

FUNDAMENTOS DE BASES DE DATOS

Semestre: 2024-1

Grupo 7081

Profesor: Víctor Manuel Corza Vargas

Proyecto final

Requerimientos

Diseñar tres bases de datos sobre el universo de discurso (minimundo) que elijas, por ejemplo: El sistema para administrar la venta de boletos de una aerolínea, el sistema de administración de una escuela, el sistema para administrar las ventas y el inventario de una tienda en línea, entre muchos otros. Puedes apoyarte de un LLM (Large Language Model) como ChatGPT en todas las etapas del proyecto.

Características del reporte escrito

Por cada base de datos se debe entregar un reporte escrito en formato PDF que incluya los siguientes puntos:

1. Lista de requerimientos para la base de datos propuesta.
2. Modelo conceptual de la base de datos, respetando la nomenclatura de Peter Chen.
 - 2.a Utilizar el software de modelado de su preferencia.
3. Modelo relacional, deben incluirse:
 - 3.a Llaves primarias
 - 3.b Llaves foráneas
 - 3.c Utilizar el software de modelado de su preferencia.
4. Script completo y sin errores para la creación de todos los elementos que conforman el esquema de la base de datos.
 - 4.a El Script debe estar diseñado para la versión 14 de Postgres.
 - 4.b Deben estar contempladas todas las llaves primarias, llaves candidatas y llaves foráneas; todas las llaves foráneas deben contar con un trigger de integridad referencial (SET NULL, CASCADE o SET DEFAULT).
5. Script con las Instrucciones “Insert” que permitan poblar la base de datos (**Este Script no se debe incluir en el reporte final, sólo en los entregables**):
 - 5.a Se deben generar 100 registros para cada tabla.
 - 5.b Si para el buen funcionamiento de la base de datos se requieren más de 100 registros o menos de 100 registros en una tabla, se debe explicar claramente la razón, sólo en este caso sí se debe incluir un apartado en el reporte final.
6. Evidencia del funcionamiento de al menos 4 restricciones de integridad referencial.

-
- 6.a Tablas involucradas en la restricción.
 - 6.b FK de la tabla que referencia y PK de la tabla referenciada.
 - 6.c Justificación del trigger de integridad referencial elegido.
 - 6.d Instrucción UPDATE o DELETE que permita evidenciar que la restricción está funcionando.
 - 6.e Captura de pantalla con el resultado de la instrucción que muestre que la restricción está funcionando.
7. Evidencia del funcionamiento de al menos 3 restricciones check para “atributos” de varias tablas.
- 7.a Tabla elegida
 - 7.b Atributo elegido
 - 7.c Breve descripción de la restricción
 - 7.d Instrucción para la creación de la restricción.
 - 7.e Instrucción que permita evidenciar que la restricción esta funcionando.
 - 7.f Captura de pantalla con el resultado de la instrucción que muestre que la restricción está funcionando.
8. Evidencia de la creación de al menos tres dominios personalizados. Se deben utilizar restricciones check en la creación de los tres dominios.
- 8.a Tabla elegida
 - 8.b Atributo elegido
 - 8.c Breve descripción del dominio y de la restricción check propuesta.
 - 8.d Instrucción para la creación del dominio personalizado.
 - 8.e Captura de pantalla de la estructura de la tabla donde se muestre el dominio personalizado en uso.
9. Evidencia del funcionamiento de al menos 2 restricciones para “tuplas” en diferentes tablas (Unidad 8 Integridad, tema “Specifying Constraints on Tuples Using CHECK”)
- 9.a Tabla elegida
 - 9.b Breve descripción de la restricción.
 - 9.c Instrucción para la creación de la restricción.
 - 9.d Instrucción “Insert” o “Update” que permita evidenciar que la restricción esta funcionando.
 - 9.e Captura de pantalla con el resultado de la instrucción que muestre que la restricción está funcionando.
- 10. Consultas**
- Plantea 3 consultas que consideres relevantes para la base de datos propuesta.
Para cada consulta planteada, incluir en el reporte los siguientes incisos:
- 10.a Redacción clara de la consulta.
 - 10.b Código en lenguaje SQL de la consulta.
 - 10.c Ejecutar la consulta en Postgres e incluir una captura de pantalla con el resultado de la consulta.

Ejemplos de consultas

Suponiendo que se trabaja con una base de datos de administración escolar, se plantean las siguientes consultas de ejemplo.

Consulta 1

Listar las asignaturas ordenadas por índices de reprobación, del más alto al más bajo. Agregar una columna que muestre el número de reprobados.

Consulta 2

Listado de alumnos ordenados por promedio del más alto al más bajo y por generación. Agregar una columna que muestre el promedio de cada alumno

Consulta 3

Listado de profesores ordenado por el índice de alumnos reprobados por semestre.

11. Vistas

Plantea 3 vistas que consideres relevantes para la base de datos propuesta.

Para cada vista planteada, incluir en el reporte los siguientes incisos:

11.a Redacción clara de la vista planteada.

11.b Código en lenguaje SQL que permita crear la vista solicitada.

11.c Ejecutar el código para la creación de la vista en Postgres e incluir una captura de pantalla con la vista creada satisfactoriamente.

11.d Incluir un ejemplo que los evaluadores puedan ejecutar para verificar el funcionamiento de las vistas.

Ejemplos.

Suponiendo que se trabaja con una base de datos de administración escolar, se plantean las siguientes vistas de ejemplo.

Vista 1

Historial académico de los alumnos, la vista debe mostrar un listado de todos los cursos en los que un alumno ha estado inscrito, los cursos deben aparecer en orden cronológico, junto con las calificaciones obtenidas en cada curso. Se debe proporcionar un ID de alumno como parámetro. La vista debe incluir los siguientes campos:

Período	Clave Asignatura	Número de créditos	Nombre de la Asignatura	Calificación
---------	------------------	--------------------	-------------------------	--------------

Vista 2

Horario de clases de los profesores o ayudantes, debe especificarse un ID de profesor o un ID de ayudante y un período como parámetros. A continuación un ejemplo de la vista:

Lunes	Martes	Miércoles	Jueves	Viernes
7:00 – 9:00	7:00 – 9:00	7:00 – 9:00	7:00 – 9:00	7:00 – 9:00

Estructuras de datos		Estructuras de Datos		
9:00 – 11:00	9:00 – 11:00 Fundamentos de bases de datos	9:00 – 11:00	9:00 – 11:00 Fundamentos de bases de datos	9:00 – 11:00
11:00 – 13:00	11:00 – 13:00	11:00 – 13:00	11:00 – 13:00	11:00 – 13:00
13:00 – 15:00	13:00 – 15:00	13:00 – 15:00	13:00 – 15:00	13:00 – 15:00
15:00 – 17:00	15:00 – 17:00	15:00 – 17:00	15:00 – 17:00	15:00 – 17:00
17:00 – 19:00	17:00 – 19:00	17:00 – 19:00 Fundamentos de bases de datos	17:00 – 19:00	17:00 – 19:00 Fundamentos de bases de datos
19:00 – 21:00	19:00 – 21:00	19:00 – 21:00	19:00 – 21:00	19:00 – 21:00

Vista 3

Horario de clases de los alumnos, debe escribirse un ID de alumno y un período como parámetro. A continuación un ejemplo de la vista:

Lunes	Martes	Miércoles	Jueves	Viernes
7:00 – 9:00	7:00 – 9:00	7:00 – 9:00	7:00 – 9:00	7:00 – 9:00
9:00 – 11:00	9:00 – 11:00	9:00 – 11:00	9:00 – 11:00	9:00 – 11:00
11:00 – 13:00	11:00 – 13:00	11:00 – 13:00	11:00 – 13:00	11:00 – 13:00
13:00 – 15:00	13:00 – 15:00	13:00 – 15:00	13:00 – 15:00	13:00 – 15:00
15:00 – 17:00 Análisis de algoritmos	15:00 – 17:00 Computación distribuida	15:00 – 17:00 Análisis de algoritmos	15:00 – 17:00 Computación distribuida	15:00 – 17:00
17:00 – 19:00	17:00 – 19:00 Inglés V	17:00 – 19:00 Fundamentos de bases de datos	17:00 – 19:00 Inglés V	17:00 – 19:00 Fundamentos de bases de datos
19:00 – 21:00 Lenguajes de Programación	19:00 – 21:00 Lógica Computacional	19:00 – 21:00 Lenguajes de Programación	19:00 – 21:00 Lógica Computacional	19:00 – 21:00

Fecha de entrega

Lunes 11 de diciembre de 2023

Entregables por cada base de datos diseñada.

1. Reporte escrito en formato PDF. El reporte debe estar en orden, respetando la numeración especificada en el planteamiento del proyecto.
2. Archivo de texto que incluya el código en lenguaje SQL con la solución a los puntos 10 y 11 del reporte. El archivo debe tener la extensión SQL y puede crearse en el editor de textos de tu preferencia.

-
3. Script SQL completo y sin errores para la creación de todos los elementos que conforman el esquema de la base de datos. Verificar que sea compatible con la versión 14 de Postgres.
 4. Script SQL con las instrucciones Insert que permitan poblar la base de datos. Verificar que sea compatible con la versión 14 de Postgres.
 5. Archivo de texto(No captura de pantalla) con los datos de conexión a la plataforma ElephantSQL:
 - (a) Usuario y base de datos (deben ser iguales).
 - (b) Servidor.
 - (c) Contraseña.

Instrucciones de entrega:

El proyecto puede realizarse en equipos de máximo 3 personas.

La entrega se realizará a través de la plataforma Google Classroom, en la asignación que corresponda al Proyecto Final.

Cada integrante del equipo deberá añadir 5 archivos correspondientes a los entregables de cada base de datos propuesta. Si se entregan tres bases de datos, entonces serán 15 archivos los que debe enviar cada integrante.

Al subir todos los archivos, no olvidar hacer clic en la opción "Marcar como tarea completada" para completar el envío de su proyecto.