**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas



**LABORATORIO No.03**

**BLOQUES ANÓNIMOS**

**ASIGNATURA:**

Administración de Base de Datos – Sección 3

**DOCENTE:**

Chávez Soto, Jorge Luis

**INTEGRANTE:**

* Muñoz Aliaga, Bryan Stephano **-** 23200038

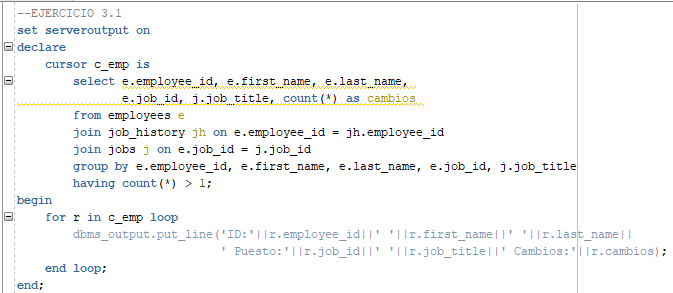
**Lima - Perú**

**02 – 10 – 2025**

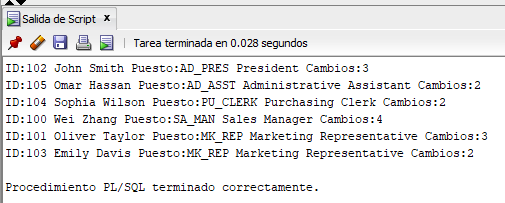
## PRIMER EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Obtener y presentar Código de Empleado, Nombres y Apellidos, Código de Puesto actual y Nombre de Puesto actual, de los N empleados que han rotado más de puesto desde que ingresaron a la empresa, para cada uno de ellos presente como columna adicional el número de veces que han cambiado de puesto.

**CÓDIGO:**

****

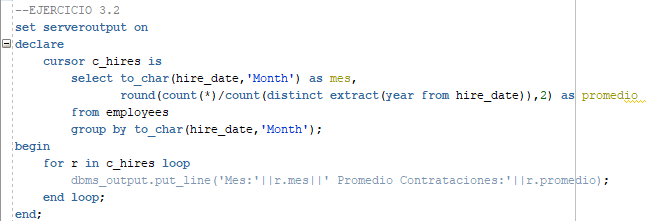
**SALIDA:**

****

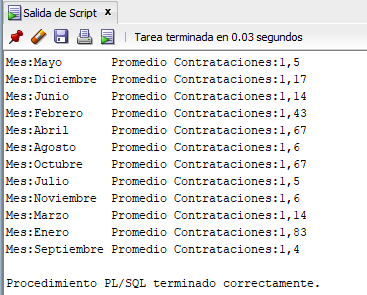
## SEGUNDO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Elabore un resumen estadístico del número promedio de contrataciones por cada mes con respecto a todos los años que hay información en la base de datos. Debe presentar sólo dos columnas: Nombre del Mes y Número Promedio de Contrataciones en ese Mes.

**CÓDIGO:**

****

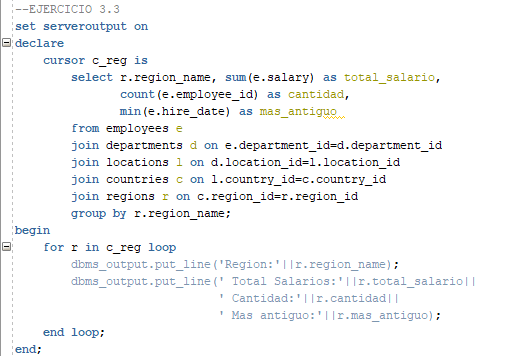
**SALIDA:**

****

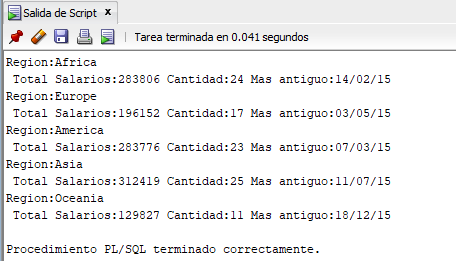
## TERCER EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Obtener y presentar la información de gastos en salario y estadística de empleados a nivel regional. Como encabezado de cada región presente el nombre de la región y en la siguiente línea indique como detalle la suma de salarios, cantidad de empleados, y fecha de ingreso del empleado más antiguo.

**CODIGO:**

****

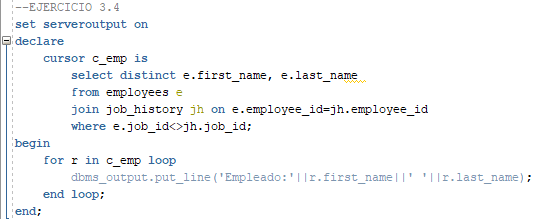
**SALIDA:**



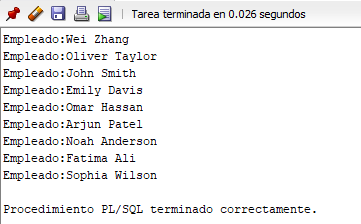
## CUARTO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Obtenga todos los nombres de los empleados que estén registrado en el historial de sueldo y que no desempeñen el mismo que el puesto actual.

**CODIGO:**

****

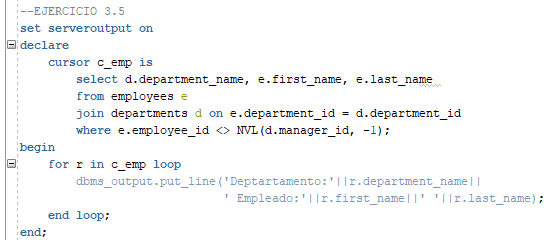
**SALIDA:**



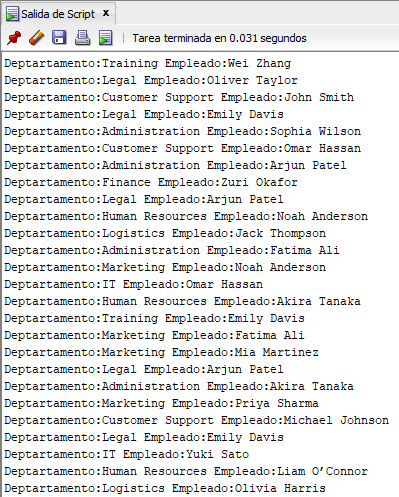
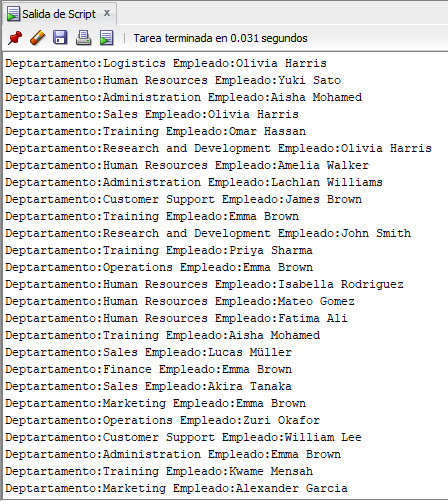
## QUINTO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

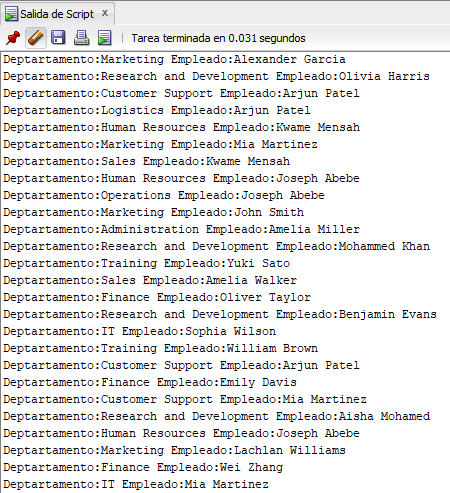
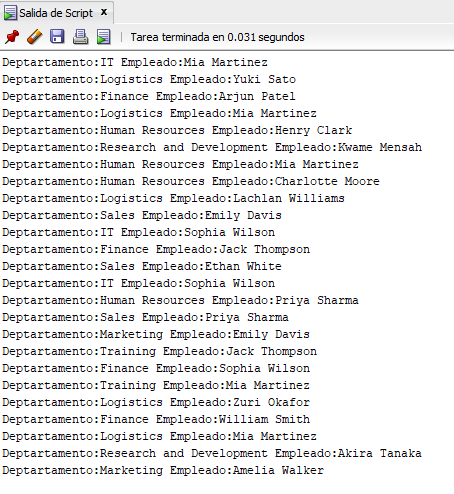
Obtenga todos los nombres de departamentos, los nombres y apellidos de los empleados que no son jefes de departamento (Managment\_ID en Departments).

**CODIGO:**



**SALIDA:**

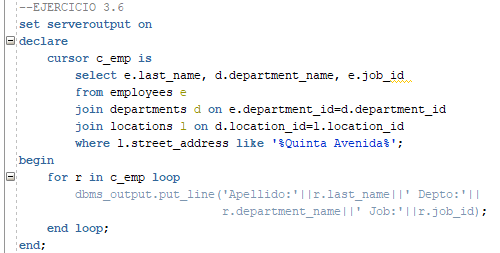
 

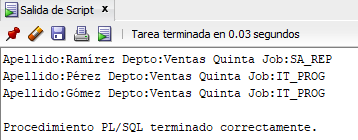
## SEXTO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Obtenga el apellido del empleado, nombre de departamento y id de puesto de todos los empleados cuya dirección de localización del departamento está en la quinta avenida (columna street\_address).

**CODIGO:**

****

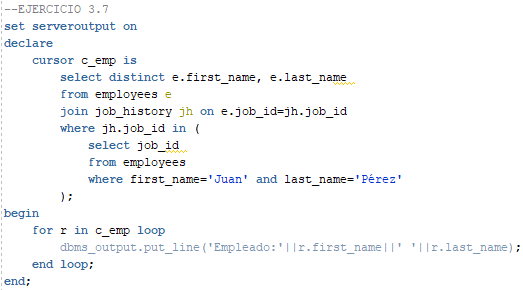
**SALIDA:**



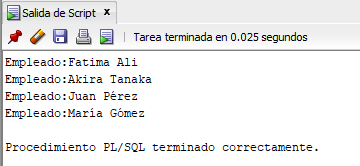
## SEPTIMO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Obtenga los nombres y apellidos de todos los empleados que hayan desempeñado los mismos puestos que el empleado Juan Pérez.

**CODIGO:**

****

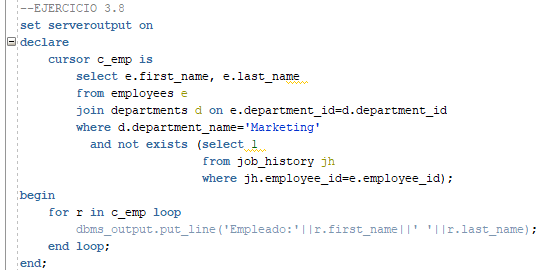
**SALIDA:**



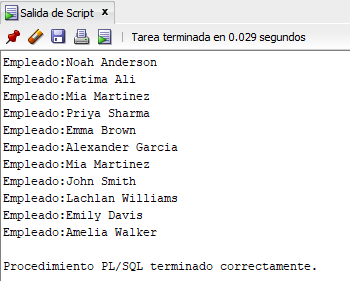
## OCTAVO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Obtenga los nombres de los empleados que no estén registrados en el historial de sueldo (Job\_history) y que trabajen en el departamento de marketing.

**CODIGO:**



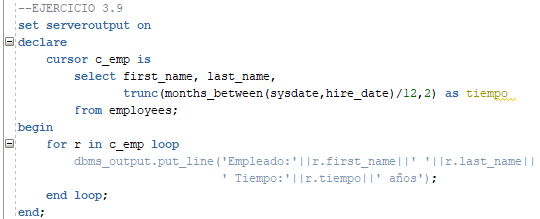
**SALIDA:**



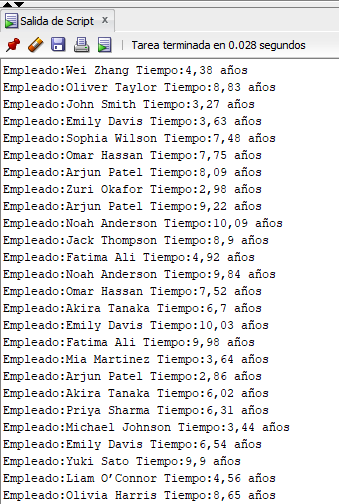
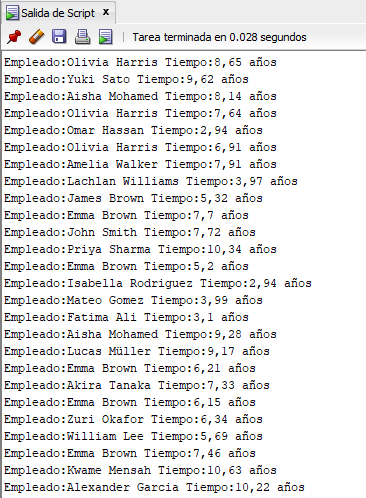
## NOVENO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

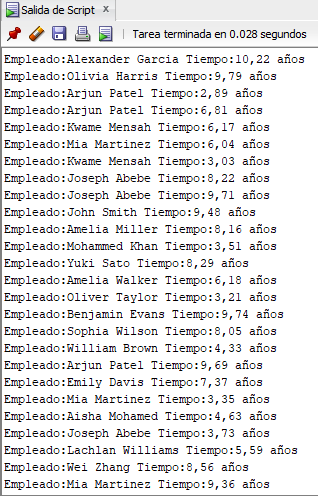
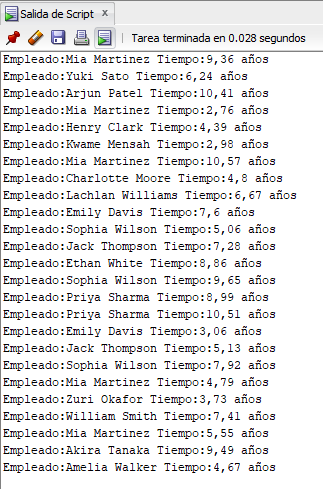
Obtenga todos los nombres de empleados y su tiempo de servicio en la empresa.

**CODIGO:**

****

**SALIDA:**

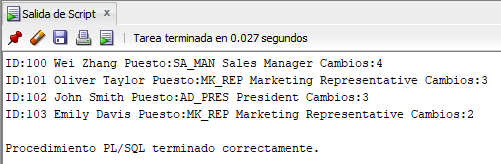
## DECIMO EJERCICIO PLANTEADO DE BLOQUES ANÓNIMOS

Obtenga el código de empleado, apellido y nombre, código de puesto actual y nombre de puesto actual, de los 4 empleados que más han rotado de puesto desde que ingresaron a la empresa (tabla Job\_History registra los cambios de puesto). Para cada uno de ellos presente como columna adicional el número de veces que han cambiado de puesto.

**CODIGO:**



**SALIDA:**



## SCRIP DE LOS EJERCICIOS

--EJERCICIO 3.1

set serveroutput on

declare

cursor c\_emp is

select e.employee\_id, e.first\_name, e.last\_name,

e.job\_id, j.job\_title, count(\*) as cambios

from employees e

join job\_history jh on e.employee\_id = jh.employee\_id

join jobs j on e.job\_id = j.job\_id

group by e.employee\_id, e.first\_name, e.last\_name, e.job\_id, j.job\_title

having count(\*) > 1;

begin

for r in c\_emp loop

dbms\_output.put\_line('ID:'||r.employee\_id||' '||r.first\_name||' '||r.last\_name||

' Puesto:'||r.job\_id||' '||r.job\_title||' Cambios:'||r.cambios);

end loop;

end;

--EJERCICIO 3.2

set serveroutput on

declare

cursor c\_hires is

select to\_char(hire\_date,'Month') as mes,

round(count(\*)/count(distinct extract(year from hire\_date)),2) as promedio

from employees

group by to\_char(hire\_date,'Month');

begin

for r in c\_hires loop

dbms\_output.put\_line('Mes:'||r.mes||' Promedio Contrataciones:'||r.promedio);

end loop;

end;

--EJERCICIO 3.3

set serveroutput on

declare

cursor c\_reg is

select r.region\_name, sum(e.salary) as total\_salario,

count(e.employee\_id) as cantidad,

min(e.hire\_date) as mas\_antiguo

from employees e

join departments d on e.department\_id=d.department\_id

join locations l on d.location\_id=l.location\_id

join countries c on l.country\_id=c.country\_id

join regions r on c.region\_id=r.region\_id

group by r.region\_name;

begin

for r in c\_reg loop

dbms\_output.put\_line('Region:'||r.region\_name);

dbms\_output.put\_line(' Total Salarios:'||r.total\_salario||

' Cantidad:'||r.cantidad||

' Mas antiguo:'||r.mas\_antiguo);

end loop;

end;

--EJERCICIO 3.4

set serveroutput on

declare

cursor c\_emp is

select distinct e.first\_name, e.last\_name

from employees e

join job\_history jh on e.employee\_id=jh.employee\_id

where e.job\_id<>jh.job\_id;

begin

for r in c\_emp loop

dbms\_output.put\_line('Empleado:'||r.first\_name||' '||r.last\_name);

end loop;

end;

--EJERCICIO 3.5

set serveroutput on

declare

cursor c\_emp is

select d.department\_name, e.first\_name, e.last\_name

from employees e

join departments d on e.department\_id = d.department\_id

where e.employee\_id <> NVL(d.manager\_id, -1);

begin

for r in c\_emp loop

dbms\_output.put\_line('Deptartamento:'||r.department\_name||

' Empleado:'||r.first\_name||' '||r.last\_name);

end loop;

end;

--EJERCICIO 3.6

set serveroutput on

declare

cursor c\_emp is

select e.last\_name, d.department\_name, e.job\_id

from employees e

join departments d on e.department\_id=d.department\_id

join locations l on d.location\_id=l.location\_id

where l.street\_address like '%Quinta Avenida%';

begin

for r in c\_emp loop

dbms\_output.put\_line('Apellido:'||r.last\_name||' Depto:'||

r.department\_name||' Job:'||r.job\_id);

end loop;

end;

--EJERCICIO 3.7

set serveroutput on

declare

cursor c\_emp is

select distinct e.first\_name, e.last\_name

from employees e

join job\_history jh on e.job\_id=jh.job\_id

where jh.job\_id in (

select job\_id

from employees

where first\_name='Juan' and last\_name='Pérez'

);

begin

for r in c\_emp loop

dbms\_output.put\_line('Empleado:'||r.first\_name||' '||r.last\_name);

end loop;

end;

--EJERCICIO 3.8

set serveroutput on

declare

cursor c\_emp is

select e.first\_name, e.last\_name

from employees e

join departments d on e.department\_id=d.department\_id

where d.department\_name='Marketing'

and not exists (select 1

from job\_history jh

where jh.employee\_id=e.employee\_id);

begin

for r in c\_emp loop

dbms\_output.put\_line('Empleado:'||r.first\_name||' '||r.last\_name);

end loop;

end;

--EJERCICIO 3.9

set serveroutput on

declare

cursor c\_emp is

select first\_name, last\_name,

trunc(months\_between(sysdate,hire\_date)/12,2) as tiempo

from employees;

begin

for r in c\_emp loop

dbms\_output.put\_line('Empleado:'||r.first\_name||' '||r.last\_name||

' Tiempo:'||r.tiempo||' años');

end loop;

end;

--EJERCICIO 3.10

set serveroutput on

declare

cursor c\_emp is

select \*

from (

select e.employee\_id, e.first\_name, e.last\_name,

e.job\_id, j.job\_title, count(\*) as cambios

from employees e

join job\_history jh on e.employee\_id=jh.employee\_id

join jobs j on e.job\_id=j.job\_id

group by e.employee\_id, e.first\_name, e.last\_name, e.job\_id, j.job\_title

order by count(\*) desc

)

where rownum<=4;

begin

for r in c\_emp loop

dbms\_output.put\_line('ID:'||r.employee\_id||' '||r.first\_name||' '||r.last\_name||

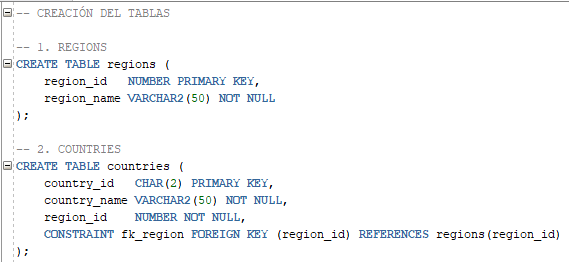
' Puesto:'||r.job\_id||' '||r.job\_title||' Cambios:'||r.cambios);

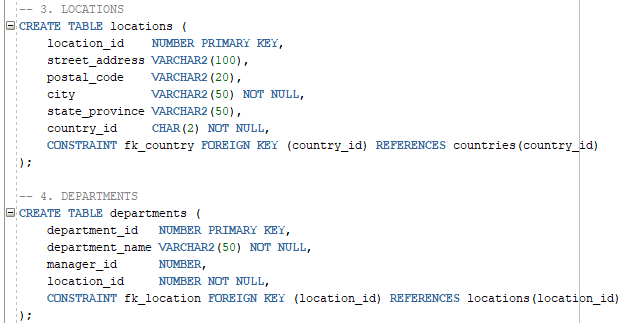
end loop;

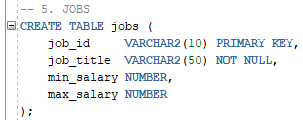
end;

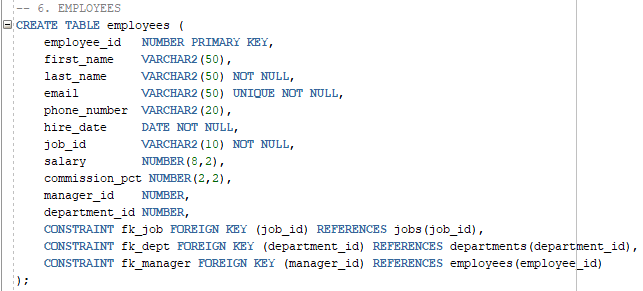
## CODIGO DE LAS TABLAS

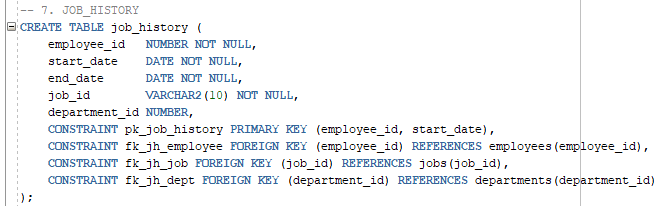
**CÓDIGO:**











**CODIGO:**

-- EMPLEADOS

CREATE TABLE EMPLEADOS (

DNI NUMBER(8) PRIMARY KEY,

NOMBRE VARCHAR(10) NOT NULL,

APELLIDO1 VARCHAR(15) NOT NULL,

APELLIDO2 VARCHAR(15),

DIRECC1 VARCHAR(40),

DIRECC2 VARCHAR(20),

CIUDAD VARCHAR(20),

PROVINCIA VARCHAR(20),

COD\_POSTAL VARCHAR(5),

SEXO VARCHAR(1) CHECK (SEXO IN ('H', 'M')),

FECHA\_NAC DATE,

PRES\_ACTUAL NUMBER,

TELEFONO VARCHAR(15),

CELULAR VARCHAR(15),

VALORACION NUMBER DEFAULT 5 CHECK (VALORACION BETWEEN 1 AND 10)

);

-- DEPARTAMENTOS

CREATE TABLE DEPARTAMENTOS (

DPTO\_COD NUMBER(5) PRIMARY KEY,

NOMBRE\_DPTO VARCHAR(30) NOT NULL UNIQUE,

DPTO\_PADRE NUMBER(5),

PRESUPUESTO NUMBER NOT NULL

);

-- UNIVERSIDADES

CREATE TABLE UNIVERSIDADES (

UNIV\_COD NUMBER(5) PRIMARY KEY,

NOMBRE\_UNIV VARCHAR(25),

CIUDAD VARCHAR(20),

MUNICIPIO VARCHAR(2),

COD\_POSTAL VARCHAR(5)

);

-- ESTUDIOS

CREATE TABLE ESTUDIOS (

EMPLEADO\_DNI NUMBER(8),

UNIVERSIDAD NUMBER(5),

AÑO NUMBER,

GRADO VARCHAR(3),

ESPECIALIDAD VARCHAR(20),

FOREIGN KEY (EMPLEADO\_DNI) REFERENCES EMPLEADOS(DNI),

FOREIGN KEY (UNIVERSIDAD) REFERENCES UNIVERSIDADES(UNIV\_COD) ON DELETE SET NULL

);

-- TRABAJOS

CREATE TABLE TRABAJOS (

TRABAJO\_COD NUMBER(5) PRIMARY KEY,

NOMBRE\_TRAB VARCHAR(20) UNIQUE,

SALARIO\_MIN NUMBER(2) NOT NULL,

SALARIO\_MAX NUMBER(2) NOT NULL

);

-- HISTORIAL\_LABORAL

CREATE TABLE HISTORIAL\_LABORAL (

EMPLEADO\_DNI NUMBER(8),

TRABAJO\_COD NUMBER(5),

FECHA\_INICIO DATE,

FECHA\_FIN DATE,

DPTO\_COD NUMBER(5),

SUPERVISOR\_DNI NUMBER(8),

PRIMARY KEY (EMPLEADO\_DNI, FECHA\_INICIO),

FOREIGN KEY (EMPLEADO\_DNI) REFERENCES EMPLEADOS(DNI),

FOREIGN KEY (TRABAJO\_COD) REFERENCES TRABAJOS(TRABAJO\_COD),

FOREIGN KEY (DPTO\_COD) REFERENCES DEPARTAMENTOS(DPTO\_COD),

FOREIGN KEY (SUPERVISOR\_DNI) REFERENCES EMPLEADOS(DNI)

);

-- HISTORIAL\_SALARIAL

CREATE TABLE HISTORIAL\_SALARIAL (

EMPLEADO\_DNI NUMBER(8),

SALARIO NUMBER NOT NULL,

FECHA\_COMIENZO DATE,

FECHA\_FIN DATE,

PRIMARY KEY (EMPLEADO\_DNI, FECHA\_COMIENZO),

FOREIGN KEY (EMPLEADO\_DNI) REFERENCES EMPLEADOS(DNI)

);

## SCRIPT DE LAS INSERCIONES

[**https://github.com/BryanSMA/LAB03\_MUNOZ\_BRYAN/blob/0da96a3a6828b6401265646f31e621c9ef69adaa/LABO03~1.sql**](https://github.com/BryanSMA/LAB03_MUNOZ_BRYAN/blob/0da96a3a6828b6401265646f31e621c9ef69adaa/LABO03~1.sql)