Ihan Marroquin - 23108

Reflexión Individual

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.

Identificamos como entidades todo aquello con ciclo de vida propio y relevancia en los reportes: estudiantes, instructores, cursos, módulos, lecciones, inscripciones, evaluaciones y progreso. Para las relaciones, reflejamos la realidad de inscripción (estudiante–curso), contenido (curso–módulo–lección) y evaluaciones (curso–estudiante) usando tablas de cruce cuando fue necesario, simplificamos elementos secundarios (por ejemplo, recursos y comentarios) para evitar sobreabstracción, y agrupamos atributos multivaluados (contactos de estudiantes) en tablas aparte, manteniendo el foco en la información necesaria para la reportería.

2. ¿Qué tan adecuadas fueron las claves primarias y foráneas que definieron en su diceño? Evalúen si estas claves facilitaron las consultas, mantuvieron la integridad de los datos y permitieron modelar correctamente las dependencias entre tablas.

Las claves primarias garantizan unicidad y facilitan índices automáticos; usar IDs numéricos sencillos agiliza los joins, las foráneas apuntan directamente a la tabla padre y llevan ON DELETE CASCADE o RESTRICT según convenga, lo que mantiene la integridad referencial y simplifica consultas con múltiples joins, haciéndolo así, este esquema permitió formular SQL claros y eficientes sin necesidad de condiciones complejas para mantener consistencia de datos.

3. ¿En qué medida aplicaron la normalización? Qúe 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 1FN al eliminar atributos repetitivos (por ejemplo, contactos de estudiantes en tabla separada), 2FN al mover atributos dependientes únicamente de parte de la clave compuesta y 3FN al extraer dependencias transitivas (por ejemplo, promedio de estudiante en inscripciones), haciendo que esto, se evitó anomalías de inserción y actualización, la única desventaja teórica es un mayor número de joins, pero consideramos aceptable el coste en un entorno de bases de tamaño medio.

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 CHECK para rangos de progreso y calificación, DEFAULT para fechas y estados iniciales, NOT NULL donde los datos son esenciales (nombre, títulos) y UNIQUE para correos y combinaciones únicas (estudiante–curso).

Los triggers actualizan automáticamente promedios y estados de inscripción, garantizando que la lógica crítica resida en la base de datos y no dependa de la aplicación, esto fortalece la validez y coherencia interna de los datos.

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.

El modelo facilita joins semánticos claros y ofrece tablas enfocadas en dimensiones clave (tiempo, progreso, ingresos) que agilizan agregaciones y filtros y gracias a las claves bien definidas y los índices, las consultas con ranges y agrupaciones (HAVING, BETWEEN) son rápidas, más sin embargo, en escenarios con millones de registros, el elevado número de joins puede penalizar el rendimiento, por lo que contemplaríamos materializar vistas o añadir índices compuestos.

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 volúmen de datos, rendimiento, integridad y escalabilidad.

Para un entorno con posible gran volumen de datos:

- Particionamiento de tablas sensibles por fecha o curso.
- Materialización de vistas para reportes pesados y cache en la capa de aplicación.
- Monitoreo y auditoría para mayor eficiencia operativa.