

Cecha	Delta Lake	Apache Iceberg
Wspierany przez Databricks	Oficjalnie rozwijany i natywnie wspierany	Działa, ale z ograniczeniami
Optymalizacja zapytań	Bardzo dobra (Z-Ordering, OPTIMIZE)	Dobra, zależna od silnika (Spark/Trino)
Format plików	Parquet + metadane Delta	Parquet/Avro/ORC + metadane Iceberg
ACID Transactions	Tak	Tak
Time Travel / Versioning	Tak	Tak
Schema Evolution	Tak (pełna obsługa)	Tak (lepszą obsługą niż Delta Lake)
Silniki obsługujące format	Spark, Databricks, Synapse	Spark, Flink, Trino, Hive, Presto
Obsługa przez inne platformy	Słabsza poza Databricks	Lepsza kompatybilność
Kompleksowość zarządzania	Mniej konfiguracji (out-of-the-box)	Więcej opcji – ale większa złożoność

Delta Lake, gdy:

- Korzystasz głównie z Databricks jako silnika obliczeniowego.
- Potrzebujesz łatwego wdrożenia i zarządzania bez zaawansowanej konfiguracji.
- Chcesz mieć dostęp do funkcji typu OPTIMIZE, Z-Ordering, MERGE INTO, UPDATE, DELETE.
- Masz przewidywalne, sekwencyjne przetwarzanie danych (np. batch/ETL).
- Wspierasz data lakehouse typowy dla środowisk Azure.

Idealny wybór dla projektów zbudowanych w 100% na Azure + Databricks.

Apache Iceberg, gdy:

- Planujesz multi-engine/multi-cloud (np. Spark + Flink + Trino).
- Potrzebujesz lepszej obsługi schematów w czasie rzeczywistym (np. zmiana nazw kolumn, typów).
- Oczekujesz bardziej elastycznego zarządzania plikami i metadanymi.
- Priorytetem jest integracja z innymi narzędziami open-source.
- Twoje dane są często modyfikowane w czasie rzeczywistym (streaming + batch).

Lepsze dla dużych, rozproszonych architektur z różnorodnymi silnikami (np. Flink + Trino + Spark)