

# Trabalho CSGBD

Carlos Gabriel de Castro Rodrigues - 509655

Jose Gildasio Freitas do O - 473901

Junho 2023

## 1 Introdução

- A base de dados utilizada é chamada de "clima" que foi retirada do link:

<https://encurtador.com.br/mpAJ3>

- Na base original temos 7 tabelas, sendo elas: north, north\_heast, south, south\_heast, central\_west, stations e columns\_description.

Sendo as 5 primeiras tabelas que tem as maiores quantidades de dados.

Na tabela columns\_description tem todas as informações dos atributos das tabelas, Dados de 2000 a 2021 de todas as cidades do brasil, divididas em regiões e com os dados de temperatura, vento, pressão e umidade, essas informações de todos os dias e hora de cada registro guardadas na base de dados.

- Normalização: Foi feito uma divisão de tabelas e dados realocados para 7 tabelas sendo elas:

info\_vento, info\_temperatura, info\_outras, info\_pressao, info\_umidade, station, columns\_description

Descrito na modelagem como ficou os atributos e as relações entre as novas tabelas, feito um insert de dados específicos a partir de uma consulta para não guardar valores nulos e revolver duplicas que aparecem na versão original.

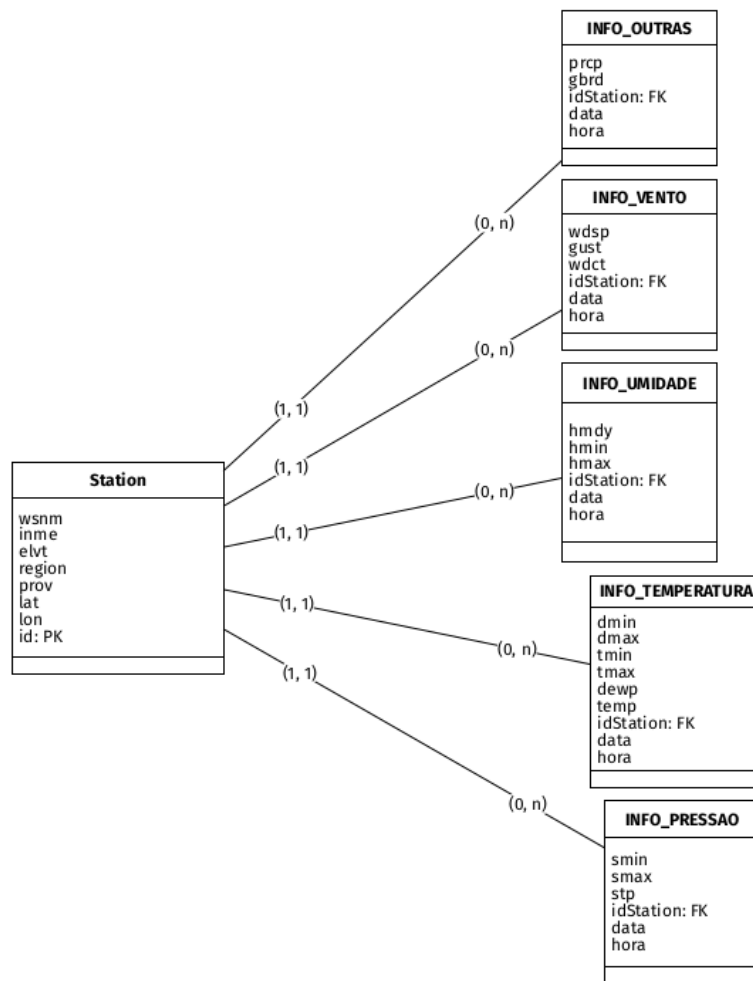
Depois do processo de normalização foi visto que o banco ainda estava grande assim reduzindo os dados para conter apenas entre os intervalos das datas de 2018 a 2021, otimizando o banco mas ainda contendo dados relevantes.

Foi criado índices para otimizar algumas consultas, mesmo com eles algumas não usaram e foi mantido o mesmo tempo para retornar os resultados, mas com os índices otimizando e retornando os dados com desempenho melhorado.

## 1.1 Atributos das tabelas

- Date (YYYY-MM-DD)
- Time (HH:00)
- Amount of precipitation in millimetres (last hour)
- Atmospheric pressure at station level (mb)
- Maximum air pressure for the last hour (mb)
- Minimum air pressure for the last hour (mb)
- Solar radiation (KJ/m2)
- Air temperature (instant) (°c)
- Dew point temperature (instant) (°c)
- Maximum temperature for the last hour (°c)
- Minimum temperature for the last hour (°c)
- Maximum dew point temperature for the last hour (°c)
- Minimum dew point temperature for the last hour (°c)
- Maximum relative humid temperature for the last hour (%)
- Minimum relative humid temperature for the last hour (%)
- Relative humid (% instant)
- Wind direction (radius degrees (0-360))
- Wind gust in metres per second
- Wind speed in metres per second
- Brazilian geopolitical regions
- State (Province)
- Station Name (usually city location or nickname)
- Station code (INMET number)
- Latitude
- Longitude
- Elevation

## 2 Modelagem



### 3 Consultas

1. Dado uma latitude e longitude, retorna as 5 estações mais próximas desta coordenada com a sua temperatura máxima registrada.

```
SELECT  s.id, s.station, s.latitude, s.longitude,
        (6371 * ACOS(COS(RADIANS(-5.19812)) *
        COS(RADIANS(s.latitude)) *
        COS(RADIANS(-39.2962) - RADIANS(s.longitude)) +
        SIN(RADIANS(-5.19812)) *
        SIN(RADIANS(s.latitude)))) AS distance,
        (
            SELECT  MAX(i.tmax)
            FROM      station AS st JOIN info_temperatura AS i
                      ON st.id = i.id_station
            WHERE     i.id_station = s.id
        ) AS temperatura
FROM      station AS s
ORDER BY distance ASC
LIMIT 5;
```

2. Dado um estado e um ano, retorna a precipitação máxima deste local no dado ano.

```
SELECT  EXTRACT(MONTH FROM IO.data) AS mes, MAX(IO.prcp)
FROM      station AS ST JOIN info_outras AS IO
          ON ST.id = IO.id_station
WHERE     ST.state = 'CE' AND
          EXTRACT(YEAR FROM IO.data) = 2019
GROUP BY mes
ORDER BY mes;
```

3. Dado um local e uma data, retorna a temperatura mínima e máxima para este local.

```
SELECT  S.state, (
    SELECT  max(tmax)
    FROM      station AS ST JOIN info_temperatura AS IT
              ON ST.id = IT.id_station
    WHERE     IT.data = '2017-12-20' AND
              ST.state = S.state
), (
    SELECT  min(tmin)
    FROM      station AS ST JOIN info_temperatura AS IT
              ON ST.id = IT.id_station
    WHERE     IT.data = '2017-12-20' AND
              ST.state = S.state AND
              tmin <> -9999
)
FROM      station AS S
WHERE     S.state = 'CE'
group by S.state;
```

4. Retorna a temperatura mais alta em cada estado.

```
SELECT ST.state, (
    SELECT MAX(I.tmax)
    FROM station AS S JOIN info_temperatura AS I
        ON S.id = I.id_station
    WHERE ST.state = S.state
) AS temperatura
FROM station AS ST
GROUP BY ST.state
ORDER BY temperatura;
```

5. Dado um estado, ano e mês, retorna a máxima radiação solar registrada por uma estação neste dado ano e mês.

```
SELECT EXTRACT(MONTH FROM IO.data) AS mes, MAX(IO.gbrd)
FROM station AS ST JOIN info_outras AS IO
    ON ST.id = IO.id_station
WHERE ST.state = 'BA' AND
    EXTRACT(YEAR FROM IO.data) = 2019
GROUP BY mes
ORDER BY mes;
```

6. Umidade mínima e máxima de um local em um intervalo de datas 2018 a 2023.

```
SELECT S.state, (
    SELECT max(hmax)
    FROM station as st join info_umidade as iu
        on st.id = iu.id_station
    WHERE iu.data between '2018-01-01' and '2023-05-30' and
        st.state = S.state
), (
    SELECT min(hmin)
    FROM station as st join info_umidade as iu
        on st.id = iu.id_station
    WHERE iu.data between '2018-01-01' and '2023-05-30' and
        st.state = S.state and
        hmin <> -9999
)
FROM station as S
WHERE S.state = 'CE'
GROUP BY S.state;
```

7. Listar as cidades no CE com a maior temperatura e temperatura mínima em um intervalo de datas.

```
SELECT  S.station, temp_min.min_tmin, temp_max.max_tmax
FROM    station AS S JOIN (
    SELECT  st.station, MIN(nh.tmin) AS min_tmin
    FROM    station AS st JOIN info_temperatura AS nh
            ON st.id = nh.id_station
    WHERE   nh.data BETWEEN '2021-01-01' AND '2022-12-31' AND
            nh.tmin IS NOT NULL AND
            nh.tmin <> -9999
    GROUP BY st.station
) AS temp_min ON S.station = temp_min.station
JOIN (
    SELECT  st.station, MAX(nh.tmax) AS max_tmax
    FROM    station AS st JOIN info_temperatura AS nh
            ON st.id = nh.id_station
    WHERE   nh.data BETWEEN '2021-01-01' AND '2022-12-31' AND
            nh.tmax IS NOT NULL AND
            nh.tmax <> -9999
    GROUP BY st.station
) AS temp_max ON S.station = temp_max.station
WHERE    S.state = 'CE' AND
        temp_min.min_tmin IS NOT NULL AND
        temp_max.max_tmax IS NOT NULL;
```

8. Listar as cidades com valor elevação ordenado em ordem decrescente com sua temperatura max e min.

```
SELECT  st.station, st.height, max(nh.tmax), min(nh.tmin), nh.data
FROM    station as st join info_temperatura as nh
        on st.id = nh.id_station
WHERE   st.state = 'CE' and
        nh.data between '2021-01-01' and '2022-12-31' and
        nh.tmin <> -9999
GROUP BY st.station, st.height, nh.data
ORDER BY st.height DESC;
```

9. Listar as cidades no CE que tiveram a temperatura máxima de 35 graus ou mais e quantas vezes aconteceu.

```
SELECT  st.station, COUNT(st.station)
FROM    station as st join info_temperatura as nh
        on st.id = nh.id_station
WHERE   st.state = 'CE' and
        nh.tmax > 35
GROUP BY st.station;
```

10. Listar as cidades no CE que tiveram a maior velocidade de ventos e a data que aconteceu.

```
SELECT  st.station, max(nh.wdsp), nh.data
FROM    station as st join info_vento as nh
        ON st.id = nh.id_station
WHERE   st.state = 'CE'
GROUP BY nh.wdsp, st.station, nh.data
ORDER BY nh.wdsp DESC;
```

## 4 Metadados

1. Quantidade de tabelas do banco de dados.

	quantidade_tabelas bigint
1	7

2. Quantidade de atributos de cada tabela.

	table_name name	quantidade_atributos bigint
1	info_vento	6
2	info_temperatura	8
3	info_outras	5
4	columns_description	5
5	info_pressao	6
6	station	9
7	info_umidade	6



3. Tamanho de cada tabela em gigabytes.

	table_name name	tamanho_tabela text
1	info_umidade	1.24GB
2	station	112.00KB
3	info_vento	1.01GB
4	info_outras	1.45GB
5	info_temperatura	2.59GB
6	info_pressao	1.37GB
7	columns_description	16.00KB

4. Quantidade de acessos sequenciais realizada em cada tabela.

	table_name name	quantidade_acessos_sequenciais bigint
1	station	2088
2	info_temperatura	1399
3	info_outras	500
4	info_umidade	12
5	columns_description	8
6	info_pressao	3
7	info_vento	3

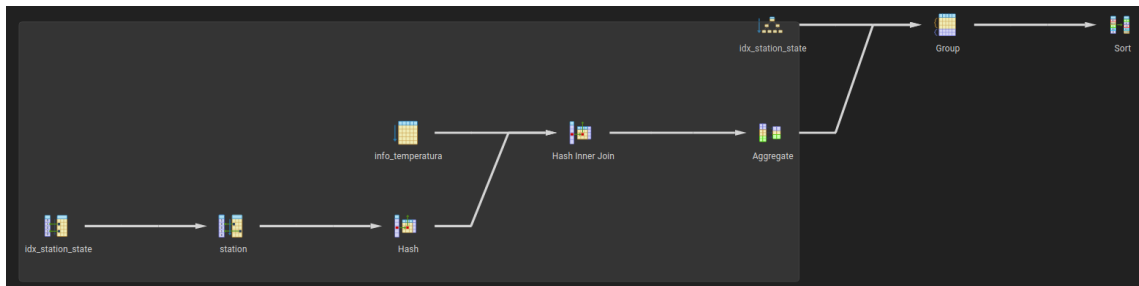
## 5 Avaliação das consultas

### 5.1 Tempo sem uso índices

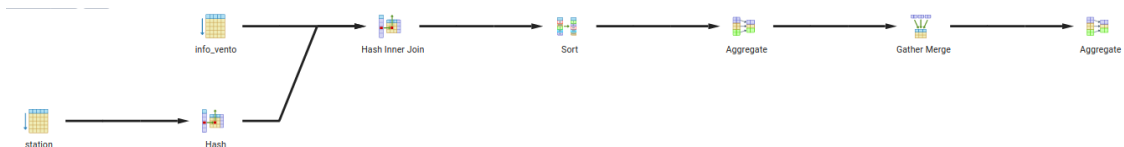
Consulta	Tempo
1	5,03462 sec
2	7,39830 sec
3	5,14162 sec
4	19,8563 min
5	8,69090 sec
6	3,50407 sec
7	2,93432 sec
8	1,52255 sec
9	0,45107 sec
10	19,2731 sec

#### 5.1.1 Consultas mais demoradas

A consulta mais demorada foi a consulta 4, tendo como tempo de execução 19,8563 min, isso se deve a grande quantidade de loops feitos na tabela info\_temperatura (27, um para cada estado e o DF), usada na subconsulta, como segue a imagem do seu esquema de execução:



A segunda consulta com mais tempo de execução foi a consulta 10, com o tempo de execução de 19,2731 sec. Esse tempo de execução é causado pela quantidade de loops nas tabelas info\_vento e station, sendo necessário 3 loops em cada para retornar o resultado da consulta, como mostrado no esquema de execução abaixo:



### 5.1.2 Consultas menos demoradas

A consulta menos demorada foi a consulta 9, tendo como tempo de execução 0,45107 sec, isso se deve a quantidade de atributos no GROUP BY fazendo com que reste poucas tuplas na tabela station, tornando menor a quantidade de tuplas a serem computadas. A segunda consulta com menos tempo de execução foi a consulta 8, tendo como tempo de execução 1,52255 sec, isso se deve as condições dadas no WHERE, que são muito seletivas, retornando poucas tuplas.

## 5.2 Criação dos índices

1. Índice criado para a consulta 1:

```
CREATE INDEX idx_info_temperatura_id_station  
ON info_temperatura (id_station);
```

2. Índices criados para a consulta 2:

```
CREATE INDEX idx_info_outras_id_station  
ON info_outras (id_station);  
CREATE INDEX idx_station_state  
ON station (state);  
CREATE INDEX idx_info_outras_data  
ON info_outras (data);
```

3. Índices criados para a consulta 3:

```
CREATE INDEX idx_info_temperatura_data  
ON info_temperatura (data);  
CREATE INDEX idx_info_temperatura_tmin  
ON info_temperatura (tmin);  
CREATE INDEX idx_info_temperatura_tmax  
ON info_temperatura (tmax);
```

Além de usar os índices criados para a consulta 1 e 2

```
idx_info_temperatura_id_station  
idx_station_state
```

4. Índices criados para a consulta 4: Faz uso dos índices criados para a consultas 1 e 2

```
idx_station_state  
idx_info_temperatura_id_station
```

5. Índices criados para a consulta 5:

```
CREATE INDEX idx_info_outras_gbrd  
ON info_outras (gbrd);
```

Além de usar os índices criados para a consulta 1 e 2

```
idx_station_state  
idx_info_outras_id_station  
idx_info_outras_data
```

6. Índices criados para a consulta 6:

```
CREATE INDEX idx_info_umidade_data
ON info_umidade (data);
CREATE INDEX idx_info_umidade_hmin
ON info_umidade (hmin);
CREATE INDEX idx_info_umidade_hmax
ON info_umidade (hmax);
```

Além de usar o índice criado para a consulta 2

`idx_station_state`

7. Índice(s) usado(s) para a consulta 7:

`idx_station_state`

8. Índice(s) usado(s) para a consulta 8: Foi usado o índice da consulta 3

`idx_station_state`

9. Índice(s) usado(s) para a consulta 9:

```
id_station_state
idx_info_temperatura_id_station
idx_info_temperatura_tmax
```

10. Índice(s) usado(s) para a consulta 10:

`idx_station_state`

### 5.3 Tempo com uso índices

Consulta	Tempo	Usou o índice
1	0,82949 sec	sim
2	5,88063 sec	sim
3	0,01544 sec	sim
4	35.0320 sec	sim
5	7,00151 sec	sim
6	3,33569 sec	sim
7	4,82780 sec	sim
8	0,50100 sec	sim
9	0,33200 sec	sim
10	2,72200 sec	sim

## 6 Índice clusterizado

Sugeriria um índice clusterizado para a consulta 4. Criaria esse índice para que ela calcule o MAX(I.tmax) de forma mais rápida, já que a tabela estaria clusterizada com base neste atributo.

Índice criado para a consulta 4:

```
CREATE INDEX idx_cluster_tmax  
ON public.info_temperatura USING btree  
(tmax DESC NULLS LAST);
```

```
ALTER TABLE IF EXISTS public.info_temperatura  
CLUSTER ON idx_cluster_tmax;
```

Tempo de execução com o índice: 38.4748 sec.

A consulta não fez o uso do índice, os índices utilizados foram os mesmos utilizados antes da criação do índice clusterizado.

