

Universidad de Puerto Rico

Recinto de Humacao

Proyecto Final

Profesor MEDINA HUAMAN

Nombre Luis E. Aponte

5/23/2022

## Introducción

ProyectoFinal es un formulario creado para una compañía. De acuerdo a la información a continuación.

*Una compañía está organizada en departamentos. Cada departamento tiene un nombre único, un número único y un empleado particular que dirige el departamento. Se mantiene el registro de la fecha de inicio en la que el empleado empieza a dirigir el departamento. No se guarda registro de los empleados que han dirigido el departamento anteriormente.*

*Un departamento controla un número de proyectos, cada uno de los cuales tiene un nombre único, un número único y una ubicación donde se lleva a cabo. Se debe guardar registro de los empleados. Por cada empleado se necesita saber el nombre del empleado, el número de seguro social, dirección, salario, sexo y fecha de nacimiento. Un empleado es asignado a un departamento, pero puede trabajar en varios proyectos, que no necesariamente son controlados por el mismo departamento. Cada empleado tiene un supervisor inmediato (que es otro empleado). Para propósitos de seguros, se debe guardar registro de los dependientes del empleado. Para cada dependiente se debe saber el nombre, sexo, fecha de nacimiento y relación con el empleado.*

El programa que se utilizó para crear la base de datos y el formulario fue Oracle Apex. Por otro lado, se muestra el diagrama de entidad-relación de las tablas. También código de SQL que se utilizó para crear las tablas, editar datos y hacer consultas. Además, podrán observar, el diccionario de atributos. Por último, se muestra un ejemplo de cómo quedarían las tablas.

## Objetivo

- Identificar datos
- Crear diagrama entidad-relación
- Crear diccionario de atributo
- Crear tabla con SQL
- Crear auto incremento y trigger
- Crear formulario
- Editar tabla
- Tabla de ejemplo
- Ejemplos de consulta en SQL
- Conclusión

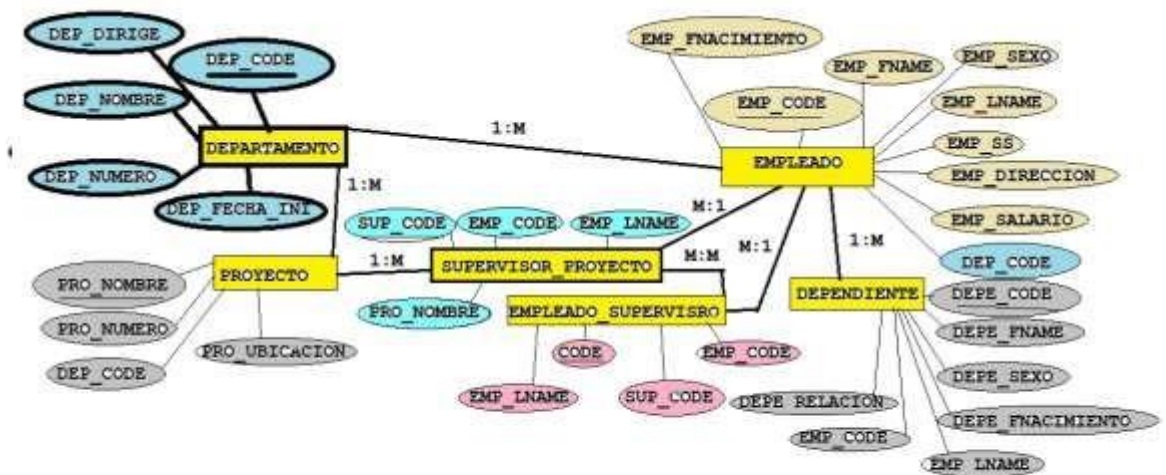
## Identificar datos

Luego de leer la información ofrecida por la compañía, se pasa a identificar que tablas se necesitan. Las tablas identificadas fueron las siguientes: departamento, empleado, dependiente, proyecto, supervisor proyecto, empleado\_proyecto. La tabla departamento necesita guardar los siguientes datos: nombre único, un número único, un empleado que dirige el departamento y fecha de inicio en la que el empleado empieza a dirigir el departamento. Pasando a la tabla empleado, se necesitan los siguientes datos: nombre del empleado, el número de seguro social, dirección, salario, sexo, fecha de nacimiento y departamento. Por otro lado, la tabla proyecto necesita: nombre único, un número único, una ubicación y departamento. Las tablas empleado y proyecto son dependientes de departamento. Luego tenemos la tabla dependiente que depende de la tabla empleado. Los datos que se quieren en esta tabla son los siguientes: nombre, sexo, fecha de nacimiento y relación con el empleado. Siguiendo esta la tabla supervisor proyecto que depende de la tabla proyecto y

empleado. Esta tabla va a guarda quien es el supervisor que está a cargo del proyecto. Por último, la tabla empleado supervisor. Dicha tabla depende de la tabla supervisor proyecto y empleado. Esta tabla guarda quien son los supervisores de cada empleado.

## Diagrama entidad-relación

Un diagrama entidad-relación es un tipo de diagrama de flujo que ilustra cómo las "entidades" se relacionan entre sí dentro de un sistema. El diagrama entidad-relación que se muestra a continuación es el que se utiliza para crear las tablas.



## Diccionario de atributo

Aquí se define la relación que hay entre tablas.

ENTIDAD	RELACIÓN	CONEXIÓN	ENTIDAD
DEPARTAMENTO	posee	1:M	Empleado
EMPLEADO	tiene	1:M	Dependiente
DEPARTAMENTO	funciona	1:M	Proyecto
PROYECTO	incluye	M:1	Supervisor_proyecto
SUPERVISOR_PROYECTO	lidera	1:M	Empleado_supervisor
EMPLEADO	se utiliza para	M:1	Supervisor_proyecto
EMPLEADO	se utiliza para	1:M	Empleado_supervisor

## Crear tabla (con SQL)

Para crear las tablas, se utiliza la aplicación de Oracle apex. Oracle APEX es una plataforma de desarrollo de bajo código que le permite crear aplicaciones sorprendentes, escalables y seguras con características de clase mundial que se pueden implementar en cualquier lugar. Además, los desarrolladores pueden desarrollar e implementar rápidamente aplicaciones atractivas que resuelven problemas reales y brindar resultados inmediatos. Hay diferente manera de crear tablas. Aquí se estará documentado como se crean con SQL.

```
CREATE TABLE DEPARTAMENTO1 (  
  DEP_CODE integer NOT NULL UNIQUE,  
  DEP_NOMBRE varchar(25) NOT NULL UNIQUE,  
  DEP_NUMERO integer NOT NULL UNIQUE,  
  DEP_DIRIGE varchar(25) ,  
  DEP_FECHA_INI DATE not null ,  
  PRIMARY KEY (DEP_CODE)  
);
```

```
CREATE TABLE EMPLEADO(  
  EMP_CODE integer NOT NULL UNIQUE,  
  EMP_FNAME varchar(25) not null ,  
  EMP_LNAME varchar(25) not null ,  
  EMP_SS integer NOT NULL UNIQUE,  
  EMP_DIRECCION varchar(50) not null,  
  EMP_SALARIO NUMERIC(5,2) not null,  
  EMP_SEXO char(1) not null,  
  EMP_FNACIMIENTO date not null ,  
  DEP_CODE integer not null, primary  
  key(EMP_CODE,EMP_LNAME),
```

FOREIGN KEY (DEP\_CODE) REFERENCES

DEPARTAMENTO1 (DEP\_CODE)

);

CREATE TABLE DEPENDIENTE (

DEPE\_CODE integer not null unique,

DEPE\_NUMERO integer not null,

DEPE\_FNAME varchar(15) not null,

DEPE\_LNAME varchar(25) not null,

DEPE\_SEXO varchar(1) not null,

EMP\_CODE integer not null,

EMP\_lname integer not null,

DEPE\_FNACIMIENTO DATE not null,

DEPE\_RELACION varchar(25) not null, primary key (DEPE\_CODE),

FOREIGN KEY(EMP\_CODE) REFERENCES

EMPLEADO (EMP\_CODE)

);

)

## Auto incremento y trigger(con SQL)

Para asegurar que no se repita ninguna id de las filas, se le programo un auto incremento. Para cuando ingresen valor. La id o código se actualice automáticamente. A continuación se muestre un ejemplo de un auto incremento y sus trigger para las tablas(se hace lo mismo para las otras 5).

CREATE SEQUENCE UNO\_SUPERVISOR\_PROYECTO

START WITH 1

```

INCREMENT BY 1;

CREATE TRIGGER TRI_SUPERVISOR_PROYECTO
BEFORE INSERT ON SUPERVISOR_PROYECTO
FOR EACH ROW

BEGIN

SELECT UNO_SUPERVISOR_PROYECTO.NEXTVAL INTO :NEW.SUP_CODE

FROM DUAL; END;

```

## Crear formulario

Los formularios son objetos de base de datos que pueden usarse para crear una interfaz de usuario para una aplicación de base de datos. Para crear un formulario, se le da a crear una aplicación y se selecciona la opción de formulario.



## Editar tabla( con SQL)

### Insertar fila

-INSERT INTO

```

DEPENDIENTE(DEPE_FNAME,DEPE_LNAME,DEPE_SEXO,DEPE_FNACIMIENTO,DEPE_RELACION,EMP_CODE,EMP_LNAME) VALUES ('Victoria','Lacen','F','5/1/2000','hija','2','LACEN CIRINO')

```

### Editar fila

UPDATE departamento

SET DEP\_NOMBRE = 'planta1'

WHERE DEP\_CODE = 1;

### Eliminar fila

DELETE FROM DEPENDIENTE

WHERE DEP\_NUMERO;

## Tabla

### Departamento

DEP_CODE	DEP_NOMBRE	DEP_NUMERO	DEP_DIRIGE	DEP_FECHA_INI
2	PLANTA2	2	Juan Lacen	3/9/2016
1	planta1	1	Luis Aponte	3/8/2016
62	PLANTA 4	4	Mercedes Navarro	4/10/2016
61	planta3	3	Eduardo Lopez	2/9/2016

### EMPLEADO

EMP_CODE	EMP_FNAME	EMP_LNAME	EMP_SS	EMP_DIRECCION	EMP_SALARIO	EMP_SEXO	EMP_FNACIMIENTO	DEP_CODE
1	Luis	Aponte Lopez	123456789	po box 275,00772	15	M	4/8/1998	1
63	Barbara	Navarro Gimenez	119156789	po box 451 , 00772	30	F	2/12/1985	62
64	Ramon	Berrios Aponte	111456781	Ba san isidro,car186,casa4a	15	M	4/21/1997	62
66	Reinaldo	Afran Gonzales	111446781	Ba san isidro,car186,casa4a	15	M	4/21/1997	62



## PROYECTO

PRO_NOMBRE	PRO_NUMERO	PRO_UBICACION	DEP_CODE
COVID19	81	LABORATORIO BAYAMON	62
MATENIMINETO PAGINA WEB	23	OFICINA BAYAMON	2
HEROKU	24	OFICINA BAYAMON	2
AB	10	OFICINA	61
ACTUALIZAR BASE DE DATOS	21	OFICINA BAYAMON	1
MANTENIMIENTO	1	cardera , edificio A	1
APP NBA	22	OFICINA BAYAMON	1

## SUPERVISOR\_PROYECTO

SUP_CODE	EMP_CODE	EMP_LNAME	PRO_NOMBRE
1	3	Aponte Lopez	Mantenimiento
25	3	Aponte Lopez	ACTUALIZAR BASE DE DATOS
26	21	Berrios Rodriguez	MATENIMINETO PAGINA WEB
27	21	Berrios Rodriguez	HEROKU
31	3	Aponte Lopez	APP NBA

## EMPLEADO\_SUPERVISOR

CODE	EMP_CODE	EMP_LNAME	SUP_CODE
3	1	Aponte Lopez	1
4	2	LACEN CIRINO	1
10	64	Berrios Aponte	38
8	61	Lopez Aponte	26
27	67	Rivera Lopez	38
13	66	Afran Gonzales	38
23	61	Lopez Aponte	50
24	65	Navarro Lopez	38

## DEPENDIENTE

DEPE_CODE	DEPE_FNAME	DEPE_LNAME	DEPE_SEX	DEPE_FNACIMIENTO	DEPE_RELACION	EMP_CODE	EMP_LNAME
3	Damian	Gonzales	M	12/13/2019	sobrino	1	Aponte Lopez
23	Victoria	Lacen	F	5/1/2000	hija	2	LACEN CIRINO
104	Damian	Berrios	M	5/1/2022	hijo	64	Berrios Aponte
5	Eduardo	Aponte	M	7/20/2021	hijo	1	Aponte Lopez
101	Milagros	Figueroa	F	5/1/2021	hija	63	Navarro Gimenez
102	Barbara	Figueroa	F	5/1/2022	hija	63	Navarro Gimenez

## Ejemplos de sentencia SQL

- **Determina el promedio del salario**

```
select AVG(EMP_SALARIO) from empleado
```

- **Determina la cantidad de empleado**

```
select COUNT(EMP_CODE) from empleado
```

- **Cuanto empleado tiene cada supervisor**

```
SELECT SUP_CODE , COUNT(CODE) ASCANTIDAD_EMPLEADO
FROM EMPLEADO_SUPERVISOR
GROUP BY SUP_CODE
```

- **Cuanto proyecto tiene cada empleado**

```
SELECT EMP_CODE , COUNT(SUP_CODE) AS TOTAL_PROYECTO
FROM EMPLEADO_SUPERVISOR
GROUP BY EMP_CODE
```

- **Devuelve el supervisor de cada departamento**

```
SELECT DISTINCT DEP_NUMERO, DEP_NOMBRE,EMP_LNAME
FROM (DEPARTAMENTO NATURAL JOIN PROYECTO) NATURAL JOIN SUPERVISOR_PROYECTO
```

## Conclusión

Se logro cumplir con cada uno del objetivo. El formulario permite que una persona que no sepa el código de SQL, pueda ingresar datos. Muy útil para cuando uno está creando una base de dato y no eres el que la va a estar manejando. Por otro lado, entiendo que se puedo mejorar, para guardar la información más eficiente.