

# UCS

## UNIVERSIDADE DE CAXIAS DO SUL

UNIVERSIDADE DE CAXIAS DO SUL

Disciplina: Fundamentos de Banco de Dados  
Professor: Daniel Luis Notari

Acadêmico: Leandro de Matos Soares, Pedro Henrique Bambi de Godoy, Rodrigo Luan  
Ferraza

**Explicação Plano de Execução**  
**Álgebra Relacional**  
**Banco de Dados**

# Introdução

No projeto apresentado utiliza-se diversas consultas, comentaremos sobre o plano de execução de algumas consultas escolhidas e as suas funcionalidades.

Um dos maiores problemas na extração de informações de um banco de dados é a velocidade no qual a disponibilização das informações ocorreram, para isso observa-se o plano de execução.

Hoje o plano de execução de uma consulta em um SGBD é de suma importância, o plano de execução auxilia na performance, qualidade, tempo de execução e eficiência de um projeto.

Assim, um plano de execução envolve a compreensão de por quê certo operador foi usado e não outro operador, envolve a compreensão do melhor tempo de resposta da consulta e quais fatores influenciaram o SQL a tomar decisões que tomou para gerar o plano.

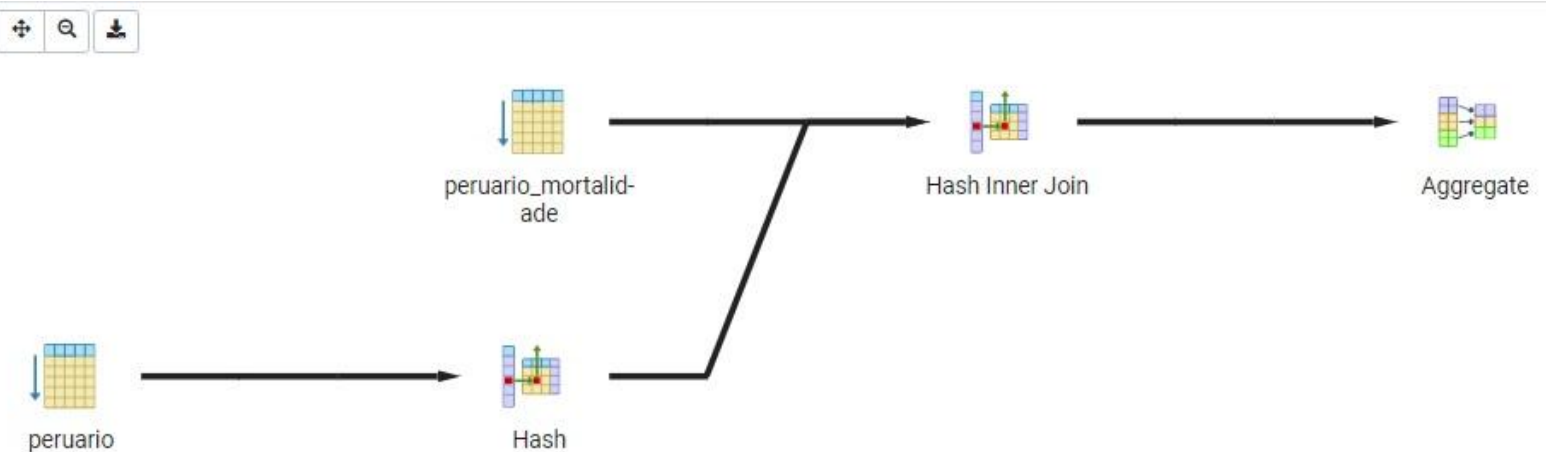
Dessa forma, segue a análise do plano de execução:

Plano de Execução da consulta:

-- select exists

-- selecionar os peruanos que houveram mortes

```
SELECT DISTINCT *  
FROM peruano_mortalidade AS a  
WHERE EXISTS (  
    SELECT *  
    FROM peruano AS b  
    WHERE b.id = a.peruano_id  
);
```

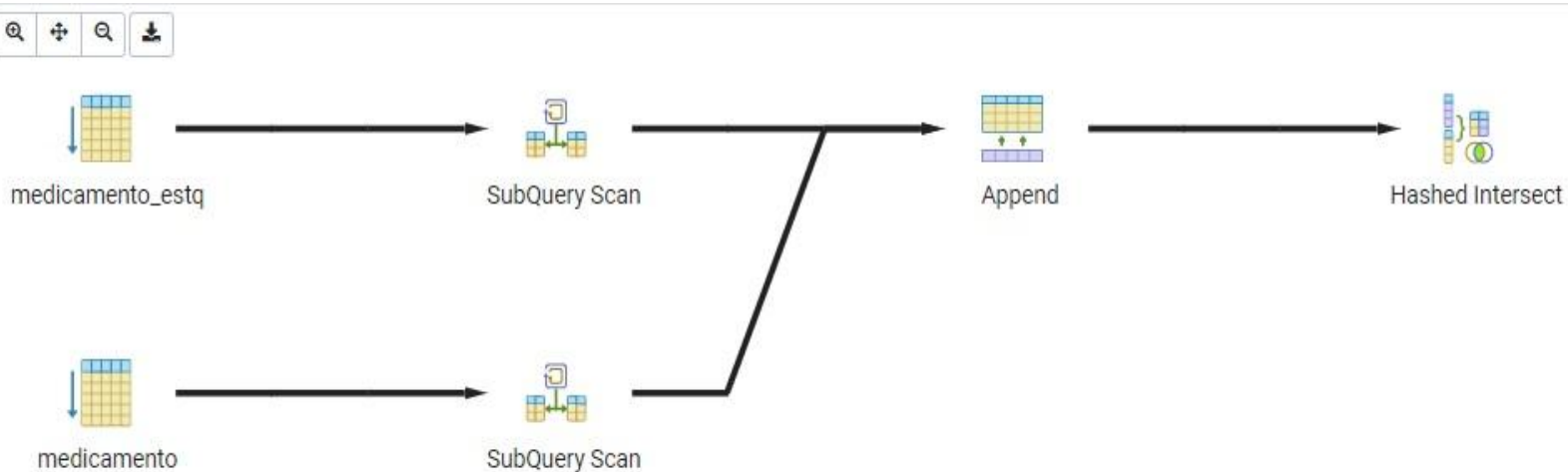


No plano de execução apresentado, primeiramente foi feita a varredura sequencial em “peruanos” e em “peruano\_mortalidade”, no Hash Inner Join, foi realizada a igualdade entre os mesmos, logo após isso, é feita a agregação.

Plano de Execução da consulta:

-- select intersect  
-- seleciona todos medicamentos os quais existem unidades disponíveis para consumo

```
SELECT a.medimento_id as numero_medimento  
FROM medicamento_estq as a  
WHERE a.quantidade > 0  
INTERSECT  
SELECT b.id  
FROM medicamento as b;
```



No exemplo apresentado, o SubQuery Scan é uma sub consulta de “medicamento” e “medicamento\_estq”, Append acrescentou a “medicamento\_estq” o “medicamento”, juntando os dois, e o Hashed Intersect fez a intersecção entre os mesmos. Também observa-se no Hashed Intersect que entre as duas junções foi pintado de verde somente os valores que estão ao mesmo tempo na intersecção entre “medicamento\_estq” e “medicamento”.

# ÁLGEBRA RELACIONAL

Consulta 1:

-- select order by  
-- seleciona todos as mortalidades em ordem decrescente

```
SELECT DISTINCT a.peruario_id,  
                a.dt_atualizacao  
FROM peruario_mortalidade as a  
ORDER BY a.dt_atualizacao DESC;
```

Álgebra Relacional:

$\pi_{dt\_atualizacao \text{ desc}} (\pi_{peruario\_id, dt\_atualizacao} (peruario\_mortalidade))$

Consulta 2:

```
-- select group by com having
-- selecina todos os peruários que tiveram mortalidade acima de 100
SELECT a.peruario_id
FROM peruario_mortalidade as a
GROUP BY a.peruario_id, a.quantidade
HAVING a.quantidade > 100;
```

Álgebra Relacional:

quantidade > 100 (peruario\_id y peruario\_id)