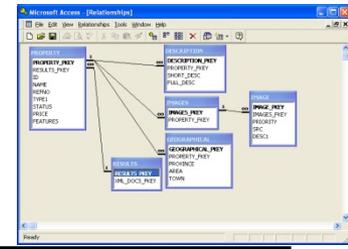
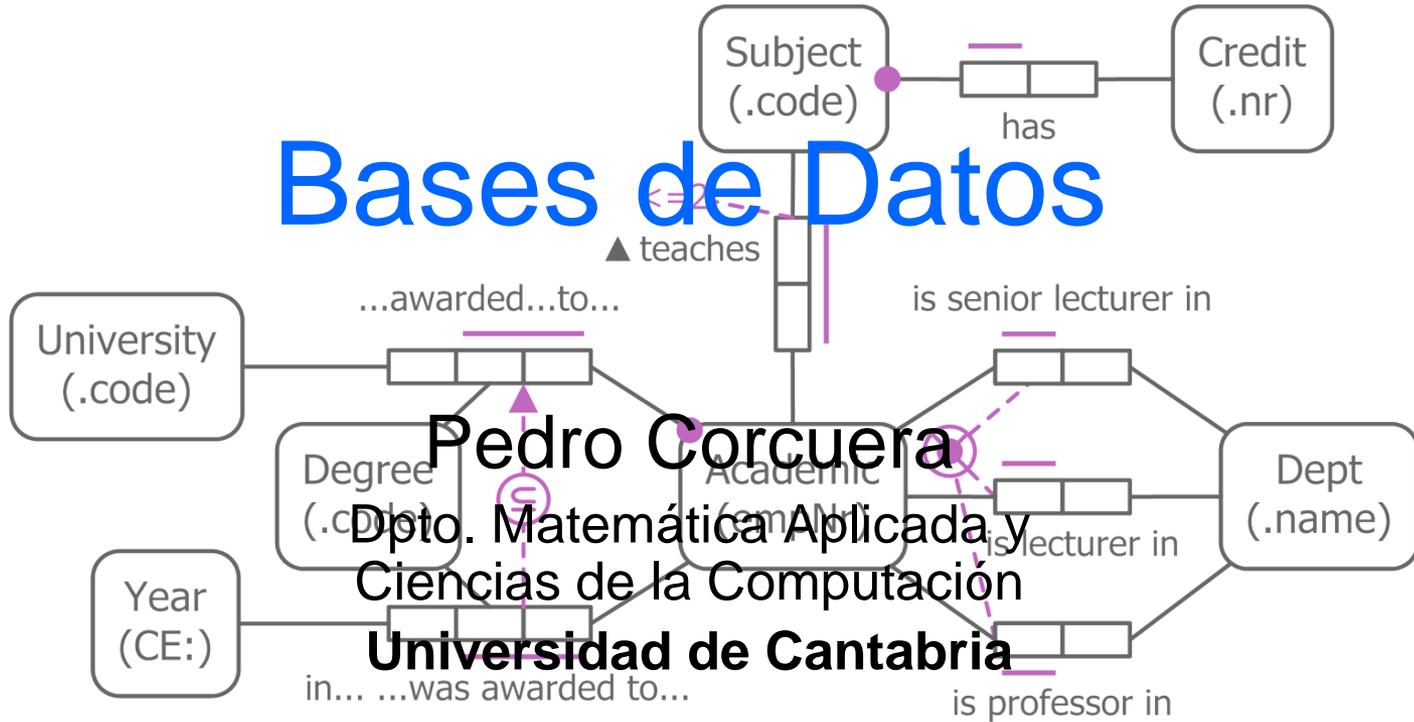


SQL



Bases de Datos



Pedro Corcuera

**Dpto. Matemática Aplicada y
Ciencias de la Computación**

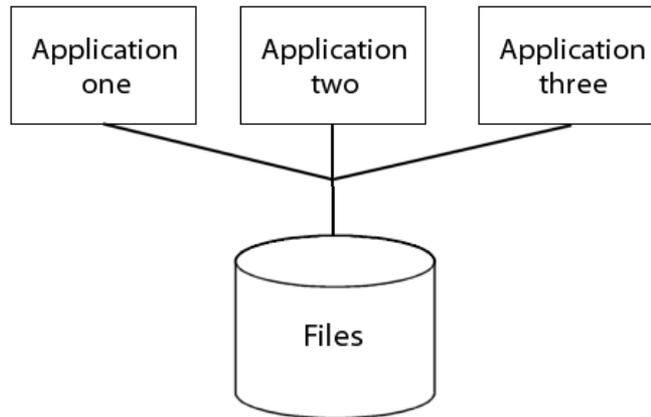
Universidad de Cantabria

corcuerp@unican.es

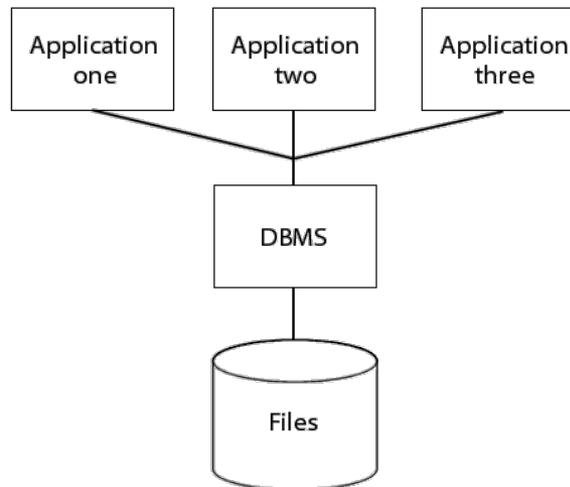
- Presentar los conceptos básicos y terminología de las bases de datos

- Gestión de la información
- Visión de los datos
- Lenguajes de bases de datos
- Diseño de bases de datos
- Almacenamiento de datos y consultas
- Arquitectura de las bases de datos
- Usuarios y administradores de bases de datos
- Estructura general de un sistema
- Historia de los sistemas de bases de datos

- Sistemas de archivos



- Bases de Datos



Sistemas Tradicionales de archivos - Desventajas

- **Inconsistencia y redundancia** de información.
 - Copias múltiples de los datos
- Carencia o pérdida en el **control de concurrencia**
 - Para usuarios múltiples
- Carencia o pérdida de **seguridad**
 - Mas allá de la que provee el sistema operativo
- Reglas de **integridad**
 - Es difícil de mantener las reglas de integridad de la información

- Dificultad al acceder los datos
 - Dependiendo de la aplicación y necesidades de consulta, reorganización de estructuras de datos pueden requerirse Datos aislados en diferentes archivos
- Dependencia de datos
- Formatos de archivo no compatibles (Cobol, C, etc.)
- Nuevo requerimiento de Consulta, nuevo programa

¿Qué es una Base de Datos?

- Una colección de datos (registros) que describen las actividades de una o más organizaciones relacionadas.
- Colección organizada de datos, relativa a un problema concreto, que puede ser compartida por un conjunto de usuarios/aplicaciones
- Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- Actualmente las BD están en formato digital

- Flexibilidad de adaptación a cada problema
- Optimización en la gestión de la información
- Independencia física y lógica de los datos
- Control de la integridad de los datos
- Garantía sobre la consistencia de la información
- Facilidad de acceso concurrente
- Protección ante fallos del sistema
- Seguridad ante accesos restringidos

¿Que es un Sistema Gestor de Base de Datos (DBMS)?

- Producto software que sirve para el mantenimiento y utilización de colecciones de datos.
- Un DBMS ayuda a los usuarios a definir, crear, mantener y controlar el acceso a los datos
- Objetivos:
 - Lograr que el desarrollo y mantenimiento de aplicaciones dentro de la organización sea fácil, económico, flexible y eficiente
 - Organizar los datos para que su explotación sea oportuna, confiable y eficiente

- Según la variabilidad de los datos almacenados
 - Bases de datos estáticas
 - Bases de datos dinámicas
- Según el contenido
 - Bases de datos bibliográficas
 - Bases de datos de texto completo
 - Directorios
 - Bases de datos o "bibliotecas" de información

- Primera Generación (años 50 y 70):
 - Procesamiento de datos con cintas magnéticas como almacenamiento (50 -60). Las cintas sólo proporcionan acceso secuencial. Tarjetas perforadas como entrada
 - (60-70) Los discos duros permiten acceso directo a los datos. Modelos de datos jerárquicos y en red en amplio uso. Ted Codd define el **modelo de datos relacional**.
 - Alto rendimiento en el procesamiento de transacciones (para su época)

Historia de los Sistemas de Bases de Datos

- Años 80:
 - Los prototipos relacionales de investigación evolucionan a sistemas comerciales. SQL se convierte en estándar. Sistemas de bases de datos paralelos y distribuidos y orientados a objetos
 - Años 90:
 - Aplicaciones de ayuda a la toma de decisiones y minería de datos. Grandes almacenes de datos multi-terabyte. Emerge el comercio Web
 - Años 2000:
 - Estándares XML XQuery. Administración automatizada de bases de datos
-

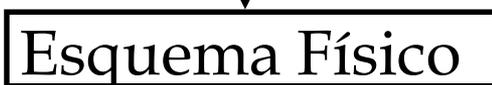
- Independencia de datos y acceso eficaz
 - Reduce la redundancia de información
 - 3 Niveles de abstracción

Las vistas describen como ven los usuarios los datos



El esquema conceptual define la estructura lógica

El esquema físico describe los archivos, estructuras e índices usados



- Independencia de Datos
 - Físico: habilidad de modificar el esquema físico sin tener que rescribir el programa de aplicación
 - Lógico: habilidad de modificar el esquema conceptual sin tener que rescribir el programa de aplicación
- Reducción del tiempo de desarrollo de aplicaciones
 - Facilita la aplicación de estándares y el desarrollo de aplicaciones
 - Permite dar respuesta a consultas no planeadas de información (sin crear nuevos programas)

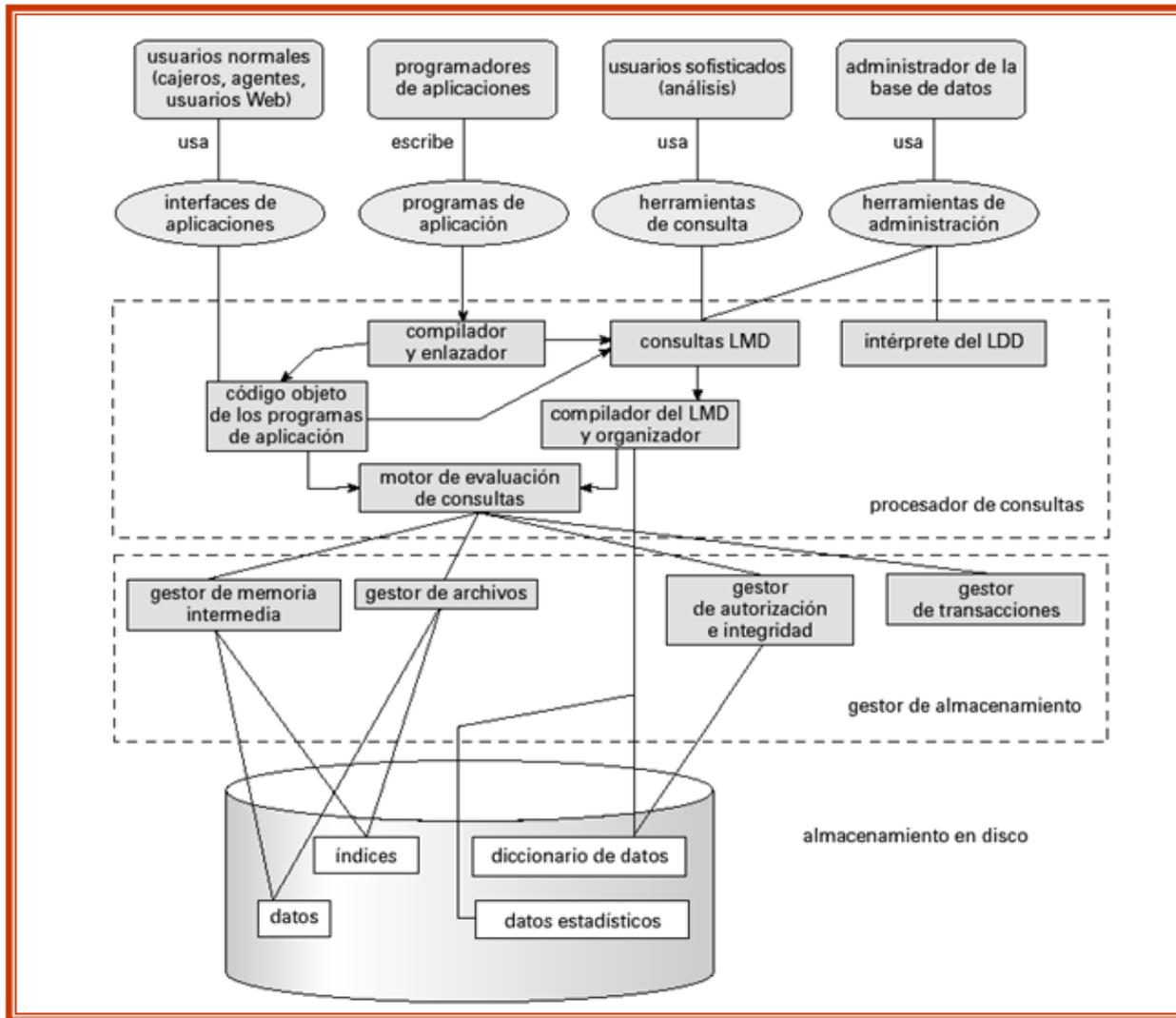
- Integridad y seguridad de la información
 - Presentar diferentes vistas de la información
 - Dar privacidad y seguridad
- Administración uniforme de los datos
 - Control integrado de la información
 - Reduce los recursos asignados al mantenimiento
- Acceso concurrente, recuperación de fallas
 - Permite compartir la información y asegura atomicidad
 - Soporta recuperación de fallas
 - Concepto clave es Transacción y uso de logs para recuperación en caso de fallas

- Hardware
 - Puede variar desde una PC a una red de computadoras
- Software
 - DBMS, sistema operativo, software de red y aplicaciones
- Datos
 - Usados por la organización, además de una descripción de los datos llamado esquema
- Procedimientos
 - Instrucciones y reglas que son aplicados al diseño y uso de la base de datos y del entorno DBMS
- Gente

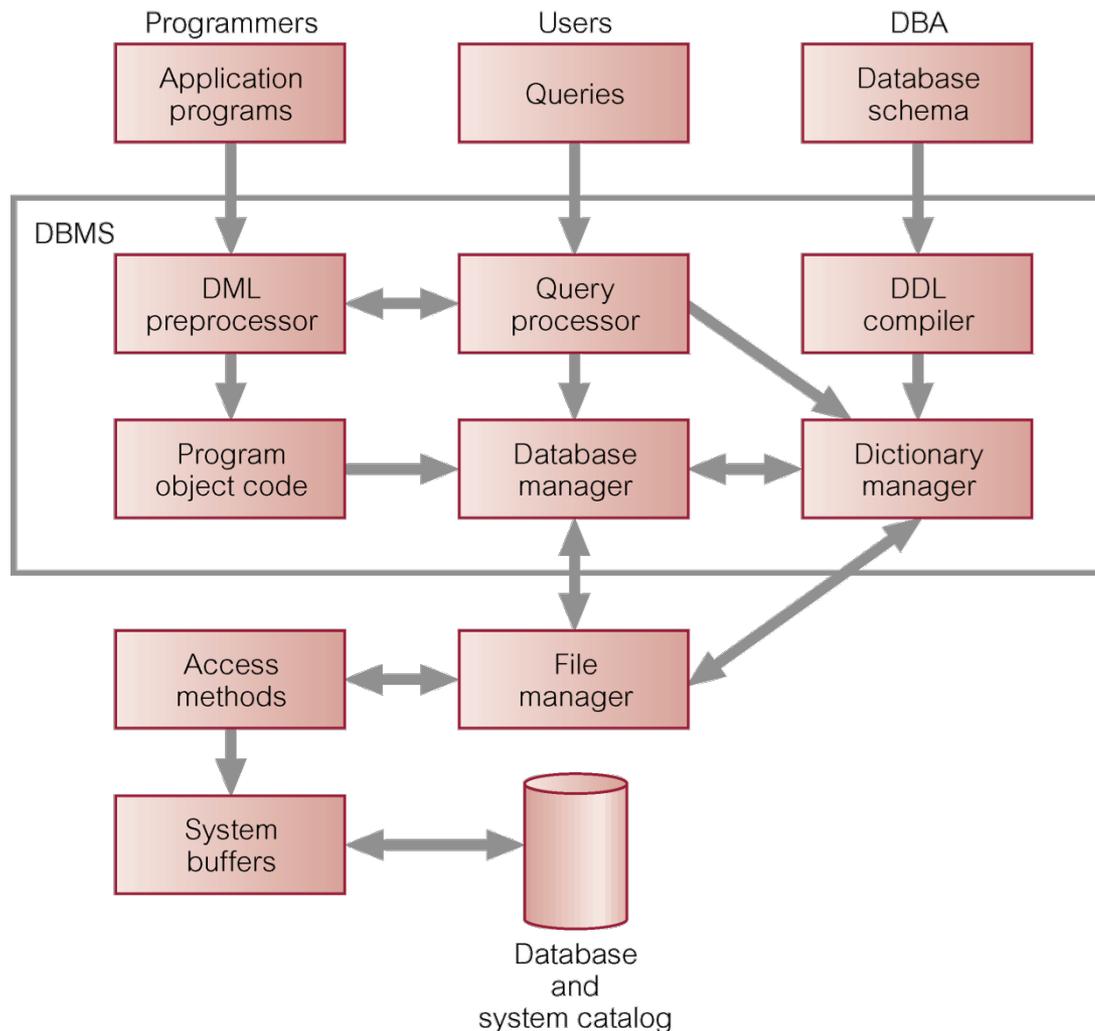
- Administrador de Datos (DA)
- Administrador de la Base de Datos (DBA)
- Diseñador de la Base de Datos (Lógico y Físico)
- Programadores de aplicaciones
- Usuarios Finales (novel y senior)

- Coordina todas las actividades del sistema de la base de datos; conoce los recursos y necesidades de información de la empresa.
- Entre las funciones del DBA se incluyen:
 - Definición/modificación del esquema y organización física
 - Estructura de almacenamiento y definición del método de acceso
 - Concesión de autorización para el acceso a los datos
 - Especificación de las restricciones de consistencia
 - Actuar como enlace con los usuarios
 - Supervisión de rendimiento y respuesta a cambios de los requisitos

Estructura general del sistema

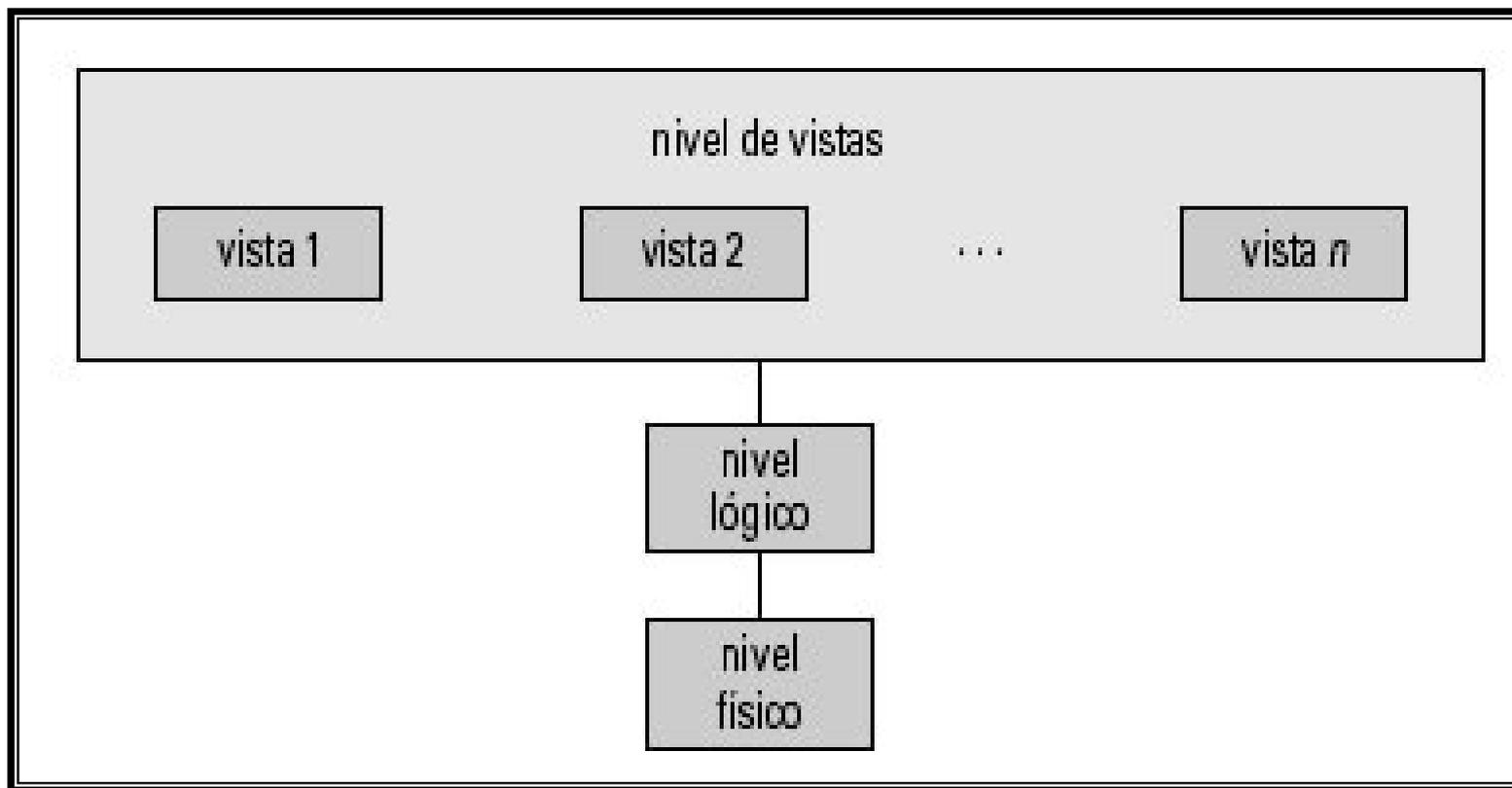


Componentes típicos de un DBMS



- **Nivel físico:** describe cómo se almacenan realmente los datos
- **Nivel lógico:** describe qué datos se almacenan en la base de datos y las relaciones existentes entre ellos
- **Nivel de vistas:** los programas de aplicación ocultan detalles de los tipos de datos. Las vistas también pueden ocultar información por razones de seguridad.

- Una arquitectura para un sistema de bases de datos



- Similar a los tipos y variables en los lenguajes de programación
- **Esquema**, la estructura lógica de la base de datos
 - Ejemplo: La base de datos se compone de información acerca de un grupo de clientes y cuentas, y de las relaciones entre ellos
 - Análogo a la información del tipo de una variable en un programa
 - **Esquema físico**: diseño de la base de datos a nivel físico
 - **Esquema lógico**: diseño de la base de datos a nivel lógico

- **Instancia**, el contenido real de la base de datos en un instante de tiempo determinado
 - Análogo al valor de una variable
- **Independencia física de los datos**, la capacidad de modificar el esquema físico sin cambiar el esquema lógico
 - Las aplicaciones dependen del esquema lógico
 - En general, las interfaces entre los diferentes niveles y componentes deben definirse adecuadamente de modo que los cambios en algunas partes no influyeran otras seriamente

- Colección de herramientas conceptuales para describir
 - los datos
 - las relaciones de datos
 - la semántica de los datos
 - consistencia entre los datos
- Modelo relacional
- Modelo entidad-relación (diseño de bases de datos)
- Modelo de datos orientado a objetos
- Modelo de datos semiestructurados (XML)
- Metodología Object Role Modeling (ORM)
 - <http://www.orm.net/>

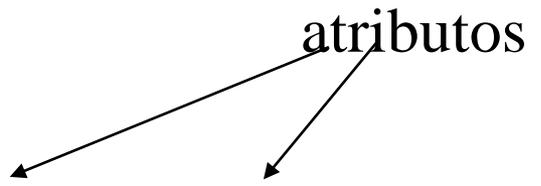
SQL Lenguaje de manipulación de datos (DML)

- Lenguaje para acceder o manipular los datos organizados mediante el modelo de datos apropiado
 - El DML también se conoce como lenguaje de consultas
- Existen dos clases de lenguajes
 - **Procedimentales** – el usuario especifica qué datos se necesitan y cómo han de obtenerse dichos datos
 - **Declarativos (no procedimentales)** – el usuario especifica qué datos se necesitan sin especificar cómo se han de obtener
- SQL es el lenguaje de consultas más utilizado

- Notación de especificación para definir el esquema de la base de datos. El compilador DDL genera un conjunto especial de tablas denominado *diccionario de datos*
 - El diccionario de datos contiene metadatos (datos acerca de los datos)
 - Esquema de base de datos
 - Lenguaje de *almacenamiento y definición* de datos
 - Especifica la estructura de almacenamiento y los métodos de acceso utilizados
 - Restricciones de integridad
 - Restricciones de dominio, Integridad referencial, Asertos
 - Autorización
-

- Ejemplo de tabla de datos en el modelo relacional

atributos



| <i>id_cliente</i> | <i>nombre_cliente</i> | <i>calle_cliente</i> | <i>ciudad_cliente</i> | <i>número_cuenta</i> |
|-------------------|-----------------------|----------------------|-----------------------|----------------------|
| 19.283.746 | González | Arenal, 12 | La Granja | C-101 |
| 19.283.746 | González | Arenal, 12 | La Granja | C-201 |
| 67.789.901 | López | Mayor, 3 | Peguerinos | C-102 |
| 18.273.609 | Abril | Preciados, 123 | Valsaín | C-305 |
| 32.112.312 | Santos | Mayor, 100 | Peguerinos | C-217 |
| 33.666.999 | Rupérez | Ramblas, 175 | León | C-222 |
| 01.928.374 | Gómez | Carretas, 72 | Cerceda | C-201 |

Ejemplo de base de datos relacional

| <i>id_cliente</i> | <i>nombre_cliente</i> | <i>calle_cliente</i> | <i>ciudad_cliente</i> |
|-------------------|-----------------------|----------------------|-----------------------|
| 19.283.746 | González | Arenal, 12 | La Granja |
| 67.789.901 | López | Mayor, 3 | Peguerinos |
| 18.273.609 | Abril | Preciados, 123 | Valsaín |
| 32.112.312 | Santos | Mayor, 100 | Peguerinos |
| 33.666.999 | Rupérez | Ramblas, 175 | León |
| 01.928.374 | Gómez | Carretas, 72 | Cerceda |

(a) La tabla *cliente*

| <i>número_cuenta</i> | <i>saldo</i> |
|----------------------|--------------|
| C-101 | 500 |
| C-215 | 700 |
| C-102 | 400 |
| C-305 | 350 |
| C-201 | 900 |
| C-217 | 750 |
| C-222 | 700 |

(b) La tabla *cuenta*

| <i>id_cliente</i> | <i>número_cuenta</i> |
|-------------------|----------------------|
| 19.283.746 | C-101 |
| 19.283.746 | C-201 |
| 01.928.374 | C-215 |
| 67.789.901 | C-102 |
| 18.273.609 | C-305 |
| 32.112.312 | C-217 |
| 33.666.999 | C-222 |
| 01.928.374 | C-201 |

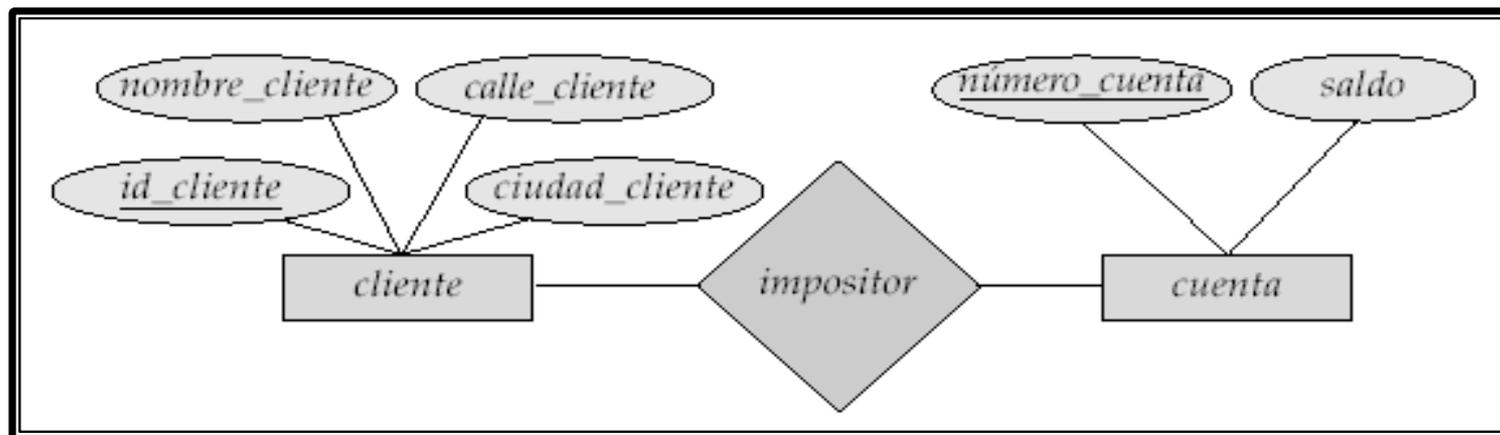
(c) La tabla *impositor*

- **SQL**: lenguaje no procedimental ampliamente utilizado
 - Ejemplo: Encontrar el nombre de el cliente con id_cliente 192-83-7465

```
select cliente.nombre_cliente
from cliente
where cliente.id_cliente = '192-83-7465'
```
- Los programas de aplicación generalmente acceden a la base de datos a través de
 - Extensiones de lenguajes que permitan SQL empotrado
 - Interfaces de programas de aplicación (p. e. ODBC/JDBC) que permiten el envío de consultas SQL a una base de datos

- Proceso de diseño de la estructura general de una BD:
- *Diseño lógico*: decidir el esquema de la base de datos. Requiere encontrar una “buena” colección de esquemas de relación.
 - Decisión de negocio – ¿Qué atributos se deberían registrar en la base de datos?
 - Decisión informática – ¿Qué relación de esquemas se deberían utilizar y cómo se deberían distribuir los atributos entre los distintos esquemas de relación?
 - *Diseño físico*: decidir sobre las características físicas de la base de datos
-

- Modela una empresa como una colección de entidades y relaciones
 - Entidad: una “cosa” o un “objeto” en la empresa distinguible de otros objetos
 - Se describe mediante un conjunto de atributos
 - Relación: una asociación entre varias entidades
- Se representa gráficamente mediante un diagrama entidad-relación:



Modelos de datos relacional orientado a objetos

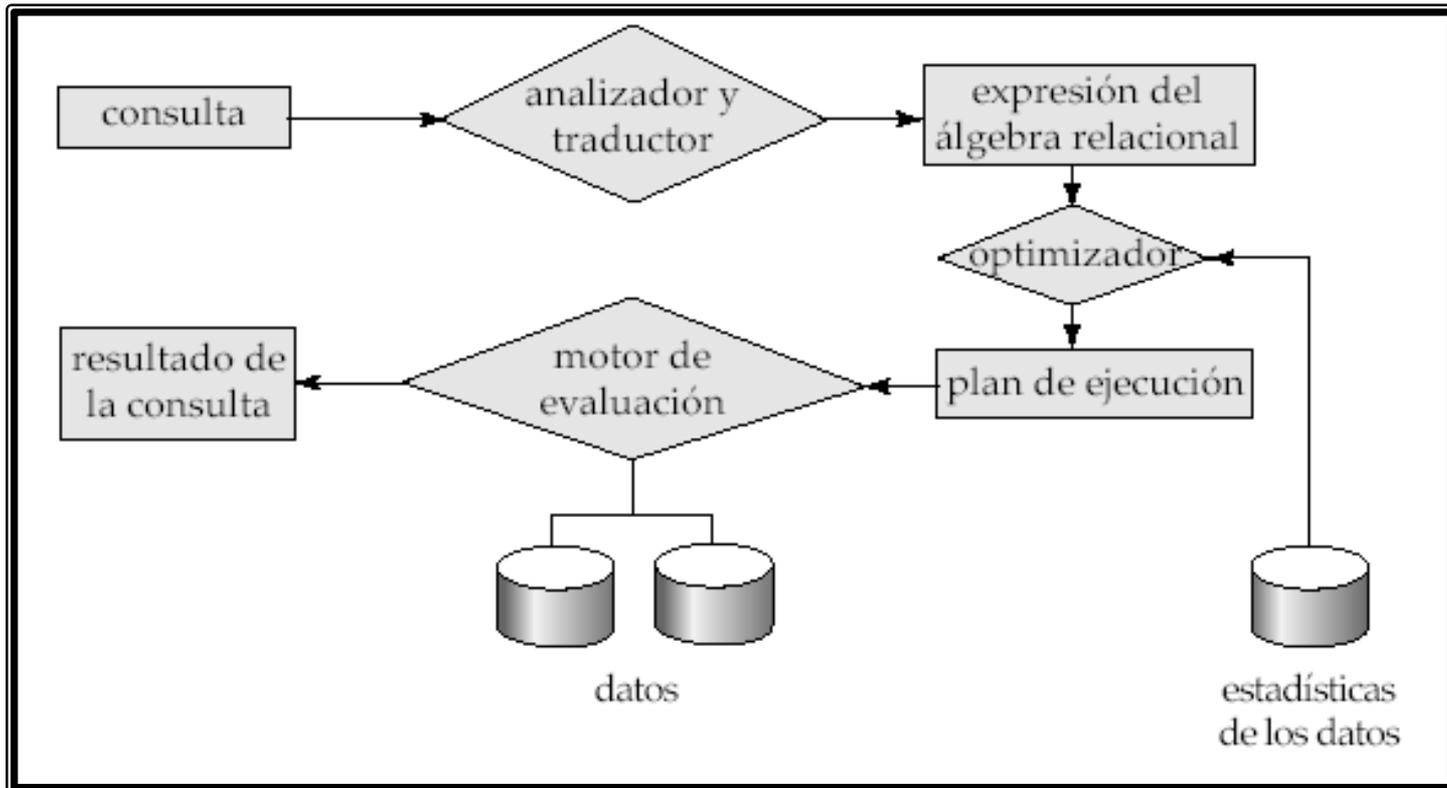
- Extiende el modelo de datos relacional incluyendo orientación a objetos y construcciones que manejan otros tipos de datos adicionales
- Permite atributos de tuplas con tipos complejos, incluyendo valores no atómicos como son las relaciones anidadas
- Preserva los fundamentos relacionales, en particular el acceso declarativo a los datos, aunque extiende la capacidad de modelado
- Proporciona compatibilidad hacia arriba con lenguajes relacionales preexistentes

- Definido por el consorcio WWW (W3C)
- Originariamente como lenguaje de marcado de documentos, no como lenguaje de base de datos
- La posibilidad de especificar nuevas etiquetas y crear estructuras de etiquetas anidadas convierten a XML en un mecanismo perfecto para el intercambio de **datos**, no solo de documentos
- XML se ha convertido en la base de una nueva generación de formatos de intercambio de datos
- Existen una gran variedad de herramientas de análisis, presentación y consulta de documentos/datos en XML

- El **gestor de almacenamiento** es el módulo que proporciona la interfaz entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema.
- El gestor de almacenamiento es responsable de las siguientes tareas:
 - La interacción con el gestor de ficheros
 - El almacenamiento, recuperación y actualización eficiente de los datos
- Temas:
 - Almacenamiento
 - Organización de archivos
 - Indexación y asociación

Procesamiento de consultas

1. Análisis y traducción
2. Optimización
3. Evaluación



- Alternativas de evaluación de una consulta
 - Expresiones equivalentes
 - Algoritmos diferentes para cada operación
- La diferencia de coste entre una forma buena y una mala de evaluar una consulta puede ser enorme
- Se necesita estimar el coste de las operaciones
 - Depende de forma crítica de la información estadística sobre las relaciones que debe mantener la base de datos
 - Se necesita estimar las estadísticas para los resultados intermedios para estimar el coste de expresiones complejas

- Una **transacción** es una colección de operaciones que se llevan a cabo como una única función lógica en una aplicación de base de datos.
- El **componente de gestión de transacciones** asegura que la base de datos permanezca en un estado consistente (correcto) a pesar de los fallos del sistema (p.e., fallos de energía y caídas del sistema operativo) y de los fallos en las transacciones.
- El **gestor de control de concurrencia** controla la interacción entre las transacciones concurrentes para asegurar la consistencia de la base de datos.

La arquitectura de una base de datos se ve muy influenciada por el sistema informático subyacente sobre el que se está ejecutando:

- Centralizado
- Cliente-servidor
- Paralelo (multi-procesador)
- Distribuido

- Oracle
- MS SQL Server
- MySql
- PostgreSQL
- IBM DB2
- Elasticsearch
- Redis
- MS Access
- Cassandra
- SQLite
- Splunk
- MariaDB

[Fuente](#)

- Esquema de la Base de Datos:
 - Diseño total de la Base de Datos
- Instancia de la Base de Datos:
 - Colección de información almacenada en la Base de Datos en un instante particular en tiempo
- Administrador de Bases de Datos (DBA):
 - Persona que tiene tanto el control central datos como el de los programas que tienen acceso a los datos
- Lenguaje de definición de datos (DDL):
 - Usados para especificar el esquema de la base de datos

- Directorio de datos:
 - Almacena los meta datos, es el resultado de compilar el DDL
- Lenguaje de manipulación de datos (DML)
 - Permite a usuarios tener acceso a datos o manipularlos
- Query:
 - Declaración que solicita la recuperación de la información
- Lenguaje de Querys :
 - Parte del DML que implica la recuperación de datos

- Usuarios de las Bases de datos :
 - Programadores de aplicaciones / informáticos
 - Usuarios capacitados en el lenguaje de querys
 - Usuarios finales
- ANSI
 - American National Standards Institute