

SQL para análise de dados



TÉCNICAS AVANÇADAS



GUIA DA AULA 3



Faça agregação por particionamento - Teoria



Introdução



Acompanhe aqui
os temas que
serão tratados
na videoaula



Introdução

Esse é um aspecto do AWS Athena utilizado para organizar e gerar as queries de maneira mais eficiente no *framework*. É uma organização hierárquica onde cada pasta contém **subpastas** com o rótulo e valores. É utilizado para economizar dados carregados no AWS Athena, aumentando a performance e reduzindo custos. Para fazer isso, siga as etapas:

- No S3, crie uma pasta no AWS com o nome do seu **dataset**.
- Vamos supor que queremos separar as lojas na nossa partição. Para isso criamos subpastas:
 - transacoes_partition/id_loja=magalu
 - transacoes_partition/id_loja=giraffas
 - transacoes_partition/id_loja=postoshell



- transacoes_partition/id_loja=subway
- transacoes_partition/id_loja=seveneleven
- transacoes_partition/id_loja=extra
- transacoes_partition/id_loja=shopee

Dentro de cada uma das subpastas, colocamos apenas aquelas informações referentes a id_loja dedicada. A geração da partição é indicada na hora da **CRIAÇÃO** da tabela com o comando **PARTITIONED by** id_loja (no exemplo).



Depois da criação, é necessário carregar as partições como o comando:

```
CREATE EXTERNAL TABLE transacoes_part(  
    id_cliente BIGINT,  
    id_transacoes BIGINT,  
    valor DOUBLE)  
PARTITIONED BY (id_loja string)  
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'  
WITH SERDEPROPERTIES (  
    'serialization.format' = ',',  
    'field.delim' = ',',  
    )  
LOCATION 's3://transacoes-partition/'  
MSCK REPAIR TABLE transacoes_part
```



Você pode verificar pela contagem de linhas na tabela completa:

```
select count(*) from transacoes_part
```

A partir disso, nós podemos seguir com os comandos de SELECT que aprendemos.

