1/10/25

Manolo Pajaro Borras

Docente

Evaluacion Bases de datos II

15%

**Enunciado del Problema:**

Una tienda online necesita una base de datos robusta y escalable para gestionar sus clientes, productos y el ciclo completo de pedidos. Se requiere un diseño que permita almacenar la siguiente información clave:

1. **Clientes:**
   * Cada cliente debe tener un identificador único.
   * Se necesita almacenar su nombre, apellido, correo electrónico (que debe ser único), y su dirección completa (calle, ciudad, país).
2. **Productos:**
   * Cada producto debe tener un identificador único.
   * Se requiere guardar el nombre del producto, una descripción detallada, su precio y la cantidad disponible en stock.
3. **Pedidos:**
   * Cada pedido debe tener un identificador único.
   * Debe estar asociado a un cliente.
   * Se debe registrar la fecha y hora en que se realizó el pedido, su estado actual (ej. "Pendiente", "Procesado", "Enviado", "Entregado") y el total final del pedido.
4. **Detalles del Pedido:**
   * Para cada pedido, es esencial conocer los productos que lo componen, cuántas unidades de cada producto se pidieron y el precio unitario exacto de cada producto en el momento de la compra.

**Tareas a Realizar:**

**Parte 1: Bases de Datos Relacionales (SQL)**

1. **Diseño del Modelo Relacional:**
   * Crea un modelo entidad-relación lógico que represente las entidades Clientes, Productos, Pedidos y Detalle\_Pedido, identificando atributos y relaciones (1:N, N:M).
2. **Script SQL para Implementación:**
   * Desarrolla un script SQL completo que contenga:
     + La creación de la base de datos (si aplica a tu SGBD).
     + La definición de todas las tablas con sus columnas, tipos de datos apropiados, claves primarias y claves foráneas.
     + La inserción de al menos **tres (3) registros de ejemplo** para cada tabla, asegurando la coherencia referencial entre ellas.

**Parte 2: Conversión a NoSQL - MongoDB**

1. **Diseño del Modelo de Documentos:**
   * Modifica el diseño relacional para adaptarlo al modelo de documentos de MongoDB. Piensa en cómo optimizar las consultas comunes (ej. ver todos los pedidos de un cliente) mediante la incrustación de documentos.
2. **Script de Inserción para MongoDB:**
   * Crea un script con comandos de insertMany o insertOne de MongoDB para replicar los datos de prueba de la Parte 1 en las colecciones diseñadas, reflejando el nuevo modelo de documentos.

**Parte 3: Conversión a NoSQL - Apache Cassandra**

1. **Diseño de KeySpace y Tablas:**
   * Diseña un KEYSPACE y las tablas necesarias en Cassandra, pensando en el acceso a datos. Es crucial definir las claves primarias (PARTITION KEY y CLUSTERING KEYS) en función de las consultas esperadas (ej. obtener pedidos por cliente). Considera la desnormalización si es apropiado.
2. **Script CQL para Implementación e Inserción:**
   * Desarrolla un script CQL que incluya:
     + La creación del KEYSPACE.
     + La creación de todas las tablas de Cassandra.
     + La inserción de al menos los mismos registros de ejemplo de la Parte 1 en las tablas de Cassandra. Utiliza UUIDs para los identificadores.

**Parte 4: Consultas y Conectividad con Python**

1. **Consultas en MongoDB (Python):**
   * Escribe un script en Python que utilice la librería pymongo para conectarse a tu base de datos MongoDB y realizar las siguientes consultas:
     + a) Recuperar todos los pedidos realizados por un cliente específico (identificado por su correo electrónico).
     + b) Encontrar todos los productos con un stock inferior a un valor dado (ej. 15 unidades).
     + c) Calcular el valor total de todos los pedidos que se encuentran en estado "Pendiente".
2. **Consultas en Cassandra (Python):**
   * Escribe un script en Python que utilice la librería cassandra-driver para conectarse a tu cluster de Cassandra y realizar las siguientes consultas:
     + a) Recuperar todos los pedidos de un cliente específico (identificado por su UUID de cliente).
     + b) Obtener la información de un producto dado su UUID de producto.
     + c) Listar todos los clientes que se registraron en una ciudad específica. (Nota: Esto podría requerir un índice secundario si no está en la clave primaria).

**Entregables (Soluciones a entregar por el estudiante):**

* **Documento de Diseño:** Un documento que explique brevemente el modelo de datos para cada tipo de base de datos (relacional, MongoDB, Cassandra) y las decisiones de diseño tomadas.
* **Script SQL:** Archivo .sql con la creación de la base de datos relacional y la inserción de datos.
* **Script MongoDB:** Archivo .js o .json con los comandos de inserción para MongoDB.
* **Script Cassandra:** Archivo .cql con la creación del keyspace, tablas y la inserción de datos.
* **Scripts Python:** Dos archivos .py separados (uno para MongoDB y otro para Cassandra) con las consultas solicitadas, incluyendo los pasos para la conexión a las bases de datos. Cada consulta debe imprimir sus resultados de forma legible.

Deben subir este documento al Polivirtual con en formato zip o rar con el pantallazo de lo realizado así como el código de Python, En la hoja de respuestas que aparece en la parte inferior

**RESPUESTAS**