

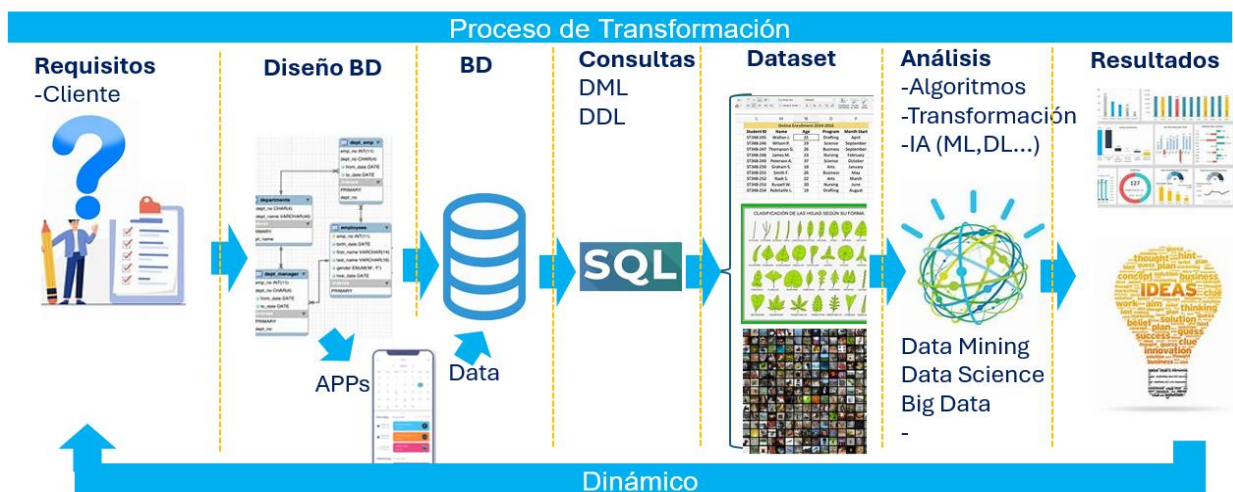
4. Diseño De Bases De Datos

El diseño de bases de datos es una fase crítica en el desarrollo de sistemas de información, ya que establece los cimientos sobre los cuales se construirá todo el sistema de gestión de datos. Un diseño de bases de datos bien planificado y ejecutado garantiza que los datos sean almacenados de manera eficiente, segura y accesible, lo cual es fundamental para el funcionamiento correcto y el éxito a largo plazo de cualquier aplicación o servicio que dependa de datos.

El diseño de bases de datos es fundamental para asegurar que el sistema de gestión de datos cumpla con sus objetivos de manera eficiente y segura, proporcionando una base sólida sobre la cual se pueden construir aplicaciones confiables y escalables. Además un buen diseño facilita la creación de dataset con información valiosa para procesos de analítica.

4.1 Importancia del diseño de bases de datos

Las bases de datos tienen un papel importante en los proceso de transformación de datos en resultados útiles.



El diseño de bases de datos es crucial como base para un buen sistema y para generar los resultados más acertados y cercanos a lo esperado, algunos ítems a tener en cuenta se mencionan a continuación.

4.1.1. Eficiencia en el almacenamiento

Un buen diseño optimiza recursos, evitando la redundancia y asegurando que los datos se almacenen de manera compacta y organizada, que orientan al uso de herramientas especializadas.

4.1.2. Mejora del rendimiento:

Mediante la correcta organización de los datos y el uso adecuado de índices, un diseño eficiente de bases de datos permite que las consultas y transacciones se realicen de manera rápida y eficaz, mejorando el performance general del sistema.

4.1.3. Integridad y consistencia de los datos:

A través de técnicas como la normalización y el establecimiento de reglas de integridad, se asegura que los datos sean precisos y coherentes, con lo que se desea representar y almacenar.

4.1.4. Facilidad de mantenimiento:

El diseño claro y bien documentado facilita el mantenimiento y la escalabilidad de la base de datos, permitiendo a los administradores y desarrolladores realizar cambios y actualizaciones con menor riesgo y mejores tiempos de respuesta.

4.1.5. Escalabilidad

La robustez del diseño permite que la base de datos crezca y se adapte a medida que aumentan las necesidades de la organización, soportando mayores volúmenes de datos cambios estructural y permite expansión a sistemas distribuidos.

4.1.6. Seguridad

La correcta implementación de medidas de seguridad en el diseño inicial de la base de datos protege la información contra accesos no autorizados y vulnerabilidades, garantizando la confidencialidad y protección de los datos sensibles.

4.1.7. Soporte a la toma de decisiones:

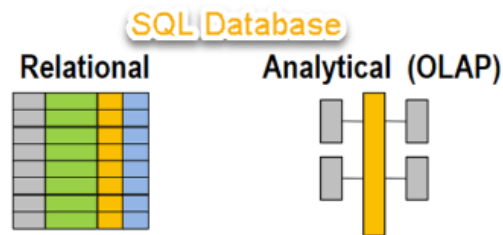
Se debe garantizar que los datos se almacenen de manera precisa y accesible, un buen diseño de base de datos facilita la consulta de datos para el análisis, la generación de informes, apoyando a la organización en la toma de decisiones.

4.2. Proceso del diseño de base de datos Relacional

Las bases de datos relacionales son conocidas también como bases de datos SQL. Para diseñar bases de dato se deben conocer algunos conceptos que se muestran en la siguiente tabla:

Concepto	Definición
Registro	Conjunto de datos almacenados en una fila dentro de una tabla de base de datos, representando una única instancia de una entidad.
Tupla	Término técnico usado para describir un registro en una base de datos relacional; una fila en una tabla.
Campo o Atributo	Columna en una tabla de base de datos que representa una propiedad o característica de la entidad modelada.
Índice	Estructura que mejora la velocidad de las operaciones de consulta en una tabla, permitiendo un acceso más rápido a los registros.
Clave Primaria	Conjunto de uno o más campos/atributos que identifican de manera única a cada registro en una tabla, asegurando la integridad y unicidad de los datos.
Clave Foránea	Campo o conjunto de campos en una tabla que crea un vínculo entre los datos en dos tablas, haciendo referencia a la clave primaria de otra tabla, asegurando la integridad referencial.
Relación	Asociación entre dos o más tablas de una base de datos, establecida mediante claves primarias y foráneas para conectar los registros correspondientes.
Tipo de Dato	Definición de la naturaleza de los datos que pueden almacenarse en un campo, como enteros, cadenas de texto, fechas, etc., determinando las operaciones que pueden realizarse sobre ellos.

Clave primaria -Primary Key = PK
 Calve Foránea -Foreing Key = Fk



4.2.1. Análisis de requisitos

Es el proceso de recopilar y definir las necesidades y expectativas de los usuarios y las partes interesadas sobre qué datos se deben almacenar y cómo se deben gestionar en la base de datos. Divide los elementos de información en entidades principales o temas, como Productos o Clientes. Implica una tarea investigativa ardua hasta conocer bien las características del sistema a desarrollar.

4.2.2. Identificación de entidades:

Consiste en determinar los objetos o conceptos principales del mundo real que deben ser representados en la base de datos y se representan en tablas que almacenaran la informacion ejemplo de ellos: como clientes, facturas, productos, pedidos, tipo de transaccion, pagos etc.

4.2.3. Organización de entidades en tablas:

Es el proceso de agrupar las entidades identificadas en tablas que reflejan de manera estructurada cómo se almacenarán los datos en la base de datos, visualizando una aproximación de la relación entre ellas.

Identificación de campos (atributos) por entidad:

Es el proceso de definir los atributos o campos que describen cada entidad, especificando qué información se almacenará en cada una(tabla), por ejemplo:

Entidad = cliente	
Campos:	Nro_id Nombre Nit Direccion Contacto

4.2.4 Identificación de claves:

Consiste en determinar los posibles atributos o combinaciones de atributos que pueden servir como identificadores únicos para las entidades, de ellos: la clave primaria, claves foráneas, además la identificación de los índices.

Aquí se realizan preguntas sobre los campos que identifican registros o los relacionan en otras tablas, por ejemplo:

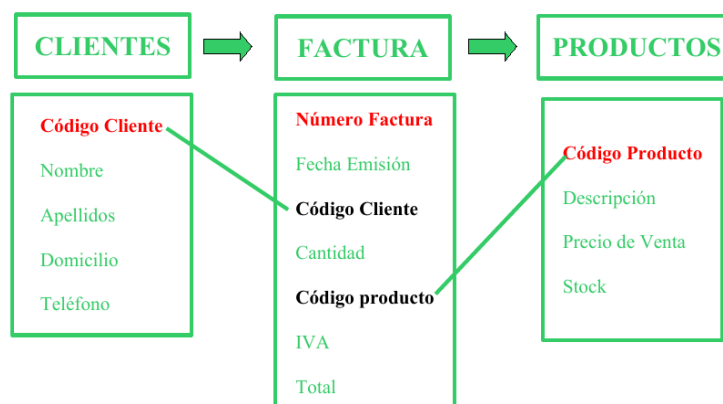
- ¿El campo se puede repetir?
- ¿El campo es único?
- ¿El campo también va en otra tabla?
- Las mismas 3 preguntas hacerlas cuando se necesiten agrupar 2 o más campos.

Ejemplo:

- ¿Se puede repetir el nombre de una persona?
- ¿Se puede repetir el número de una factura?
- ¿Se puede repetir el código del cliente?
- Se puede repetir la fecha+código?

4.2.5. Identificación de relaciones entre tablas

Es el proceso de establecer las conexiones y dependencias entre las tablas, definiendo cómo interactúan y se relacionan entre sí las entidades, mediante las claves primarias (PK) y Foráneas (FK).



4.2.6. Normalización

Es el proceso de organizar los datos en las tablas de manera que se minimice la redundancia y se asegure la integridad de los datos y eliminar redundancias, dividiendo las tablas según ciertas reglas y formas normales.

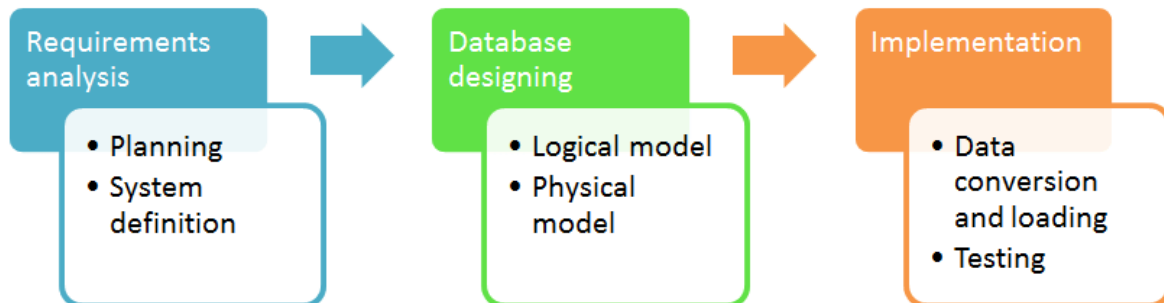
En otras palabras, es el proceso de simplificar el global de los datos separándolos en otras entidades y asignado las claves respectivas. Existen varias formas normales las más comunes son las 3 primeras formas normales.

Formas normales (1,2,3)
<p>La primera forma normal – 1NF</p> <p>Para una tabla ser la primera forma normal, debe cumplir el siguiente criterio:</p> <ul style="list-style-type: none"> una sola celda no debe contener más de un valor (atomicidad) debe haber una clave primaria para identificación no filas o columnas duplicadas cada columna debe tener solamente un valor por cada fila en la tabla <p>La segunda forma normal – 2NF</p> <p>El 1NF solamente elimina los grupos repetitivos, no la redundancia. Por eso hay 2NF.</p> <ul style="list-style-type: none"> Una tabla se dice que está en 2NF si cumple el siguiente criterio: ya está en 1NF no tiene dependencia parcial. Es decir, todos los atributos no claves son totalmente dependientes de la clave primaria <p>La tercera forma normal – 3NF</p> <p>Cuando una tabla está en 2NF, elimina los grupos repetitivos y la redundancia, pero no elimina la dependencia parcial transitiva.</p> <p>Esto significa que un atributo no principal (un atributo que no forma parte de la clave del candidato) es dependiente de otro atributo no principal. Esto es lo que la tercera forma normal (3NF) elimina.</p>

Sitios de referencia normalización:

- <https://www.youtube.com/watch?v=CdUsNUAdl8E>
- <https://www.freecodecamp.org/espanol/news/normalizacion-de-base-de-datos-formas-normales-1nf-2nf-3nf-ejemplos-de-tablas/>

4.3 Ciclo de vida del diseño de BD



4.3.1. Planeación y Definición del Sistema:

El Objetivo es determinar los requisitos y el alcance del sistema de base de datos.

Actividades:

- Reunir y analizar los requisitos del negocio y de los usuarios.
- Definir el propósito de la base de datos y sus objetivos.
- Identificar las entidades, procesos y datos necesarios.

Resultado: Un documento de requisitos del sistema que guiará las siguientes fases del diseño.

4.3.2. Modelo Lógico

A aquí se crea un modelo de datos conceptual que represente la estructura lógica de la base de datos, independiente del DBMS

Actividades:

- Diseñar un esquema conceptual utilizando diagramas entidad-relación (ER).
- Definir entidades, atributos y relaciones.
- Establecer reglas de integridad y restricciones.

Resultado: Un modelo lógico de datos que es independiente del sistema de gestión de bases de datos (DBMS).

4.3.3 Modelo Físico

Se traduce el modelo lógico en un diseño físico específico para un DBMS determinado.

Actividades:

- Convertir el esquema lógico en tablas, columnas, índices y otros objetos de base de datos.
- Optimizar el diseño físico para el rendimiento y almacenamiento.
- Definir estructuras de almacenamiento y métodos de acceso.

Resultado: Un esquema físico de base de datos listo para ser implementado en un DBMS (Mysql, Oracle, Ms Sql Server, etc..)



4.3.4. Implementación:

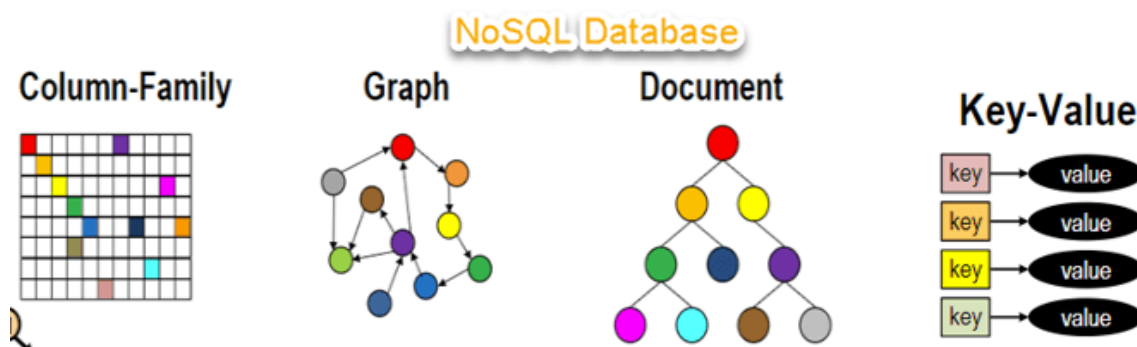
Objetivo: Construir la base de datos física en el DBMS y cargar los datos iniciales.

Actividades:

- Crear las estructuras de la base de datos (tablas, índices, vistas, etc.) en el DBMS.
- Implementar restricciones de integridad y reglas de negocio.
- Cargar datos iniciales y realizar pruebas de funcionalidad y rendimiento.

Resultado: Una base de datos operativa y lista para ser utilizada por aplicaciones y usuarios.

4.4 Modelado NoSql



Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y almacenan los datos en esquemas flexibles que se escalan con facilidad para aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala. Esta página incluye recursos que lo ayudan a comprender mejor las bases de datos NoSQL y comenzar a usarlas.

4.4.1 Ventajas de las bases de datos NoSQL

Las aplicaciones modernas se enfrentan a varios desafíos que las bases de datos NoSQL pueden resolver. Por ejemplo, las aplicaciones procesan un gran volumen de datos de fuentes dispares, como las redes sociales, los sensores inteligentes y las bases de datos de terceros. Todos estos datos dispares no encajan perfectamente en el modelo relacional. La aplicación de estructuras tabulares puede provocar redundancia, duplicación de datos y problemas de rendimiento a escala.

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos no relacionales y tienen esquemas flexibles para crear aplicaciones modernas. Son ampliamente reconocidas por su facilidad de desarrollo, su funcionalidad y el rendimiento a escala. Los beneficios de las bases de datos NoSQL se enumeran a continuación.

Flexibilidad

Las bases de datos NoSQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido y más iterativo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados.

Escalabilidad

Las bases de datos NoSQL generalmente están diseñadas para escalar horizontalmente usando clústeres distribuidos de hardware, en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado.

Alto rendimiento

Las bases de datos NoSQL están optimizadas para modelos de datos y patrones de acceso específicos. Esto permite un mayor rendimiento que si intentara lograr una funcionalidad similar con bases de datos relacionales.

Altamente funcional

Las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos.

4.4.2 Usos de BD NoSQL

Puede usar bases de datos NoSQL para crear una amplia variedad de aplicaciones móviles, de Internet of Things (IoT), de juegos y web de alto rendimiento que proporcionan excelentes experiencias de usuario a escala.

Terminología de SQL vs NoSQL

SQL	MongoDB	DynamoDB	Cassandra	Couchbase
Tabla	Conjunto	Tabla	Tabla	Bucket de datos
Fila	Documento	Elemento	Fila	Documento
Columna	Campo	Atributo	Columna	Campo
Clave principal	ObjectId	Clave principal	Clave principal	ID del documento
Índice	Índice	Índice secundario	Índice	Índice
Ver	Ver	Índice secundario global	Vista materializada	Ver
Tabla u objeto anidado	Documento incrustado	Mapa	Mapa	Mapa
Matriz	Matriz	Lista	Lista	Lista

Anexo: -Ejercicio

Planteamiento de requisitos para diseñar un modelo de datos

Requisitos Generales (Sistema para Gestionar una Liga de Fútbol)

Para implementar un sistema de gestión de una liga de fútbol, es crucial identificar los requisitos clave que nos permitirán definir las entidades, atributos y relaciones necesarias en el modelo de datos. Primero, es fundamental registrar todos los equipos de fútbol, incluyendo detalles como el nombre del equipo, ciudad, estadio, entrenador principal y año de fundación. Cada equipo debe estar vinculado a jugadores, con información detallada sobre el nombre del jugador.

Cada temporada debe estar registrada, y cada temporada tendrá múltiples partidos programados. Los partidos deben incluir los detalles, equipos participantes, y el resultado del partido (goles anotados por cada equipo). Además, debe ser posible registrar eventos importantes durante los partidos, como goles, tarjetas (amarillas y rojas), sustituciones, y lesiones, especificando el minuto del evento y el jugador involucrado.

Los árbitros, quienes supervisan los partidos, deben estar registrados con su nombre, teléfono, email y nivel de certificación. Los árbitros pueden ser asignados a partidos específicos y sus decisiones durante los partidos deben ser documentadas.

Referencias

- <https://entidadrelacion.com/guia-diseno-bases-datos-relacionales/>
- <https://appmaster.io/es/blog/diagrama-de-diseno-de-la-base-de-datos-guia-completa>
- <https://ia902808.us.archive.org/8/items/fundamentosdesistemasdebasesdedatos/Fundamentos-de-Sistemas-de-Bases-de-Datos.pdf>
- RAMEZ ELMASRI, Fundamentos de Sistemas de Bases de Datos, Quinta Edición

Fecha Creación	Enero 25 2024
Responsable	Plinio Neira Vargas
Revisado por	Sonia Escobar
Fecha Revisión	Febrero 10 2024