1. Características

- Formato de arquivo columnar (colunar), voltado para armazenamento de dados analíticos.
- Projetado para trabalhar com grandes volumes de dados em sistemas distribuídos como Hadoop, Spark, Hive, Presto, etc.
- Suporta compressão eficiente e leitura seletiva de colunas (não carrega tudo de uma vez).
- É um formato aberto, parte do ecossistema Apache.
- Otimizado para uso com linguagens como Python (Pandas), R, Scala e ferramentas de Big Data.

2. Vantagens e Desvantagens

Vantagens:

- Leitura rápida de colunas específicas: ideal para análises que não precisam do dataset todo.
- Compressão de dados muito eficiente, reduzindo uso de disco e tempo de leitura.
- Integração com ferramentas de Big Data (Spark, AWS Athena, Google BigQuery, etc).
- Suporta tipos de dados complexos (listas, mapas, estruturas aninhadas).

Desvantagens:

- Não é um banco de dados tradicional: é um formato de arquivo - não tem transações, autenticação, etc.
- Não é ideal para leitura linha a linha ou para operações OLTP (transacionais).
- Curva de aprendizado para quem nunca trabalhou com formatos columnar ou ferramentas de Big Data.

3. Caso de Uso

- Data Lakes: usado para armazenar grandes volumes de dados de forma otimizada (em S3, HDFS, etc).
- ETL e pipelines de dados: muito usado com Apache Spark para transformar dados.
- Plataformas de análise como AWS Athena e Google BigQuery leem diretamente arquivos Parquet, sem precisar importar para um banco.
- Machine Learning: datasets em Parquet são leves e rápidos de carregar em frameworks como Pandas e PyTorch.

4. Comparativo de Desempenho (Amazon S3)

Formato	Tamanho no S3	Tempo de Consulta	Dados Lidos	Custo Estimado
CSV	1 TB	236 segundos	1.15 TB	$5.75

Parquet	130 GB	6.78 segundos	2.51 GB	$0.01

Resultado da Comparação:

- 87% menos espaço em disco com Parquet
- Consulta 34x mais rápida
- 99% menos dados lidos
- 99.7% de economia no custo de consulta

- O Databricks é baseado em Apache Spark, que usa Parquet como formato padrão para armazenar e consultar dados.
- Parquet é essencial em ambientes com Data Lakes e Delta Lake, acelerando consultas e reduzindo custo em nuvem (como S3).
- Ferramentas como AWS Athena, BigQuery, e Azure Synapse também leem Parquet diretamente - é amplamente compatível.