

Integrantes:

- Cortés Macías Gretel Penélope
- García Landa Brenda Yareli
- Montaña Pérez Joshua Said
- Mora Abonce Samantha

Pregunta

Especificación de las consideraciones que tuvieron al escoger el tipo de dato y las restricciones de dominio para cada columna.

• Efectivo:

idDinero CHAR(18) NOT NULL CHECK(CHAR_LENGTH(idDinero) = 18) UNIQUE \Rightarrow Para tener un id de la transacción del efectivo. Debe de ser única y no vacía para poder ser llave.

curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) \Rightarrow Para que concuerde con la persona que está pagando, como un cliente puede hacer múltiples transacciones no debe de ser única.

monto MONEY NOT NULL \Rightarrow Sólo dinero para poder operarlo así, no vacío por lo mismo.

• Tarjeta:

numero CHAR(16) NOT NULL CHECK(CHAR_LENGTH(numero) = 16) UNIQUE \Rightarrow Las tarjetas tienen 16 números, como queremos que sea llave no puede repetirse ni ser vacía.

curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) \Rightarrow Para que concuerde con los datos de la llave foránea. No podemos unique porque puede que un cliente tenga, por ejemplo, tarjeta de crédito y de débito.

titular VARCHAR(50) CHECK(titular <> "") \Rightarrow No habíamos corregido desde hace varias prácticas este para que fuera Nombre, Paterno y Materno por separado, pero debería ser un nombre.

vencimiento DATE NOT NULL \Rightarrow Para poder modelar la fecha. No debe de ser vacía.

ccv INT CHECK (ccv between 100 and 999) \Rightarrow Es un INT y debe de tener 3 dígitos, así que el más pequeño es 100 y el más grande es 990.

tipo VARCHAR(20) CHECK(tipo <> "") \Rightarrow Sólo hay dos tipos, de débito y de crédito. Con 20 es más que suficiente.

• Pedido:

idPedido CHAR(18) NOT NULL CHECK(CHAR_LENGTH(idPedido) = 18) UNIQUE \Rightarrow Igual para poder identificar cada transacción, no creemos que necesiten más de esta cantidad para sus transacciones. Como va a ser llave no queremos que esté vacía y queremos que sea única.

curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) \Rightarrow Para que concuerde con los datos de la llave foránea, como un cliente puede hacer varios pedidos no es única.

fecha DATE NOT NULL \Rightarrow Como buscamos la fecha en la que se pidió el producto.

hora TIME NOT NULL \Rightarrow Como queremos modelar el tiempo en el que se pidió el producto.

- **Express:**

idEnvio CHAR(18) NOT NULL CHECK(CHAR_LENGTH(idEnvio) = 18) UNIQUE \Rightarrow En teoría habrá un envío por cada pedido, así que concuerde. Como es llave no debe de ser nulo y debe de ser único.

idPedido CHAR(18) NOT NULL CHECK(CHAR_LENGTH(idPedido) = 18) \Rightarrow Para que concuerde con los datos de la llave foránea.

curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) \Rightarrow Para que concuerde con los datos de la llave foránea.

hora TIME NOT NULL \Rightarrow Para modelar el tiempo del envío.

fecha DATE NOT NULL \Rightarrow Para modelar la fecha en la que se realizó el envío

costoExtra MONEY NOT NULL \Rightarrow El costo extra del envío express, como es dinero lo podemos usar.

tiempo TIME NOT NULL \Rightarrow Para tener el tiempo del envío, como es express debe de durar menos de 24 horas así que lo podemos manipular con este tipo de dato (pensando por ejemplo 03:32 es un envío que duró 3 horas y media).

estatus CHAR(15) NOT NULL \Rightarrow Tiene posibles estatus de 'Terminado', 'En proceso', 'Cancelado'

- **Normal:**

idEnvio CHAR(18) NOT NULL CHECK(CHAR_LENGTH(idEnvio) = 18) UNIQUE \Rightarrow Por las mismas razones que el de express. También es llave y por eso única y no nula.

idPedido CHAR(18) NOT NULL CHECK(CHAR_LENGTH(idPedido) = 18) \Rightarrow Para que concuerde con la llave foránea.

curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) \Rightarrow Para que concuerde con la llave foránea.

hora TIME NOT NULL \Rightarrow Son los mismos atributos que la tabla de express, así que la misma justificación.

fecha DATE NOT NULL \Rightarrow Análogo, de express.

estatus CHAR(15) NOT NULL \Rightarrow Análogo al de express.

- **Persona:**

curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) UNIQUE \rightarrow Porque el Curp tiene 18 dígitos

curpSupervisor CHAR(18) CHECK(CHAR_LENGTH(curp) = 18) \rightarrow Porque el Curp tiene 18 dígitos

nombre VARCHAR(50) NOT NULL CHECK(nombre <> '') \rightarrow Si la persona está entonces su nombre debería ser distinto de null, consideramos que con 50 caracteres es suficiente

apellidoPaterno VARCHAR(50) NOT NULL CHECK(apellidoPaterno <> '') \rightarrow Si la persona está entonces su apellido paterno debería ser distinto de null, consideramos que con 50 caracteres es suficiente

apellidoMaterno VARCHAR(50) NOT NULL CHECK(apellidoMaterno <> '') → Si la persona está entonces su apellido materno debería ser distinto de null, consideramos que con 50 caracteres es suficiente

genero CHAR(1) NOT NULL CHECK(genero <> '') → Si la persona está entonces el género debería ser distinto de null, consideramos que con 1 caracter es suficiente(M, F)

fechaNacimiento DATE NOT NULL → Si la persona está entonces su fecha de nacimiento debería ser distinto de null, consideramos que con date podíamos modelar bien esta parte

calle VARCHAR(50) NOT NULL CHECK(calle <> '') → Si la persona está entonces su calle debería ser distinto de null, consideramos que con 50 caracteres es suficiente

codigoPostal INT NOT NULL CHECK(codigoPostal between 10000 and 99999) → Si la persona está entonces su código postal debería ser distinto de null, consideramos ese rango porque el número más pequeño es 10000 y el más grande 99999

estado VARCHAR(50) NOT NULL CHECK(estado <> '') → Si la persona está entonces su estado debería ser distinto de null, consideramos que con 50 caracteres es suficiente

municipio VARCHAR(50) NOT NULL CHECK(municipio <> '') → Si la persona está entonces su municipio debería ser distinto de null, consideramos que con 50 caracteres es suficiente

numero INT NOT NULL → Si la persona está entonces su nombre debería ser distinto de null

contrasenia VARCHAR(50) CHECK(contrasenia <> '') → Si la persona está entonces su contraseña debería ser distinta de null, consideramos que con 50 caracteres es suficiente

email VARCHAR(50) CHECK(email <> '') → Si la persona está entonces su email debería ser distinto de null, consideramos que con 50 caracteres es suficiente

horaSalida TIME → Elegimos TIME porque queremos modelar una hora en específico

horaEntrada TIME Elegimos TIME porque queremos modelar una hora en específico

calificacion REAL CHECK (calificacion between 0 and 5) → El rango que elegimos es de 1 (malo) hasta 5 (excelente)

numEnvios BIGINT CHECK (numEnvios >= 0) → El número de envíos van desde 0 en adelante, ya que pueden o no existir los mismos

esCliente BOOLEAN NOT NULL → Para saber si la persona es cliente elegimos un booleano

esRepartidor BOOLEAN NOT NULL → Para saber si una persona es repartidor elegimos un booleano

- **TeléfonoProveedor:**

telefono CHAR(10) NOT NULL CHECK(CHAR_LENGTH(telefono) = 10) UNIQUE → El proveedor debe tener mínimo un número, los dígitos de los teléfonos son 10

rfc CHAR(13) NOT NULL CHECK(CHAR_LENGTH(rfc) = 13) UNIQUE → El rfc tiene 13 caracteres y un proveedor debe contar con uno, por eso debe de ser no nulo

- **Proveedor:**

rfc VARCHAR(13) NOT NULL CHECK(CHAR_LENGTH(rfc) = 13) UNIQUE → Un proveedor debe de contar con un rfc por eso no puede ser nulo y el rfc tiene 13 caracteres

razonSocial VARCHAR(75) NOT NULL → Un proveedor debe de tener una razon social que no debe ser nula y consideramos que 75 caracteres son suficientes

cp CHAR(5) NOT NULL CHECK(CHAR_LENGTH(cp) = 5) → El código postal debe ser distinto de null y este tiene 5 dígitos

calle VARCHAR(50) NOT NULL → La calle no puede ser nula y consideramos que 50 caracteres son suficientes

municipio VARCHAR(50) NOT NULL → El municipio no puede ser nulo y consideramos que 50 caracteres son suficientes

estado VARCHAR(50) NOT NULL → El estado debe de ser distinto de null y consideramos que 50 caracteres son suficientes

numero int CHECK(numero ≥ 0) → El número no puede ser negativo, es por eso que se toman de 0 en adelante

- **Categoría:**

nombre VARCHAR(50) NOT NULL CHECK(CHAR_LENGTH(nombre) ≤ 50) UNIQUE → Si la categoría existe su nombre no debe ser nulo, consideramos que 50 caracteres son suficientes

descripcion VARCHAR(75) CHECK(descripcion <> '') → Consideramos que 75 caracteres son suficientes para la descripción

- **Producto:**

idProducto CHAR(18) NOT NULL CHECK(CHAR_LENGTH(idProducto) = 18) UNIQUE → Si el producto existe su id no debe ser nulo y tiene que tener 18 caracteres

nombre VARCHAR(50) CHECK(nombre <> '') → Consideramos que 50 caracteres son suficientes para el nombre del producto

rfc CHAR(13) NOT NULL CHECK(CHAR_LENGTH(rfc) = 13) → El rfc del producto tiene que tener 13 caracteres y no debe ser nulo

descActivo REAL NOT NULL CHECK(descActivo between 0 and 100)

nomArchivo VARCHAR(75) CHECK(nomArchivo <> '') → Consideramos que 75 caracteres son suficientes para el nombre del archivo

descripcion VARCHAR(250) CHECK(descripcion <> '') → Consideramos que para la descripción del producto son suficientes 250 caracteres

uniDispo int CHECK(uniDispo ≥ 0) → Los productos pueden o no existir en una categoría, por eso consideramos de 0 en adelante

precio MONEY NOT NULL → Si el producto existe entonces forzosamente tiene que tener un precio,

por eso no puede ser nulo

nombreP VARCHAR(50) CHECK(nombreP <> ") → Consideramos que 50 caracteres son suficientes para el nombre del Producto