

KIOSKSDB

Curso: SQL

Comisión: 53175

Nombre del alumno: Vazquez Velasquez, Henry José

Temática de la Base de Datos

Sistema de Base de Datos para que usarán varios locales de mercadería (kioscos) donde se pueden ingresar los productos que se adquieren para ponerlos a la venta, crear las facturas automáticamente desde el sistema y generar estadísticas que sirvan para mejorar el rendimiento de los locales.

Objetivo de la Base de Datos

Almacenar los datos más importantes del negocio para gestionarlo y generar procedimientos automáticos que permitan agilizar la realización de compras y ventas de los locales de manera que sea escalable y eficiente.

Descripción de las tablas

{ }: Lo que se puede agregar en las próximas entregas.

1. **Clientes**: Almacenará la información más relevante de los clientes que le compren un producto a los kioscos, principalmente tendrán la función de recuperar los datos ingresados en esta tabla para generar la factura con sus datos relevantes.

Atributos:

- DNI del cliente [varchar(8)]
- Nombre del cliente [varchar(100)]
- Apellido del cliente [varchar(100)]
- {El email [varchar(50)] del cliente}
- {El teléfono [varchar(30)] del cliente}

2. **Productos**: Tendrá la información básica de los productos que se compran y que después pondrán venderse.

Atributos:

- ID del producto [int autonum] [Primary Key]
- Descripción del producto [varchar(100)]
- Marca del producto [varchar(100)]
- Cantidad disponible en stock del producto [int]
- Precio al que se compró el producto [float]
- Precio en el que se venderá el producto* [float]
- {La sucursal [int autonum] que compró el producto}
- {La fecha [date] en que se compró el producto}

KIOSKSDB

***Aclaración:** El precio de venta siempre será mayor al precio de compra en un 35% al 70% para que sea rentable la compra del producto.

3. **Kioscos:** Contendrá la información relevante sobre los locales que lleven a cabo el modelo de negocio.

Atributos:

- ID del kiosco [int autonum] [Primary Key]
- Dirección [varchar(100)]
- {Nombre del local [varchar(100)]}
- {Cantidad de empleados [int]}
- {Nombre de dueño [varchar(100)]}

4. **Métodos de Pago:** Entidad que se especializará en guardar la información de las modalidades de pago (por ahora son 3; efectivo, débito y crédito).

Atributos:

- ID del método de pago [tinyint 1 a 3] [Primary Key]
- Descripción del método de pago [varchar(100)]
- Valor de las cuotas con ese método de pago [float]*
- Cantidad de cuotas con ese método de pago [float]*

* **Aclaración:** Servirán principalmente para los pagos que se realicen con crédito.

5. **Facturas:** Tabla de hechos que reunirá los datos más importantes de las 4 tablas anteriores con el objetivo de que sirva como factura válida de las ventas del local.

Atributos:

- ID de la factura [int autonum] [Primary Key]
- DNI del cliente [varchar(8)] [Foreign Key]
- {Nombre del cliente [varchar(100)]}
- Fecha en la que se realizó la factura [date]
- Id del producto [int autonum] [Foreign Key]
- Cantidad comprada del producto [int]
- {Precio unitario [float] del producto}
- Total de la factura [float]
- ID del método de pago [tinyint 1 a 3] [Foreign Key]
- Estado de la factura [bool]*
- ID del kiosco donde se realizó la compra [int autonum] [Foreign Key]

KIOSKSDB

- {Dirección del kiosco [varchar(100)] donde se realizó la compra}
- {Categoría [char] de la factura}
- {Descripción de la categoría [varchar(50)] de la factura*}

* **Aclaración 1:** Cancelada o Válida.

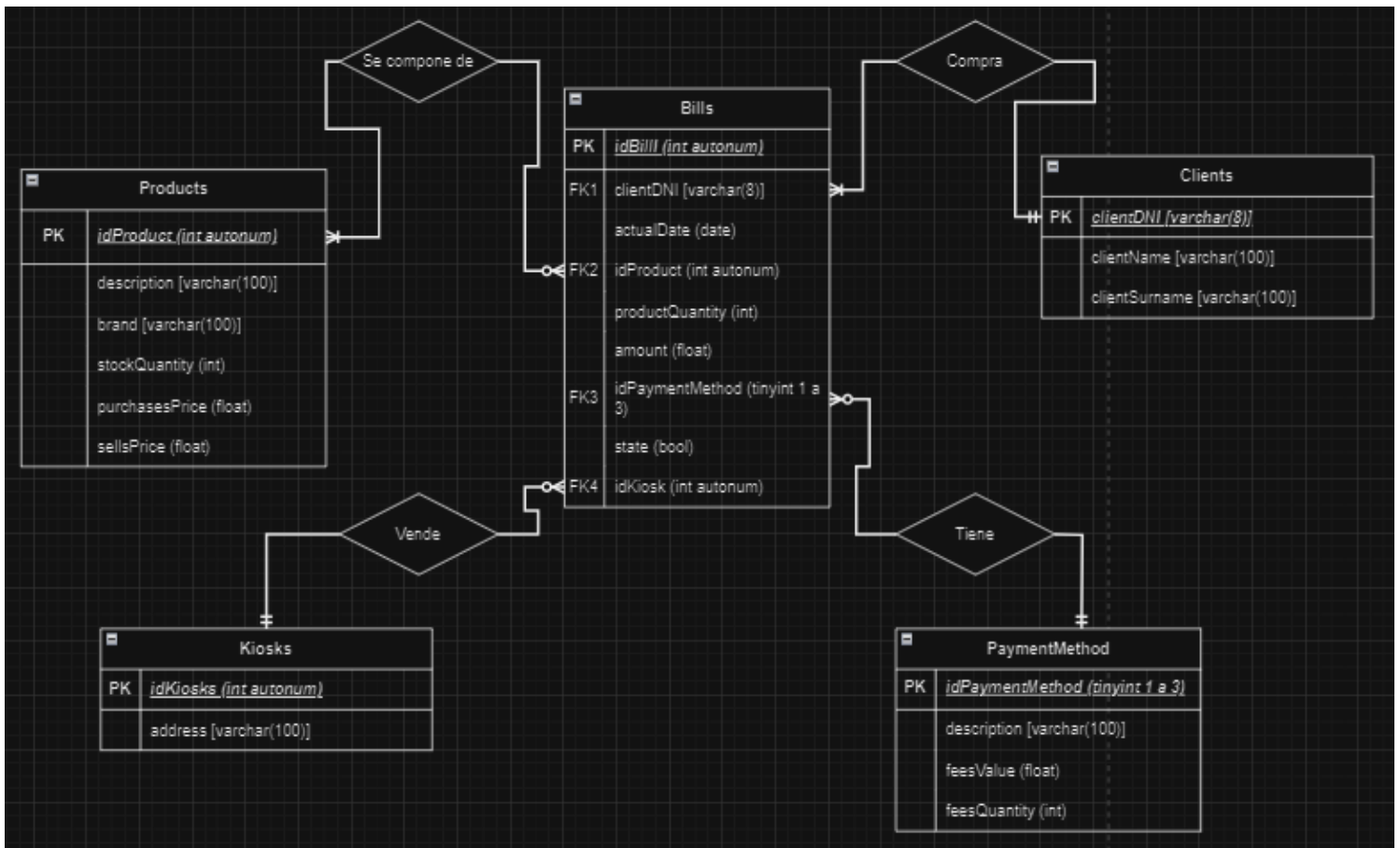
* **Aclaración 2:** Responsable inscripto, Monotributo, etc.

6. {**Empleados:** Guardará los datos relevantes de los trabajadores que tenga el local.

Atributos:

- DNI del empleado [varchar(8)]
- Nombre del empleado [varchar(100)]
- Apellido del empleado [varchar(100)]
- Teléfono del empleado [varchar(30)]
- Email del empleado [varchar(50)]
- Horas trabajadas en la semana [int]
- Salario por hora trabajada del empleado [float]
- Fecha en que empezó a trabajar en el kiosco [date]

Diagrama Entidad – Relación (DER)



KIOSKSDB

Nombre del archivo para editar el DER: ***DER Proyecto Final CoderHouse.drawio***