

Instrucciones

1. Para resolver este taller deberá utilizar la herramienta Oracle Live[\[link\]](#)
2. La carpeta contiene un archivo que se llama DDL.sql, este archivo tiene definidas unas tablas, así como unos registros para cada una de ellas.

Integrantes:

CARLOS JARAMILLO

ANDREA TIMARAN

Actividades

Para las siguientes actividades:

SQL para modificar los datos

1. Incluya en el esquema la relación Dependent con las siguientes características
Dependent(Name, EmpNo, Gender, DOB, Allowance, Relationship)

En donde:

- Name es el nombre del dependiente, una cadena de tamaño variable de tamaño 30.
- EmpNo es el identificador del empleado que registra el dependiente.
- Gender es el género del dependiente. El género puede ser uno de los siguientes valores:
 - 'M', por Male,
 - 'F', por Female
 - 'NB', por Non binary
 - 'T', por Transgender
 - 'E', por Intersex
 - 'NS', por "I prefer not to say"
- DOB es la fecha de nacimiento del dependiente. La fecha de nacimiento es muy importante, por tanto no puede dejarse el campo vacío.
- Allowance es el monto del subsidio que recibe el dependiente, es un número de máximo 5 cifras con dos cifras decimales.
- Relationship es el parentesco entre el dependiente y el empleado. El parentesco puede ser uno de los siguientes: hijo, cónyuge, padre.

Pegue en el cuadro a continuación la definición de la nueva relación con sus atributos, llaves y restricciones

```
CREATE TABLE Dependent(  
    Name VARCHAR(30),  
    EmpNo INTEGER,  
    Gender VARCHAR(2) CHECK (Gender in ('M','F','NB','T','E','NS')),  
    DOB DATE NOT NULL,  
    Allowance NUMERIC(7,2),  
    Relationship VARCHAR(10) CHECK (Relationship in ('hijo','conyuge','padre')),  
    PRIMARY KEY (Name, EmpNo),  
    FOREIGN KEY (EmpNo) REFERENCES EMPLOYEE(EmpNo)  
);  
  
INSERT INTO Dependent VALUES ('Blake Junio',7698,'M',DATE '2003-09-28',285.10,'hijo');
```

```
INSERT INTO Dependent VALUES ('Queen King',7839,'F',DATE '1980-12-12',250.98,'conyuge');
INSERT INTO Dependent VALUES ('Jones 1',7566,'NS',DATE '1950-07-01',974.99,'padre');
INSERT INTO Dependent VALUES ('Clarkend',7782,'NB',DATE '2023-01-01',2000.00,'hijo');
```

2. Incluya en la tabla Employee los siguientes empleados:

- Parker, quien es subordinado de Blake en el departamento de 'RESEARCH'. Elija los valores para los demás campos, ningún campo debe ser vacío.
- Banner, quien es subordinado de CLARK en el departamento de 'OPERATIONS'. Elija los valores para los demás campos, ningún campo debe ser vacío.
- Stark, quien es subordinado de Jones en el departamento de 'SALES'. Elija los valores para los demás campos, ningún campo debe ser vacío.
- Strange, de quién no se tiene información del jefe o del departamento en que trabaja. Elija los valores para los demás campos.

Pegue en el cuadro a continuación los inserts

```
-----
INSERT INTO Employee VALUES (8001, 'PARKER', 'SALESMAN', 7698,DATE '1983-11-17', 1550,
0, 20);
INSERT INTO Employee VALUES (8105, 'BANNER', 'SALESMAN', 7782,DATE '1984-07-11', 2000,
0, 40);
INSERT INTO Employee VALUES (8527, 'STARK', 'SALESMAN', 7566,DATE '2000-01-28', 3000, 0,
30);
INSERT INTO Employee VALUES (8671, 'STRANGE', 'NOOBY', NULL,DATE '2023-08-10', 100, 0,
NULL);
```

SQL para manipular datos

Escriba el código SQL que permita resolver las siguientes consultas.

3. Consulta 1: Genere una tabla con dos columnas llamadas Manager y EmployeeId, EmployeeName y EmployeeLastname que en cada fila contiene los datos de cada empleado y el apellido del jefe ordenado alfabéticamente de acuerdo a la columna Manager

Código. Copie y pegue la sentencia SELECT-FROM-WHERE que resuelve la consulta.

```
SELECT MG.ENAME || ' ' || MG.EMPNO MANAGER, E.ENAME || ' ' || E.EMPNO EMPLOYEE FROM
EMPLOYEE E LEFT JOIN EMPLOYEE MG ON (E.MGR = MG.EMPNO) ORDER BY MANAGER ASC;
```

4. Consulta 2: Genere una tabla con dos columnas Manager y NumEmployee que contiene los apellidos de los jefes y la cantidad de empleados que tiene subordinados. La tabla está ordenada de menor a mayor de acuerdo a la cantidad de subordinados.

Código. Copie y pegue la sentencia SELECT-FROM-WHERE que resuelve la consulta.

```
SELECT MG.ENAME MANAGER, COUNT(E.EMPNO) NUMBER_EMPLOYEE FROM EMPLOYEE E LEFT JOIN
EMPLOYEE MG ON (E.MGR = MG.EMPNO) GROUP BY MG.ENAME ORDER BY NUMBER_EMPLOYEE ASC;
```

5. Consulta 3: Genere una tabla con el id, la fecha de inicio y la fecha de finalización de los proyectos que se llevan a cabo en los departamentos ubicados en 'NEW YORK'. El resultado debe estar ordenado de acuerdo al identificador del proyecto.

Código. Copie y pegue la sentencia SELECT-FROM-WHERE que resuelve la consulta.

```
SELECT PRO.ProjId, PRO.StartDate,PRO.EndDate FROM PROJECT PRO LEFT JOIN EMPLOYEE E ON  
(PRO.EMPNO = E.EMPNO) JOIN DEPARTMENT DEP ON(DEP.DEPTNO = E.DEPTNO) WHERE  
DEP.LOCALIZATION = 'NEW YORK' ORDER BY PRO.ProjId ASC
```

6. Consulta 4: Genere una tabla con el id de cada departamento, el nombre del departamento y la cantidad de empleados que allí trabajan. El resultado debe estar ordenado de acuerdo a la cantidad de empleados

Código. Copie y pegue la sentencia SELECT-FROM-WHERE que resuelve la consulta.

```
SELECT NVL(DEPT.DEPTNO,0) DEPARTMENT_ID, NVL(DEPT.DNAME,0) DEPARMENT_NAME,  
COUNT(E.EMPNO) NUMBER_EMPLOYEE FROM EMPLOYEE E LEFT JOIN DEPARTMENT DEPT ON (E.DEPTNO =  
DEPT.DEPTNO) GROUP BY DEPT.DEPTNO, DEPT.DNAME ORDER BY NUMBER_EMPLOYEE ASC;
```