

Base de datos propuesta 3er parcial:

Jesse Isaac Guerrero Jiménez

Luis Mario Mauricio Jiménez

Omar Reyes Morales

Universidad Politécnica de Tulancingo, Calle Ingenierías #100, Huapalcalco,

Tulancingo

Asignatura: Base de datos

Nombre del profesor: Víctor Hugo Fernández Cruz

Tulancingo Hidalgo 11/02/2024

Contenido

Enunciado.....	3
Lista de entidades según enunciado de requerimientos	3
Lista de atributos	3
Relaciones	4
Grafos binarios.....	4
Grafo sin redundancias.....	4
Matriz de relación	5
Primera propuesta de diagrama MER (Microsoft Visio)	6
Diagrama MERE o MER.....	7
Diccionario de Datos	8
Esquema transición	9
Diagrama ER utilizando workbench.....	11
Descripción de las 3 formas normales	12
Primera forma normal:	12
Segunda forma normal:.....	13
Tercera forma normal:	14
Codigo	15
Datos de registro	17
Joins	23
REFERENCIAS.....	24

Enunciado

El restaurante “La Brava” es un establecimiento de comida mexicana, ubicado en la Ciudad de México, por lo cual ha experimentado un incremento en sus ventas, por lo tanto necesita gestionar su registro en el cual se tiene un inventario de cada producto (gestionado por varios empleados); además se lleva a cabo un registro de los pedidos a los proveedores de nuevos productos, que se verán reflejados cuando se agreguen al inventario.

Lista de entidades según enunciado de requerimientos

- REGISTRO
- PRODUCTO
- EMPLEADO
- PEDIDO
- PROVEEDOR

Lista de atributos

- REGISTRO: ID_Registro, Cantidad, Fecha del Registro.
- PRODUCTO: Id_Producto, Nombre del Producto, Unidad de Medida, Precio.
- EMPLEADO: Id_Empleado, Nombre(s), Apellido, Núm. Teléfono, Fecha de contratación, Usuario, contraseña.
- PEDIDO: ID_Pedido, Fecha de entrega estimada, Fecha de Pedido, Estado del Pedido.
- PROVEEDOR: ID_Proveedor, Nombre de la Empresa, RFC, Dirección, Correo electrónico, Teléfono.

Relaciones

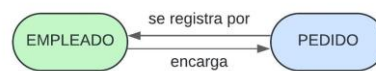
-EMPLEADO-----PEDIDO

-PROVEEDOR-----PEDIDO

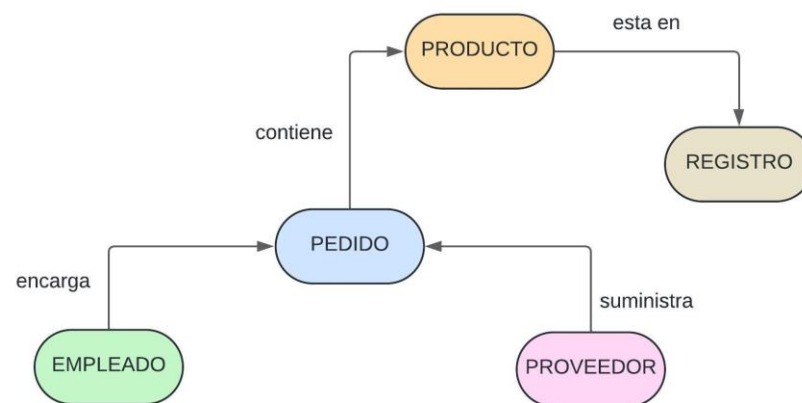
-PRODUCTO-----PEDIDO

-PRODUCTO-----REGISTRO

Grafos binarios



Grafo sin redundancias



Matriz de relación

	EMPLEADO	PROVEEDOR	PRODUCTO	PEDIDO	REGISTRO
EMPLEADO				X	
PROVEEDOR				X	
PRODUCTO				X	X
PEDIDO	X	X	X		
REGISTRO			X		

Primera propuesta de diagrama MER (Microsoft Visio)

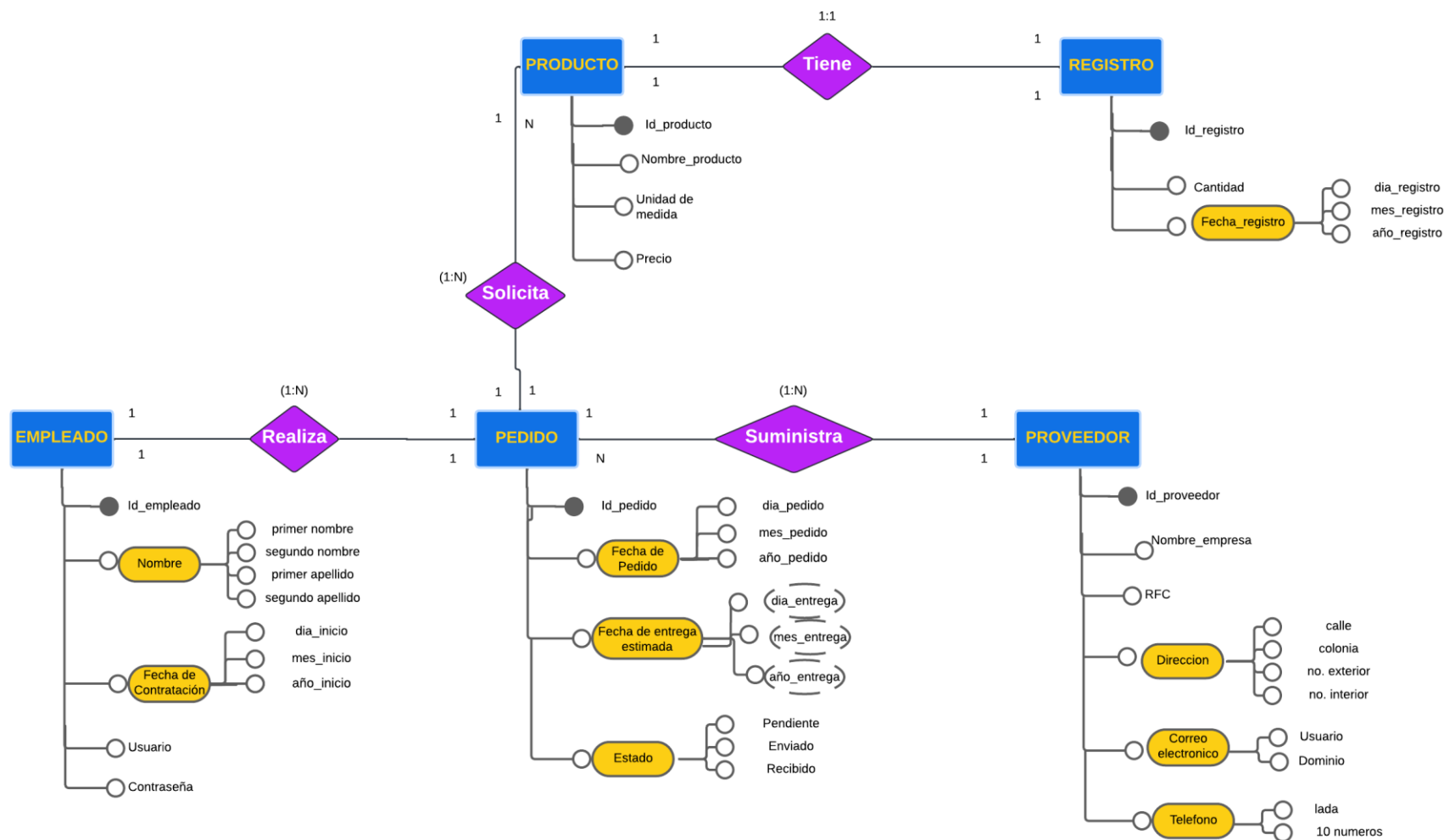
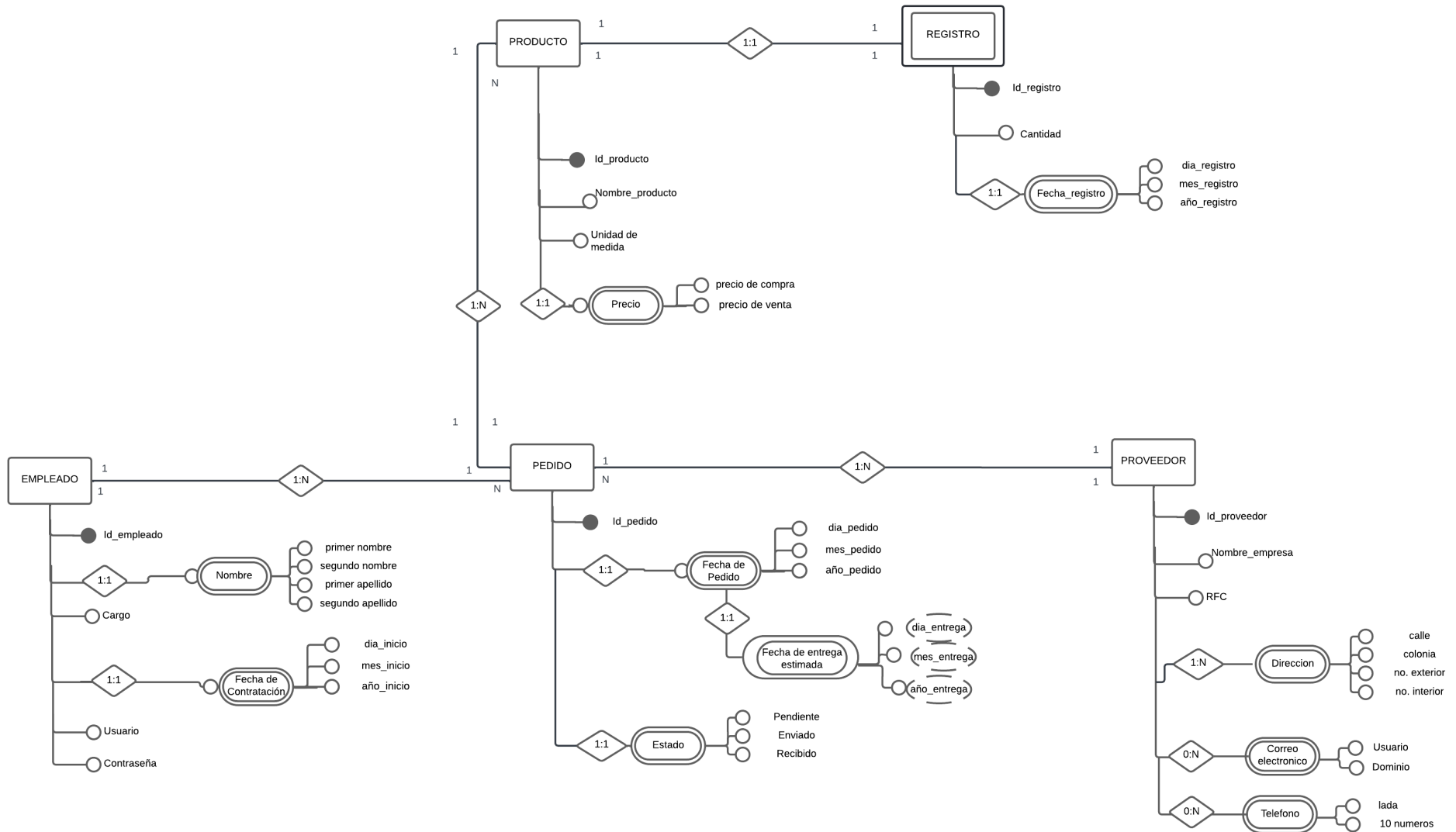


Diagrama MERE o MER



Diccionario de Datos

INFORMACIÓN CORRESPONDIENTE A LA BASE DE DATOS										
Inicio del formato										
Nombre de la base de datos:		ALMACEN		Preparado por:		guerrero jimenez jesse isaac			Fecha de preparación:	
1. Identificación del archivo de datos				2. Reglas de validación						
Nombre del Archivo	Descripción del Archivo	Nombre del Campo	Descripción del Campo	Llave Primaria	Llave Foránea	Campo obligatorio	Dominio o lista de valores	Tipo de datos	Longitud del campo	Regla de validación (en lenguaje natural)
				[Indicar si el campo hace parte de la llave primaria] [SI/NO]	[Si el campo hace parte de una llave foránea, indique el archivo y el campo de referencia]	[SI/NO]	[Liste los valores válidos, patrón o rango para el campo]	[Numérico, Cadena, Carácter, Entero, otro]	[Según el tipo de dato indicar la longitud del campo]	[Enumere claramente cada regla que se debe aplicar a la Variable]
EMPLEADO	Contiene información sobre los empleados de la empresa.	id_empleado	Identificador único del empleado.	SI	NO	SI	N/A	Entero	10	Debe ser un número entero positivo y único.
		cargo	Cargo del empleado.	NO	NO	SI	N/A	Cadena	Variable	-
		usuario	Usuario para acceder al sistema.	NO	NO	SI	N/A	Cadena	Variable	Debe ser único y no nulo.
		contraseña	Contraseña para acceder al sistema.	NO	NO	SI	N/A	Cadena	Variable	Debe tener un tamaño mínimo de 6 caracteres.
NOMBRE_EMPLEADO	Contiene información sobre los nombres de los empleados.	id_empleado	Identificador único del empleado.	NO	SI	SI	Referencia a tabla EMPLEADO.	Entero	10	Debe ser un número entero positivo.
		nombre1	Primer nombre del empleado.	NO	NO	SI	N/A	Cadena	Variable	No se permite que esté vacío.
		nombre2	Segundo nombre del empleado.	NO	NO	NO	N/A	Cadena	Variable	No se permite que esté vacío.
		apellidoM	Apellido materno del empleado.	NO	NO	SI	N/A	Cadena	Variable	No se permite que esté vacío.
FECHA_CONTRATACION_EMPLEADO	Contiene información sobre la fecha de contratación de los empleados.	id_empleado	Identificador único del empleado.	NO	SI	SI	Referencia a tabla EMPLEADO.	Entero	10	Debe ser un número entero positivo.
		dia_inicio	Día de inicio del contrato.	NO	NO	SI	1-31	Entero	2	Debe ser un número entre 1 y 31.
		mes_inicio	Mes de inicio del contrato.	NO	NO	SI	1-12	Entero	2	Debe ser un número entre 1 y 12.
		año_inicio	Año de inicio del contrato.	NO	NO	SI	1900-9999	Entero	4	Debe ser un año entre 1900 y 9999.
PROVEEDOR	Contiene información sobre los proveedores de la empresa.	id_proveedor	Identificador único del proveedor.	SI	NO	SI	N/A	Entero	10	Debe ser un número entero positivo y único.
		nombre_empresa	Nombre de la empresa proveedora.	NO	NO	SI	N/A	Cadena	Variable	No se permite que esté vacío.
		RFC	Registro Federal de Contribuyentes.	NO	NO	SI	N/A	Cadena	Variable	Debe ser un RFC válido.
		id_proveedor	Identificador único del proveedor.	NO	SI	SI	Referencia a tabla	Entero	10	Debe ser un número entero positivo.
DIRECCIÓN_PROVEEDOR	Contiene información sobre la dirección de los proveedores.	calle	Calle de la dirección.	NO	NO	SI	N/A	Cadena	Variable	No se permite que esté vacío.
		colonia	Colonia de la dirección.	NO	NO	SI	N/A	Cadena	Variable	No se permite que esté vacío.
		no_exterior	Número exterior de la dirección.	NO	NO	NO	N/A	Cadena	Variable	Debe ser un número o alfanumérico válido.
		no_interior	Número interior de la dirección.	NO	NO	NO	N/A	Cadena	Variable	Opcional.
CORREO_PROVEEDOR	Contiene información sobre los correos de los proveedores.	id_proveedor	Identificador único del proveedor.	NO	SI	SI	Referencia a tabla	Entero	10	Debe ser un número entero positivo.
		usuario	Usuario del correo.	NO	NO	SI	N/A	Cadena	Variable	-
		dominio	Dominio del correo.	NO	NO	SI	N/A	Cadena	Variable	Debe ser un dominio de correo válido.
		id_proveedor	Identificador único del proveedor.	NO	SI	SI	Referencia a tabla	Entero	10	Debe ser un número entero positivo.
TELEFONO_PROVEEDOR	Contiene información sobre los números de teléfono de los proveedores.	lado	Lado del número telefónico.	NO	NO	SI	N/A	Entero	3	Debe ser un número entero positivo.
		id_proveedor	Identificador único del proveedor.	NO	SI	SI	Referencia a tabla	Entero	10	Debe ser un número entero positivo.
		10_numeros	Número de teléfono (10 dígitos)	NO	NO	SI	N/A	Entero	10	Debe ser un número entero de 10 dígitos.
		id_pedido	Identificador único del pedido.	SI	NO	SI	N/A	Entero	10	Debe ser un número entero positivo y único.
PEDIDO	Contiene información sobre los pedidos.	id_pedido	Identificador único del pedido.	NO	SI	SI	Referencia a tabla PEDIDO.	Entero	10	Debe ser un número entero positivo.
		dia_pedido	Día del pedido.	NO	NO	SI	1-31	Entero	2	Debe ser un número entre 1 y 31.
		mes_pedido	Mes del pedido.	NO	NO	SI	1-12	Entero	2	Debe ser un número entre 1 y 12.
		año_pedido	Año del pedido.	NO	NO	SI	1900-9999	Entero	4	Debe ser un año entre 1900 y 9999.
FECHA_ENTREGA	Contiene información sobre la fecha de entrega de los pedidos.	id_pedido	Identificador único del pedido.	NO	SI	SI	Referencia a tabla PEDIDO.	Entero	10	Debe ser un número entero positivo.
		dia_entrega	Día de la entrega del pedido.	NO	NO	SI	1-31	Entero	2	Debe ser un número entre 1 y 31.
		mes_entrega	Mes de la entrega del pedido.	NO	NO	SI	1-12	Entero	2	Debe ser un número entre 1 y 12.
		año_entrega	Año de la entrega del pedido.	NO	NO	SI	1900-9999	Entero	4	Debe ser un año entre 1900 y 9999.
ESTADO_ENTREGA	Contiene información sobre el estado de los pedidos.	id_pedido	Identificador único del pedido.	NO	SI	SI	Referencia a tabla PEDIDO.	Entero	10	Debe ser un número entero positivo.
		pendiente	Estado de pendiente.	NO	NO	SI	N/A	Bool	Variable	-
		enviado	Estado de enviado.	NO	NO	SI	N/A	Bool	Variable	-
		recibido	Estado de recibido.	NO	NO	SI	N/A	Bool	Variable	-
REGISTRO	Contiene información sobre los registros realizados.	id_registro	Identificador único del registro.	SI	NO	SI	N/A	Entero	10	Debe ser un número entero positivo y único.
		cantidad	Cantidad del registro.	NO	NO	SI	N/A	Entero	10	Debe ser un número entero positivo.
FECHA_REGISTRO	Contiene información sobre la fecha de los registros.	id_registro	Identificador único del registro.	NO	SI	SI	Referencia a tabla	Entero	10	Debe ser un número entero positivo.
		dia_registro	Día del registro.	NO	NO	SI	1-31	Entero	2	Debe ser un número entre 1 y 31.
		mes_registro	Mes del registro.	NO	NO	SI	1-12	Entero	2	Debe ser un número entre 1 y 12.
		año_registro	Año del registro.	NO	NO	SI	1900-9999	Entero	4	Debe ser un año entre 1900 y 9999.
PRODUCTO	Contiene información sobre los productos de la empresa.	id_producto	Identificador único del producto.	SI	NO	SI	N/A	Entero	10	Debe ser un número entero positivo y único.
		nombre_producto	Nombre del producto.	NO	NO	SI	N/A	Cadena	Variable	No se permite que esté vacío.
		unidad_medida	Unidad de medida del producto.	NO	NO	SI	N/A	Cadena	Variable	No se permite que esté vacío.
PRECIO	Contiene información sobre los precios de los productos.	id_producto	Identificador único del producto.	NO	SI	SI	Referencia a tabla	Entero	10	Debe ser un número entero positivo.
		precio_compra	Precio de compra del producto.	NO	NO	SI	N/A	Decimal	Variable	No se permite que esté vacío.
		precio_venta	Precio de venta del producto.	NO	NO	SI	N/A	Decimal	Variable	No se permite que esté vacío.
				NO	NO	SI	N/A	Decimal	Variable	No se permite que esté vacío.

Esquema transición

EMPLEADO (id_empleado, cargo, usuario, contraseña) EMPLEADO.id_empleado es una clave primaria.

NOMBRE (id_empleado, nombre1, nombre2, apellidoM, apellidoP) NOMBRE.id_empleado es una clave ajena que hace referencia a EMPLEADO.

FECHA_CONTRATACION (id_empleado, dia_inicio, mes_inicio, año_inicio)

FECHA_CONTRATACION.id_empleado es una clave ajena que hace referencia a EMPLEADO.

PROVEEDOR (id_proveedor, nombre_empresa, RFC) PROVEEDOR.id_proveedor es una clave primaria.

DIRECCIÓN (id_proveedor, calle, colonia, no.exterior, no.interior) DIRECCION.id_proveedor es una clave ajena que hace referencia a PROVEEDOR.

CORREO (id_proveedor, usuario, dominio) CORREO.id_proveedor es una clave ajena que hace referencia a PROVEEDOR.

TELEFONO (id_proveedor, lado, 10_numeros) TELEFONO.id_proveedor es una clave ajena que hace referencia a PROVEEDOR.

PEDIDO (id_pedido) PEDIDO.id_pedido es una clave primaria.

FECHA_PEDIDO (id_pedido, dia_pedido, mes_pedido, año_pedido) FECHA_PEDIDO.id_pedido es una clave ajena que hace referencia a PEDIDO.

FECHA_ENTREGA (id_pedido, dia_entrega, mes_entrega, año_entrega)

FECHA_ENTREGA.id_pedido es una clave ajena que hace referencia a PEDIDO.

ESTADO (id_pedido, pendiente, enviado, recibido) ESTADO.id_pedido es una clave ajena que hace referencia a PEDIDO.

REGISTRO (id_registro, cantidad) REGISTRO.id_registro es una clave primaria.

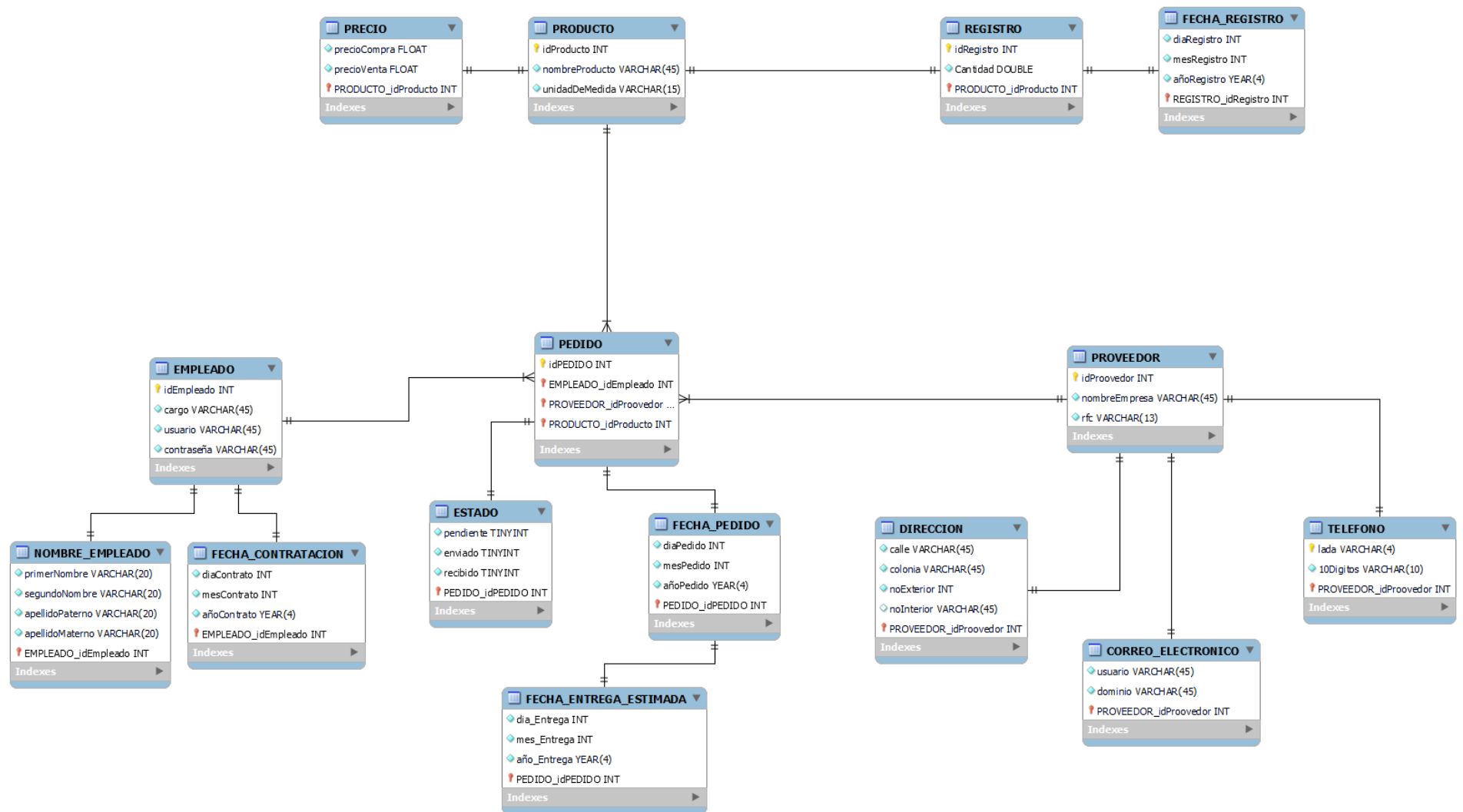
FECHA_REGISTRO (id_registro, dia_registro, mes_registro, año_registro)

FECHA_REGISTRO.id_registro es una clave ajena de REGISTRO

PRODUCTO(id_producto, nombre_producto, unidad_medida) PRODUCTO.id_producto es una clave primaria.

PRECIO (id_producto, precio_compra, precio_venta) PRECIO.id_producto es una clave ajena de PRODUCTO

Diagrama ER utilizando workbench



Descripción de las 3 formas normales

La normalización de bases de datos en simples palabras es organizar nuestro conjunto de datos para evitar redundancias y duplicaciones.

Es un proceso que busca eliminar la redundancia, es decir, la duplicación innecesaria de información. Para lograr esto utilizaremos las formas normales; recordando que las formas normales son un conjunto de reglas que sirven como guía para estructurar y organizar las bases de datos de manera eficiente. Su objetivo principal es evitar la redundancia, mejorar la integridad de los datos y facilitar su acceso y manejo.

En nuestra propuesta sobre la gestión del almacén del restaurante “La Brava” hemos aplicado estas formas normales de la siguiente manera:

Primera forma normal: En esta etapa, nos enfocamos en eliminar la repetición innecesaria de datos dentro de la base de datos. Para ello comenzamos identificando y desglosando los atributos compuestos de cada entidad:

- **PRODUCTO:** En este caso se desglosó el apartado del “precio” para indicar que el producto debe tener un precio de compra y un precio de venta, a diferencia de un único precio fijo.
- **REGISTRO:** Teníamos como un atributo la “fecha de registro” y se desglosó por su “día, mes y año de registro”.
- **EMPLEADO:** El atributo “Nombre” se tenía que desglosar en sus distintos elementos que son “primer nombre”, “segundo nombre” (con la posibilidad de que sea nulo si el empleado no tiene dos nombres), “apellido paterno” y “apellido materno”. De la misma forma se descompuso el atributo “fecha de contratación” para tener una mejor gestión del “día, mes y año de contratación”.

- PEDIDO: Se ha desglosado tanto la “fecha de pedido” como la de “entrega” en sus respectivos “día, mes y año”, además del atributo “estado” que servirá para conocer en qué punto de la entrega se encuentra el pedido (“pendiente, enviado y/o entregado”)
- PROVEEDOR: Se desgloso el atributo “dirección” considerando que al ser una empresa puede tener una o muchas sucursales y así solicitando la “calle,colonia, no. Exterior y no. Interior” (considerando la posibilidad de que puede o no tener no. Interior). También se aplica la primera forma normal al descomponer el atributo teléfono (“lada y 10_dígitos”) y el “correo electrónico” (“usuario y dominio”) y a su vez aclarando que estos últimos dos atributos compuestos tienen una cardinalidad de cero a muchos (0:N), ya que pueden no tenerlos o tener varios.

Segunda forma normal: En la segunda forma normal cada atributo no-clave debe depender de la clave primaria completa, no solo de una parte de ella. Una dependencia funcional es una relación entre dos atributos, donde un atributo (el determinante) determina el valor del otro (el dependiente).

Para que una tabla cumpla con la segunda forma normal debe cumplir con dos requisitos:

1. Que cumpla con la primera forma normal.
2. No tener dependencias parciales.

Una vez que se había cumplido la primera condición en nuestro modelo procedimos a verificar que efectivamente no existieran dependencias parciales en nuestras tablas.

En un principio el atributo de la FECHA DE ENTREGA estaba siendo dependiente de PEDIDO, pero este a su vez dependía de la FECHA DE PEDIDO para determinar los valores que le corresponden a este atributo, por lo que no se cumplía la segunda forma normal, así que hicimos que FECHA DE ENTREGA fuera una tabla aparte que dependiera únicamente de FECHA DE PEDIDO

Tercera forma normal: La tercera forma normal se basa en la segunda forma normal y busca eliminar las dependencias transitivas, que pueden generar inconsistencias en los datos.

Para que una tabla cumpla con la tercera forma normal, debe cumplir con dos requisitos:

1. Estar en la segunda forma normal, en otras palabras, no tener dependencias parciales.
2. No tener dependencias transitivas, es decir que un atributo no-clave no puede depender de otro atributo no-clave, a menos que este último dependa de la clave primaria completa.

Llegados a este punto lo que teníamos que hacer era verificar que no existieran dependencias transitivas en nuestro modelo, y dado que esto se cumplía no fue necesario aplicar más modificaciones

Codigo

```
create table Registro (
id_registro INT PRIMARY KEY,
cantidad INT);
```

```
create table Fecha_Registro (
id_registro INT primary key,
dia_registro int,
mes_registro int,
año_registro int,
foreign key (id_registro) references
Registro(id_registro)
);
```

```
create table Producto (
id_producto int primary key,
nombre_producto varchar(255),
unidad_medida varchar(50)
);
```

```
create table Precio (
id_producto int primary key,
precio_compra decimal(10, 2),
precio_venta decimal (10, 2),
foreign key(id_producto) references
Producto(id_producto)
);
```

```
create table Pedido (
id_pedido int primary key,
id_producto int,
cantidad int,
foreign key (id_producto) references
Producto(id_producto)
);
```

```
create table Fecha_pedido (
id_pedido int primary key,
dia_pedido int,
mes_pedido int,
año_pedido int,
foreign key (id_pedido) references
Pedido(id_pedido)
);
```

```
create table Fecha_entrega (
id_pedido int primary key,
dia_entrega int,
mes_entrega int,
año_entrega int,
foreign key (id_pedido) references Pedido
(id_pedido)
);
```

```

create table Estado (
id_estado int primary key,
estado_nombre varchar(50)
);

create table Empleado (
id_empleado int primary key,
cargo varchar (50),
usuario varchar (50),
contraseña varchar (50)
);

create table Nombre (
id_empleado int primary key,
primer_n varchar(50),
segundo_n varchar(50),
primer_ap varchar(50),
segundo_ap varchar(50),
foreign key(id_empleado) references
Empleado(id_empleado)
);

create table Fecha_contratacion (
id_empleado int primary key,
dia_inicio int,
mes_inicio int,
año_inicio int,
foreign key (id_empleado) references
Empleado (id_empleado)
);

```

```

create table Proveedor (
id_proveedor int primary key,
nombre_empresa varchar(100),
RFC varchar(50)
);

create table Direccion (
id_proveedor int,
calle varchar(100),
colonia varchar(100),
No_exterior varchar(20),
No_interior varchar(20),
foreign key (id_proveedor) references
Proveedor(id_proveedor)
);

create table Correo_electronico (
id_proveedor int,
usuario varchar (50),
dominio varchar (50),
foreign key (id_proveedor) references
Proveedor(id_proveedor)
);

create table Telefono (
id_proveedor int,
lada varchar(50),
dominio varchar(50),
foreign key (id_proveedor) references
Proveedor(id_proveedor)
);

```


Datos de registro

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| new |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> use new;
Database changed
```

Ilustración 1 mostrar las bases de datos, cambiando a la nueva base

```
mysql> show tables;
+-----+
| Tables_in_new |
+-----+
| correo_electronico |
| direccion |
| empleado |
| estado |
| fecha_contratacion |
| fecha_entrega |
| fecha_pedido |
| fecha_registro |
| nombre |
| pedido |
| precio |
| producto |
| proveedor |
| registro |
| telefono |
+-----+
15 rows in set (0.01 sec)
```

Ilustración 2 mostrando todas las tablas creadas

```
mysql> describe correo_electronico;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id_proveedor | int | YES | MUL | NULL | |
| usuario | varchar(50) | YES | | NULL | |
| dominio | varchar(50) | YES | | NULL | |
+-----+
3 rows in set (0.01 sec)
```

Ilustración 3 describe la tabla de correo_electronico

```
mysql> describe direccion;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id_proveedor | int | YES | MUL | NULL | |
| calle | varchar(100) | YES | | NULL | |
| colonia | varchar(100) | YES | | NULL | |
| No_exterior | varchar(20) | YES | | NULL | |
| No_interior | varchar(20) | YES | | NULL | |
+-----+
5 rows in set (0.01 sec)
```

Ilustración 4 describe la tabla dirección

```
mysql> describe empleado;
```

Field	Type	Null	Key	Default	Extra
id_empleado	int	NO	PRI	NULL	
cargo	varchar(50)	YES		NULL	
usuario	varchar(50)	YES		NULL	
contraseña	varchar(50)	YES		NULL	

4 rows in set (0.01 sec)

Ilustración 5 describe la tabla empleada

```
mysql> describe fecha_contratacion;
```

Field	Type	Null	Key	Default	Extra
id_empleado	int	NO	PRI	NULL	
día_inicio	int	YES		NULL	
mes_inicio	int	YES		NULL	
año_inicio	int	YES		NULL	

4 rows in set (0.01 sec)

Ilustración 6 describe la tabla fecha_contratacion

```
mysql> describe fecha_entrega;
```

Field	Type	Null	Key	Default	Extra
id_pedido	int	NO	PRI	NULL	
día_entrega	int	YES		NULL	
mes_entrega	int	YES		NULL	
año_entrega	int	YES		NULL	

4 rows in set (0.01 sec)

Ilustración 7 describe la tabla fecha_entrega

```
mysql> describe fecha_pedido;
```

Field	Type	Null	Key	Default	Extra
id_pedido	int	NO	PRI	NULL	
día_pedido	int	YES		NULL	
mes_pedido	int	YES		NULL	
año_pedido	int	YES		NULL	

4 rows in set (0.01 sec)

Ilustración 8 describe la fecha_pedido

```
mysql> describe fecha_registro;
```

Field	Type	Null	Key	Default	Extra
id_registro	int	NO	PRI	NULL	
día_registro	int	YES		NULL	
mes_registro	int	YES		NULL	
año_registro	int	YES		NULL	

4 rows in set (0.01 sec)

Ilustración 9 describe la tabla fecha_registro

```
mysql> describe nombre;
```

Field	Type	Null	Key	Default	Extra
id_empleado	int	NO	PRI	NULL	
primer_n	varchar(50)	YES		NULL	
segundo_n	varchar(50)	YES		NULL	
primer_ap	varchar(50)	YES		NULL	
segundo_ap	varchar(50)	YES		NULL	

5 rows in set (0.01 sec)

Ilustración 10 describe la tabla nombre

```
mysql> describe pedido;
```

Field	Type	Null	Key	Default	Extra
id_pedido	int	NO	PRI	NULL	
id_producto	int	YES	MUL	NULL	
cantidad	int	YES		NULL	

3 rows in set (0.01 sec)

Ilustración 11 describe la tabla pedido

```
mysql> describe precio;
```

Field	Type	Null	Key	Default	Extra
id_producto	int	NO	PRI	NULL	
precio_compra	decimal(10,2)	YES		NULL	
precio_venta	decimal(10,2)	YES		NULL	

3 rows in set (0.01 sec)

Ilustración 12 describe la tabla precio

```
mysql> describe producto;
```

Field	Type	Null	Key	Default	Extra
id_producto	int	NO	PRI	NULL	
nombre_producto	varchar(255)	YES		NULL	
unidad_medida	varchar(50)	YES		NULL	

3 rows in set (0.01 sec)

Ilustración 13 describe la tabla producto

```
mysql> describe proveedor;
```

Field	Type	Null	Key	Default	Extra
id_proveedor	int	NO	PRI	NULL	
nombre_empresa	varchar(100)	YES		NULL	
RFC	varchar(50)	YES		NULL	

3 rows in set (0.01 sec)

Ilustración 14 describe la tabla proveedor

```
mysql> describe registro;
```

Field	Type	Null	Key	Default	Extra
id_registro	int	NO	PRI	NULL	
cantidad	int	YES		NULL	

2 rows in set (0.00 sec)

Ilustración 15 describe la tabla registro

```
mysql> describe telefono;
```

Field	Type	Null	Key	Default	Extra
id_proveedor	int	YES	MUL	NULL	
lada	varchar(50)	YES		NULL	
dominio	varchar(50)	YES		NULL	

3 rows in set (0.01 sec)

Ilustración 16 describe la tabla teléfono

```
mysql> INSERT INTO Registro (id_registro, cantidad) VALUES (1, 10),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (2, 20),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (3, 15),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (4, 8),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (5, 12),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (6, 30),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (7, 5),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (8, 18),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (9, 25),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (10, 14),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (11, 22),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (12, 17),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (13, 9),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (14, 28),
-> INSERT INTO Registro (id_registro, cantidad) VALUES (15, 11);
```

Ilustración 17 código con datos ingresados a la tabla Registro

```
mysql> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (1, 15, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (2, 10, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (3, 5, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (23, 21, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (19, 20, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (23, 21, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (6, 22, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (7, 23, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (8, 24, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (9, 25, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (10, 26, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (11, 27, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (12, 28, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (13, 29, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (14, 30, 4, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (15, 1, 5, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (16, 2, 5, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (17, 3, 5, 2024),
-> INSERT INTO Fecha_registro (id_registro, dia_registro, mes_registro, año_registro) VALUES (18, 4, 5, 2024);
```

Ilustración 18 código con datos ingresados a la tabla Fecha_registro

```
mysql> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (1, 'Leche', 'Litro'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (2, 'Pan', 'Pieza'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (3, 'Huevos', 'Docena'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (4, 'Azúcar', 'Kilogramo'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (5, 'Café', 'Gramo'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (6, 'Arroz', 'Kilogramo'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (7, 'Aceite de oliva', 'Litro'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (8, 'Sal', 'Gramo'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (9, 'Harina de trigo', 'Kilogramo'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (10, 'Vinagre', 'Litro'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (11, 'Pasta', 'Gramo'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (12, 'Salsa de tomate', 'Mililitro'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (13, 'Miel', 'Kilogramo'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (14, 'Lechuga', 'Pieza'),
-> INSERT INTO Producto (id_producto, nombre_producto, unidad_medida) VALUES (15, 'Papas', 'Kilogramo');
```

Ilustración 19 código con datos ingresados a la tabla Producto

```
mysql> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (1, 10.50, 12.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (2, 2.00, 2.50),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (3, 15.00, 18.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (4, 15.00, 20.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (5, 5.00, 8.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (6, 10.00, 15.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (7, 25.00, 30.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (8, 3.00, 5.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (9, 12.00, 18.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (10, 8.00, 12.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (11, 4.00, 6.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (12, 6.00, 10.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (13, 20.00, 25.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (14, 2.50, 4.00),
-> INSERT INTO Precio (id_producto, precio_compra, precio_venta) VALUES (15, 10.00, 15.00);
```

Ilustración 20 código con datos ingresados a la tabla Precio

```
mysql> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (1, 1, 2),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (2, 2, 3),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (3, 3, 1),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (4, 4, 3),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (5, 5, 2),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (6, 6, 5),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (7, 7, 1),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (8, 8, 4),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (9, 9, 3),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (10, 10, 2),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (11, 11, 6),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (12, 12, 3),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (13, 13, 2),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (14, 14, 5),
-> INSERT INTO Pedido (id_pedido, id_producto, cantidad) VALUES (15, 15, 4);
```

Ilustración 21 código con datos ingresados a la tabla Pedido


```
mysql> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (1, 15, 4, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (2, 16, 4, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (3, 17, 4, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (4, 5, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (5, 6, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (6, 7, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (7, 8, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (8, 9, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (9, 10, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (10, 11, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (11, 12, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (12, 13, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (13, 14, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (14, 15, 5, 2024),
-> INSERT INTO Fecha_pedido (id_pedido, dia_pedido, mes_pedido, año_pedido) VALUES (15, 16, 5, 2024);_
```

Ilustración 22 código con datos ingresados a la tabla Fecha_pedido

```
mysql> INSERT INTO Estado (id_estado, estado_nombre) VALUES (1, 'Pendiente'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (2, 'Enviado'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (3, 'Recibido'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (4, 'Preparando'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (5, 'En tránsito'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (6, 'Entregado'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (7, 'Devuelto'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (8, 'Cancelado'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (9, 'Esperando confirmación'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (10, 'Listo para enviar'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (11, 'En espera'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (12, 'Procesando'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (13, 'Recibido con daños'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (14, 'Recibido sin problemas'),
-> INSERT INTO Estado (id_estado, estado_nombre) VALUES (15, 'En revisión');
```

Ilustración 23 código con datos ingresados a la tabla Estado

```
mysql> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (1, 'Gerente', 'gerentel', 'clave123'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (2, 'Cajero', 'cajero1', 'clave456'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (3, 'Almacenero', 'almacen1', 'clave789'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (4, 'Vendedor', 'vendedor1', 'contraseña123'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (5, 'Recepcionista', 'recepcion1', 'clave456'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (6, 'Gerente de almacén', 'gerente_almacen', 'password789'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (7, 'Asistente de ventas', 'asistente_ventas', 'clave987'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (8, 'Gerente de tienda', 'gerente_tienda', 'password654'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (9, 'Contador', 'contador1', 'clave321'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (10, 'Recursos humanos', 'rrhh1', 'password1234'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (11, 'Supervisor de ventas', 'supervisor_ventas', 'clave567'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (12, 'Analista de datos', 'analista_datos', 'password890'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (13, 'Técnico de soporte', 'soporte_tecnico', 'clave246'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (14, 'Encargado de compras', 'compras1', 'password135'),
-> INSERT INTO Empleado (id_empleado, cargo, usuario, contraseña) VALUES (15, 'Marketing', 'marketing1', 'clave579');
```

Ilustración 24 código con datos ingresados a la tabla Empleado

Joins

```
mysql> SELECT Pedido.id_pedido, Pedido.cantidad, Producto.nombre_producto, Producto.unidad_medida
-> FROM Pedido
-> INNER JOIN Producto ON Pedido.id_producto = Producto.id_producto,
->
-> SELECT Empleado.id_empleado, Nombre.primer_nombre, Nombre.segundo_nombre, Nombre.primer_apellido, Nombre.segundo_apellido
-> FROM Empleado
-> INNER JOIN Nombre ON Empleado.id_empleado = Nombre.id_empleado,
->
-> SELECT Registro.id_registro, Registro.cantidad, Fecha_registro.dia_registro, Fecha_registro.mes_registro, Fecha_registro.anio_registro
egistro
-> FROM Registro
-> INNER JOIN Fecha_registro ON Registro.id_registro = Fecha_registro.id_registro,
->
-> SELECT Pedido.id_pedido, Pedido.cantidad, Fecha_pedido.dia_pedido, Fecha_pedido.mes_pedido, Fecha_pedido.anio_pedido
-> FROM Pedido
-> INNER JOIN Fecha_pedido ON Pedido.id_pedido = Fecha_pedido.id_pedido,
->
-> SELECT Pedido.id_pedido, Pedido.cantidad, Estado.estado_nombre
-> FROM Pedido
-> INNER JOIN Estado ON Pedido.id_estado = Estado.id_estado;_
```

Ilustración 25 código con datos ocupando “join”

REFERENCIAS

Helencu. (n.d.). *Descripción de la normalización de la base de datos - Microsoft 365 Apps*.

Microsoft Learn. <https://learn.microsoft.com/es-es/office/troubleshoot/access/database-normalization-description>

Pérez, S. D. (2023, March 23). Gestor de Base de Datos: Qué es, funcionalidades y ejemplos. Intelequia. <https://intelequia.com/es/blog/post/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos>

Inc., L. (2010, January 18). Qué es un modelo de base de datos. <https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos#:~:text=Un%20modelo%20de%20base%20de%20datos%20muestra%20la%20estructura%20l%C3%B3gica,c%C3%B3mo%20se%20accede%20a%20ellos>.

Torrejón, H. C. (2022, November 24). Cómo realizar la Normalización de bases de Datos y Por Qué. OpenWebinars.net. <https://openwebinars.net/blog/como-realizar-la-normalizacion-de-bases-de-datos-y-por-que/>