Operador de ordenação e outros operadores relacionados

Sumário

- Ordenação
- Remoção de Duplicatas
- Merge Join
- Agrupamento
- Operadores de Conjunto
  - Substituição por junções
- Materialização em vez de ordenação

Ordenação

- A ordenação dos registros pode ser necessária devido à necessidade de retornar dados ordenados

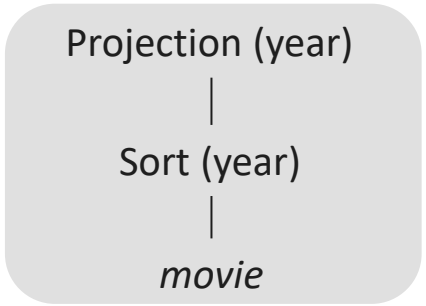
- No exemplo abaixo
  - o ORDER BY indica que os filmes devem ser ordenados por ano

```
SELECT year
FROM movie
ORDER BY year
```

Ordenação

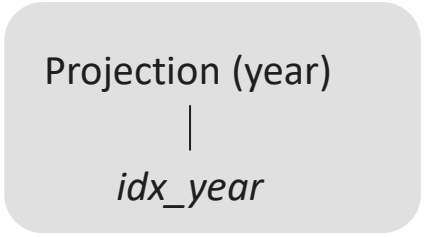
- A ordenação pode ser alcançada usando o operador de ordenação Sort
- Custo do operador de ordenação
  - Overhead de memória: os registros precisam ser materializados
  - Overhead de processamento: os registros precisam ser ordenados

- Caso haja muitos registros a ordenar, pode ser necessário recorrer à ordenação externa!

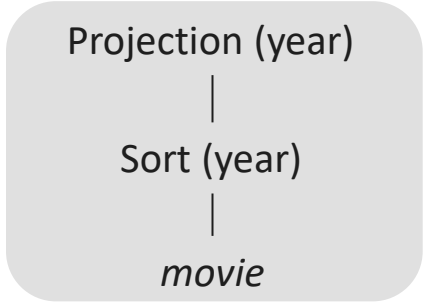


Ordenação

- Qual estratégia é melhor?



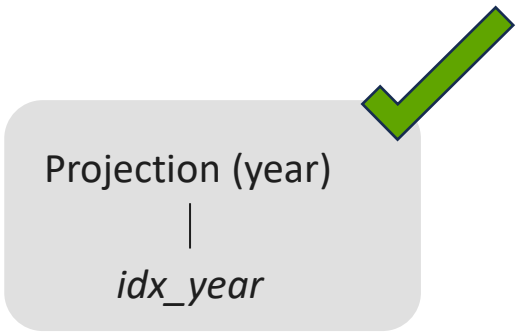
Plano A



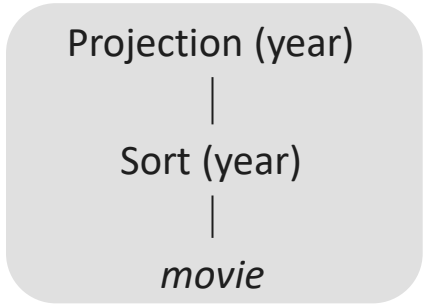
Plano B

Ordenação

- Plano A é melhor
  - O índice já traz os dados ordenados por year
  - Isso evita o custo de ordenação explícita



Plano A

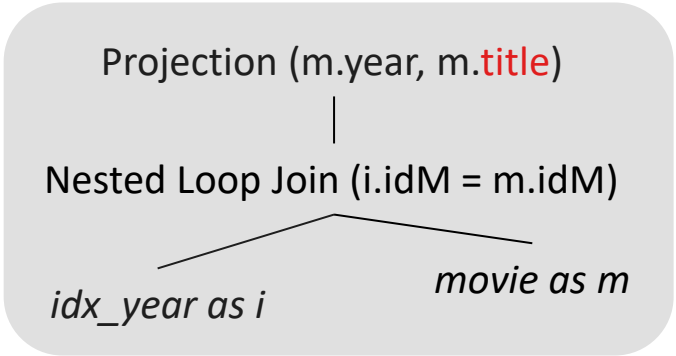


Plano B

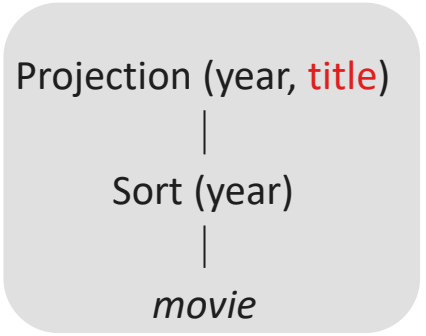
Ordenação

- E se a coluna title também precisar ser devolvida?
- Agora, o plano A exige uma etapa de complementação

- Qual é melhor?



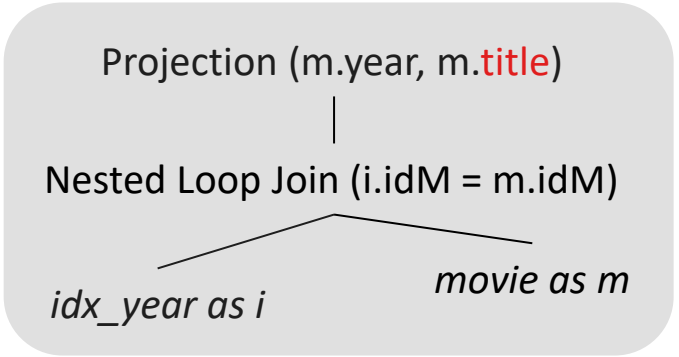
Plano A



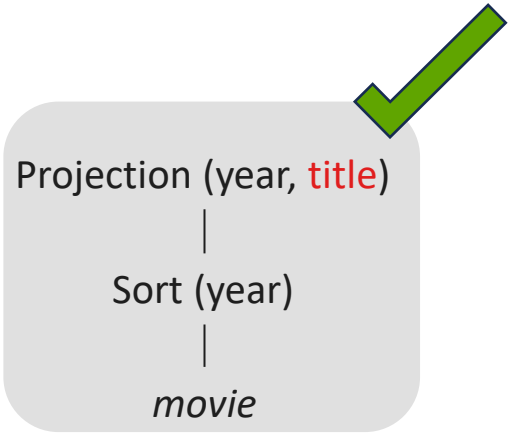
Plano B

Ordenação

- O plano B é melhor
  - O plano A requer muitos acessos aleatórios à tabela movie
- Para que o índice seja mais vantajoso, os SGBDs podem usar técnicas que minimizam a aleatoriedade no acesso aos dados
  - Ex.
    - Multi Range Read (MySQL)



Plano A



Plano B

Ordenação

- Além do ORDER BY, alguns operadores do plano de execução podem ser valer de dados ordenados
  - Remoção de duplicatas
  - Agregação
  - Operadores de conjunto
  - Algoritmo Merge Join
  - ...

- Todos eles podem
  - Usar índices para ter acesso aos registros já ordenados
  - Ou ordenar os registros explicitamente

Sumário

- Ordenação
- Remoção de Duplicatas
- Merge Join
- Agrupamento
- Operadores de Conjunto
  - Substituição por junções
- Materialização em vez de ordenação

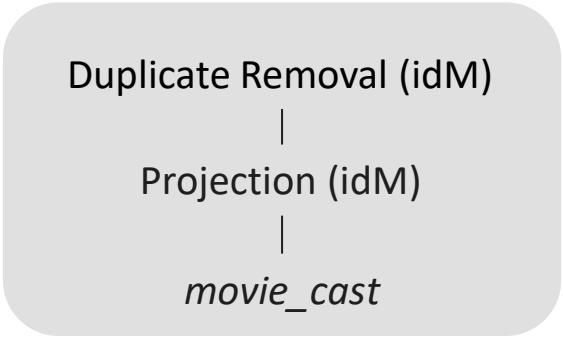
Remoção de duplicatas

- A remoção de duplicatas é realizada pelo operador Duplicate Removal
- Esse operador exige que os dados esteja ordenados
- Cenários de uso
  - Quando a consulta exige a remoção de duplicatas
  - Quando se deseja materializar resultados intermediários
    - Nesse caso, se aplicável, pode-se remover duplicatas antes da materialização para reduzir o consumo de memória

Remoção de duplicatas

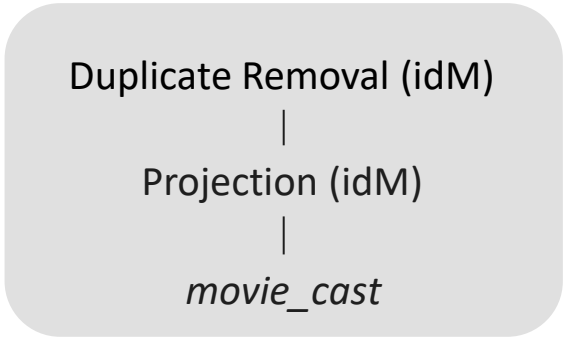
- Exemplo: Retornar ids de filmes que tenham elenco, sem repetir os ids

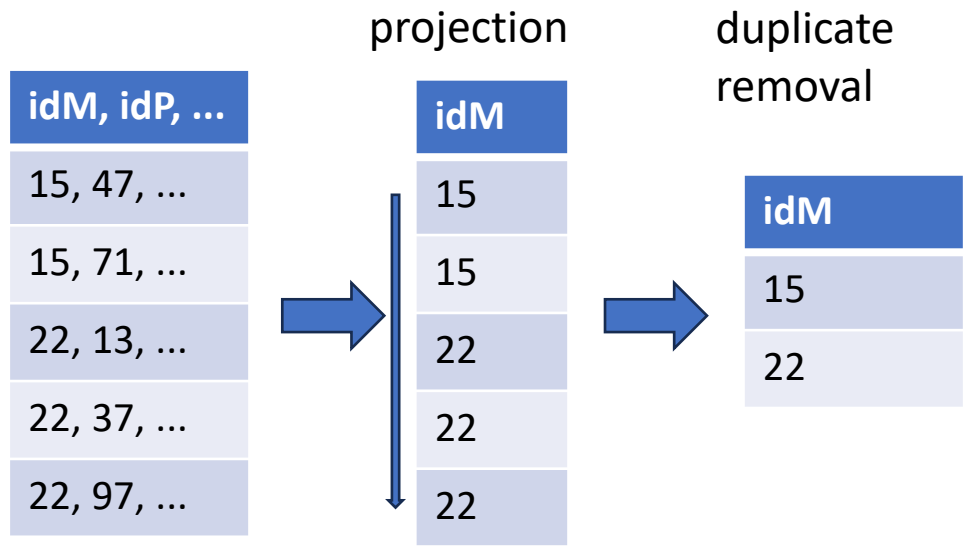
```
SELECT DISTINCT idM
FROM movie_cast
```



Remoção de duplicatas

- Se as tuplas estiverem ordenadas por idM
  - dá para calcular a resposta percorrendo sequencialmente a lista



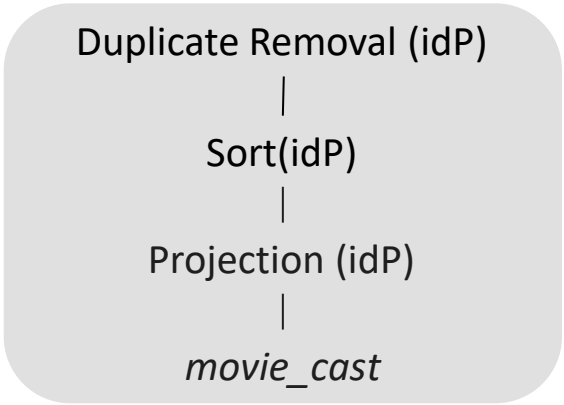


Remoção de duplicatas

- Exemplo: Retornar ids de pessoas que atuaram em filmes, sem repetir os ids

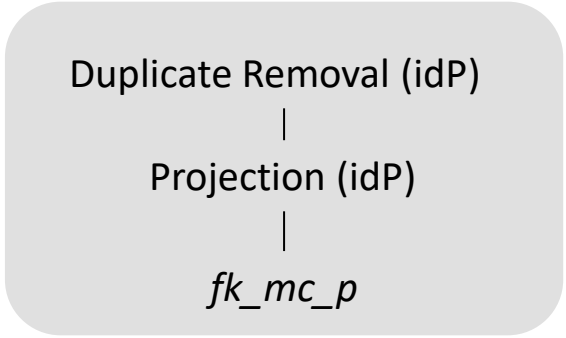
- Como movie_cast não está ordenada por idP, foi necessário fazer a ordenação

```
SELECT DISTINCT idP
FROM movie_cast
```



Remoção de duplicatas

- Uma alternativa seria recorrer ao índice secundário



Sumário

- Ordenação
- Remoção de Duplicatas
- Merge Join
- Agrupamento
- Operadores de Conjunto
  - Substituição por junções
- Materialização em vez de ordenação

Merge Join

- O Merge Join é um algoritmo de junção
- Aplicável para junções do tipo equi-Join
  - O algoritmo parte do pressuposto que os dados estejam ordenados pelas colunas usadas na junção

- Variações
  - Merge Join
  - Merge Left Outer Join
  - Merge Right Outer Join
  - Merge Full Outer Join
  - Merge Left Semi Join
  - Merge Right Semi Join
  - Merge Left Anti Join
  - Merge Right Anti Join

Merge Join

- Exemplo: Retorne título de filmes e nomes de personagens

```
SELECT m.title, mc.c_name
FROM movie m JOIN movie_cast USING (idM)
```

Merge Join

- Se as tuplas estiverem ordenadas pela coluna da junção
  - dá para combinar as tuplas percorrendo sequencialmente as listas

movie

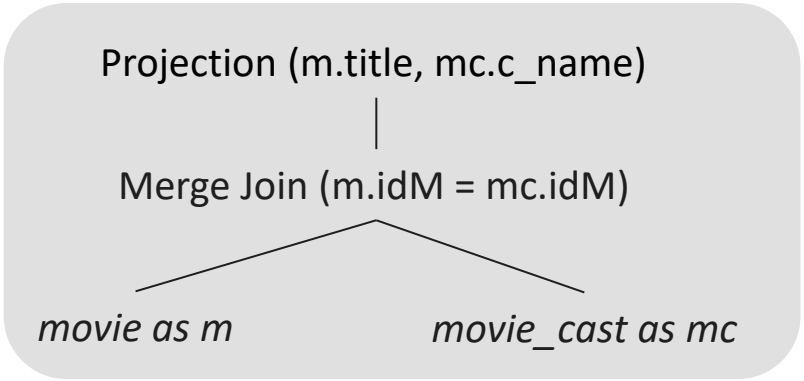
idM	title
5	Star Wars
8	Forrest Gump
12	Pulp Fiction

movie_cast

idM	c_name
5	Skywalker
5	H. Solo
5	Leia
8	F. Gump
8	J. Curran

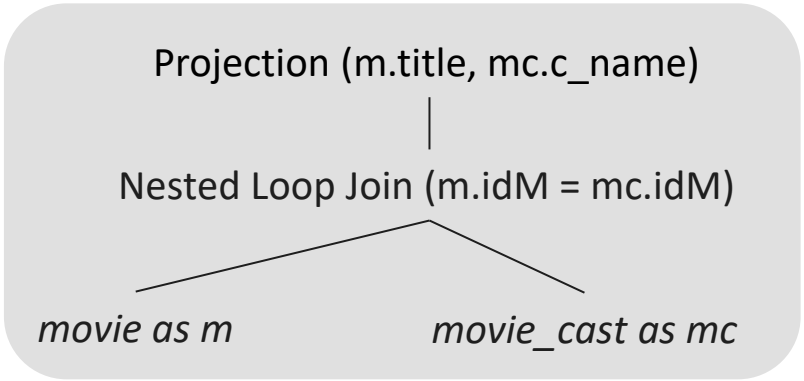
merge join

idM	title	idM	C_name
5	Star Wars	5	Skywalker
5	Star Wars	5	H. Solo
5	Star Wars	5	Leia
8	Forrest Gump	8	F. Gump
8	Forrest Gump	8	J. Curran

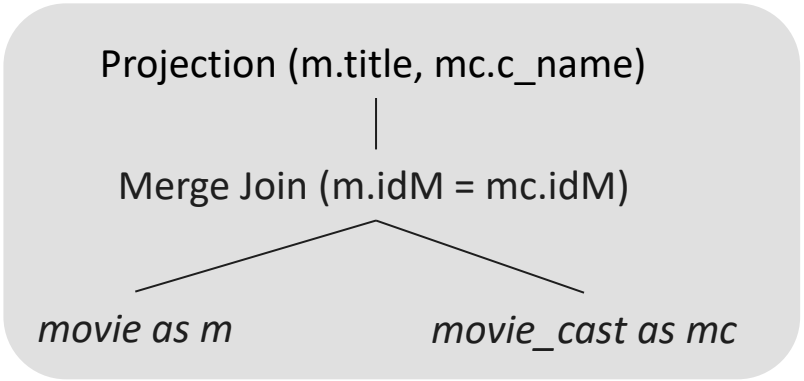


Merge Join

- Qual estratégia é melhor?



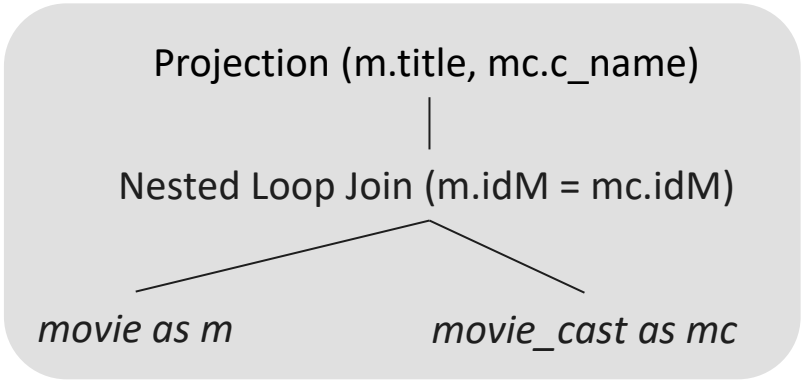
Plano A



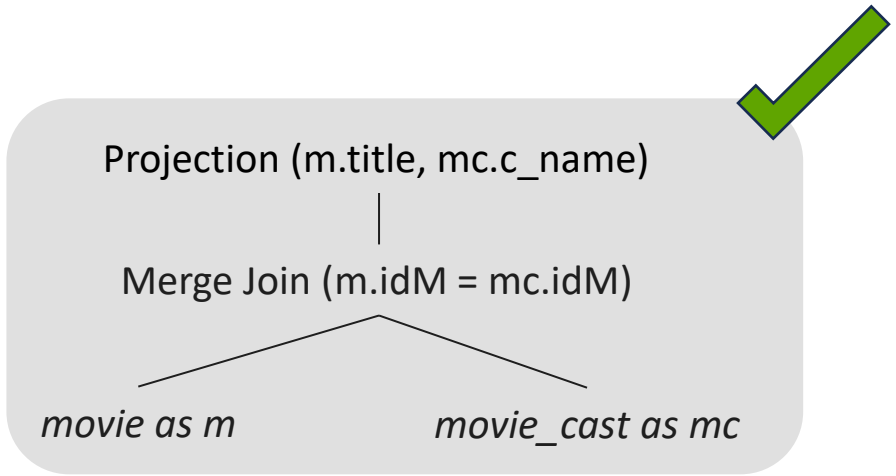
Plano B

Merge Join

- Plano B é melhor
  - Não acessa a mesma página mais de uma vez



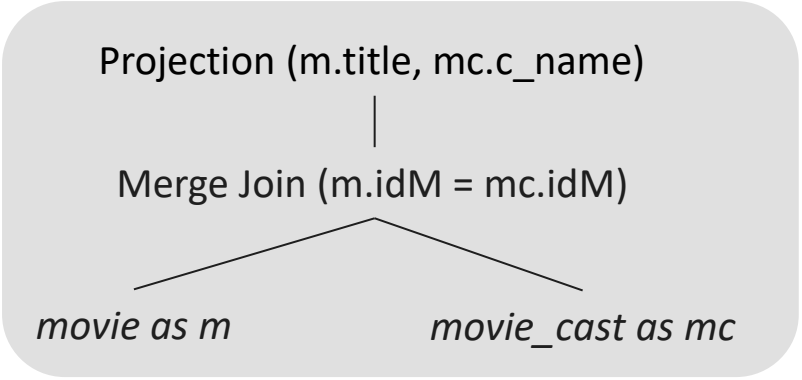
Plano A



Plano B

Merge Join

- O plano B funciona se as tabelas estiverem ordenadas pelo atributo de junção
  - É o caso do DBest/MySQL, onde as tabelas acessadas são índices primários que tem idM como prefixo da chave de busca



Plano B

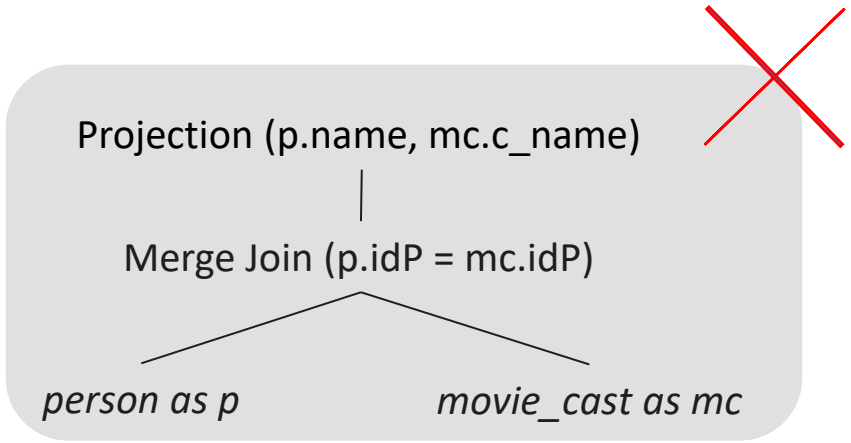
Merge Join

- Exemplo: Retorne nomes de pessoas e os personagens que elas desempenharam

```
SELECT p.name, mc.c_name
FROM person p JOIN movie_cast USING (idP)
```

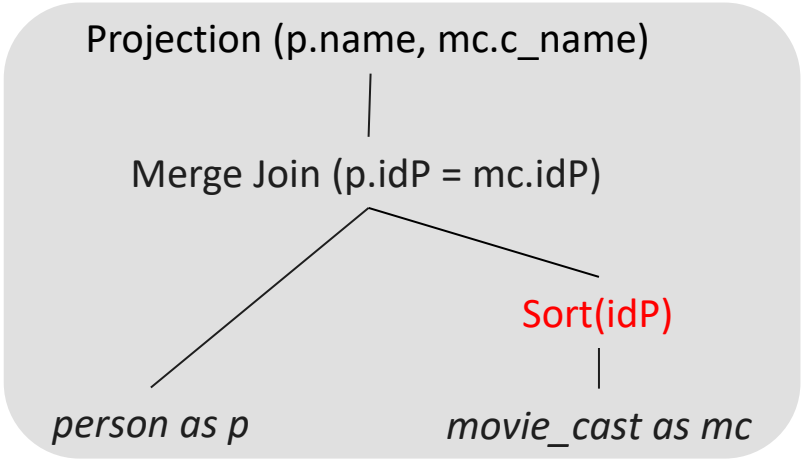
Merge Join

- O plano abaixo não funciona
  - A tabela movie_cast não está ordenada por idP



Merge Join

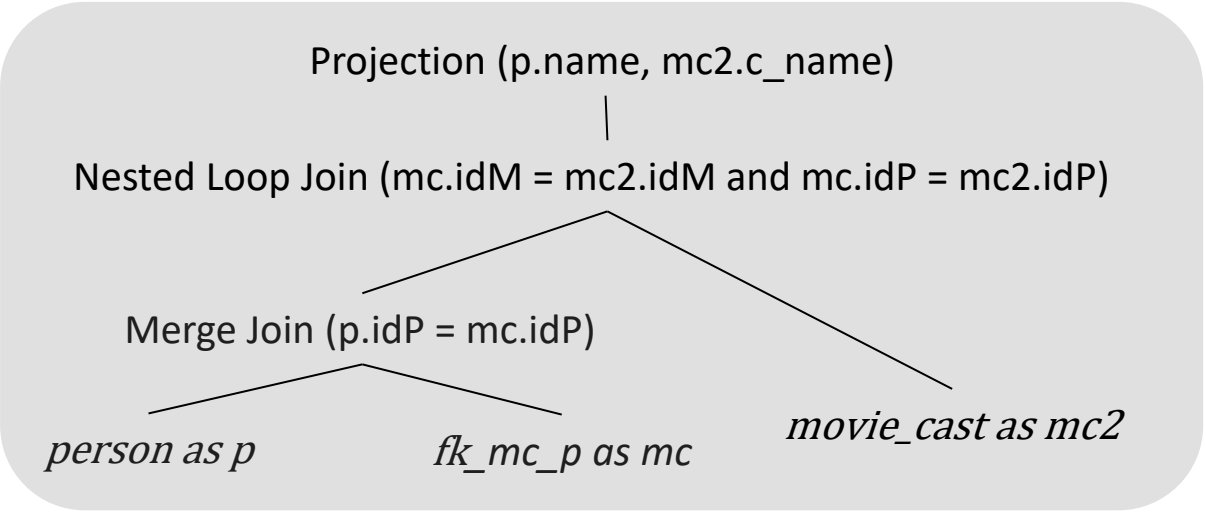
- Uma saída é ordenar os dados antes do Merge Join
- No entanto, agora existe o custo da ordenação
  - Overhead de processamento
  - Overhead de memória



Plano A

Merge Join

- Também pode-se recorrer à índices secundários
- No entanto
  - Isso leva à necessidade de complementação, com acessos aleatórios à tabela



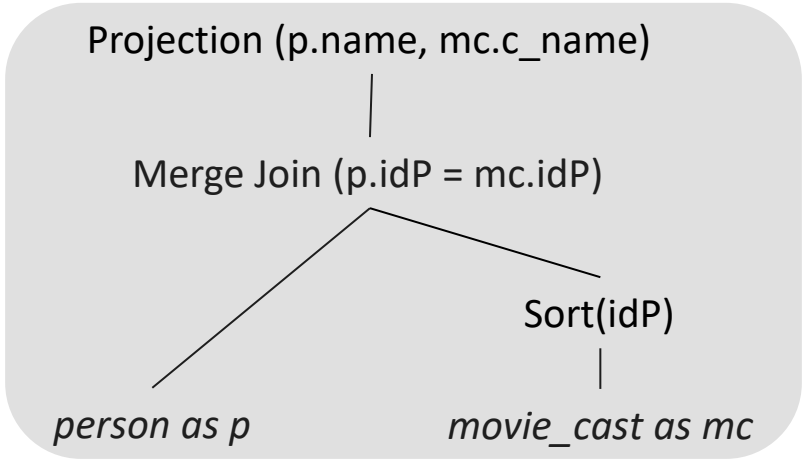
Plano B

Estruturas de dados

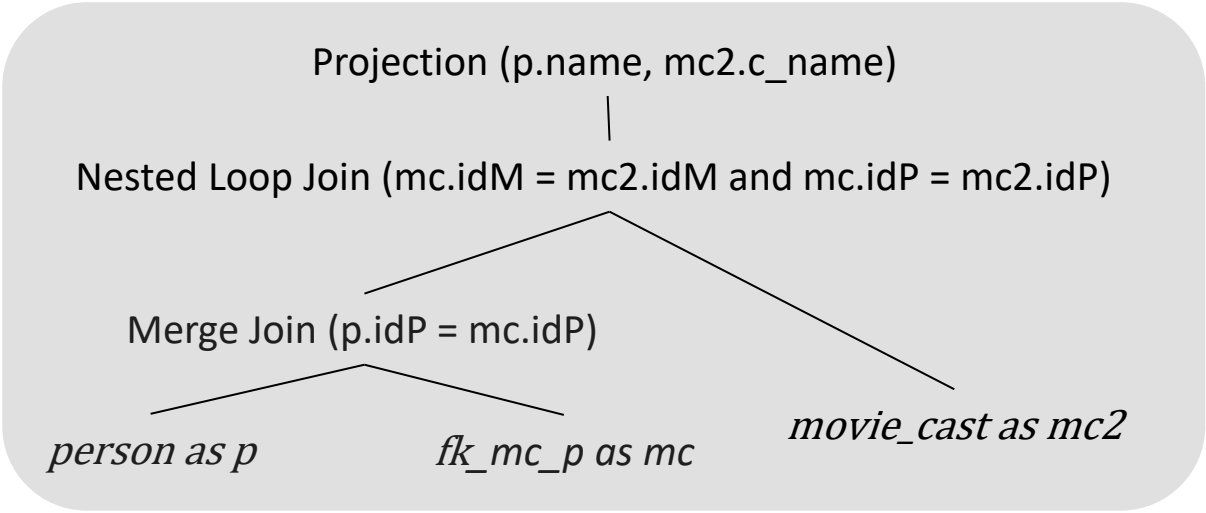
nome	Tipo	tipo	chave	valor
person	tabela	B+tree clust.	idP	idP, p_name
movie_cast	tabela	B+tree clust.	idM, idP	idM, idP, c_name, ...
Fk_mc_p	índice	B+tree ñ clust.	idP	idM, idP

Merge Join

- Qual é melhor?



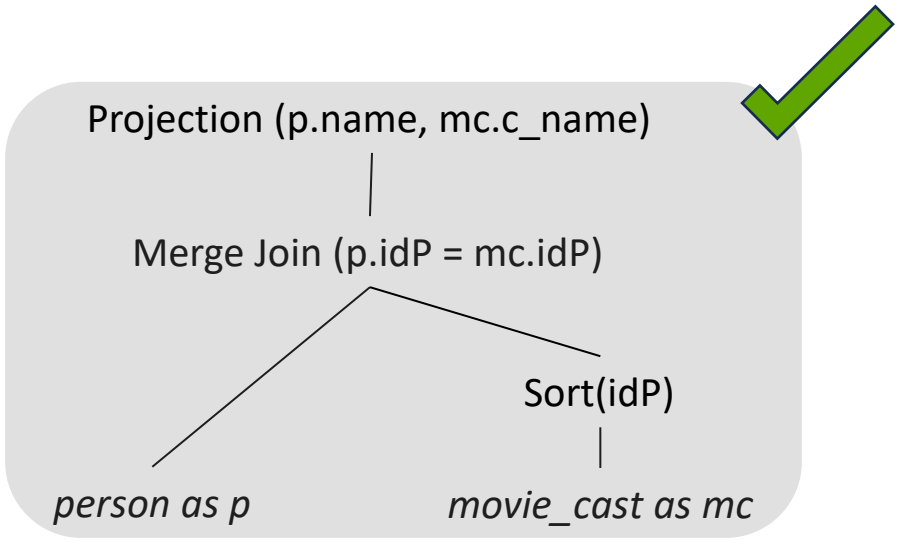
Plano A



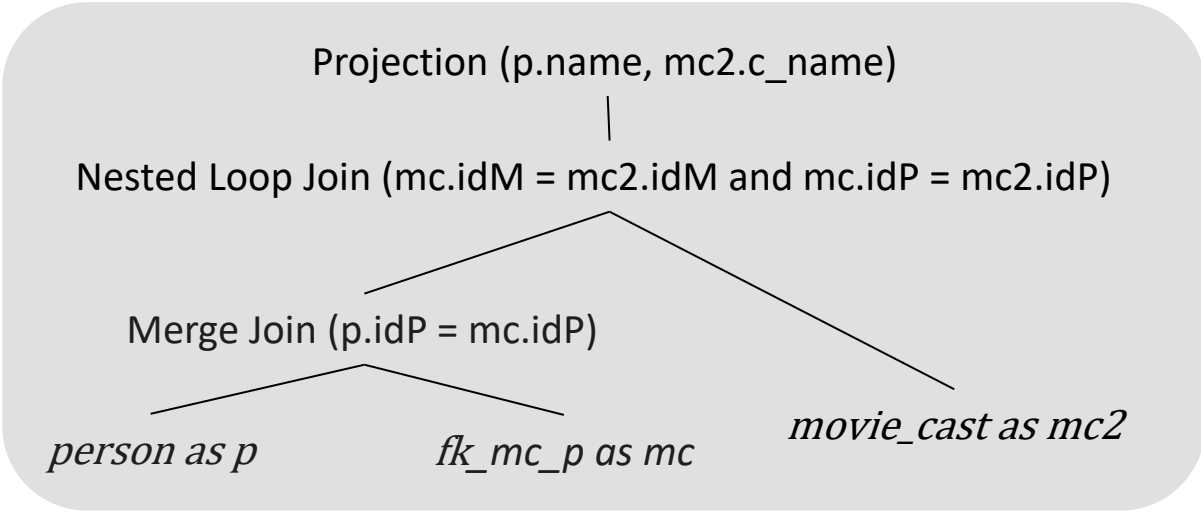
Plano B

Merge Join

- O plano A é melhor
  - A quantidade de acessos aleatórios do plano B é muito alto
- Caso o lado externo fosse menor (um filtro seletivo sobre person), o plano B seria melhor



Plano A



Plano B

Merge Join

- Quando usar cada algoritmo de junção?
  - Nested Loop Join
    - Lado externo possui poucos registros e pode-se usar índice
  - Merge Join
    - Ambos os lados já estão ordenados pelo atributo de junção
  - Hash Join
    - Quando os dois lados possuem muitos registros ou
    - Quando não há índice que possa ser usado

Sumário

- Ordenação
- Remoção de Duplicatas
- Merge Join
- Agrupamento
- Operadores de Conjunto
  - Substituição por junções
- Materialização em vez de ordenação

Agrupamento

- O agrupamento é representado pelo operador Group by
- Ele é usado quando os registros precisam ser agrupados por alguma coluna, sendo que para cada grupo são aplicadas funções de agregação
- Em SQL, esse recurso é acionado pela cláusula GROUP BY
- O operador Group by exige que os dados estejam ordenados pela coluna de agrupamento

Agrupamento

- Exemplo: retornar quantidade de membros de elenco por idM

```
SELECT idM, count(*)
FROM movie_cast
GROUP BY idM
```

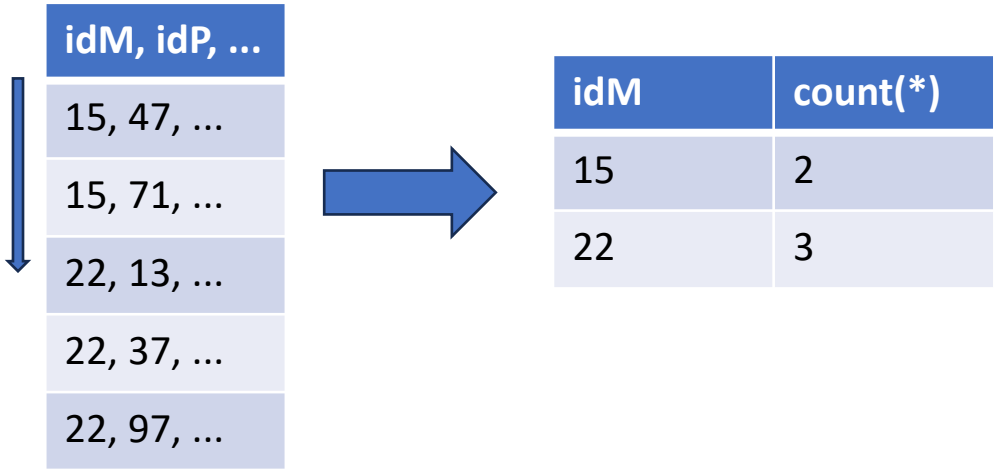
Agrupamento

- Se as tuplas estiverem ordenados por idM
  - dá para calcular a resposta percorrendo sequencialmente a lista
  - O mesmo vale para outras funções de agregação

Group By (idM) - return idM, count(*)

|

movie_cast

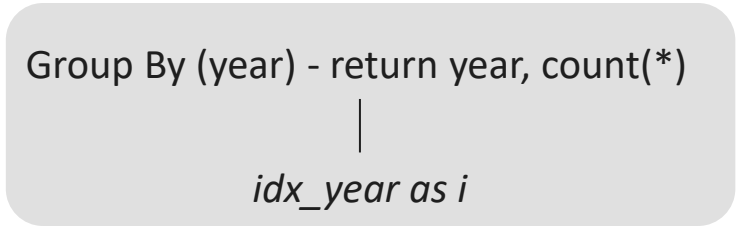


Agrupamento

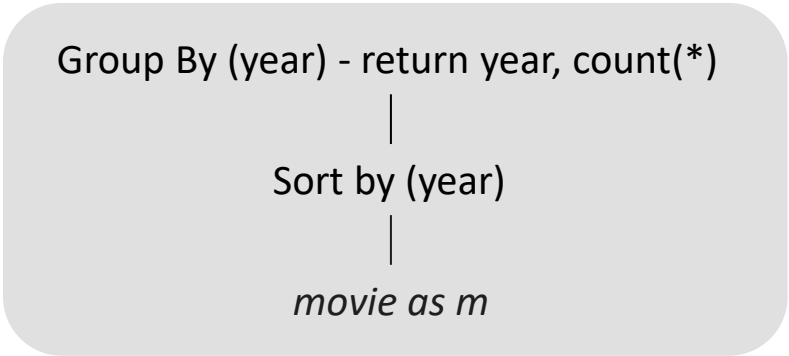
- Exemplo: A consulta abaixo retorna o ano e a quantidade de filmes que ocorreram naquele ano

```
SELECT year, COUNT(*)
FROM movie
GROUP BY year
```

- Qual plano é melhor?



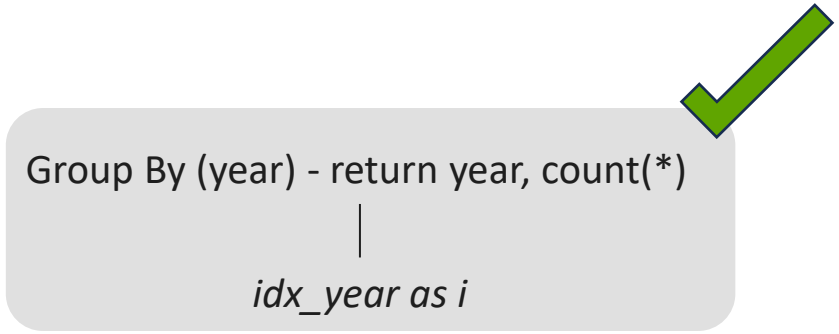
Plano A



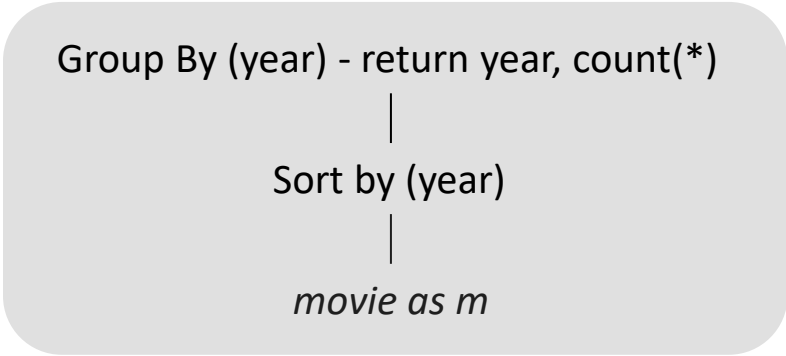
Plano B

Agrupamento

- O plano A é melhor
  - O índice já recupera os dados ordenados



Plano A



Plano B

Agrupamento

- Ao agrupar, se possível, escolha atributos indexados como fatores de agrupamento

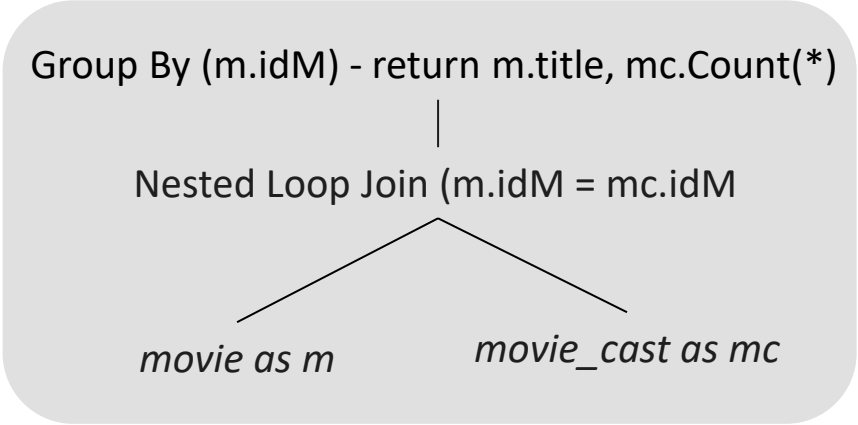
- Isso auxilia o otimizador a escolher planos melhores
  - Por exemplo, que resolvam o agrupamento por meio de um índice

Agrupamento

- Exemplo:
  - No plano A, o agrupamento é feito por idM
  - A junção já entrega os registros agrupados por idM.
  - Ou seja, a ordenação explícita aqui é desnecessária

Consulta: título do filme e a quantidade de artistas

```
SELECT m.idM, COUNT(*)
FROM movie m
JOIN movie_cast mc ON m.idM = mc.idM
GROUP BY m.idM
```



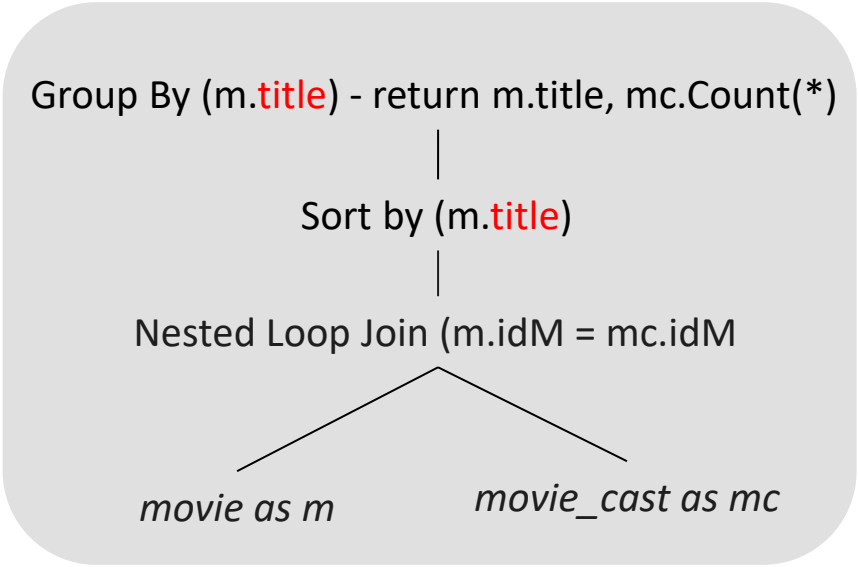
plano A

Agrupamento

- No plano B, o agrupamento é feito por título
  - O plano precisa ordenar os registros por título

Consulta: título do filme e a quantidade de artistas

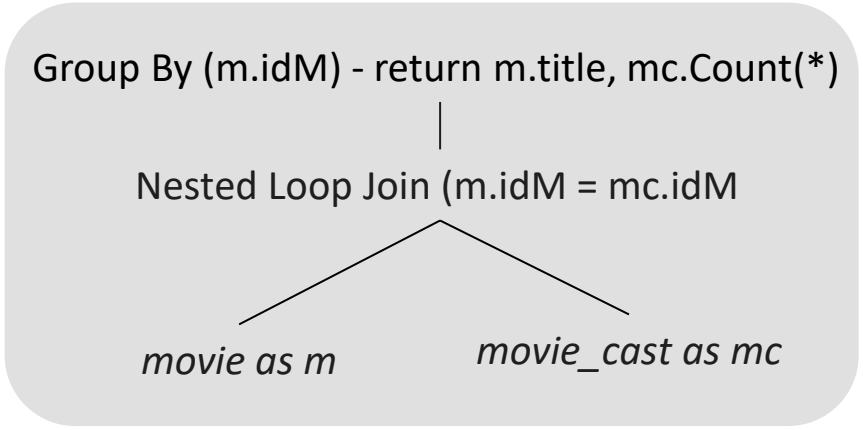
```
SELECT m.idM, COUNT(*)
FROM movie m
JOIN movie_cast mc ON m.idM = mc.idM
GROUP BY m.title
```



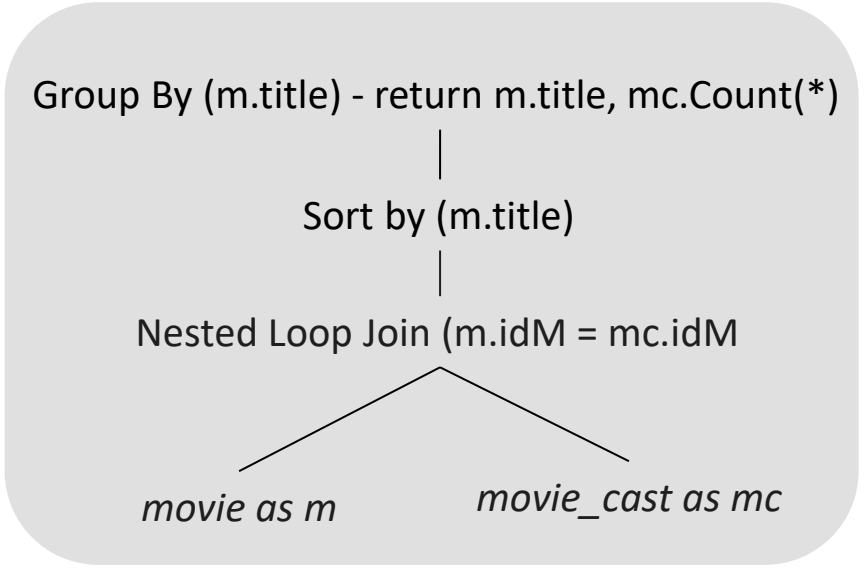
plano B

Agrupamento

- Qual é melhor?



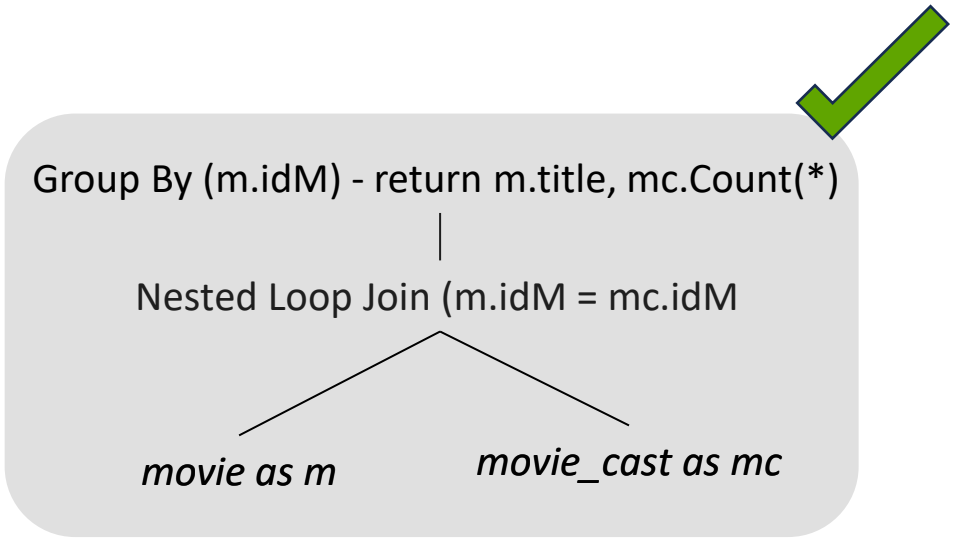
plano A



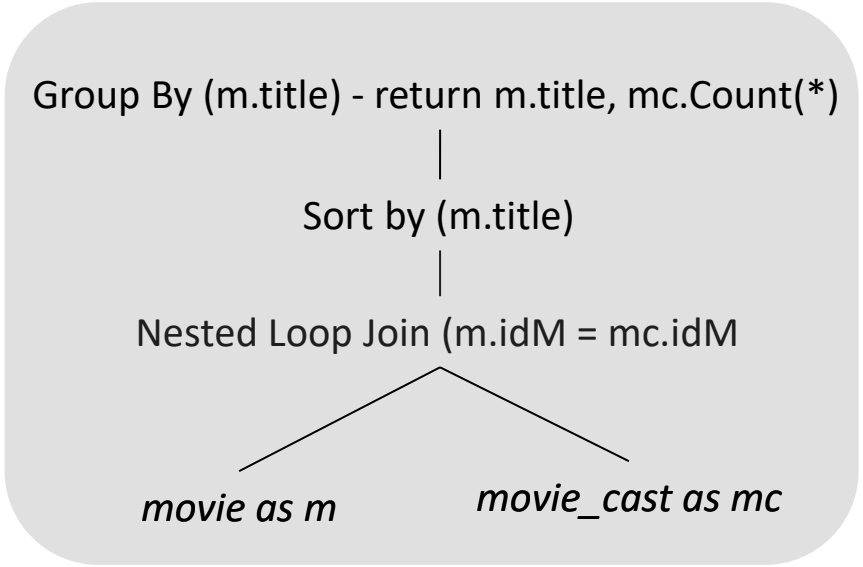
plano B

Agrupamento

- O plano A é melhor
  - consegue agrupar sem precisar de uma ordenação explícita



plano A



plano B

Sumário

- Ordenação
- Remoção de Duplicatas
- Merge Join
- Agrupamento
- Operadores de Conjunto
  - Substituição por junções
- Materialização em vez de ordenação

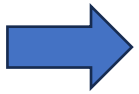
Operadores de Conjunto

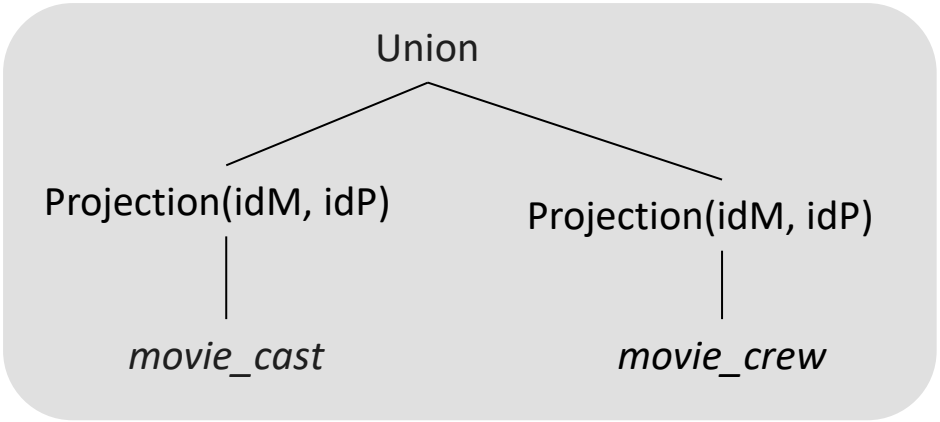
- Operadores de conjunto disponíveis
  - Union
  - Difference
  - Intersect
  - Append
- Os operadores possuem correspondência direta com SQL
  - Append = UNION ALL
- Todos eles exigem que os dados estejam ordenados

Operadores de Conjunto - União

- Exemplo: Retorne ids de filmes e ids de pessoas para todas relações de pessoa e filme, seja como membros da equipe de atuação ou de produção

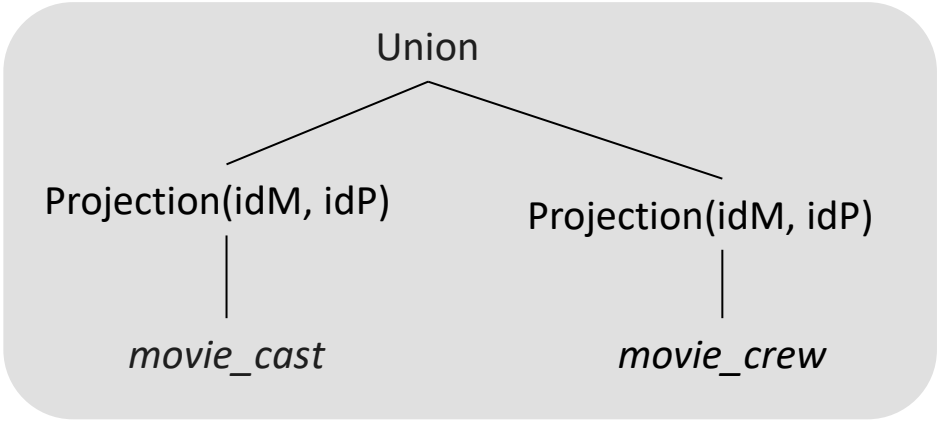
```
SELECT idM, idP
FROM movie_cast
UNION
SELECT idM, idP
FROM movie_crew
```





Operadores de Conjunto - União

- Se os registros estiverem ordenados
  - dá para calcular a resposta percorrendo sequencialmente as listas



movie_cast

idM	idP
15	47
15	71
22	13
22	37
22	97

movie_crew

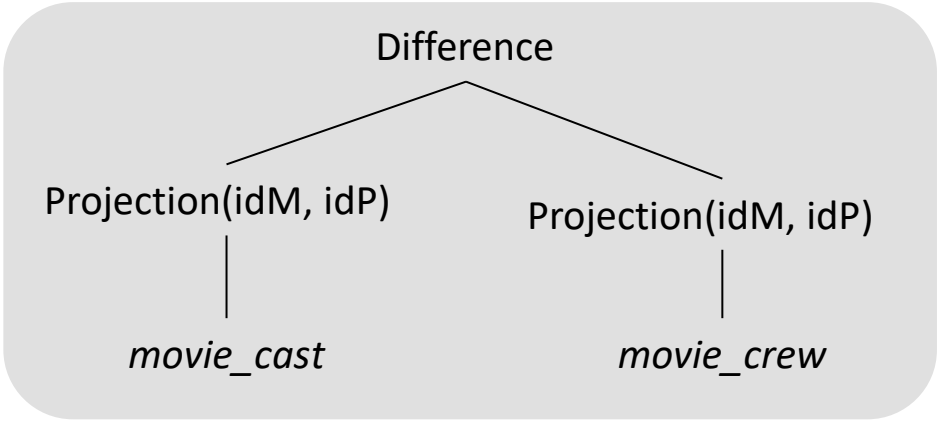
idM	idP
15	22
22	13
40	14

união

idM	idP
15	22
15	47
15	71
22	13
22	37
...	...

Operadores de Conjunto - Diferença

- O mesmo se aplica para a diferença



movie_cast

idM	idP
15	47
15	71
22	13
22	37
22	97

movie_crew

idM	idP
15	22
22	13
40	14

diferença

year	name
15	47
15	71
22	37
22	97

Operadores de Conjunto - Interseção

- E para a interseção

movie_cast

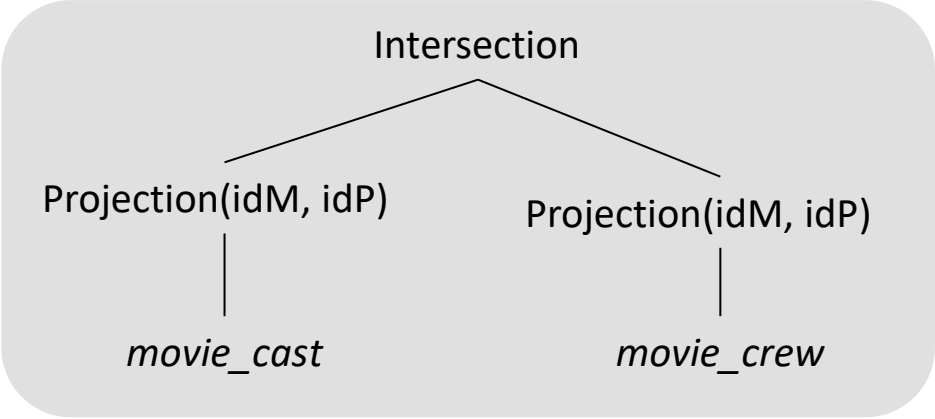
idM	idP
15	47
15	71
22	13
22	37
22	97

movie_crew

idM	idP
15	22
22	13
40	14

interseção

year	name
22	13



Sumário

- Ordenação
- Remoção de Duplicatas
- Merge Join
- Agrupamento
- Operadores de Conjunto
  - Substituição por junções
- Materialização em vez de ordenação

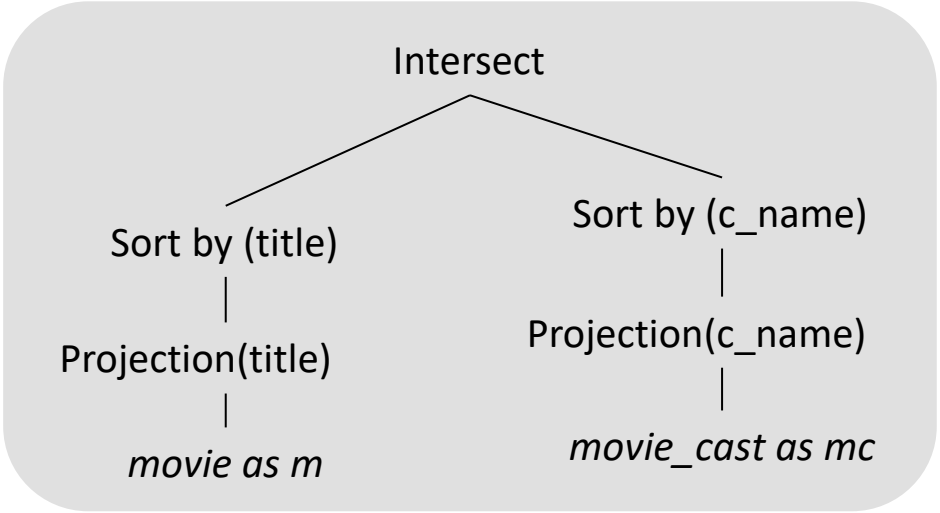
Substituição por junções

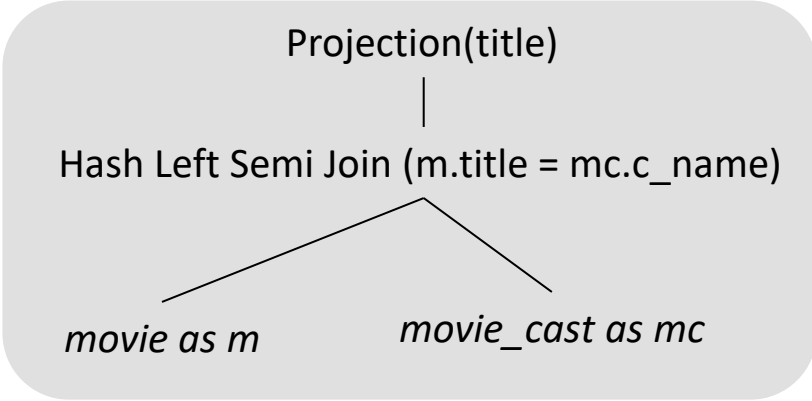
- SQL disponibiliza três operadores de conjunto
  - UNION
  - INTERSECT
  - EXCEPT

- Dois deles podem ser resolvidos de outras formas
  - INTERSECT
    - Ex. Left Semi-Join
  - EXCEPT
    - Ex. Left Anti-Join

Substituição por junções

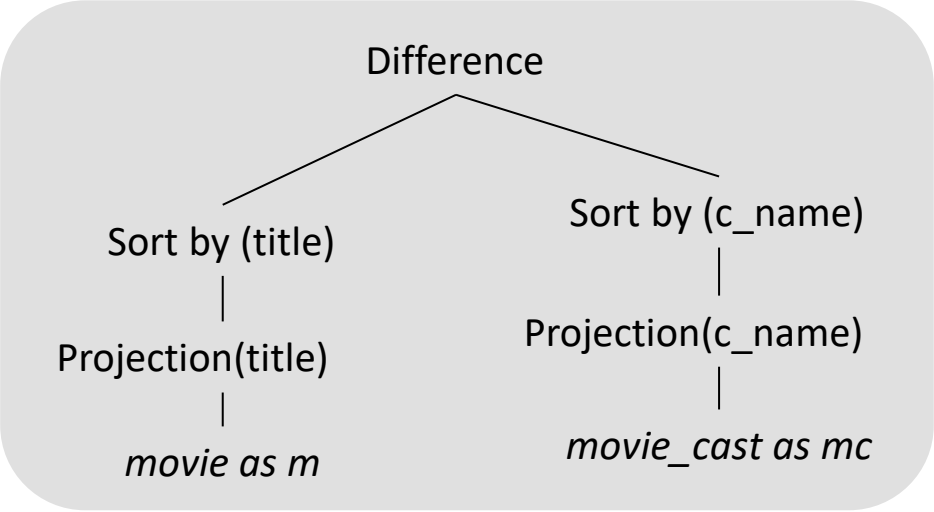
- Plano A:
  - Baseado no operador Intersection
- Plano B:
  - Baseado em algum operador do tipo Left Semi Join (ex. Hash)
- Os dois são equivalentes

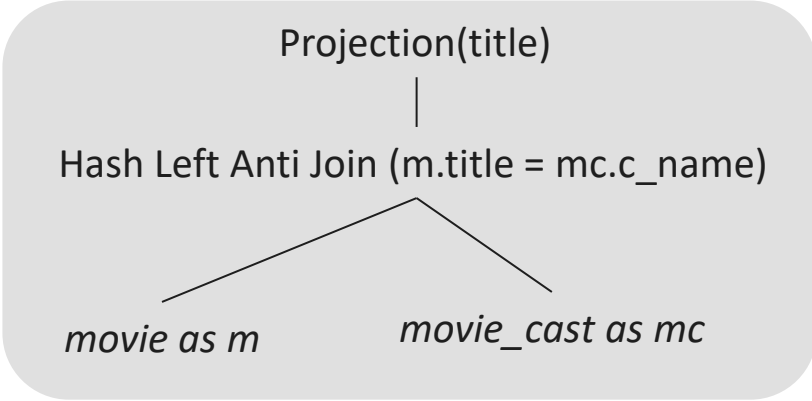




Substituição por junções

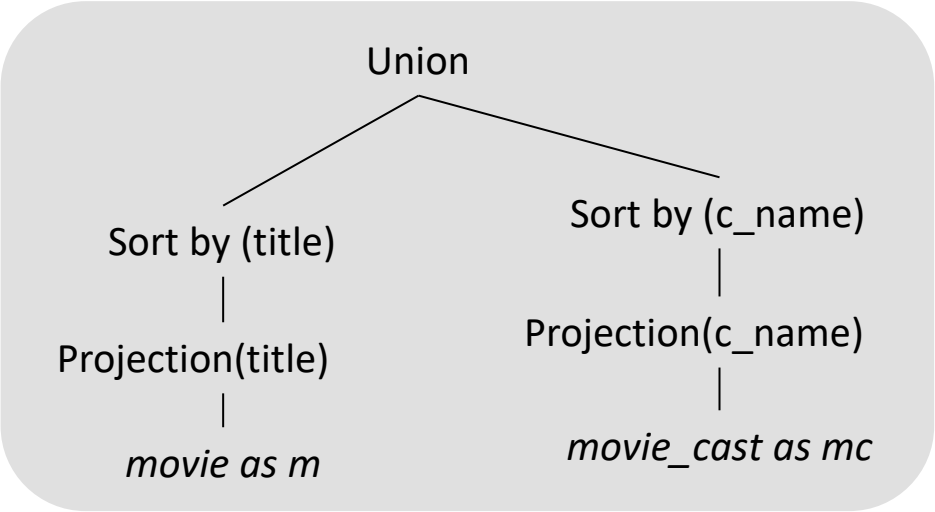
- Plano A:
  - Baseado no operador Difference
- Plano B:
  - Baseado em algum operador do tipo Left Anti Join (ex. Hash)
- Os dois são equivalentes





Substituição por junções

- Operador de União
  - É o único que não tem substituto



Sumário

- Ordenação
- Remoção de Duplicatas
- Merge Join
- Agrupamento
- Operadores de Conjunto
  - Substituição por junções
- Materialização em vez de ordenação

Materialização em vez de Ordenação

- Os algoritmos que dependem de dados ordenados
  - Merge Join
  - Duplicate Removal
  - Group by
  - Intersection
  - Difference
  - Union

- E se os dados não estiverem ordenados, e a ordenação for uma operação muito cara?
  - Nesse caso, pode-se empregar variações baseadas em tabelas hash

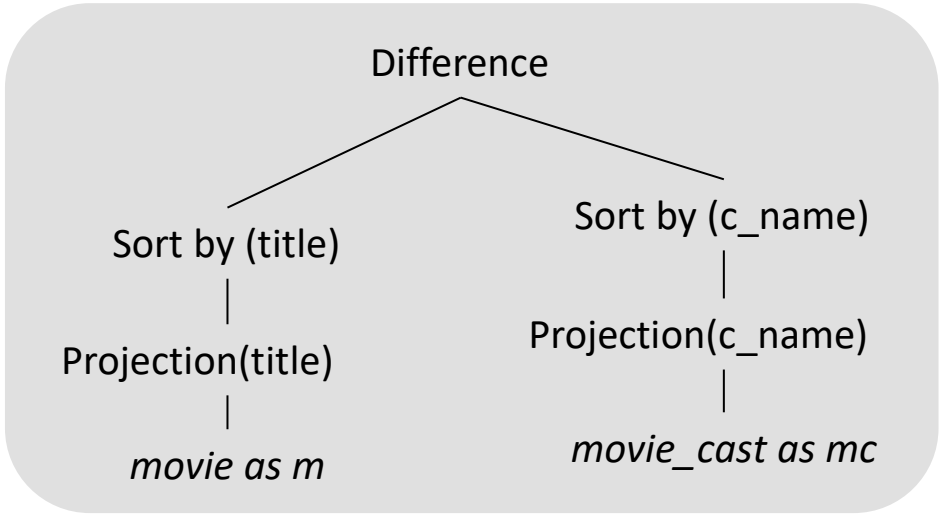
Materialização em vez de Ordenação

- Variações

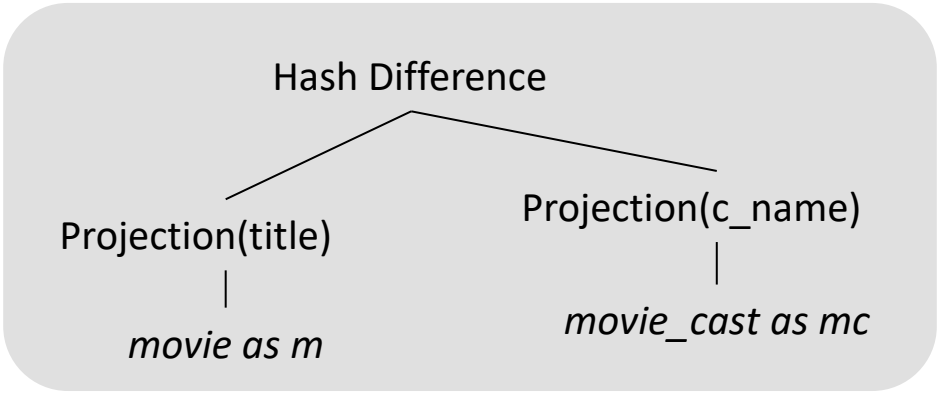
Versão que precisa de dados ordenados	Versão que não precisa de dados ordenados
Merge Join	Hash Join
Duplicate Removal	Hash Duplicate Removal
Aggregation	Hash Aggregation
Intersection	Hash Intersection
Difference	Hash Difference
Union	Hash Union

Materialização em vez de Ordenação

- Plano A: Baseado em dados ordenados
- Plano B: Baseado em tabelas hash
- Qual é melhor?



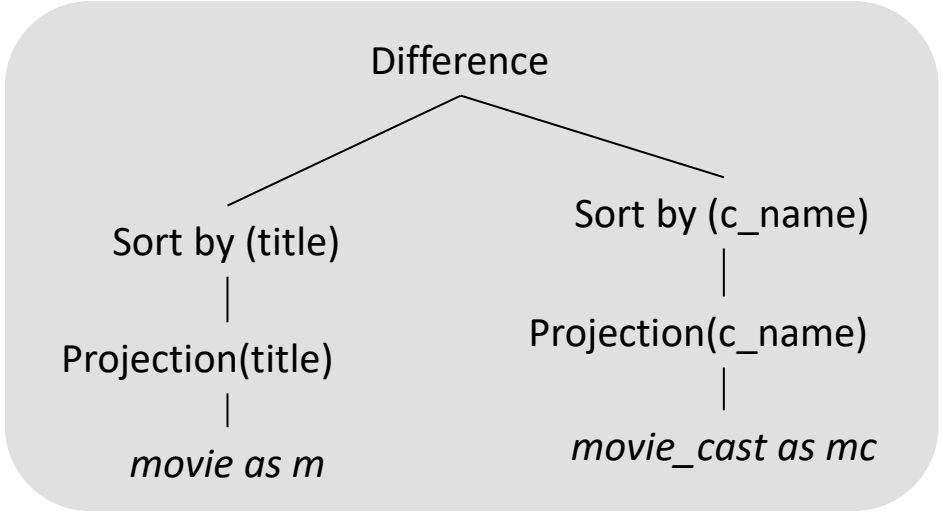
Plano A



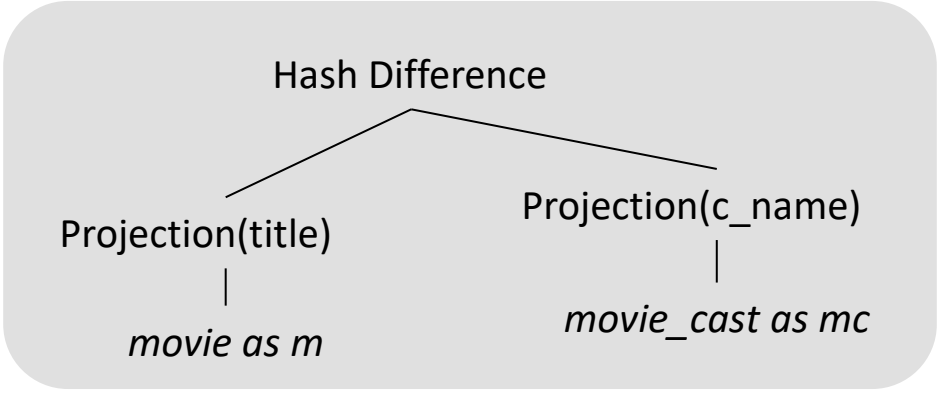
Plano B

Materialização em vez de Ordenação

- Plano A: Baseado em dados ordenados
- Plano B: Baseado em tabelas hash
- Qual é melhor?
  - Vamos descobrir



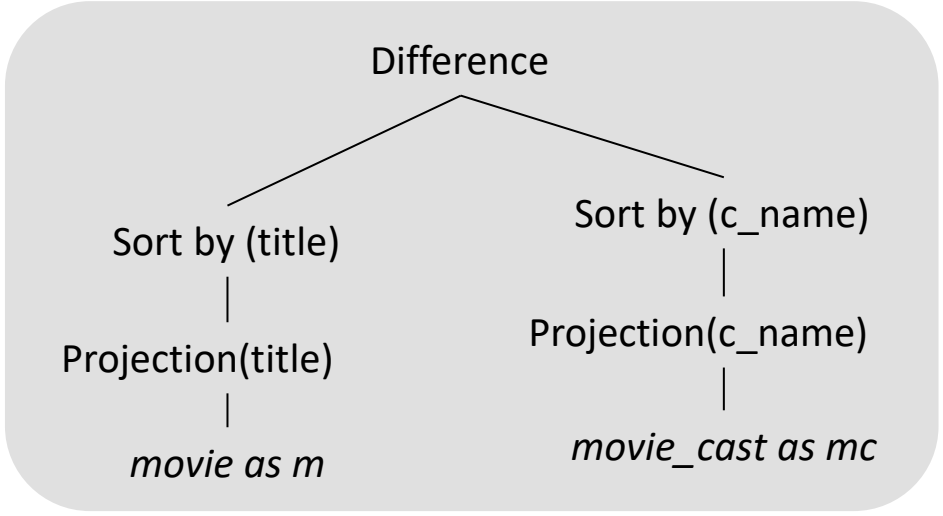
Plano A



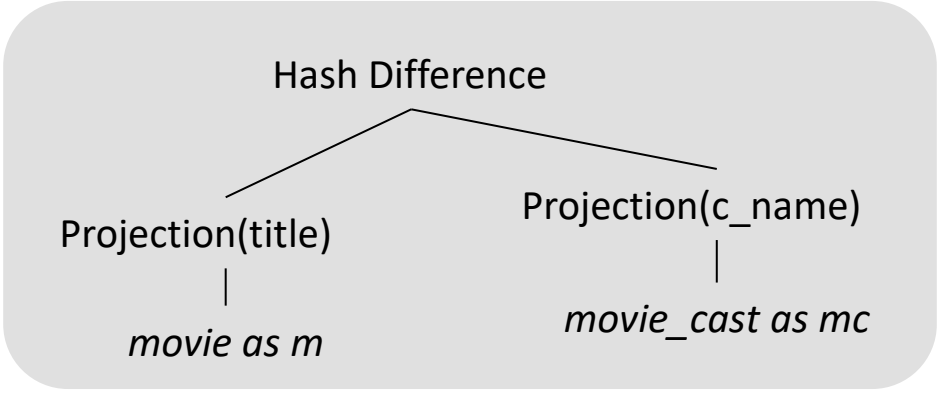
Plano B

Atividade Individual

- Crie os planos A e B
- Analise-os e indique qual é melhor
- Explique o custo dos dois planos.
  - O que levou um deles a ter um indicador de custo pior do que o outro?



Plano A



Plano B