

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Laboratorio Sistemas de Bases de Datos 1
Segundo Semestre 2023
Ing. Luis Fernando Espino
Aux. Edin Emanuel Montenegro Vásquez



Enunciado

Proyecto #1

Introducción

En la actualidad, muchas organizaciones se enfrentan al desafío de migrar de sistemas de almacenamiento de datos físicos a entornos virtuales debido a la falta de un sistema de información consolidado. Esta transición hacia lo digital requiere una cuidadosa planificación y diseño de una estructura eficiente que garantice la persistencia y facilidad de acceso a la información.

Un escenario común es el de las pequeñas empresas que inicialmente optan por gestionar sus datos mediante archivos de Excel. No obstante, a medida que la cantidad de datos aumenta con el tiempo, la gestión de estos archivos se vuelve compleja y laboriosa. El crecimiento en tamaño y complejidad de estos archivos dificulta las tareas de manejo y manipulación.

Además, otra problemática recurrente es la existencia de bases de datos que carecen de un proceso adecuado de modelado. Esto resulta en bases de datos ineficientes que dificultan

la realización de operaciones básicas con la información, convirtiendo el proceso en una tarea engorrosa.

En este contexto, resulta esencial abordar estos desafíos y considerar soluciones que permitan una transición exitosa hacia sistemas de información virtualizados y bases de datos eficientes, asegurando la integridad, accesibilidad y fluidez en la gestión de datos empresariales.

Objetivos

General:

- Dominar la habilidad de obtener información necesaria de una base de datos relacional mediante consultas efectivas

Específicos:

- Aprender a realizar cargas masivas desde archivos que son proporcionados.
- Aplicar los principios del modelado de bases de datos para la creación y optimización de estructuras eficientes.
- Diseñar y ejecutar consultas avanzadas, asegurando la creación de informes precisos y útiles a partir de la base de datos.
- Desarrollar una API para el consumo de la base de datos y mostrar resultados.

Descripción del problema

El Tribunal Supremo Electoral (TSE) tiene la responsabilidad crucial de recopilar los resultados de las elecciones de diversas mesas distribuidas en todo el territorio nacional. En este contexto, se busca su colaboración para implementar un sistema de bases de datos capaz de realizar consultas que permitan verificar la coherencia entre los reportes manuales y los generados por el sistema de bases de datos. El objetivo primordial es establecer un proceso transparente y confiable que garantice la integridad y precisión de los resultados electorales.

Para llevar a cabo esta tarea, el TSE le proporcionará información esencial sobre los candidatos, los partidos políticos y los ciudadanos que tomaron parte en el proceso electoral. Para las mesas de votación, contamos con un registro que se identifica mediante un número de identificación (ID) y el departamento de su ubicación geográfica.

En cuanto a los partidos políticos, dispondrá de detalles clave, como el nombre, las siglas y la fecha de fundación. Respecto a los candidatos, se le proveerá de información que incluye el nombre, la fecha de nacimiento, el partido al que están afiliados y el cargo que ocupan en la planilla del partido.

Además, tendrá acceso a la lista de ciudadanos que participaron en el proceso electoral. Esta lista contiene información importante, como el número de identificación personal (DPI), el nombre, la dirección, el número de teléfono, la edad y el género de los ciudadanos. Cabe destacar que es de suma importancia que trate toda esta información con la máxima profesionalidad y discreción, dado que se trata de datos sensibles y confidenciales.

Este proyecto se plantea como un plan piloto que busca simular y perfeccionar los procedimientos electorales, con el objetivo de lograr elecciones futuras más eficientes y transparentes.

Requerimientos del proyecto

Diseño:

Se debe realizar todo el proceso de modelado de la base de datos necesaria para su creación. Este proceso consta de:

1. Modelo conceptual.
2. Modelo lógico.
3. Modelo relacional (Diagrama ER).

El modelo lógico y relacional deben ser realizados en Oracle Data Modeler.

Funcionalidades:

Una vez analizados los requerimientos y el archivo de carga, se necesita que usted diseñe el modelo de datos para almacenar toda la información. Aunque este proceso queda totalmente a discreción del estudiante como realizar la carga y su modelo, se sugiere realizar el proceso siguiente:

1. Crear una tabla temporal en donde cargar toda la información del archivo fuente.
2. Haciendo uso de consultas a la tabla temporal debe de llenar el modelo que usted proporcione 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 REPORTERÍA los cuales serán evaluados para verificar que la información mostrada sea consistente y verídica.

Reportería:

Se deben entregar los siguientes reportes:

1. Mostrar el nombre de los candidatos a presidentes y vicepresidentes por partido (en este reporte/consulta se espera ver tres columnas: "nombre presidente", "nombre vicepresidente", "partido").
2. Mostrar el número de candidatos a diputados (esto incluye lista nacional, distrito electoral, parlamento) por cada partido.
3. Mostrar el nombre de los candidatos a alcalde por partido.
4. Cantidad de candidatos por partido (presidentes, vicepresidentes, diputados, alcaldes).
5. Cantidad de votaciones por departamentos.
6. Cantidad de votos nulos.
7. Top 10 de edad de ciudadanos que realizaron su voto.
8. Top 10 de candidatos más votados para presidente y vicepresidente (el voto por presidente incluye el vicepresidente).
9. Top 5 de mesas más frecuentadas (mostrar no. Mesa y departamento al que pertenece).
10. Mostrar el top 5 la hora más concurrida en que los ciudadanos fueron a votar.
11. Cantidad de votos por genero (Masculino, Femenino).

Consumo de la API:

Las consultas deben verse a través del uso de una API REST utilizando el lenguaje de su preferencia (entre los tres lenguajes que se mencionan más adelante), las consultas serán visualizadas a través de peticiones HTTP a los siguientes endpoints:

ACCION	NOMBRE 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
Mostrar consulta 11	/consulta11
Eliminar datos de la tabla temporal	/eliminarabtemp
Carga masiva de datos a tabla temporal	/cargartabtemp
Eliminar las tablas de la base de datos (modelo de datos)	/eliminarmodelo
Crear tablas del modelo	/crearmodelo
Cargar datos a modelo	/cargarmodelo

Las peticiones HTTP serán todas de tipo GET y podrán ejecutarse a través de cualquier asistente de pruebas como Postman, Thunder o cualquier navegador.

Entregables

- Modelo conceptual, lógico y Físico (ER)
- Script de la creación de base de datos, la carga del modelo y consultas creadas (.sql).
- Código fuente de la API desarrollada.
- Manual técnico (todo lo relacionado al modelo y creación de la DB)

Restricciones

- El motor de base de datos a utilizar puede ser SQL Server, MYSQL u ORACLE.
- La API puede ser realizada en Python, Node JS o Golang.
- Para el modelo lógico y relacional se debe utilizar la herramienta de Data Modeler de Oracle.
- El proyecto es individual.
- Sistema operativo es libre.
- Pueden utilizar Docker, una instancia local o la nube para su base de datos.
- El código y todo lo relacionado a su proyecto debe alojarse en un repositorio de GitHub agregando a su auxiliar como colaborador (**usuario github: edinfusion**).
- No se permiten ediciones o modificaciones después de la entrega.
- En caso de copias totales o parciales será anulado y reportado a escuela.

Entrega

- Fecha límite: domingo 17 de septiembre de 2023 hasta las 23:59 hrs.
- La entrega se hará vía UEDI por medio del enlace a su repositorio.