

Desarrollo de Software para Sistemas Distribuidos

Curso 2015

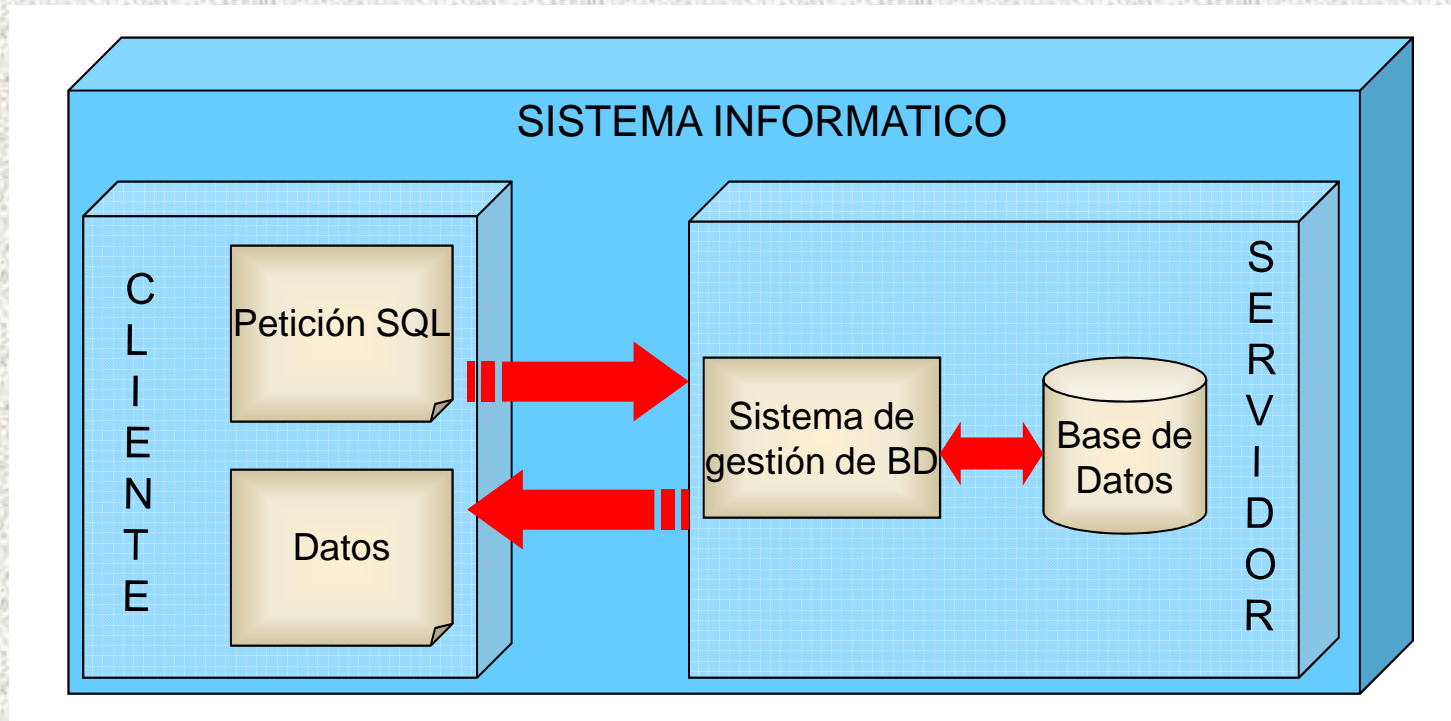
Desarrollo de Software para
Sistemas Distribuidos - 2015

Servidores de base de datos

- Sin NOS y un mecanismo de comunicación que cuente con al menos el nivel de transporte, es imposible crear la ilusión de tener un sistema totalmente transparente de la distribución
- Las bases de datos SQL o servidores SQL son el modelo dominante para crear aplicaciones cliente/servidor.

El Lenguaje SQL

SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una BD relacional.

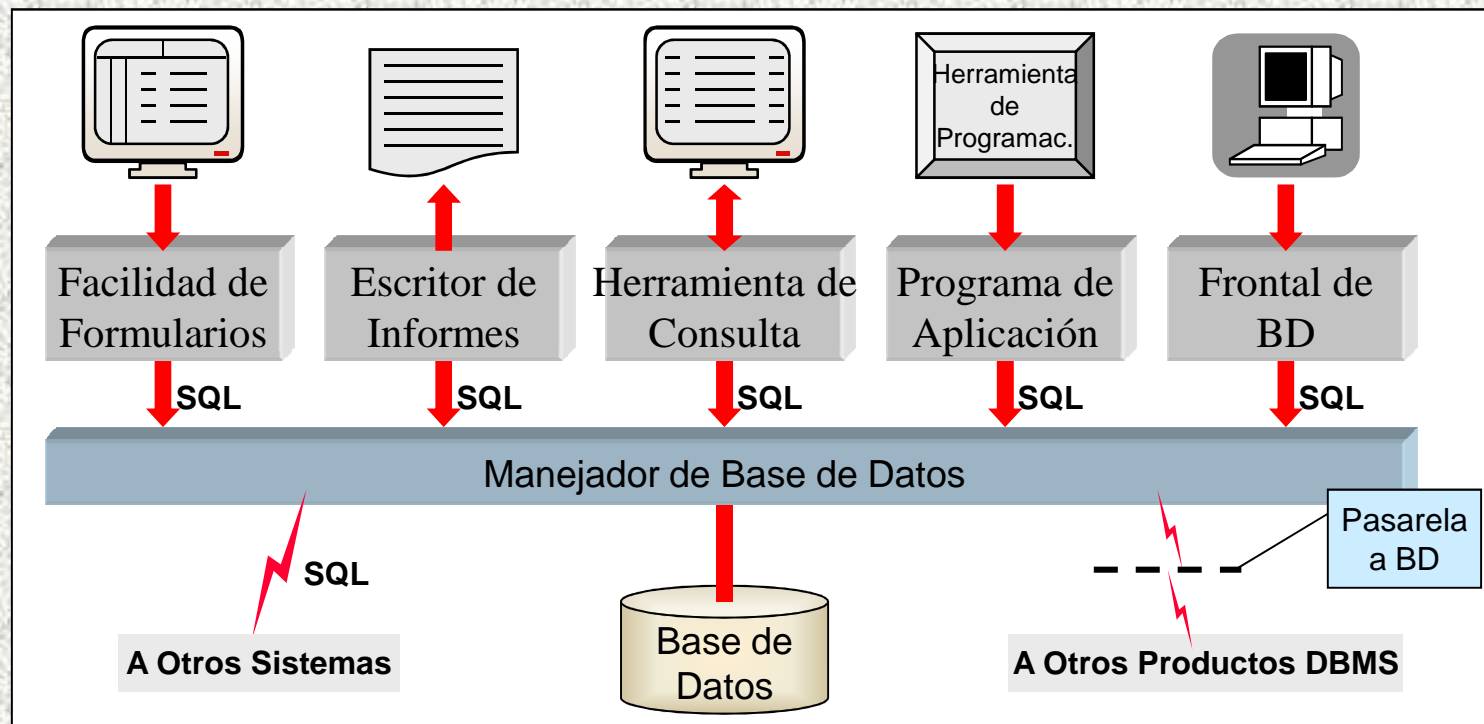


¿Qué hace el SQL?

- Es un lenguaje de consulta interactivo
- Es un lenguaje de programación de bases de datos.
- Es un lenguaje de definición y administración (manipulación) de datos.
- Ayuda a proteger los datos en un ambiente multiusuario (control de acceso).
- Garantiza la integridad y consistencia de datos en un ambiente multiusuario

Roles del SQL

- ↓ Lenguaje de consultas interactivas.
- ↓ Lenguaje de programación para Base de Datos
- ↓ Administración de Base de Datos.
- ↓ Lenguaje Cliente / Servidor.
- ↓ Lenguaje de Base de Datos distribuidas.

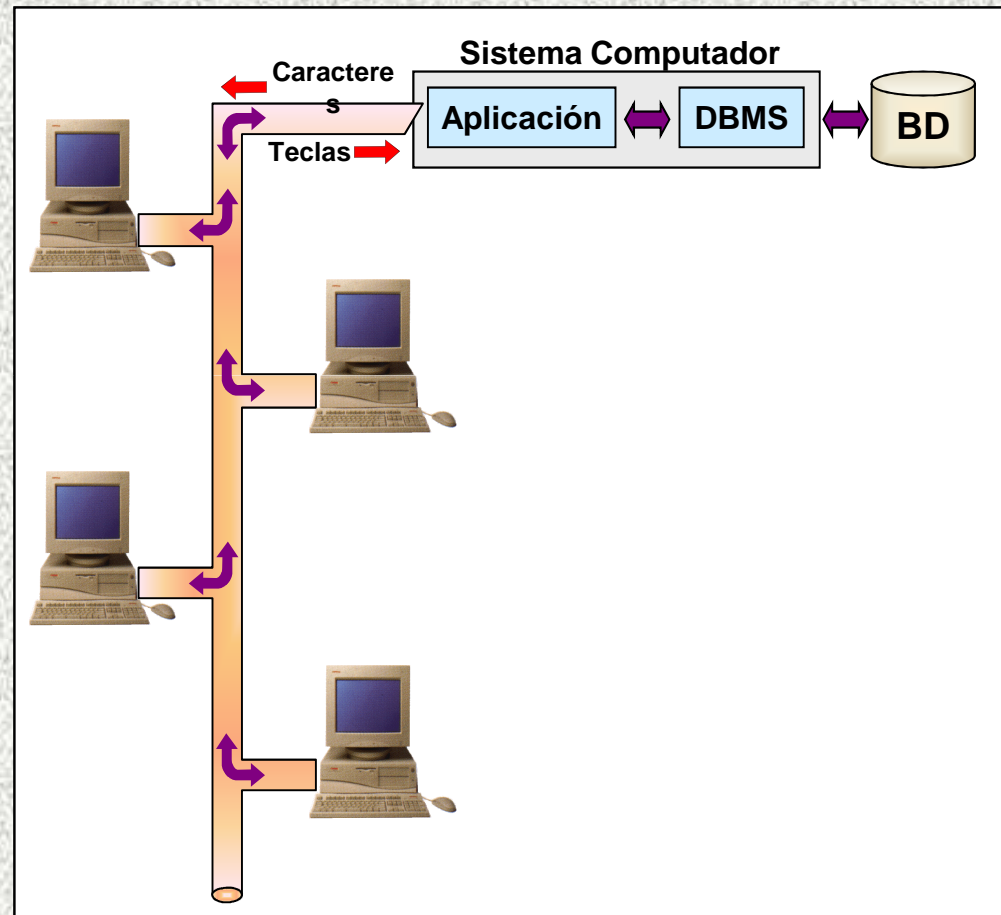


¿Qué hace un servidor SQL?

- **Controla y ejecuta comandos SQL**
- **Provee una visión lógica y física de los datos**
- **Genera planes optimizados de ejecución de los comandos SQL**
- **Manejan la recuperación, concurrencia, seguridad y consistencia de la base de datos.**
- **Controlan la ejecución de transacciones**
- **Obtienen y liberan lockeos durante la ejecución de la transacción en curso**

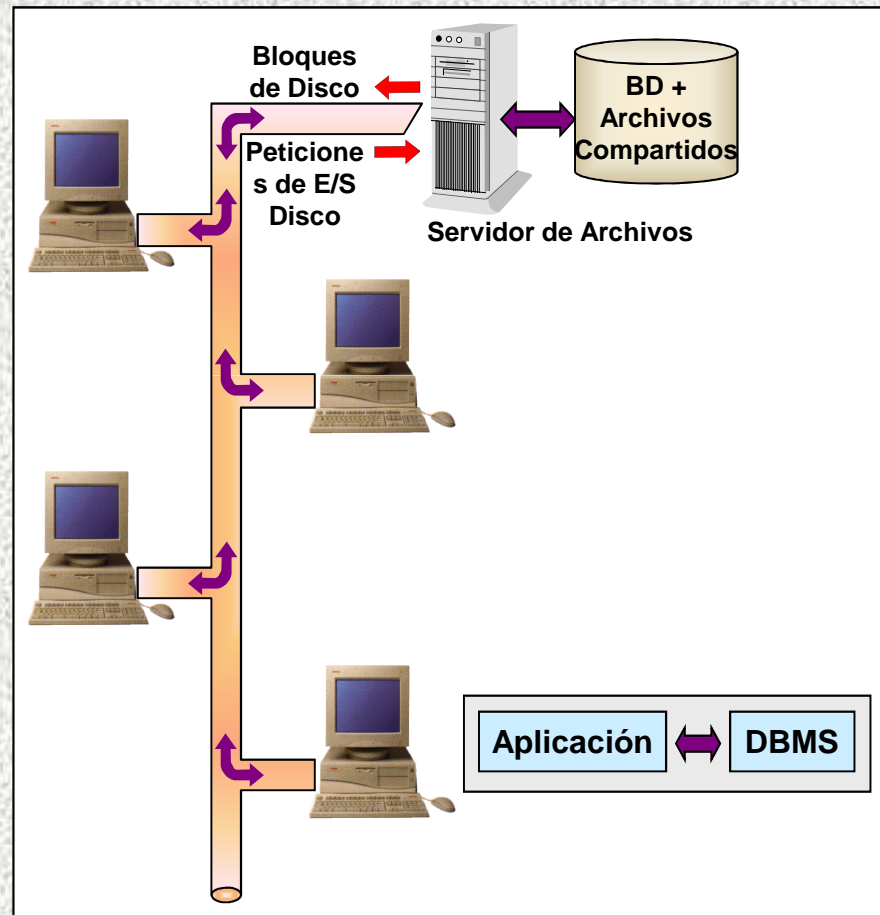
SQL y la Conexión por Red

Arquitectura Centralizada



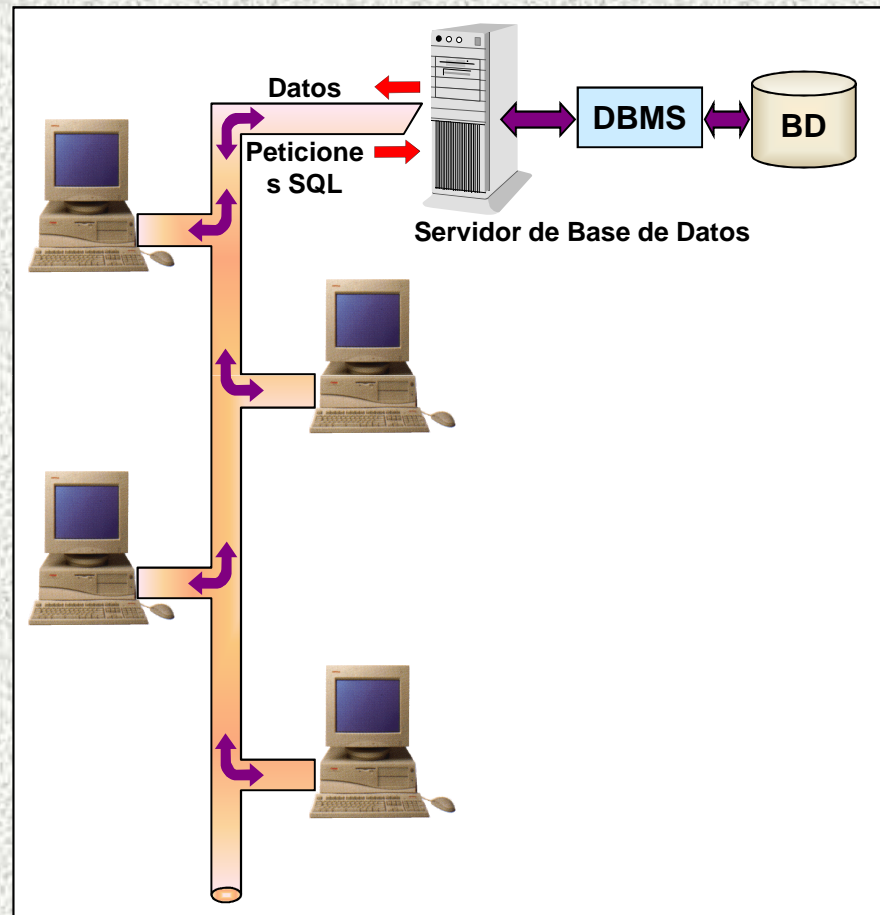
SQL y la Conexión por Red

Arquitectura de Servidor de Archivos



SQL y la Conexión por Red

Arquitectura Cliente / Servidor



¿Qué es un servidor SQL?

- Un servidor SQL es un híbrido del SQL estándar y el dialecto propio del vendedor.
- Poseen extensiones que otorgan nuevas funcionalidades mas allá del modelo relacional, como extensiones procedurales, monitores de transacciones, bases de datos de objetos y administradores de objetos distribuidos.

Arquitectura de los servidores SQL

La arquitectura se puede definir desde dos puntos de vista:

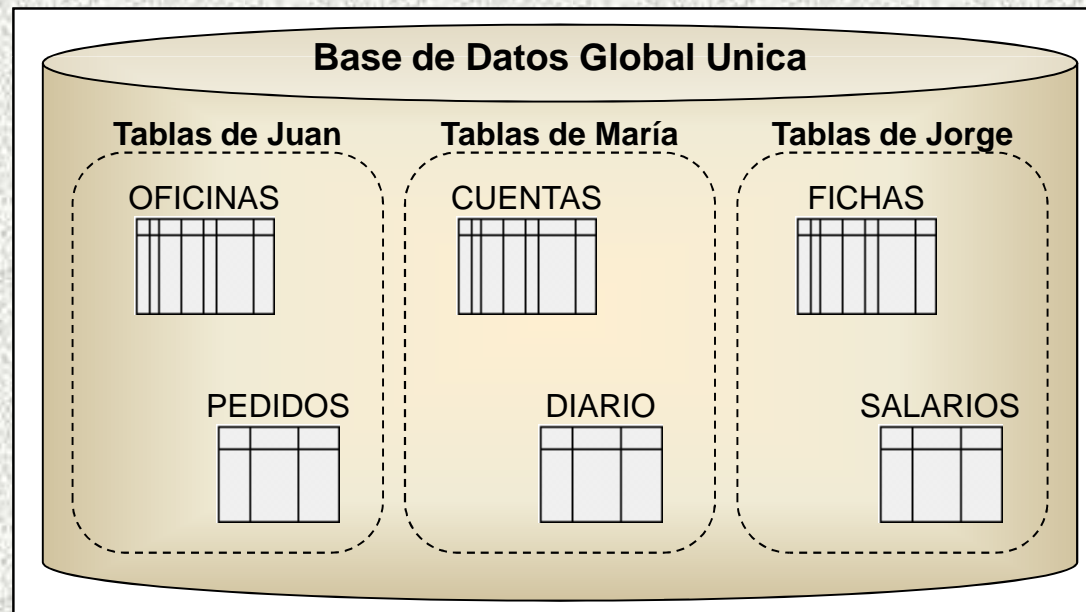
- Desde el punto de vista del almacenamiento
- Desde el punto de vista del procesamiento

Arquitectura según el almacenamiento

- Arquitectura de base de datos única
- Arquitectura de base de datos múltiple
- Arquitectura de ubicación múltiple

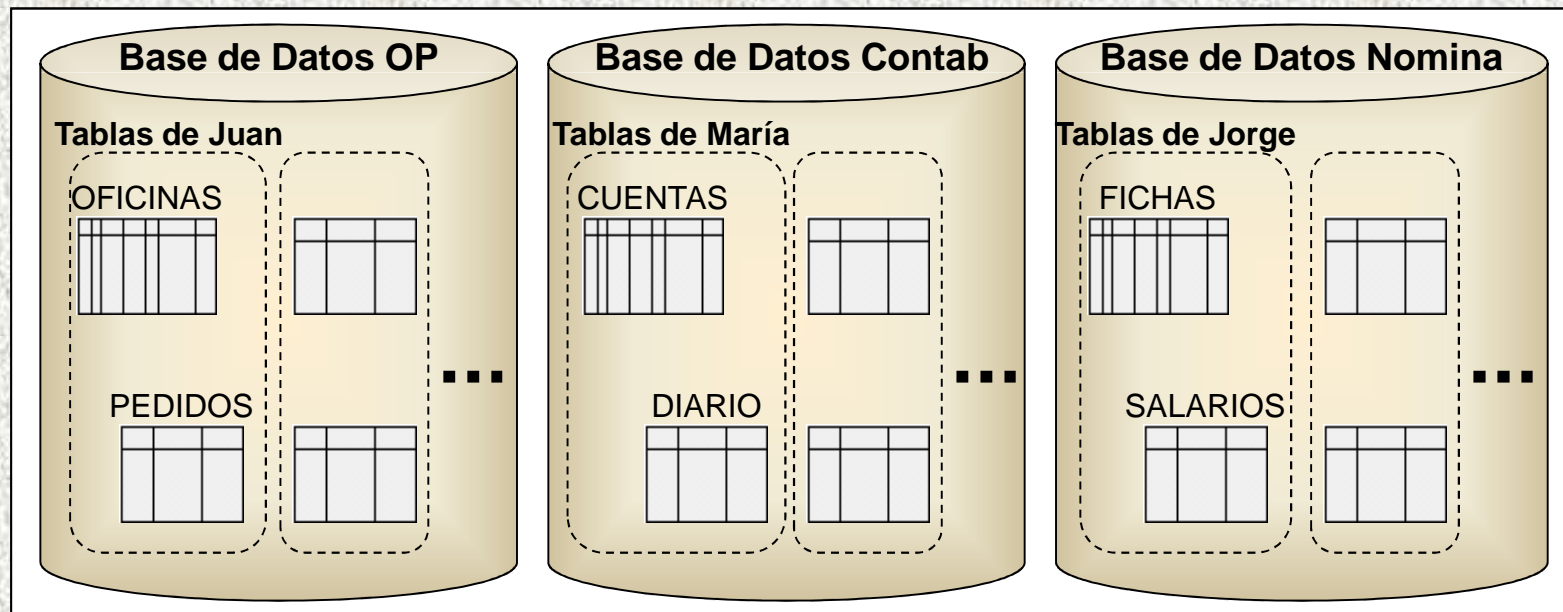
Arquitectura de BD Unica

- **Ventajas:** las tablas de diferentes aplicaciones pueden referenciarse fácilmente.
- **Desventajas:** el tamaño de la BD torna compleja su gestión en cuanto a copias de seguridad, recuperación de datos y análisis de Rendimiento.



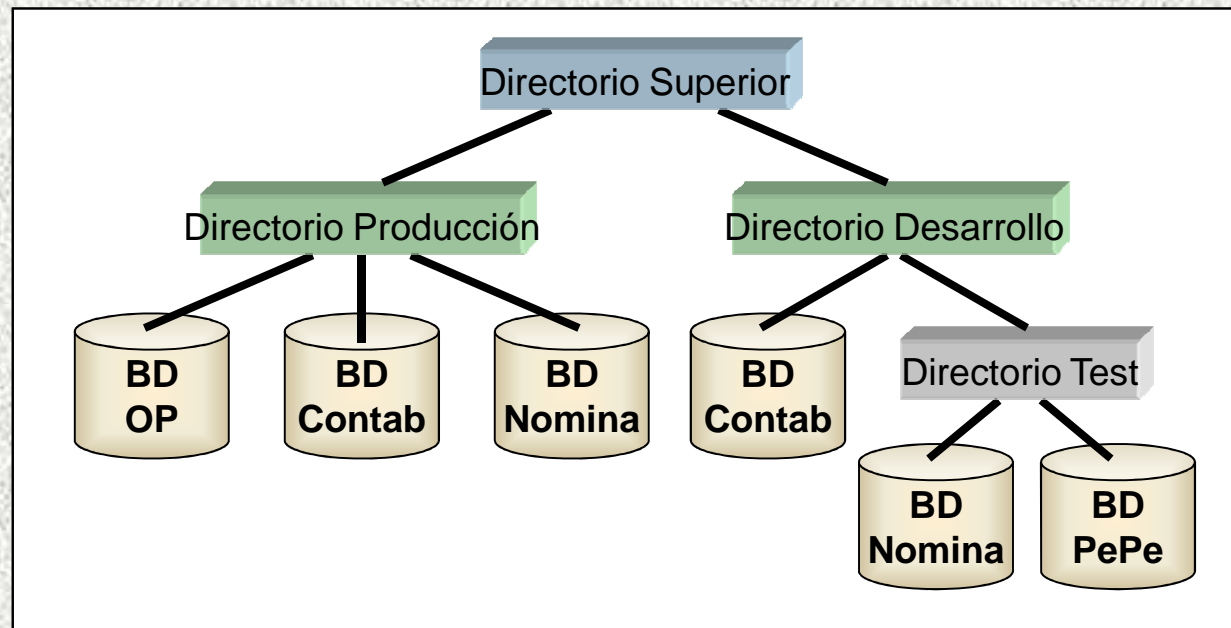
Arquitectura de BD Múltiples

- **Ventajas:** divide las tareas de gestión en partes mas pequeñas y mas manejables.
- **Desventajas:** las BD individuales pueden convertirse en islas de información.



Arquitectura de Ubicación Múltiple

- **Ventajas:** es flexible y adecuada en aplicaciones donde los usuarios requieran el uso de diversas BD y estructurar su propia información.
- **Desventajas:** además de las específicas de la arquitectura, torna compleja la administración centralizada de todas las BD.



Arquitectura según el procesamiento

- Arquitectura proceso por cliente
- Arquitectura multihilada
- Arquitectura híbrida

Arquitectura proceso por cliente

- Otorga a cada cliente su propio espacio de direcciones por proceso y la base de datos corre en un o mas procesos background separados.
- Son ejemplos de esta arquitectura: DB2, Informix y Oracle6

Arquitectura multihilada

- Provee la mejor performance para ejecutar todas las conexiones de usuarios, aplicaciones y la base de datos misma en el mismo espacio de direcciones.
- Provee su propio scheduler independiente del manejo de tareas del SO local. Ejemplos de esta arquitectura son Sybase y SQL Server

Arquitectura híbrida

Posee tres componentes:

- Escuchas multithilos que participan de una conexión inicial asignando el cliente al dispatcher
- Tareas del dispatcher que ubican los mensajes en una cola de mensajes interna y desencolan las respuestas para retornarlas al cliente.
- Administrador de procesos que toman de la cola los pedidos, los ejecutan y los ubican en la cola de salida.
- Un ejemplo de esta arquitectura es Oracle 7 en adelante.

SQL como lenguaje

- Es un lenguaje de consulta interactivo
- Es un lenguaje de definición y administración (manipulación) de datos.
- Es un lenguaje de programación de bases de datos.
 - Triggers
 - Stored Procedures

Stored Procedure

- Es una colección de sentencias SQL y lógica procedural (sentencias de control) que son compiladas, verificadas y almacenadas en el servidor de base de datos.
- Es un objeto mas de la base de datos y es almacenado en el catalogo junto a los demás (tablas, indices, vistas, etc.)

Stored Procedure

- Acepta parámetros de entrada
- Se gana enormemente en eficiencia y reducción de tráfico en la red.
- El concepto de SP fue introducido por Sybase en 1986 para mejorar la performance del SQL sobre redes.

Ejemplo de SP

```
Create procedure Asignar_org (p_uni int,  
p_cong int);  
update congreso  
set id_universidad = p_uni  
where id_congreso= p_cong;  
End procedure;  
  
create procedure registrar( @nombre varchar(40),  
                             @precio money,  
                             @stock int,  
                             @mensaje varchar(60)output  
                             ) as  
  
BEGIN  
    if(exists(select*from PRODUCTO where nombre=@nombre))  
        set @mensaje='ESTE PRODUCTO YA EXISTE'  
    else  
        begin  
            insert into PRODUCTO VALUES(@nombre,@precio,@stock)  
            set @mensaje='PRODUCTO REGISTRADO'  
        end  
END; // fin del proceso  
  
//Se debe invocar explicitamente al sp  
  
EXECUTE PROCEDURE registrar ("Mayonesa", "15.25", "50", "");  
|
```

Triggers y reglas

- Un trigger es una acción especial definida por el usuario (usualmente en la forma de SP) que son automáticamente invocadas cuando ocurre un evento relacionado con los datos de la base de datos.
- Una regla es un caso especial de trigger que se usa para chequeos simples sobre los datos.

Triggers y reglas

- Realizan tareas relacionadas con auditoria, seteo de columnas por default y también control de integridad. Se habilitan **implícitamente** por las operaciones de DELETE, INSERT y UPDATE sobre las tablas para las cuales están definidos.

Triggers: modelo de ejecución

- Existen dos modelos de ejecución de los triggers:
 - after (después de la operación para la que fue definido)
 - before (antes de la operación para la que fue definido)
- Según el modelo, se altera el orden de ejecución de las restricciones de integridad

Ejemplo de triggers

```
//Trigger de Delete

CREATE TRIGGER Elimina_auditoria_clientes
AFTER DELETE ON clientes

FOR EACH ROWINSERT INTO auditoria_clientes(nombre_anterior, seccion_anterior, usuario,
                                           modificado, proceso, Id_Cliente)
VALUES (OLD.nombre, OLD.seccion, CURRENT_USER(), NOW(), 'Eliminado', OLD.id_cliente);

INSERT INTO ausentes (nombre, seccion) VALUES(OLD.id_cliente,OLD.seccion);

-----
//Triger de insert

CREATE TRIGGER Inserta_auditoria_clientes
AFTER INSERT ON clientes

FOR EACH ROWINSERT INTO auditoria_clientes(nombre_nuevo, seccion_nueva, usuario,
                                           modificado, proceso, Id_Cliente)
VALUES (NEW.nombre, NEW.seccion, CURRENT_USER(), NOW(), 'Incluido', NEW.id_cliente);
```

SQL estático y SQL dinámico

- El SQL estático son sentencias definidas en el código y compiladas en el momento de su definición. Si los objetos que referencia no existen, no compila.
- El SQL dinámico es creado en tiempo de ejecución y ofrece mas flexibilidad a expensas de velocidad de ejecución

SQL incorporado y SQL programado

- SQL incorporado: las sentencias SQL están incorporadas al código fuente del programa, entremezcladas con las otras sentencias del lenguaje de programación
- SQL programado: las sentencias SQL se almacenan como objetos de la base de datos (triggers, Stored procedure, reglas)

SQL programado Vs SQL embebido(dinámico o estático)

	SP	SQL Remoto	
		Embebido Estatico	Dinamico
Nombres para las funciones	SI	NO	NO
Funciones compartidas	SI	NO	NO
Parametros de E/S	SI	NO	NO
Catalogado	SI	SI	NO
Logica procedural	En el objeto	Externa	Externa
Flexibilidad	Baja	Baja	Alta
Nivel de abstraccion	Alto	Bajo	Bajo
Estandarizacion	NO	SI	SI
Performance	Rapido	Mediano	Lento
Trafico	1 Request/Reply por varios comandos SQL	1 Request/Reply por cada comando SQL	1 Request/Reply por cada comando SQL

SQL y el acceso concurrente

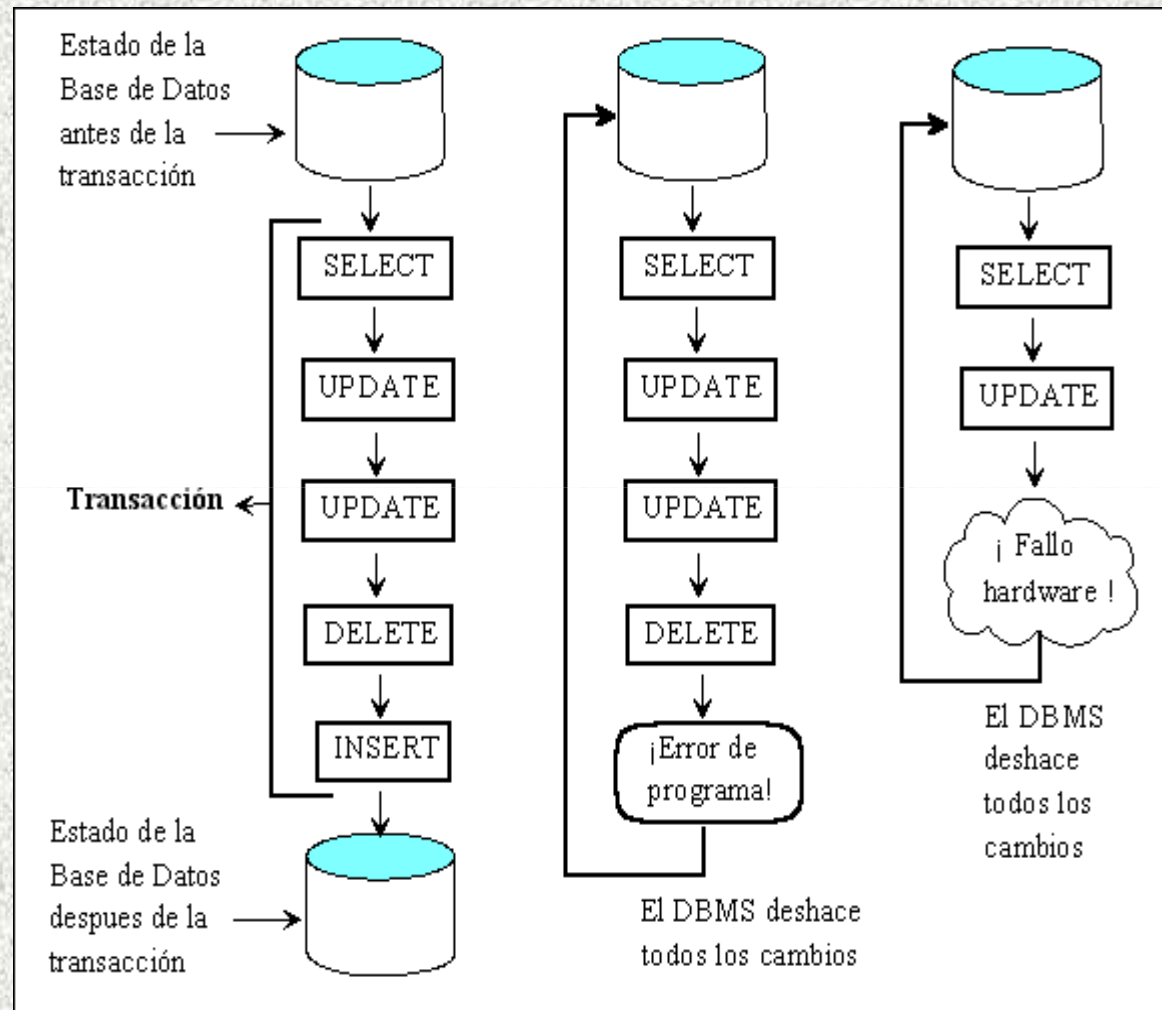
- Ayuda a proteger los datos en un ambiente multiusuario (control de acceso).
- Garantiza la integridad y consistencia de datos en un ambiente multiusuario

A través de los servicios brindados por el Servidor SQL que:

- Controla la ejecución de transacciones
- Obtiene y libera lockeos durante la ejecución de la transacción en curso
- Protege la base de datos de accesos no autorizados.

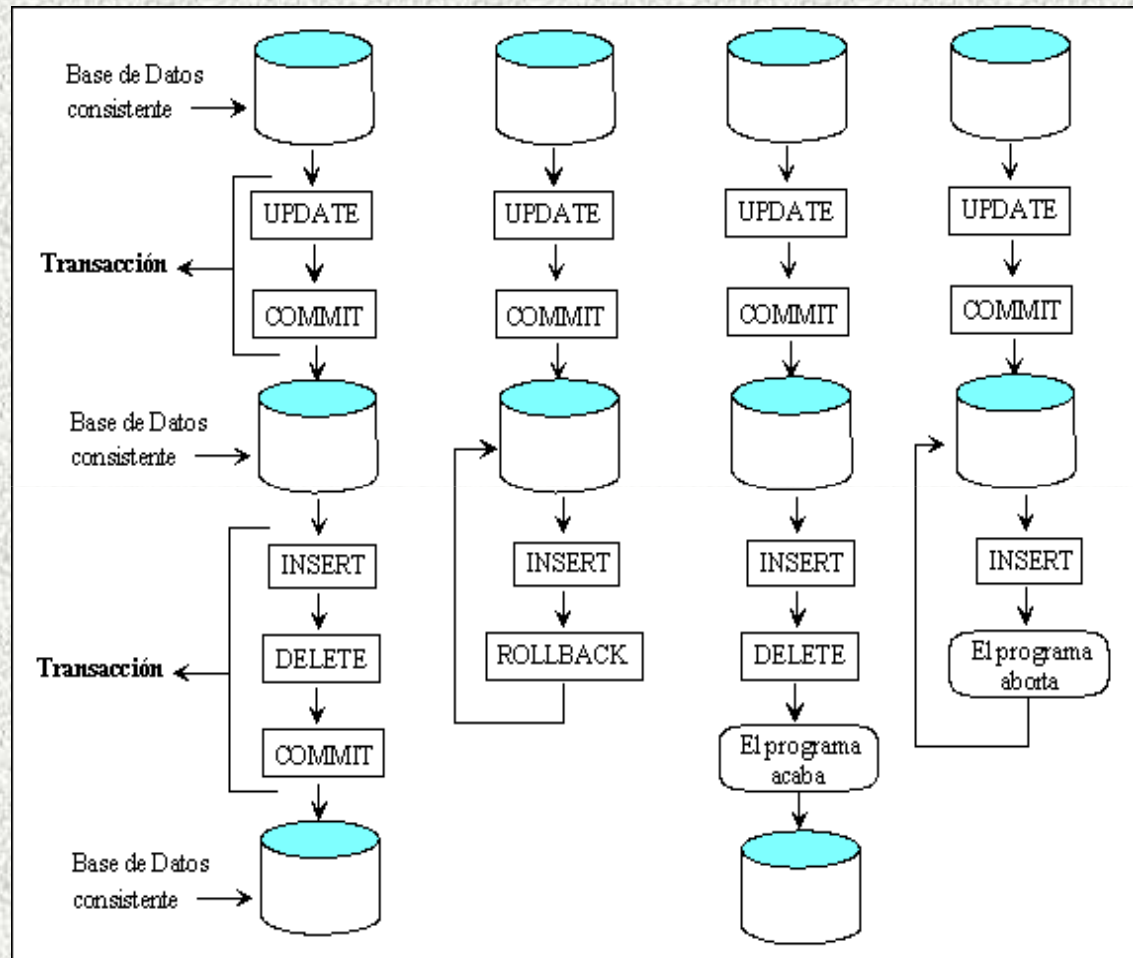
Transacciones

Una transacción es una secuencia de una o más sentencias SQL que juntas forman una unidad de trabajo para el DBMS, el cuál asume que todas las sentencias deben completarse para asegurar la consistencia e integridad de la BD.



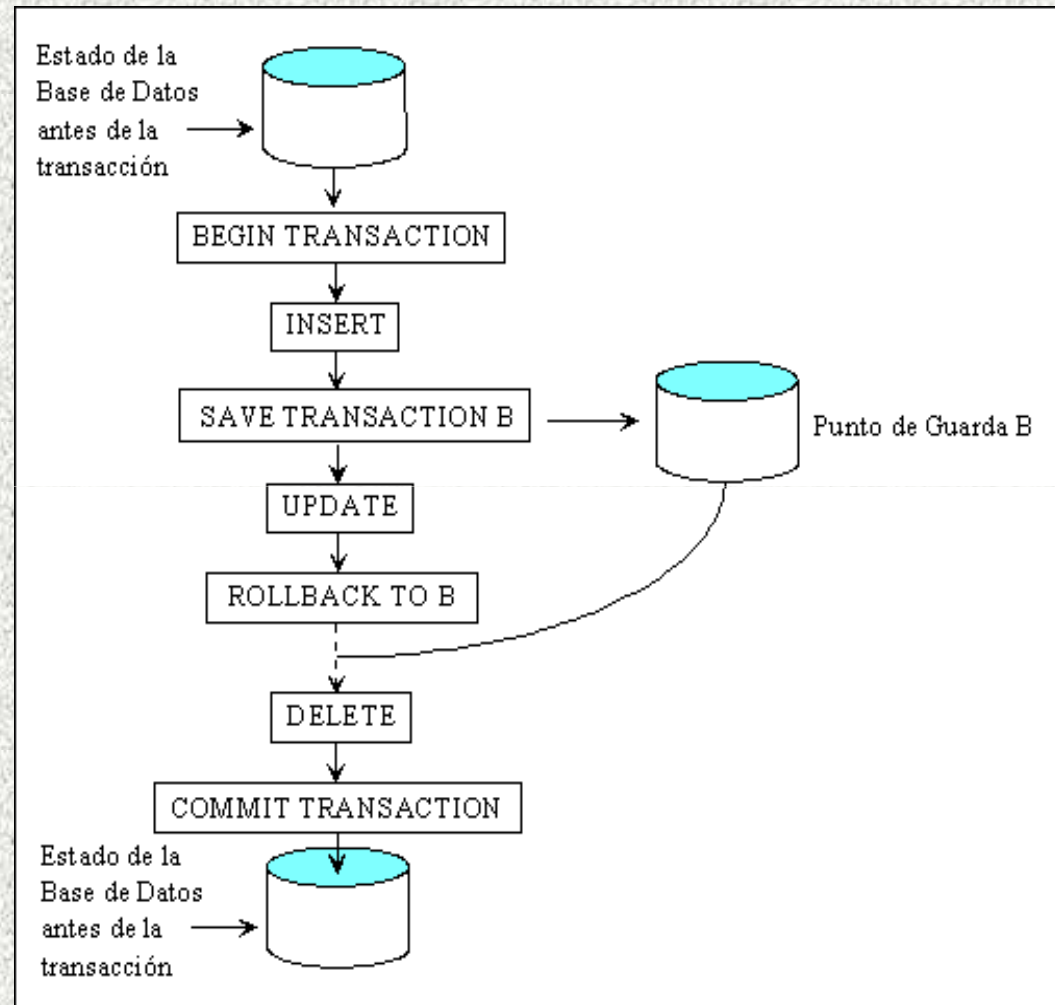
Transacciones Estándar

- SQL estándar soporta las transacciones de base de datos mediante las sentencias COMMIT y ROLLBACK.



Transacciones No Estándar

En SQL interactivo se asume que cada sentencia en una transacción Sybase extiende el modelo de transacciones del estándar SQL incluyendo otras sentencias para el procesamiento de transacciones



Transacciones y Procesamiento Multiusuario

- Cuando dos o mas usuarios acceden concurrentemente, se pueden presentar básicamente tres problemas:

- **Actualización Perdida:** cuando dos aplicaciones leen datos de la BD, los utilizan para un cálculo y luego los actualizan.
- **Datos No Cumplimentados:** cuando una aplicación lee actualizaciones no cumplimentadas de otra y actúa en base a los datos leídos.
- **Datos Inconsistentes:** cuando durante una consulta otra aplicación cumple transacciones que afectan a los datos de la consulta.

Transacciones y Procesamiento Multiusuario

"Durante un transacción el usuario verá un vista completamente consistente de la base de datos. El usuario nunca verá modificaciones no cumplimentadas de otros usuarios, e incluso los cambios cumplimentados efectuados por otros, no afectarán a los datos examinados por el usuario en mitad de una transacción."

Niveles de Cerramiento

- El cerramiento puede ser implementado a distintos niveles de la base de datos. Los cuales pueden ser:
 - Cerramiento a nivel de la base de datos.
 - Cerramiento a nivel de tabla.
 - Cerramiento a nivel de fila.
 - Cerramiento a nivel de columnas.

Esquemas de Cerramientos

Para aumentar el acceso concurrente a una BD, se utilizan esquemas con mas de un tipo de cierre, siendo habitual en tal sentido implementar al menos dos tipos de cierre: Cierre Compartido y Cierre Exclusivo.

T r a n s a c c i ó n A		Transacción B		
	Acción	No Cerrado	Cierre Compartido	Cierre Exclusivo
	No Cerrado	SI	SI	SI
	Cierre Compartido	SI	SI	NO
	Cierre Exclusivo	SI	NO	NO

SQL como middleware

- ¿Cómo resuelve cada cliente el hecho de acceder a datos en servidores SQL de múltiples vendedores?
- La respuesta es a través del middleware específico de la base de datos (API)

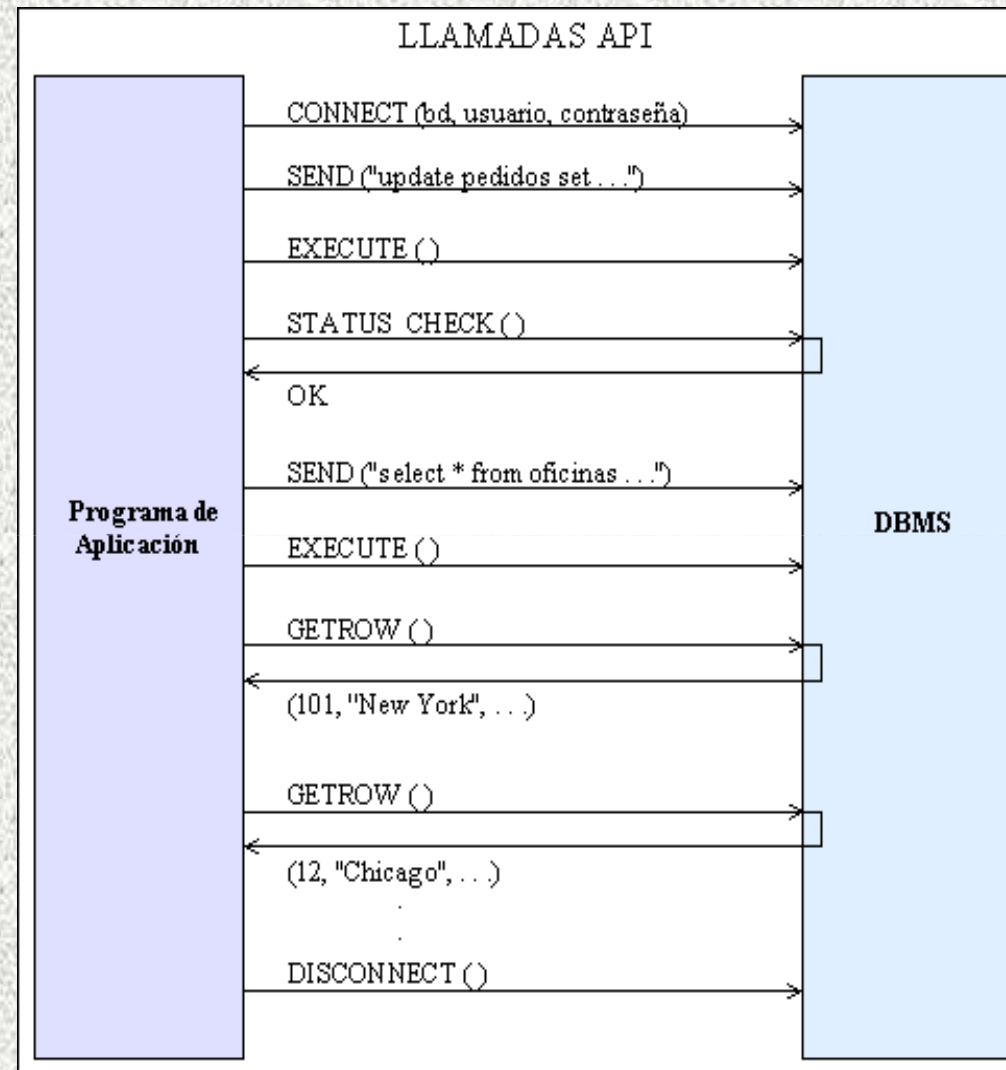
SQL como middleware

- Un middleware comienza con una API del lado del cliente usada para invocar el servicio y cubre la transmisión del requerimiento sobre la red y la respuesta resultante.
- Lo que necesitamos conocer es cual es la API y como intercambia los request/reply con el servidor.

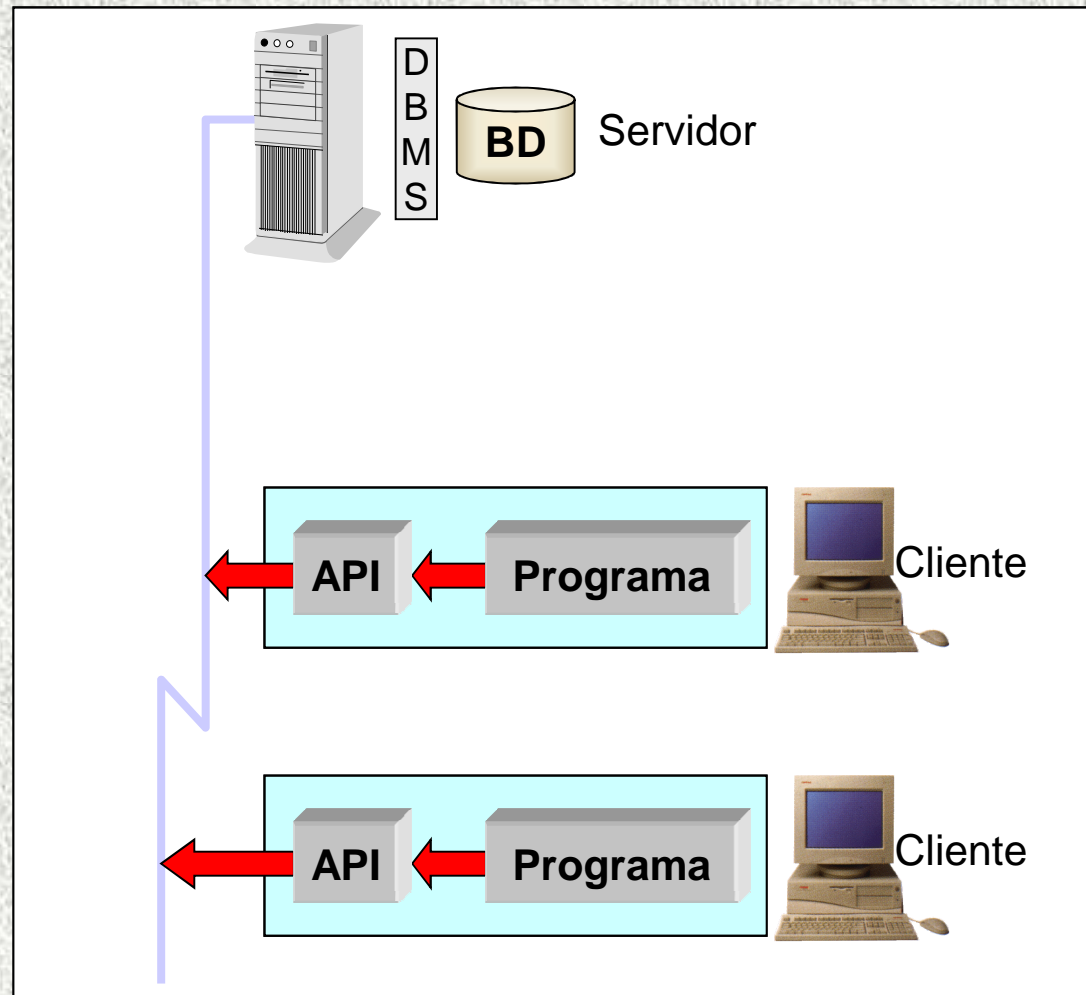
Interfaz de Programa de Aplicación (API)

- El programa se comunica con el DBMS mediante un conjunto de llamadas de función denominadas interfaz de programa de aplicación (API).
 - Análisis Sintáctico.
 - Valida la sentencia contra el catálogo del sistema.
 - Optimiza la sentencia, determinando la mejor forma de ejecutarla.
 - Genera un plan de aplicación para la sentencia, esto es un secuencia de los pasos a seguir para su realización.
 - Ejecuta el plan de aplicación.

API de SQL



Un API SQL en una Arquitectura Cliente / Servidor



¿En que consiste realmente el API SQL?

- El API surge por las características procedurales que se apartan del estándar.
- Hay dos mecanismos para soportar el uso de SQL desde un lenguaje de programación:
 - SQL Embebido (ESQL)
 - Call-Level Interface (CLI)

Open DataBase Connectivity

- El ESQL se sustenta el estándar ISO ya que no existen características procedurales. En cuanto al CLI hay distintas alternativas, una de las cuales es ODBC de MicroSoft.
- ODBC (Open DataBase Connectivity) es una API estandar multivendedor.

Open Client

- Es el software propietario (de cada BD) que garantiza la conectividad entre cliente y servidor
- En definitiva dependerá de las aplicaciones cual de ellos usara y a su vez, la aplicación se verá condicionada por el API que utilice la herramienta de desarrollo