databricks aggreg_gold_data

(https://databricks.com)

breweries_aggregated - table load

```
# Importar funções necessárias
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
```

```
# Inicializar Spark Session
spark = SparkSession.builder.appName("BreweryETL").getOrCreate()
```

```
# Carregar dados da camada Silver
silver_path = "/dbfs/tmp/silver/"
silver_df = spark.read.parquet(silver_path)
```

```
# Agregar dados
gold_df = silver_df.groupBy("state", "city", "brewery_type").count()
# Caminho para salvar os dados da camada Gold como tabela Delta
gold_path = "/dbfs/tmp/gold/breweries_aggregated"
```

```
# Salvar dados agregados como tabela Delta
gold_df.write.format("delta").mode("overwrite").save(gold_path)
```

```
# Registrar a tabela Delta no catálogo de metadados
spark.sql(f"""
    CREATE TABLE IF NOT EXISTS breweries_aggregated
    USING DELTA
    LOCATION '{gold_path}'
""")
```

```
Out[19]: DataFrame[]
```

```
# Exibir os dados para garantir que foram carregados corretamente
display(gold_df)
```

Table

	state	city	brewery_type	count
1	Nevada	Reno	micro	1
2	Massachusetts	Abington	micro	1
3	Arizona	Mesa	micro	1
4	Idaho	Boise	large	1
5	Oregon	Bend	large	3
6	Colorado	Denver	large	1

7	California	Westlake Villa...	micro	1
8	Colorado	Denver	proprietor	1
9	Oregon	Portland	brewpub	1
10	Arizona	Gilbert	micro	1
11	Texas	Houston	micro	1
12	California	Petaluma	closed	1
13	New York	Williamsville	brewpub	1
14	Indiana	Knox	micro	1
15	California	Mariposa	micro	1

48 rows

Validations

```
%sql
select * from breweries_aggregated
order by count desc
limit 10;
```

Table

	state	city	brewery_type	count
1	Oregon	Bend	large	3
2	Nevada	Reno	micro	1
3	Massachusetts	Abington	micro	1
4	Arizona	Mesa	micro	1
5	Idaho	Boise	large	1
6	Colorado	Denver	large	1
7	California	Westlake Villa...	micro	1
8	Colorado	Denver	proprietor	1
9	Oregon	Portland	brewpub	1
10	Arizona	Gilbert	micro	1

10 rows

Table

	brewery_type	count
1	micro	30
2	brewpub	9
3	large	5
4	closed	2
5	proprietor	1
6	contract	1

6 rows