

PROMETEO

Unidad 1: Fundamentos de bases de datos

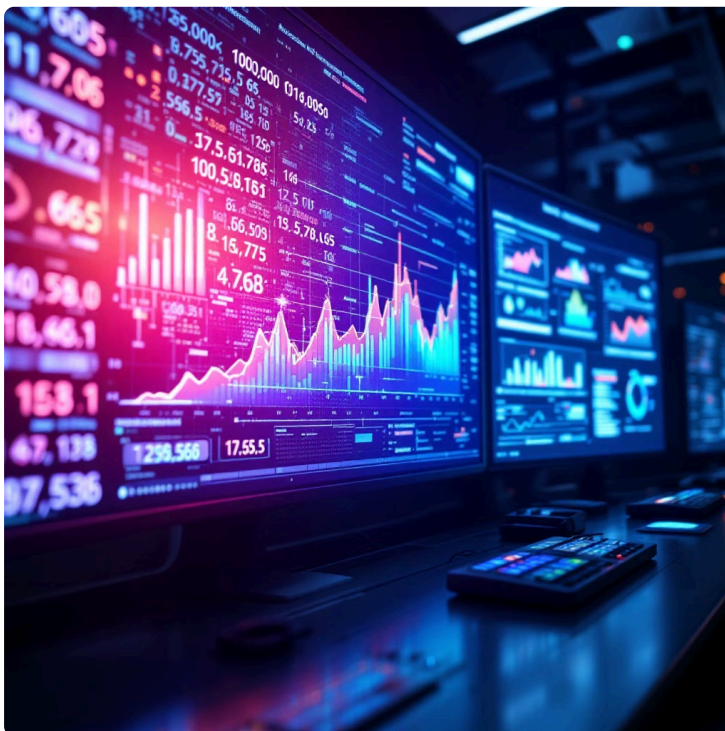
El proceso de transformación de datos en información no es automático ni trivial. Requiere un procesamiento inteligente que incluye la contextualización, la organización, la validación y la interpretación.

Sesión 1: Concepto de datos, información y sistemas de información

Fundamentos

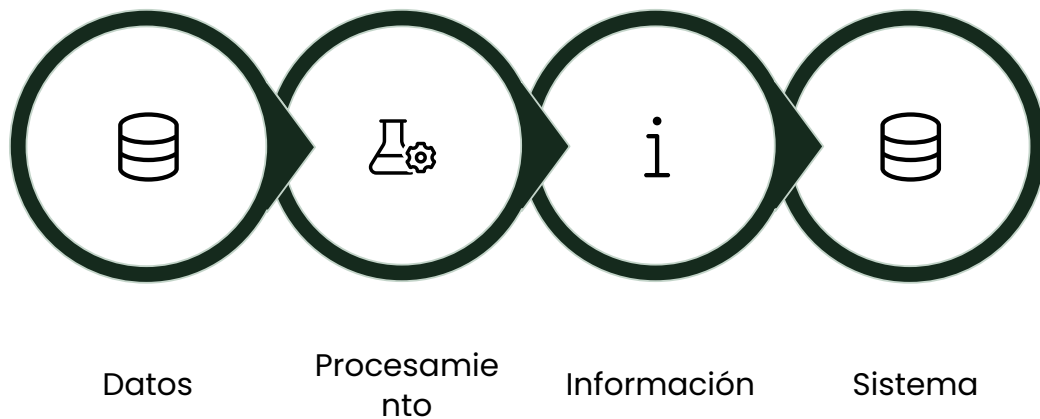
En el mundo digital actual, todo se construye sobre una jerarquía fundamental que debemos comprender: **datos** → **información** → **sistemas de información**. Un **dato** es un hecho en bruto, un valor sin contexto, como el número "1984" o la palabra "Orwell". Por sí solo, este dato no nos comunica nada significativo. La **información**, en cambio, es un conjunto de datos procesados y organizados que adquieren un significado específico. Por ejemplo: "El libro 1984 fue escrito por George Orwell en 1949". Aquí, los datos aislados se han contextualizado y relacionado para crear información valiosa.

Un **sistema de información** es el conjunto completo de componentes (hardware, software, personas, procesos y procedimientos) que trabajan de manera coordinada para recopilar, procesar, almacenar y distribuir información con un propósito específico. Estos sistemas son omnipresentes en nuestra vida: desde el catálogo digital de una biblioteca hasta el complejo sistema de gestión de ventas de una multinacional. La comprensión de esta jerarquía es fundamental porque establece la base conceptual sobre la cual se construyen todas las bases de datos modernas.



El proceso de transformación de datos en información no es automático ni trivial. Requiere un procesamiento inteligente que incluye la contextualización, la organización, la validación y la interpretación. Este procesamiento es lo que añade valor a los datos brutos, convirtiéndolos en un activo estratégico para cualquier organización. Sin este entendimiento, es imposible diseñar sistemas de bases de datos efectivos que realmente sirvan a las necesidades del negocio.

Esquema Visual



Este diagrama ilustra el flujo de transformación de valor desde los datos hasta los sistemas de información. En la parte superior, vemos el proceso conceptual: los **datos brutos** pasan por un **procesamiento** que los convierte en **información**, la cual se gestiona mediante un **sistema de información**. En la parte inferior, se muestra un ejemplo concreto: datos aislados como "1984", "Orwell" y "1949" se procesan mediante contextualización y relación para crear información significativa sobre un libro, que finalmente se gestiona a través de un catálogo digital. Esta transformación es lo que permite que los datos se conviertan en conocimiento accionable.



Caso de Estudio: Amazon y la personalización basada en datos

1

El Contexto/Escenario

Amazon recopila miles de millones de datos cada día de sus usuarios: clics en productos, tiempo de permanencia en páginas, búsquedas realizadas, compras previas, productos añadidos al carrito pero no comprados, valoraciones y reseñas. Estos son datos en bruto, números y registros sin significado aparente cuando se ven de forma aislada.

2

La Estrategia/Ejecución

El verdadero valor surge cuando el sistema de información de Amazon procesa estos datos para convertirlos en información valiosa. Utilizan algoritmos de machine learning para relacionar tu historial de compras con el de millones de otros usuarios que tienen patrones de comportamiento similares. Por ejemplo, si compraste un libro de ciencia ficción y pasaste mucho tiempo viendo reseñas de novelas distópicas, el sistema identifica que tienes un perfil similar al de usuarios que también compraron "Blade Runner" o "Fahrenheit 451". Esta información procesada se convierte en conocimiento sobre tus preferencias.

3

El Resultado/Impacto

La información resultante se utiliza para personalizar completamente tu experiencia de compra. Las famosas recomendaciones "Los clientes que compraron X también compraron Y" no son casuales, sino el resultado de procesar millones de datos para crear información predictiva sobre tus gustos. Este sistema de información no solo mejora la experiencia del usuario, sino que impulsa las ventas de manera masiva. Amazon reporta que aproximadamente el **35% de sus ingresos** provienen de estas recomendaciones personalizadas, demostrando cómo la transformación efectiva de datos en información se traduce directamente en beneficios económicos.

Herramientas y Consejos

1

Comienza con herramientas familiares

Utiliza una hoja de cálculo como Excel o Google Sheets para experimentar con la organización básica de datos en filas y columnas. Esto te ayudará a comprender cómo los datos se estructuran antes de pasar a sistemas más complejos. Crea una tabla simple con información personal (nombre, edad, ciudad) y observa cómo cada celda contiene un dato, cada fila representa una entidad y cada columna un atributo.

2

Experimenta con herramientas de visualización

Descarga Tableau Public o utiliza Power BI Desktop (ambos tienen versiones gratuitas) para experimentar cómo la representación gráfica de los datos los convierte en información comprensible. Importa un conjunto de datos simple y crea diferentes tipos de gráficos para ver cómo la misma información puede comunicar diferentes insights según cómo se presente.

3

Identifica sistemas de información en tu vida diaria

Desarrolla el hábito de reconocer los sistemas de información que utilizas constantemente. Spotify gestiona datos sobre canciones, artistas y tus preferencias de escucha para crear información sobre tus gustos musicales. Tu aplicación bancaria gestiona datos de transacciones para proporcionarte información sobre tus patrones de gasto. Google Maps gestiona datos geoespaciales para darte información sobre rutas y tráfico. Esta práctica te ayudará a entender cómo los conceptos teóricos se aplican en el mundo real.

Mitos y Realidades

✗ MITO: "Más datos siempre es mejor para tomar decisiones"

→ **FALSO.** Esta es una de las creencias más peligrosas en la era del Big Data. La calidad de los datos es infinitamente más importante que la cantidad. Tener millones de datos irrelevantes, incorrectos o desactualizados solo generará información errónea y decisiones equivocadas. Es el principio fundamental de "Garbage In, Garbage Out" (GIGO): si introduces basura en tu sistema, solo obtendrás basura como resultado. Es mejor tener 1,000 datos precisos y relevantes que 1,000,000 de datos de dudosa calidad.

✗ MITO: "Los sistemas de información son solo para grandes empresas"

→ **FALSO.** Los sistemas de información existen en todas las escalas y presupuestos. Cualquier persona que utilice una aplicación de notas para organizar sus tareas diarias, una hoja de cálculo para gestionar sus gastos personales, o incluso una agenda física para planificar su semana, está utilizando un sistema de información. La diferencia está en la escala y la sofisticación, pero los principios fundamentales son los mismos: recopilar datos, procesarlos para crear información útil y utilizarla para tomar mejores decisiones.

Resumen Final

Los datos son hechos brutos sin contexto, la información son datos procesados con significado, y los sistemas de información son los conjuntos de componentes que gestionan esta transformación. La jerarquía **datos → información → sistema de información** es fundamental para entender cómo funcionan las bases de datos modernas. La calidad de los datos es más importante que la cantidad, y los sistemas de información son herramientas universales que utilizamos en múltiples escalas y contextos.

Sesión 2: Evolución de los sistemas de gestión de datos.

Ventajas de las BD

Fundamentos

La evolución de los sistemas de gestión de datos representa una de las transformaciones más significativas en la historia de la informática. En los primeros días de la computación, los datos se almacenaban en **archivos planos** (como ficheros .txt o .csv) que eran gestionados directamente por las aplicaciones individuales. Cada programa tenía sus propios archivos de datos, creando un ecosistema fragmentado y problemático.

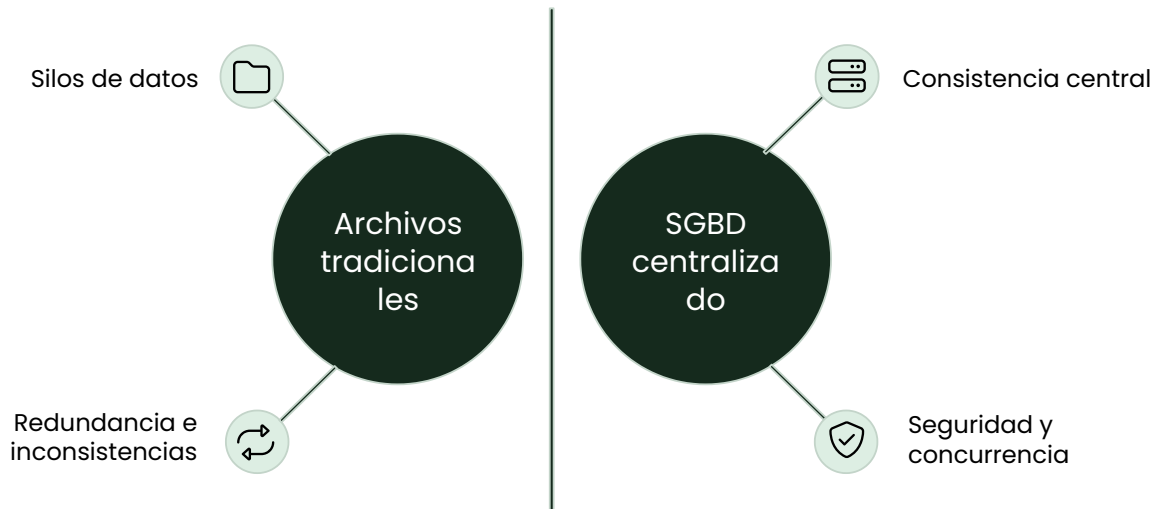
Este enfoque primitivo generaba problemas masivos: **redundancia** (los mismos datos se repetían en múltiples archivos), **inconsistencia** (si se actualizaba un dato en un archivo pero no en otros, se creaban contradicciones), **dependencia de datos** (cambiar la estructura de un archivo requería modificar todos los programas que lo utilizaban), y **dificultad de acceso** (cada nueva consulta requería escribir un programa específico desde cero).

Para resolver estos problemas fundamentales, surgieron los **Sistemas de Gestión de Bases de Datos (SGBD)**. Un SGBD es un software sofisticado que actúa como intermediario inteligente entre los usuarios/aplicaciones y los datos físicos almacenados. Proporciona un lenguaje estándar para definir y manipular datos (como SQL), garantiza la integridad y consistencia de la información, gestiona la seguridad y el control de acceso, y permite que múltiples usuarios accedan a los datos de forma concurrente sin conflictos.



Las ventajas de los SGBD son transformadoras: eliminan la redundancia mediante la centralización, aseguran la consistencia a través de mecanismos de integridad, proporcionan independencia de datos (las aplicaciones no necesitan conocer cómo se almacenan físicamente los datos), facilitan enormemente el acceso y las consultas, mejoran la seguridad mediante sistemas de autenticación y autorización, y permiten la concurrencia controlada para que múltiples usuarios trabajen simultáneamente sin interferencias.

Esquema Visual



Este diagrama contrasta claramente los dos paradigmas. En la parte superior, vemos el **modelo de archivos tradicional** donde cada aplicación gestiona sus propios archivos de datos de forma independiente, creando silos de información que generan redundancia, inconsistencia y dependencia. En la parte inferior, el **modelo de SGBD** muestra cómo todas las aplicaciones interactúan con una base de datos centralizada a través del SGBD, que actúa como un intermediario inteligente. Esta centralización elimina los problemas del modelo anterior y proporciona ventajas significativas en términos de consistencia, independencia, seguridad y capacidad de concurrencia.

Caso de Estudio: La migración de Netflix de Oracle a Cassandra



El Contexto/Escenario

En sus primeros años como servicio de streaming, Netflix utilizaba una arquitectura monolítica tradicional con una base de datos Oracle centralizada. Esta configuración funcionaba adecuadamente cuando tenían unos pocos millones de usuarios, pero a medida que su servicio crecía exponencialmente a nivel global, este sistema centralizado se convirtió en un cuello de botella masivo y un punto único de fallo crítico.



La Estrategia/Ejecución

Netflix tomó la decisión estratégica de migrar a una arquitectura de microservicios distribuidos y adoptó Apache Cassandra, una base de datos NoSQL distribuida diseñada para manejar grandes volúmenes de datos a través de múltiples servidores. En lugar de mantener un único y gigantesco almacén de datos centralizado, distribuyeron los datos de los usuarios (historiales de visualización, perfiles de usuario, preferencias, metadatos de contenido) a través de cientos de servidores distribuidos geográficamente por todo el mundo. Esta migración no fue solo tecnológica, sino también arquitectónica, pasando de un modelo centralizado a uno completamente distribuido.



El Resultado/Impacto

Esta transformación les proporcionó una escalabilidad y disponibilidad sin precedentes en la industria del streaming. Si un servidor o incluso un centro de datos completo falla, el sistema continúa funcionando sin interrupción perceptible para los usuarios. Pudieron manejar un crecimiento explosivo de usuarios (de millones a más de 200 millones de suscriptores globales) y volúmenes de datos masivos sin degradación del rendimiento. Esta migración demostró las ventajas de los SGBD modernos y distribuidos sobre los sistemas tradicionales centralizados, especialmente para aplicaciones que requieren escalabilidad global y alta disponibilidad.

Herramientas y Consejos

1 Experimenta con la diferencia práctica

Para comprender realmente las limitaciones de los sistemas de archivos, crea un simple "directorio telefónico" en un archivo de texto (.txt). Añade nombres, números de teléfono y ciudades. Ahora intenta buscar todas las personas de una ciudad específica o actualizar el número de teléfono de alguien que aparece en múltiples lugares. Rápidamente experimentarás las frustraciones de redundancia, inconsistencia y dificultad de consulta que motivaron la creación de los SGBD.

2 Instala SQLite para experimentación

SQLite es un SGBD completo contenido en un solo archivo, no requiere configuración de servidor y es perfecto para aprender los conceptos básicos. Puedes descargarlo gratuitamente y comenzar a experimentar con SQL inmediatamente. Es la puerta de entrada ideal al mundo de las bases de datos relacionales sin la complejidad de configurar un servidor completo.

3 Utiliza DBeaver como cliente universal

DBeaver es un cliente de base de datos gratuito y de código abierto que te permite conectarte a prácticamente cualquier tipo de base de datos (MySQL, PostgreSQL, SQLite, Oracle, SQL Server, MongoDB, etc.) desde una única aplicación. Esto te permitirá experimentar con diferentes SGBD y comparar sus características sin tener que aprender múltiples herramientas de administración.

Mitos y Realidades

✗ MITO: "Las bases de datos son herramientas exclusivas para programadores"

→ **FALSO.** Esta percepción está completamente desactualizada. Profesionales de marketing utilizan bases de datos para segmentar clientes y analizar campañas. Especialistas en finanzas las usan para análisis de riesgo y reporting. Profesionales de recursos humanos las emplean para gestión de talento y análisis de rendimiento. Incluso roles aparentemente no técnicos como ventas, operaciones y gestión de proyectos se benefician enormemente de entender cómo funcionan las bases de datos, ya que les permite tomar decisiones más informadas basadas en datos.

✗ MITO: "Para pequeñas empresas, Excel es suficiente"

→ **FALSO.** Aunque este enfoque puede parecer más simple inicialmente, no escala y se vuelve rápidamente inmanejable y propenso a errores críticos. Los archivos de Excel tienen limitaciones severas en términos de tamaño, concurrencia (solo una persona puede editarlos a la vez), integridad de datos y capacidades de consulta. Un SGBD, incluso uno simple como SQLite, está diseñado específicamente para resolver estos problemas y proporciona una base sólida para el crecimiento futuro.

Resumen Final

- ❏ Los sistemas de archivos tradicionales causaban redundancia, inconsistencia y dependencia de datos. Los SGBD surgieron como software intermediario que gestiona bases de datos centralizadas, eliminando estos problemas. Las ventajas incluyen **consistencia, independencia de datos, seguridad mejorada y capacidad de concurrencia**. El caso de Netflix demuestra cómo la migración a SGBD modernos permite escalabilidad masiva y alta disponibilidad.

Sesión 3: Arquitectura ANSI/SPARC, modelos de datos, SGBD, tipos de bases de datos

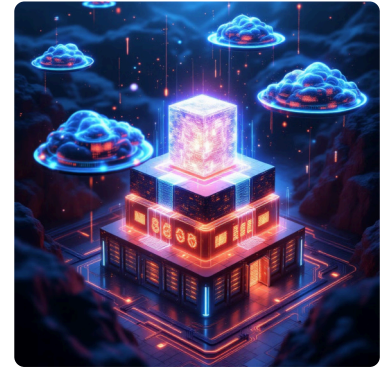
Fundamentos

Para estandarizar el diseño y la implementación de los sistemas de bases de datos, la comunidad académica y profesional desarrolló la **arquitectura ANSI/SPARC** (American National Standards Institute/Standards Planning and Requirements Committee). Esta arquitectura revolucionaria organiza cualquier SGBD en tres niveles conceptuales distintos pero interconectados, proporcionando una separación clara de responsabilidades y una independencia crucial entre diferentes aspectos del sistema.

El **Nivel Interno (o físico)** es la capa más baja y describe cómo se almacenan físicamente los datos en los dispositivos de almacenamiento (discos duros, SSDs, memoria). Incluye detalles como la organización de archivos, los métodos de acceso, los índices físicos y las estructuras de almacenamiento. Este nivel es invisible para la mayoría de los usuarios y es gestionado por administradores de bases de datos especializados.

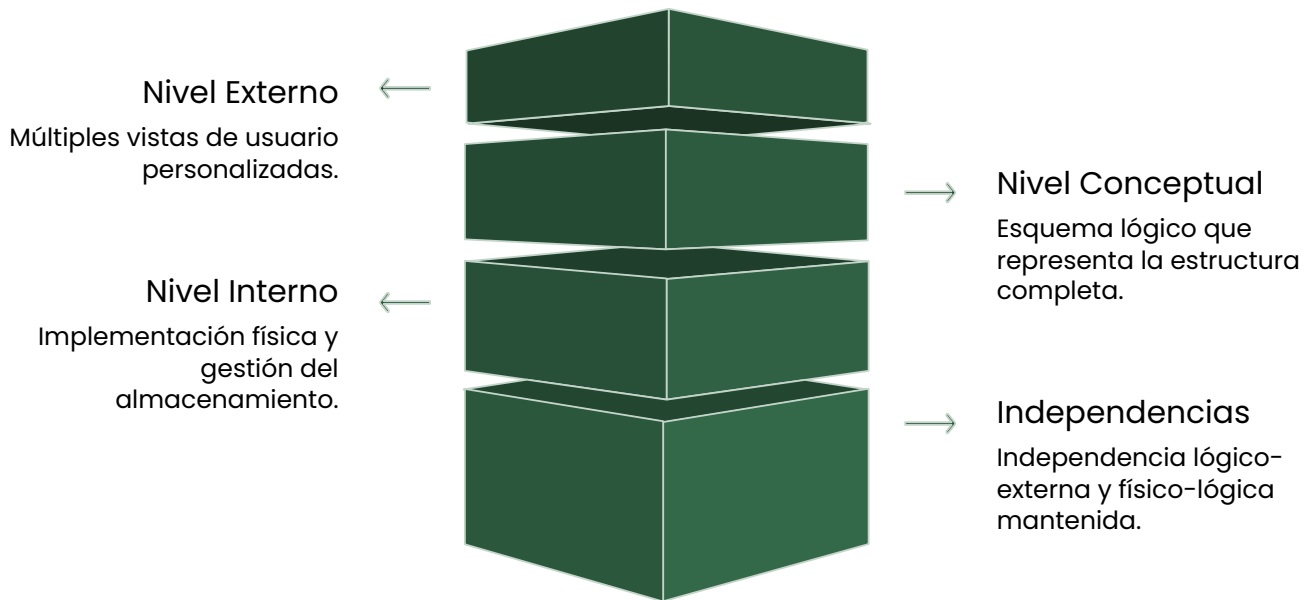
El **Nivel Conceptual (o lógico)** describe la estructura completa de la base de datos para toda la comunidad de usuarios, pero oculta los detalles de almacenamiento físico. Define qué datos se almacenan, las relaciones entre ellos, las restricciones de integridad y las reglas de negocio. Es el nivel donde se aplican los modelos de datos como el relacional, orientado a objetos o NoSQL.

El **Nivel Externo (o de vistas)** describe la parte de la base de datos que es relevante para grupos específicos de usuarios, proporcionando vistas personalizadas y seguras de los datos. Diferentes usuarios ven diferentes "ventanas" de la misma base de datos según sus necesidades y permisos.



Esta arquitectura garantiza dos tipos cruciales de independencia: la **independencia físico-lógica** (cambios en el almacenamiento físico no afectan al esquema conceptual) y la **independencia lógico-externa** (cambios en el esquema conceptual no necesariamente afectan a las vistas de usuario).

Esquema Visual



Este diagrama ilustra la arquitectura de tres niveles y las independencias que proporciona. En la parte superior, múltiples **vistas de usuario** (Nivel Externo) proporcionan perspectivas personalizadas de los datos. Estas vistas se mapean al **esquema conceptual** (Nivel Conceptual), que representa la estructura lógica completa de la base de datos. Este esquema conceptual se implementa físicamente en el **Nivel Interno**, que gestiona el almacenamiento real en dispositivos físicos. Las flechas punteadas muestran las dos independencias clave: los cambios en el nivel conceptual no afectan necesariamente a las vistas (independencia lógico-externa), y los cambios en el almacenamiento físico no afectan al esquema lógico (independencia físico-lógica).

Caso de Estudio: Un banco y la arquitectura de tres niveles

19

El Contexto/Escenario

Un gran banco como BBVA o Santander gestiona una base de datos masiva y compleja con información crítica de millones de clientes, cuentas, préstamos, transacciones y productos financieros. Esta información debe ser accesible para diferentes tipos de usuarios con necesidades y permisos muy diferentes.



La Estrategia/Ejecución

El banco implementa la arquitectura ANSI/SPARC de forma natural y necesaria. En el Nivel Interno, los administradores de sistemas deciden la distribución física de los datos: qué información se almacena en servidores de alta velocidad (datos de transacciones frecuentes), qué se archiva en almacenamiento más lento pero económico (transacciones históricas), cómo se configuran los índices para optimizar las consultas más comunes, y cómo se implementan las copias de seguridad y la replicación geográfica.



Nivel Conceptual - Modelo de Datos

Es el modelo de datos relacional completo diseñado por arquitectos de datos, que incluye todas las tablas (Clientes, Cuentas, Transacciones, Productos, etc.), sus relaciones, restricciones de integridad y reglas de negocio. Este esquema es único y coherente para todo el banco.



Nivel Externo - Vistas Especializadas

Se manifiesta en las diferentes aplicaciones: la aplicación móvil del cliente muestra solo sus propias cuentas y transacciones, la interfaz del cajero proporciona acceso a más información del cliente pero con restricciones, el sistema de análisis de riesgos accede a datos agregados y patrones de comportamiento, y los auditores tienen vistas especializadas para supervisión y cumplimiento normativo.

El Resultado/Impacto: Esta arquitectura permite al banco operar de manera eficiente y segura. Pueden cambiar la infraestructura física (migrar a nuevos servidores, cambiar sistemas de almacenamiento) sin afectar las aplicaciones de usuario. Pueden modificar el esquema conceptual (añadir nuevos productos financieros) sin romper las aplicaciones existentes. Y pueden crear nuevas vistas para diferentes tipos de usuarios sin comprometer la seguridad o la integridad de los datos. La arquitectura ANSI/SPARC es fundamental para la operación de sistemas financieros complejos y críticos.

Herramientas y Consejos

Experimenta con vistas en un SGBD real

Instala MySQL o PostgreSQL y crea una base de datos simple con información de empleados (nombre, salario, departamento). Luego crea diferentes vistas: una para recursos humanos que muestre toda la información, otra para managers que muestre solo empleados de su departamento, y otra para empleados que oculte los salarios. Esto te ayudará a entender prácticamente el concepto de nivel externo.

Utiliza herramientas de modelado conceptual

Descarga MySQL Workbench o utiliza herramientas online como Lucidchart o Draw.io para crear diagramas de modelos de datos. Estas herramientas te permiten diseñar el esquema conceptual visualmente antes de implementarlo físicamente, separando claramente el diseño lógico de la implementación física.

Estudia diferentes modelos de datos

Experimenta con diferentes tipos de bases de datos para entender los modelos de datos. Usa SQLite para el modelo relacional, MongoDB para el modelo de documentos, y Redis para el modelo clave-valor. Cada uno representa una forma diferente de organizar y acceder a los datos en el nivel conceptual.

Mitos y Realidades

✗ MITO: "La arquitectura de tres niveles es solo teoría académica"

→ **FALSO.** Todos los SGBD comerciales y de código abierto implementan implícitamente esta arquitectura. Cuando usas vistas en SQL, estás trabajando en el nivel externo. Cuando diseñas tablas y relaciones, estás en el nivel conceptual. Cuando el administrador configura índices y particiones, está trabajando en el nivel interno. La arquitectura ANSI/SPARC es la base conceptual de todos los sistemas de bases de datos modernos.

✗ MITO: "Solo necesito entender el nivel conceptual"

→ **FALSO.** Aunque es cierto que diferentes roles se enfocan en diferentes niveles, un profesional completo debe entender los tres niveles y sus interacciones. Un desarrollador que no entiende las implicaciones del nivel físico puede escribir consultas ineficientes. Un administrador que no comprende el nivel conceptual puede tomar decisiones de optimización que rompan la lógica de negocio. La comprensión integral de la arquitectura es clave para el éxito profesional.

Resumen Final

La arquitectura ANSI/SPARC organiza los SGBD en tres niveles: **Interno (almacenamiento físico)**, **Conceptual (estructura lógica global)** y **Externo (vistas de usuario)**. Esta separación proporciona independencia físico-lógica e independencia lógico-externa, permitiendo cambios en un nivel sin afectar otros. Los modelos de datos (relacional, NoSQL, etc.) se aplican en el nivel conceptual, y diferentes tipos de SGBD implementan diferentes modelos.