

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



“LABORATORIO 02”

ASIGNATURA

Administración de Base de Datos

DOCENTE

Chávez Soto, Jorge Luis

ESTUDIANTE

Carhuaricra Anco, Heidy Nicole - 23200150

Lima, Perú

2025

ÍNDICE

LABORATORIO 02.....	3
I. OBJETIVOS	3
II. ENUNCIADO	3
III. ACTIVIDADES QUE REALIZAR.....	4

LABORATORIO 02

I. OBJETIVOS

El presente laboratorio tiene por objetivos:

- Implementar las sentencias SQL DDL para solucionar un problema de diseño de base de datos.
- Crear los objetos de base de datos en base al tipo de dato gestionado.
- Implementar las sentencias SQL DML para solucionar cargar los datos en forma masiva.

II. ENUNCIADO

Se desea tener una base de datos que almacene la información sobre los empleados de una empresa, los departamentos en los que trabajan y los estudios de que disponen. Guardaremos el historial laboral y salarial de todos los empleados. Para ello contamos con las siguientes tablas:

Empleados

Column Name	Data Type
DNI	NUMBER (8)
NOMBRE	VARCHAR (10)
APELLIDO1	VARCHAR (15)
APELLIDO2	VARCHAR (15)
DIRECC1	VARCHAR (25)
DIRECC2	VARCHAR (20)
CIUDAD	VARCHAR (20)
PROVINCIA	VARCHAR (20)
COD_POSTAL	VARCHAR (5)
SEXO	VARCHAR (1)
FECHA_NAC	DATE

Departamentos

Column Name	Data Type
DPTO_COD	NUMBER (5)
NOMBRE_DPTO	VARCHAR (30)
DPTO_PADRE	NUMBER (5)
PRESUPUESTO	NUMBER
PRES_ACTUAL	NUMBER

Estudios

Column Name	Data Type
EMPLEADO_DNI	NUMBER (8)
UNIVERSIDAD	NUMBER (5)
AÑO	NUMBER
GRADO	VARCHAR (3)
ESPECIALIDAD	VARCHAR (20)

Universidades

Column Name	Data Type
UNIV_COD	NUMBER(5)
NOMBRE_UNIV	VARCHAR(5)
CIUDAD	VARCHAR(20)
MUNICIPIO	VARCHAR(2)
COD_POSTAL	VARCHAR(5)

Historial_Laboral

Column Name	Data Type
EMPLEADO_DNI	NUMBER(8)
TRABAJO_COD	NUMBER(5)
FECHA_INICIO	DATE
FECHA_FIN	DATE
DPTO_COD	NUMBER(5)
SUPERVISOR_DNI	NUMBER(8)

Historial_Salarial

Column Name	Data Type
EMPLEADO_DNI	NUMBER(8)
SALARIO	NUMBER
FECHA_COMIENZO	DATE
FECHA_FIN	DATE

Trabajos

Column Name	Data Type
TRABAJO_COD	NUMBER(5)
NOMBRE_TRAB	VARCHAR(20)
SALARIO_MIN	NUMBER(2)
SALARIO_MAX	NUMBER(2)

III.ACTIVIDADES QUE REALIZAR

- Asignamos los tablespaces permanentes y temporales. Asimismo, otorgamos permisos al administrador SYSTEM para crear el usuario que la llamaremos *empresa*.

```
---Creación de tablespace
CREATE TABLESPACE empresa_tbs
  DATAFILE 'empresa_tbs.dbf'
  SIZE 20M AUTOEXTEND ON NEXT 5M
  MAXSIZE 500M EXTENT MANAGEMENT LOCAL
  SEGMENT SPACE MANAGEMENT AUTO;

---Creación de tablespace temporal
CREATE TEMPORARY TABLESPACE empresa_temp
  TEMPFILE 'empresa_temp.dbf'
  SIZE 20M AUTOEXTEND ON NEXT 5M
```

```

MAXSIZE 500M EXTENT MANAGEMENT LOCAL;

---Creación de usuario CICLISMO
CREATE USER empresa IDENTIFIED BY empresa
      DEFAULT TABLESPACE empresa_tbs
      TEMPORARY TABLESPACE empresa_temp
      QUOTA UNLIMITED ON empresa_tbs;

---Otorgar permisos al usuario
GRANT CONNECT, RESOURCE TO empresa;

```

```

TABLESPACE EMPRESA_TBS creado.

TABLESPACE EMPRESA_TEMP creado.

User EMPRESA creado.

Grant correcto.

```

- Ahora, procedemos a realizar los scripts de creación a partir de las tablas planteadas.
 1. Los siguientes atributos son obligatorios:
 - ✓ NOMBRE (en todas las tablas)
 - ✓ APELLIDO1 en EMPLEADOS
 - ✓ PRESUPUESTO en DEPARTAMENTOS
 - ✓ SALARIO en HISTORIAL_SALARIAL
 - ✓ SALARIO_MIN y SALARIO_MAX en TRABAJOS.

Al crear las tablas correspondientes especificar la opción NOT NULL. Si la tabla ya estuviese creada: ALTER TABLE nombre_tabla ADD nombre_campo TIPO NOT NULL;

```

---Scripts de Creación de Tablas
CREATE TABLE EMPLEADOS (
      DNI                NUMBER(8) PRIMARY KEY,
      NOMBRE              VARCHAR2(10) NOT NULL,
      APELLIDO1           VARCHAR2(15) NOT NULL,
      APELLIDO2           VARCHAR2(15),
      DIRECC1             VARCHAR2(25),
      DIRECC2             VARCHAR2(20),
      CIUDAD              VARCHAR2(20),
      PROVINCIA           VARCHAR2(20),
      COD_POSTAL          VARCHAR2(5),
      SEXO                VARCHAR2(1),
      FECHA_NAC           DATE
);

```

```

CREATE TABLE DEPARTAMENTOS (
    DPTO_COD          NUMBER(5) PRIMARY KEY,
    NOMBRE_DPTO       VARCHAR2(10) NOT NULL,
    DPTO_PADRE        NUMBER(5),
    PRESUPUESTO        NUMBER NOT NULL,
    PRES_ACTUAL        NUMBER
);

CREATE TABLE ESTUDIOS (
    EMPLEADO_DNI       NUMBER(8),
    UNIVERSIDAD        NUMBER(5),
    ANIO               NUMBER,
    GRADO              VARCHAR2(3),
    ESPECIALIDAD       VARCHAR2(20)
);

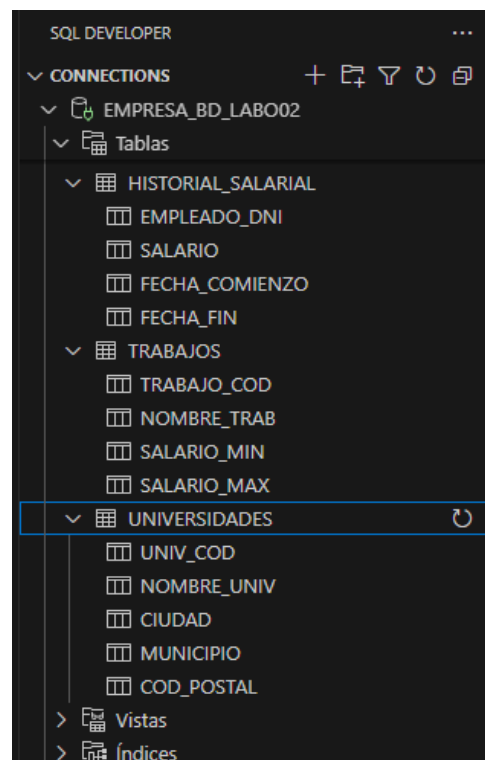
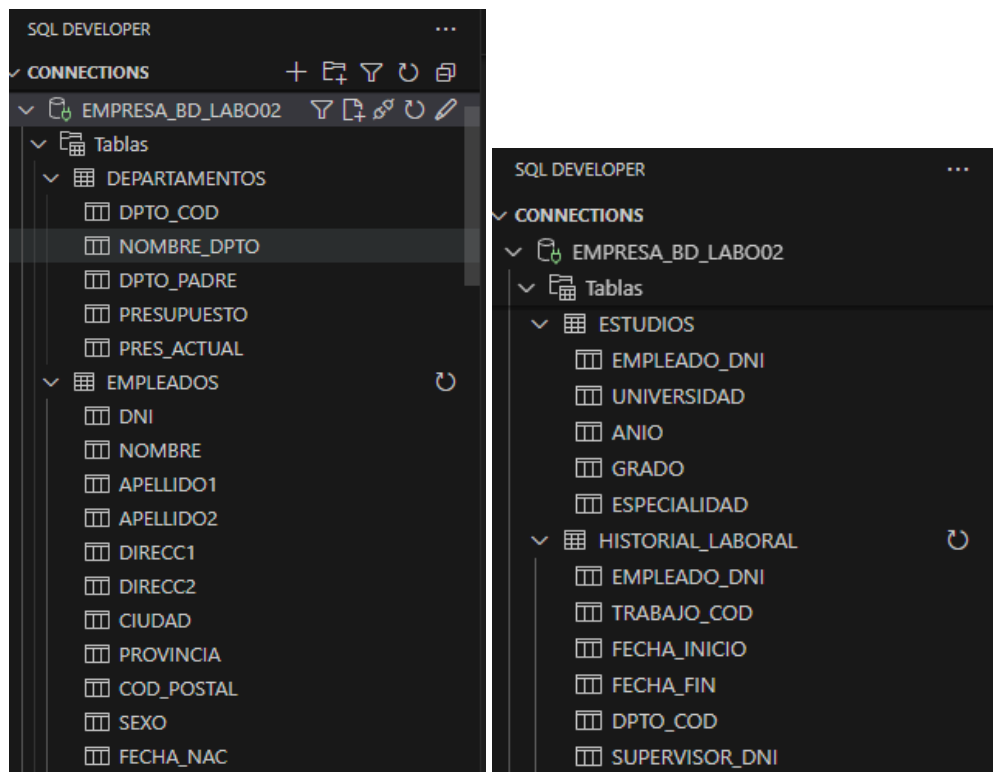
CREATE TABLE UNIVERSIDADES (
    UNIV_COD          NUMBER(5) PRIMARY KEY,
    NOMBRE_UNIV       VARCHAR2(5) NOT NULL,
    CIUDAD            VARCHAR2(20),
    MUNICIPIO         VARCHAR2(2),
    COD_POSTAL        VARCHAR2(5)
);

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)
);

CREATE TABLE HISTORIAL_SALARIAL (
    EMPLEADO_DNI       NUMBER(8),
    SALARIO            NUMBER NOT NULL,
    FECHA_COMIENZO     DATE,
    FECHA_FIN          DATE
);

CREATE TABLE TRABAJOS (
    TRABAJO_COD        NUMBER(5) PRIMARY KEY,
    NOMBRE_TRAB        VARCHAR2(20) NOT NULL,
    SALARIO_MIN        NUMBER(2) NOT NULL,
    SALARIO_MAX        NUMBER(2) NOT NULL
);

```



2. El atributo SEXO en EMPLEADOS sólo puede tomar los valores H para hombre y M para mujer.

```
ALTER TABLE EMPLEADOS
ADD CONSTRAINT check_sexo
```

```
CHECK (SEXO IN ('H', 'M'));
```

EMPRESA_BD_LABO02 > object > EMPRESA > TABLE > EMPLEADOS

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
→ 1	CHECK_SEXO	Check	SEXO IN ('H', 'M')	(null)	(null)	(null)
2	SYS_C008297	Check	"NOMBRE" IS NOT NULL	(null)	(null)	(null)
3	SYS_C008298	Check	"APELLIDO1" IS NOT NULL	(null)	(null)	(null)
4	SYS_C008299	Check	"FECHA_NAC" IS NOT NULL	(null)	(null)	(null)
5	SYS_C008300	Primary_Key	(null)	(null)	(null)	(null)

3. Dos DEPARTAMENTOS no se llaman igual. Dos TRABAJOS tampoco.

```
ALTER TABLE DEPARTAMENTOS
ADD CONSTRAINT unique_nombre_dpto
UNIQUE (NOMBRE_DPTO);
```

```
ALTER TABLE TRABAJOS
ADD CONSTRAINT unique_nombre_trab
UNIQUE (NOMBRE_TRAB);
```

EMPRESA_BD_LABO02 > object > EMPRESA > TABLE > DEPARTAMENTOS

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	SYS_C008301	Check	"NOMBRE_DPTO" IS NOT NULL	(null)	(null)	(null)
2	SYS_C008302	Check	"PRESUPUESTO" IS NOT NULL	(null)	(null)	(null)
3	SYS_C008303	Primary_Key	(null)	(null)	(null)	(null)
→ 4	UNIQUE_NOMBRE_DPTO	Unique	(null)	(null)	(null)	(null)

EMPRESA_BD_LABO02 > object > EMPRESA > TABLE > TRABAJOS

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	SYS_C008307	Check	"NOMBRE_TRAB" IS NOT NULL	(null)	(null)	(null)
2	SYS_C008308	Check	"SALARIO_MIN" IS NOT NULL	(null)	(null)	(null)
3	SYS_C008309	Check	"SALARIO_MAX" IS NOT NULL	(null)	(null)	(null)
4	SYS_C008310	Primary_Key	(null)	(null)	(null)	(null)
→ 5	UNIQUE_NOMBRE_TRAB	Unique	(null)	(null)	(null)	(null)

4. Cada empleado tiene un solo salario en cada momento. También, cada empleado tendrá asignado un solo trabajo en cada momento.

```
ALTER TABLE HISTORIAL_SALARIAL
ADD CONSTRAINT unique_historial_salarial
UNIQUE (EMPLEADO_DNI, FECHA_COMIENZO);
```

```
ALTER TABLE HISTORIAL_LABORAL
ADD CONSTRAINT unique_historial_laboral
UNIQUE (EMPLEADO_DNI, FECHA_INICIO);
```


EMPRESA_BD_LABO02 > object > EMPRESA > TABLE > HISTORIAL_SALARIAL						
Columns Data Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL						
	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	SYS_C008306	Check	"SALARIO" IS NOT NULL	(null)	(null)	(null)
2	UNIQUE_HISTORIAL_SALARIAL	Unique	(null)	(null)	(null)	(null)

EMPRESA_BD_LABO02 > object > EMPRESA > TABLE > HISTORIAL_LABORAL						
Columns Data Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL						
	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	UNIQUE_HISTORIAL_LABORAL	Unique	(null)	(null)	(null)	(null)

- Se ha de mantener la regla de integridad de referencia y pensar una clave primaria para cada tabla.

```

---Primary Key compuesta
ALTER TABLE ESTUDIOS
    ADD CONSTRAINT pk_estudios
    PRIMARY KEY (EMPLEADO_DNI) ;

---Foreign Key
ALTER TABLE ESTUDIOS
    ADD CONSTRAINT fk_estudios_empleados
    FOREIGN KEY (EMPLEADO_DNI)
    REFERENCES EMPLEADOS (DNI) ;

ALTER TABLE ESTUDIOS
    ADD CONSTRAINT fk_estudios_universidades
    FOREIGN KEY (UNIVERSIDAD)
    REFERENCES UNIVERSIDADES (UNIV_COD) ;

ALTER TABLE HISTORIAL_LABORAL
    ADD CONSTRAINT fk_historial_laboral_empleados
    FOREIGN KEY (EMPLEADO_DNI)
    REFERENCES EMPLEADOS (DNI) ;

ALTER TABLE HISTORIAL_LABORAL
    ADD CONSTRAINT fk_historial_laboral_trabajos
    FOREIGN KEY (TRABAJO_COD)
    REFERENCES TRABAJOS (TRABAJO_COD) ;

ALTER TABLE HISTORIAL_LABORAL
    ADD CONSTRAINT fk_historial_laboral_departamentos
    FOREIGN KEY (DPTO_COD)
    REFERENCES DEPARTAMENTOS (DPTO_COD) ;

ALTER TABLE HISTORIAL_LABORAL
    ADD CONSTRAINT fk_historial_laboral_empleados_supervisor
    FOREIGN KEY (SUPERVISOR_DNI)
    REFERENCES EMPLEADOS (DNI) ;

```

```
ALTER TABLE HISTORIAL_SALARIAL
ADD CONSTRAINT fk_historial_salarial_empleados
FOREIGN KEY (EMPLEADO_DNI)
REFERENCES EMPLEADOS (DNI) ;
```

```
ALTER TABLE DEPARTAMENTOS
ADD CONSTRAINT fk_departamentos_departamento_padre
FOREIGN KEY (DPTO_PADRE)
REFERENCES DEPARTAMENTOS (DPTO_COD) ;
```

EMPRESA_BD_LAB002 > object > EMPRESA > TABLE > HISTORIAL_LABORAL

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
1	FK_HISTORIAL_LABORAL_DEPARTAMENTOS	Foreign_Key	(null)	EMPRESA	DEPARTAMENTOS	SYS_C008303	NO ACTION	ENABLED
2	FK_HISTORIAL_LABORAL_EMPELADOS	Foreign_Key	(null)	EMPRESA	EMPELADOS	SYS_C008300	NO ACTION	ENABLED
3	FK_HISTORIAL_LABORAL_EMPELADOS_SUPERVISOR	Foreign_Key	(null)	EMPRESA	EMPELADOS	SYS_C008300	NO ACTION	ENABLED
4	FK_HISTORIAL_LABORAL TRABAJOS	Foreign_Key	(null)	EMPRESA	TRABAJOS	SYS_C008310	NO ACTION	ENABLED
5	UNIQUE_HISTORIAL_LABORAL	Unique	(null)	(null)	(null)	(null)	(null)	ENABLED

EMPRESA_BD_LAB002 > object > EMPRESA > TABLE > HISTORIAL_SALARIAL

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
1	FK_HISTORIAL_SALARIAL_EMPELADOS	Foreign_Key	(null)	EMPRESA	EMPELADOS	SYS_C008300	NO ACTION	ENABLED
2	SYS_C008306	Check	"SALARIO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
3	UNIQUE_HISTORIAL_SALARIAL	Unique	(null)	(null)	(null)	(null)	(null)	ENABLED

EMPRESA_BD_LAB002 > object > EMPRESA > TABLE > ESTUDIOS

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE
1	FK_ESTUDIOS_EMPELADOS	Foreign_Key	(null)	EMPRESA	EMPELADOS	SYS_C008300	NO ACTION	ENABLED	NOT DEFERRABLE
2	SYS_C008305	Check	(null)	EMPRESA	UNIVERSIDADES	SYS_C008305	NO ACTION	ENABLED	NOT DEFERRABLE
3	PK_ESTUDIOS	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

EMPRESA_BD_LAB002 > object > EMPRESA > TABLE > DEPARTAMENTOS

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
1	FK_DEPARTAMENTOS_DEPARTAMENTO_PADRE	Foreign_Key	(null)	EMPRESA	DEPARTAMENTOS	SYS_C008303	NO ACTION	ENABLED
2	SYS_C008301	Check	"NOMBRE_DPTO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
3	SYS_C008302	Check	"PRESUPUESTO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
4	SYS_C008303	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED
5	UNIQUE_NOMBRE_DPTO	Unique	(null)	(null)	(null)	(null)	(null)	ENABLED

- Agregue a la tabla empleados los campos de teléfono y celular para tener como ubicar rápidamente al empleado.

```
ALTER TABLE EMPLEADOS
ADD TELEFONO VARCHAR2(15) ;
```

```
ALTER TABLE EMPLEADOS
ADD CELULAR NUMBER(9) ;
```

EMPRESA_BD_LAB002 > object > EMPRESA > TABLE > EMPLEADOS

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DNI	NUMBER(8,0)	No	(null)	1	(null)
2	NOMBRE	VARCHAR2(10 BYTE)	No	(null)	2	(null)
3	APELLIDO1	VARCHAR2(15 BYTE)	No	(null)	3	(null)
4	APELLIDO2	VARCHAR2(15 BYTE)	Yes	(null)	4	(null)
5	DIRECC1	VARCHAR2(25 BYTE)	Yes	(null)	5	(null)
6	DIRECC2	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)
7	CIUDAD	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)
8	PROVINCIA	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)
9	COD_POSTAL	VARCHAR2(5 BYTE)	Yes	(null)	9	(null)
10	SEXO	VARCHAR2(1 BYTE)	Yes	(null)	10	(null)
11	FECHA_NAC	DATE	No	(null)	11	(null)
12	TELEFONO	VARCHAR2(15 BYTE)	Yes	(null)	12	(null)
13	CELULAR	NUMBER(9,0)	Yes	(null)	13	(null)

7. Inserte las siguientes filas (las columnas que no aparecen se considerarán nulas).

Empleados				
NOMBRES	APELLIDO1	APELLIDO2	DNI	SEXO
Sergio	Palma	Entrena	111222	H
Lucia	Ortega	Plus	222333	M

Historial Laboral					
EMPLEADO DNI	TRABAJO COD	FECHA_ INICIO	FECHA FIN	DPTO_ COD	SUPERVISOR_ DNI
111222		16/06/96		222333	

```

---Inserción de datos de la tabla Empleados
INSERT INTO EMPLEADOS (NOMBRE, APELLIDO1, APELLIDO2, DNI, SEXO)
VALUES ('Sergio', 'Palma', 'Entrena', 111222, 'H');
INSERT INTO EMPLEADOS (NOMBRE, APELLIDO1, APELLIDO2, DNI, SEXO)
VALUES ('Lucia', 'Ortega', 'Plus', 222333, 'M');

---Inserción de datos de la tabla Historial_Laboral,
previamente con una inserción de datos de la tabla
Departamentos porque el atributo DPTO_COD es una PK
INSERT INTO DEPARTAMENTOS (DPTO_COD, NOMBRE_DPTO, PRESUPUESTO)
VALUES (22233, 'PRUEBA', 10000);
INSERT INTO HISTORIAL_LABORAL (EMPLEADO_DNI, TRABAJO_COD,
FECHA_INICIO, FECHA_FIN, DPTO_COD, SUPERVISOR_DNI)
VALUES (111222, NULL, TO_DATE('16/06/96', 'DD/MM/YY'),
NULL, 22233, NULL);

```

DNI	NOMBRE	APELLIDO1	APELLIDO2	DIRECC1	DIRECC2	CIUDAD	PROVINCIA	COD_POSTAL	SEXO	FECHA_NAC	TELEFONO	CELULA
1	111222	Sergio	Palma	Entrena	(null)	(null)	(null)	(null)	H	(null)	(null)	(n
2	222333	Lucia	Ortega	Plus	(null)	(null)	(null)	(null)	M	(null)	(null)	(n

DPTO_COD	NOMBRE_DPTO	DPTO_PADRE	PRESUPUESTO	PRES_ACTUAL
1	22233	PRUEBA	(null)	10000
				(null)

EMPLEADO_DNI	TRABAJO_COD	FECHA_INICIO	FECHA_FIN	DPTO_COD	SUPERVISOR_DNI
1	111222	(null)	16/06/96	(null)	22233
					(null)

8. ¿Qué ocurre si se modifica esta última fila de historial_laboral asignándole al empleado 111222 un supervisor que no existe en la tabla de empleados?

---Como ya se ejecutó el script de inserción, intentamos actualizar la tupla con un SUPERVISOR_DNI inexistente en la tabla Empleados.

```
UPDATE HISTORIAL_LABORAL
  SET SUPERVISOR_DNI = 999999
  WHERE EMPLEADO_DNI = 111222;
```

---Visualizamos la terminal

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  QUERY RESULT  SCRIPT OUTPUT  SQL HISTORY  TASK MONITOR  PORTS

Error que empieza en la línea: 1 del comando :
UPDATE HISTORIAL_LABORAL
  SET SUPERVISOR_DNI = 999999
  WHERE EMPLEADO_DNI = 111222
Informe de error -
ORA-02291: integrity constraint (EMPRESA.FK_HISTORIAL_LABORAL_EMPLEADOS_SUPERVISOR) violated - parent key not found

https://docs.oracle.com/error-help/db/ora-02291/

More Details :
https://docs.oracle.com/error-help/db/ora-02291/
```

El error nos indica:

ORA-02291: Restricción de integridad (*string.string*) violada - clave principal no encontrada.

Causa: Un valor de clave externa no tiene un valor de clave principal coincidente.

Acción: Eliminar la clave externa o agregar una clave principal coincidente.

9. Borre una universidad de la tabla de UNIVERSIDADES ¿Qué le sucede a la restricción de clave ajena de la tabla ESTUDIOS? Altere la definición de la tabla para que se mantenga la restricción, aunque se borre una universidad.

---Modificamos nuestro Foreign Key incluyendo la restricción solicitada.

```
ALTER TABLE ESTUDIOS
  DROP CONSTRAINT fk_estudios_universidades;
```

```
ALTER TABLE ESTUDIOS
ADD CONSTRAINT fk_estudios_universidades
FOREIGN KEY (UNIVERSIDAD)
REFERENCES UNIVERSIDADES (UNIV_COD)
ON DELETE SET NULL;
```

EMPRESA_BD_LABO02 > object > EMPRESA > TABLE > ESTUDIOS

Columns	Data	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE		
1	FK_ESTUDIOS_EMPLEADOS	Foreign_Key	(null)	EMPRESA	EMPLEADOS	SYS_C008300	NO ACTION	ENABLED	NOT DEFERRABLE		
2	FK_ESTUDIOS_UNIVERSIDADES	Foreign_Key	(null)	EMPRESA	UNIVERSIDADES	SYS_C008305	SET NULL	ENABLED	NOT DEFERRABLE		
3	PK_ESTUDIOS	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE		

```
---Insertamos datos a las tablas
INSERT INTO UNIVERSIDADES (UNIV_COD, NOMBRE_UNIV, CIUDAD,
MUNICIPIO, COD_POSTAL)
VALUES (100, 'UNMSM', 'Lima', '01', '15001');
INSERT INTO UNIVERSIDADES (UNIV_COD, NOMBRE_UNIV, CIUDAD,
MUNICIPIO, COD_POSTAL)
VALUES (200, 'UNI', 'Lima', '01', '15002');
INSERT INTO ESTUDIOS (EMPLEADO_DNI, UNIVERSIDAD, ANIO, GRADO,
ESPECIALIDAD)
VALUES (111222, 100, 2015, 'LIC', 'Sistemas');
INSERT INTO ESTUDIOS (EMPLEADO_DNI, UNIVERSIDAD, ANIO, GRADO,
ESPECIALIDAD)
VALUES (222333, 200, 2018, 'MTR', 'Industrial');
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

All rows fetched: 2 in 0.127 seconds

	UNIV_COD	NOMBRE_UNIV	CIUDAD	MUNICIPIO	COD_POSTAL
1	100	UNMSM	Lima	01	15001
2	200	UNI	Lima	01	15002

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

All rows fetched: 2 in 0.107 seconds

	EMPLEADO_DNI	UNIVERSIDAD	ANIO	GRADO	ESPECIALIDAD
1	111222	100	2015	LIC	Sistemas
2	222333	200	2018	MTR	Industrial

```
---Eliminamos una universidad
DELETE FROM UNIVERSIDADES
WHERE UNIV_COD = 100;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

All rows fetched: 2 in 0.171 seconds

	EMPLEADO_DNI	UNIVERSIDAD	ANIO	GRADO	ESPECIALIDAD
1	111222	(null)	2015	LIC	Sistemas
2	222333	200	2018	MTR	Industrial

10. Añada una restricción que obligue a que las personas que hayan introducido la CIUDAD deban tener el campo COD_POSTAL a NOT NULL. ¿Qué ocurre con las filas ya introducidas?

```
ALTER TABLE EMPLEADOS
ADD CONSTRAINT check_ciudad_codpostal
CHECK (CIUDAD IS NULL OR COD_POSTAL IS NOT NULL);
```

```

---Intentamos añadir el atributo ciudad a un empleado
UPDATE EMPLEADOS
  SET CIUDAD = 'Lima'
  WHERE DNI = 111222;

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  QUERY RESULT  SCRIPT OUTPUT  SQL H
Error que empieza en la línea: 1 del comando :
UPDATE EMPLEADOS
  SET CIUDAD = 'Lima'
  WHERE DNI = '111222'
Informe de error -
ORA-02290: check constraint (EMPRESA.CHECK_CIUADAD_CODPOSTAL) violated

https://docs.oracle.com/error-help/db/ora-02290/

More Details :
https://docs.oracle.com/error-help/db/ora-02290/

```

El error nos indica:

ORA-02290: Restricción de comprobación (string. string) violada
Causa: Los valores que se insertan no satisfacen la restricción de verificación nombrada.
Acción: No inserte valores que violen la restricción.

```

UPDATE EMPLEADOS
  SET CIUDAD = 'Lima', COD_POSTAL = '15001'
  WHERE DNI = 111222;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

All rows fetched: 2 in 0.114 seconds

	DNI	NOMBRE	APELLIDO1	APELLIDO2	DIRECC1	DIRECC2	CIUDAD	PROVINCIA	COD_POSTAL	SEXO	FECHA_NAC	TELEFONO	CELULA
1	111222	Sergio	Palma	Entrena	(null)	(null)	Lima	(null)	15001	H	(null)	(null)	(n
2	222333	Lucia	Ortega	Plus	(null)	(null)	(null)	(null)	(null)	M	(null)	(null)	(n

```

UPDATE EMPLEADOS
  SET COD_POSTAL = '15002'
  WHERE DNI = 222333;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

All rows fetched: 2 in 0.094 seconds

	DNI	NOMBRE	APELLIDO1	APELLIDO2	DIRECC1	DIRECC2	CIUDAD	PROVINCIA	COD_POSTAL	SEXO	FECHA_NAC	TELEFONO	CELULA
1	111222	Sergio	Palma	Entrena	(null)	(null)	Lima	(null)	15001	H	(null)	(null)	(n
2	222333	Lucia	Ortega	Plus	(null)	(null)	(null)	(null)	15002	M	(null)	(null)	(n

- Añada un nuevo atributo VALORACIÓN en la tabla de EMPLEADOS que indique de 1 a 10 la valoración que obtuvo el empleado en su entrevista de trabajo al iniciar su andadura en la empresa. Ponga el valor por defecto 5 para ese campo.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

All rows fetched: 2 in 0.094 seconds

	DNI	NOMBRE	APELLIDO1	APELLIDO2	DIRECC1	DIRECC2	CIUDAD	PROVINCIA	COD_POSTAL	SEXO	FECHA_NAC	TELEFONO	CELULA
1	111222	Sergio	Palma	Entrena	(null)	(null)	Lima	(null)	15001	H	(null)	(null)	(n
2	222333	Lucia	Ortega	Plus	(null)	(null)	(null)	(null)	15002	M	(null)	(null)	(n

```

--Como en las tuplas no se tenía este atributo, al momento de
ejecutarlo se agregará con el valor por defecto 5
ALTER TABLE EMPLEADOS
    ADD VALORACION NUMBER(2)
    DEFAULT 5
    CONSTRAINT check_valoracion
    CHECK (VALORACION BETWEEN 1 AND 10);

```

	APELLIDO1	APELLIDO2	DIRECC1	DIRECC2	CIUDAD	PROVINCIA	COD_POSTAL	SEXO	FECHA_NAC	TELEFONO	CELULAR	VALORACION
1	Palma	Entrena	(null)	(null)	Lima	(null)	15001	M	(null)	(null)	(null)	5
2	Ortega	Plus	(null)	(null)	(null)	(null)	15002	M	(null)	(null)	(null)	5

12. Elimine la restricción de que el atributo NOMBRE de la tabla EMPLEADOS no puede ser nulo.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STI
1	CHECK_CIUDAD_CODPOSTAL	Check	CIUDAD IS NULL OR COD_POSTAL IS NOT NULL	(null)	(null)	(null)	(null)	EN
2	CHECK_SEXO	Check	SEXO IN ('M', 'H')	(null)	(null)	(null)	(null)	EN
3	CHECK_VALORACION	Check	VALORACION BETWEEN 1 AND 10	(null)	(null)	(null)	(null)	EN
4	SYS_C008297	Check	"NOMBRE" IS NOT NULL	(null)	(null)	(null)	(null)	EN
5	SYS_C008298	Check	"APELLIDO1" IS NOT NULL	(null)	(null)	(null)	(null)	EN
6	SYS_C008300	Primary_Key	(null)	(null)	(null)	(null)	(null)	EN

```

--Eliminamos el constraint de NOT NULL del atributo NOMBRE
ALTER TABLE EMPLEADOS
    MODIFY NOMBRE NULL;

```

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STI
1	CHECK_CIUDAD_CODPOSTAL	Check	CIUDAD IS NULL OR COD_POSTAL IS NOT NULL	(null)	(null)	(null)	(null)	EN
2	CHECK_SEXO	Check	SEXO IN ('M', 'H')	(null)	(null)	(null)	(null)	EN
3	CHECK_VALORACION	Check	VALORACION BETWEEN 1 AND 10	(null)	(null)	(null)	(null)	EN
4	SYS_C008298	Check	"APELLIDO1" IS NOT NULL	(null)	(null)	(null)	(null)	EN
5	SYS_C008300	Primary_Key	(null)	(null)	(null)	(null)	(null)	EN

13. Modificar el tipo de datos de DIRECC1 de la tabla EMPLEADOS a cadena de caracteres de 40 como máximo.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DNI	NUMBER(8,0)	No	(null)	1	(null)
2	NOMBRE	VARCHAR2(10 BYTE)	Yes	(null)	2	(null)
3	APELLIDO1	VARCHAR2(15 BYTE)	No	(null)	3	(null)
4	APELLIDO2	VARCHAR2(15 BYTE)	Yes	(null)	4	(null)
5	DIRECC1	VARCHAR2(25 BYTE)	Yes	(null)	5	(null)
6	DIRECC2	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)
7	CIUDAD	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)
8	PROVINCIA	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)
9	COD_POSTAL	VARCHAR2(5 BYTE)	Yes	(null)	9	(null)
10	SEXO	VARCHAR2(1 BYTE)	Yes	(null)	10	(null)
11	FECHA_NAC	DATE	Yes	(null)	11	(null)

```

--Modificamos el tipo de dato de DIRECC1: VARCHAR2(25) ->
VARCHAR2(40)
ALTER TABLE EMPLEADOS
    MODIFY DIRECC1 VARCHAR2(40);

```

EMPRESA_BD_LAB002 > object > EMPRESA > TABLE > EMPLEADOS

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DNI	NUMBER(8,0)	No	(null)	1	(null)
2	NOMBRE	VARCHAR2(10 BYTE)	Yes	(null)	2	(null)
3	APELLIDO1	VARCHAR2(15 BYTE)	No	(null)	3	(null)
4	APELLIDO2	VARCHAR2(15 BYTE)	Yes	(null)	4	(null)
5	DIRECC1	VARCHAR2(40 BYTE)	Yes	(null)	5	(null)
6	DIRECC2	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)
7	CIUDAD	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)
8	PROVINCIA	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)
9	COD_POSTAL	VARCHAR2(5 BYTE)	Yes	(null)	9	(null)
10	SEXO	VARCHAR2(1 BYTE)	Yes	(null)	10	(null)
11	FECHA_NAC	DATE	Yes	(null)	11	(null)

14. ¿Podría modificar el tipo de datos del atributo FECHA_NAC de la tabla EMPLEADOS y convertirla a tipo cadena?

---En caso no se tenga datos en las tablas, se podría ejecutar este script

```
ALTER TABLE EMPLEADOS
    MODIFY FECHA_NAC VARCHAR2(10);
```

---En caso ya tengamos datos almacenados. Seguir los pasos.

---1. Crear una nueva columna

```
ALTER TABLE EMPLEADOS
    ADD FECHA_NAC_STR VARCHAR2(10);
```

---2. Transformamos el tipo de dato DATE a una cadena con formato DD/MM/YY y copiamos a la nueva columna

```
UPDATE EMPLEADOS
    SET FECHA_NAC_STR = TO_CHAR(FECHA_NAC, 'DD/MM/YY');
```

---3. Eliminamos la columna con tipo de dato DATE

```
ALTER TABLE EMPLEADOS
    DROP COLUMN FECHA_NAC;
```

---4. Renombramos la nueva columna para evitar confusiones

```
ALTER TABLE EMPLEADOS
    RENAME COLUMN FECHA_NAC_STR TO FECHA_NAC;
```

15. Cambiar la clave primaria de EMPLEADOS al NOMBRE y los dos APELLIDOS.

---Buscamos el nombre del constraint

```
SELECT CONSTRAINT_NAME
    FROM USER_CONSTRAINTS
    WHERE TABLE_NAME = 'EMPLEADOS' AND CONSTRAINT_TYPE = 'P';
```


All rows fetched: 1 in 6.945 seconds

	CONSTRAINT_NAME
1	SYS_C008300

```

---Eliminamos los foreign keys relacionadas al pk
ALTER TABLE ESTUDIOS
    DROP CONSTRAINT fk_estudios_empleados;
ALTER TABLE HISTORIAL_LABORAL
    DROP CONSTRAINT fk_historial_laboral_empleados;
ALTER TABLE HISTORIAL_LABORAL
    DROP CONSTRAINT fk_historial_laboral_empleados_supervisor;
ALTER TABLE HISTORIAL_SALARIAL
    DROP CONSTRAINT fk_historial_salarial_empleados;

---Eliminamos el constraint de dni (pk)
ALTER TABLE EMPLEADOS
    DROP CONSTRAINT SYS_C008300;

```

```

---Crear la PK compuesto
ALTER TABLE EMPLEADOS
    ADD CONSTRAINT pk_empleados_nombre_apellido1_apellido2
    PRIMARY KEY (NOMBRE, APELLIDO1, APELLIDO2);

```

EMPRESA_BD_LABO02 > object > EMPRESA > TABLE > EMPLEADOS

Columns	Data	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
		CONSTRAINT_NAME		CONSTRAINT_TYPE		SEARCH_CONDITION		R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DEF
1	CHECK_CIUAD_CDOPOSTAL	Check				CIUDAD IS NULL OR COD_POSTAL IS NOT NULL		(null)	(null)	(null)	(n
2	CHECK_SEXO	Check				SEXO IN ('H', 'M')		(null)	(null)	(null)	(n
3	CHECK_VALORACION	Check				VALORACION BETWEEN 1 AND 10		(null)	(null)	(null)	(n
4	PK_EMPLEADOS_NOMBRE_APELLIDO1_APELLIDO2	Primary_Key				(null)		(null)	(null)	(null)	(n
5	SYS_C008298	Check				"APELLIDO1" IS NOT NULL		(null)	(null)	(null)	(n

```

---Reescribir los foreign keys
ALTER TABLE ESTUDIOS
    ADD NOMBRE VARCHAR2(10);
ALTER TABLE ESTUDIOS
    ADD APELLIDO1 VARCHAR2(15);
ALTER TABLE ESTUDIOS
    ADD APELLIDO2 VARCHAR2(15);
ALTER TABLE HISTORIAL_LABORAL
    ADD NOMBRE VARCHAR2(10);
ALTER TABLE HISTORIAL_LABORAL
    ADD APELLIDO1 VARCHAR2(15);
ALTER TABLE HISTORIAL_LABORAL
    ADD APELLIDO2 VARCHAR2(15);
ALTER TABLE HISTORIAL_SALARIAL
    ADD NOMBRE VARCHAR2(10);
ALTER TABLE HISTORIAL_SALARIAL
    ADD APELLIDO1 VARCHAR2(15);
ALTER TABLE HISTORIAL_SALARIAL

```

```
ADD APELLIDO2 VARCHAR2 (15) ;
```

```
ALTER TABLE ESTUDIOS
ADD CONSTRAINT fk_estudios_empleados
FOREIGN KEY (NOMBRE, APELLIDO1, APELLIDO2)
REFERENCES EMPLEADOS (NOMBRE, APELLIDO1, APELLIDO2) ;
ALTER TABLE HISTORIAL_LABORAL
ADD CONSTRAINT fk_historial_laboral_empleados
FOREIGN KEY (NOMBRE, APELLIDO1, APELLIDO2)
REFERENCES EMPLEADOS (NOMBRE, APELLIDO1, APELLIDO2) ;
ALTER TABLE HISTORIAL_LABORAL
ADD CONSTRAINT fk_historial_laboral_empleados_supervisor
FOREIGN KEY (NOMBRE, APELLIDO1, APELLIDO2)
REFERENCES EMPLEADOS (NOMBRE, APELLIDO1, APELLIDO2) ;
ALTER TABLE HISTORIAL_SALARIAL
ADD CONSTRAINT fk_historial_salarial_empleados
FOREIGN KEY (NOMBRE, APELLIDO1, APELLIDO2)
REFERENCES EMPLEADOS (NOMBRE, APELLIDO1, APELLIDO2) ;
```

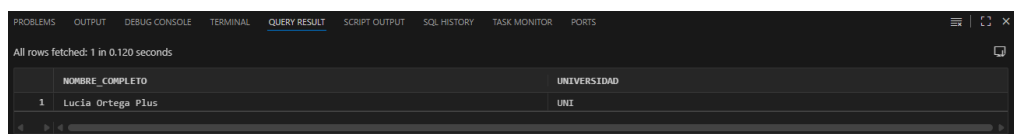
16. Crear una nueva tabla llamada INFORMACIÓN UNIVERSITARIA que tenga el NOMBRE y los dos APELLIDOS (en un solo atributo) de todos los EMPLEADOS junto con la UNIVERSIDAD donde estudiaron. Cárguela con los datos correspondientes.

---Creación de la tabla

```
CREATE TABLE INFORMACION_UNIVERSITARIA (
    NOMBRE_COMPLETO VARCHAR2 (100) ,
    UNIVERSIDAD VARCHAR2 (50)
) ;
```

---Configurar correctamente los atributos

```
INSERT INTO INFORMACION_UNIVERSITARIA (NOMBRE_COMPLETO,
UNIVERSIDAD)
SELECT
    e.NOMBRE || ' ' || e.APELLIDO1 || ' ' || e.APELLIDO2 AS
NOMBRE_COMPLETO,
    u.NOMBRE_UNIV
FROM EMPLEADOS e
JOIN ESTUDIOS es
    ON e.DNI = es.EMPLEADO_DNI
JOIN UNIVERSIDADES u
    ON es.UNIVERSIDAD = u.UNIV_COD ;
```



NOMBRE_COMPLETO		UNIVERSIDAD	
1	Lucia Ortega Plus	UNI	

17. Crear una vista llamada NOMBRE_EMPLEADOS con el NOMBRE y los dos APELLIDOS (en un solo atributo) de todos los EMPLEADOS que son de Málaga.

```

---Crear la vista
CREATE OR REPLACE VIEW NOMBRE_EMPLEADOS AS
SELECT
    NOMBRE || ' ' || APELLIDO1 || ' ' || APELLIDO2 AS
NOMBRE_COMPLETO
FROM EMPLEADOS
WHERE CIUDAD = 'Málaga';

```

EMPRESA_BO_LAB002 > object > EMPRESA > VIEW > NOMBRE_EMPLEADOS

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE
1	NOMBRE_COMPLETO	VARCHAR2(42)	Yes	(null)	1	(null)	NO	NO	NO

18. Crear otra vista llamada INFORMACION_EMPLEADOS con el NOMBRE y los dos APELLIDOS (en un solo atributo) y EDAD (no fecha de nacimiento) de todos los EMPLEADOS.

```

---Crear la vista
CREATE OR REPLACE VIEW INFORMACION_EMPLEADOS AS
SELECT
    NOMBRE || ' ' || APELLIDO1 || ' ' || APELLIDO2 AS
NOMBRE_COMPLETO,
    TRUNC(MONTHS_BETWEEN(SYSDATE, FECHA_NAC) / 12) AS EDAD
FROM EMPLEADOS;

```

```

---Insertar la fecha de nacimiento para poder visualizar al
momento de realizar la consulta respectiva
UPDATE EMPLEADOS
SET FECHA_NAC = '25/06/2006'
WHERE DNI = 111222;
UPDATE EMPLEADOS
SET FECHA_NAC = '21/08/1985'
WHERE DNI = 222333;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR PORTS

All rows fetched: 2 in 0.135 seconds

	NOMBRE_COMPLETO	EDAD
1	Sergio Palma Entrena	19
2	Lucia Ortega Plus	40

19. Crear otra vista sobre la anterior llamada INFORMACION_ACTUAL que dispone de toda la información de INFORMACION_EMPLEADOS junto con el SALARIO que está cobrando en este momento.

```

---Crear la vista
CREATE OR REPLACE VIEW INFORMACION_ACTUAL AS
SELECT
    ie.NOMBRE_COMPLETO,
    ie.EDAD,
    hs.SALARIO
FROM INFORMACION_EMPLEADOS ie
JOIN HISTORIAL_SALARIAL hs

```

```

ON ie.NOMBRE_COMPLETO = (
    SELECT e.NOMBRE || ' ' || e.APELLIDO1 || ' ' ||
e.APELLIDO2
    FROM EMPLEADOS e
    WHERE e.DNI = hs.EMPLEADO_DNI
)
WHERE hs.FECHA_FIN IS NULL;

```

EMPRESA_BD_LAB002 > object > EMPRESA > VIEW > INFORMACION_ACTUAL

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE
1	NOMBRE_COMPLETO	VARCHAR2(42)	Yes	(null)	1	(null)	NO	NO	NO
2	EDAD	NUMBER	Yes	(null)	2	(null)	NO	NO	NO
3	SALARIO	NUMBER	No	(null)	3	(null)	NO	NO	NO

20. Borrar todas las tablas. ¿Hay que tener en cuenta las claves ajenas a la hora de borrar las tablas?

Sí, en Oracle y cualquier SGBD relacional, no se puede borrar una tabla que esté referenciada por una clave foránea en otra tabla. Al intentar compilar, aparecería error.

ORA-02449: Claves únicas/primarias en la tabla referenciada por claves externas

Causa: Se intentó eliminar una tabla con claves únicas o primarias referenciadas por claves externas en otra tabla.

Acción: Antes de realizar las operaciones anteriores en la tabla, elimine las restricciones de clave foránea en otras tablas. Puede ver qué restricciones hacen referencia a una tabla ejecutando el siguiente comando: `SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME = "tabnam";`

En este caso que, se creó un nuevo PK compuesto para EMPLEADOS y no se llegó a insertar datos en las otras tablas que involucraban al FK, por lo tanto, no presentó restricción de borrar todas las tablas con el siguiente script.

```

DROP TABLE EMPLEADOS;
DROP TABLE HISTORIAL_SALARIAL;
DROP TABLE HISTORIAL_LABORAL;
DROP TABLE DEPARTAMENTOS;
DROP TABLE ESTUDIOS;
DROP TABLE UNIVERSIDADES;
DROP TABLE TRABAJOS;
DROP TABLE INFORMACION_UNIVERSITARIA;

```