

Universidad del Valle de Guatemala

Facultad de Ingeniería

Departamento de Ciencias de la Computación

Cristian Túnchez - 231359



Base de Datos 1

### Proyecto 3: Reflexión

---

Instrucciones: Cada integrante entregará un documento individual que analice críticamente la base de datos construida. La reflexión debe responder de forma justificada las siguientes preguntas.

1. ¿Qué criterios usaron para decidir qué entidades y relaciones debían formar parte del modelo? Justifiquen cómo identificaron qué debía representarse en la base de datos y qué decisiones tomaron para simplificar o abstraer ciertos aspectos del sistema real.

Para definir las entidades y relaciones, partimos del análisis funcional del sistema: entendimos que el propósito era gestionar reservas, contratos, vehículos y clientes en un entorno de alquiler. Usamos técnicas de modelado como diagramas entidad-relación y mapeamos directamente conceptos clave del dominio (por ejemplo, Cliente, Vehículo, Contrato, Reserva) como entidades. Las relaciones surgieron de interacciones naturales entre estas entidades, como una Reserva asociada a un Cliente y un Vehículo, o un Contrato ligado a una Reserva.

En algunos casos hicimos abstracciones para simplificar el modelo. Por ejemplo:

- Centralizamos información de ubicación en una tabla Address, para ser reutilizada por varias entidades.

- Separamos Model, Manufacturer, y Vehicle\_Type para evitar redundancia y facilitar mantenimiento.

Este enfoque nos permitió mantener un diseño limpio, reutilizable y extensible.

2. ¿Qué tan adecuadas fueron las claves primarias y foráneas que definieron en su diseño?

Evalúen si estas claves facilitaron las consultas, mantuvieron la integridad de los datos y permitieron modelar correctamente las dependencias entre tablas.

Las claves primarias fueron apropiadamente definidas como serial (autoincrementales), lo cual nos facilitó la generación de registros únicos y simplificó la indexación. Las claves foráneas se usaron consistentemente para modelar dependencias entre tablas, y gracias a ello:

- Se mantuvo la integridad referencial.
- Las consultas con JOINS fueron claras y eficientes.
- Fue posible representar relaciones N:M (como Model\_Color) y 1:N sin ambigüedades.

Sin embargo, detectamos que en algunos casos (por ejemplo, phone\_number), las claves no numéricas requerían más cuidado en su tipo de dato y restricciones únicas.

3. ¿En qué medida aplicaron la normalización? ¿Qué beneficios y limitaciones experimentaron? Aplicar 1FN, 2FN y 3FN fue suficiente o si surgieron situaciones donde tuvieran que decidir entre rendimiento y diseño teórico.

Aplicamos las tres primeras formas normales:

- 1FN: Evitamos atributos multivaluados o compuestos.

- 2FN: Separando información dependiente de claves primarias compuestas (por ejemplo, Vehicle\_Maintenance).
- 3FN: Evitamos dependencias transitivas separando manufacturer, color, tipo de vehículo, etc.

Esto trajo beneficios importantes:

- Eliminamos redundancia
- Facilitamos el mantenimiento de datos
- Aumentamos la coherencia del modelo

Como limitación, notamos que, al momento de realizar consultas complejas, se requerían múltiples JOINS, lo cual podría impactar el rendimiento en una base de datos con millones de registros. No obstante, preferimos seguir un diseño teóricamente correcto sobre una optimización prematura.

4. ¿Qué restricciones y reglas del negocio implementaron directamente en la base de datos y por qué? Describe el uso de CHECK, DEFAULT, NOT NULL, UNIQUE, claves foráneas y triggers, y justifica su implementación.

Implementamos varias restricciones directamente en la base de datos:

- NOT NULL para campos obligatorios como fechas o identificadores.
- UNIQUE en correos electrónicos y licencias para evitar duplicados.
- CHECK en campos como payment\_method y amount para validar rangos y opciones válidas.
- DEFAULT en fechas de registro para registrar automáticamente la fecha actual.

Estas restricciones:

- Ayudan a capturar errores de validación antes de que lleguen al backend.
- Simplifican la lógica de aplicación.
- Protegen los datos incluso si múltiples aplicaciones acceden a la base.

No usamos triggers aún, pero podrían ser útiles para auditoría o cálculo automático de estados.

5. ¿Qué ventajas o desventajas identificas del modelo que construyeron al momento de hacer consultas complejas? Piensen en consultas con múltiples filtros, joins, subconsultas o agrupaciones, y comenten si el modelo fue flexible y escalable.

Ventajas:

- El modelo es muy modular y relacional, lo que permite escribir consultas robustas con JOIN, GROUP BY, y subconsultas.
- La estructura facilita filtrar datos en tiempo real, útil para reportes dinámicos.

Desventajas:

- Algunas consultas requieren múltiples JOINS, especialmente aquellas que cruzan varias relaciones (ej. historial de pagos por cliente con multas y contratos).
- Para consultas agregadas de alto volumen, se podría necesitar indexación adicional o vistas materializadas.

6. ¿Qué cambiarían en el diseño de la base de datos si tuvieran que escalar este sistema a un entorno de producción? Reflexionen sobre aspectos como volumen de datos, rendimiento, integridad y escalabilidad.

En un entorno real de alta carga, haría algunos cambios clave:

- Indexación estratégica en campos de búsqueda frecuentes (fecha, estado, id de cliente).
- Uso de particiones de tablas si el volumen crece significativamente (ej. Reservations por año).
- Evaluar cachés para reportes frecuentes.
- Agregar auditoría o bitácoras de cambio (trigger o tabla histórica).
- Considerar ENUMs o catálogos para campos con valores limitados como estado o método de pago.
- Añadir validación de integridad temporal (por ejemplo, que no se superpongan contratos para el mismo vehículo).