

Actividad | #1| Creación de la Base

de Datos

Ingeniería en Desarrollo de Software



TUTOR: Felix Acosta Hernandez

ALUMNO: Edgar Enrique Cuamea Ochoa

FECHA: 27 de marzo del 2024

Contenido

Introducción	3
Descripción	4
Justificación	5
Desarrollo.....	6
Modelo entidad relación	6
Modelo lógico relacional.	8
Base de Datos.....	10
Conclusión.....	17
Referencias.....	18

Introducción.

En esta actividad veremos conceptos básicos sobre la programación orientada a objetos tales como las herencias simples o herencias múltiples, así como declarar la clase y los atributos de los diferentes objetos que estableceremos mas adelante, por lo que es necesario graficar el modelo de entidad relación, así como el modelo lógico relacional para utilizar como base para la realización de la base de datos en Microsoft SQL server , realizaremos en total 10 diferentes clases, todas estas con sus propios atributos, realizaremos los modelos de entidad relación y lógico relacional antes de realizar la base de datos en SQL server ya que tenemos que analizar el contexto de la actividad antes de realizar las diferentes tablas para estas clases así como la asignación de los atributos ya que tendremos que diferenciar cuantas clases realizaremos así como cuales serán sus atributos, además de realizar los modelos con anterioridad para realizar las clases de manera correcta así como realizarlo de manera más fácil al tener los modelos terminados.

Descripcion.

Realizaremos una base de datos en Microsoft SQL server para la realización de la actividad pues tendremos que analizar un contexto donde se nos indicara cuales clases y atributos tendremos que realizar por lo que es necesario descargar e instalar Microsoft SQL server para la realización de esta actividad, analizaremos el contexto y realizaremos diferentes graficas donde realizaremos el modelo de entidad relación así como el modelo lógico relacional antes de realizar la base de datos por lo que utilizaremos estos modelos como base para realizar la base de datos, agregare diferentes capturas donde se muestre el proceso de la creación de la base de datos en Microsoft SQL server, agregaremos 5 diferentes registros a una tabla por lo que mostraremos el registro guardado en la tabla seleccionada, mediante esta actividad aprenderemos a utilizar SQL server para la creación de las bases de datos así como agregar información en estas tablas como las clases y atributos además de realizar registros en estas bases de datos.

Justificación.

Utilizaremos el programa de Microsoft SQL Server para la creación de nuestra base de datos ya que este programa es bastante conocido para la gestión de las bases de datos además de ofrecer herramientas integradas que nos simplifican la administración de estas bases de datos además de proporcionar un ecosistema de Microsoft ya que funcionan con diferentes programas de esta compañía por lo que la seguridad de este programa es robusta con un buen rendimiento ya que maneja un volumen alto de datos de manera eficiente además de que el lenguaje SQL es un estándar para la creación de bases de datos por lo que podremos aplicarla al crear una base de datos local así como realizar sistemas donde se necesite una gestión de datos como una tienda de ropa y así poder clasificar clases y atributos de diferentes objetos por lo que aprenderemos a realizar estas bases de datos aplicando los modelos de entidad relación y lógico relacional para poder ver de forma gráfica como es que estas clases se relacionan con diferentes clases así como los atributos que pueden ser compartidos de diferentes clases.

Desarrollo.

Se necesita una estructura de clases que permita a la empresa UNI controlar los distintos tipos de empleados, así como sus datos personales. Esto se hará a través de clases, herencia de clases y atributos. Las clases, por su parte, deberán ser usadas desde una aplicación donde se gestione la siguiente información:

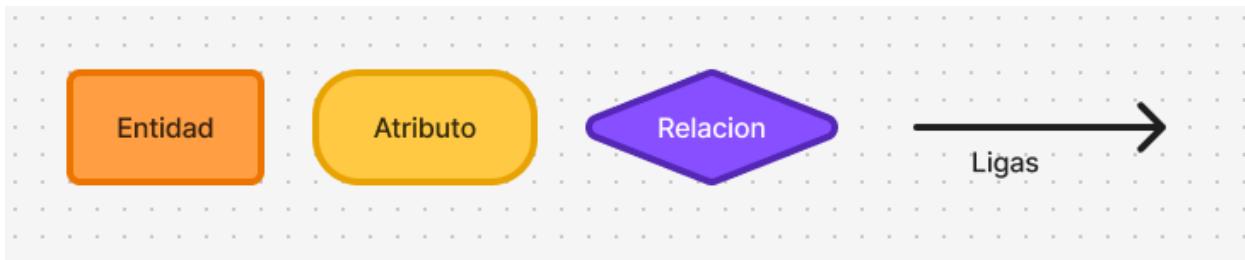
- Número de Empleado: (autogenerado, numérico)
- Nombre: (capturable, alfanumérico)
- Apellido Paterno: (capturable, alfanumérico)
- Apellido Materno: (capturable, alfanumérico)
- Fecha de Nacimiento: (capturable tipo fecha)
- RFC: (calculado conforme al nombre y fecha de nacimiento, alfanumérico)
- Centro de Trabajo: (capturable, alfanumérico, elegible desde el número de clave con base en el catálogo de puestos)
- Puesto: (capturable, alfanumérico)
- Descripción del Puesto: (capturable, alfanumérico)
- Directivo: (bandera para indicar tipo de empleado; para directivo 1; para empleado normal 0)

Con base al contexto presentado creamos nuestro modelo entidad relación, así como el modelo lógico relacional;

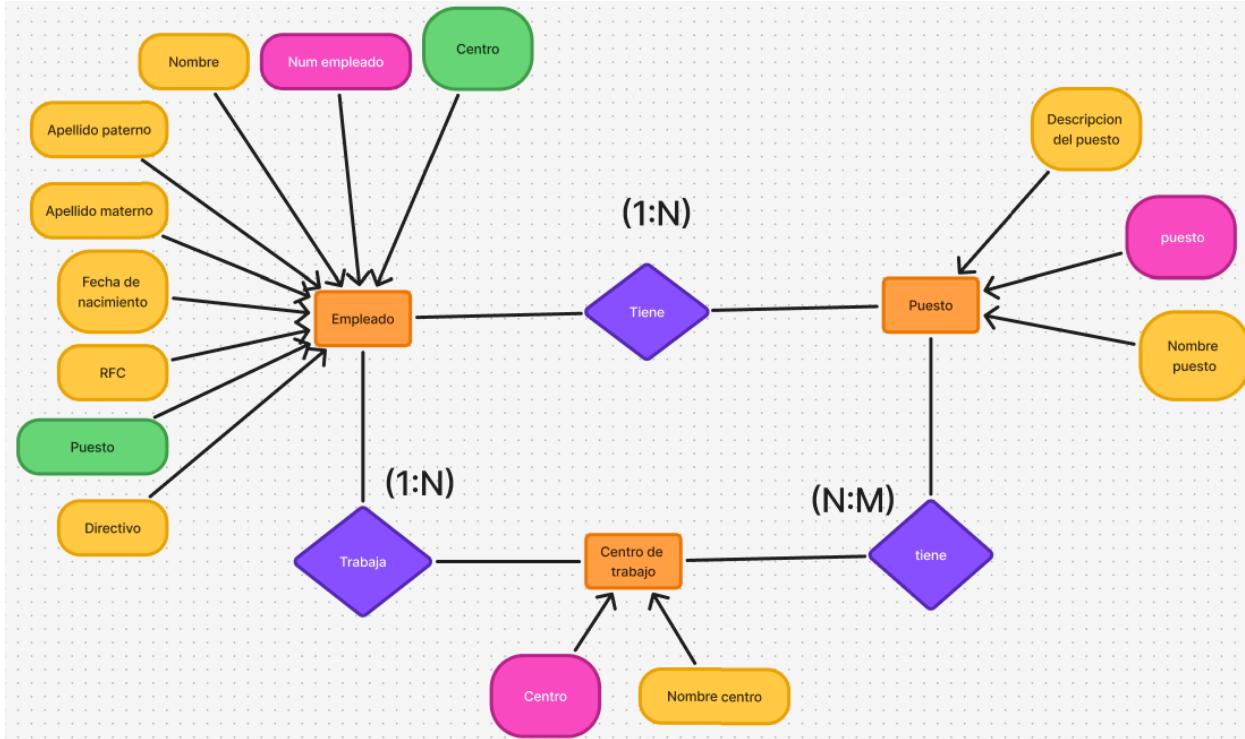
Modelo entidad relacion

Para este modelo crearemos diferentes clases, así como los atributos y como es que se relacionan entre ellos utilizando diferentes figuras geométricas, así como distintos colores y líneas

indicando como se relacionan las diferentes clases.



Usaremos estas formas de diferentes colores para diferenciar las entidades de los atributos y que relaciones tienen entre ellos por lo que realizamos nuestro modelo entidad relación de la siguiente forma

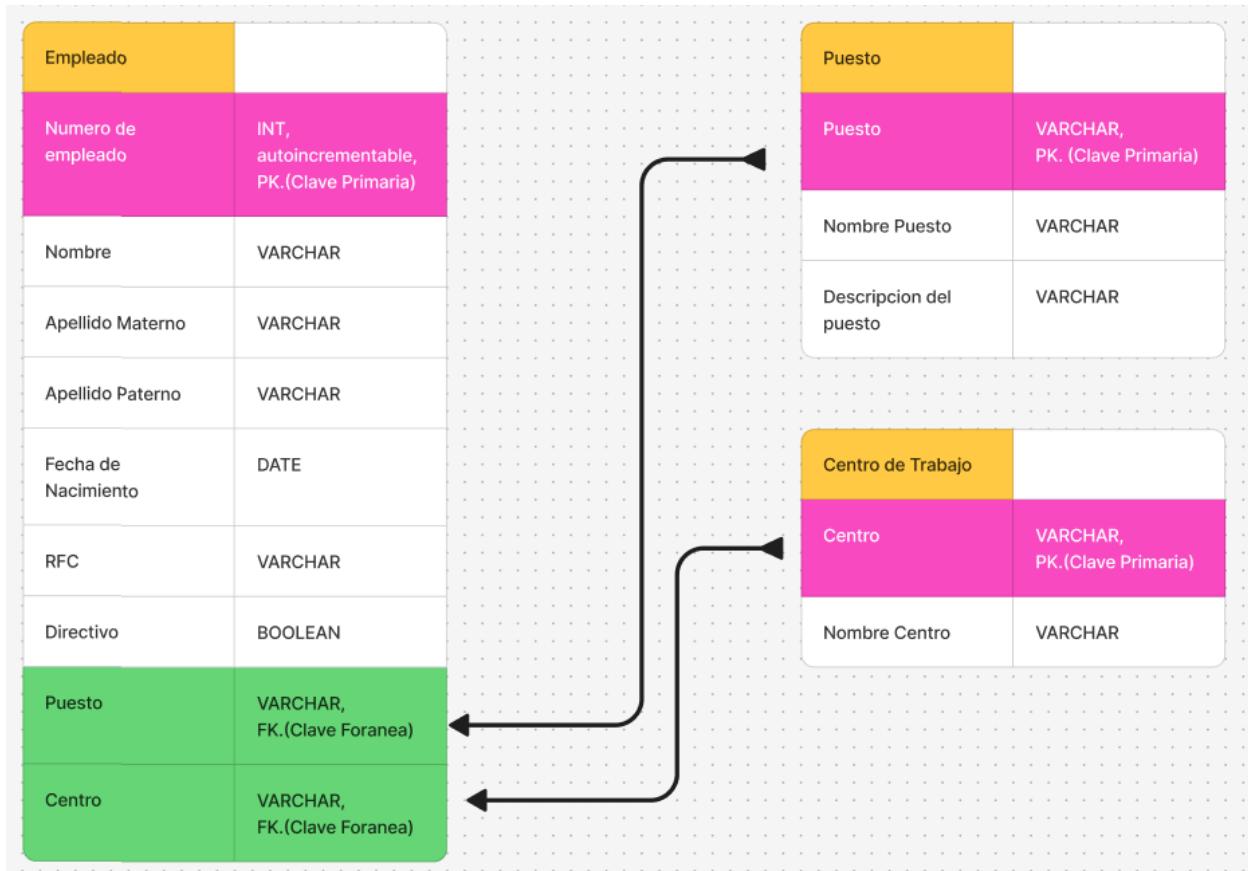


Tenemos nuestra entidad empleado donde este cuenta con nombre, apellidos, fecha de nacimiento, RFC, directivo, puesto, numero de empleado y centro, estos últimos 3 con diferentes colores ya que representan las claves primaria y foránea que se necesitaran mas adelante por lo que el color rosa es la clave primaria y el color verde la clave foránea, en este caso el centro de la

entidad empleado es una clave foránea de la entidad centro de trabajo que contiene el nombre del centro y la clave primaria de centro, igual con la clave foránea puesto de la entidad empleado que esta viene de la entidad puesto que contiene la clave primaria de puesto, nombre del puesto y descripción de este, los rombos presentan las relaciones entre estas entidades por lo que un empleado tiene un solo puesto pero el puesto puede ser ocupados por muchos empleados por lo que se representa de la siguiente forma (1:N) uno a muchos, por lo que un empleado trabaja en un centro de trabajo y el centro de trabajo puede tener muchos empleados, de la misma forma con la relación de centro de trabajo con puesto, ya que un centro de trabajo puede tener muchos puestos y un puesto puede tener muchos centros de trabajo que se representa de la siguiente forma (N:M) muchos a muchos. Por lo que también se puede representar como (1:N) ya que un solo centro de trabajo puede tener muchos puestos, lo que hace que utilizar un solo centro de trabajo haga que el proceso de la creación de la base de datos sea más fácil al eliminar la necesidad de realizar mas tablas para los diferentes centros de trabajo.

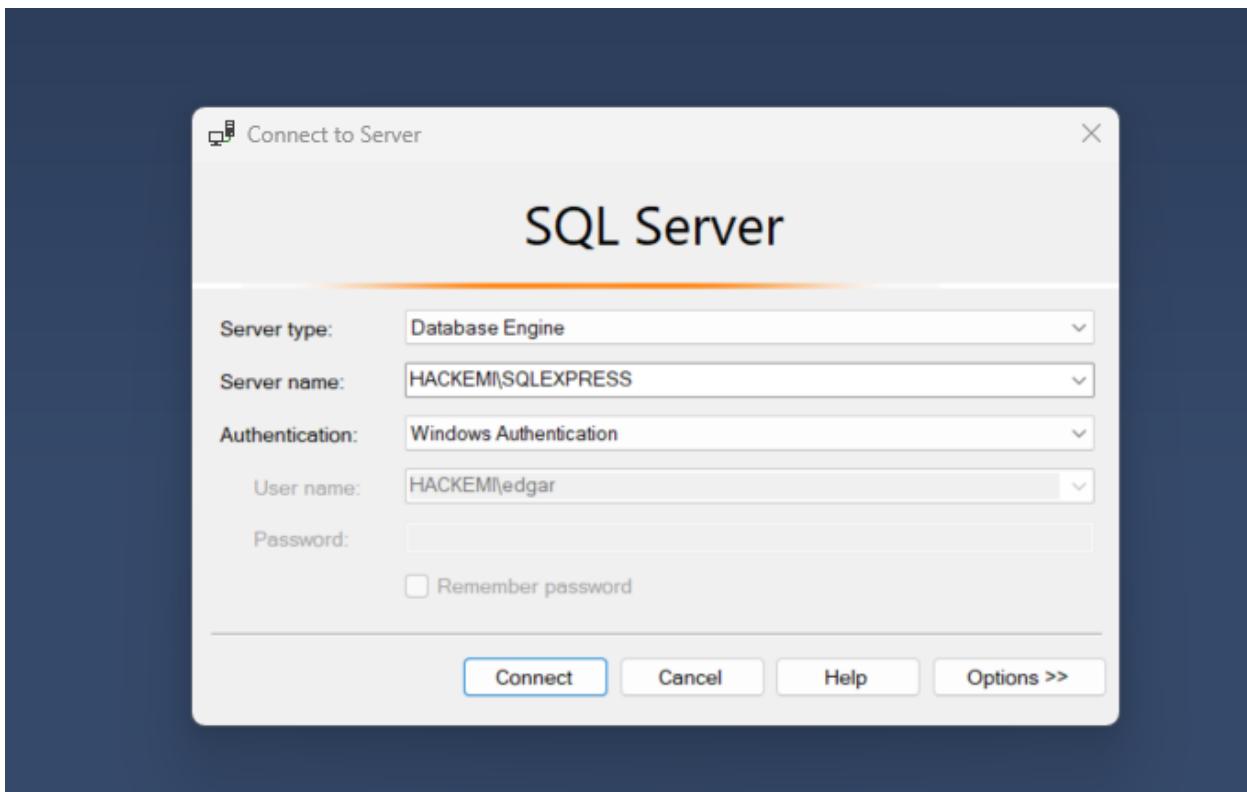
Modelo lógico relacional.

El modelo lógico relacional es parecido al modelo entidad relación, en cambio, este modelo utiliza tablas y la asignación de claves foráneas y claves primarias que vimos anteriormente, utilizaremos tablas para representar las entidades y sus atributos, así como representar las claves primarias y claves foráneas para establecer las relaciones, así como el tipo de dato que este contendrá por lo que realizamos la siguiente tabla:

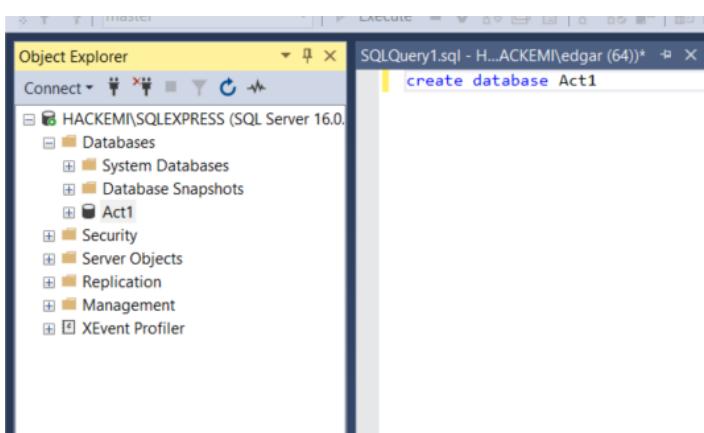


Tenemos nuestras tablas de empleado, puesto y centro donde cada una de estas contiene su clave primaria marcado en color rosa, por lo que la clave foránea está representada con color verde, cuales viene de las tablas puesto y centro de trabajo. Una vez que realizamos estos diagramas, realizaremos nuestra base de datos en Microsoft SQL server.

Base de Datos.



Creamos nuestra nueva base de datos donde estaremos agregando las tablas y los datos presentadas en el contexto de nuestra actividad por lo que generamos nuestro servidor y creamos nuestra base de datos



Usamos el comando create database para crear una base de datos asignandole el nombre de Act1 y ejecutamos el comando presionando f5 para crear nuestra base de datos.

```
SQLQuery1.sql - H:\CKEM\edgar (64)» ▶ X
USE Act1
CREATE TABLE Empleados(
    Num_Empleado INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    Nombre VARCHAR(50) NOT NULL,
    Apellido_Paterno VARCHAR(50) NOT NULL,
    Apellido_Materno VARCHAR(50) NOT NULL,
    Fecha_Nacimiento DATE NOT NULL,
    RFC AS (UPPER(SUBSTRING(Nombre,1,2) + SUBSTRING(Apellido_Paterno,1,2) + SUBSTRING(Apellido_Materno,1,1) + FORMAT(Fecha_Nacimiento, 'yyMMdd'))),
    Centro_Trabajo VARCHAR(50) NOT NULL,
    Puesto VARCHAR(50) NOT NULL,
    Descripcion_Puesto VARCHAR(100) NOT NULL,
    Directivo BIT NOT NULL );
```

Usamos el comando create table empleados para crear nuestra tabla de empleados agregando todas las columnas que necesita nuestra tabla así como la llave primaria y el tipo de datos que se registrara en la columna como el dato de nombre que regisra el tipo de dato como una cadena de caracteres (varchar) donde entre los parentesis especificamos cuantos caracteres maximos puede tener, así como especificar el tipo not null para evitar que se registre un nombre vacio, por lo que al ajecutar el codigo, nos guardara la siguiente tabla

Columna	Tipo de Dato	Propiedades
Num_Empleado	INT	PK, int, not null
Nombre	VARCHAR(50)	not null
Apellido_Paterno	VARCHAR(50)	not null
Apellido_Materno	VARCHAR(50)	not null
Fecha_Nacimiento	DATE	not null
RFC	Computed, nvarchar(4000)	null
Centro_Trabajo	VARCHAR(50)	not null
Puesto	VARCHAR(50)	not null
Descripcion_Puesto	VARCHAR(100)	not null
Directivo	BIT	not null

Aquí podremos ver las llave primaria así como las columnas y el tipo de dato que contiene por lo que realizaremos el mismo proceso para la tabla de puesto y centro de trabajop cambiando los datos que estos registrarán

```

USE Act1
CREATE TABLE Centro_Trabajo(
    Centro varchar(50) primary key not null,
    Nombre_centro varchar(50) not null);

USE Act1
CREATE TABLE Centro_Trabajo(
    Centro varchar(50) primary key not null,
    Nombre_puesto varchar(50) not null,
    Descripcion_puesto varchar(50) not null);

```

Una vez escrito y ejecutado el código, este creará las diferentes tablas que necesitamos por lo que revisamos que se hayan creado correctamente

```

dbo.Centro_Trabajo
  Columns:
    Centro (PK, varchar(50), not null)
    Nombre_centro (varchar(50), not null)
  Keys:
  Constraints:

dbo.Puesto
  Columns:
    Puesto (PK, varchar(50), not null)
    Nombre_puesto (varchar(50), not null)
    Descripcion_puesto (varchar(50), not null)
  Keys:

```

Vemos que los datos de las tablas están creados correctamente verificando que estos no acepten valores nulos además de tener su propia llave primaria para estas tablas por lo que al tener la tabla de empleado, este necesariamente usa llaves foraneas de la tabla puesto y centro_Trabajo por lo que relacionaremos las tablas de la siguiente forma:

```

alter table Empleados add constraint FK_Centro_Trabajo foreign key (Centro_Trabajo) references Centro_Trabajo (Centro);
alter table Empleados add constraint FK_ID_Puesto foreign key (ID_Puesto) references Puesto (ID_Puesto);

```

Con estos comandos alteramos la tabla empleados donde agregamos una restricción donde estableceremos el nombre de la llave foránea empezando por FK para diferenciar la llave foránea donde usaremos foreign key para relacionar la llave foránea, indicamos la columna que se

convertira en llave foranea, en este caso (ID_Puesto) por lo que seleccionamos la referencia de las llaves primarias indicando en que tabla se encuentra y cual es la llave primaria, en este caso indicndo que la llave primaria se encuentra en la tabla puesto y la llave primaria se llama (ID_Puesto) entre parentesis, por lo que realizamos el mismo proceso con la llave foranea de centro de trabajo por lo que verificamos en la tabla:

En esta tabla podemos ver que la tabla empleados cuenta con las llaves foraneas de Centro_Trabajo así como la llave foranea de ID_Puesto

Podremos usar el comando: Alter table Empleados add foreign key (ID_Puesto) references Puesto (ID_Puesto) para asignarle la llave foranea, lo que realiza el mismo proceso con un comando mas corto.

Una vez creadas nuestras tablas, agregaremos 5 registros para comprobar la funcionalidad de la base de datos que creamos por lo que agregamos datos a puesto con el siguiente comando

```
insert into Puesto (ID_Puesto,Nombre_puesto,Descripcion_puesto) values ('01','cajero','cobra en punto de venta'),
('02','vendedor','atiende al cliente'),
('03','gerente','dirige tienda')

select * from Puesto
```

Este nos creara la tabla donde establecimos los datos anteriores

Results Messages		
ID_Puesto	Nombre_puesto	Descripcion_puesto
1	01	cajero
2	02	vendedor
3	03	gerente
		cobra en punto de venta
		atiende al cliente
		dirige tienda

Una vez agregados los puestos, agregaremos el centro de trabajo

```
insert into Centro_Trabajo (Centro,Nombre_centro) values ('0281','Colosio');
select * from Centro_Trabajo
```

Results Messages	
Centro	Nombre_centro
1	0281 Colosio

Por ultimo, agregaremos a los empleados utilizando el mismo codigo

```
insert into Empleados (Nombre,Apellido_Paterno,Apellido_Materno,Fecha_Nacimiento,Centro_Trabajo,ID_Puesto,Descripcion_Puesto,Directivo)
values('Edgar','Cuamea','Ochoa','2001-02-15','0281','02','atiende al cliente','0');

select * from Empleados
```

	Num_Emppleado	Nombre	Apellido_Paterno	Apellido_Materno	Fecha_Nacimiento	RFC	Centro_Trabajo	ID_Puesto	Descripcion_Puesto	Directivo
1	1	Edgar	Cuamea	Ochoa	2001-02-15	EDCUU010215	0281	02	atiende al cliente	0

En este caso el RFC esta calculado de manera erronea por lo que eliminamos la columna de RFC y agregamos una nueva con los siguientes valores

```
alter table Empleados
add RFC AS (UPPER(SUBSTRING(Apellido_Paterno,1,2) + SUBSTRING(Apellido_Materno,1,1) + SUBSTRING(Nombre,1,1) + FORMAT(Fecha_Nacimiento, 'yyMMdd')))
```

Es importante tener escrito correctamente el calculo de RFC ya que hay reglas establecidas para este calculo por lo que al actualizar los datos, vemos que se calcula el RFC de manera correcta

100 %										
Results Messages										
	Num_Emppleado	Nombre	Apellido_Paterno	Apellido_Materno	Fecha_Nacimiento	Centro_Trabajo	ID_Puesto	Descripcion_Puesto	Directivo	RFC
1	1	Edgar	Cuamea	Ochoa	2001-02-15	0281	02	atiende al cliente	0	CUOE010215
2	2	Edgar	Cuamea	Ochoa	2001-02-15	0281	02	atiende al cliente	0	CUOE010215
3	3	Edgar	Cuamea	Ochoa	2001-02-15	0281	02	atiende al cliente	0	CUOE010215
4	4	Edgar	Cuamea	Ochoa	2001-02-15	0281	02	atiende al cliente	0	CUOE010215

Por lo que una vez arreglado este detalle, agregaremos 4 empleados mas

```
insert into Empleados (Nombre,Apellido_Paterno,Apellido_Materno,Fecha_Nacimiento,Centro_Trabajo,Id_Puesto,Descripcion_Puesto,Directivo)
values
('Edwin','Martinez','Marquez','2002-01-25','0281','01','cobra en punto de venta','0'),
('Julio','Perez','Gomez','2000-04-24','0281','02','atiende al cliente','0'),
('Victoria','Hernandez','Diaz','2003-08-11','0281','02','atiende al cliente','0'),
('Jose','Martinez','Sanchez','2001-09-11','0281','01','cobra en punto de venta','0'),
('Alejandro','Ramirez','Torrez','2002-12-24','0281','03','dirige tienda','1');

select * from Empleados
select * from Puesto
select * from Centro_Trabajo
```

Ejecutamos el codigo y vemos que en la tabla de empleados aparecen los empleados registrados con sus datos que agregamos así como los centros de trabajo y los puestos

The screenshot shows the results of the executed SQL code. It displays three tables:

- Empleados** (Employees):

Num_Emppleado	Nombre	Apellido_Paterno	Apellido_Materno	Fecha_Nacimiento	RFC	Centro_Trabajo	ID_Puesto	Descripcion_Puesto	Directivo
1	Edgar	Cuamea	Ochoa	2001-02-15	EDCU0010215	0281	02	atiende al cliente	0
2	Edgar	Cuamea	Ochoa	2001-02-15	EDCU0010215	0281	02	atiende al cliente	0
3	Edgar	Cuamea	Ochoa	2001-02-15	EDCU0010215	0281	02	atiende al cliente	0
4	Edgar	Cuamea	Ochoa	2001-02-15	EDCU0010215	0281	02	atiende al cliente	0
5	Edwin	Martinez	Marquez	2002-01-25	EDMAM020125	0281	01	cobra en punto de venta	0
6	Julio	Perez	Gomez	2000-04-24	JUPEG000424	0281	02	atiende al cliente	0
7	Victoria	Hernandez	Diaz	2003-08-11	VIHED030811	0281	02	atiende al cliente	0
8	Jose	Martinez	Sanchez	2001-09-11	JOMAS010911	0281	01	cobra en punto de venta	0
9	Alejandro	Ramirez	Torrez	2002-12-24	ALRAT021224	0281	03	dirige tienda	1
- Puesto** (Position):

ID_Puesto	Nombre_puesto	Descripcion_puesto
01	cajero	cobra en punto de venta
02	vendedor	atiende al cliente
03	gerente	dirige tienda
- Centro_Trabajo** (Work Center):

Centro	Nombre_centro
000201	Angel Flores Ropa
000202	Angel Flores Muebles
000203	Angel Flores Cajas
0281	Colosio
049001	La Primavera Ropa
049002	La Primavera Muebl...
049003	La Primavera Cajas

Por lo que vemos los puestos que definimos, los centros de trabajo y los empleados, vemos que edgar esta repetido 4 veces por lo que podemos eliminarlos utilizando el siguiente codigo

```
delete from Empleados where Num_Emppleado <4 por lo que eliminaria los primeros 3 registros ya que su numero de empleado son menores a 4
```

```
l 045005 , la primavera ladas ;j
delete from Empleados where Num_Empleado < 4
```

100 % ▶

Results Messages

	Num_Empleado	Nombre	Apellido_Paterno	Apellido_Materno	Fecha_Nacimiento	RFC	Centro_Trabajo	ID_Puesto	Descripcion_Puesto	Directivo
1	4	Edgar	Cuamea	Ochoa	2001-02-15	EDCU0010215	0281	02	atiende al cliente	0
2	5	Edwin	Martinez	Marquez	2002-01-25	EDMAM020125	0281	01	cobra en punto de venta	0
3	6	Julio	Perez	Gomez	2000-04-24	JUPEG000424	0281	02	atiende al cliente	0
4	7	Victoria	Hernandez	Diaz	2003-08-11	VIHED030811	0281	02	atiende al cliente	0
5	8	Jose	Martinez	Sanchez	2001-09-11	JOMAS010911	0281	01	cobra en punto de venta	0
6	9	Alejandro	Ramirez	Torrez	2002-12-24	ALRAT021224	0281	03	dirige tienda	1

	ID_Puesto	Nombre_puesto	Descripcion_puesto
1	01	cajero	cobra en punto de venta
2	02	vendedor	atiende al cliente

El ejecutar el código, vemos que efectivamente elimino los registros que le especificamos

Conclusion.

En conclusión es importante aprender la sintaxis al momento de crear las tablas usando el código así como la construcción del RFC ya que hay una serie de reglas que necesitamos seguir para la creación de un RFC correcto además de entender de manera fácil el funcionamiento de las llaves primarias y llaves foráneas por lo que el asignarlos es una tarea fácil conociendo la sintaxis de los comandos utilizados ya que hay diferentes formas de agregar tablas o editarlas así como poder asignar estas llaves, ya que el equivocarse en alguna parte del código como un símbolo faltante o una letra que escribimos mal, hará que el código no se ejecute correctamente por lo que al realizar los modelos lógico relacional y entidad relación antes de realizar la base de datos hace que el proceso de creación de la bases de datos sea más fácil al tener como base unas tablas donde definamos las entidades con sus atributos, se puede complicar la tarea al agregar más tablas así como agregar mas datos y sus relaciones ya que tomara mas tiempo el realizar las relaciones de manera correcta ya que la inserción de los datos en las tablas es fácil de realizar por código además de que Microsoft SQL management nos permite realizar la base de datos sin código de una manera un poco más lenta pero gráficamente fácil de realizar.

Referencias.

- Franklin García. (2020, 17 febrero). *Como Crear BASE DE DATOS en SQL SERVER Desde Cero* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=fyyEhDgKI7E>
- Franklin García. (2020b, febrero 25). *Crear TABLAS en SQL Server con y sin CODIGO* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=rHcB0KGoysw>