

Typ : Podtyp	Opis i główne cechy	Najczęstsze zastosowania	Przykładowe silniki	Udział w rynku
Relacyjne (RDBMS)	- Dane przechowywane w tabelach (wiersze i kolumny). - Schematy z góry zdefiniowane (sztywne modele). - Wsparcie dla ACID (Atomicity, Consistency, Isolation, Durability). - Posługują się językiem SQL.	- Systemy transakcyjne (bankowość, księgowość). - Aplikacje biznesowe (ERP, CRM). - Analizy i raportowanie w oparciu o modele relacyjne (OLAP).	Oracle, MySQL, PostgreSQL, Microsoft SQL Server	60–70%
NoSQL: Dokumentowe	- Dane w formacie dokumentów (np. JSON, BSON, XML). - Brak sztywnego schematu, możliwość przechowywania obiektów o zróżnicowanej strukturze. - Dedykowane języki zapytań (np. MQL w MongoDB) lub zbliżone do SQL.	- Aplikacje webowe i mobilne (dynamiczne struktury danych). - Systemy CMS. - Logi i dane pół-strukturyzowane.	MongoDB, CouchDB, Couchbase	10–15%
NoSQL: Klucz–wartość	- Dane w postaci par klucz → wartość. - Najprostszy model w rodzinie NoSQL. - Przeznaczone do szybkiego odczytu/zapisu w rozproszonych środowiskach.	- Przechowywanie sesji i cache. - Tymczasowe przechowywanie konfiguracji. - Proste dane klucz–definicja (np. słowniki, parametry).	Redis, Riak, Amazon DynamoDB	5–10%
NoSQL: Kolumnowe	- Dane przechowywane kolumnowo, co pozwala na wydajną kompresję i szybkie agregacje. - Przeznaczone do ogromnych zbiorów danych (Big Data). - Zazwyczaj działają w klastrach rozproszonych.	- Hurtownie danych. - Analiza wielkich zbiorów (Big Data). - Szybkie operacje agregujące (SUM, COUNT, AVG) na setkach milionów rekordów.	Apache Cassandra, HBase, Google Bigtable, ClickHouse	5–10%
NoSQL: Grafowe	- Dane w postaci węzłów (nodes) i krawędzi (edges) z atrybutami. - Świetnie odwzorowują relacje i powiązania między obiektami. - Dedykowane zapytania grafowe (Cypher w Neo4j, Gremlin).	- Analiza sieci społecznościowych. - Systemy rekomendacyjne (powiązania, zależności). - Modelowanie i analiza złożonych relacji (np. w logistyce, sieciach transportu).	Neo4j, OrientDB, ArangoDB	1–5%
Obiektowe	- Dane przechowywane jako obiekty (z atrybutami i metodami), zgodnie z paradygmatem OOP. - Eliminacja tzw. „impedance mismatch” między kodem obiektywnym a bazą. - Integracja z językami obiektowymi (Java, C++, C#).	- Systemy, w których bezpośrednio odwzorowuje się strukturę obiektową. - Aplikacje wymagające elastycznego przechowywania złożonych klas (zagnieżdżenia obiektów).	ObjectDB, db4o	1–2%
Wektorowe	- Specjalne bazy do przechowywania i wyszukiwania wektorów (embeddings). - Kluczowe w zastosowaniach AI, ML i wyszukiwaniu semantycznym. - Oferują m.in. nearest neighbor search w wysokowymiarowych przestrzeniach.	- Systemy rekomendacyjne. - Wyszukiwanie semantyczne (teksty, obrazy, dźwięki). - Analiza podobieństwa w przestrzeniach wielowymiarowych (klasyfikacja, klastrowanie).	Milvus, Qdrant, Pinecone	1–3%
Hierarchiczne	- Jeden z najstarszych modeli; dane w formie drzewa (hierarchii). - Bardziej sztywna struktura niż modele relacyjne czy NoSQL. - Głównie w starszych systemach mainframe.	- Starsze systemy bankowe, ubezpieczeniowe (mainframe). - Aplikacje korzystające z ustalonych, niezmiennych struktur danych.	IBM IMS	<1%
Sieciowe	- Uogólnienie modelu hierarchicznego, gdzie każdy węzeł może łączyć się z wieloma innymi. - Prototyp baz grafowych, ale mniej elastyczny i oparty na starszych standardach (CODASYL). - Spotykany w starszych systemach mainframe.	- Dziedzictwo starszych aplikacji finansowych, kadrowych. - Historyczne systemy w dużych przedsiębiorstwach (często nadal utrzymywane z powodów biznesowych).	IDMS (Integrated Database Management System)	<1%