



Universidad San Carlos de Guatemala Facultad
de Ingeniería

Escuela de Ciencias y Sistemas

Manejo e Implementación de

Archivos Segundo Semestre 2021

Catedráticos: Ing. Álvaro Díaz, Ing. Oscar Paz, Ing. William Escobar

Tutores académicos: Bruno Coronado, Renato Flores, Elizabeth Zavala, Alan

Morataya

PRÁCTICA

INTRODUCCIÓN

La necesidad de tener sistemas escalables y óptimos depende principalmente de un buen diseño de base de datos. Si bien existen sistemas antiguos que funcionan así, con una metodología de almacenamiento de información en archivos planos con datos redundantes y con datos no atómicos, éstos a lo largo del tiempo llegan a impactar de forma considerable el rendimiento del propio sistema. Por lo anterior muchas organizaciones aceptan el reto de migrar sus sistemas a bases de datos formalizadas que les permita mejorar su rendimiento y su escalabilidad.

OBJETIVOS

- Diseñar y desarrollar un modelo relacional a partir del planteamiento de un problema y el análisis de un archivo de datos.
- Realizar una carga masiva de datos de la información proporcionada a una nueva plataforma mediante la creación de una tabla temporal distribuyendo la información de la carga a el modelo relacional propuesto.
- Generar consultas avanzadas en lenguaje SQL que cumplan con los reportes solicitados.

ENUNCIADO

La empresa “BlockBuster” es una franquicia de videoclubes, especializada en el alquiler de cine a través de tiendas físicas. Fue fundada en 1985 por David Cook y tuvo un gran crecimiento en los años 1990. Ésta empresa sufrió el famoso “golpe tecnológico” y se declaró en bancarrota en el año 2010 con una deuda de más de

1.00 millones de dólares. Debido a ese golpe, se vio en la necesidad de dejar a un lado los sistemas costosos y manejar nuevamente su información dentro de una hoja de cálculo, la cual, les está ocasionando muchos problemas al momento de querer obtener información de forma específica.

Ésta empresa se pone en contacto con usted para proporcionarle la hoja de cálculo donde actualmente manejan toda su información y usted pueda analizarla, interpretarla y poder llevar a cabo un diseño de base datos para poder manejar esa información de una forma más sencilla y poder generar reportes sobre ella que son de vital importancia para la franquicia. La información actualmente se encuentra segmentada de la siguiente forma:

a. Información de los clientes

Para la empresa es vital la información de sus clientes ya que es gracias a ellos que pueden seguir en pie. De los clientes se necesita almacenar su nombre, su apellido, su correo electrónico, su dirección, su fecha de registro en el sistema, identificar si el cliente se encuentra activo o no y su tienda de BlockBuster favorita o la que se le hace más cómodo llegar para rentar una película.

Para la empresa es necesario tener bien completas las direcciones de las personas y no basta con solo saber su dirección por defecto, también necesitan almacenar datos como el distrito de dicha dirección y su código postal. Debido a que la empresa recibe visitas de personas de diferentes países en sus diferentes tiendas, cada dirección que proporcione una persona está asociada a una ciudad en particular. De la ciudad se necesita saber su nombre y el país al que pertenece.

b. Información del negocio

Para la empresa es necesario también almacenar la información de todos sus empleados. De los empleados se necesita almacenar su nombre, apellido, dirección, correo electrónico, identificar si el empleado se encuentra activo o no, nombre de usuario, una contraseña que sirve para ingresar en las puertas de personal autorizado y la tienda en donde está asignado para trabajar. De las tiendas únicamente se desea almacenar su dirección. Es importante mencionar que las tiendas tienen a un empleado del personal como jefe de tienda, por lo que también es importante manejar esa información.

A cada empleado se le paga una cierta cantidad de dinero por cada rentado de película que hace un cliente. Para ello se debe almacenar información como la cantidad a pagar y la fecha de pago y, por supuesto, el empleado que realizó la venta. Es muy importante para la tienda saber en qué fecha fue rentada una película a un cliente y en qué fecha debe regresar el cliente dicha película a la tienda.

c. Información del inventario

Si bien la empresa se dedica a la renta de películas, lastimosamente no todas las películas que una vez entraron a la tienda se tienen en stock, por lo que para la empresa es importante manejar la información de su inventario de películas. El inventario es manejado por tiendas.

De las películas, la franquicia necesita almacenar información como su título, su descripción, el año en el que fue lanzada, la duración (en minutos) de la película, la cantidad de días en la que puede ser rentada la película, el costo de rentar la película, el costo que debe ser cargado al cliente si en caso regresa la película en mal estado y su clasificación, la cual puede ser: A, A+, B, B+15 o C.

Otra información importante para la empresa es el manejo de las categorías de las películas. Una película puede ser parte de diferentes categorías, por ejemplo, la película “Freddy vs Jason” pertenece a la categoría de “Terror” y “Thriller”. En cada película, participan diferentes actores, de los cuales se necesita almacenar información como su nombre y su apellido. Cada película es recibida en su idioma original, pero cada película puede tener diferentes traducciones en diferentes lenguajes, por lo que para la empresa es importante manejar también esta información.

REQUERIMIENTOS

a. Documentación

La empresa “Blockbuster” solicita que realice un documento de análisis que le permita determinar el alcance de la nueva plataforma. El documento deberá contener como mínimo una descripción de las reglas de normalización aplicadas al archivo de excel fuente. Así mismo deberá incluir el modelo relacional explicado detalladamente de tal forma que permita visualizar de mejor manera la estructurapropuesta y el cómo se almacenará la información.

b. Scripts

La empresa “Blockbuster” solicita que se entregue el script completo de la base de datos para la nueva plataforma en estado normalizado con su respectiva creación de tablas, constraint’s, llaves primarias, llaves candidatas únicas, check de referencias y demás consideraciones que usted considere necesario dentro del diseño del modelo relacional.

c. Carga de datos

La empresa “Blockbuster” le hace entrega de un archivo excel (csv) con toda la información de su sistema de forma desnormalizada. Se solicita que genere un script que permita cargar la información del archivo de exportación (csv) a una tabla temporal y con base al modelo entidad relación solicitado en el inciso (a) para luego ser ejecutado mediante el inciso (b) distribuyendo así toda la información hacia la nueva estructura propuesta. El archivo de carga se encuentra en la siguiente dirección: <https://tinyurl.com/vsa4cw9>

d. Reportería solicitada

La empresa “Blockbuster” tiene 20 consultas de uso frecuente. Deberá crear un script con las veinte consultas que se solicitan a continuación:

1. Mostrar la cantidad de copias que existen en el inventario para la película “Sugar Wonka”.
2. Mostrar el nombre, apellido y pago total de todos los clientes que han rentado películas por lo menos 40 veces.
3. Mostrar el nombre y apellido (en una sola columna) de los actores que contienen la palabra “SON” en su apellido, ordenados por su primer nombre.
4. Mostrar el nombre y apellido de los actores que participaron en una película cuya descripción incluye la palabra “crocodile” y “shark” junto con el año de lanzamiento de la película, ordenados por el apellido del actor en forma ascendente.
5. Mostrar el país y el nombre del cliente que más películas rentó así como también el porcentaje que representa la cantidad de películas que rentó con respecto al resto de clientes del país.
6. Mostrar el total de clientes y porcentaje de clientes por ciudad y país. El ciento por ciento es el total de clientes por país. (Tip: Todos los porcentajes por ciudad de un país deben sumar el 100%).
7. Mostrar el nombre del país, la ciudad y el promedio de rentas por ciudad. Por ejemplo: si el país tiene 3 ciudades, se deben sumar todas las rentas de

la ciudad y dividirlo dentro de tres (número de ciudades del país).

8. Mostrar el nombre del país y el porcentaje de rentas de películas de la categoría “Sports”. El porcentaje es sobre el número total de rentas de cada país.
9. Mostrar la lista de ciudades de Estados Unidos y el número de rentas de películas para las ciudades que obtuvieron más rentas que la ciudad “Dayton”.
10. Mostrar todas las ciudades por país en las que predomina la renta de películas de la categoría “Horror”. Es decir, hay más rentas que las otras categorías.

PRESENTACIÓN DE DATOS - API REST

Las consultas deben verse a través del uso de una API REST utilizando el framework Flask que está basado en el lenguaje Python.

La API estará conectada a PostgreSQL, las consultas serán visualizadas a través de HTTP Requests a los siguientes endpoints:

Acción	Endpoint
Mostrar consulta 1	/consulta1
Mostrar consulta 2	/consulta2
Mostrar consulta 3	/consulta3
Mostrar consulta 4	/consulta4
Mostrar consulta 5	/consulta5
Mostrar consulta 6	/consulta6
Mostrar consulta 7	/consulta7
Mostrar consulta 8	/consulta8
Mostrar consulta 9	/consulta9
Mostrar consulta 10	/consulta10
Eliminar datos de la tabla temporal	/eliminarTemporal
Elimina las tablas del modelo de datos	/eliminarModelo
Carga masiva de datos a tabla temporal	/cargarTemporal
Crear tablas del modelo y cargarle los datos	/cargarModelo

Las peticiones HTTP podrán ser ejecutadas a través de Chrome(GET) o a través de POSTMAN(GET, POST, PUT, DELETE).

ENTREGABLES

- Documentación de análisis solicitada. Debe tener el siguiente formato:
[MIA]Documentación_#CARNET.pdf
- Documentación (breve y concisa) sobre cómo realizó la carga de datos del archivo de excel (csv) a la tabla temporal. Debe tener el siguiente formato:
[MIA]CargaMasiva_#CARNET.pdf.
- Script (DDL) con el **borrado** y **creado** de la tabla temporal. Debe tener el siguiente formato: **[MIA]ScriptTT_#CARNET.sql**
- Script (DDL) con el **borrado** y **creado** de la base de datos propuesta en el modelo relacional. Debe tener el siguiente formato:
[MIA]ScriptMR_#CARNET.sql
- Script (DML) con los COUNT's de todas las tablas que utilizaron en su modelo relacional. Debe tener el siguiente formato:
[MIA]Conteos_#CARNET.sql
- Script (DML) con las diferentes consultas utilizadas para realizar la carga y distribución de la información en la nueva base de datos. Debe tener el siguiente formato: **[MIA]CargaDeDatos_#CARNET.sql**
- Script (DML) con las diferentes consultas solicitadas por la empresa que corresponden a la reportería. Debe tener el siguiente formato:
[MIA]Reportería_#CARNET.sql
- Carpeta con el código fuente de la API REST. **Formato: [MIA]API**

RESTRICCIONES

- La práctica es individual.
- El sistema operativo debe ser una distribución de Linux.
- La práctica debe ser implementada en una base de datos **PostgreSQL**, se puede hacer uso de servicios en la nube para su implementación.
- Se utilizará un repositorio de git para que suban su práctica y se habilitará una opción en UEDI para que puedan subir el link de su repositorio, los auxiliares de cada curso deberán tener acceso a los repositorios respectivos en cualquier momento de la duración del laboratorio, si no se cuenta con acceso se anulara la práctica.

- El servicio de hosting de Git (Github, Gitlab, Bitbucket, etc) queda a discreción del estudiante. Se calificará del último commit que suban a la hora estipulada
- La API REST debe ser desarrollada con Flask.
- No se recibirán prácticas después de la fecha y hora de entrega establecida.
- Toda carga masiva (de csv a tabla temporal) realizada con el asistente que proporcionan los DBMS's de PostgreSQL tendrán automáticamente 0 puntos en el área de carga masiva a tabla temporal.
- Toda carga de información realizada de forma incorrecta al modelo relacional (información inválida o no presente en el archivo csv) tendrán automáticamente 0 puntos en el área de carga de tabla temporal al modelorelacional.
- Copias de prácticas tendrán automáticamente nota de 0 puntos y se reportará a los involucrados a la Escuela de Ingeniería en Ciencias y Sistemas.

ENTREGA

- La entrega será por medio de la plataforma de UEDI.
- Tomar en cuenta que no habrá prórroga, por lo que se solicita al estudiante distribuir adecuadamente su tiempo para la elaboración de su práctica.
- Para subir el proyecto deberán crear un archivo comprimido con todos los entregables detallados anteriormente. El archivo debe tener el siguiente formato: **[MIA]Practica_#CARNET.zip**
- Fecha de entrega:
30 de septiembre de 2021 antes de las 11:59 p.m.