where nombre_univ = v_nom_univ;
    if v_cantidad = 0 then
        dbms_output.put_line( 'Nombre de universidad no registrado en la base
                                de datos' );
    else
        select count(*) into v_cantidad
        from UNIVERSIDADES U, ESTUDIOS E
        where U.univ_cod = E.universidad and
              U.nombre_univ = v_nom_univ;
        dbms_output.put_line( v_cantidad || ' empleados han estudiado en la
                                Universidad ' || v_nom_univ );
    end if;
end;
```

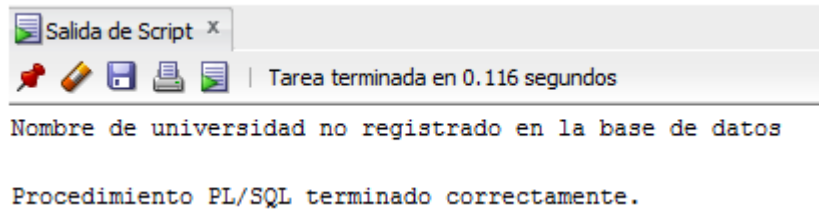
set SERVEROUTPUT ON

exec p_cant_empleados_universidad('Nacional Mayor San Marcos');



set SERVEROUTPUT ON

exec p_cant_empleados_universidad('ABC');



Ejercicio 4

Elaborar un procedimiento que permita incrementar el presupuesto del departamento que tenga el menor presupuesto. El monto del incremento debe ser un parámetro del procedimiento.

```
create or replace procedure p_incrementar_presupuesto( v_monto number )
is
    v_menor_presupuesto number;
begin
    select min(presupuesto) into v_menor_presupuesto
    from DEPARTAMENTOS;
    update DEPARTAMENTOS
    set presupuesto = presupuesto + v_monto
    where presupuesto = v_menor_presupuesto;
end;
```

Obtenemos los presupuestos de todos los departamentos:

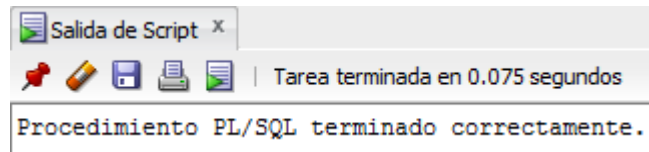
```
select * from DEPARTAMENTOS;
```

| DPTO_COD | NOMBRE_DPTO | JEFE | PRESUPUESTO | PRES_ACTUAL |
|----------|--------------|----------|-------------|-------------|
| 1 | MARKETING | 10684781 | 80000 | 50000 |
| 2 | CONTABILIDAD | 07845147 | 30000 | 10520 |
| 3 | INGENIERIA | 38744043 | 50000 | 38790 |
| 4 | LOGISTICA | 37480048 | 10000 | 5790 |

Se observa que el departamento de Logística tiene el menor presupuesto.

Ejecutamos el procedimiento:

```
exec p_incrementar_presupuesto( 3000 );
```



Obtenemos nuevamente los presupuestos de todos los departamentos:

`select * from DEPARTAMENTOS;`

| | DPTO_COD | NOMBRE_DPTO | JEFE | PRESUPUESTO | PRES_ACTUAL |
|---|----------|--------------|----------|-------------|-------------|
| 1 | 1 | MARKETING | 10684781 | 80000 | 50000 |
| 2 | 2 | CONTABILIDAD | 07845147 | 30000 | 10520 |
| 3 | 3 | INGENIERIA | 38744043 | 50000 | 38790 |
| 4 | 4 | LOGISTICA | 37480048 | 13000 | 5790 |

Se observa que se ha incrementado en 3000 soles el presupuesto del departamento de Logística.

Ejercicio 5

Elaborar un procedimiento que devuelva dos valores: el mayor y el menor salarios históricos de un empleado cuyo nombre, apellido paterno y apellido materno se pasan como parámetros.

```
create or replace procedure p_obtener_mayor_menor_salario( v_ape_paterno
    varchar2, v_ape_materno varchar2, v_nombre varchar2,
    v_mayor_salario out number, v_menor_salario out number )
is
begin
    select max(salario), min(salario) into v_mayor_salario, v_menor_salario
    from empleados E, historial_salarial HS
    where E.dni = HS.empleado_dni and nombre = v_nombre
        and apellido1 = v_ape_paterno and apellido2 = v_ape_materno;
end;
```

`set SERVEROUTPUT ON`

`declare`

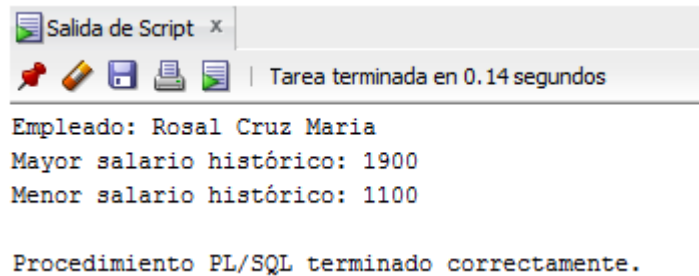
```
    v_ape_paterno varchar2(15);
    v_ape_materno varchar2(15);
    v_nombre varchar2(40);
    v_mayor_salario number;
    v_menor_salario number;
```



```

begin
    v_ape_paterno := 'Rosal';
    v_ape_materno := 'Cruz';
    v_nombre := 'Maria';
    dbms_output.put_line( 'Empleado: ' || v_ape_paterno || ' ' || v_ape_materno || ' ' || v_nombre);
    p_obtener_mayor_menor_salario( v_ape_paterno, v_ape_materno, v_nombre,
                                   v_mayor_salario, v_menor_salario );
    if ( v_mayor_salario is not null and v_menor_salario is not null ) then
        dbms_output.put_line( 'Mayor salario histórico: ' || v_mayor_salario );
        dbms_output.put_line( 'Menor salario histórico: ' || v_menor_salario );
    else
        dbms_output.put_line( 'No se tienen registrados datos para este empleado' );
    end if;
end;

```



Ejercicio 6

Elaborar un procedimiento que permita mostrar en pantalla los nombres de los jefes de cada departamento.

```

create or replace procedure p_mostrar_jefes_departamentos
is
    cursor c is
        select nombre_dpto, nombre || ' ' || apellido1 || ' ' || apellido2,
               sexo
        from DEPARTAMENTOS D, EMPLEADOS E
        where D.jefe = E.dni;
    v_departamento varchar2(20);
    v_jefe varchar2(50);
    v_sexo char(1);
    v_frase varchar2(20);
begin
    open c;
    loop
        fetch c into v_departamento, v_jefe, v_sexo;
        exit when c%NOTFOUND;
    
```

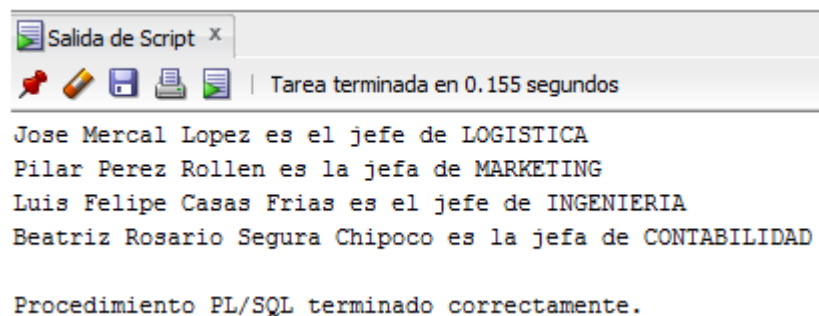
```

        if ( v_sexo = 'M' ) then
            v_frase := ' es la jefa de ';
        else
            v_frase := ' es el jefe de ';
        end if;
        dbms_output.put_line( v_jefe || v_frase || v_departamento );
    end loop;
    close c;
end;

```

set SERVEROUTPUT ON

exec p_mostrar_jefes_departamentos;



```

Salida de Script x
Tarea terminada en 0.155 segundos

Jose Mercal Lopez es el jefe de LOGISTICA
Pilar Perez Rollen es la jefa de MARKETING
Luis Felipe Casas Frias es el jefe de INGENIERIA
Beatriz Rosario Segura Chipoco es la jefa de CONTABILIDAD

Procedimiento PL/SQL terminado correctamente.

```

Ejercicio 7

Elaborar un procedimiento que permita mostrar en pantalla los nombres de todos los empleados que han laborado en cada departamento indicando el rango de fechas en que trabajaron en el mismo.

```

create or replace procedure p_mostrar_empleados_x_dpto
is
    cursor c1 is
        select dpto_cod, nombre_dpto
        from DEPARTAMENTOS;

    cursor c2( p_id_dpto number ) is
        select apellido1 || ' ' || apellido2 || ' ' || nombre || ' ( ' ||
            fecha_inicio || ' - ' || nvl(to_char(fecha_fin), 'Hoy') || ' ) '
        from EMPLEADOS E, HISTORIAL_LABORAL HL
        where E.dni = HL.empleado_dni and HL.dpto_cod = p_id_dpto;

    v_departamento varchar2(20);
    v_id_dpto number;
    v_empleado varchar2(50);

```


```

begin
  open c1;
  loop
    fetch c1 into v_id_dpto, v_departamento;
    exit when c1%NOTFOUND;
    dbms_output.put_line( '-----');
    dbms_output.put_line( v_departamento );
    open c2( v_id_dpto );
    loop
      fetch c2 into v_empleado;
      exit when c2%NOTFOUND;
      dbms_output.put_line( v_empleado );
    end loop;
    close c2;
  end loop;
  close c1;
end;

```

set SERVEROUTPUT ON

exec p_mostrar_empleados_x_dpto;



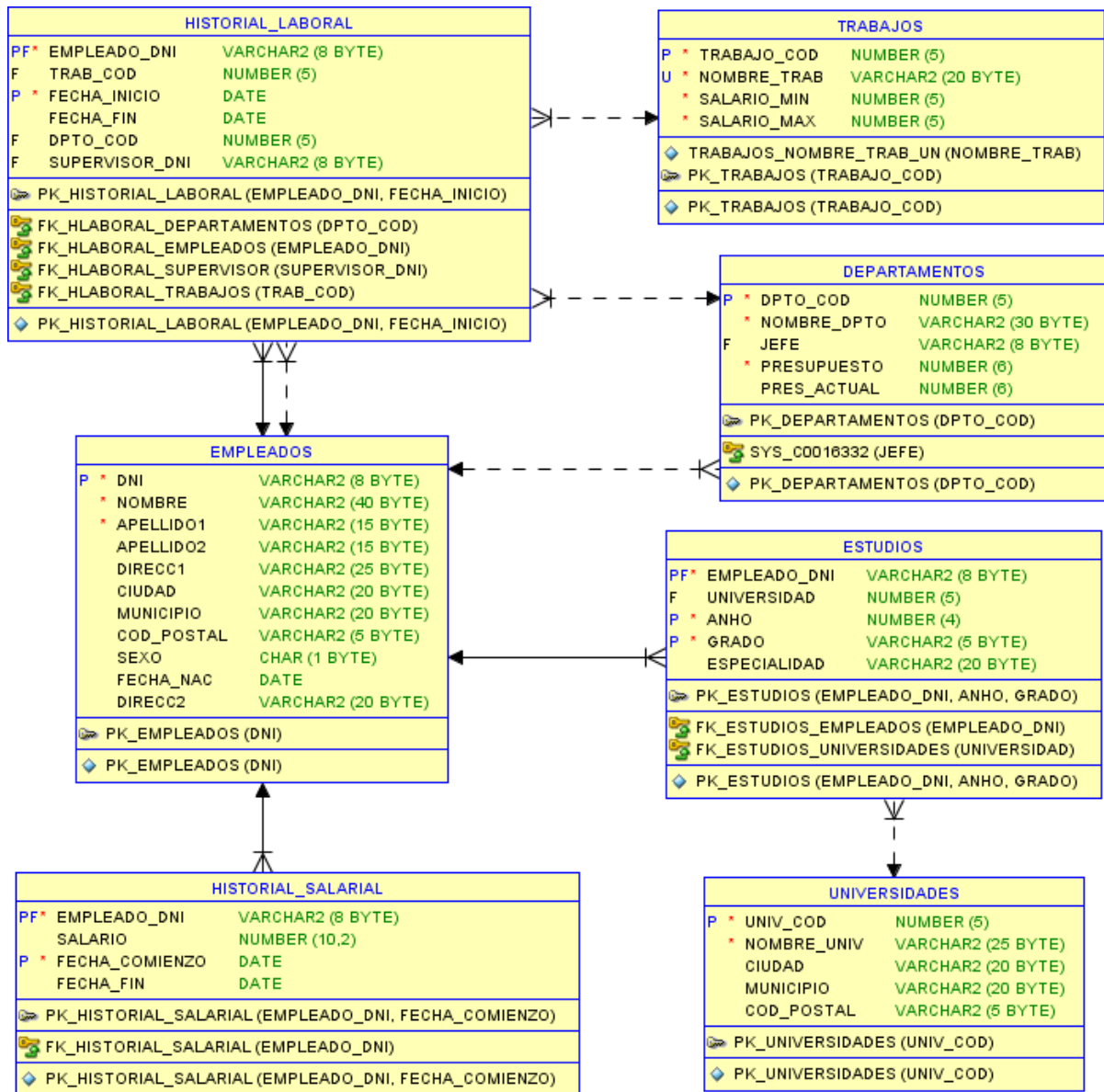
```

-----
MARKETING
Mercal Lopez Jose ( 05/01/03 - Hoy )
Rosal Cruz Maria ( 03/11/05 - Hoy )
Perez Rollen Pilar ( 03/01/04 - Hoy )
Casas Frias Luis Felipe ( 15/01/01 - 15/09/02 )
-----
CONTABILIDAD
Rosal Cruz Maria ( 03/11/04 - 30/10/05 )
-----
INGENIERIA
Casas Frias Luis Felipe ( 16/09/02 - Hoy )
Segura Chipoco Beatriz Rosario ( 02/07/02 - Hoy )
Palomino Cuya Mercedes ( 04/05/04 - Hoy )
-----
LOGISTICA
Perez Rollen Pilar ( 03/11/99 - 30/12/03 )
Soto Aguilar Harry ( 03/08/07 - Hoy )

Procedimiento PL/SQL terminado correctamente.

```

Anexo 1



16 de junio del 2022

HH