



INSTITUTO FEDERAL  
FARROUPILHA

# Admin de Banco de Dados

## Processamento e otimização de consultas

Prof.: Ícaro Lins Iglesias

[icaro.iglesias@iffarroupilha.edu.br](mailto:icaro.iglesias@iffarroupilha.edu.br)

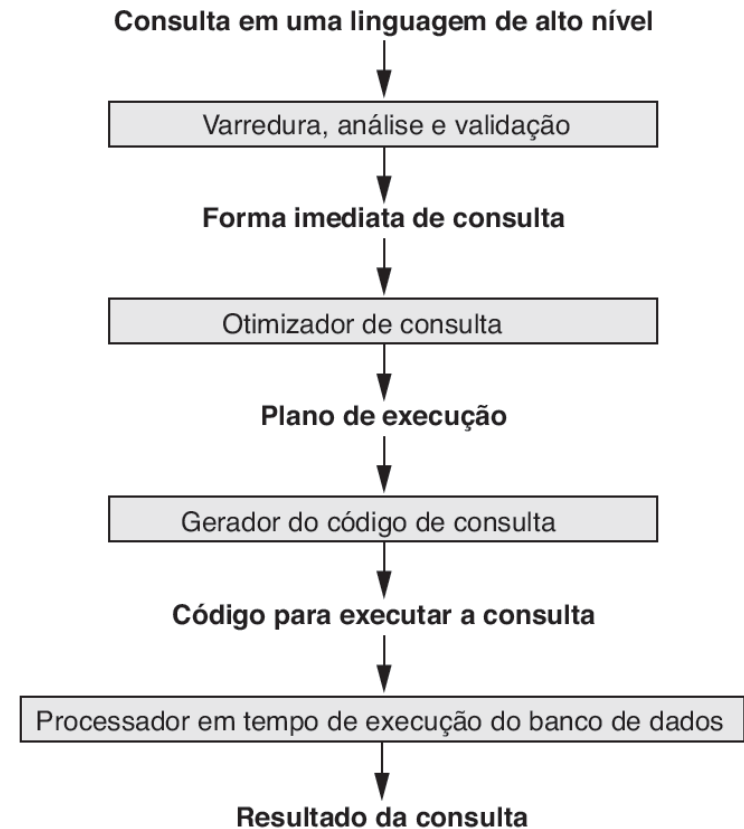
# Introdução ao processamento de consultas

- Técnicas usadas internamente por um SGBD para processar, otimizar e executar consultas de alto nível
- Uma consulta expressa em uma linguagem de consulta de alto nível, como SQL, primeiro precisa ser lida, analisada e validada.

Existem diferentes etapas no processamento de uma consulta de alto nível.

O módulo otimizador de consulta tem a tarefa de produzir um bom plano de execução, e o gerador de código dá origem ao código para executar esse plano.

# Etapas no processamento de uma consulta de alto nível



## Código pode ser:

Executado diretamente (modo interpretado)

Armazenado e executado mais tarde,  
sempre que possível (modo compilado)

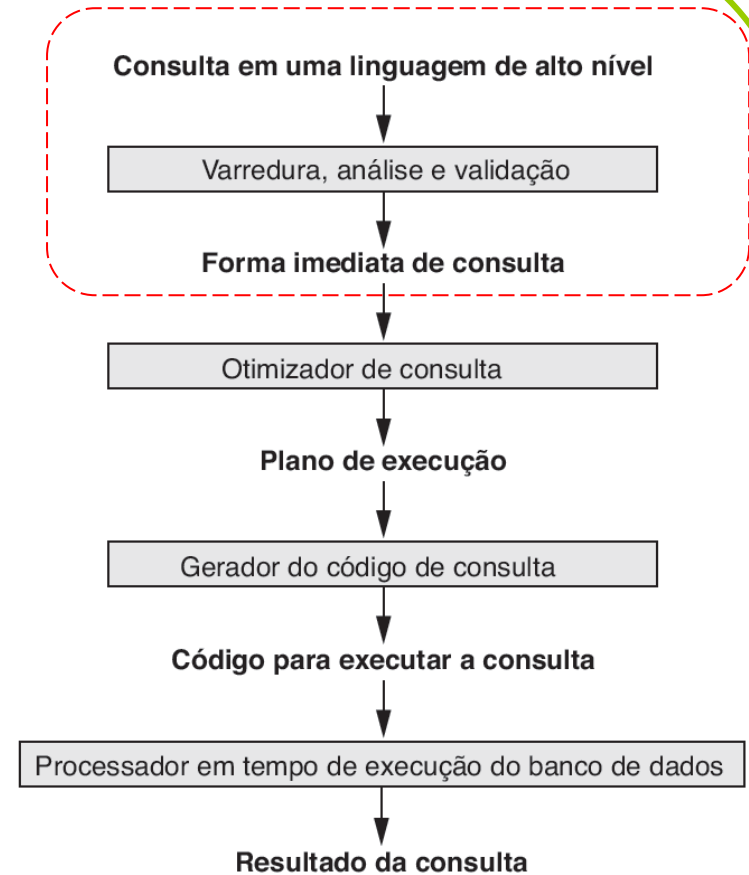
# Varredura, Análise e Validação

## Análise e Validação

- Análise léxica (scanner -> *tokens, alfabeto*)
- Análise sintática (*parser*)
- Validação (semântica)

## Representações internas

- Árvore de Consulta
- Grafo de consulta



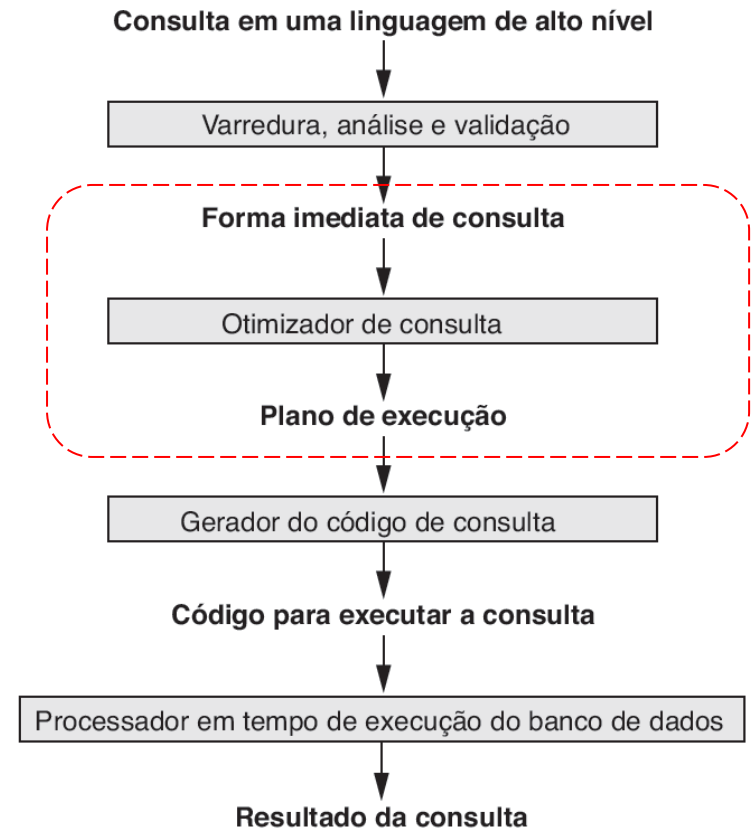
### Código pode ser:

Executado diretamente (modo interpretado)

Armazenado e executado mais tarde, sempre que possível (modo compilado)

# Estratégia de execução

- Existem muitas estratégias possíveis
- Planejamento da estratégia : otimização
- Escolha da situação mais adequada razoavelmente:
  - Existe Índice?
  - Qual a melhor forma de organização para acesso eficiente da memória?



## Código pode ser:

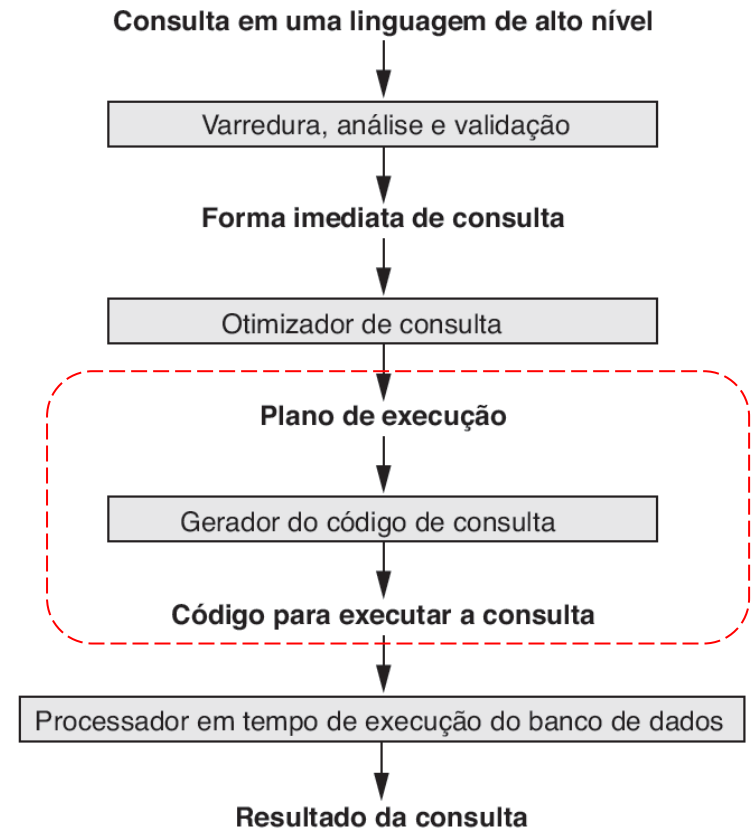
Executado diretamente (modo interpretado)

Armazenado e executado mais tarde,  
sempre que possível (modo compilado)

# Código da Consulta

## Modos:

1. Executado diretamente (interpretado) – gera o código interno e dispara o evento para o processador
2. Armazenado e executado quando necessário (compilado) – armazena o evento para posterior envio ao processador - *view*



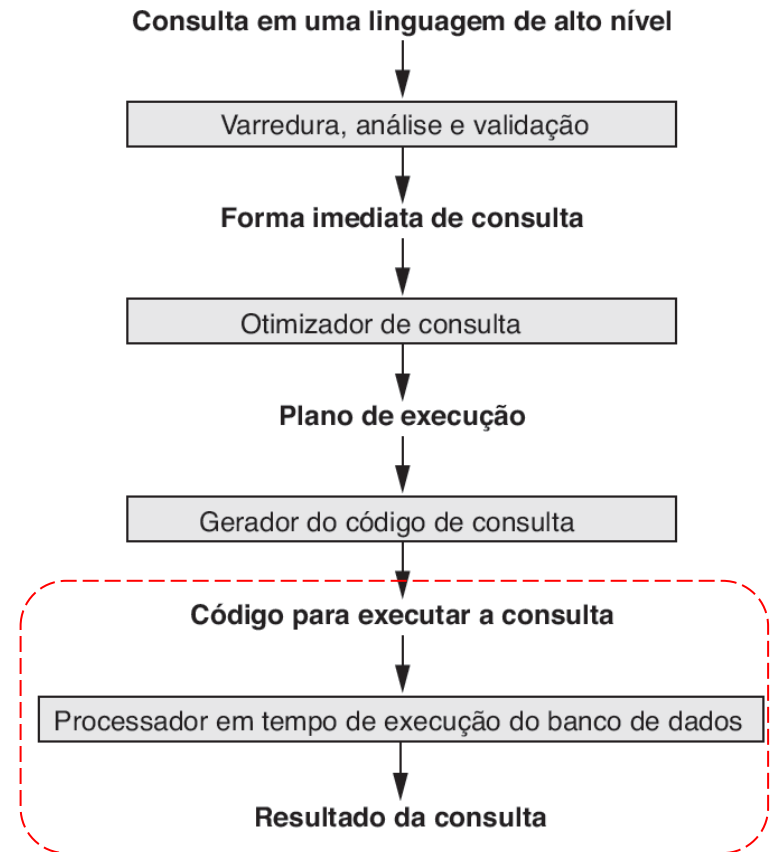
### Código pode ser:

Executado diretamente (modo interpretado)

Armazenado e executado mais tarde,  
sempre que possível (modo compilado)

# Execução do Código

- Processador executa o código da consulta
- É produzido um resultado desta consulta



## Código pode ser:

Executado diretamente (modo interpretado)  
Armazenado e executado mais tarde,  
sempre que possível (modo compilado)

# Otimização das consultas

- Consulta: extração de informações do BD
- Consultas em linguagem de alto nível
  - ❑ Não descrevem uma estratégia eficiente para a implementação de cada passo -> Acesso ao nível físico (arquivos do BD)
- O SGBD deve se preocupar com a estratégia
  - ❑ Módulo de Processamento de Consultas



# Otimização das consultas

- Consultas em SQL, são traduzidas em consultas da álgebra relacional e então otimizadas.
- Otimização é o processo de escolha de uma estratégia de execução para o processamento da consulta
  - Tradução, transformação e geração de uma estratégia (plano) de execução e acesso
  - Considera algoritmos predefinidos para implementação de passos do processamento e estimativas

# Otimização das consultas

- Encontrar a estratégia ótima requer:
  - ☐ Muito tempo
  - ☐ Pode necessitar informações de como os arquivos estão implementados, do conteúdo dos arquivos e as informações podem não estar disponíveis
  - ☐ Os SGBDs têm diferentes algoritmos de acesso ao BD para implementar as operações
- *Em geral, a escolha do plano de execução não é pela melhor estratégia, mas pela estratégia de eficiência razoável.*

# Traduzindo SQL para Alg. Relacional

- Uma consulta SQL é primeiro traduzido para uma expressão da AR estendida
- A representação é dada como uma estrutura de dados de árvore de consulta
- Um bloco de consulta contém uma única expressão SELECT-FROM-WHERE, bem como GROUP BY e HAVING quando aplicado.
- As consultas aninhadas são identificadas como blocos de consultas individuais.
- Operações de agregação, como MAX, MIN, SUM, AVG e COUNT, são incluídos no conceito de AR estendida.
- Pode-se definir uma função agregada usando o símbolo  $\mathfrak{F}$  (pronuncia-se *F script*)

# Traduzindo SQL para Alg. Relacional

## Exemplo:

- Considere a consulta SQL para a relação Funcionário, onde são listados os funcionários (de qualquer departamento) que recebem salário maior que o maior salário do departamento 5:

**SELECT** Pnome, Unome

**FROM** FUNCIONÁRIO

**WHERE** Salario > (**SELECT** MAX(Salario) **FROM**  
FUNCIONARIO **WHERE** Dnr=5);

# Traduzindo SQL para Alg. Relacional

- Esta consulta inclui uma subconsulta aninhada e precisa ser decomposta em dois blocos:
- **Mais interno:** recupera o salário mais alto do departamento 5

**(SELECT MAX(Salario) FROM FUNCIONARIO  
WHERE Dnr=5);**

- **Mais externo:** recupera o salário mais alto do departamento 5

**SELECT Pnome, Unome  
FROM FUNCIONÁRIO  
WHERE Salario > c**

Onde  $c$  representa o resultado do bloco interno

# Traduzindo SQL para Alg. Relacional

- Em álgebra relacional estendida, o bloco interno poderia ser traduzido para a expressão:

$$c \leftarrow \mathcal{J} \text{ MAX Salario } (\sigma_{\text{Dnr}=5}(\text{FUNCIONARIO}))$$

- E o externo para:

$$\pi_{\text{Pnome, Unome}} (\sigma_{\text{Salario} > c} (\text{FUNCIONARIO}))$$

- O bloco interno só precisou ser avaliado uma vez para encontrar o maior salário do departamento 5.
- Desta forma, o otimizador de consulta escolheria um plano de execução adequado para cada bloco de consulta.

# Notação para árvore de consulta

## Geração da Árvore Algébrica (canônica)

- Estrutura que representa o mapeamento da consulta em álgebra relacional
  - Nós folha: relações do BD
  - Nós internos: operações da álgebra
- É equivalente à Consulta
- Não tem preocupação com otimização

## Processamento da árvore

- Nós internos:
  - Executados quando seus operandos estão disponíveis
- Nó raiz: último a ser executado

# Notação para árvore de consulta

## Definição de uma árvore de consulta equivalente

- Entrada: árvore da consulta inicial (canônica)
- Saída: árvore da consulta otimizada

## Árvore de consulta final que seja eficiente para executar

- Uma consulta pode corresponder a várias expressões em álgebra relacional
- Cada expressão corresponde a uma árvore de consulta diferente



# Notação para árvore de consulta

Considere a consulta: *Para cada projeto localizado em Mauá, liste o número do projeto, seu departamento controlador, bem como o último nome, endereço e data de nascimento do gerente deste departamento*

## ✓ Expressão em AR:

$\pi$  Projnumero, Dnum, Unome, Endereco, Datanasc( $((\sigma$  Projlocal='Mauá' (PROJETO))  $\bowtie$  Dnum=Dnumero (DEPARTAMENTO))  $\bowtie$  Cpf\_gerente=Cpf (FUNCIONARIO))

## ✓ Em SQL:

**SELECT** Projnumero, P.Dnum, F.Unome, F.Endereco, F.data\_nasc  
**FROM** PROJETO **AS** P, DEPARTAMENTO **AS** D, FUNCIONARIO **AS** F  
**WHERE** P.Dnum=D.Dnumero **AND** D.Cpf\_ger=F.Cpf **AND** P.Projlocal =  
"Mauá";

# Árvore de consulta inicial (Canônica) - C2

$\pi$  Projnumero, Dnum, Unome, Endereco, Datanasc

$\sigma$  Projlocal='Mauá'  $\wedge$  Dnum=Dnumero  $\wedge$  Cpf\_gerente=Cpf

3

Nós internos  
(operações da AR)

Nós folha  
(relações do  
BD)

FUNCIONARIO

PROJETO

DEPARTAMENTO

# Árvore Otimizada

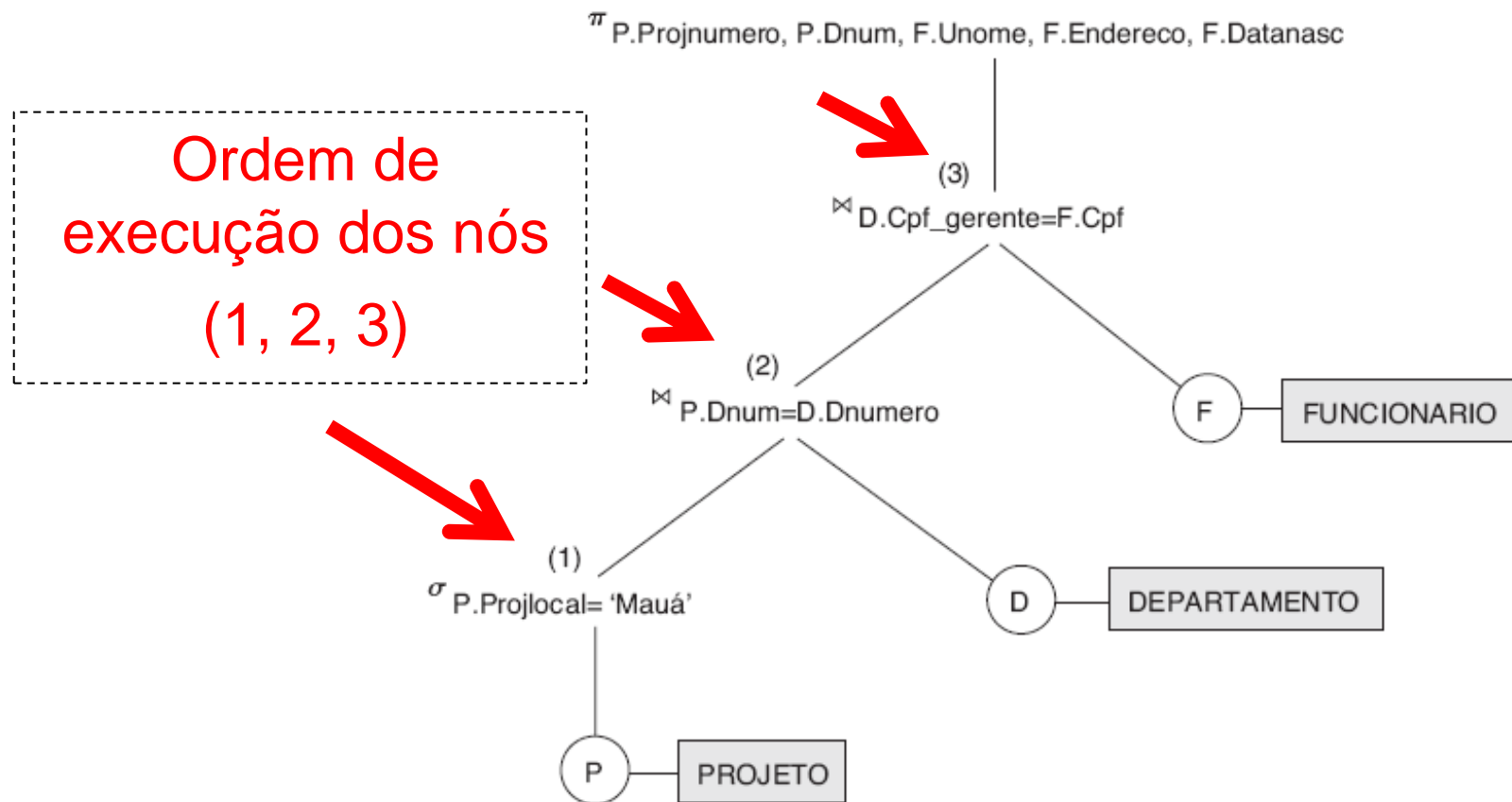


Figura 6.9

Árvore de consulta correspondente à expressão da álgebra relacional para C2.

# Otimização heurística das árvores de consulta

- Em geral, há muitas expressões diferentes da AR, mas possuem como resultado árvores de consultas equivalentes (representam a mesma consulta).
- Analisador de consulta -> cria uma árvore de consulta inicial, correspondente a consulta SQL, sem otimização.
- O otimizador de consulta heurística é responsável por transformar esta árvore de consulta inicial em uma **árvore de consulta final** equivalente, eficiente para execução.
- Deve incluir regras para *equivalência entre expressões da AR* capazes de transformar a árvore inicial em uma árvore final otimizada.

# Exemplo de transformação de árvores

Considere a consulta: *Encontre os sobrenomes dos funcionários nascidos após 1957 que trabalham em um projeto chamado “Aquarius”.*

✓ **Expressão correspondente em SQL:**

**SELECT** Unome

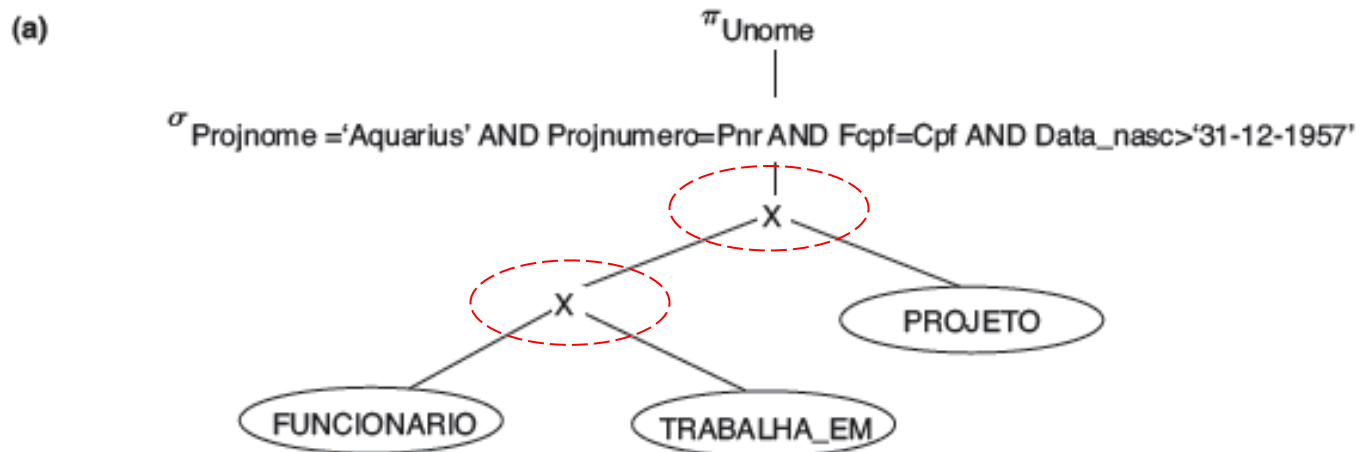
**FROM** FUNCIONARIO, TRABALHA\_EM, PROJETO

**WHERE** Projnome=“Aquarius” **AND** Projnumero=Pnr

**AND** Fcpf=Cpf **AND** Data\_nasc > “31-12-1957”;

# Exemplo de transformação de árvores (A)

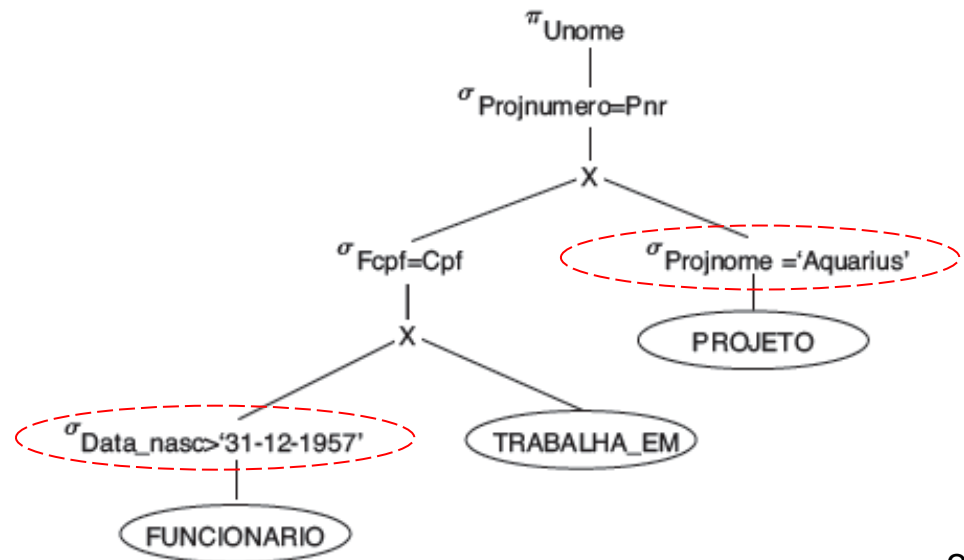
- A figura (a) ilustra a árvore inicial (canônica) para a consulta SQL.
- É criado um arquivo muito grande contendo o PRODUTO CARTESIANO dos arquivos FUNCIONARIO, TRABALHA\_EM e PROJETO inteiros.
- Esta árvore nunca será executada desta forma, e sim transformada em outra árvore equivalente, eficiente para execução.



## Exemplo de transformação de árvores (B)

- A consulta precisa apenas de um registro da relação PROJETO (“Aquarius”) e de registros de FUNCIONÁRIOS que nasceram após “31/12/1957”.
- A figura (b) apresenta uma árvore já melhorada, aplicando primeiramente as operações de SELEÇÃO, de modo a reduzir o número de tuplas do PRODUTO CARTESIANO.

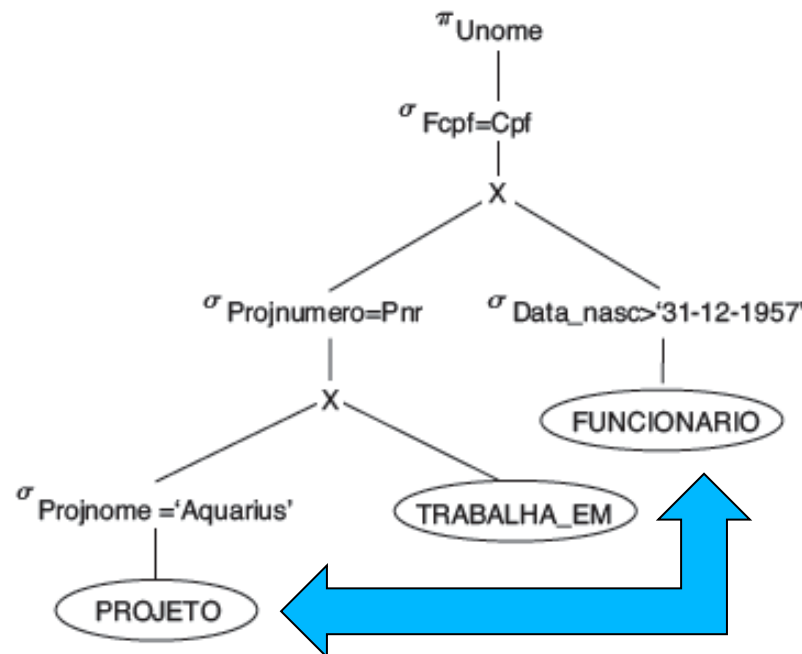
(b)



## Exemplo de transformação de árvores (C)

- Outra melhoria é alcançada trocando as posições das relações FUNCIONÁRIO e PROJETO, ilustrado em (c).
- Desta forma, usa-se a informação de que *Projnumero* é um atributo chave de PROJETO, fazendo com que a SELEÇÃO nesta relação retorne um único registro.

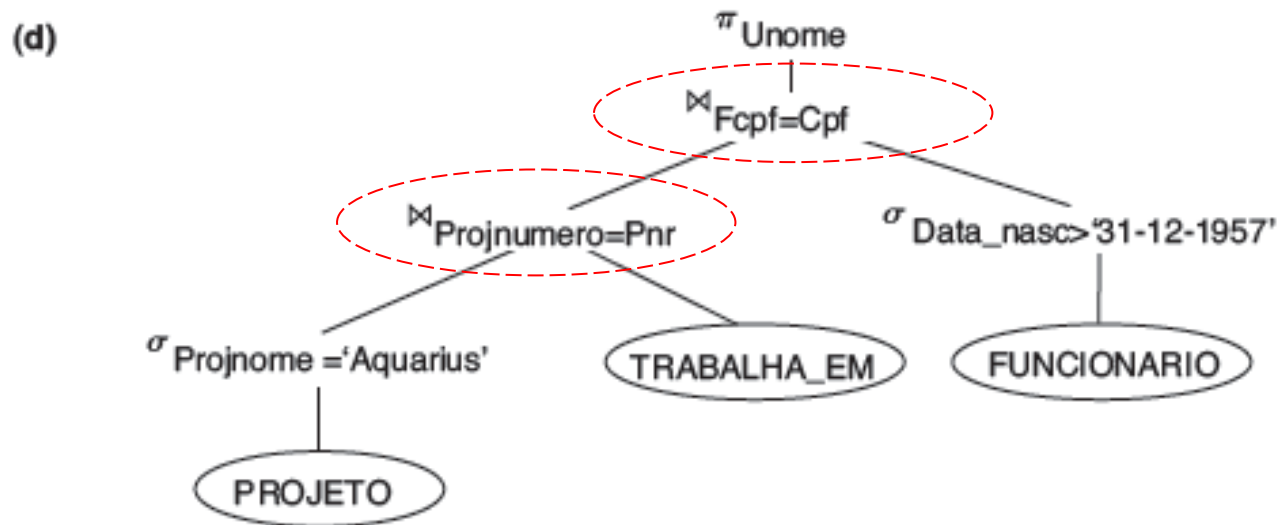
(c)





## Exemplo de transformação de árvores (D)

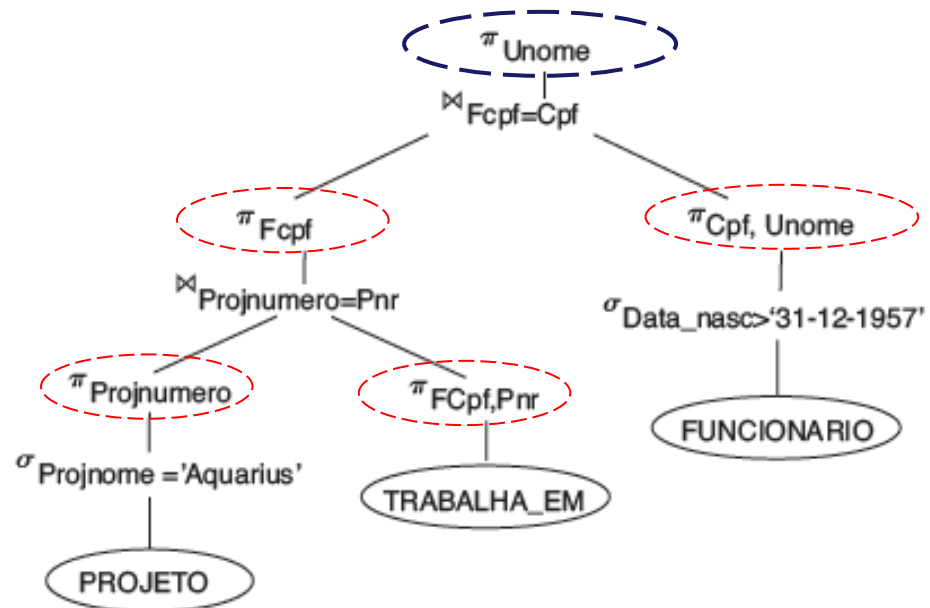
- Para melhorar a árvore, substituem-se as operações de PRODUTO CARTESIANO acompanhadas de condições de igualdade por operações de JUNÇÃO.



## Exemplo de transformação de árvores (E)

- Por fim, mantém-se apenas os atributos necessários pelas operações subsequentes nas relações intermediárias, incluindo as operações de PROJEÇÃO o mais cedo possível na árvore de consulta.
- Esta tarefa reduz as colunas intermediárias enquanto que as operações de SELEÇÃO reduzem o número de tuplas (registros).

(e)



# Atividade de Aprendizagem

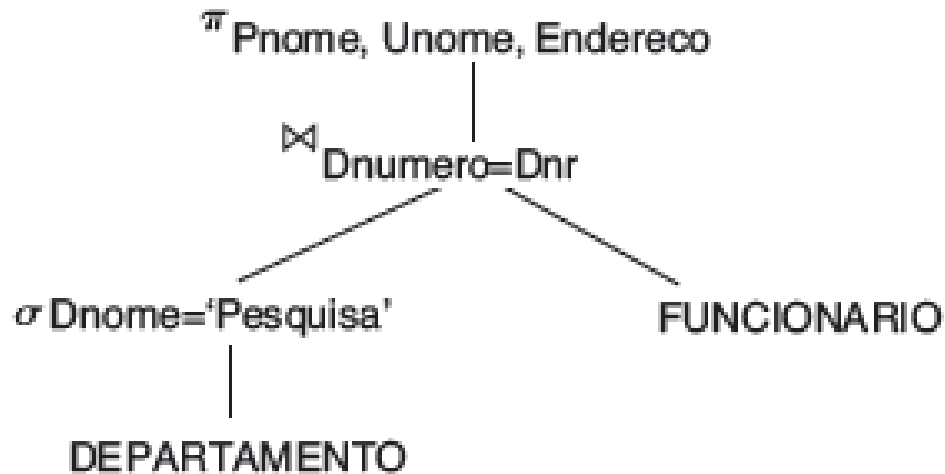
## 1) Considere a seguinte consulta:

*Liste o nome e grau de parentesco dos dependentes, bem como o primeiro e último nome dos funcionários do sexo Masculino que atuam no departamento de Pesquisa.*

- a) Apresente o comando SQL
- b) A expressão correspondente em AR
- c) A árvore canônica (inicial)
- d) A árvore otimizada final (conforme exemplificado)

# Atividade de Aprendizagem

2) Considere a árvore de consulta otimizada que segue:



**A partir desta árvore, crie:**

- a) A expressão em AR correspondente;
- b) Sua consulta SQL equivalente;
- c) A árvore Canônica inicial.