

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE CIENCIAS, 2025-II
FUNDAMENTOS DE BASES DE DATOS



PRACTICA04

Modelo Relacional

NOMBRE DEL EQUIPO:
NAMEisNULL

INTEGRANTES:

Flores Mata Ricardo - 422127808

Matute Cantón Sara Lorena - 319331622

Sánchez Cruz Norma Selene - 320198508

Suárez Ortiz Joshua Daniel - 320151260

Villegas Martínez Vania Victoria - 418003114

Restricciones del Modelo Relacional y consideraciones de diseño y traducción.

Entidades Fuertes y Atributos Multivaluados:

Bazar (IdBazar, NombreBazar, Calle, NúmeroInterior, NúmeroExterior, Colonia, Estado, Modalidad, FechaInicio, FechaFin)

1. Tiene como llave primaria IdBazar, el cual se considera como un entero positivo consecutivo.
2. Para los atributos NombreBazar, Calle, NúmeroInterior, NúmeroExterior, Colonia y Estado; su dominio es *varchar* y su longitud se considera en el modelo relacional.
3. En el atributo Modalidad guardamos si el bazar es al aire libre o en un lugar cerrado.
4. FechaInicio y FechaFin son de tipo *date* en un formato DD/MM/YYYY.

AmenidadBazar (IdBazar, AmenidadBazar)

1. Tabla del atributo Multivaluado Amenidades.
2. AmenidadBazar es de tipo *varchar*, para poder registrar todas las amenidades de un Bazar.
3. Llave primaria compuesta por IdBazar y AmenidadBazar.
4. Llave Foránea IdBazar la cual proviene como llave primaria de **Bazar**.

Negocio (IdNegocio, NombreNegocio, Descripción, PrecioMínimo, PrecioMáximo)

1. Tiene como llave primaria IdNegocio y lo consideramos como un entero positivo consecutivo.
2. Los atributos NombreNegocio y Descripción son de dominio *varchar* y su longitud se especifica en modelo relacional.
3. PrecioMínimo y PrecioMáximo se consideran como dominio *numeric*, el cual tiene una precisión de 2 decimales.

TelefonoNegocio (IdNegocio, Teléfono)

1. Tabla del atributo Multivaluado Teléfono de **Negocio**.
2. Teléfono es de tipo *char* de longitud 10.
3. Llave primaria compuesta por IdNegocio y Teléfono.
4. Llave Foránea IdNegocio la cual proviene como llave primaria de **Negocio**.

CorreoNegocio (IdNegocio, Correo)

1. Tabla del atributo Multivaluado Correo **Negocio**.
2. Correo es de tipo *varchar* de longitud 50.
3. Llave primaria compuesta por IdNegocio y Correo.
4. Llave Foránea IdNegocio la cual proviene como llave primaria de **Negocio**.

RedSocialNegocio (IdNegocio, RedSocial)

1. Tabla del atributo Multivaluado RedSocial **Negocio**.
2. Correo es de tipo *varchar* de longitud 50.
3. Llave primaria compuesta por IdNegocio y RedSocial.
4. Llave Foránea IdNegocio la cual proviene como llave primaria de **Negocio**.

Estand (Número, Paquete, Precio)

1. Tiene como llave primaria Número y lo consideramos como int entre 1 y 20.
2. El atributo Paquete lo consideramos para guardar el Paquete que eligieran los emprendedores al momento de reservar el Estand para ocupar en el Bazar, por lo que su dominio es char(1), únicamente para elegir el número del paquete.
3. El atributo Precio es para guardar el precio del paquete que se haya elegido, por lo que su dominio es money.

AmenidadEstand (Número, AmenidadEstand)

1. Tabla del atributo Multivaluado Amenidad de **Estand**.
2. AmenidadEstand es de tipo *varchar* de longitud 100.
3. Llave primaria compuesta por Número y AmenidadEstand.
4. Llave Foránea Número la cual proviene como llave primaria de **Estand**.

Ticket (IdTicket, idCliente, idBazar, ComisiónBazar, idNegocio, RFC)

1. Su identificador es **id_Ticket** y lo consideramos como un entero positivo consecutivo.
2. El atributo **Precio Total** no se traduce porque es calculado (y depende del 20% de comisión para el bazar).
3. Se agrega idCliente por la traducción de la relación **Cliente - Comprar - Ticket**.
4. Se agregan idBazar y ComisiónBazar por la traducción de la relación **Bazar - Pagar - Ticket**.
5. Se agrega idNegocio por la traducción de la relación **Negocio - Imprimir - Ticket**.
6. Se agrega RFC por la traducción de la relación **Emprendedor - Cobrar - Ticket**.

Trabajador:

En la traducción del modelo entidad-relación al modelo relacional, siguiendo las reglas estándar, la entidad fuerte Trabajador se representaría como una superentidad de las entidades especializadas **Personal Organizador** y **Emprendedor**. A su vez, Personal Organizador se especializa en **Seguridad, Limpieza y Médico**, lo que resultaría en la creación de cuatro tablas separadas en la base de datos:

1. Emprendedor
2. Seguridad
3. Limpieza
4. Médico

Este diseño permitiría que un mismo RFC (Registro Federal de Contribuyentes, que identifica a los trabajadores) pudiera aparecer en múltiples tablas si un trabajador quisiera tener más de un rol, pero no es lo que nos piden los casos de uso. Sin embargo, queremos evitar esto por dos razones principales:

- Mejor organización de los empleados: Mantener la información de los trabajadores en una única tabla facilita la administración y consulta de datos.
- Garantizar unicidad en la categoría del trabajador: Cada trabajador debe pertenecer únicamente a una categoría. Es decir, no puede ser simultáneamente un Médico y un Empleado de Limpieza, por lo que cada RFC debe aparecer solo una vez en la base de datos.

Para lograr esto, en lugar de dividir la información en múltiples tablas, optaremos por un diseño que mantenga una única tabla **Trabajador** con un atributo que indique a qué categoría pertenece (Emprendedor, Seguridad, Limpieza o Médico). Así, cada trabajador tendrá un solo registro en la base de datos, asegurando que cada RFC esté asociado exclusivamente a un tipo de Trabajador.

Por lo que, siguiendo con la versión reducida del Modelo Relacional, tenemos que: .

. **Trabajador** (RFC, NombreTrabajador, APaternoTrabajador, AMaternoTrabajador, Calle, NúmeroExterior, NúmeroInterior, Colonia, Estado, Género, FechaNacimiento, Salario, esEmprendedor, esSeguridad, esLimpieza, esMédico, idNegocio)

1. Tiene como identificador **RFC**, considerado como un tipo de dato *char*: 4 letras, 6 dígitos, 2 letras y 1 dígito.
2. Los atributos Nombre, APaterno, AMaterno, Calle, NúmeroExterior, NúmeroInterior, Colonia y Estado; son de tipo *varchar* y su longitud se define en el diagrama del modelo relacional.
3. Género es de tipo *char* de longitud 1.
4. FechaNacimiento es de tipo *date* en un formato DD/MM/YYYY.
5. Salario es de tipo *money*
6. esEmprendedor, esSeguridad, esLimpieza, esMédico; son de tipo *boolean* y solamente uno puede tener el valor *true*. Esto nos permite cumplir con la restricción de los casos de uso.
7. Se agrega la llave primaria de **Negocio** por la traducción de la relación **Emprendedor - Emprender - Negocio**.

HorarioTrabajador (RFC, Horario)

1. Tabla del atributo Multivaluado Horario de **Trabajador**.
2. Horario es de tipo *varchar* de longitud 30.
3. Llave primaria compuesta por RFC y Horario.
4. Llave Foránea RFC la cual proviene como llave primaria de **Trabajador**.

TeléfonoTrabajador (RFC, TeléfonoTrabajador)

1. Tabla del atributo Multivaluado Teléfono de **Trabajador**.
2. TeléfonoTrabajador es de tipo *char* de longitud 10.
3. Llave primaria compuesta por RFC y TeléfonoTrabajador.
4. Llave Foránea RFC la cual proviene como llave primaria de **Trabajador**.

CorreoTrabajador (RFC, CorreoTrabajador)

1. Tabla del atributo Multivaluado Correo de **Trabajador**.
2. CorreoTrabajador es de tipo *varchar* de longitud 50.
3. Llave primaria compuesta por RFC y CorreoTrabajador.
4. Llave Foránea RFC la cual proviene como llave primaria de **Trabajador**.

Cliente (idCliente, NombreCliente, APaternoCliente, AMaternoCliente, Calle, NúmeroExterior, NúmeroInterior, Colonia, Estado, esFísico, esVirtual)

1. Es superentidad de: **Físico y Virtual**, consideramos una especialización total con traslape. Por lo que se genera una única tabla Cliente con todos los atributos
2. Su identificador es **id_Cliente** y lo consideramos como entero positivo consecutivo.
3. NombreCliente, APaternoCliente, AMaternoCliente, Calle, NúmeroExterior, NúmeroInterior, Colonia, Estado son atributos de tipo *varchar* y su longitud se especifica en el modelo relacional.
4. esFísico y esVirtual son atributos de tipo *boolean* y no hay problema si ambos están en modo *true*.

MétodoPago (idCliente, MétodoPago)

1. Tabla del atributo multivaluado MétodoPago de **Cliente**.
2. MétodoPago es de tipo *varchar* para elegir entre efectivo o tarjeta.

3. Llave primaria compuesta por idCliente y MétodoPago.
4. Llave foránea idCliente que proviene de **Cliente** que es llave primaria.

Tarjeta (NúmeroTarjeta, Vencimiento, CVV, idCliente)

1. NúmeroTarjeta es la llave primaria y lo consideramos como tipo de dato *char* y su longitud es de 16 caracteres.
2. Vencimiento es de tipo *date* en un formato DD/MM/YYYY.
3. CVV es de tipo *varchar* y tiene una longitud de 4 caracteres (porque existen tarjetas con un cvv de longitud 3 y otras de 4).
4. Se agrega idCliente por la traducción de la relación **Cliente - Domiciliar - Tarjeta**.

Entidades débiles:

Producto (idNegocio, idProducto, NombreProducto, Descripción, Tipo, Precio, Presentación, Stock)

1. idNegocio es heredado de Negocio porque **Producto** es una entidad débil.
2. idProducto es de tipo *int* como entero positivo consecutivo.
3. NombreProducto, Descripción, Tipo, PrecioProducto, Presentación son de tipo *varchar* y su longitud se especifica en el modelo relacional.
4. Stock se agrega por la traducción de la relación **Negocio - Vender - Producto**.

Servicio (idNegocio, idServicio, NombreServicio, Descripción, Tipo, PrecioServicio, Duración)

1. idNegocio es heredado de Negocio porque **Producto** es una entidad débil.
2. idServicio es de tipo *int* como entero positivo consecutivo.
3. NombreServicio, Descripción, Tipo son de tipo *varchar* y su longitud se especifica en el modelo relacional.
4. PrecioServicio es de tipo *money*.
5. Duración es de tipo *time* (dado que solamente necesitamos las horas y/o minutos).

Perecedero (idNegocio, idProducto, FechaPreparación, FechaCaducidad)

1. Hereda idNegocio y idProducto de la tabla **Producto**.
2. FechaPreparación y FechaCaducidad son de tipo *date* en un formato DD/MM/YYYY.

Nota: Separamos Producto y Servicio en dos entidades débiles separadas debido a que pensamos que un negocio puede ofrecer servicios y vender productos al mismo tiempo sin que una afecte la otra.

Relaciones que generan tabla:

Trabajar (idBazar, RFC)

- idBazar, RFC son llaves foráneas por la cardinalidad n:m de la relación Trabajar entre Bazar y Personal Organizador.

RegistrarProducto (idTicket, idProducto, Cantidad)

- idTicket y idProducto son llaves foráneas por la cardinalidad n:m de la relación Registrar entre Ticket y Producto.
- Cantidad es un dato de tipo *int*

RegistrarServicio (idTicket, idServicio, Duración)

- idTicket y idServicio son llaves foráneas por la cardinalidad n:m de la relación Registrar entre Ticket y Servicio.
- Duración es un atributo de tipo *time* (dado que solamente necesitamos las horas y/o minutos que dura el servicio).

Agendar (idBazar, idNegocio, NúmeroEstand, FechaAsistencia)

- idBazar, idNegocio, NúmeroEstand son llaves que se agregan a la tabla porque es una relación n-aria.
- Tiene como discriminante el atributo **FechaAsistencia** para garantizar que un Negocio no registre dos veces que asistirá el mismo día al mismo Bazar.

Relaciones que modifican tabla:

Comprar

- Relación entre Cliente y Ticket.
- Como tiene cardinalidad n:1, modifica la tabla **Ticket** mandando la llave primaria de Cliente, es decir, idCliente se agrega a la tabla Ticket.

Domiciliar

- Relación entre ClienteVirtual (que por la traducción, la tabla se llama Cliente) y Tarjeta.
- Por su cardinalidad 1:n, modifica la tabla **Tarjeta**, mandando la llave primaria de **Cliente**, es decir, idCliente se agrega a la tabla Tarjeta.

Pagar

- Relación entre Bazar y Ticket.
- Por su cardinalidad 1:n, modifica la tabla Ticket mandando la llave primaria de Bazar, es decir, idBazar se agrega a la tabla Ticket.
- El atributo de la relación pagar, **ComisiónBazar**, también se agrega a la tabla Ticket.

Imprimir

- Relación entre Negocio y Ticket.
- Por su cardinalidad 1:n modifica la tabla Ticket, agregando la llave primaria de Negocio, es decir, idNegocio se agrega a la tabla Ticket.

Cobrar

- Relación entre Emprendedor y Ticket.

- Por su cardinalidad 1:n, agrega la llave primaria de Emprendedor a la tabla Ticket, es decir, el RFC del Emprendedor se agrega a Ticket.

Emprender

- Relación entre Negocio y Emprendedor.
- Por su cardinalidad 1:n, modifica la tabla Trabajador (ya que por reglas de traducción, aquí se estarán guardando los emprendedores), agregando la llave primaria de Negocio.
- Como su único atributo es calculado, no se agrega al modelo relacional.

Vender

- Relación débil entre Negocio y Producto.
- Por su debilidad, no se genera tabla (porque ya está considerada en la tabla **Producto**). Sin embargo, al tener atributos, se agregan a la tabla **Producto**, en este caso, solamente se agrega Stock.

Ofrecer

- Relación entre Negocio y Servicio.
- Como es una relación débil, no se traduce porque ya está considerada en la tabla **Servicio** y como no tiene atributos, tampoco modifica ninguna tabla.