step 1 & 2: Scenario Analysis and Decision Table

Scenario	Selected Format	Primary Justification	Read/Write Pattern	Schema	Interoperability	Trade-offs
1. Streaming Clickstream	**JSON**	Fast writes & schema flexibility.	Low-latency writes; Exploratory reads.	**Evolving:** Handles new events easily.	High (Universal).	Large file size; slow for heavy aggregations.
2. Daily Sales Analytics	**Parquet**	Columnar pruning for 50+ columns.	Heavy Batch writes; Analytical reads.	**Stable:** Fixed daily schema.	Medium (Big Data tools).	Binary format; not human-readable.
3. External Exchange	**CSV**	Universal support & simple tools.	Moderate volume; Full row reads.	**Stable:** Well-defined.	**High (Universal).**	Fragile; no nested data; larger size.
4. Ad-Hoc Exploration	**JSON**	Self-describing & easy to inspect.	Irregular; Full row inspection.	**Flexible:** Discovering structure.	High (Developer friendly).	Verbose; repeating keys waste space.

5. Long-Term Archive	**ORC**	Superior compression & metadata.	Rare reads; Append-only.	**Stable:** Preservation focus.	Low (Hadoop/Hive focus).	Specialized; difficult to read outside Hadoop.

Step 3: Detailed Justifications

Scenario 1: Streaming Clickstream Ingestion

- **Selected Format: JSON**
- **Justification:** In a streaming context, the system cannot afford the CPU overhead of "building" a columnar Parquet file every few seconds. JSON allows the application to dump events as-is. Since clickstream data evolves (new buttons, new tags), JSON's self-describing nature ensures the pipeline doesn't break when the schema changes.
- **Performance:** Optimized for **Write Throughput**.
- **Trade-off:** You will likely need a downstream "compaction" job to convert these JSONs into Parquet for long-term analytics.

Scenario 2: Daily Sales Analytics

- **Selected Format: Parquet**
- **Justification:** With billions of rows and 50+ columns, reading a CSV or JSON would be a disaster for cost and speed. Parquet's **Column Pruning** allows the engine to ignore the 40+ columns not needed for the query. Its internal statistics (Min/Max) allow for **Predicate Pushdown**, skipping massive chunks of data.
- **Performance:** Optimized for **Read/Analytical Speed**.
- **Trade-off:** Requires a schema definition; you cannot simply "append" a column without updating the metadata.

Scenario 3: Data Exchange with External Partners

- **Selected Format: CSV**
- **Justification:** You cannot assume your partner has a Spark cluster or Parquet viewer. Every system on Earth (Excel, SQL, Python, Notepad) can read a CSV. Since the schema is stable and volume is moderate, the lack of compression is an acceptable price for **Interoperability**.
- **Trade-off:** Risk of data corruption if text fields contain commas or newlines.

Scenario 4: Ad-Hoc Analyst Exploration

- **Selected Format: JSON**
- **Justification:** When data is "unstructured" or new, an analyst needs to see the keys and values together. JSON allows them to explore nested arrays (like "items in a cart") that

are difficult to flatten into CSVs or visualize in binary Parquet without pre-defining a schema.
- **Trade-off:** High memory usage for large files during exploration.

Scenario 5: Long-Term Archival Storage

- **Selected Format: ORC (Optimized Row Columnar)**
- **Justification:** While Parquet and ORC are similar, ORC often provides slightly better compression ratios for Hadoop-native environments. Since storage cost is the #1 priority, ORC's ability to compress data into tiny footprints while maintaining analytical headers (for the occasional query) makes it the winner for "Cold Storage."
- **Trade-off:** Less "universal" than Parquet in the modern cloud (Snowflake/BigQuery prefer Parquet).

Step 4: Decision Framework Summary

When choosing a format, follow this priority logic:

1. **Is human readability or universal compatibility required?**
    - Yes $\rightarrow$ **CSV** (Simple/Stable) or **JSON** (Complex/Nested).
2. **Is this for high-performance analytics on large data?**
    - Yes $\rightarrow$ **Parquet** (General Cloud) or **ORC** (Hadoop Ecosystem).
3. **Is the data streaming and the schema constantly changing?**
    - Yes $\rightarrow$ **JSON** or **Avro** (for binary streaming).
4. **Is storage cost the primary concern for a stable dataset?**
    - Yes $\rightarrow$ **Parquet/ORC** with high compression (Zstd/Gzip).