

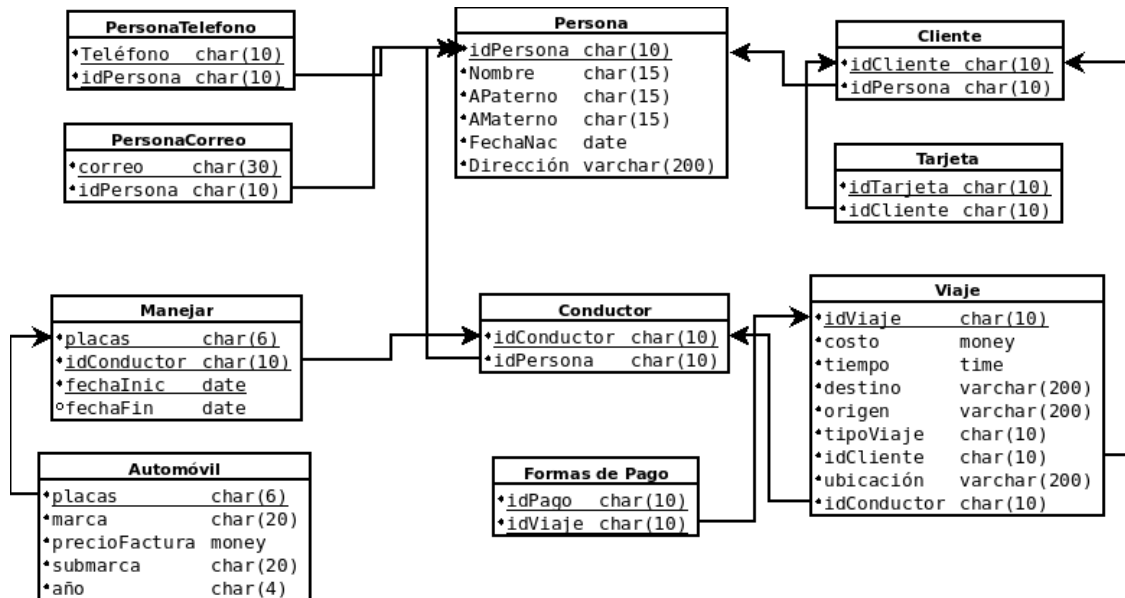
# Reporte de conversión de Entidad Relación a tablas

## Práctica 3

Jorge Francisco Cortés López  
Diego Estrada Mejia

Agosto 30, 2019

### Conversión



### Descripción y restricciones

En el diagrama están indicados, por medio de notación, la mayoría de las restricciones de integridad. Las que faltan las indicaremos mas abajo. La notación consta de:

- ° el atributo puede ser null
- • el atributo no puede ser null
- atributo si el atributo está subrayado y con un rombo al inicio es una llave primaria (o parte de una llave compuesta).

Indicaremos, tabla por tabla las restricciones que faltan, junto con la explicación de los dominios de cada atributo y de las referencias de sus llaves. Además en las relaciones que surgieron de la conversión haremos una descripción de estas.

## Persona

- **idPersona** tiene como dominio **char(10)** porque pensamos que un identificador de 10 carácter para cada persona que se registre es suficiente. Esta misma decisión se tomó para todos los identificadores de la base de datos.
- **Nombre** tiene como dominio **char(15)** porque un nombre es una cadena de caracteres de una longitud medianamente larga.
- **APaterno** tiene como dominio **char(15)** porque un apellido es una cadena de caracteres.
- **AMaterno** lo mismo que APaterno.
- **FechaNac** tiene como dominio **date** pues queremos guardar la fecha de nacimiento.
- **Dirección** tiene como dominio **varchar(200)** pues una dirección a veces puede ser muy simple o puede ser una cadena bastante larga, el límite es porque pensamos que ninguna dirección supera los 200 caracteres.

## Viaje

- **idViaje** tiene como dominio **char(10)** pues es un identificador.
- **costo** tiene como dominio **money** pues aquí guardaremos el precio de cada viaje.
- **tiempo** tiene como dominio **time** pues guardaremos el tiempo que tardó cada viaje.
- **destino** tiene como dominio **varchar(200)** pues al igual que la dirección en *Persona* puede ser una cadena muy larga.
- **origen** lo mismo que destino.
- **tipoViaje** tiene como dominio **char(10)** porque el nombre de ningún tipo de viaje rebasa los 10 caracteres.
- **idCliente** es una llave foránea de *Cliente*, esto porque cada valor en esta tabla debe tener una tupla en la tabla que refiera a un cliente específico. Como es una llave foránea debe tener el mismo dominio que el valor al que referencia.
- **ubicación** tiene como dominio **varchar(200)** pues es una ubicación.
- **idConductor** es una llave foránea de *Conductor*, por lo tanto tiene el mismo dominio que el valor al que referencia.

## Automóvil

- **placas** tiene como dominio **char(7)** pues son cadenas de caracteres con a lo más 7 letras.
- **marca** tiene como dominio **char(20)** pues el nombre de las marcas de carro normalmente no es tan largo.
- **precioFactura** tiene como dominio **money** pues es el precio del auto.
- **submarca** tiene como dominio **char(20)** pues normalmente el nombre de las submarcas no son largo.
- **año** tiene como dominio **char(4)** pues solo guardaremos los 4 dígitos del año.

## Manejar

- **placas** es una llave foránea de *Automóvil*, por lo que comparten dominios.
- **idConductor** es una llave foránea de *Conductor*, comparten dominios.
- **fechaInic** es una **date** pues es la fecha en la que el conductor empezó a viajar con el carro específico.
- **fechaFin** es una **date** por el mismo motivo que la **fechaInic**.

## Conductor

- **idConductor** es un **char(10)** pues es un identificador.
- **idPersona** es llave foránea que referencia a un atributo de *Persona*, por lo que comparten atributos.

## PersonaTeléfono

Ésta relación es nueva gracias a la conversión, necesitamos representar que el atributo teléfono en *Persona* es multivaluado

- **Teléfono** es un **char(10)** pues los números de teléfono son de esa longitud.
- **idPersona** es una llave foránea de *Persona*, por lo que comparte dominio con el atributo que referencia.

## PersonaCorreo

Otra relación nueva por la conversión que necesitamos porque podemos tener varios correos.

- **correo** es un **char(30)** porque creemos que es una longitud adecuada para cualquier correo electrónico.
- **idPersona** es una llave foránea de *Persona*, comparte dominio con el atributo al que referencia.

## Formas de Pago

Ésta relación se formó en la conversión pues el atributo forma de pago de *Viaje* es multivaluado, por lo que necesitamos crear una nueva relación.

- **idPago** es un identificador, por lo que tiene dominio **char(10)**.
- **idViaje** es una llave foránea de *Viaje*, por lo que comparte el dominio del identificador.

## Cliente

- **idCliente** es un identificador, por lo que tiene dominio en **char(10)**.
- **idPersona** es una llave foránea de *Persona*, comparten dominios.

## Tarjeta

- **idTarjeta** es un identificador, por lo que su dominio es `char(10)`.
- **idCliente** es una llave foránea de *Cliente*, comparte dominios con el identificador.

## Conversión de Modelo Entidad Relación a tablas.

«Deberán hacer una descripción de las relaciones que resulten de la conversión, explicando las referencias de sus llaves y por qué razón le asignaron el dominio a cada atributo de la relación.»

## Preguntas.

1. ¿Cuáles son las principales diferencias entre el diagrama E-R y Relacional?

La diferencia principal entre el diagrama E-R y relacional es que el diagrama E-R se enfoca entre las entidades y sus relaciones. Por otro lado, el modelo relacional se enfoca en tablas y la relación entre los datos de las tablas.

Un diagrama E-R describe los datos con el conjunto de entidades, relaciones y atributos. Sin embargo, el modelo relacional describe los datos con las tuplas, atributos y el dominio de los atributos.

Uno puede entender fácilmente las relaciones entre los datos en el diagrama E-R comparado con el modelo relacional.

El diagrama E-R tiene una cardinalidad asociada como una restricción en donde el modelo relacional no tiene esa restricción.

2. ¿Cuáles son las principales características del diagrama E-R? Un diagrama E-R es una representación gráfica que modela relaciones entre personas, objetos, conceptos o eventos dentro de un sistema. Existen cinco componentes básicos de un diagrama entidad-relación. Componentes similares estarán diseñados por la misma forma. Por ejemplo, todas las entidades estarán encerradas por un rectángulo, mientras que todos los atributos estarán encerrados por un rombo. Las componentes incluyen:

- (a) Entidades, las cuales son objetos o conceptos que pueden tener datos guardados sobre ellos. Las entidades se refieren a las tablas que usamos en las BDD.
- (b) Atributos, los cuales son propiedades o características de las entidades. Un atributo puede ser denotado como una llave primaria, la cual identifica un atributo único, o una llave foránea, la cual puede ser asignada a múltiples atributos.
- (c) Las relaciones entre y sobre esas entidades
- (d) Acciones, las cuales describen como las entidades comparten información a la base de datos.
- (e) Conexión de líneas.

Una notación de cardinalidad puede entonces definir los atributos de la relación entre las entidades. Las cardinalidades pueden denotar que una entidad es opcional (Por ejemplo, un cliente puede o no tener una tarjeta de cliente distinguido) u obligatoria (Por ejemplo, debe haber al menos un producto listado en un almacén).

Las tres principales cardinalidades son:

- (a) Una relación uno-a-uno (1:1). Por ejemplo, si cada cliente en una bases de datos está asociado con una sola dirección de envío.
- (b) Una relación uno a varios (1:M). Por ejemplo, un sólo cliente puede hacer una orden para múltiples productos. El cliente está asociado con varias entidades, pero todas esas entidades tienen una sola conexión de regreso con el mismo cliente.
- (c) Una relación de varios a varios (M:N). Por ejemplo, una compañía en donde todos los agentes del call center trabajan con múltiples clientes, cada agente está asociado con varios clientes, y varios clientes pueden estar relacionados con muchos otros agentes.

3. ¿Es necesario tener llaves primarias en cada entidad de nuestro diagrama? Explica.

Las llaves primarias y foráneas son una manera en las cuales podemos restringir datos relacionados para asegurarnos que la BDD se mantenga consistente y para asegurarnos de que no hayan datos redundantes en la BDD como resultado de eliminar una tabla o una columna en una de las tablas que puede afectar los datos en otras tablas que puedan estar relacionadas. Puede causar problemas de integridad de datos así como problemas en la aplicación que haga uso de la base de datos. Por lo que siempre es recomendable asignar una llave primaria o foránea a cada entidad.

4. Explica con tus palabras que es una llave primaria (PK), llave candidata (UNIQUE) y llave foránea (FK)

**PK** Una llave primaria es un atributo o conjunto de atributos que nos ayudan a identificar un ejemplar de una entidad, una llave primaria en general debería de cumplir con:

Tener un valor no vacío para cada ejemplar de la entidad, el valor debe ser único para cada ejemplar de la entidad y los valores no deben cambiar o convertirse en vacíos a lo largo de la existencia de los ejemplares de la entidad.

**UNIQUE** Una llave candidata es una combinación de atributos que pueden ser usados únicamente para identificar un ejemplar de la entidad sin referirse a otros datos. Todas las llaves candidatas tienen algunas propiedades en común. Una de las propiedades es que durante la existencia de la llave candidata, el atributo usado para la identificación debe mantenerse igual, luego otra propiedad es que el valor no puede ser vacío y que además la llave candidata debe ser única.

**FK** Una llave foránea es un atributo que completa una relación identificando la entidad padre. Las llaves foráneas proveen un método para mantener la integridad de los datos (llamada integridad referencial) y para navegar entre diferentes instancias de una entidad. Cada relación en el modelo debe estar sostenida por una llave foránea.

## Tabla de tipos de datos.

A continuación se muestra una tabla con los diferentes tipos de datos con los que cuenta PostgreSQL. Se pueden consultar en la [documentación](#).

Lista de tipos de datos en PostegreSQL		
Nombre de tipo	Alias	Descripción
bigint	int8	entero de 8 bits con signo
bigserial	serial8	entero de 8 bits auto-incrementante
bit [ (n) ]		cadena de bits de tamaño fijo
bit varying [ (n) ]	varbit	cadena de bits de tamaño variable
boolean	bool	valor booleano
box		caja rectangular en un plano
bytea		datos binarios ("arreglo de bytes")
character [ (n) ]	char [ (n) ]	cadena de caracteres de tamaño fijo
character varying [ (n) ]	varchar [ (n) ]	cadena de caracteres de tamaño variable
cidr		dirección de red IPv4 o IPv6
circle		círculo en un plano
date		fecha de calendario (año, mes, día)
double precision	float8	número real con doble precisión
inet		dirección host IPv4 o IPv6
integer	int, int4	entero de cuatro bits con signo
interval [fields] [(p)]		Intervalo de tiempo
json		dato JSON
line		línea infinita en un plano
lseg		segmento de línea en un plano
macaddr		dirección MAC
money		valor monetario
numeric [(p,s)]	decimal [(p,s)]	selección precisa de un valor numérico
path		trayectoria geométrica en un plano
point		punto geométrico en un plano
polygon		trayectoria cerrada en un plano
real	float4	número real con precisión simple
smallint	int2	entero con signo de dos bytes
smallserial	serial2	entero autoincrementante de dos bytes
serial	serial4	entero autoincrementante de cuatro bytes
text		cadena de tamaño variable
time [(p)] [without time zone]		tiempo del día (sin zona horaria)
time [(p)] with time zone	timetz	tiempo del día. con zona horaria
timestamp [(p)] [without time zone]		fecha y hora (sin zona horaria)
tsquery		consulta de texto
tsvector		búsqueda de texto en documento
txid_snapshot		ID de transacción a nivel de usuario de una copia
uuid		identificador único universal
xml		dato XML