

INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

BASES DE DATOS

GRUPO: 3CV5

Profesor: Gabriel Hurtado Avilés

“Práctica 3 Modelo Entidad-Relación (Extendido) al modelo relacional”

Alumno:

- **González González Erick Emiliano**
- **De La Rosa Hernández Tania**

Fecha de Entrega: 21/10/25

Ejercicio 1: Análisis del Modelo EER Original

El proyecto *Sillas y Mesas Hernández* modela un sistema de gestión para la renta de mobiliario para eventos sociales. El modelo EER inicial representa los elementos fundamentales del negocio, las relaciones entre ellos y la especialización de ciertos tipos de entidades.

1.1 Revisión del Modelo

- Entidades fuertes:
Cliente, Artículo, Pedido, Pago y Paquete.
- Entidades débiles:
DetallePedido (dependiente de Pedido y Artículo).
- Relaciones principales:
 - Cliente realiza uno o varios Pedidos (1:N).
 - Cada Pedido puede contener múltiples Artículos (N:M).
 - Un Pedido puede estar compuesto por uno o varios Paquetes (N:M).
 - Cada Pedido genera uno o más Pagos (1:N).
- Jerarquías ISA:
 - *Cliente* se especializa en *ClienteParticular* y *ClienteEmpresa*.
 - *Artículo* se especializa en *Silla*, *Mesa* y *Accesorio*.
En ambos casos se trata de jerarquías disjuntas y parciales, pues cada instancia pertenece a un solo subtipo y no todos los clientes o artículos necesariamente deben pertenecer a uno.
- Atributos multivaluados:
 - *Teléfono*, asociado al cliente.
- Atributos compuestos o derivados:
Ninguno identificado.

1.2 Preparación para la Transformación

Durante el proceso de transformación, se generaron tablas adicionales para representar las relaciones N:M mediante entidades intermedias. Las jerarquías se resolvieron aplicando la Estrategia A (múltiples tablas con relación 1:1), de forma que cada subtipo tiene su propia tabla enlazada a su supertipo.

Ejercicio 2: Reglas de Transformación Básicas

2.1 Entidades Fuertes → Tablas

Cada entidad fuerte del modelo EER se convierte directamente en una tabla con una clave primaria:

- Cliente(idCliente PK, nombre, direccion, correo)
- Artículo(idArticulo PK, nombre, cantidadTotal, estado, costoRenta)
- Pedido(idPedido PK, fechaEvento, fechaEntrega, fechaDevolucion, montoTotal, idCliente FK)
- Pago(idPago PK, fechaPago, monto, estadoPago, idPedido FK)
- Paquete(idPaquete PK, nombre, precioEspecial)

2.2 Atributos Compuestos

No hay atributos compuestos.

2.3 Atributos Multivaluados

El atributo *Teléfono* del cliente se representa como una tabla independiente: Telefono(idTelefono PK, idCliente FK, numero).

Ejercicio 3: Transformación de Relaciones

3.1 Relaciones 1:N

- Cliente → Pedido
La clave primaria de Cliente (idCliente) pasa como FK en Pedido.
- Pedido → Pago
La clave primaria de Pedido (idPedido) pasa como FK en Pago.

3.2 Relaciones N:M

Se crean tablas intermedias con claves foráneas compuestas:

- DetallePedido(idPedido FK, idArticulo FK, cantidad)
- PaqueteArticulo(idPaquete FK, idArticulo FK, cantidad)
- PedidoPaquete(idPedido FK, idPaquete FK, cantidad)

3.3 Relaciones 1:1 (Generalización/Especialización)

- Cliente–ClienteParticular / Cliente–ClienteEmpresa
 - Artículo–Silla / Artículo–Mesa / Artículo–Accesorio
- Cada subtipo conserva la clave del supertipo como su PK y FK, garantizando la correspondencia uno a uno.

Ejercicio 4: Transformación de Entidades Débiles

DetallePedido se define como una entidad débil porque depende de Pedido y Artículo para su existencia.

Su clave primaria es compuesta (idPedido, idArticulo), y ambas son llaves foráneas.

Ejercicio 5: Transformación de Jerarquías de Generalización/Especialización

Cliente → ClienteParticular / ClienteEmpresa

Se aplica la estrategia de múltiples tablas (supertipo + subtipos), con relación 1:1.

Justificación: evita valores nulos y refleja de manera precisa las diferencias entre tipos de clientes.

Restricción: *disjunta y parcial*.

Artículo → Silla / Mesa / Accesorio

Cada subtipo tiene su tabla propia y una relación 1:1 con Artículo.

Justificación: Esto evita redundancia y facilita la extensión futura de nuevos tipos de artículos.

Ejercicio 6: Esquema Relacional Completo

```

CLIENTE (
    idCliente INT PK,
    nombre VARCHAR(100),
    direccion VARCHAR(150),
    correo VARCHAR(100)
)
CLIENTE_PARTICULAR (
    idCliente INT PK, FK → CLIENTE(idCliente),
    fechaNacimiento DATE,
    CURP VARCHAR(20)
)
CLIENTE_EMPRESA (
    idCliente INT PK, FK → CLIENTE(idCliente),
    razonSocial VARCHAR(100),
    RFC VARCHAR(15),
    contactoEmpresa VARCHAR(100)
)
TELEFONO (
    idTelefono INT PK,
    idCliente INT FK → CLIENTE(idCliente),
    numero VARCHAR(15)
)
ARTICULO (
    idArticulo INT PK,
    nombre VARCHAR(100),
    cantidadTotal INT,
    estado VARCHAR(20),
    costoRenta DECIMAL(10,2)
)
SILLA (
    idArticulo INT PK, FK → ARTICULO(idArticulo),
    tipoSilla VARCHAR(50),
    material VARCHAR(50)
)
MESA (
    idArticulo INT PK, FK → ARTICULO(idArticulo),
    forma VARCHAR(50),
    capacidadPersonas INT,
    tamaño VARCHAR(50)
)
ACCESORIO (
    idArticulo INT PK, FK → ARTICULO(idArticulo),
    descripcion VARCHAR(100),
    fragilidad VARCHAR(30)
)
PEDIDO (
    idPedido INT PK,
    fechaEvento DATE,
    fechaEntrega DATE,
    fechaDevolucion DATE,
    montoTotal DECIMAL(10,2),
    idCliente INT FK → CLIENTE(idCliente)
)
DETALLE_PEDIDO (
    idPedido INT FK → PEDIDO(idPedido),
    idArticulo INT FK → ARTICULO(idArticulo),
    cantidad INT,
    PRIMARY KEY (idPedido, idArticulo)
)
PAGO (
    idPago INT PK,
    fechaPago DATE,
    monto DECIMAL(10,2),
    estadoPago VARCHAR(20),
    idPedido INT FK → PEDIDO(idPedido)
)
PAQUETE (
    idPaquete INT PK,
    nombre VARCHAR(100),
    precioEspecial DECIMAL(10,2)
)
PAQUETE_ARTICULO (
    idPaquete INT FK → PAQUETE(idPaquete),
    idArticulo INT FK → ARTICULO(idArticulo),
    cantidad INT,
    PRIMARY KEY (idPaquete, idArticulo)
)
PEDIDO_PAQUETE (
    idPedido INT FK → PEDIDO(idPedido),
    idPaquete INT FK → PAQUETE(idPaquete),
    cantidad INT,
    PRIMARY KEY (idPedido, idPaquete)
)

```

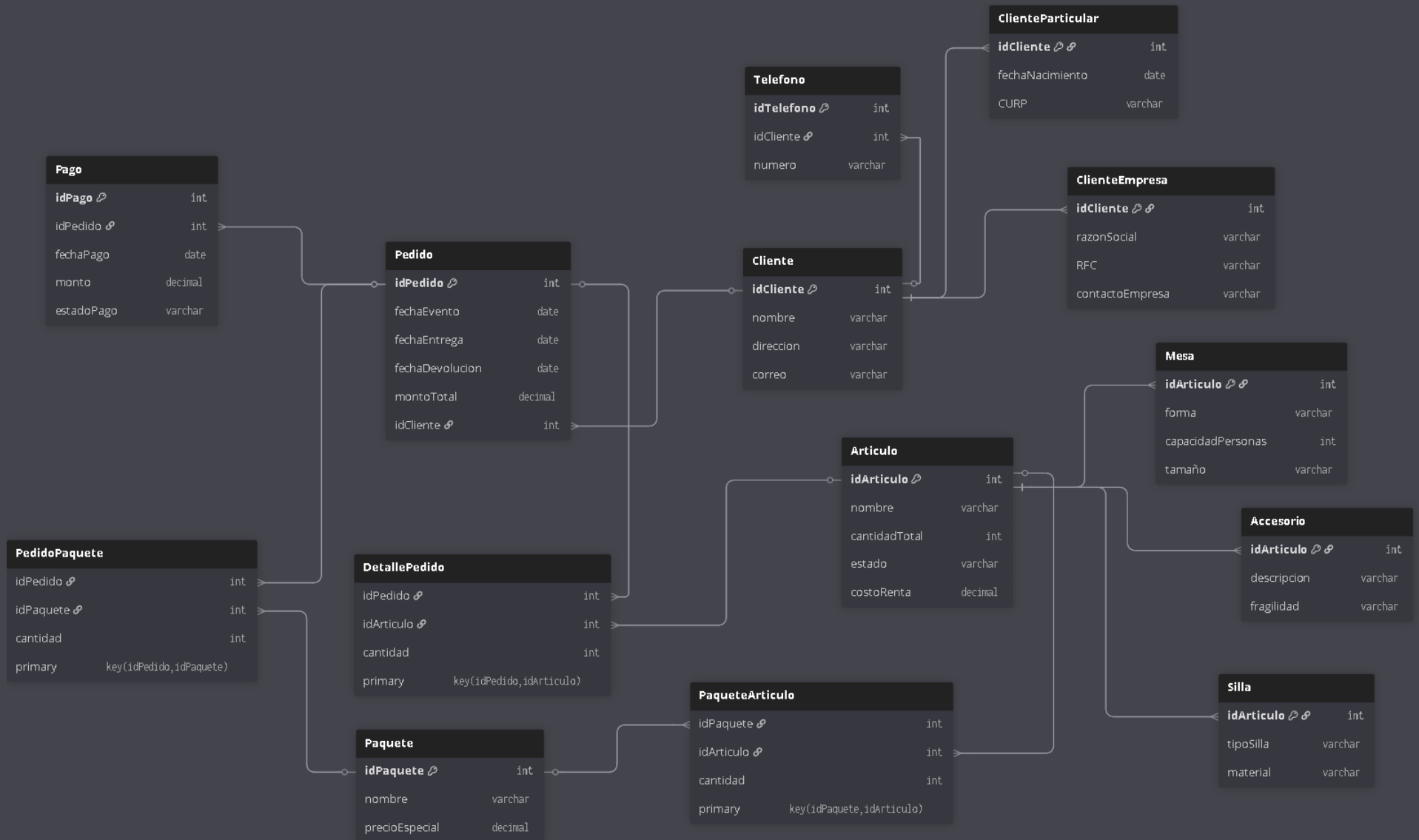


Tabla	Columna	Tipo de dato	Restricciones	Descripción
Cliente	idCliente	INT	PK, AUTO_INCREMENT	Identificador único del cliente.
	nombre	VARCHAR(100)	NOT NULL	Nombre completo del cliente.
	direccion	VARCHAR(150)	NULL	Dirección de domicilio o empresa.
	correo	VARCHAR(100)	NULL	Correo electrónico del cliente.
ClienteParticular	idCliente	INT	PK, FK → Cliente.idCliente	Identificador del cliente particular.
	fechaNacimiento	DATE	NULL	Fecha de nacimiento del cliente.
	CURP	VARCHAR(20)	NULL	CURP del cliente particular.
ClienteEmpresa	idCliente	INT	PK, FK → Cliente.idCliente	Identificador del cliente empresa.
	razonSocial	VARCHAR(100)	NULL	Nombre legal de la empresa.
	RFC	VARCHAR(15)	NULL	Registro Federal de Contribuyentes.
	contactoEmpresa	VARCHAR(100)	NULL	Nombre de la persona de contacto de la empresa.
Telefono	idTelefono	INT	PK, AUTO_INCREMENT	Identificador único del teléfono.
	idCliente	INT	FK → Cliente.idCliente	Cliente al que pertenece el número.

	numero	VARCHAR(15)	NOT NULL	Número telefónico del cliente.
Articulo	idArticulo	INT	PK, AUTO_INCREMENT	Identificador único del artículo.
	nombre	VARCHAR(100)	NOT NULL	Nombre o tipo del artículo (silla, mesa, etc.).
	cantidadTotal	INT	NOT NULL	Cantidad total disponible para renta.
	estado	VARCHAR(20)	DEFAULT 'Disponible'	Estado actual (disponible, rentado, mantenimiento).
	costoRenta	DECIMAL(10,2)	NOT NULL	Precio de renta por unidad.
Silla	idArticulo	INT	PK, FK → Articulo.idArticulo	Identificador de la silla.
	tipoSilla	VARCHAR(50)	NULL	Tipo de silla (plástica, plegable, etc.).
	material	VARCHAR(50)	NULL	Material del que está hecha la silla.
Mesa	idArticulo	INT	PK, FK → Articulo.idArticulo	Identificador de la mesa.
	forma	VARCHAR(50)	NULL	Forma de la mesa (redonda, cuadrada, etc.).
	capacidadPersonas	INT	NULL	Número de personas que caben en la mesa.
	tamaño	VARCHAR(50)	NULL	Dimensiones o tamaño general.
Accesorio	idArticulo	INT	PK, FK → Articulo.idArticulo	Identificador del accesorio.
	descripcion	VARCHAR(100)	NULL	Breve descripción del accesorio.

	fragilidad	VARCHAR(30)	NULL	Nivel de fragilidad del artículo.
Pedido	idPedido	INT	PK, AUTO_INCREMENT	Identificador único del pedido.
	fechaEvento	DATE	NOT NULL	Fecha en la que se realizará el evento.
	fechaEntrega	DATE	NULL	Fecha de entrega del pedido.
	fechaDevolucion	DATE	NULL	Fecha de devolución del pedido.
	montoTotal	DECIMAL(10,2)	NULL	Monto total del pedido.
	idCliente	INT	FK → Cliente.idCliente	Cliente que realizó el pedido.
DetallePedido	idPedido	INT	PK, FK → Pedido.idPedido	Pedido al que pertenece el detalle.
	idArticulo	INT	PK, FK → Articulo.idArticulo	Artículo incluido en el pedido.
	cantidad	INT	NOT NULL	Cantidad de unidades rentadas del artículo.
Pago	idPago	INT	PK, AUTO_INCREMENT	Identificador único del pago.
	idPedido	INT	FK → Pedido.idPedido	Pedido al que corresponde el pago.
	fechaPago	DATE	NULL	Fecha en que se realizó el pago.
	monto	DECIMAL(10,2)	NULL	Monto abonado o pagado.

	estadoPago	VARCHAR(20)	DEFAULT 'Pendiente'	Estado del pago (pendiente, completado).
Paquete	idPaquete	INT	PK, AUTO_INCREMENT	Identificador único del paquete.
	nombre	VARCHAR(100)	NOT NULL	Nombre o título del paquete.
	precioEspecial	DECIMAL(10,2)	NULL	Precio total preferencial del paquete.
PaqueteArticulo	idPaquete	INT	PK, FK → Paquete.idPaquete	Identificador del paquete.
	idArticulo	INT	PK, FK → Articulo.idArticulo	Artículo incluido en el paquete.
	cantidad	INT	NOT NULL	Cantidad de artículos incluidos.
PedidoPaquete	idPedido	INT	PK, FK → Pedido.idPedido	Pedido que incluye el paquete.
	idPaquete	INT	PK, FK → Paquete.idPaquete	Paquete incluido en el pedido.
	cantidad	INT	NOT NULL	Cantidad de paquetes rentados.

Ejercicio 7: Validación y Verificación del Modelo

Aspecto	Modelo EER	Modelo Relacional	Observaciones
Entidades fuertes	Cliente, Artículo, Pedido, Pago, Paquete	Cada una tiene su tabla	Preservadas
Entidades débiles	DetallePedido	Tabla con PK compuesta	Correcta conversión
Relaciones N:M	Pedido–Artículo, Pedido–Paquete, Paquete–Artículo	Tablas intermedias	Cumple integridad
Jerarquías ISA	Cliente/Artículo	Subtablas con 1:1	Sin redundancia
Atributos multivaluados	Telefono	Tabla separada	Correcta implementación

Conclusiones

La elaboración del modelo EER y su posterior transformación al modelo relacional permitió estructurar de forma lógica la información de nuestro negocio hipotético Sillas y Mesas Hernández.

Este proceso nos mostró cómo las jerarquías ISA, las entidades débiles y las relaciones de cardinalidad múltiple pueden representarse dentro de un esquema relacional, dándole mayor integridad y coherencia al proyecto en general.

El resultado final es una base de datos funcional (al menos hasta lo que llevamos implementado), flexible y escalable, capaz de manejar clientes particulares y empresariales, artículos de distintos tipos, pedidos, pagos y paquetes. El modelo relacional lo podemos considerar como una base sólida para la implementación de un sistema web.

Bibliografía

de Educación y FP, M. (s/f). *Generalización y especialización*. Caib.es. Recuperado el 19 de octubre de 2025, de https://sarreplec.caib.es/pluginfile.php/9827/mod_resource/content/2/72_generalizacin_y_especializacin.html

Entidad/Relación, G.-U. 2-M. (s/f). *[Unidad 2] Modelo Entidad/Relación [2.1] Modelo Entidad/Relación Extendido*. Jorgesanchez.net. Recuperado el 19 de octubre de 2025, de <https://jorgesanchez.net/presentaciones/bases-de-datos/entidad-relacion/tipos-isa.pdf>

What is a Relational Database (RDBMS)? (2021, junio 18). Oracle.com; Oracle. <https://www.oracle.com/mx/database/what-is-a-relational-database/>