# TEMA 4.6: Procesamiento y optimización de consultas

1. Introducción
2. Procesamiento de una consulta
   - 2.1.- Análisis de la consulta
   - 2.2.- Reescritura de la consulta
   - 2.3.- Generación de un plan físico de ejecución. Optimizador.
3. Procesamiento de consultas en ORACLE
   - 3.1.- Herramientas: EXPLAIN PLAN



## 1.- Introducción

- Objetivo: disminuir el **tiempo de ejecución** de las consultas que se realizan más frecuentemente sobre una base de datos
- ¿Cuál es el camino de acceso a los datos?
  - Modificar el diseño físico
    - Añadir redundancia y modificar la organización: añadir o cambiar índices, dividir tablas, particionar tablas.
    - Reorganizar las estructuras para mantener las característcas a pesar de borrados o actualizaciones

## 2.- Procesamiento de consultas

- Análisis de la consulta
  - representación arbórea de la estructura de la consulta
- Reescritura de la consulta
  - representación en álgebra relacional
  - transformación a un plan lógico más eficiente
- Generación de un plan físico de ejecución
  - Selección de los algoritmos para la ejecución de cada operación lógica



## 2.1.- Análisis de la consulta

- Generación del árbol de la consulta
  - Comprobar la sintáctica de SQL
- Preprocesado: Comprobaciones semánticas
  - Relaciones: las tablas existen en la BD
  - Atributos: están definidos para las tablas
  - Tipos: los tipos de los atributos usados en las condiciones son compatibles

## 2.1.- Análisis de la consulta
## Ejemplo de consulta

ACTOR (nombre,direccion,genero,fecha_nacimiento)

ACTUA (pelicula, año_estreno, actor)

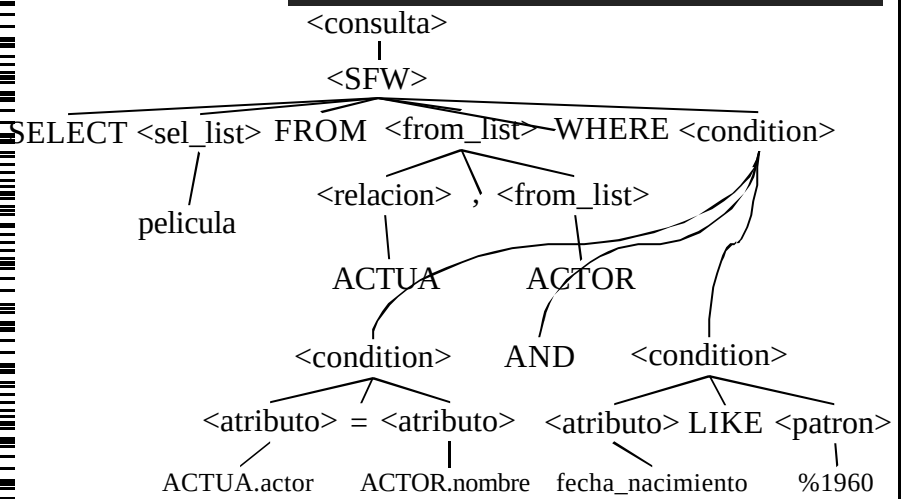
- Obtener los títulos de las peliculas en las que participan actores nacidos en 1960

```
SELECT pelicula
FROM ACTUA
WHERE actor IN (
 SELECT nombre
 FROM actor
 WHERE
 fecha_nacimiento LIKE '%1960')
```

```
SELECT pelicula
FROM ACTUA, ACTOR
WHERE actor = nombre AND
 fecha_nacimiento LIKE '%1960'
```



## 2.1.- Análisis de la consulta
## Ejemplo de análisis de consulta

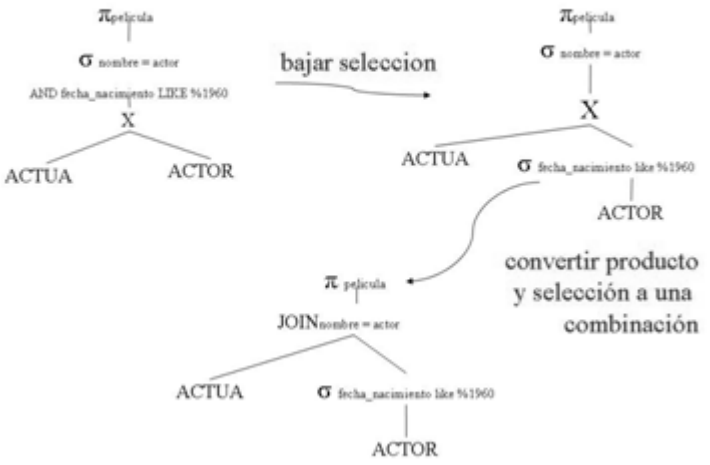


## 2.2.- Reescritura de la consulta
## Plan lógico de la consulta

- Transformación del árbol a álgebra relacional
- Reglas de optimización algebraicas
  - Algunas reglas generales
    - Conmutatividad / Asociatividad
    - Bajar las selecciones (reduce el número de tuplas)
    - Bajar las proyecciones (reduce el tamaño)
    - Eliminar subconsultas de las condiciones



## 2.2.- Reescritura de la consulta
## Plan lógico de la consulta



## 2.3.- Generación de un plan físico de ejecución

- Transformar cada operación lógica a un algoritmo de recuperación concreto
- Los algoritmos pueden ser de tres tipos:
  - **Memoria** (si las tablas pueden almacenarse completamente en memoria)
  - **De una pasada** (cuando las tablas no se pueden almacenar en memoria pero si el resultado)
  - **De dos o más pasadas** (cuando ni las tablas ni los resultados se pueden mantener en memoria)



## 2.3.- Generación de un plan físico de ejecución Algoritmos físicos

- Según el tipo de la estructura (auxiliar) utilizada
  - Recorrido total (**scan**)
  - Basada en índices (**index**): ej. join natural
  - Basada en una tabla de dispersión (**hash**)
  - Basado en ordenación (**sort**): algunas operaciones se simplifican si los resultados están ordenados: ej. agrupación
  - ¿Es posible realizar en paralelo?

## 2.3.- Generación de un plan físico de ejecución Tipos de optimización

- Optimización basada en reglas
  - Heurísticas generales basadas en la experiencia del administrador o de los diseñadores
- Optimización basada en costes
  - Estimación de los costes de realizar una operación física. Depende:
    - Tamaño de la estructura: estadísticas
    - Tamaño de la memoria: depende de los procesos que se ejecuten simultáneamente



## 3.- Procesamiento de consultas en ORACLE.

- Almacenamiento de consultas y su plan de ejecución en memoria
- Antes de analizar una consulta comprueba que está en memoria:
  - Comprueba que una consulta es la misma carácter por carácter
  - ¿Qué ocurre con las variables?
    - películas con actores nacidos en 1960
    - películas con actores nacidos en 1977

## 3.1.- Optimización de consultas en ORACLE: Herramientas

- Optimización basada en reglas:
  - SQL*Expert: Sistema experto para el diseño físico y la optimización
- Optimización basada en costes
  - SQL-Analyze (EXPLAIN PLAN)
    - Planes de ejecución lógicos/físicos y depuración
  - Recolección de estadísticas:
    - Paquete SQL_STAT : Es necesario contar con estadísticas actualizadas



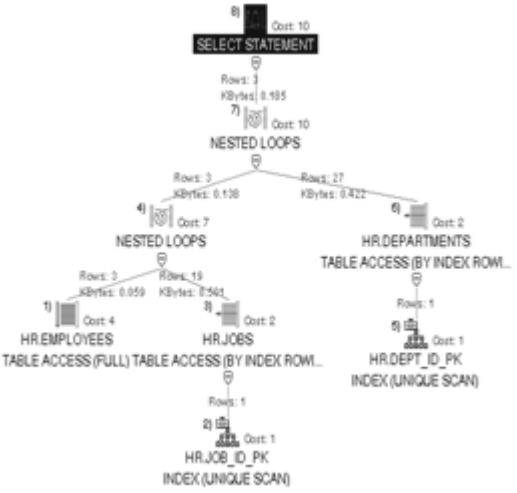
## 3.1.- Herramientas ORACLE: EXPLAIN PLAN

```
EXPLAIN PLAN FOR
SELECT e.employee_id, j.job_title, e.salary, d.department_name
FROM employees e, jobs j, departments d
WHERE  e.employee_id < 103
AND e.job_id = j.job_id
AND e.department_id = d.department_id;
```

```
-------------------------------------------------------------------------------
| Id  | Operation                    | Name        | Rows  | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |             |     3 |   189 |    10  (10)|
|   1 |  NESTED LOOPS                |             |     3 |   189 |    10  (10)|
|   2 |   NESTED LOOPS               |             |     3 |   141 |     7  (15)|
|*  3 |    TABLE ACCESS FULL         | EMPLOYEES   |     3 |    60 |     4  (25)|
|   4 |    TABLE ACCESS BY INDEX ROWID| JOBS       |    19 |   513 |     2  (50)|
|*  5 |     INDEX UNIQUE SCAN        | JOB_ID_PK   |     1 |       |            |
|   6 |   TABLE ACCESS BY INDEX ROWID | DEPARTMENTS |    27 |   432 |     2  (50)|
|*  7 |    INDEX UNIQUE SCAN         | DEPT_ID_PK  |     1 |       |            |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
   3 - filter("E"."EMPLOYEE_ID"<103)
   5 - access("E"."JOB_ID"="J"."JOB_ID")
   7 - access("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
```

## 3.1.- Herramientas ORACLE: EXPLAIN PLAN




1) Recorrido serial de toda la tabla employees
2) Recorrido del índice primario de la tabla jobs
3) Aceeso mediante ROWID a las filas obtenidas en 2)
4) Bucle anidado (por bloques) de los resultados 2) y 3)
5) Recorrido del índice primario de la tabla departament
6) Aceeso mediante ROWID a las filas obtenidas en 5)
7) Bucle anidado (por bloques) de los resultados 4) y 6)
8) Sentencia SELECT que devuelve los resultados al usuario



## 3.2.- Pistas de ejecución (Hints) en ORACLE

- El administrador puede tener una mejor percepción del problema: puede dar pistas para mejorar el plan de ejecución física
- SELECT /*+<lista de hints> */ .....
  - +INDEX +NO_INDEX
  - +ALL_ROWS +FIRST_ROWS
  - +USE_MERGE(merge_sort)
  - +USE_NL (nested loop)