

MEDICIÓN – MANEJO DE DATOS Y CREACIÓN DE QUERIES



Objetivo

Medir conocimiento en consultas SQL y entendimiento de los datos en base a un modelo

Pregunta 1: Manipulación de Datos con API

Se proporciona la siguiente URL de una API pública que devuelve información sobre libros en formato JSON:

`https://api.example.com/books`

Escriba un script en Python que haga una solicitud a esta API, obtenga los datos y realice las siguientes tareas:

1. Calcule el número total de libros en la lista.
2. Guarde los datos en un archivo CSV llamado `books.csv`.

Pregunta 2: SQL y Modelado de Bases de Datos

Se proporciona el siguiente esquema simplificado de una base de datos de una librería:

Tabla: Libros

- `id_libro` (PK)
- `titulo`
- `autor_id` (FK a `Autores.id_autor`)
- `categoria_id` (FK a `Categorias.id_categoria`)
- `precio`

Tabla: Autores

- `id_autor` (PK)
- `nombre`
- `nacionalidad`

Tabla: Categorias

- `id_categoria` (PK)
- `nombre`

Escriba consultas SQL para realizar las siguientes operaciones:

1. Seleccione el título y el nombre del autor de todos los libros de la categoría "Ficción".
2. Calcule el precio promedio de todos los libros en la tabla `Libros`.
3. Actualice el precio de todos los libros escritos por el autor con `id_autor = 5` en un 10% de descuento.

Pregunta 3: Integraciones ERP vía API (Utilizando JSONPlaceholder API)

Supongamos que necesitamos integrar datos de JSONPlaceholder utilizando su API RESTful para obtener información detallada sobre usuarios y publicaciones.

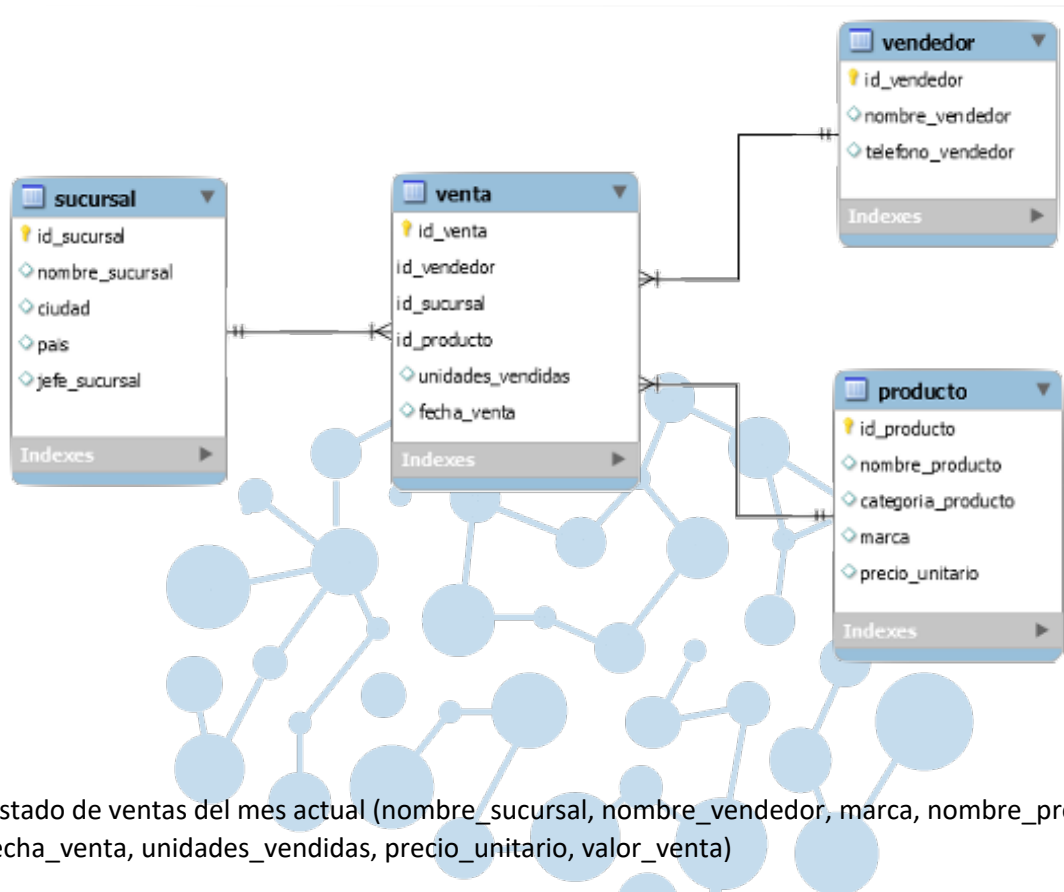
1. Diseñe una función en Python que:
 - Autentique y obtenga acceso a la API de JSONPlaceholder.
 - Extraiga los detalles de usuarios y publicaciones (nombre de usuario, email, título de publicación, cuerpo de publicación, etc.) y guarde esta información en las tablas correspondientes (Usuarios, Publicaciones).
 - Modele los datos en un esquema de base de datos estrella.
2. Se necesita la información en una esquema NearRealtime con actualización diaria, realice un función Python que diariamente sea ejecutada (sugiera el mecanismo de ejecución en la nube).

Instrucciones Adicionales:

- Utilice los siguientes endpoints de JSONPlaceholder para obtener los datos:
 - <https://jsonplaceholder.typicode.com/users> para datos de usuarios.
 - <https://jsonplaceholder.typicode.com/posts> para datos de publicaciones.
 - Modele los datos en tablas separadas según el modelo propuesto (Usuarios, Publicaciones).
 - Se solicita además utilizar las mejores prácticas de programación y mecanismos de log .
3. Finalmente se requiere que las tablas obtenidas sean creadas en una BD PostgreSQL 16. implementada en un esquema dockerizado.
 - Entregue los elementos que sean requeridos para crear:
 - Contenedor de la Base de Datos vía DockerCompose que incluya el SQL de creación de las tablas y relaciones.
 - Programa PYTHON que cargue los datos del punto 1.
 - Archivo docker-compose.yml para implementar el contenedor de Postgresql.
 - NOTA: Al crear el contenedor Docker de Postgresql debe crear la BD a utilizar.

Pregunta 4: Considerando el modelo de datos de la imagen, genere consultas en SQL que permitan obtener:

*Supuesto: El valor de venta se calcula por producto como unidades_vendidas * precio_unitario*



1. Listado de ventas del mes actual (nombre_sucursal, nombre_vendedor, marca, nombre_producto, fecha_venta, unidades_vendidas, precio_unitario, valor_venta)
2. Ventas totales por sucursal, vendedor y marca, incluyendo los vendedores que no tuvieron ventas (nombre_sucursal, nombre_vendedor, marca, total_venta)
3. Productos con más de 1000 unidades vendidas en los últimos 2 meses (nombre_producto, marca, unidades_vendidas)
4. Productos sin ventas en el presente año (nombre_producto, marca)
5. De los productos sin ventas en el presente año, monto total de ventas en el año anterior (nombre_producto, marca, total_venta)