

**Introducción**

Se está desarrollando un sistema para comercios, en el cuál puedan llevar registro de sus ventas y los vendedores que las realizan. Esta información servirá para ser visualizada desde una app móvil y generar reportes de ingresos de cada comercio.

**Requerimientos**

Es necesario definir el modelo de base de datos, para enviarla al equipo de backend y puedan desarrollar los microservicios para desplegar el API.

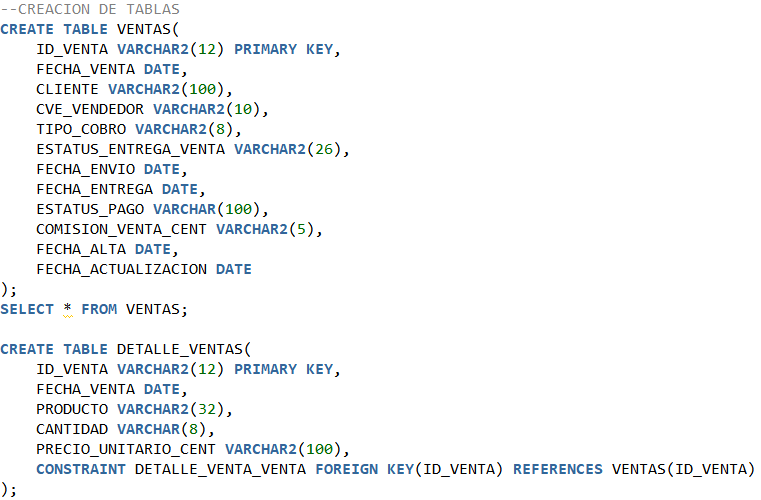
**No funcionales**

* Se espera que entregues un código limpio, fácil de mantener, leer y autodocumentado.
* Se espera que el código siga una guía de estilos definidos (<https://oracle.readthedocs.io/en/latest/sql/basics/style-guide.html>)
* Las tablas estarán montadas sobre una base de datos Oracle 12.1.0.1

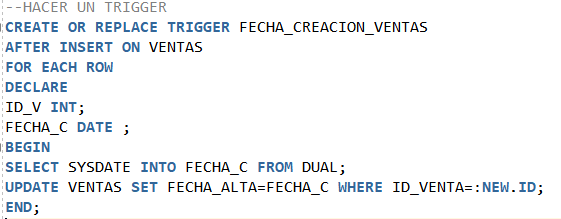
**Funcionales**

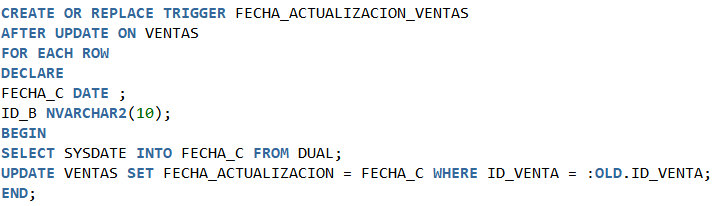
De acuerdo al siguiente diagrama, realizar lo siguiente:

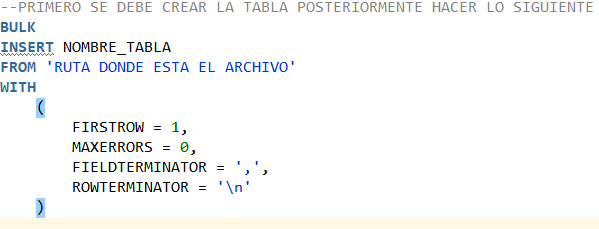
1. Scripts para creación de tablas.



1. Script para habilitar un trigger en FECHA\_ALTA o FECHA\_ACTUALIZACION, según sea el caso (INSERT o UPDATE).





1. Desarrollar un paquete incluyendo las siguientes funciones o stored procedures:
   1. Consulta de Ventas por tipo de cobro, fecha de envío, filtro por vendedor, fecha de entrega, estatus de pago y rango de fechas de alta
   2. Insertar un nuevo registro de venta con 5 productos
   3. Actualizar el estatus de pago de una venta
   4. Consulta de Detalle de ventas con base en el id de venta
2. Realizar la propuesta de un modelo para registro de productos de tienda y asignarlos a la venta (Agregar el diagrama en el examen).
3. Script para insertar datos a partir de un archivo csv.
4. 
5. Adicional, puedes hacer la propuesta de un modelo para registro de ventas de comercio.

**Instrucciones para la prueba**

1. El código debe quedar almacenado en un repositorio git.
2. Como tu primer commit, incluye un README especificando el modelo de base de datos y la estructura del proyecto.
3. Al terminar la prueba, responde al mismo correo en el cual se te envío la prueba, incluyendo los archivos anexos y una breve explicación a “tu cliente” de cómo puede consumir la información de las tablas.

**Puntos extra**

1. El código puede incluir pruebas unitarias.
2. Proponer un mejor modelo de la estructura actual.
3. Proponer un modelo para registro de vendedores del comercio.