

Multi-model Database 是什麼？

編撰：屏東大學 周國華老師 2026-03-04

簡單來說，**Multi-model DB**（多模態資料庫）就是資料庫界的「瑞士刀」。在過去，如果你有不同的資料需求，通常需要針對每一種資料類型建立一個獨立的資料庫（例如：用 MySQL 存關聯式資料、用 MongoDB 存文件資料、用 Neo4j 存圖譜）。而 **Multi-model DB** 則是將這些不同的資料模型整合進同一個後端引擎中。

1. 核心概念：一站式處理

多模態資料庫允許你在**單一查詢接口**或單一系統中，同時操作多種不同的資料模型。常見支援的模型包括：

- **關聯式 (Relational)**: 傳統的表格與 Row/Column。
- **文件 (Document)**: 如 JSON 格式的非結構化資料。
- **圖形 (Graph)**: 處理實體之間複雜的關聯網路。
- **鍵值對 (Key-Value)**: 簡單的高速快取結構。
- **時間序列 (Time-series)**: 追蹤隨時間變化的數據（如 IoT 感測器）。

2. 為什麼需要它？（解決「聚合式持久化」問題）

在沒有多模態資料庫之前，開發者常面臨 **Polyglot Persistence**（聚合式持久化）的困擾：為了不同的功能，你必須維護好幾套資料庫系統。

傳統做法 (Polyglot)	多模態做法 (Multi-model)
複雜度高：需學習多種查詢語言 (SQL, Cypher, Gremlin)。	一致性：通常使用統一的查詢語法或 API。
資料冗餘：資料常需在不同 DB 間搬運 (ETL)。	零 ETL：資料存一份，換個視角（模型）就能查。
維護成本：需部署、更新、備份多個系統。	降低成本：簡化維護成本與架構複雜度。

3. 常見的 Multi-model 資料庫實例

現在市面上許多主流資料庫都已經走向多模態化：

- **Azure Cosmos DB:** 支援文件、圖形、Key-value 等。
- **ArangoDB:** 原生設計就是為了整合圖形、文件與搜尋。
- **PostgreSQL:** 雖然是關聯式起家，但透過 JSONB、PostGIS、以及各式 Extension，它現在也是非常強大的多模態資料庫。
- **Redis:** 除了 Key-value，也支援 JSON、Graph 和時序資料。

4. 總結

多模態資料庫的出現，是為了應對現代應用程式資料結構越來越混雜的情況。它讓你不必在「為了效能選特定 DB」與「為了管理方便選單一 DB」之間痛苦掙扎。

小提醒： 雖然它很方便，但並非銀彈。對於極端極致的效能需求（例如超大規模的圖運算），專門的圖資料庫 (Native Graph) 可能還是會比多模態 DB 的圖模式略勝一籌。