

Sesión N° 17

Tema (contenido) Técnicas de revisión de modelos y diseños de bases de datos.

Fecha Lunes, 03 noviembre, 2025 17:00

I INTRODUCCIÓN

Las técnicas de revisión de modelos y diseños de bases de datos son fundamentales para garantizar que una base de datos sea eficiente, coherente, escalable y alineada con los requerimientos del negocio.

1. Revisión del Modelo Conceptual

- Validación con usuarios: Asegurarse de que el modelo refleja correctamente las reglas del negocio.
- Chequeo de entidades y relaciones: Verificar que todas las entidades necesarias estén representadas y que las relaciones sean correctas.
- Cardinalidades y restricciones: Confirmar que las cardinalidades (uno a uno, uno a muchos, muchos a muchos) estén bien definidas.

2. Revisión del Modelo Lógico

- Normalización: Aplicar las formas normales (1FN, 2FN, 3FN, etc.) para evitar redundancias y anomalías.
- Integridad referencial: Verificar claves primarias y foráneas, asegurando que las relaciones estén bien estructuradas.

- Tipos de datos: Evaluar si los tipos de datos asignados son los más adecuados para cada atributo.

3. Revisión del Modelo Físico

- Índices: Evaluar la necesidad de índices para mejorar el rendimiento de consultas.
- Particionamiento: Considerar la división de tablas grandes para mejorar la eficiencia.
- Espacio de almacenamiento: Analizar el uso de espacio y prever crecimiento futuro.

4. Técnicas de Validación y Verificación

- Revisión por pares (peer review): Otros diseñadores o desarrolladores revisan el modelo para detectar errores o mejoras.
- Prototipado: Crear una versión preliminar de la base de datos para probar su funcionamiento.
- Pruebas de carga y rendimiento: Simular escenarios reales para evaluar la eficiencia del diseño.

5. Herramientas de Apoyo

- Modeladores como ERwin, MySQL Workbench, Oracle Data Modeler
- Sistemas de control de versiones para modelos
- Documentación detallada del modelo

Ejemplo 1: Revisión del Modelo Conceptual

Caso: Sistema de gestión de biblioteca

Modelo inicial:

- Entidades: Libro, Autor, Usuario
- Relaciones: Usuario presta Libro

Errores detectados:

- Falta la entidad Préstamo, que debería registrar fecha de préstamo y devolución.
- Relación entre Libro y Autor es de muchos a muchos, pero no está representada correctamente.

Corrección:

- Crear entidad Préstamo con atributos: fecha_prestamo, fecha_devolución.
- Crear tabla intermedia LibroAutor para representar la relación muchos a muchos.

Ejemplo 2: Revisión del Modelo Lógico

Caso: Base de datos de empleados

Modelo inicial:

Empleado(id, nombre, departamento, jefe_id)

Errores detectados:

- Redundancia: el campo departamento se repite para muchos empleados.
- Falta de normalización: mezcla de datos de empleados y departamentos.

Corrección:

- Separar en dos tablas:
 - Empleado(id, nombre, jefe_id, departamento_id)
 - Departamento(id, nombre)
- Aplicar 2FN y 3FN para eliminar dependencias parciales y transitivas.

Ejemplo 3: Revisión del Modelo Físico

Caso: Base de datos de ventas

Modelo inicial:

- Ventas(id, fecha, cliente_id, producto_id, cantidad, total)

Errores detectados:

- No hay índices en cliente_id ni producto_id, lo que ralentiza las consultas.
- El campo total se calcula como cantidad × precio, pero se almacena directamente.

Corrección:

- Crear índices en cliente_id y producto_id.
- Eliminar el campo total si se puede calcular en tiempo de consulta, para evitar inconsistencias.

Ejemplo 4: Revisión por Pares

Caso: Modelo de base de datos para una tienda en línea

Proceso:

- Otro analista revisa el modelo y detecta que:
- No se contemplan devoluciones.
- Las direcciones de envío están dentro de la tabla de clientes, lo que complica múltiples direcciones.

Corrección:

- Agregar entidad Devolución.
- Crear tabla Dirección relacionada con Cliente (uno a muchos).

Lista de verificación completa para revisar modelos y diseños de bases de datos, organizada por etapas. Puedes usarla como guía para validar tus propios proyectos o como herramienta de auditoría en equipos de desarrollo.

Lista de Verificación para Revisión de Modelos de Bases de Datos

1. Modelo Conceptual

- [] ¿Todas las entidades relevantes del negocio están representadas?
- [] ¿Las relaciones entre entidades están correctamente definidas?
- [] ¿Las cardinalidades (uno a uno, uno a muchos, muchos a muchos) son precisas?
- [] ¿Se han identificado correctamente los atributos clave de cada entidad?
- [] ¿Se han considerado las reglas del negocio en el diseño?

2. Modelo Lógico

- [] ¿El modelo está normalizado (al menos hasta 3FN)?
- [] ¿Se han eliminado redundancias y dependencias innecesarias?
- [] ¿Las claves primarias están bien definidas?
- [] ¿Las claves foráneas están correctamente establecidas para mantener la integridad referencial?
- [] ¿Los tipos de datos asignados a cada atributo son adecuados?
- [] ¿Se han definido restricciones como NOT NULL, UNIQUE, CHECK, etc.?

3. Modelo Físico

- [] ¿Se han creado índices en campos utilizados frecuentemente en búsquedas?
- [] ¿Se ha considerado el uso de particiones o segmentación de tablas grandes?
- [] ¿El diseño contempla el crecimiento futuro de los datos?
- [] ¿Se han evaluado las opciones de almacenamiento y rendimiento?

- [] ¿Se han definido procedimientos de respaldo y recuperación?

4. Validación y Verificación

- [] ¿Se ha realizado una revisión por pares del modelo?
- [] ¿Se han hecho pruebas de carga y rendimiento?
- [] ¿Se ha validado el modelo con usuarios o expertos del dominio?
- [] ¿Se ha documentado adecuadamente el modelo y sus decisiones?
- [] ¿Se ha probado el modelo con datos reales o simulados?

5. Herramientas y Buenas Prácticas

- [] ¿Se ha utilizado software de modelado (ERwin, Workbench, etc.)?
- [] ¿El modelo está versionado y controlado?
- [] ¿Se han aplicado estándares de nomenclatura y estilo?
- [] ¿Se ha considerado la seguridad y los permisos de acceso?

Ejercicio

Vamos a construir y revisar paso a paso el diseño de una base de datos para una **biblioteca**, comenzando desde el **modelo conceptual** hasta el **modelo físico**. A lo largo del proceso, aplicaremos la lista de verificación que se presentó antes.

1. Modelo Conceptual

Objetivo del sistema:

Gestionar préstamos de libros, usuarios, autores y control de inventario en una biblioteca.

Entidades principales:

- **Libro**
- **Autor**
- **Usuario**
- **Préstamo**
- **Ejemplar** (copia física de un libro)
- **Categoría**

Relaciones:

- Un **Libro** puede tener uno o varios **Autores** (relación muchos a muchos).
- Un **Libro** pertenece a una **Categoría**.
- Un **Libro** puede tener varios **Ejemplares**.
- Un **Usuario** puede realizar varios **Préstamos**.
- Un **Préstamo** se asocia a un **Ejemplar** y a un **Usuario**.

✓ **Validación conceptual:**

- [x] Entidades clave identificadas
- [x] Relaciones bien definidas
- [x] Cardinalidades claras
- [x] Reglas del negocio representadas

 **2. Modelo Lógico (Tablas y atributos)**

Tablas propuestas:

Autor(id_autor, nombre, nacionalidad)

Libro(id_libro, titulo, año_publicacion, id_categoria)

Categoría(id_categoria, nombre)

LibroAutor(id_libro, id_autor) -- relación muchos a muchos

Ejemplar(id_ejemplar, id_libro, estado)

Usuario(id_usuario, nombre, correo, tipo_usuario)

Prestamo(id_prestamo, id_usuario, id_ejemplar, fecha_prestamo, fecha_devolucion, devuelto)

✓ **Validación lógica:**

- [x] Aplicada 3FN (sin redundancias ni dependencias transitivas)
- [x] Claves primarias y foráneas definidas
- [x] Tipos de datos adecuados (por definir en modelo físico)
- [x] Integridad referencial asegurada

3. Modelo Físico (Detalles técnicos)

Ejemplo de definición en SQL:

```
CREATE TABLE Libro (
    id_libro INT PRIMARY KEY,
    titulo VARCHAR(255) NOT NULL,
    año_publicacion INT,
    id_categoria INT,
    FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria)
);
```

```
CREATE TABLE Prestamo (
    id_prestamo INT PRIMARY KEY,
    id_usuario INT,
    id_ejemplar INT,
    fecha_prestamo DATE,
    fecha_devolucion DATE,
    devuelto BOOLEAN,
```

FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),

FOREIGN KEY (id_ejemplar) REFERENCES Ejemplar(id_ejemplar)

);

Optimización:

- Índices en id_usuario, id_ejemplar, id_libro para mejorar búsquedas.
- Restricciones NOT NULL y CHECK para campos como devuelto.
- Considerar auditoría de préstamos (historial).

Validación física:

- [x] Índices definidos
- [x] Restricciones aplicadas
- [x] Escalabilidad considerada

Ejercicio Propuesto

Caso de Estudio: Biblioteca Universitaria "San Marcos"

Contexto:

La biblioteca de la Universidad San Marcos desea modernizar su sistema de gestión. Actualmente, todo se maneja en hojas de cálculo y formularios físicos. El nuevo sistema debe permitir registrar libros, autores, préstamos, devoluciones, usuarios y controlar el inventario de ejemplares.

Requerimientos funcionales:

1. Un libro puede tener uno o varios autores.
2. Cada libro puede tener múltiples ejemplares físicos.
3. Los usuarios pueden ser estudiantes, docentes o administrativos.
4. Un usuario puede tomar prestado hasta 3 ejemplares a la vez.
5. Cada préstamo debe registrar la fecha de préstamo y la fecha de devolución.
6. Se debe poder consultar el historial de préstamos por usuario y por libro.
7. Los libros deben clasificarse por categoría (ciencia, literatura, historia, etc.).
8. Se debe registrar el estado de cada ejemplar (disponible, prestado, dañado, perdido).

Actividades propuestas:

1. Modelo Conceptual

- Identifica las entidades principales y sus atributos.
- Define las relaciones entre entidades y sus cardinalidades.
- Representa el modelo en un diagrama entidad-relación (puede ser textual o gráfico).

2. Modelo Lógico

- Transforma el modelo conceptual en tablas.
- Aplica normalización hasta 3FN.
- Define claves primarias y foráneas.
- Asegúrate de que no haya redundancias ni dependencias innecesarias.

3. Modelo Físico

- Define los tipos de datos para cada campo.
- Establece restricciones (NOT NULL, UNIQUE, CHECK).

- Propón índices para mejorar el rendimiento.
- Considera cómo manejar el crecimiento de datos y la seguridad.

4. Validación

- Usa la lista de verificación que vimos antes para revisar tu diseño.
- Simula una revisión por pares: intercambia tu modelo con otro compañero o revísalo tú mismo con mirada crítica.
- Propón al menos dos consultas SQL que el sistema debería poder responder (por ejemplo: "¿Qué libros tiene prestados el usuario X?").

Preguntas de reflexión:

- ¿Qué decisiones tomaste al modelar la relación entre libros y autores?
- ¿Cómo garantizas que un usuario no pueda tomar más de 3 libros?
- ¿Qué harías si se quiere agregar una funcionalidad de reservas en el futuro?

Bibliografía

- "Fundamentos de Bases de Datos" de Abraham Silberschatz, Henry F. Korth y S. Sudarshan
- "Sistemas de Bases de Datos: un enfoque práctico" de Thomas M. Connolly y Carolyn Begg
- "Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación" de Dolores Cuadra, Elena Castro, Ana M. Iglesias
- "Tecnología y Diseño de Bases de Datos" de Marcos, C. Calero y B. Vela
- <https://docs.oracle.com/en/database/>