



PRÁCTICA NO. 1

1. OBJETIVOS

- **General**

- Utilizar consultas para presentar la información requerida de una base de datos relacional.

- **Específicos**

- Desarrollar la habilidad de generar consultas avanzadas en lenguaje SQL.
- Mostrar información específica de una base de datos.
- Aprender a realizar cargas masivas desde archivos separados por coma (csv) a una base de datos.
- Desarrollar el modelo entidad relación a partir de un conjunto de datos.
- Aprender a utilizar una API REST para hacer consultas a una base de datos.

2. DESCRIPCIÓN DE LA PRÁCTICA

El centro de datos "El Eje del Mundo" lo contrata a usted para realizar reingeniería al proceso de almacenamiento que actualmente manejan para guardar la información de las diferentes compañías que tiene a cargo ya que actualmente hay demasiada inconsistencia que puede provocarles serios problemas legales. El centro de datos hace el esfuerzo de exportar toda la información actual a un archivo delimitado por comas (csv) que usted tendrá a disposición para realizar el análisis correspondiente con el objetivo de eliminar redundancia de los datos y crear un nuevo modelo de base de datos estable para almacenar la información.

3. ESTRUCTURA DEL ARCHIVO CSV

A continuación, se muestra cómo está estructurado el archivo separado por comas que contiene los datos a cargar. Con esta información se debe crear el modelo de datos que permita realizar los reportes de la sección 5. El archivo puede ser descargado desde el siguiente enlace:

<https://drive.google.com/file/d/1akZTduJbmfp2-OLU0E3HTRquxmCMh0yy/view?usp=sharing>

Columna	Descripción
nombre_compania	Nombre de la compañía Tipo: texto
contacto_compania	Nombre de la persona con la que se tiene contacto dentro de la empresa. Tipo: texto
correo_compania	Correo electrónico de la compañía. Tipo: texto
telefono_compania	Teléfono de la compañía. Tipo: texto
tipo	Carácter que representa si los siguientes campos (nombre, correo, teléfono y campos de dirección) corresponde a un Cliente [C] o a un proveedor [P]. Tipo: carácter
nombre	Nombre del cliente ó proveedor. Tipo: texto
correo	Correo electrónico del cliente ó proveedor. Tipo: texto
teléfono	Teléfono del cliente ó proveedor. Tipo: texto
fecha_registro	Fecha en la que se registro al sistema el cliente ó proveedor. Tipo: fecha
dirección	Dirección del cliente ó proveedor. Tipo: texto
ciudad	Ciudad de donde es el cliente ó proveedor.

	Tipo: texto
codigo_postal	Código postal de la dirección del cliente ó proveedor. Tipo: numérico
región	Región correspondiente de la ciudad de donde es el cliente ó proveedor. Tipo: texto
producto	Nombre del producto que se compró/vendió según corresponda. Tipo: texto
categoria_producto	Categoría a la que pertenece el producto. Tipo: texto
cantidad	Número que representa la cantidad de producto adquirido/vendido. Tipo: numérico
precio_unitario	Costo que representa el adquirir una única unidad de un producto en específico. Tipo: decimal

4. SOLICITUD

Una vez analizado el archivo anterior, se solicita que usted diseñe un nuevo modelo de datos para almacenar toda esa información. El proceso es el siguiente:

1. Debe crear una tabla temporal en donde debe cargar toda la información del CSV.
2. Haciendo uso de consultas a la tabla temporal debe de llenar el modelo que usted propone para almacenar la información de forma correcta de modo que no exista redundancia de los datos.
3. Una vez lleno todo el modelo, debe realizar los reportes de la sección 5 los cuales serán evaluados para verificar que la información mostrada sea consistente y verídica.

5. REPORTES

Se deben de entregar los siguientes reportes:

1. Mostrar el nombre del proveedor, número de teléfono, número de orden, total de la orden por la cual se haya pagado la mayor cantidad de dinero.
2. Mostrar el número de cliente, nombre, apellido y total del cliente que más productos ha comprado.
3. Mostrar la dirección, región, ciudad y código postal hacia la cual se han hecho más solicitudes de pedidos y a cuál menos (en una sola consulta).
4. Mostrar el número de cliente, nombre, apellido, el número de órdenes que ha realizado y el total de cada uno de los cinco clientes que más han comprado productos de la categoría 'Cheese'.
5. Mostrar el número de mes de la fecha de registro, nombre y apellido de todos los clientes que más han comprado y los que menos han comprado (en dinero) utilizando una sola consulta.
6. Mostrar el nombre de la categoría más y menos vendida y el total vendido en dinero (en una sola consulta).
7. Mostrar el top 5 de proveedores que más productos han vendido (en dinero) de la categoría de productos 'Fresh Vegetables'.
8. Mostrar la dirección, región, ciudad y código postal de los clientes que más han comprado y de los que menos (en dinero) en una sola consulta.
9. Mostrar el nombre del proveedor, número de teléfono, número de orden, total de la orden por la cual se haya obtenido la menor cantidad de producto.
10. Mostrar el top 10 de los clientes que más productos han comprado de la categoría 'Seafood'.

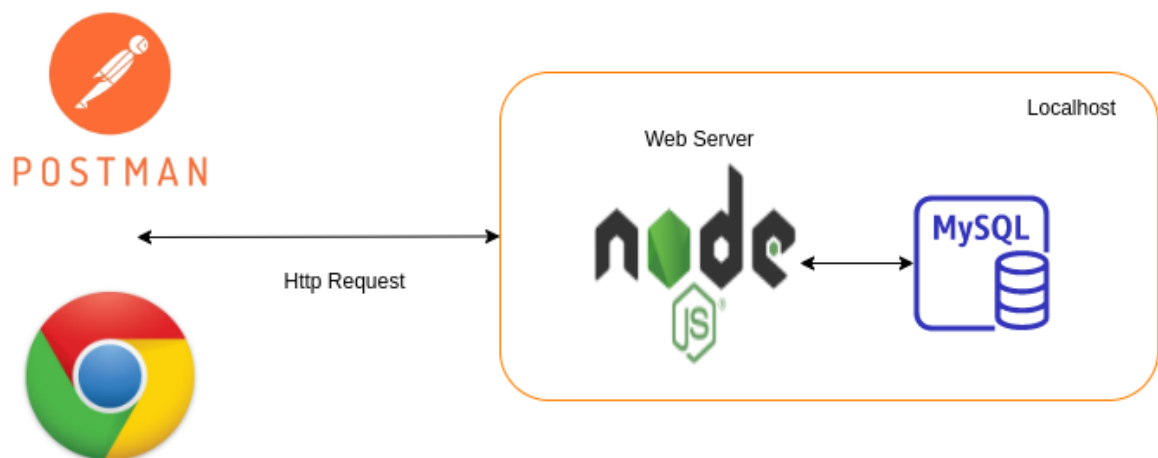
6. PRESENTACIÓN DE DATOS - API REST

Las consultas deben verse a través del uso de una API REST utilizando Node.JS, la api estará conectada con MySQL, 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).



REQUERIMIENTOS Y RESTRICCIONES

- El modelo debe ser desarrollado en MySQL en un sistema operativo basado en Linux.
- Deben diseñar el modelo relacional.
- No se debe separar en diferentes archivos csv el archivo proporcionado. Debe ser cargado únicamente el que fue proporcionado.
- Los reportes deben ser realizados a la entidad relación creada por el estudiante, reportes hechos a la tabla temporal serán anulados.
- La práctica es individual.
- Copias totales o parciales tendrán nota de 0 puntos y serán reportados a escuela.
- La API REST debe ser desarrollada con Node.JS
- Entregas tarde tendrán una penalización del 25% de la nota total.

ENTREGABLES

- Los entregables deben de ser los siguientes:
 - El modelo relacional propuesto en formato de imagen. Formato: **[MIA]ModeloPropuesto.png**
 - Script (con extensión .sql) que contiene las instrucciones DDL para crear el modelo relacional. Formato: **[MIA]InstruccionesDDL.sql**
 - Script (con extensión .sql) que contiene la carga masiva y todas las consultas necesarias para llenar el modelo relacional propuesto a partir de la tabla temporal. Formato: **[MIA]CargaDeDatos.sql**
 - Script (con extensión .sql) que contiene todas las consultas necesarias para realizar los reportes. Formato: **[MIA]Consultas.sql**
 - Carpeta con el código fuente de la API REST. Formato: **[MIA]API**
- Todos los archivos anteriormente mencionados deben de ir dentro de una carpeta comprimida con el siguiente formato:
[MIA]Practica1_#carnet.rar

FECHA DE ENTREGA

- Domingo 4 de octubre de 2020, antes de las 11:59 pm.
- La entrega será a través de la plataforma UEDI
- No habrá prórroga.