

UNIVERSIDAD DON BOSCO

BASE DE DATOS PARA MOVILES



Investigación 1 – BDM441

Grupo:

G01L

Nombres	Apellidos	N° Carnet
William Ernesto	Ramos Valladares	RV200068
Javier Eliseo	Gutiérrez Flores	GF220089
Axel Giovanni	Ramírez Alfaro	RA160395
Kenya Elizabeth	Parada Palma	PP220664

DOCENTE:

Kevin Jiménez

INDICE

Portada-----	1
Introducción-----	3
Objetivos-----	4
1. Tabla Producto	
1.1 Ejemplo-----	5
2. Campo Calculado	
2.1 Ejemplo-----	6-8
Diagrama Entidad Relación-----	9
Diccionario de Datos-----	9-12
Diagrama de la Base de Datos-----	13
Conclusión-----	14

INTRODUCCIÓN

Este trabajo de investigación, hablamos sobre una solución al momento de presentar la problemática con respecto a un supermercado, siempre que vemos la cantidad de productos y variedades en artículos que existen y seguirán sacando, como poder organizarlo, de qué manera llevar un archivo ordenado.

Presentamos con la ayuda de SQL Server, una base de datos, donde se puede considerar el primer paso para realizar un gran proyecto, para insertar, modificar, eliminar, y tener un control eficaz del supermercado.

También, con la investigación darnos cuenta de cómo internamente funciona un supermercado, como se hacen las cosas, los roles tan importantes de cada empleado, las normas que se deben cumplir, los patrones y procesos que se siguen, es por eso que diseñamos el proyecto desde cero y donde nos dimos cuenta que las posibilidades son enormes.

OBJETIVOS

- Desarrollo e implantación de un sistema de información global, que permite gestionar e interconectar los diferentes sistemas y dispositivos que componen un supermercado.
 - Tener un mayor control con el Stock de las tiendas.
- Organizar todo el producto, de una forma eficaz, incluso al momento de hacer inventario y ventas.
 - Aumentar resultados y disponer de una rentabilidad mejorada ahorrando, tiempo y dinero.

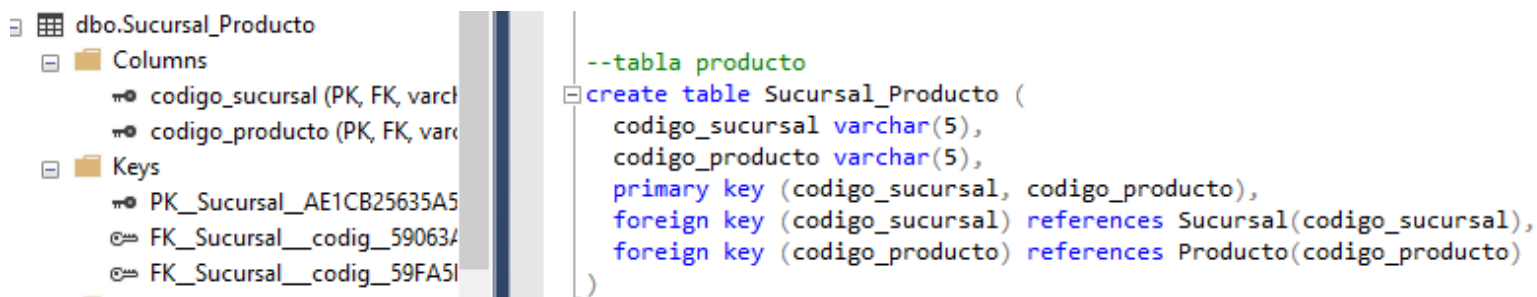
1. Tabla Producto

Una tabla unitaria es necesaria en una relación muchos a muchos porque permite representar de manera eficiente las relaciones entre dos conjuntos de entidades, sin tener que duplicar información.

Esta tabla actúa como una tabla intermedia que vincula las tablas principales de cada entidad, permitiendo que se establezcan las relaciones necesarias y que se realicen consultas de manera sencilla y eficiente. Además, al utilizar una tabla unitaria, se evita la redundancia de datos y se asegura la integridad de la información, lo que resulta fundamental para garantizar la calidad y la consistencia de los datos en cualquier sistema de gestión de bases de datos.

1.1 EJEMPLO:

```
--tabla producto
create table Sucursal_Producto (
    codigo_sucursal varchar(5),
    codigo_producto varchar(5),
    primary key (codigo_sucursal, codigo_producto),
    foreign key (codigo_sucursal) references Sucursal(codigo_sucursal),
    foreign key (codigo_producto) references Producto(codigo_producto)
)
```



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'dbo.Sucursal_Producto' table is selected, and the 'Keys' folder is expanded, showing three keys: 'PK_Sucursal_AE1CB25635A5', 'FK_Sucursal_codig_590634', and 'FK_Sucursal_codig_59FA51'. On the right, the SQL script for creating the table is shown, matching the code in the example above. The script defines a primary key on 'codigo_sucursal' and 'codigo_producto', and two foreign keys: 'FK_Sucursal_codig_590634' referencing 'Sucursal(codigo_sucursal)' and 'FK_Sucursal_codig_59FA51' referencing 'Producto(codigo_producto)'.

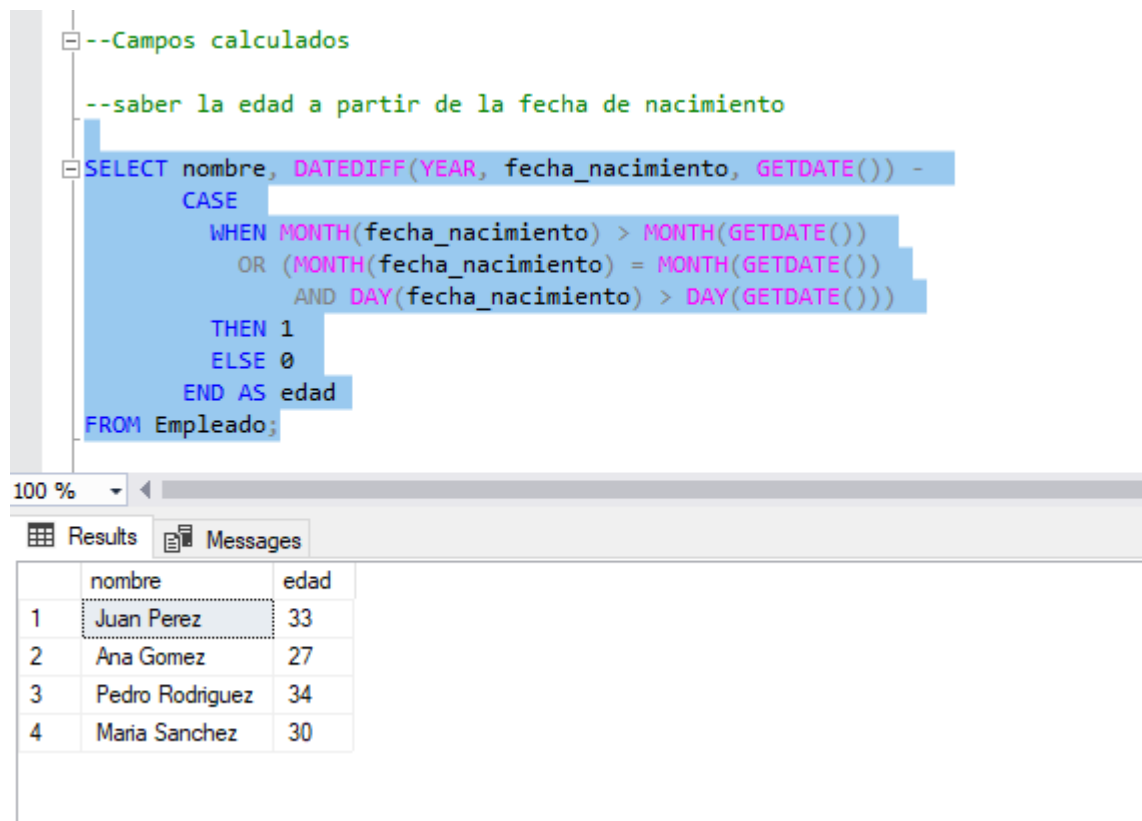
2. Campo Calculado

Los campos calculados proporcionan la capacidad de definir campos en un objeto cuyo valor se calcula a partir de los valores de otros campos. Estos otros campos pueden existir en el mismo objeto o en otro objeto relacionado.

Estos cálculos estarán disponibles en la aplicación así como para fines de creación de informes.

Los campos calculados te permiten crear métricas y dimensiones a partir de los datos. Con ellos puedes ampliar y transformar la información procedente de tus fuentes de datos y ver los resultados en los informes.

2.1 EJEMPLO:



```
--Campos calculados
--saber la edad a partir de la fecha de nacimiento
SELECT nombre, DATEDIFF(YEAR, fecha_nacimiento, GETDATE()) -
CASE
    WHEN MONTH(fecha_nacimiento) > MONTH(GETDATE())
    OR (MONTH(fecha_nacimiento) = MONTH(GETDATE())
    AND DAY(fecha_nacimiento) > DAY(GETDATE()))
    THEN 1
    ELSE 0
END AS edad
FROM Empleado;
```

100 %

Results Messages

	nombre	edad
1	Juan Perez	33
2	Ana Gomez	27
3	Pedro Rodriguez	34
4	Maria Sanchez	30

--saber el descuento de 5% del seguro medico al salario

```
SELECT e.codigo_empleado, r.nombre, r.salario_base AS "Salario Base",  
       r.salario_base * 0.05 AS "Seguro Médico (5%)"  
FROM Empleado e  
JOIN Rol_Empleado r ON e.codigo_rol = r.codigo_rol;
```

100 %

Results Messages

	codigo_empleado	nombre	Salario Base	Seguro Médico (5%)
1	EMP01	Cajero	800.00	40.0000
2	EMP02	Reponedor	700.00	35.0000
3	EMP03	Supervisor	900.00	45.0000
4	EMP04	Limpieza	600.00	30.0000

-- saber cuanto lleva trabajando a partir de la fecha de contratacion

```
SELECT nombre, DATEDIFF(YEAR, fecha_de_contratacion, GETDATE()) -  
CASE  
    WHEN MONTH(fecha_de_contratacion) > MONTH(GETDATE())  
    OR (MONTH(fecha_de_contratacion) = MONTH(GETDATE())  
        AND DAY(fecha_de_contratacion) > DAY(GETDATE()))  
    THEN 1  
    ELSE 0  
END AS [Años contratado]  
FROM Empleado;
```

100 %

Results Messages

	nombre	Años contratado
1	Juan Perez	3
2	Ana Gomez	2
3	Pedro Rodriguez	4
4	Maria Sanchez	1

-- saber la cantidad de dinero generado por las ventas de un producto

```
select nombre, (precio_unitario * cantidad_vendida) as [Ganancia de venta] from Producto
```

100 %

Results Messages

	nombre	Ganancia de venta
1	Aroz	125.00
2	Leche	36.00
3	Pan	120.00
4	Came	50.00

```
--saber la ganacia estimada de productos en almacen
```

```
select nombre, (precio_unitario * cantidad_en_almacen) as [Ganacia estimada de productos en reserva] from Producto
```

Results		Messages
	nombre	Ganacia estimada de productos en reserva
1	Aroz	250.00
2	Leche	90.00
3	Pan	160.00
4	Came	150.00

```
-- saber el impuesto del 15% a pagar por todo lo vendido
```

```
SELECT SUM(precio_unitario * cantidad_vendida) * 0.15 AS "Impuesto a Pagar"
FROM Producto;
```

100 %

Results Messages

	Impuesto a Pagar
1	49.6500

```
-- saber el salari anual de cada rol
```

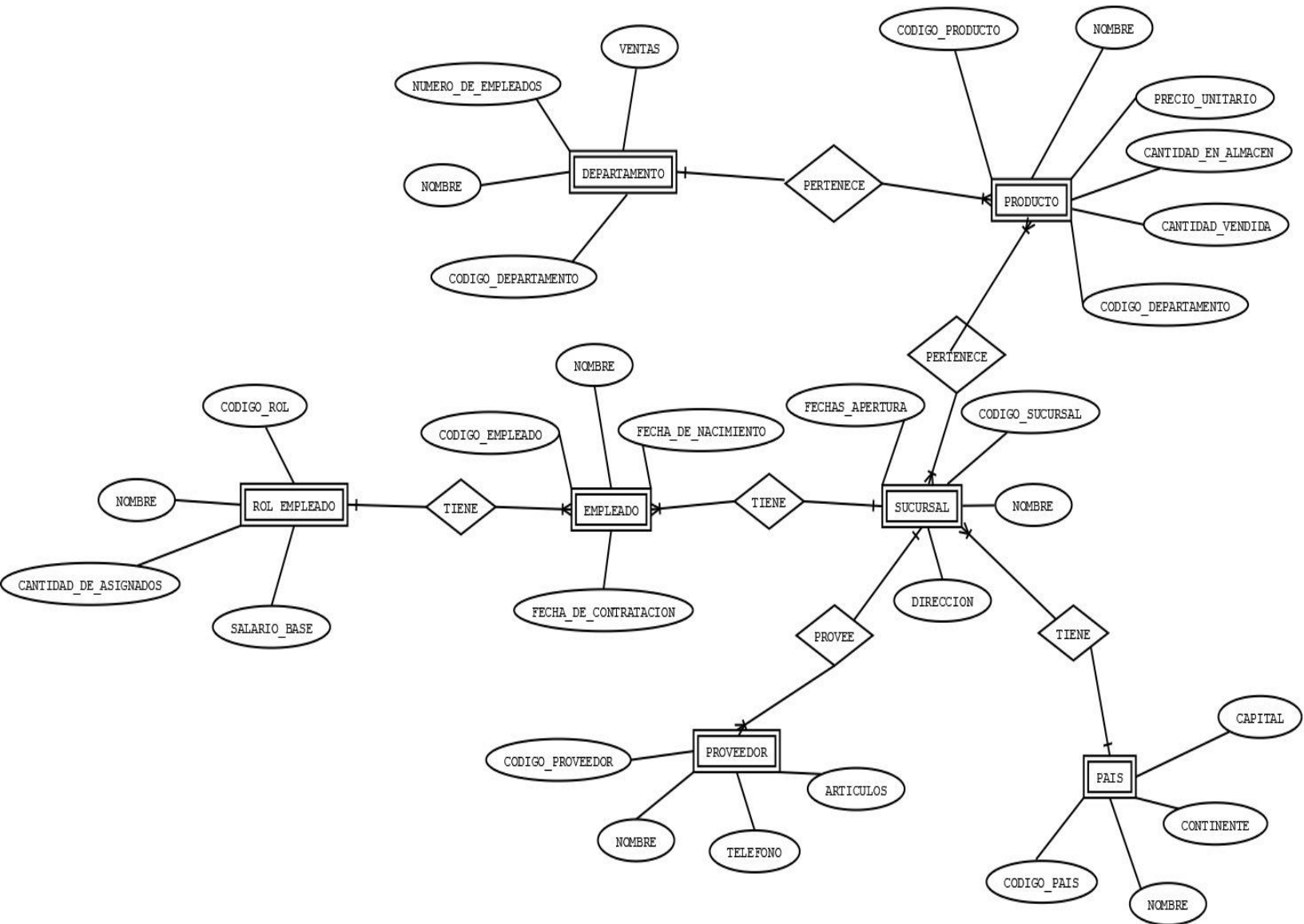
```
SELECT e.codigo_empleado as [Codigo empleado], r.nombre as Cargo, r.salario_base * 12 AS "Salario Anual"
FROM Empleado e
JOIN Rol_Empleado r ON e.codigo_rol = r.codigo_rol;
```

100 %

Results Messages

	Codigo empleado	Cargo	Salario Anual
1	EMP01	Cajero	9600.00
2	EMP02	Reponedor	8400.00
3	EMP03	Supervisor	10800.00
4	EMP04	Limpieza	7200.00

DIAGRAMA ENTIDAD RELACIÓN



DICCIONARIO DE DATOS

Tabla sucursal

PK	Código_sucursal	Varchar(5)	Código de la sucursal	NO
	Nombre	Char(50)	Nombre de la sucursal	NO
	Dirección	Char(50)	Dirección de la sucursal	NO
	Fecha_de_apertura	Date	Fecha en que se creo	NO
FK	Codigo_pais	Varchar(5)	País donde se ubica	NO

Tabla Producto

	Columna	Tipo de Dato	Descripción	null
PK	Código_producto	varchar(5)	Código del producto	NO
	Nombre	char(50)	Nombre del producto	NO
	Precio_unitario	decimal(10,2)	Precio por unidad del producto	NO
	Cantidad_en_almacen	Int	Cantidad disponible	NO
	Cantidad_vendida	Int	Cantidad vendida	NO
FK	Código_departamento	varchar(5)	Departamento al que pertenece	NO

Tabla Proveedor

	Columna	Tipo de Dato	Descripción	Null
PK	codigo_proveedor	varchar(5)	Código del proveedor	NO
	nombre	varchar(50)	Nombre del proveedor	NO
	telefono	varchar(20)	Teléfono de contacto	NO
	articulos	text	Artículos que vende	NO
FK	codigo_sucursal	varchar(5)	Código de la sucursal	NO

Tabla Rol_empleado

	Columna	Tipo de Dato	Descripción	null
PK	codigo_rol	varchar(5)	Código del rol del empleado	NO
	nombre	varchar(50)	Nombre del rol	NO
	cantidad_de_asignados	Int	Cantidad de empleados con ese rol	NO
	salario_base	decimal(10,2)	Salario base del rol	NO

Tabla Empleado

	Columna	Tipo de Dato	Descripción	null
PK	codigo_empleado	varchar(5)	Código del empleado	NO
	nombre	char(50)	Nombre del empleado	NO
	fecha_nacimiento	date	Nacimiento del empleado	NO
	fecha_de_contratacion	date	Fecha en que se contrato	NO
FK	codigo_rol	varchar(5)	Rol del empleado	NO
Fk	codigo_sucursal	varchar(5)	Sucursal donde trabaja	NO

Tabla Pais

	Columna	Tipo de Dato	Descripción	null
PK	codigo_pais	varchar(5)	Código del país	NO
	nombre	char(50)	Nombre del país	NO
	capital	char(50)	Capital del país	NO
	continente	char(50)	Continente del país	NO

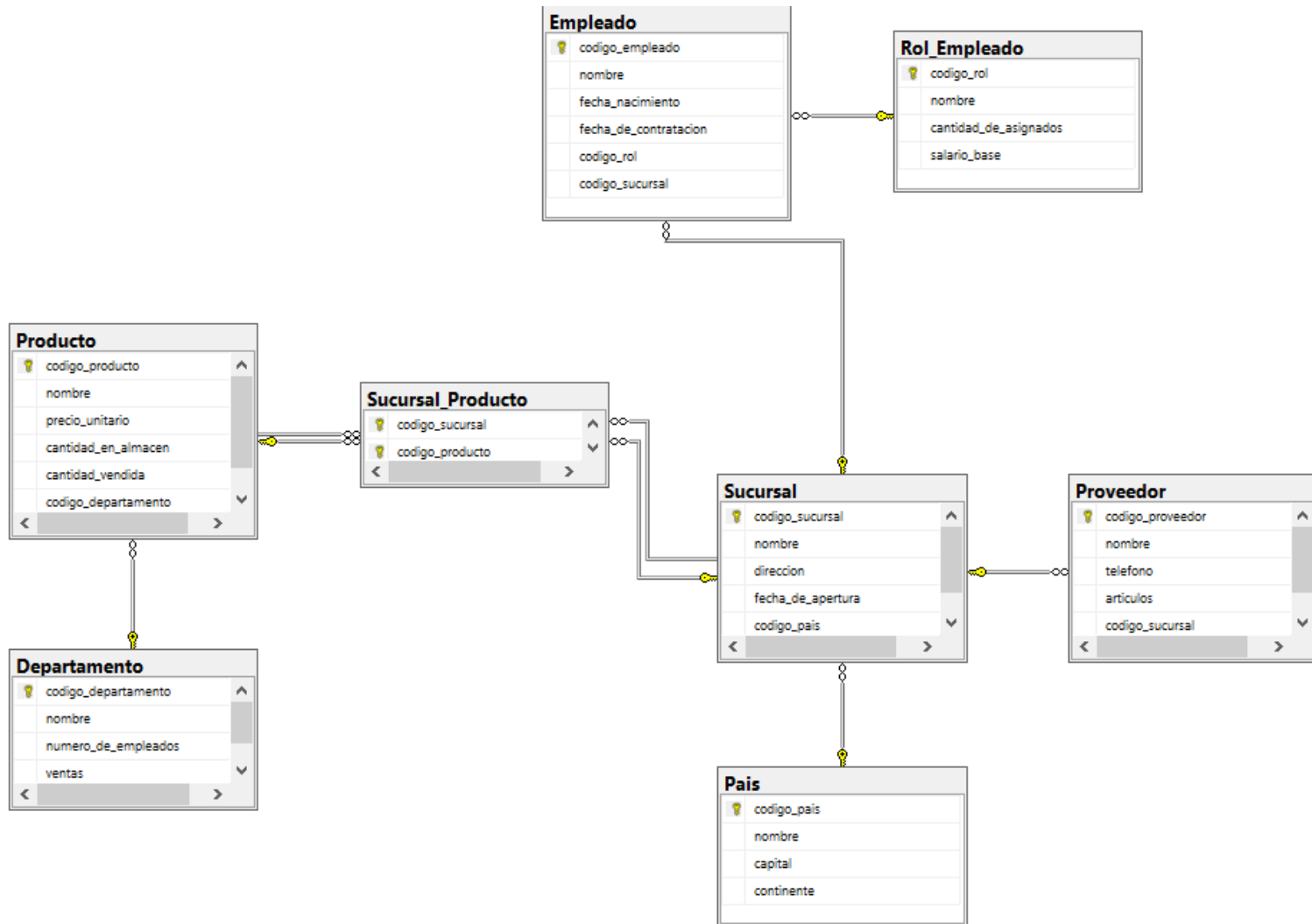
Tabla Departamento

	Columna	Tipo de Dato	Descripción	null
PK	codigo_departamento	varchar(5)	Código del departamento	NO
	nombre	char(50)	Nombre del departamento	NO
	numero_de_empleados	Int	Empleados del departamento	NO
	ventas	decimal(10,2)	Ventas del departamento	NO

Tabla Producto

	Columna	Tipo de Dato	Descripción	Null
PK	codigo_sucursal	varchar(5)	Código de la sucursal	NO
PK	codigo_producto	varchar(5)	Código del producto	NO

DIAGRAMA DE LA BASE DE DATOS



CONCLUSIÓN

Este trabajo ha sido creado con el propósito de poder documentar las partes más relevantes sobre el proyecto, con las funcionalidades de cada tienda, departamento, proveedor.

También, después de realizar el análisis de los datos y con base a los objetivos planteados para esta investigación se llegan a la conclusión, de que para un proyecto tan complejo era necesario, una base de datos, se realiza un control de inventario con los productos existentes, de igual manera el salario que cada empleado gana con su rol depende la asignación.

Además, posiblemente el proyecto, está en una etapa de auge con las ofertas comerciales, a sus preferencias en cada momento, proporcionándole una respuesta más certera y a medida de sus necesidades.