

Base de datos Bancaria en beta Informe

Universitario, Nicolas Angel Rojas Yañez

Universidad Simon Bolivar-sede 2 Cucuta

Base de datos-9883-T02

Profesora, Heidy Lisbeth Adarme Romero

Octubre 18,2024

Base de datos

Información Principal:

Nombre: BancoDB

Motor de base de datos: MYSQL

Total de tablas: 10

Nombre de ls tablas:

*Sucursal

*Departamento

*Cargo

*Cliente

*Empleado

*Sucursal_departamento

*Cuenta

*Transaccion

*Prestamo

*Tarjeta_Credito

Triggers de la base de datos

Existen dos triggers en la base de datos los cuales establecen condicionales distintas para los registros que le muestran al usuario algunas cosas que no puede hacer:

*Solo un gerente por sucursal

Tiene en cuenta la tabla empleado en la cual el campo gerente a causal de la conexión con sucursal establece esta condición

*Solo tres cuentas por cliente

Teniendo en cuenta la tabla Cuenta consulta el numero de cuentas y establece un condicional IF que le muestra al usuario si q va a ingresar una 4 cuenta no puede por la razón inicial

Conexiones

La base de datos fue conectada a través de un archivo Javascript

Database.js

Uso MYSLQ2 como librería para acceder a los datos de consulta

Creo la conexión como constante y uso el motor mysql con el método crear conexión, allí se almacenan los datos de acceso a la base de datos del banco y en caso de error muestra el mensaje 'Error al conectarse a la base de datos' + la depuración o el error por el cual no se pudo ingresar a los datos además de mostrar en el caso de funcionar mostrar el nombre de la base de datos y la configuración de esta.

Server.js

Contiene variables de definición para la administración y rutas sobre la base de datos y el uso de node.js incluyendo archivos json además el server contiene un llamado a la base de datos al definir su ruta, además de mostrar el puerto en el que se esta escuchando la base de datos para mostrarlo al conectar la base de datos con Node.js.

Métodos:

Obtener el usuario:

Alli se encuentra la ruta para ingresar desde el usuario llamándolo luego de la base de datos se hace el llamado a los datos de sucursal además de prevenir en caso de errores con el err.Message para mostrar el motivo por el cual esta fallando, tiene un condicional que comprueba si no hay sucursales y en caso de que no le indica al usuario que esta tabla se encuentra sin registros además de traducir a cadena json el arreglo que contiene los resultados de la consulta sql.

Obtener los clientes

Alli se encuentra la ruta hacia los clientes generando una consulta sql que obtiene todos los datos de la tabla Cliente y en caso de error mostrar el motivo además de revisar si existen o no clientes a través de el condicional if.

Index.html

Se creó un archivo html:5 para el index en el cual en el head se establece el estilo de fondo de la pagina de color azul acorde en el método body

Extensiones Head:

Body:

Se establece el color azul de fondo además de centrar textos y de usar de fuente Arial

#ListaClientes:

En esta se tiene una cuadrícula responsiva que ajusta las columnas en función del tamaño del contenedor además de ajustar el espaciado de las mismas

.Cliente:

En este se establecen los arreglos de Front-end o diseño de los datos del cliente al pasar el mouse acorde al método .Cliente:over hace que este dato se vuelva gris

Body del index

En este se hacen unas importaciones mediante el id para mostrar los datos de las consultas

Métodos

ObtenerNombreBd()

Hace solicitud a través del fetch a la base de datos por el puerto 3000 Para mostrar el nombre de la base de datos y teniendo en cuenta mediante el condicional if si ocurren errores mostrarlos al usuario

Mostrar Clientes()

Evalua si responde o no la base de datos para avisarle al usuario un posible error en la red además de limpiar la lista antes de mostrarla o en caso de que no hayan datos avisarle al usuario.

Actualizardatos()

Llama los métodos del body q luego son llamados por programa para mostrar sus datos además de tener un setInterval para actualizar los datos cada 5 segundos

Mapas de la base de datos

Contiene dos mapas, el conceptual y el Lógico

Mapa conceptual:

Este se puede ver abriéndose desde el sitio web draw.io (link del programa: draw.io (diagrams.net)) este archivo se llama Mapa conceptual Banco.drawio

Mapa Lógico:

este esta en formato Pdf y se llama Mapa Lógico

Base de datos

En formato SQL text file se encuentra la base de datos solo es cuestión de importarla llamada Base de datos banco

Script

En este se encuentran los comandos SQL para crear tablas y almacenar los datos se llama Banco_BD