

DB 欄位命名與比較

資料庫欄位命名的選擇對於系統的可讀性、一致性和維護性有著重要的影響。不同的資料庫可能有不同的命名慣例，以下是常見的命名方式及其在各種資料庫中的應用。

以下是常見命名方式與對應資料庫的整理表格：

命名方式	命名特徵	應用資料庫
駝峰式命名法	單字首字母小寫，後續單字首字母大寫，如： <code>firstName</code>	MongoDB 、 Cassandra ：與程式碼結合較緊密的 NoSQL 資料庫常使用
蛇形命名法	單字之間用底線 <code>_</code> 分隔，全小寫，如： <code>first_name</code>	PostgreSQL 、 MySQL 、 Oracle 、 SQLite ：傳統 SQL 資料庫中最常見
帕斯卡命名法	每個單字首字母大寫，沒有分隔符號，如： <code>FirstName</code>	Microsoft SQL Server 、 Entity Framework ：與程式碼結合時偶爾使用
中線命名法	單字之間用連字符 <code>-</code> 分隔，全小寫，如： <code>first-name</code>	Elasticsearch ：主要用於索引和類型，但不常用於欄位
全小寫命名法	全部字母小寫，無分隔符號，如： <code>firstname</code>	MySQL 、 SQLite ：為了避免大小寫區分問題，常用於輕量或無區分大小寫的系統
前綴或後綴命名法	欄位名稱加上描述性前綴或後綴，如： <code>user_id</code> 、 <code>created_at</code>	PostgreSQL 、 MySQL 、 SQL Server 、 MongoDB ：常用於表示主鍵、外鍵或時間戳等特定欄位

此表格總結了各種命名方式的特徵和常用的資料庫，選擇命名方式時，應根據資料庫特性和開發需求來做決定，並保持專案中的一致性。

常見欄位命名方式及應用

1. 駝峰式命名法（Camel Case）

- 命名法**：單字首字母小寫，接下來每個單字的首字母大寫。
- 範例**：`firstName`、`lastName`、`orderDate`
- 應用資料庫**：
 - MongoDB**：作為 NoSQL 資料庫，MongoDB 使用 JSON/BSON 結構，開發者常使用駝峰式命名與程式碼一致。
 - Cassandra**：由於 Cassandra 支援靈活的資料模式，駝峰式命名法也被頻繁採用，尤其在與程式碼整合時。

2. 蛇形命名法（Snake Case）

- 命名法**：單字之間用底線 `_` 分隔，所有字母小寫。
- 範例**：`first_name`、`last_name`、`order_date`
- 應用資料庫**：
 - PostgreSQL**：最常見的命名方式，與 SQL 語法兼容性好，避免大小寫問題。
 - MySQL**：由於 MySQL 預設不區分大小寫，蛇形命名法常被使用以保持命名的一致性和可讀性。
 - Oracle**：儘管 Oracle 支援大小寫區分，蛇形命名法依然流行。
 - SQLite**：輕量級資料庫中也常見此命名方式，方便管理。

3. 帕斯卡命名法（Pascal Case）

- 命名法**：每個單字首字母大寫，沒有分隔符號。
- 範例**：`FirstName`、`LastName`、`OrderDate`
- 應用資料庫**：
 - Microsoft SQL Server**：部分開發者，特別是使用 Microsoft 技術堆疊的，會選擇帕斯卡命名法以與程式命名風格保持一致。
 - Entity Framework**：當與 SQL Server 或其他資料庫結合時，帕斯卡命名法常用於映射資料庫欄位和程式屬性名稱。

4. 中線命名法（Kebab Case）

- 命名法**：單字之間用連字符 `-` 分隔，所有字母小寫。
- 範例**：`first-name`、`last-name`、`order-date`
- 應用資料庫**：
 - Elasticsearch**：儘管大部分資料庫不支援欄位中使用連字符，中線命名偶爾會出現在 Elasticsearch 的索引或類型中，但在欄位名稱中較少見。

5. 全小寫命名法

- 命名法**：所有字母均小寫，無任何分隔符號。
- 範例**：`firstname`、`lastname`、`orderdate`
- 應用資料庫**：
 - MySQL**：由於 MySQL 不區分大小寫，許多開發者會選擇全小寫命名來避免混淆。
 - SQLite**：小型應用中，開發者常使用全小寫命名來簡化欄位管理。

6. 前綴或後綴命名法

- **命名法**：在欄位名稱前或後添加描述性前綴或後綴。
- **範例**：`user_id`、`created_at`
- **應用資料庫**：
 - **PostgreSQL、MySQL、SQL Server**：此命名法常見於資料表中，特別是主鍵、外鍵或時間戳等欄位，如 `id`、`created_at`。
 - **MongoDB**：在 NoSQL 環境中，這種命名法也被用來標記欄位的特定用途，便於區分不同類型的資料。

小結

- **SQL 資料庫 (PostgreSQL、MySQL、SQL Server、Oracle)**：推薦使用 **蛇形命名法**，這是一種最常見的標準化命名方式，能在 SQL 查詢中避免大小寫問題，並提高可讀性。
- **NoSQL 資料庫 (MongoDB、Cassandra、Elasticsearch)**：**駝峰式命名法** 和 **全小寫命名法** 較為常見，這些資料庫的靈活數據模型和與 JSON 的緊密結合，使得駝峰式命名更符合程式開發習慣。

選擇合適的命名方式應根據具體資料庫、開發團隊的需求和維護便利性來決定，並保持整個專案中的一致性。