**✅ 1. What is a Parquet File?**

* **Parquet** is a **columnar** storage file format.
* Developed by **Apache**, optimized for big data processing tools like **Apache Spark**, **Hive**, **Hadoop**, and **Databricks**.
* Files have a **.parquet** extension.

**📌 Features:**

* **Columnar format:** Reads only the needed columns (not the full row).
* **Highly compressed**: Smaller size compared to CSV/JSON.
* **Supports schema**: Strong data typing and structure.
* **Efficient for analytics & queries.**

**✅ 2. Why Use Parquet?**

| **Benefit** | **Explanation** |
| --- | --- |
| 💡 Compression | Reduces file size drastically. |
| ⚡ Speed | Faster read/write due to column-wise storage. |
| 🧠 Schema Support | Stores data with its schema (data types). |
| 📊 Big Data Ready | Works great with Spark, Hive, Presto, AWS Athena, etc. |
| 🎯 Selective Reading | Can read specific columns only → boosts performance. |

**✅ 3. Difference Between CSV, JSON, and Parquet**

| **Feature** | **CSV** | **JSON** | **Parquet** |
| --- | --- | --- | --- |
| **Format Type** | Row-based | Semi-structured (hierarchical) | Columnar |
| **Compression** | ❌ No compression | ⚠️ Low compression | ✅ High compression |
| **Read Speed** | ❌ Slower for big data | ⚠️ Moderate | ✅ Fast (column pruning) |
| **Supports Schema** | ❌ No | ⚠️ Partially | ✅ Yes |
| **Data Size** | 🔺 Larger | 🔺 Medium | 🔻 Smaller |
| **Use Case** | Simple tabular data | Nested/complex data | Big data, analytics, Spark |
| **Tools** | Excel, Pandas | APIs, NoSQL | Spark, Hadoop, AWS, etc. |

**✅ 4. What is a Delta Table?**

* **Delta Table** is a **transactional storage layer** on top of **Parquet files**.
* Developed by **Databricks** and used in **Delta Lake** (Apache project).
* Uses **ACID transactions**, **versioning**, and **schema enforcement**.

**📌 Key Features:**

| **Feature** | **Description** |
| --- | --- |
| ✅ **ACID Transactions** | Ensures data reliability and integrity. |
| 🔁 **Time Travel** | You can query old versions of the table. |
| 🔄 **Upserts/Merges** | Can update and insert at once. |
| 🔐 **Schema Enforcement** | Rejects bad data that violates schema. |
| 🔍 **Efficient Reads** | Optimized reads using parquet & metadata. |