

# Ejecución de Consultas



# Ejecución de Consultas

## Tarea:

Encontrar un plan de consulta física eficiente (plan de ejecución) para una consulta SQL

## Objetivo:

Reducir al mínimo el tiempo de evaluación para la consulta, es decir, calcular resultado de la consulta lo más rápido posible

## Factores de costo:

Accesos a disco, operaciones de lectura / escritura, [I / O, la página de transferencia]



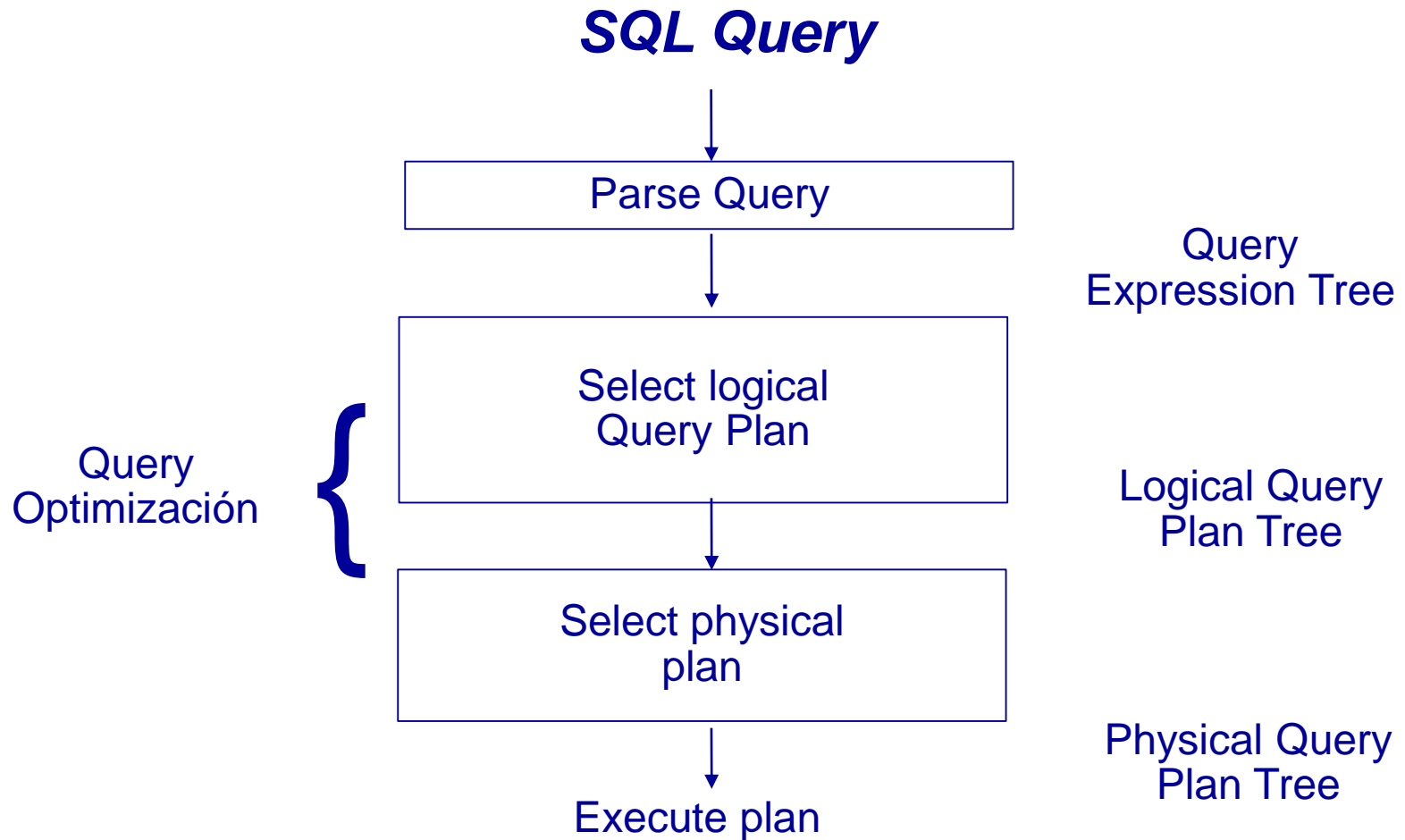
# Ejecución de Consultas

---

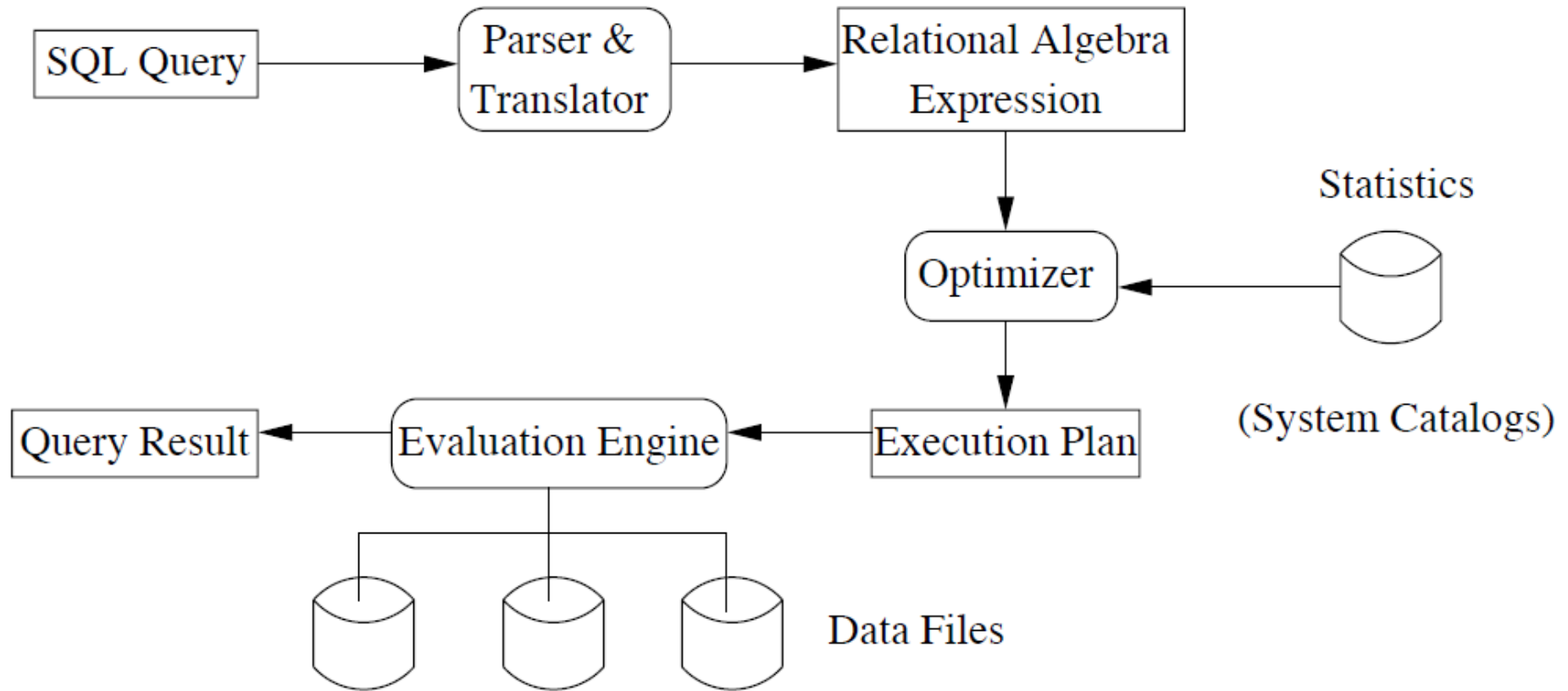
- ❑ Diferentes expresiones algebraicas por consulta:  
planes de consulta lógicos
- ❑ Representación: árboles de consulta
- ❑ Métodos ejecución operaciones algebraicas  
diferentes en estrategias básicas: Scanning,  
hashing, sorting, indexing, diferentes propuestas  
sobre memoria disponible



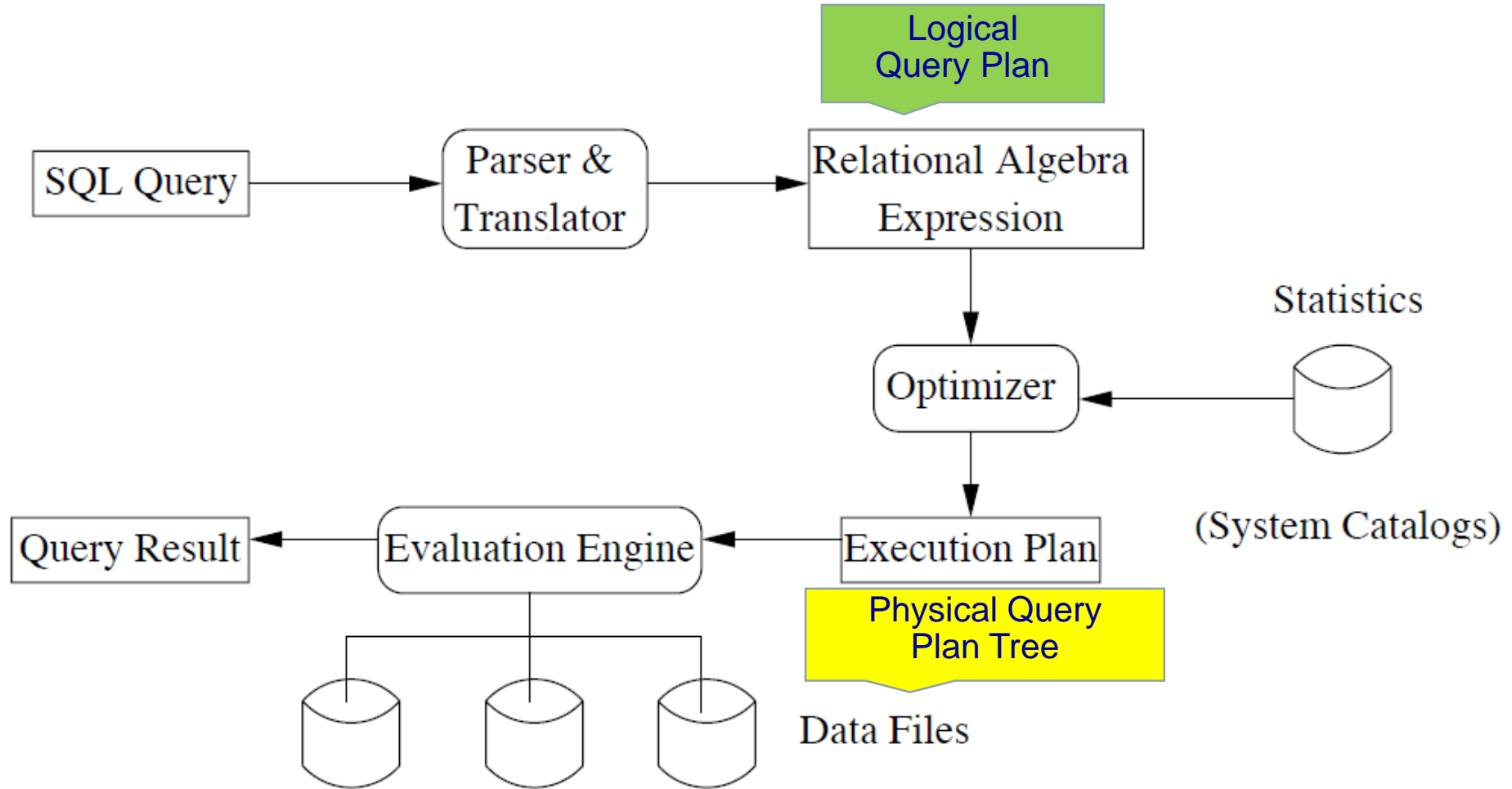
# Ejecución de Consultas: Etapas



# Ejecución de Consultas: Etapas



# Ejecución de Consultas: Etapas



# Compilación de Consultas

---

- ❑ Parsing: construye *parse tree* de la consulta
- ❑ Query rewrite: *Parse tree* se convierte en plan de consulta inicial (representación algebraica de consulta)
  - Plan inicial: Plan equivalente más barato
- ❑ Physical Plan Generation: Plan de consulta se convierte en plan de consulta físico
  - Selección algoritmos operador
  - Selección de orden de ejecución
  - Estrategias acceder relaciones



# Selección de mejor plan

---

- ❑ Formas algebraicas equivalentes de consulta producen algoritmos más eficientes?
- ❑ Para cada operación cuál algoritmo usar para implementarlo?
- ❑ Cómo pasar datos entre operaciones? (pipelined, main memory buffers, ..)





# Arboles de expresiones

---

MovieStar(name, addr, gender, birthdate)

StarIn(title, year, starname)

**SELECT** title, birthdate

**FROM** MovieStar, StarIn

**WHERE** year = 1996 **AND** gender = 'F' **AND**  
starname = name;



# Arboles de expresiones

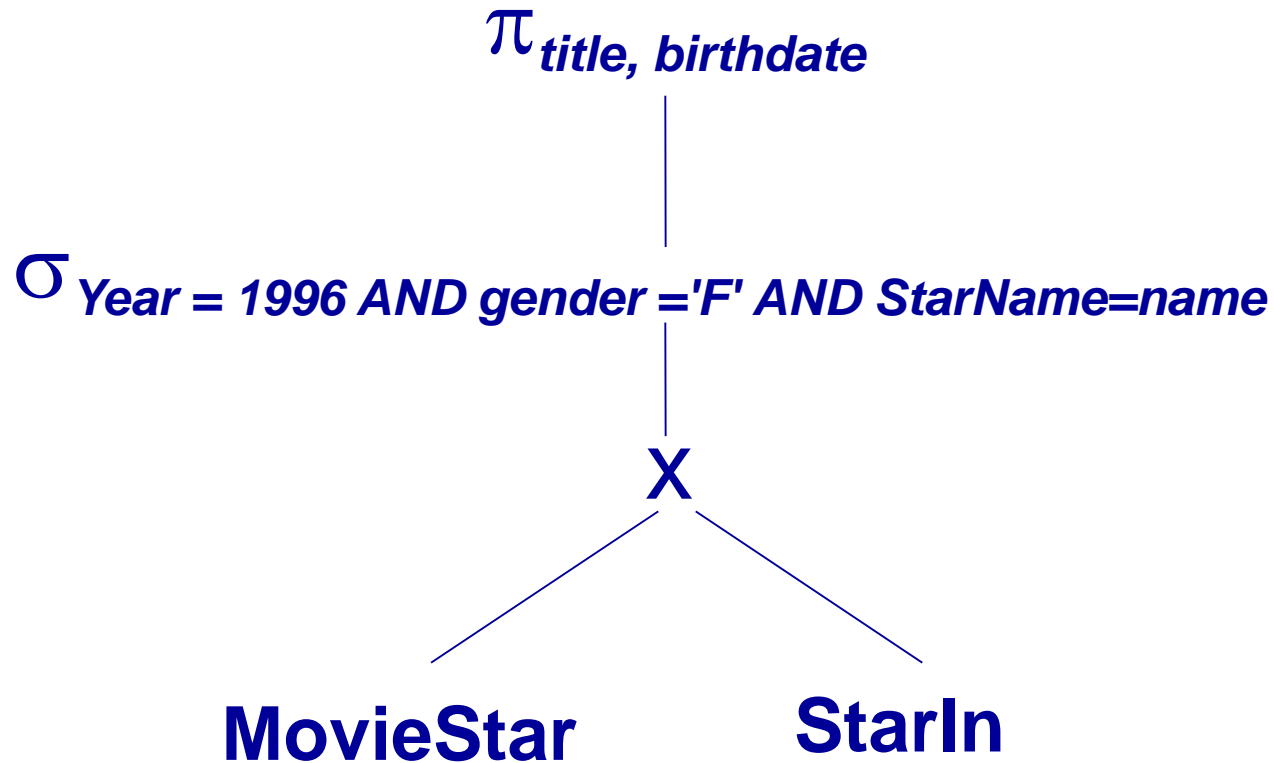
---

Parser: convierte consulta en plan lógico

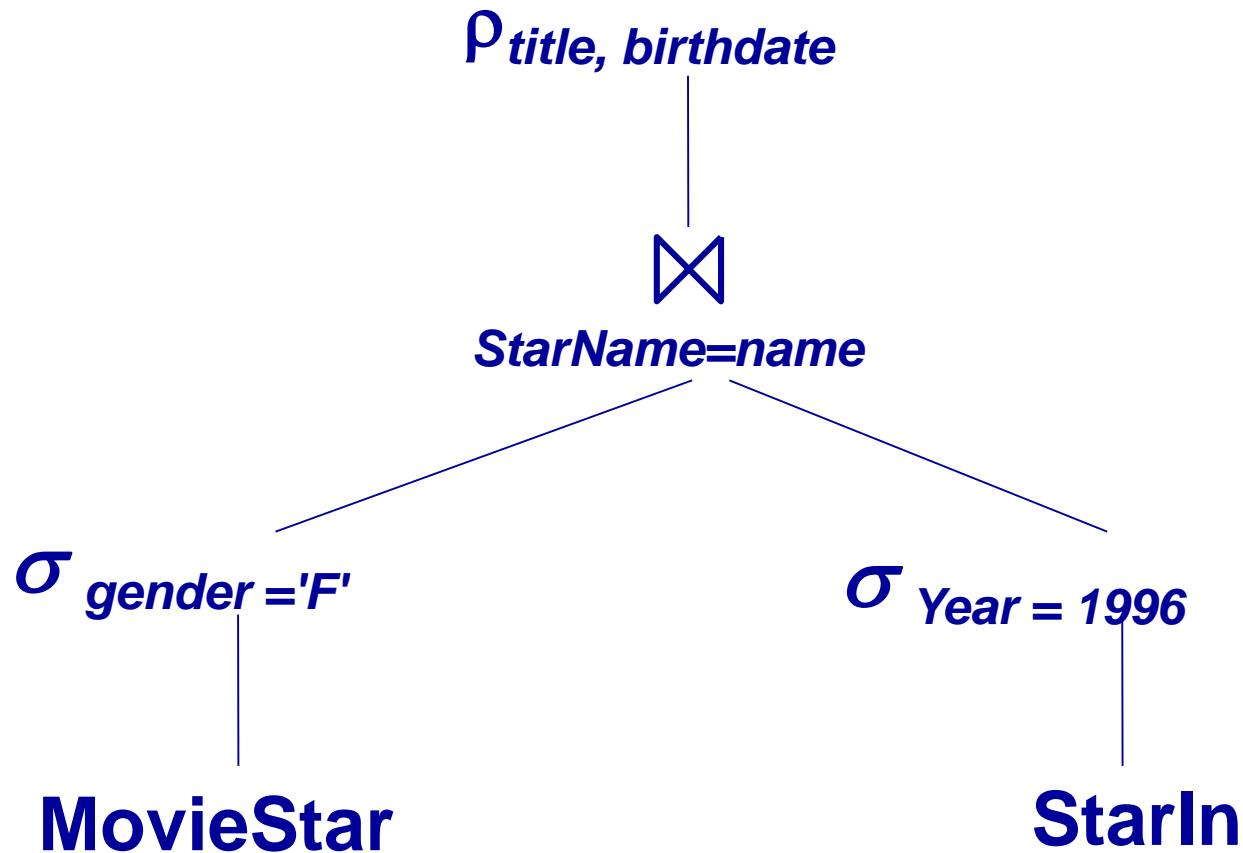
- Combinar relaciones de lista **FROM** (producto)
- Aplica selección en **WHERE**
- Proyecta usando lista cláusula **SELECT**



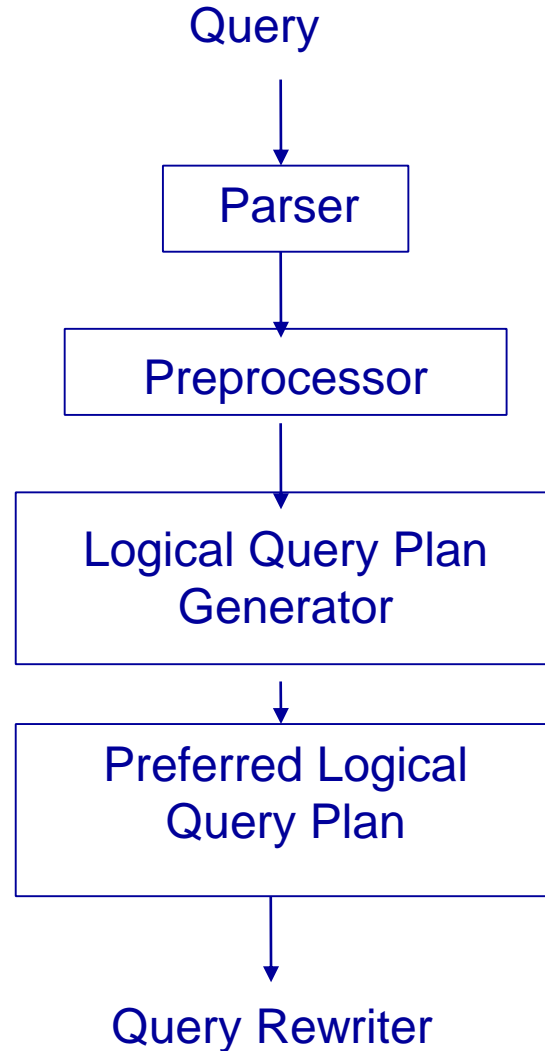
# Arboles de expresiones



# Arboles de expresiones



# Compilador de Consultas



# Compilador de Consultas

---

- ❑ Consulta: parse tree representando estructura de la consulta
- ❑ **Parse tree:** árbol expresión de plan lógico de consulta
- ❑ **Logical query plan**



# Parsing

---

## Análisis sintáctico y parse tree

**Tarea del parser:** convierte consulta SQ en *parse tree* con nodos:

- **Átomos:** elementos lexicográficos (keywords, nombres de atributo, constantes, operadores como + o <)
- **Categorías sintácticas:** Nombres de familias de subpartes de la consulta ⟨SFW⟩ y ⟨Condition⟩



# Parsing

---

- ❑ Si un nodo es un átomo entonces no tiene hijos
- ❑ Si el nodo es una categoría sintáctica, sus hijos se describen por medio de una de las reglas de la gramática del lenguaje.





# Categorías Sintácticas: Ejemplo

---

⟨ Query ⟩ : representa consulta bien formada SQL

Algunas de sus reglas son:

⟨ Query ⟩ ::= ⟨ SFW ⟩

⟨ Query ⟩ ::= (⟨ Query ⟩)

⟨ SFW ⟩ ::= **SELECT** ⟨ SelList ⟩

**FROM** ⟨ From List ⟩

**WHERE** ⟨ Condition ⟩



# Categorías Sintácticas: Ejemplo

---

## Select-List

$\langle \text{SelList} \rangle ::= \langle \text{Attribute} \rangle, \langle \text{SelList} \rangle$

$\langle \text{SelList} \rangle ::= \langle \text{Attribute} \rangle$

## From-List

$\langle \text{FromList} \rangle ::= \langle \text{Relation} \rangle, \langle \text{FromList} \rangle$

$\langle \text{FromList} \rangle ::= \langle \text{Relation} \rangle$



# Categorías Sintácticas: Ejemplo

---

## Conditions

$\langle \text{Condition} \rangle ::= \langle \text{Condition} \rangle \text{ AND } \langle \text{Condition} \rangle$

$\langle \text{Condition} \rangle ::= \langle \text{Tuple} \rangle \text{ IN } \langle \text{Query} \rangle$

$\langle \text{Condition} \rangle ::= \langle \text{Attribute} \rangle = \langle \text{Attribute} \rangle$

$\langle \text{Condition} \rangle ::= \langle \text{Attribute} \rangle \text{ LIKE } \langle \text{Pattern} \rangle$



# Consultas

Ejemplo: Stars in (Title, year, StarName)

MovieStar(name, address, gender, birthdate)

Películas con estrellas nacidas en 1960

Select title

From StarsIn

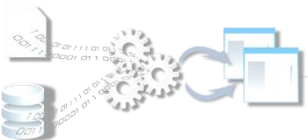
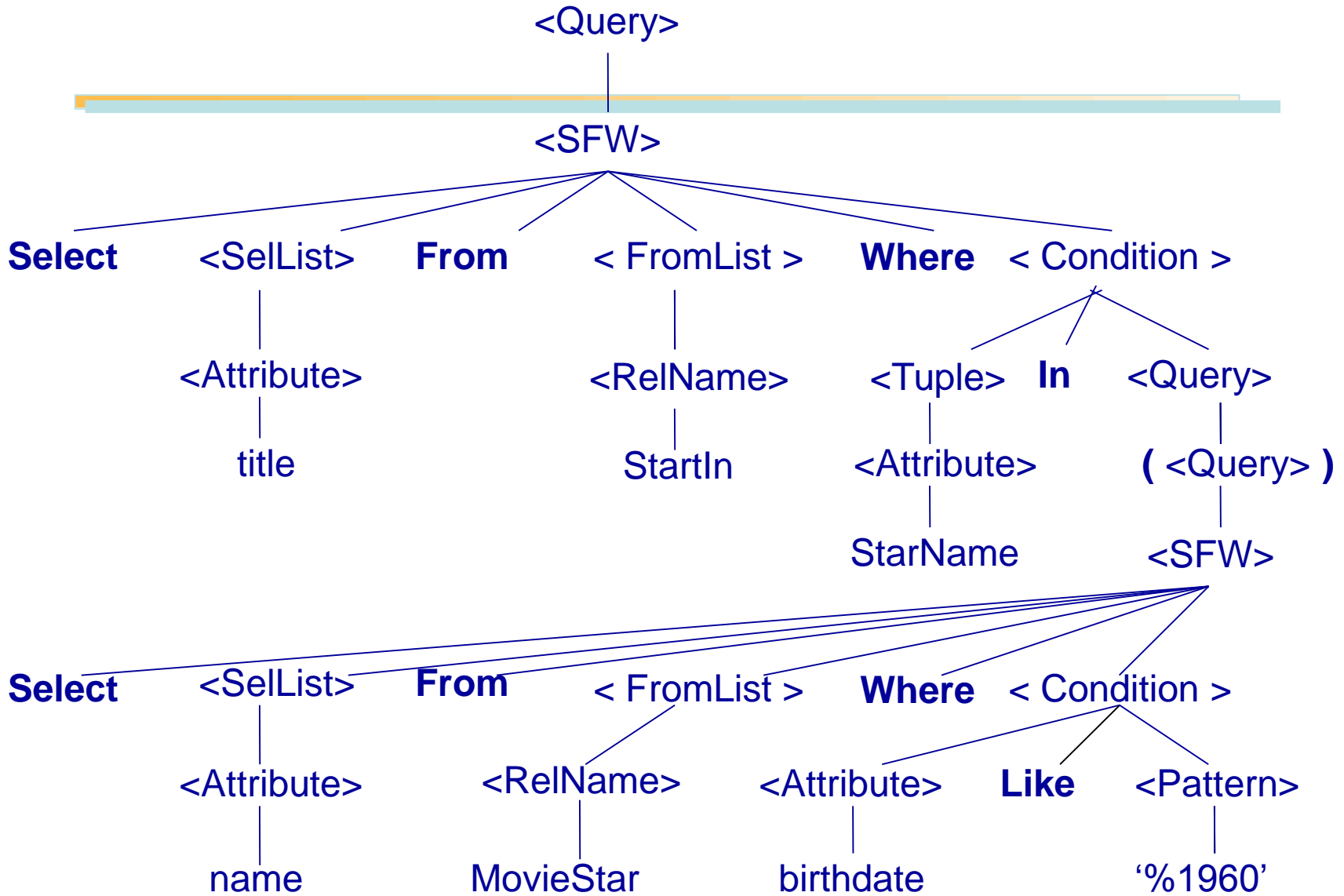
Where StarName IN (

Select name

From MovieStar

Where birthdate LIKE '%1960');





# Consultas

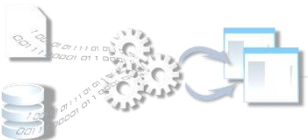
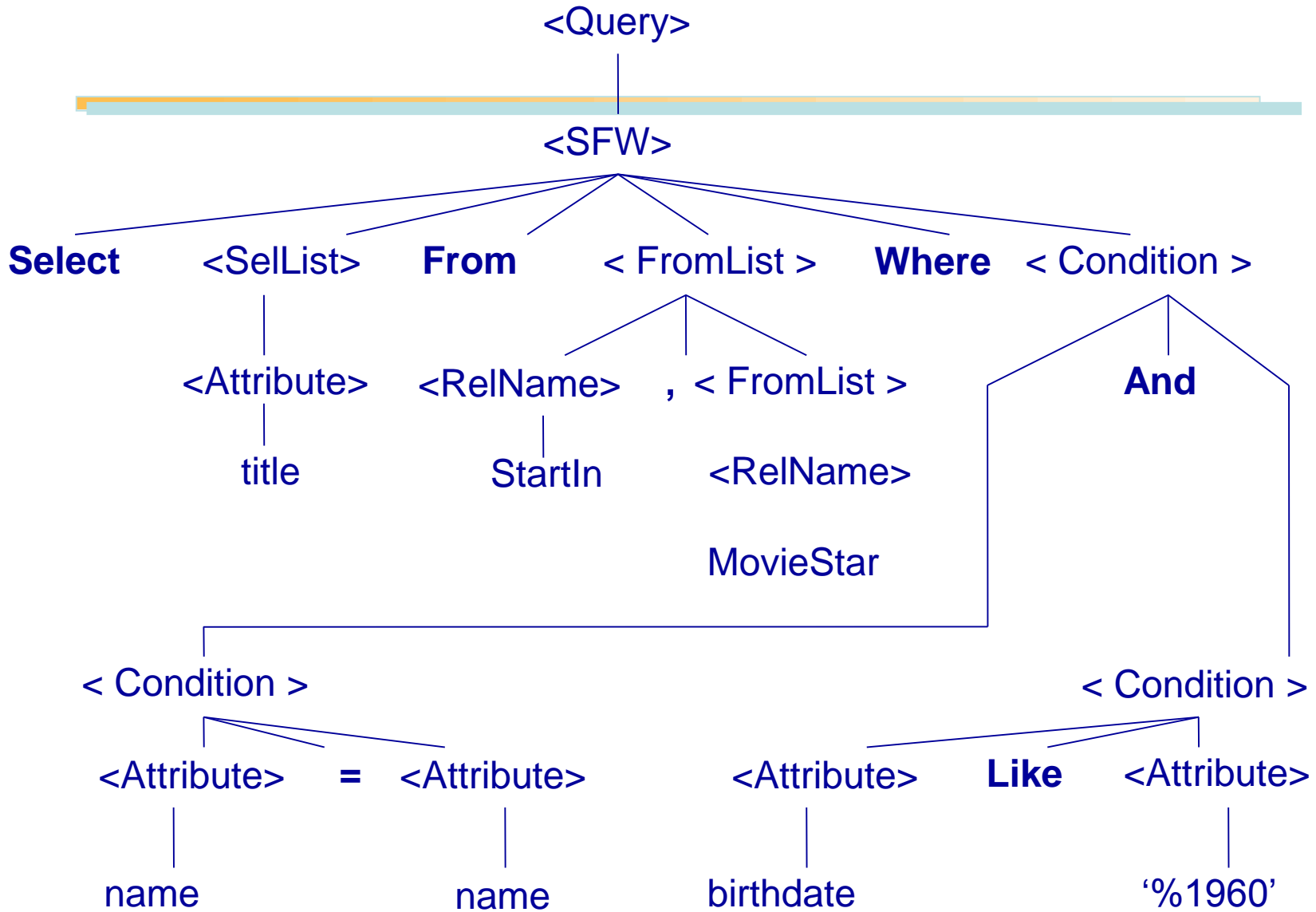
---

Select title

From StarsIn, MovieStar

Where StarName = name And birthdate  
LIKE '%1960';





# El Preprocesador

---

Si relación usada en query es una vista,  
cada uso de una relación en el from-list se  
debe reemplazar por un parse tree que  
describe la vista





# El Preprocesador: Chequeo semántico

---

- Check relation uses

Cada relación en From-Clause debe ser una relación o una vista del esquema de la consulta

- Check and resolve attribute uses

Cada atributo en cláusula SELECT o WHERE debe ser un atributo de alguna relación (si no lo está explícitamente en la consulta)

- Check types

