

Orientações:

- Deve ser realizado em grupos de **até** duas pessoas;
- Entrega exclusivamente via Tidia4 no formato PDF (Relatório) e TXT (SQL utilizados na confecção do Relatório);
- Plágio e soluções copiadas de outros colegas, implicarão na nota ZERO a todos os envolvidos.

1 Questão (60%) - Análise Geral

Como será o comportamento dos métodos de agregação à medida que o número de grupos distintos aumenta? Curioso em responder essa pergunta, aplique um experimento para colher dados de tempo de execução (*Execution Time*) sobre o comportamento dos operadores *HashAggregate* e *GroupAggregate*. Para esse experimento iremos precisar:

- Uma relação *Rel* com dois atributos, sendo o *att1* do tipo *bigint* e o *att2* do tipo *numeric*. Não é necessário chave primária ou índice. Essa relação deverá conter 10.000.000 de tuplas;
- O conteúdo do atributo *att1* deverá apresentar 10.000, 100.000, 1.000.000 e 5.000.000 valores distintos. Logo, você pode ter uma relação *Rel* para cada conjunto de valores distintos (*Rel*₁₀₀₀₀, *Rel*₁₀₀₀₀₀ e assim por diante);
- Em cada relação *Rel*, uma instrução SQL de agrupamento deverá ser executada **3 vezes** (no mínimo) e em sequência para cada operador. Como cada estação pode apresentar flutuações, trabalhe com a média dos tempos de execução obtidos. **Atenção, durante a execução do SQL, evite utilizar qualquer recurso de sua estação (Whatzap, Word, jogos, calculadora do Windows, etc.) uma vez que esse procedimento determinará flutuações gritantes nos seus dados coletados.** A instrução SQL deverá apresentar o seguinte formato abaixo:

```
SELECT att1, SUM(att2) FROM Rel GROUP BY att1;
```

Como você é novo na empresa, um antigo funcionário deixou-lhe algumas dicas passo-a-passo de como realizar o seu experimento, bem como configurar o *query planner* e acrescentar dados aleatórios. As dicas são:

Passo 1. Crie a relação *Rel*;

Passo 2. Acrescente os dados randomicamente com o SQL abaixo. Note que o '10000' remete ao número de valores distintos

```
insert into Rel select (10000*random())::bigint, random()
from generate_series(1,10000000) s(i);
```

Passo 3. Atualize as estatísticas com...

```
vacuum analyze Rel;
```

Passo 4. Ajuste os parâmetros da sua sessão com...

```
set work_mem = '640MB';
set max_parallel_workers_per_gather = 0;
```

Passo 5. Force o *HashAggregate*

```
set enable_hashagg = on;
set enable_sort = off;
```

Passo 6. Submeta o SQL (3x ou mais) e colete o tempo de execução em cada submissão

Passo 7. Force o *GroupAggregate*

```
set enable_hashagg = off;
set enable_sort = on;
```

Passo 8. Submeta o SQL (3x ou mais) e colete o tempo de execução em cada submissão

2 Questão (40%) - Análise focada no Projeto

Uma série de reclamações relacionadas a operações do tipo $R \bowtie (\sigma_{\langle condicao \rangle} S)$ levou sua equipe a identificar no ambiente de produção as respectivas variantes mais utilizadas, conforme apresenta a Tabela 1. De posse desses dados, sua equipe deverá determinar (por meio de um experimento) qual variante (ou variantes) é melhor utilizar no **contexto presente** e no **contexto futuro**. Sabe-se que no contexto presente as sessões estão com o limite de memória de trabalho (*work_mem*) de 64Kb. No contexto futuro, as sessões poderão contar com 1Mb de memória de trabalho.

Tabela 1: Lista das variantes de junções condicionadas

Transação	Frequência Diária de Uso
T1. select alocado.* from alocado inner join emp on (alocado.codemp = emp.codemp) where vlsalario >= 1300;	405
T2. select alocado.* from alocado where codemp not in (select codemp from emp where vlsalario < 1300);	175
T3. select alocado.* from alocado where codemp in (select codemp from emp where vlsalario >= 1300);	89
T4. select alocado.* from alocado where not exists (select 1 from emp where emp.codemp = alocado.codemp and vlsalario < 1300);	25
T5. select alocado.* from alocado where exists (select 1 from emp where emp.codemp = alocado.codemp and vlsalario >= 1300);	158
T6. with DP as (select codemp from emp where vlsalario >= 1300) select alocado.* from alocado inner join DP on (alocado.codemp = DP.codemp);	41
T7. select alocado.codemp from (select codemp from emp where vlsalario >= 1300) D inner join alocado on (d.codemp = alocado.codemp)	81

Boa análise!