

PROYECTO COMPILANDO CONOCIMIENTO

PROGRAMACIÓN

Bases de Datos

Una Pequeña (Gran) Introducción

AUTORES:

Rosas Hernandez Oscar Andrés

Lopez Manriquez Angel

Índice general

I	Parte Abstracta	2
1.	Definiciones	3
1.1.	Repositorio de Datos	4
1.2.	Base de Datos	4
1.3.	Proposito para una Base de Datos	4
1.4.	Elementos de un Sistema de Base de Datos	6
1.5.	Abstracción de los Datos	7
1.6.	Usuarios de una Base de Datos	8
1.6.1.	Usuarios Finales	9
1.7.	DML vs DDL	10
1.8.	Arquitectura del Sistema Gestor	12
2.	SQL	13
2.1.	SQL como Lenguaje	14

Parte I

Parte Abstracta

Capítulo 1

Definiciones

1.1. Repositorio de Datos

Son un conjunto de datos, donde definio a un dato como cualquier información que sea válida.

1.2. Base de Datos

Son un conjunto de datos interrelacionado definido por un modelo de datos, esto no lo tiene necesariamente un repositorio de datos. Así como progrmas que nos permitan acceder y manipular esa información.

1.3. Proposito para una Base de Datos

Todo muy bien, pero ¿porque debería importarme un comino?¿Porqué preferir una base de datos sobre simplemente guardar los datos de manera “común”?

- **Redundancia de datos e inconsistencia:** Ya que tu no vas a programar un sistema entero habra muchas maneras en que los demás programadores crearán aplicaciones y sobretodo en como van a guardar los datos.

Peor aún, ¿Qué pasa cuando tengamos un motón de archivos con casi la misma información? es decir cuando tengamos un montón de archivos con tu mismo número de telefono, con tu misma información de contacto.

- **Problemas para acceder a la información que queriamos en primer lugar:** Supongamos que queremos acceder a los datos, digamos que tenemos un montón de registros de sobre alumnos.

¿Como haríamos para tener todos los alumnos que hayan reprobado? No hay forma fácil de hacerlo, incluso la forma mas “correcta” sería desarrollar un pequeño programa que se encargue de hacer lo que queremos.

Y esto podría servir muy bien.... Hasta que necesitemos algo más. Entonces tenemos que estar haciendo programas y programas ¡Que cansado!

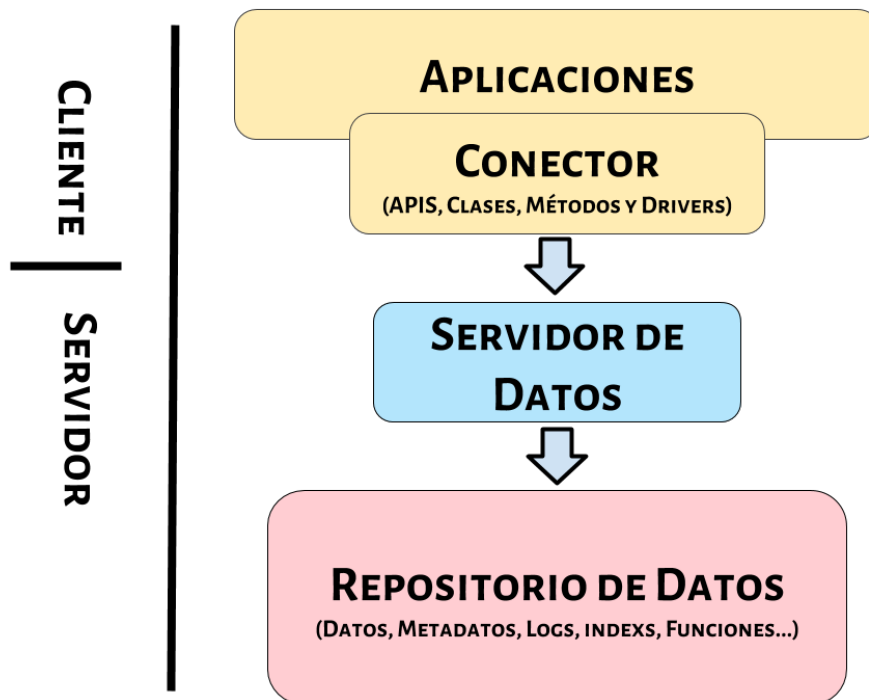
- **Aislamiento de la Información:** Ya que puede estar repartida por todos lados, no se exactamente por donde tengo que empezar a buscar. Y esto se vuelve un verdadero desastre cuando intentamos modificar la información guardada.
- **Problemas con Atomicidad:** Si ocurriera un problema en el sistema al momento de estar modificando la información me gustaria que al volver a arrancar todo, este debería regresar a un punto de respaldo.

Como si el sistema fallara justo al hacer una transacción bancaria, me interesa que se haya realizado o no, pero que no le haya quitando dinero a una cuenta pero no se la haya dado al otro cliente.

Podemos resumir esto con que a diferencia de un sistema de archivos, una base de datos busca:

- Evitar o miniza la redundancia
- Evitar inconsistencias en los datos
- Eliminar inconsistencias en los datos
- Comunicacion con distintos repositorios de datos
- Control de concurrencia

1.4. Elementos de un Sistema de Base de Datos



- **Aplicaciones:** Son las aplicaciones que usan la Base de Datos, generalmente creado en lenguajes de Alto Nivel.
- **Conector:** Son las Apis, o las funciones que permiten a la aplicación interactuar con nuestra base de datos.
- **Servidor de Datos:** Son el software especializado que nos permite manipular inteligentemente nuestros datos:
 - Creación de Repositorios
 - Creación de Cuentas de Usuarios
 - Se encarga de crear archivos lógicos, físicos y objetos de la BD.
 - Se encarga de administrar las transacciones, bloqueos, etc.

1.5. Abstracción de los Datos

- **Nivel Físico:**

Es el nivel base de abstracción que describe como es que la información es guardada de manera actual. Esto nos permite describir como de complejo es que son las estructuras en la realidad.

- **Nivel Lógico:**

Es el nivel que nos muestra como es que se almacena la información dentro de la base de datos y como es que se da la relaciones entre la información.

Aunque la implementación de las estructuras simples en el nivel lógico puede implicar complejas estructuras de nivel físico, el usuario de este nivel no necesita ser consciente de esta complejidad. Esto se conoce como independencia de datos físicos. Los administradores de bases de datos, que deben decidir qué información debe conservar en la base de datos, utilizan el nivel lógico de abstracción.

- **Nivel Visual:**

El nivel más alto de abstracción describe sólo una parte de la base de datos completa.

Aunque el nivel lógico utiliza estructuras más simples, la complejidad se mantiene debido a la variedad de información almacenada en una base de datos grande.

Muchos usuarios del sistema de base de datos no necesitan toda esta información, sólo necesitan acceder a una parte de la base de datos. El nivel de vista de la abstracción existe para simplificar su interacción con el sistema.

1.6. Usuarios de una Base de Datos

■ Diseñadores de la Base de Datos

- Encargado de grabar los propios Modelos de Datos:
 - Conceptuales: E/S
 - Lógicos: Relacionados
 - Físicos: Índices o Estructuras de Datos
- Esquema de la Base de Datos

■ Administrador de la Base de Datos

Encargado de:

- Monitorear el Performance
- Herramientas Administrativas
 - Creacion de cuentas de usuarios
 - Objetos accedidos.
 - Matriz de autorizacion.
- Definir tiempos de respaldo
- Reorganización físico
- Llevar a cabo las tecnicas de recuperación

■ Programadores de Aplicaciones

- Interfaces de los usuarios finales
 - Facilitar el acceso a ciertos "objetos" de la BD
- Interfaces para la gestión de la aplicaciones
 - Operaciones de escritura sobre altas ó bajas
 - IDE desarrollo (Java, .NET)
 - Lenguaje Scripts
 - Conectividad servidores de datos (API s)

1.6.1. Usuarios Finales

- **Casuales**
 - Data Mining
 - Big Data
- **Principiantes - Paramétricos** :Puntos de venta.
- **Sofisticados**
 - Experiencia en aplicaciones
 - Programadores de aplicación
 - DBA
 - Investigadores
- **Independientes (Stand Alone)**
 - Sistemas Escolares
 - Sistemas de prueba
 - Sin conectividad a otros nodos

1.7. DML vs DDL

■ DML

Data Manipulation Lenguaje, es un lenguaje que nos permite modificar los datos guardados.

Los tipos de acceso que tenemos disponibles son:

- Recuperar información
- Insertar nueva información
- Eliminar la información
- Modificación de la información

Una consulta o query es una sentencia que solicita la recuperación de información. La parte de un DML que implica recuperación de información se denomina lenguaje de consulta.

Aunque técnicamente incorrecto, solemos utilizar los términos lenguaje de consulta y lenguaje de manipulación de datos como sinónimos.

■ DDL

Data Definition Lenguaje, es Lenguaje de definición de datos.

Especificamos un esquema de base de datos mediante un conjunto de definiciones expresadas por un lenguaje especial denominado DDL.

El DDL también se utiliza para especificar propiedades adicionales de los datos.

Podemos simplificar en el sentido de SQL que:

DDL: Data Definition Language, mediante el cual puede definir nuevos objetos de base de datos, como Table, Views, Stored Procedures, etc. Algunos comandos comunes son:

- CREATE
- ALTER
- DROP
- etc...

DML: Data Manipulation Language, mediante el cual puede realizar cambios en los objetos creados anteriormente por DDLs. Algunos comandos comunes son:

- INSERT
- UPDATE
- DELETE
- SELECT
- etc...

1.8. Arquitectura del Sistema Gestor

Podemos acceder al Sistema Gestor de nuestra Base de Datos de muchas maneras, desde formularios web, aplicaciones de escritorio e interpretes de SQL, todos se comunican con la Base de Datos mediante SQL.

■ Motor de Evaluación de Consultas

- Analizador: Este se encarga de analizar las sentencias SQL a nivel sintactico y lexico, así como una validación de que existan dichas relaciones y atributos. Es lo que mucha gente conoce como un compilador de DDL ó DML.
- Evaluador de Operaciones: Este se encarga de crear el árbol canónico, es decir, la forma formal que tendría el árbol necesario para acceder a la información.
- Optimizador: Se encarga de tomar el árbol canónico y optimizarlo para que se tenga que hacer la menor cantidad de operación a nivel lógico, este se conoce como árbol de consulta.
- Ejecutor de Planes: Se encarga de planear la mejor estrategia para ejecutar la consulta de tal manera que se optimize de manera física.

■ Gestores

- Gestor de Transacciones: Este es el que se encarga de organizar varias sentencias SQL para ejecutarlas como una transacción.
- Gestor de Bloqueos: Este es el que se encarga de ver si es que cierta relación esta bloqueada porque esta ocurriendo una transacción en ese momento.

Capítulo 2

SQL

2.1. SQL como Lenguaje

SQL **no** es tan potente como una máquina universal de Turing.

Es decir, hay algunos cálculos que son posibles utilizando un lenguaje de programación de propósito general, pero no son posibles con SQL. SQL también no admite acciones como la entrada de usuarios, la salida a las pantallas ó la comunicación a través de la red. Dichas computaciones y acciones deben escribirse en un lenguaje principal (C, C++ ó Java) con consultas SQL incorporadas que acceden a los datos de la base de datos

Los programas de aplicación son programas que se utilizan para interactuar con la base de datos de esta manera.