



# UNIDAD V

Arquitectura y  
Componentes de un SGBD

# Sistema de Gestión de BD – SGBD

## (Data Base Management System - DBMS)

1. Brinda a los usuarios una vista abstracta de los datos almacenados, es decir, oculta detalles:

- como están almacenados y
- como se los mantiene

¿Cómo puede hacerlo?

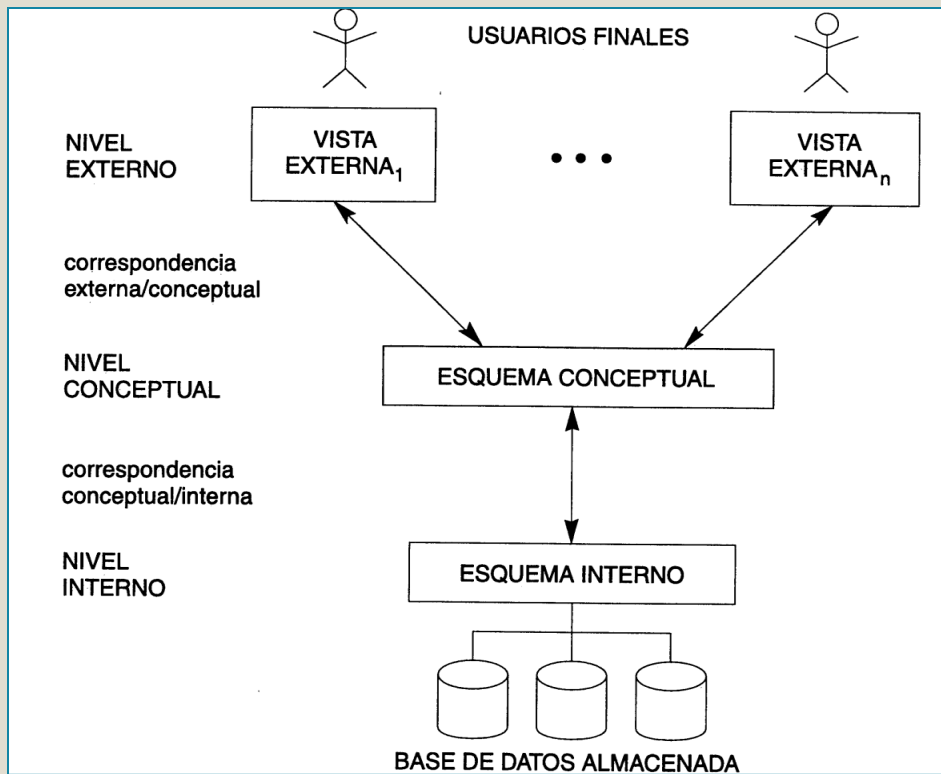
A través de la arquitectura ANSI-SPARK



**El SGBD actúa como interfaz entre los Usuarios y la Base de Datos**

# Vista Abstracta de los Datos

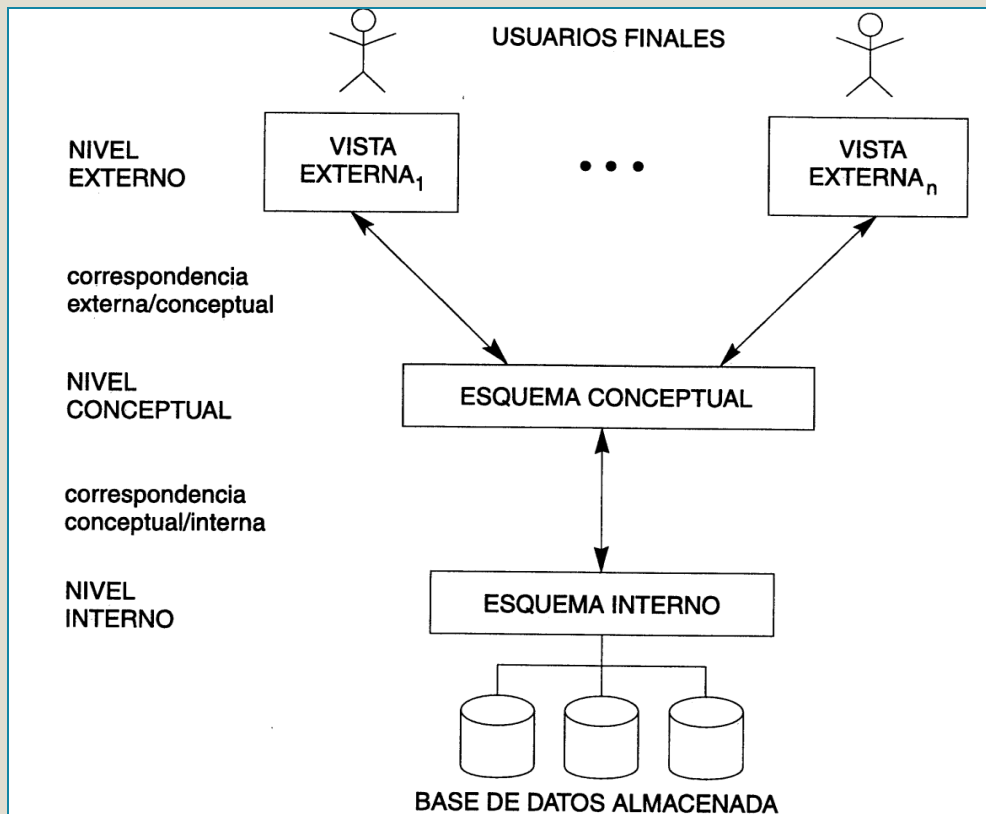
## Arquitectura ANSI-SPARK



- **Nivel Externo (Tablas):** Varios esquemas/vistas externas (**OJO! no refiere a tablas virtuales, es decir, vistas de SQL generadas por un create view...**)
  - Cada vista deja ver la porción de los datos que se necesita
  - Permisos de acceso
- **Nivel Conceptual (Tablas):**
  - Un solo esquema/vista
  - Incluye:
    - Totalidad de los datos de la bd
    - Restricciones de seguridad e integridad definidas
- **Nivel Interno (Archivos Físicos):**
  - Un esquema/vista
  - Incluye:
    - La totalidad de los datos organizados en archivos de registros

# Vista Abstracta de los Datos

## Arquitectura ANSI-SPARK



### ■ Mapeo/Correspondencia Externo-Conceptual:

- Implementado por la definición de las tablas virtuales del SQL (*create view ...*)

### ■ Mapeo/Correspondencia Conceptual-Interno:

- Implementado de diferentes formas:
  - En el create de las tablas
  - Configuración
  - Etc.

# Sistema de Gestión de BD – SGBD

2. **Software de propósito general:** El mismo SGBD puede administrar diferentes bases de datos como por ej. de Alumnos, de Bancos, etc.

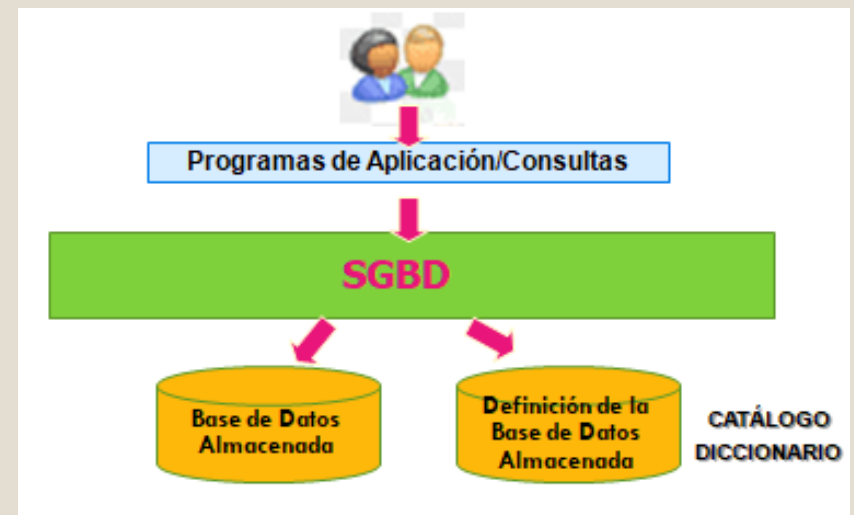
## ¿Cómo puede hacerlo?

Gracias a la información de los esquemas y correspondencias

## ¿Dónde está guardada la información de los esquemas y correspondencias?

### Catálogo o Diccionario de Datos

Ver en PostgreSQL

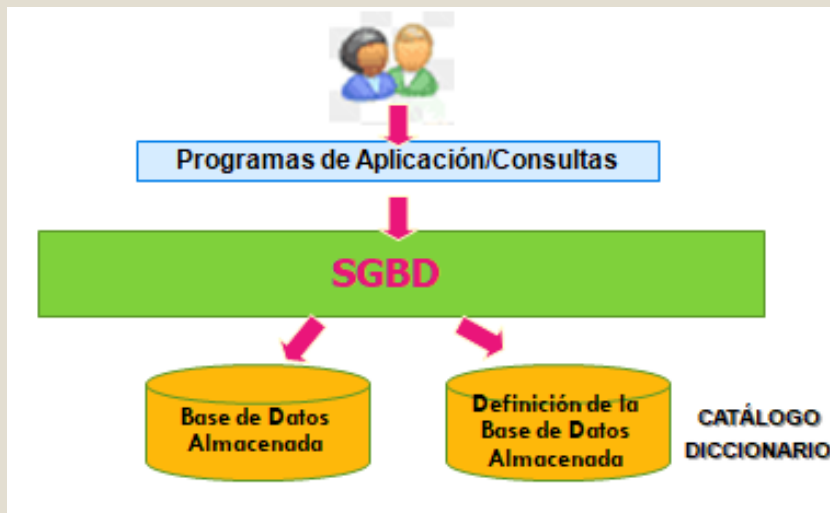


# Sistema de Gestión de BD – SGBD

¿Quién mantiene los esquemas y las correspondencias?

El **Administrador de la Base de Datos (DBA)**

Es responsable de que el sistema opere y lo haga con la performance adecuada



**++ Contribuye a la Independencia de los Datos!!!**

**-- Gasto extra en compilación y ejecución de los programas, es decir, menor eficiencia**

Nota: Mapeo y Correspondencias se utilizan como sinónimos.

# Independencia de los Datos

Inmunidad de las aplicaciones ante cambios en el nivel conceptual y/o interno



- El nivel externo permanece intacto
- Las aplicaciones no se ven afectadas ante estos cambios

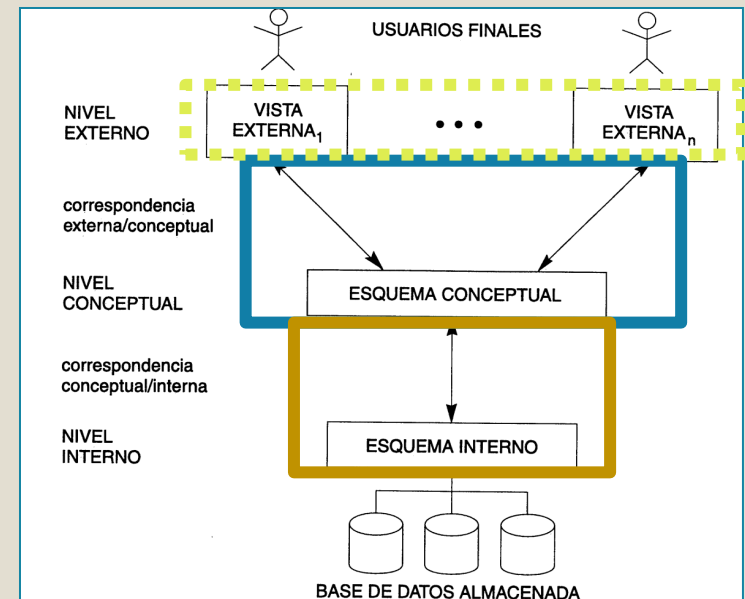
# Independencia de los Datos

- ¿Qué elementos intervienen?
  - Arquitectura de 3 niveles
  - Lenguaje SQL - Lenguaje declarativo (qué y no cómo)



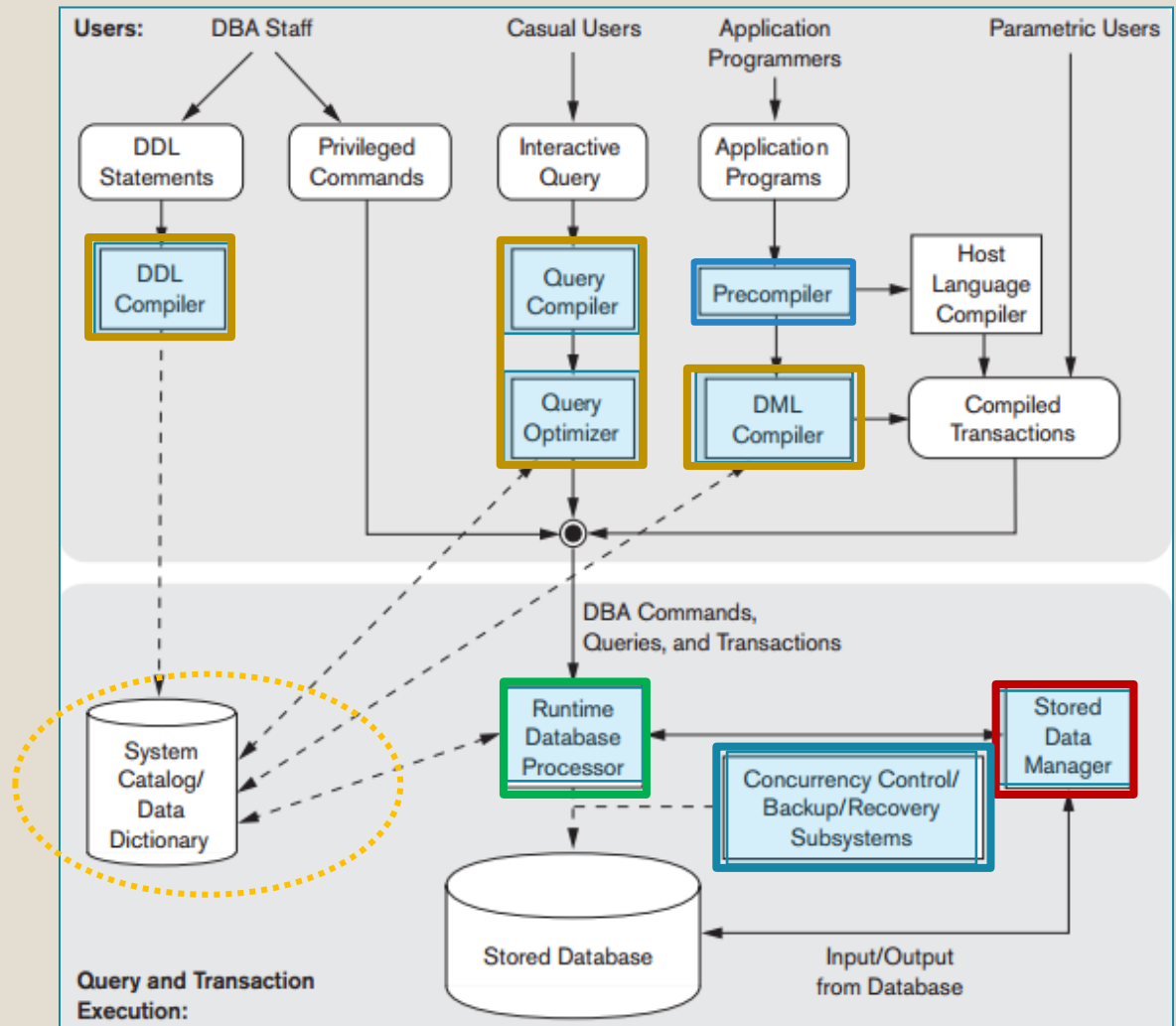
# Tipos de Independencia

- **Independencia Lógica:** Capacidad de cambiar el esquema lógico sin afectar los esquemas externos ni las aplicaciones.
  - Implica modificar el/los mapeo/s externos/conceptuales
- **Independencia Física:** Capacidad de cambiar el esquema físico sin afectar el esquema conceptual.
  - Ej. Reorganizar algún archivo.
  - Implica modificar el mapeo conceptual/interno

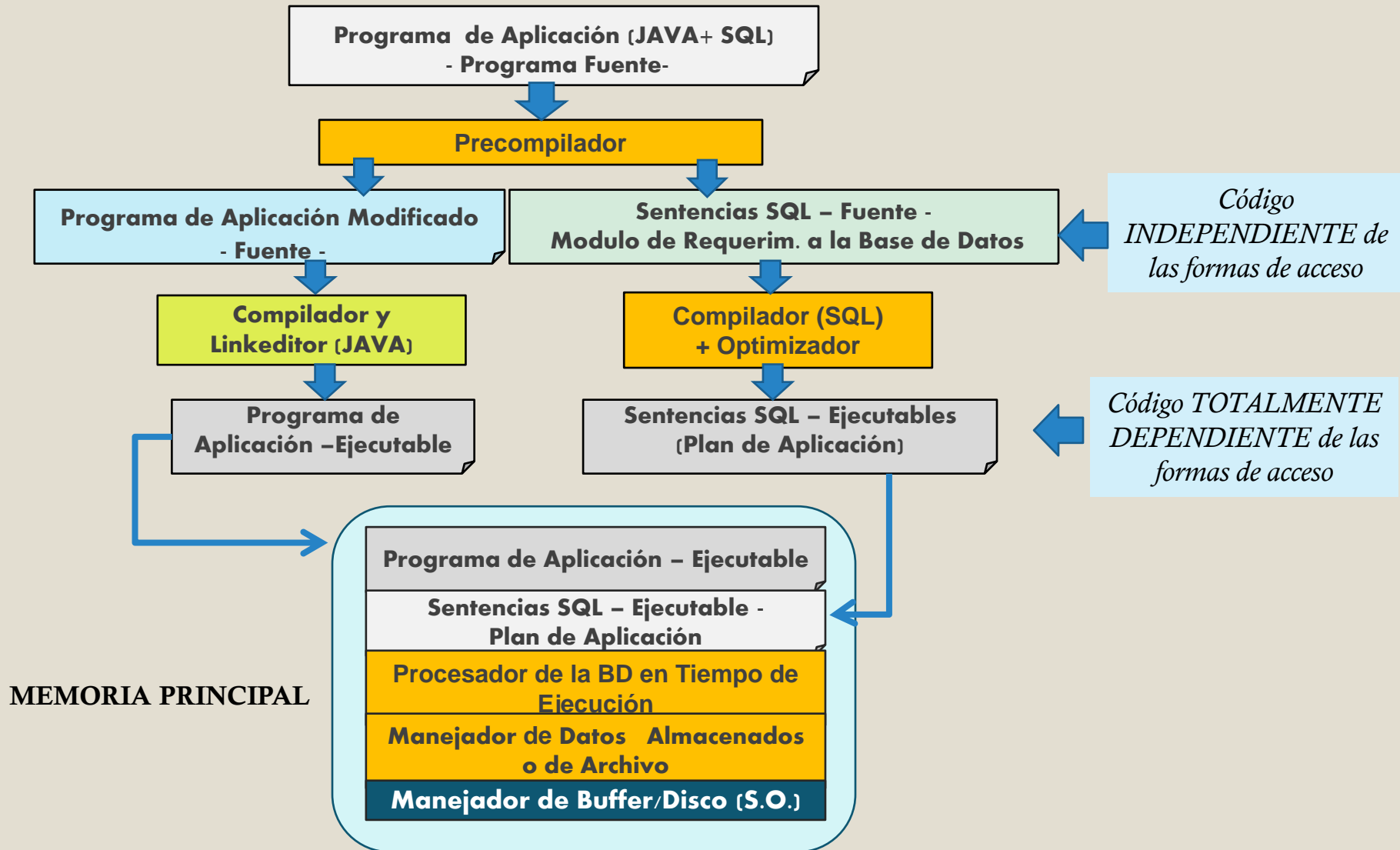


# SGBD: Componentes

- PreCompilador
- Compilador (+Optimizador)
- Procesador de BD en Tiempo de Ejecución
- Manejador de Datos Almacenados
- Subsistema de Recuperación/Concurrencia/Respaldo



# Compilación y Ejecución de un Programa que contiene SQL embebido



# Compilador SQL (Optimizador)

## Etapas en el Procesamiento de Consultas:

1. Identificar y controlar los componentes del lenguaje (sintaxis, nombres de tablas y atributos)
2. Traducir la consulta a una representación interna (Algebra). Consulta: Obtener los productos de la sucursales de San Juan.

*select num, cod, nombre, precio  
from producto natural join sucursal  
where provincia='San Juan'*

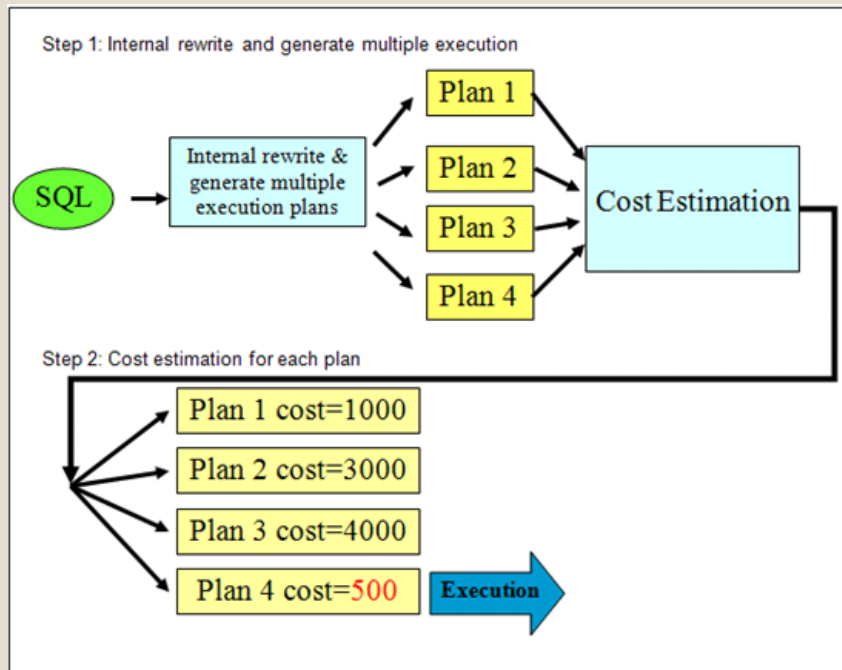


$\pi_{num, cod, nombre, precio} \sigma_{provincia='San Juan'} \text{ producto } \bowtie \text{ sucursal}$

QUERY PLAN
text
Hash Join (cost=13.78..31.88 rows=4 width=94)
Hash Cond: (producto.num = sucursal.num)
-> Seq Scan on producto (cost=0.00..16.40 rows=640 width=94)
-> Hash (cost=13.75..13.75 rows=2 width=4)
-> Seq Scan on sucursal (cost=0.00..13.75 rows=2 width=4)
Filter: ((provincia)::text = 'San Juan'::text)

3. Buscar el Plan de Ejecución Optimo (2 pasos)

# Búsqueda del Plan de Aplicación Óptimo



## Paso1: Optimización heurística

Ordena las operaciones considerando diferentes estrategias de ejecución.

Ejemplo:

- Ejecutar operaciones de **restricción tan pronto como sea posible**
- Ejecutar **primero las restricciones más restrictivas** (producen menor cantidad de tuplas)

## Paso2: Estimación de costes

Estima el costo de cada estrategia de ejecución encontrada y elige el plan (estrategia) con menor costo estimado.

Ejemplo, para implementar la restricción:

- Búsqueda Lineal
- Búsqueda Binaria
- Empleo de Índice
- Etc.

# Repasando...

- Arquitectura de 3 niveles
- Componentes:
  - PreCompilador
  - Compilador
  - Procesador de Base de Datos en Tiempo de Ejecución
  - Manejador de Datos Almacenados (Archivos): Usa al Manejador de Buffer (Disco) propio del SO
  - Subsistema de Recuperación/ Concurrencia/ Respaldo

