

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3088 – Base de datos I

Sección 20

Ing. Bacilio Bolaños



Laboratorio No. 2

Leonardo Dufrey Mejía Mejía, 23648

María José Girón Isidro, 23559

GUATEMALA, 17 de mayo de 2025

Laboratorio No. 2

Link de GitHub: <https://github.com/DufreyM/Laboratorio2-BD1.git>

1. ¿Qué ventajas encontraron al encapsular lógica en funciones en lugar de repetir consultas SQL?

La ventaja principal como todo en el mundo de la programación es poder reutilizar cierta cantidad de código, con esto a largo plazo o en proyectos muy grandes se mejora la legibilidad y el mantenimiento de la base de datos. Además de que se puede empezar a ver cómo mejorar la seguridad.

2. ¿Qué criterios usaron para decidir cuándo implementar una función y cuándo una vista?

Las vistas por lo aprendido en clase se utilizan cuando los datos no varían por medio de cálculos intermedios mientras que las funciones sí, por lo que utilizamos funciones cuando el resultado podía variar según cierto rango como por ejemplo los tipos de sangre, pero utilizamos una vista para mostrar toda la información del personaje.

3. ¿Qué limitaciones encontraron al trabajar con procedimientos almacenados en comparación con funciones?

Los procedimientos no pueden usarse directamente con un SELECT, lo que puede dificultar su integración con la curva de aprendizaje, lo cual empieza a tener limitaciones en cuanto a la escalabilidad.

4. ¿Creen que el trigger que implementaron garantiza la integridad de los datos en todos los escenarios posibles? Justifiquen su respuesta.

El trigger garantiza la integridad de los datos cuando no tienen modificaciones, es decir lo creamos para que cubra las fechas de campaña o cuando la donación se aprueba que genere la evaluación, pero por ejemplo cuando podrían existir modificaciones cuando la campaña ya inicio porque este tipo de campañas por lo que tenemos entendido no son de un día.

5. ¿Cómo adaptarían su solución para que escale en una base de datos con millones de registros?

Depende, implementaríamos índices en las columnas clave como los id o las fechas de donación para poder realizar consultas más fácilmente, pero creo que para evitar la saturación se crearían tablas por año o por un rango de fechas útiles para los médicos o las personas encargadas.

6. ¿Qué escenarios podrían romper su lógica actual si no existiera el trigger?

Se perdería la automatización de las evaluaciones al aprobar una donación, lo que podría generar inconsistencias entre las tablas de las donaciones y las evaluaciones por reportes incorrectos y decisiones erróneas.

7. ¿Qué dificultades enfrentaron al definir funciones que devuelven conjuntos de resultados?

La nueva estructura del returns table, nos generó problemas por los nombres y las columnas de las columnas que habíamos definido al inicio.

**8. ¿Consideran que su diseño sería compatible con una arquitectura de microservicios?
¿Por qué sí o por qué no?**

Sí, el diseño es compatible si se abstraen las funciones vistas y procedimientos en una capa de acceso a datos, pero sería importante migrar la lógica de la aplicación, pero se pueden hacer operaciones pequeñas internas o pequeños reportes.

9. ¿Cómo reutilizarían las vistas que definieron en reportes o en otros sistemas?

Creo que lo más útil sería con fines educativos o profesionales que puedan ser utilizadas como APIs que expongan los datos en otras aplicaciones con el acceso a la información, evitando duplicar este tipo de lógica en cada sistema.

10. ¿Qué aprendieron sobre la separación entre la lógica de negocio y la lógica de persistencia al hacer este laboratorio?

La lógica de negocio como las reglas de validación pueden residir en la aplicación, pero se deben tener capas de seguridad como los triggers y las diferentes funciones para evitar sobrecarga en alguna capa, manteniendo buena lógica.