

Examen introducción al backend y las bases de datos

Jose Gabriel Álvarez Latorre

C4

Skills

2025

CampusBike

Introducción – Problema a solucionar

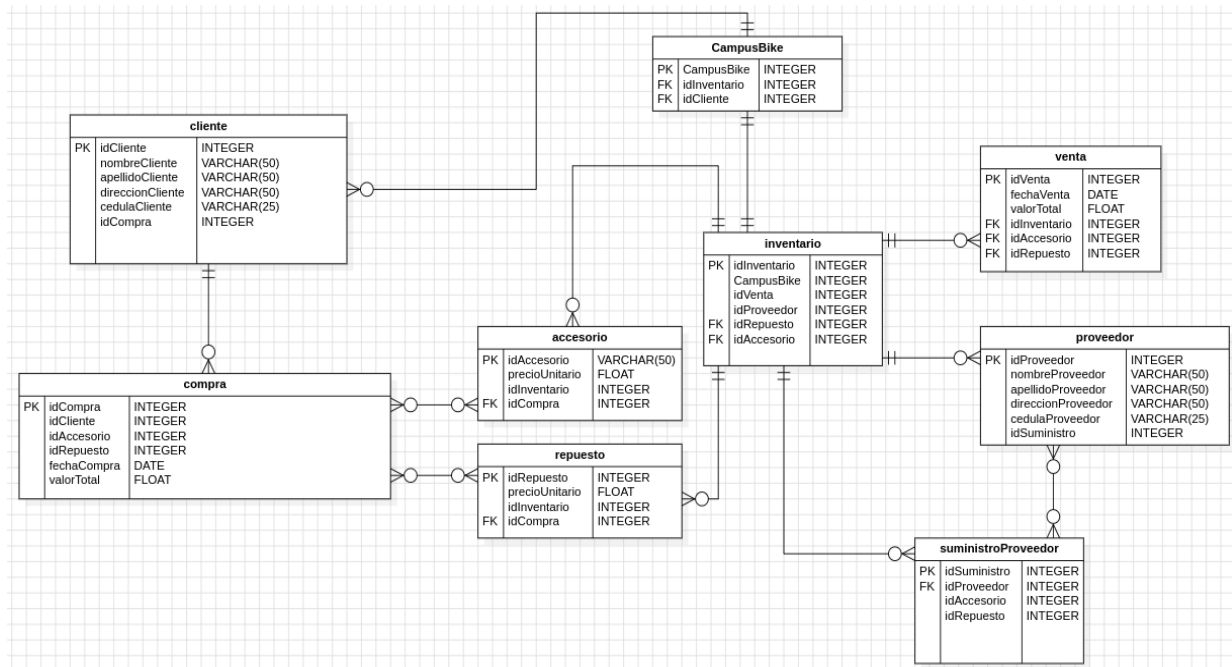
CampusBike es un negocio de venta de bicicletas que sufre problemas con la gestión de su información. La falta de renovación de su sistema hacia uno que organice, analice y gestione toda la información de manera eficiente es el objetivo de esta renovación de base de datos para mejorar los siguientes aspectos, de la siguiente manera.

1. Gestión deficiente de clientes → La falta de un seguimiento adecuado de la información de los clientes hacia un sistema que detalle su información , historial de compras e interacciones.
2. Ineficiencia en la cadena de suministro → La gestión manual de la información de los proveedores hacia un sistema que detalle su información , sus términos de suministro y seguimiento de pedidos.
3. Dificultades en el análisis de ventas y compras → La falta de una base de datos robusta para generar análisis de tendencias y comportamientos de las ventas hacia un sistema que imprima todos los datos de manera organizada.
4. Gestión ineficiente de repuestos y accesorios → El poco control de la información de entradas y salidas y repuestos genera problemas de exceso de stock o por el contrario, de desabastecimiento. El objetivo es el control preciso de inventarios de repuestos y accesorios, optimizando el estado de disponibilidad de cada elemento.

Desarrollo del problema

1. Modelo relacional de la base de datos ‘CampusBike’

Con todos los datos entregados se realizó el modelo relacional de la base de datos para quedar compuesto de la siguiente manera:



2. Código SQL

Para la construcción física de la base de datos se usó la consola de linux para acceder al host, crear la base de datos, usarla y crear las tablas. Los comandos usados en total fueron los siguientes (Aunque también estarán dentro del archivo .sql del repositorio de este proyecto):

```

create database CampusBike;
use CampusBike;
create table cliente(
    idCliente int primary key,
    nombreCliente varchar(50) not null,
    apellidoCliente varchar(50) not null,
    direccionCliente varchar(50) not null,
    cedulaCliente varchar(25) not null,
    idCompra int
);
create table proveedor(
    idProveedor int primary key,
    nombreProveedor varchar(50) not null,
    apellidoProveedor varchar(50) not null,
    direccionProveedor varchar(50) not null,
    cedulaProveedor varchar(25) not null,
    idSuministro int
);
create table suministroProveedor(
    idSuministro int primary key,
    idProveedor int,

```

```

    idAccesorio int,
    idRepuesto int,
    foreign key (idProveedor) references proveedor(idProveedor)
);
create table compra(
    idCompra int primary key,
    idCliente int,
    idAccesorio int,
    idRepuesto int,
    fechaCompra date,
    valorTotal float check (valorTotal > 0)
);
create table accesorio(
    idAccesorio int primary key,
    precioUnitario float check (precioUnitario > 0),
    idCompra int,
    idInventario int,
    foreign key (idCompra) references compra(idCompra)
);
create table repuesto(
    idRepuesto int primary key,
    precioUnitario float check (precioUnitario > 0),
    idCompra int,
    idInventario int,
    foreign key (idCompra) references compra(idCompra)
);
create table inventario(
    idInventario int primary key,
    CampusBike int,
    idAccesorio int,
    idRepuesto int,
    idVenta int,
    foreign key (idAccesorio) references accesorio(idAccesorio),
    foreign key (idRepuesto) references repuesto(idRepuesto)
);
create table venta(
    idVenta int primary key,
    idInventario int,
    idAccesorio int,
    idRepuesto int,
    fechaVenta date,
    foreign key (idInventario) references inventario(idInventario),
    foreign key (idAccesorio) references accesorio(idAccesorio),
    foreign key (idRepuesto) references repuesto(idRepuesto)
);
create table CampusBike(
    CampusBike int primary key,
    idInventario int,
    idCliente int,

```

```
foreign key (idInventario) references inventario(idInventario),  
foreign key (idCliente) references cliente(idCliente)  
);
```