



2023.06.13

MongoDB Developer Workshop

Part I. MongoDB 實用開發技能超展開活動



Silver Su,
Senior Solution Architect, MongoDB APAC

Agenda

1. MongoDB Fundamental
2. Index and Shard Key
3. Schema Design
4. Event-driven Architecture Design

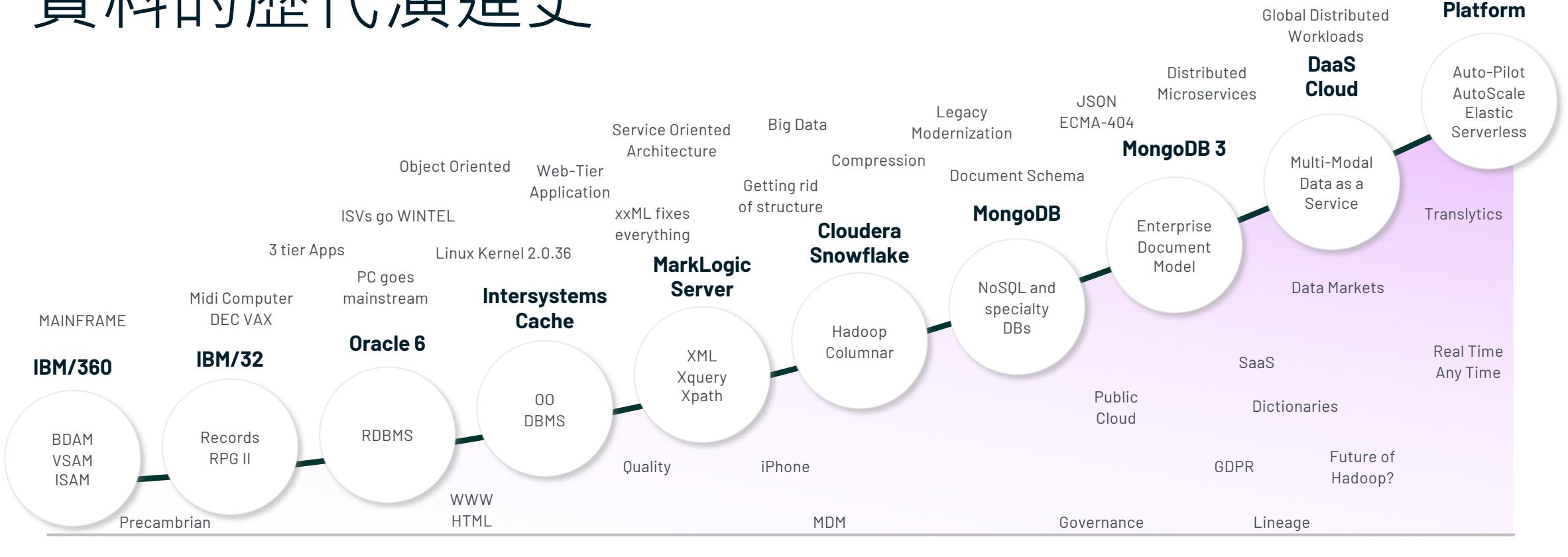


{ 1: “MongoDB Fundamental” }





資料的歷代演進史



The beginning

- Small amounts of data
- Very Structured use cases
- Storage was expensive and optimization of storage was critical

The days of relational

- Introduction of indexing, Views and Data Types
- Growing Data
- First cases of “unstructured” with the emergence of dot.com

The wild west and big data

- Object-Oriented Data
- XPath & XQuery miss the mark
- Unstructured with Hadoop

Wisdom prevails

- Acknowledgement of benefits of 50 years of RDBMS values
- Unification and Convergence of flat and hierarchical data structures into documents



RDBMS 的思維

On Normalization:

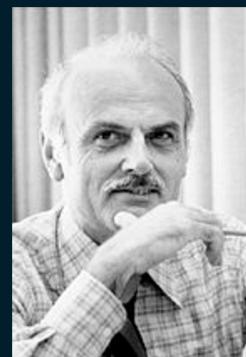
"The simplicity of the array representation which becomes feasible when all relations are cast in normal form is not only an **advantage for storage** purposes but also for **communication of bulk data** between systems which use widely different representations of the data."

On Denormalization:

"If the strong redundancies in the named set are directly reflected in strong redundancies in the stored set (or if other strong redundancies are introduced into the stored set), then, generally speaking, **extra storage space and update time are consumed with a potential drop in query time for some queries and in load on the central processing units.**"

On defining an ERD:

"Ideally, the variety of permitted data representations should be just adequate to cover the spectrum of performance requirements of the total collection of installations. Too great a variety leads to **unnecessary overhead in storage and continual reinterpretation of descriptions** for the structures currently in effect."



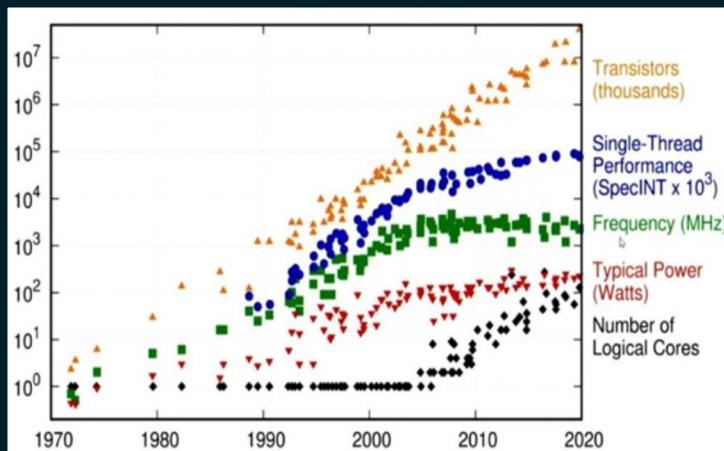
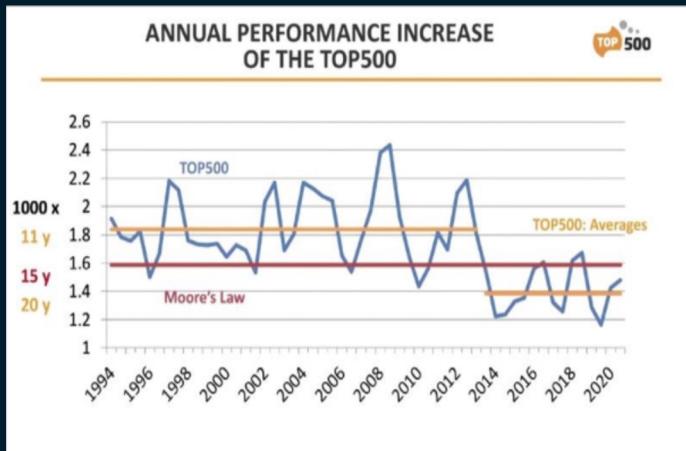
A Relational Model of Data for Large Shared Data Banks

Edgar Frank "Ted" Codd
19 August 1923 – 18 April 2003

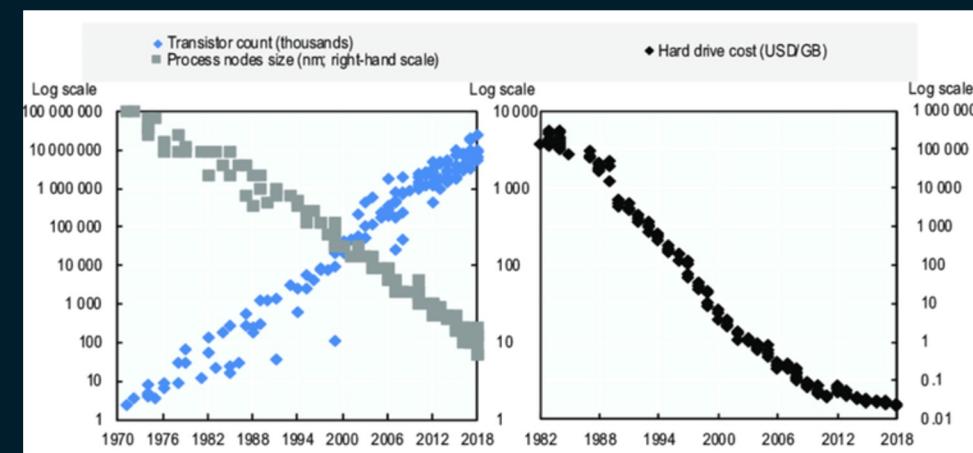


Moore's Law no longer an enabler

CPU performance is flattening



Cost of storage continues to fall





When NoSQL?

SQL

Optimized for storage

Normalized/relational

Ad hoc queries

Scale vertically

Good for OLAP

NoSQL

Optimized for compute

De-normalized/hierarchical

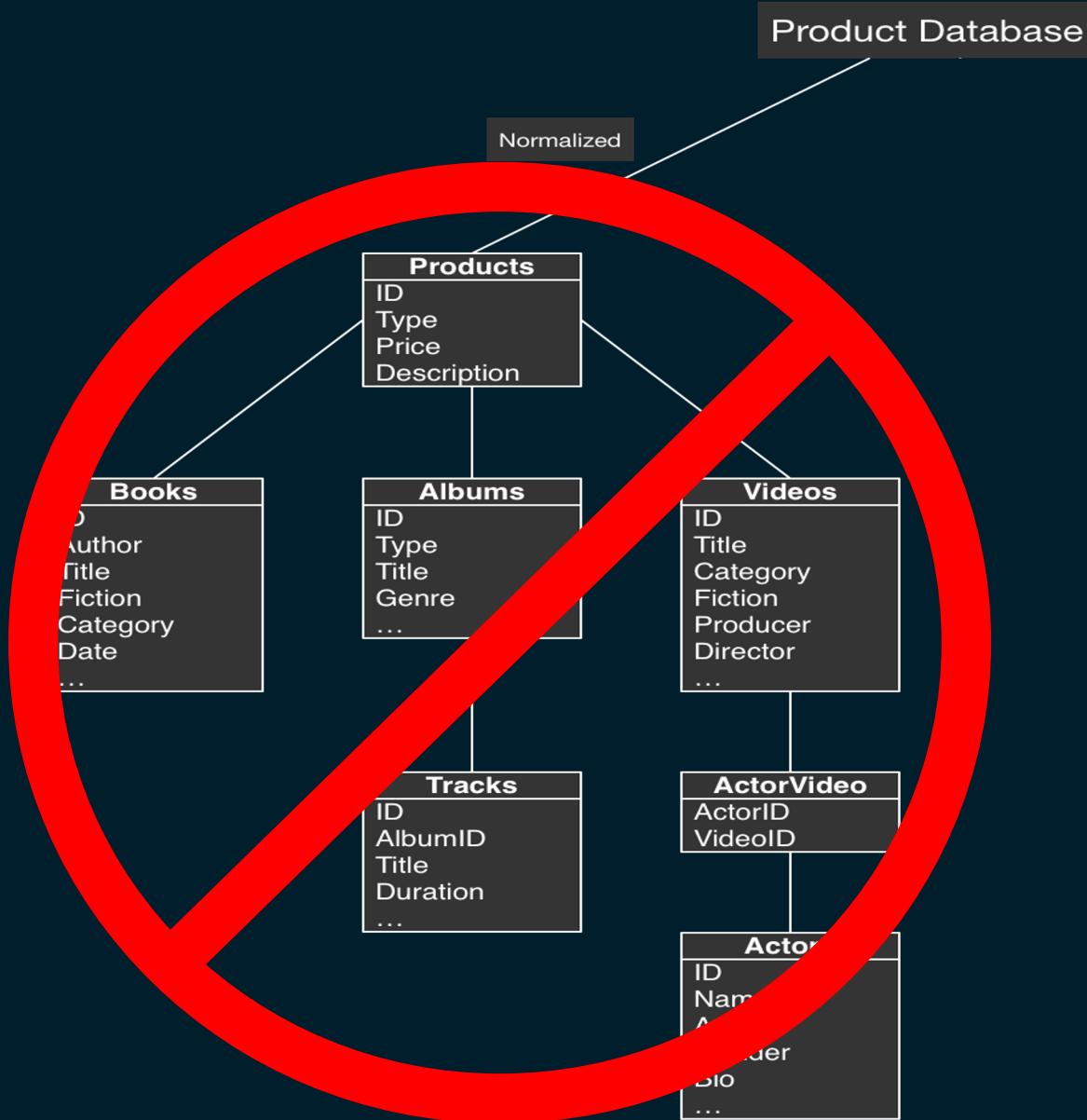
Instantiated views

Scale horizontally

Built for OLTP at scale



SQL vs. NoSQL design pattern





Ad hoc “joins” in SQL

```

SELECT * FROM PRODUCTS
INNER JOIN BOOKS
  ON productId = productId
SELECT * FROM PRODUCTS
WHERE name = "Book Title"
INNER JOIN ALBUMS
  ON productId = productId
INNER JOIN TRACKS
SELECT * FROM PRODUCTS
WHERE name = "Album Title"
  ON productId = productId
INNER JOIN ACTORVIDEO
  ON videoId = videoId
INNER JOIN ACTORS
  ON actorId = actorId
WHERE name = "Movie Title"
  
```

productId	name	type	price
1	Frankenstein	Book	11.99
2	Dire Straits	Album	17.49
3	Big	Video	14.99
4	Jane Eyre	Book	10.99
5	The Dark Side of the Moon	Album	17.49
6	Saving Private Ryan	Video	18.99

bookId	productId	author	publisher	ISBN-10
1	1	Mary Shelley	Bantam	553212478
2	4	Charlotte Brontë	Wordsworth	1853260207

albumId	productId	artist	producer	releaseDate
1	2	Dire Straits	Muff Winwood	10/7/78
2	5	Pink Floyd	Pink Floyd	3/1/73

videoId	productId	writer	director	releaseDate
1	3	Ann Spielberg	Penny Marshall	6/5/88
2	6	Robert Rodat	Steven Spielberg	7/21/98

actorVideoId	videoId	actorId	character
1	1	1	Josh
2	2	1	Captain Miller
3	1	2	Susan
4	1	3	MacMillan

actorId	gender	name	birthDate
1	M	Tom Hanks	7/9/56
2	F	Elizabeth Perkins	11/18/60
3	M	Robert Loggia	1/3/30

trackId	albumId	song	duration
1	1	Down to the Waterline	3:55
2	1	Water of Love	5:23
3	1	Setting Me Up	3:18
4	1	Six Blade Knife	4:10
5	1	Southbound Again	2:58
6	1	Sultans of Swing	5:47
7	1	In the Gallery	6:16
8	1	Wild West End	4:42
9	1	Lions	5:05
10	2	Speak to Me	1:13
11	2	Breathe	2:43
12	2	On the Run	3:36
13	2	Time	4:36
14	2	The Great Gig in the Sky	19:27
15	2	Money	6:23
16	2	Us and Them	7:49
17	2	Any Colour You Like	3:26
18	2	Brain Damage	3:49
19	2	Eclipse	2:03

Time Complexity

$$O(\log(N) \log(M) \log(M) \log(M) \log(M))$$



Modeled “joins” in NoSQL

```
SELECT * WHERE _id = { book: "Frankenstein" }
```

```
SELECT * WHERE _id = { album: "Dire Straits" }
```

```
SELECT * WHERE _id = { movie: "Big" }
```

Time Complexity

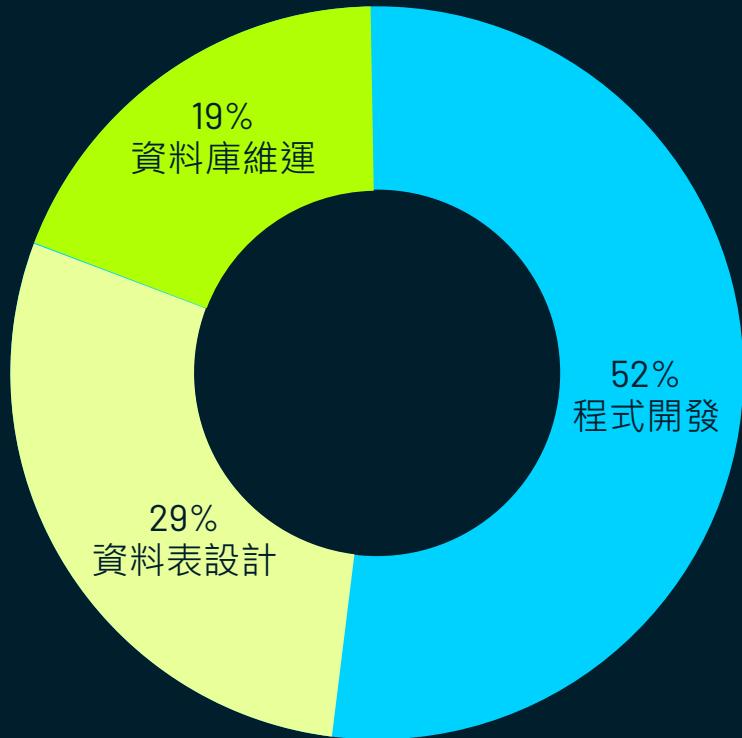
$O(\log(N))$

```
{
  "_id": "book",
  "_id": "Frankenstein",
  "type": "book",
  "data": {
    "price": 11.99,
    "isbn-10": 553212478
  },
  "target": [
    {"id": "Frankenstein"},
    {"id": "Mary Shelley", "type": "author"},
    {"id": "Bantam", "type": "publisher"}
  ],
  "target": [
    {"id": "Ann Spielberg", "type": "writer"},
    {"id": "Tom Hanks", "type": "actor"},
    {"id": "Elizabeth Perkins", "type": "actor"},
    {"id": "Robert Loggia", "type": "actor"}
  ]
},
```

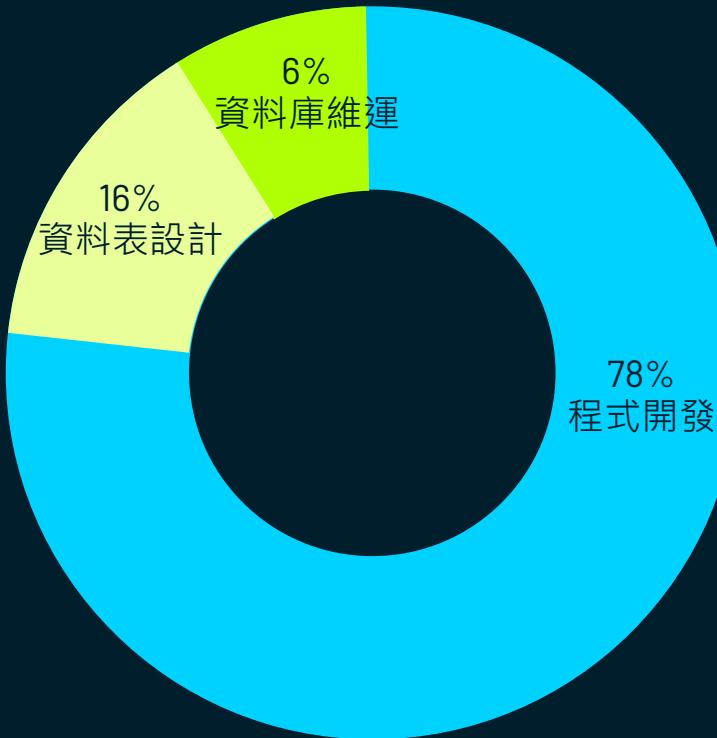


彈性資料結構，加速應用系統開發

RDBMS



NoSQL



減少修改 ORM 程式



資料結構調整不停機

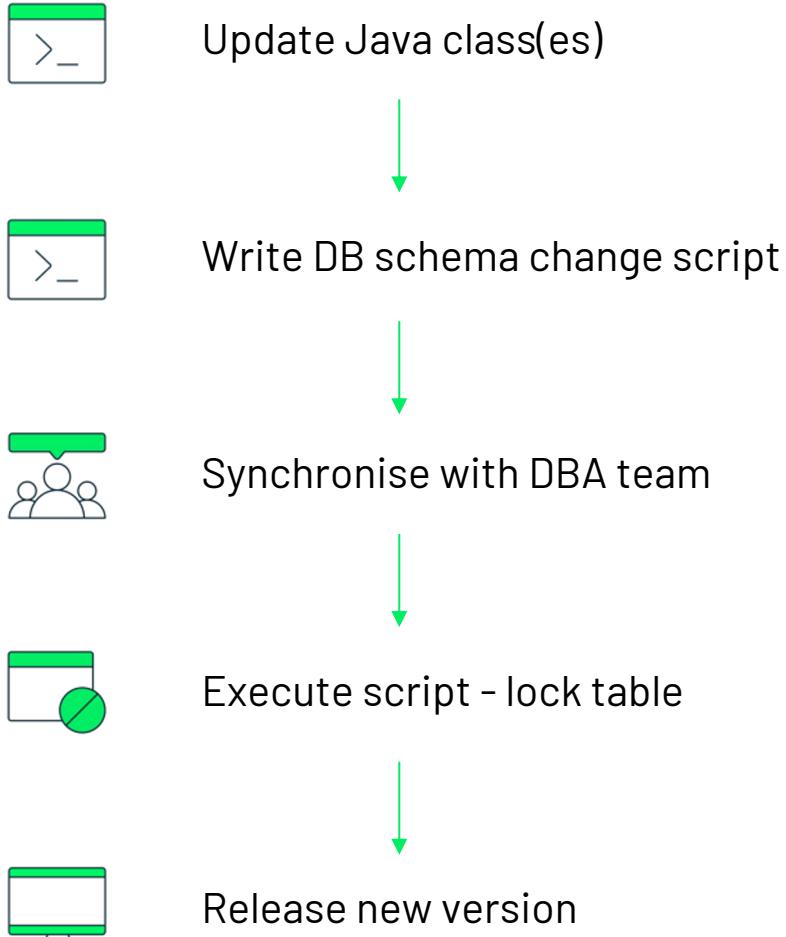


減低 DBA 維運作業



Release new feature

RELATIONAL





MongoDB 是應用最廣泛的 NoSQL 資料庫

MONGODB 下載次數

265,000,000+

線上課程註冊人次

1,500,000+

MONGODB ATLAS CLUSTERS

2,000,000+

全球服務商業合作夥伴

1,000+

全球跨產業客戶數

37,000+



連續 8 年蟬聯
NoSQL Database 第一名

DBMS

Oracle

MySQL

Microsoft SQL Server

PostgreSQL

MongoDB

IBM DB 2

DB-Engines Rankings:
<https://db-engines.com/en/ranking>



MongoDB 持續演進並推陳出新

STACK OVERFLOW SURVEY | MOST WANTED DATABASE

2015

2016

2017

2018

2019

2020

2021-2022

2022-2023

2015	2016	2017	2018	2019	2020	2021-2022	2022-2023
3.0 & 3.2	3.4	3.6	4.0	4.2	4.4	5.x	6.x
Doc-Level Concurrency RAFT / Fast Failover \$lookup Ops Manager Compression ≤50 replicas Aggregation ++ Encrypted and In-Memory storage engines BI Connector Compass	Views Graph Processing Zones ++Aggregation ++ Auto-balancing ++ Linearizable Reads Decimal Intra-cluster Compression Log Redaction Spark Connector ++ BI Connector ++	Change Streams Retryable Writes Schema Validation Expressive \$lookUp Query Expressivity Causal Consistency Consistent Sharded Secondary Reads Query Advisor End to End Compression WiredTiger 1m+ Collections MongoDB BI Connector ++ R Driver Charts (post GA) Atlas X-Region Replication Auto Storage Scaling	Replica set transactions Atlas global clusters 40% Faster Shard Migrations Atlas HIPAA Atlas LDAP Atlas Audit Atlas Enc. Storage Engine Atlas Backup Snapshots Type Conversions Snapshot Reads Non-Blocking Sec. Reads SHA-2 & TLS 1.1+ Compass Agg Pipeline Builder Compass Export to Code Free Monitoring Cloud Service Ops Manager K8s Beta	Distributed Transactions Client-Side Field Level Encryption Materialized Views Wildcard Indexes Global PIT Reads Large Transactions Mutable Shard Key Values Atlas Data Lake (Beta) Atlas Auto Scaling (Beta) Atlas Search (Beta) Multi-CAs Expressive Updates Apache Kafka Connector MongoDB Charts GA Retryable Reads & Writes New Index Builds 10x Faster stepDown Storage Node Watchdog Zstandard Compression	Union Custom Agg Expressions Refinable Shard Keys Compound Hashed Shard Keys Mirrored Reads Hedged Reads Resumable Initial Sync Time-Based Oplog Retention Simultaneous Indexing Hidden Indexes Streaming Replication Global Read/Write Concerns Rust & Swift Drivers GA TLS 1.3 & Faster Client Auth OCSP Stapling Kerberos Utility Atlas Online Archive Auto-Scaling Schema Recommendations AWS IAM Auth & Atlas x509 Federated Queries Multi-cloud clusters	Time Series collections Clustered indexes Window functions Live resharding Client-Side FLE KMIP & cloud KMS Atlas Serverless (preview) Atlas Search fast facets, function scores, synonyms Long running snapshot reads Sharded \$lookup & \$graphlookup Hidden Indexes Streaming Replication Majority write concern default 4x faster initial sync Schema validation diagnostics Rust & Swift Drivers GA TLS 1.3 & Faster Client Auth OCSP Stapling Kerberos Utility Atlas Online Archive Auto-Scaling Schema Recommendations AWS IAM Auth & Atlas x509 Federated Queries Multi-cloud clusters	Queryable encryption (preview) Atlas Serverless GA Column store indexes Cluster-to-cluster sync Independent analytic nodes scaling Flexible sync GA Asymmetric sync preview Kotlin and Flutter SDKs for Realm \$lookup perf improvements Atlas CLI GA Atlas Data API GA Secondary index on measurements for time series Time series read perf + Atlas Data Lake with fully managed storage Atlas Data Federation Atlas Charts dashboard embedding Atlas SQL interface Encrypted audit log Change streams improvements Announcement of relational to MongoDB migrator
WiredTiger (Acquisition + Integration)		MongoDB Atlas		ACID (Transactions)		Multi-Cloud	
Stable API & Rapid Releases		Developer Data Platform					

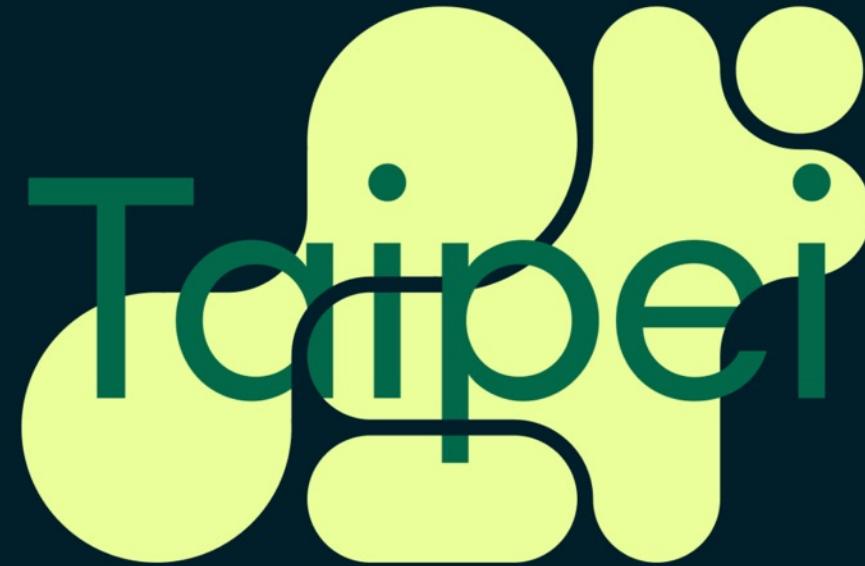


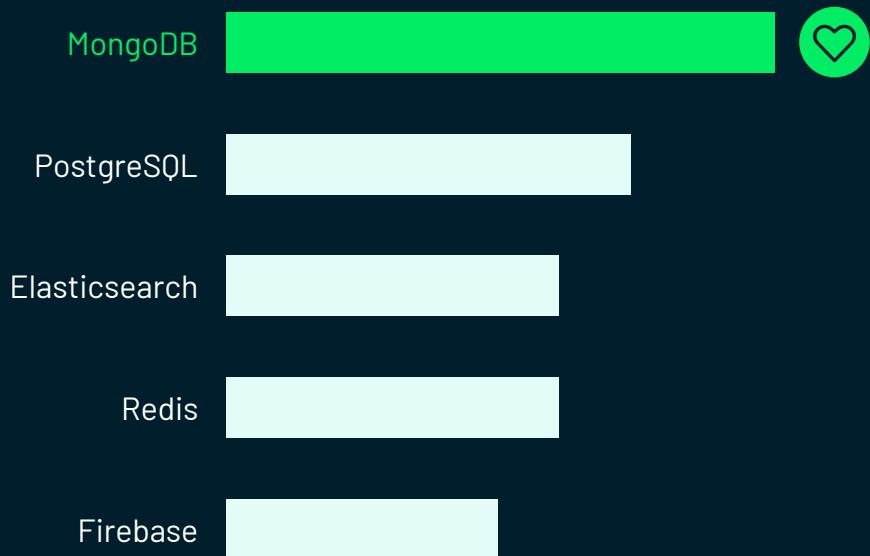
MongoDB.local Taipei

2023年9月7日，來自MongoDB的專家、社區技術大咖、各產業精英客戶和合作夥伴齊聚一堂，共同學習最新技術和成功案例，探討業務和資料庫科技的未來。

立即註冊

9/7 台北萬豪





Consistently voted
“most wanted” database

STACK OVERFLOW | DEVELOPER SURVEY

MongoDB 是討論度最高 NoSQL 資料庫



彈性多變能適應各種場景的
資料模型



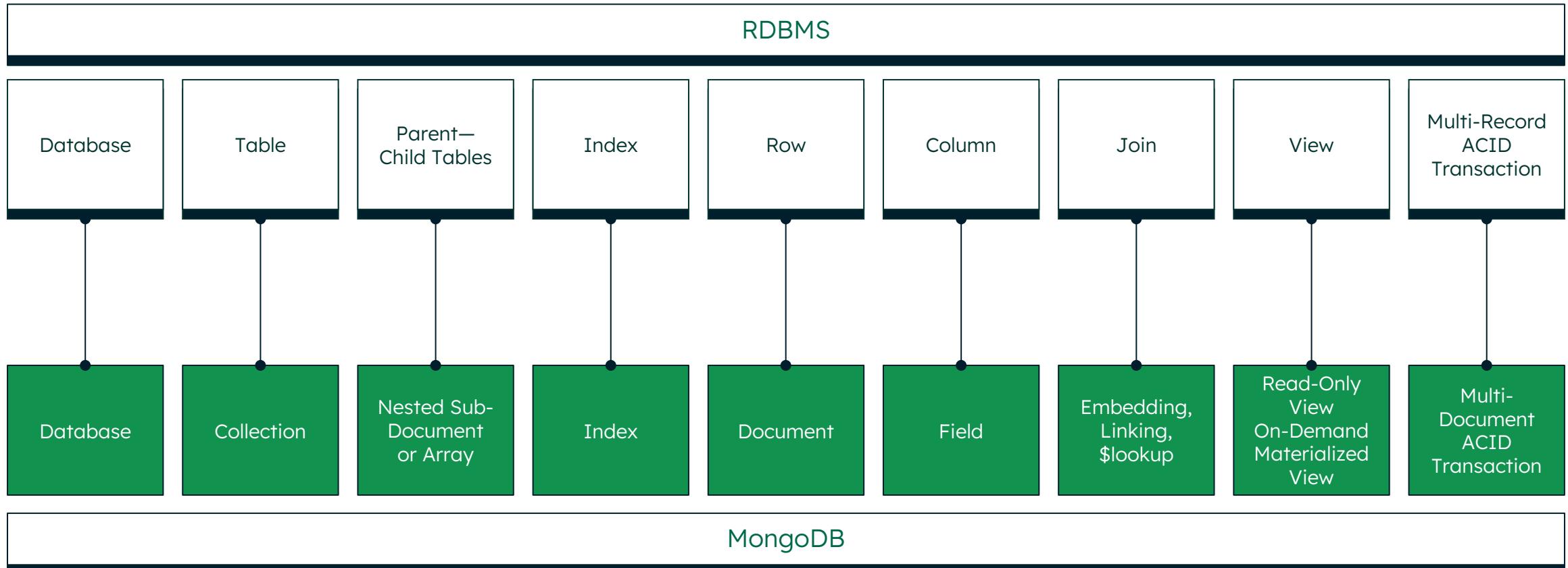
高速存取效能且支援高併發量的
資料庫引擎



分散式儲存且服務不中斷的
部署架構



MongoDB 是最像 RDBMS 的 NoSQL 資料庫





JSON Document

未來資料模型趨勢

JSON 是微服務架構的資料格式標準

```
{  
  "_id" : ObjectId("5ad88534e3632e1a35a"),  
  "name" : {  
    "first" : "John",  
    "last" : "Doe" },  
  "address" : [  
    { "location" : "work",  
      "address" : {  
        "street" : "16 Hatfields",  
        "city" : "London",  
        "postal_code" : "SE1 8DJ"},  
        "geo" : { "type" : "Point",  
          "coord" : [  
            51.5065752,-0.109081]}},  
    + [...]  
  ],  
  "dob" : ISODate("1977-04-01T05:00:00Z"),  
  "retirement_fund" :  
    NumberDecimal("1292815.75")  
}
```

```
{  
  "_id" : ObjectId("5ad88534e3632e1a35a"),  
  "name" : {  
    "first" : "John",  
    "last" : "Doe" },  
  "address" : [  
    { "location" : "work",  
      "address" : {  
        "street" : "16 Hatfields",  
        "city" : "London",  
        "postal_code" : "SE1 8DJ"},  
        "geo" : { "type" : "Point",  
          "coord" : [  
            51.5065752,-0.109081]}},  
    + [...]  
  ],  
  "phones" : [  
    { "location" : "work",  
      "number" : "+44-1234567890"},  
    + [...]  
  ],  
  "dob" : ISODate("1977-04-01T05:00:00Z"),  
  "retirement_fund" :  
    NumberDecimal("1292815.75")  
}
```

根據需求快速且彈性的調整資料樣貌



JSON Documents



Tabular



Key-Value



Text



Geospatial



Graph



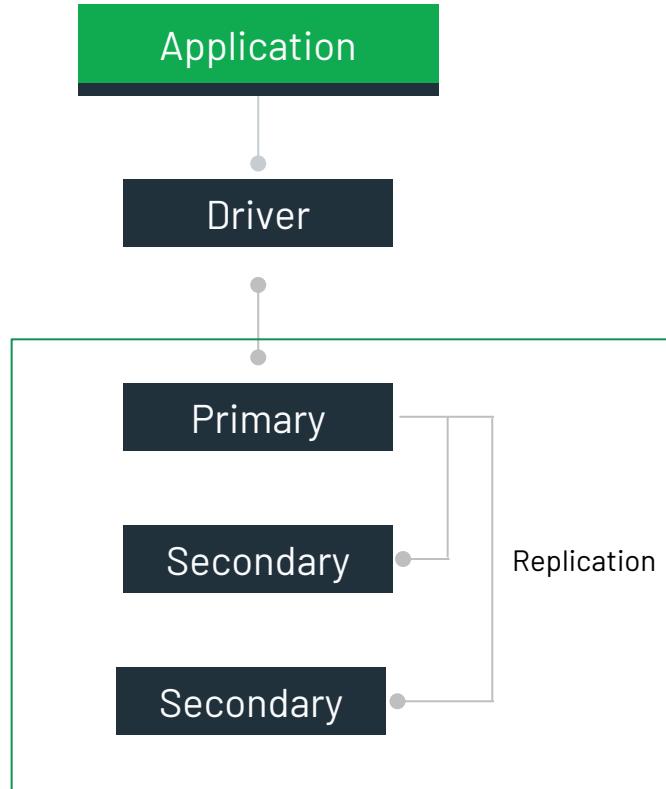
Time Series



Events



Replicaset 架構：高可用、資料不遺失



預設架構採一主二從模式：

- 自主切換：異常數秒內自動切換，交易不會中斷
- 奇數節點：最小為 3 個節點，可擴展至 50 個節點
- 自我修復：異常恢復自動加入集群

透過交易日誌同步進行資料抄寫：

- 加密傳輸：資料交換透過 TLS 加密傳輸
- 壓縮傳輸：資料先壓縮再傳輸，壓縮比例約 3 倍~4 倍

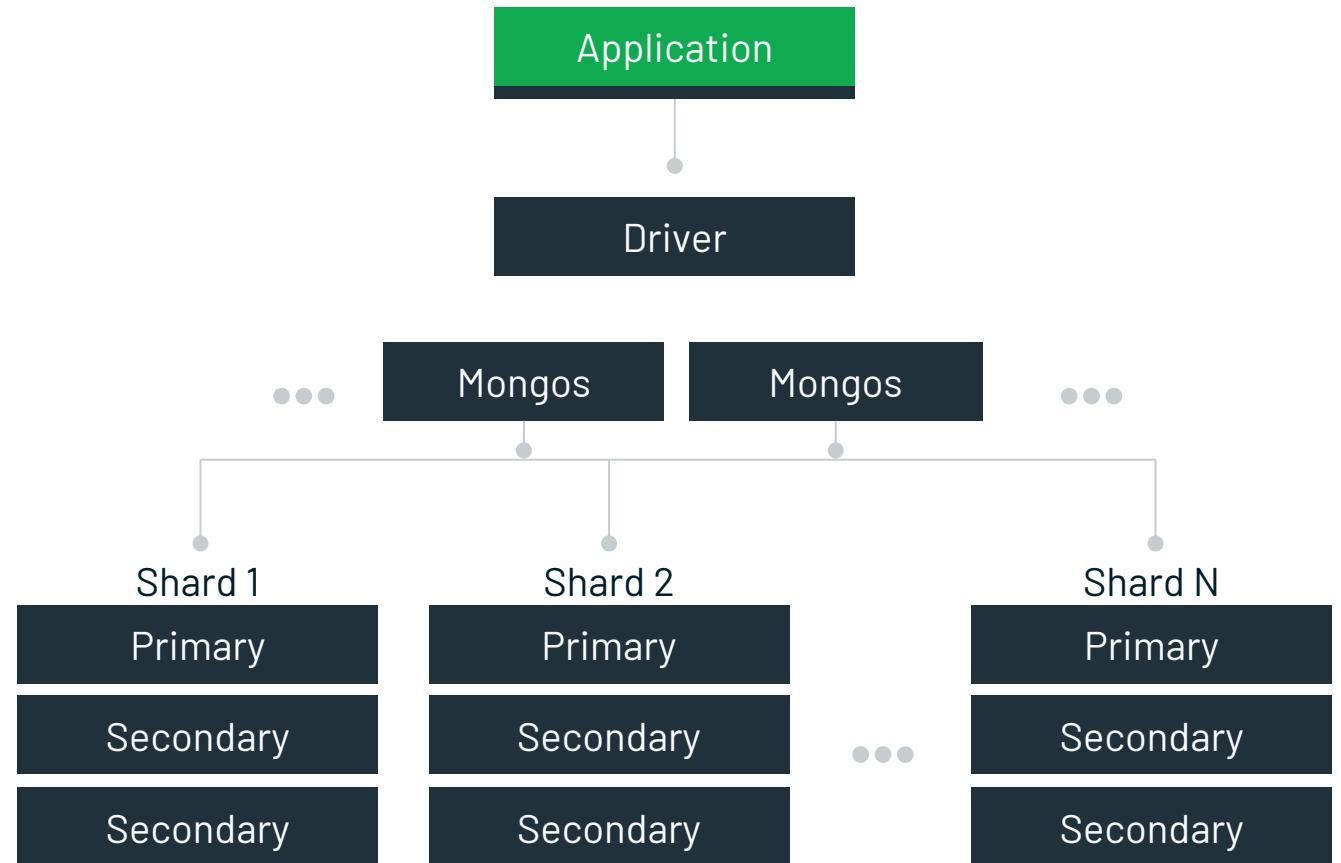


Shard Cluster: 橫向擴充，滿足海量數據需求



支援資料**分散讀寫處理**能力，為應用隨時**保持最佳狀態**

- 使用時機：
 - 資料量成長已經無法由單機負荷
 - 資料伺服器眾多，難以個別管理
- 使用方式：
 - **自動水平擴展**
 - **不停機，資料背景搬移**
 - **應用程式無需改變**
- 使用效益：
 - 分散讀寫，提升效能
 - 隨需擴充，符合成本效益





LEARN MONGODB

MongoDB University

Our free, online learning content addresses needs throughout the project lifecycle. We make it easy for your team to pick up knowledge quickly, develop a comprehensive MongoDB skill set, and get certified.

<https://learn.mongodb.com/>



UNIVERSITY

Learning Experience

- Learn how to use MongoDB with programming language-specific courses (C#, Java, Node.js, Python)
- Pick up knowledge quickly with short video lectures
- Practice skills with interactive, hands-on labs
- Test knowledge with embedded quizzes
- Build a foundational skill set with learning paths for developers and DBAs
- Learn about new features with short video tutorials
- Access global subtitles (English, Chinese [Traditional & Simplified], Spanish, French, Korean, and Portuguese)

The screenshot shows the MongoDB Learning Experience platform. At the top, a dark header features the text "Learn MongoDB and Advance Your Career." in large white font, followed by a smaller line of text: "Gain MongoDB product knowledge, practice skills with hands-on labs and quizzes, and get certified." Below the header is a search bar with a placeholder "Search for content" and a "Search" button. A "See All Content" link is located in the top right corner. The main content area is titled "Featured Training" and includes four cards: "MongoDB Developer Learning Paths" (purple card), "Introduction to MongoDB" (light green card), "Node.js Developer Path" (pink card), and "Associate Developer Exam" (light pink card). Each card has a small icon and a brief description.

Learn MongoDB and Advance Your Career.

Gain MongoDB product knowledge, practice skills with hands-on labs and quizzes, and get certified.

Search for content

Search

See All Content →

Featured Training

MongoDB Developer Learning Paths

These learning paths combine a series of courses to teach you MongoDB skills with your programming language of choice.

Introduction to MongoDB

You'll be guided through the foundational skills and knowledge you need to get started with MongoDB.

Node.js Developer Path

In this path, you'll learn the basics of building modern applications with Node.js, using MongoDB as your database.

Associate Developer Exam

When you become MongoDB Certified, you're achieving the same level of expertise that we require of our own developers.



UNIVERSITY

Certification

- Validate ability as a developer or DBA
- Ensure teams share a baseline skill set and common understanding
- Promote professional development
- Robust prep materials: study guides, videos, and practice questions
- Complete a learning path and receive 50% off the exam
- Digital badges to publicly share credential



Shareable to:





Step 0: 申請免費的 MongoDB Atlas

今日開始之前，你應該已經完成：

- ✓ 透過 Google 帳號申請 MongoDB Atlas
- ✓ 設定一組 M0 免費的 MongoDB Cluster
- ✓ 完成 Security Quickstart
- ✓ 完成 Sample Data 載入

MongoDB Atlas

- ✓ **Work with your data as code**
Documents in MongoDB map directly to objects in your programming language. Modify your schema as your apps grow over time.
- ✓ **Focus on building, not managing**
Let MongoDB Atlas take care of the infrastructure operations you need for performance at scale, from always-on security to point-in-time recovery.
- ✓ **Simplify your data dependencies**
Leverage application data for full-text search, real-time analytics, rich visualizations and more with a single API and minimal data movement.

First Name*

Last Name*

Company

Email*

Password*

I agree to the [Terms of Service](#) and [Privacy Policy](#).

Create your Atlas account

or

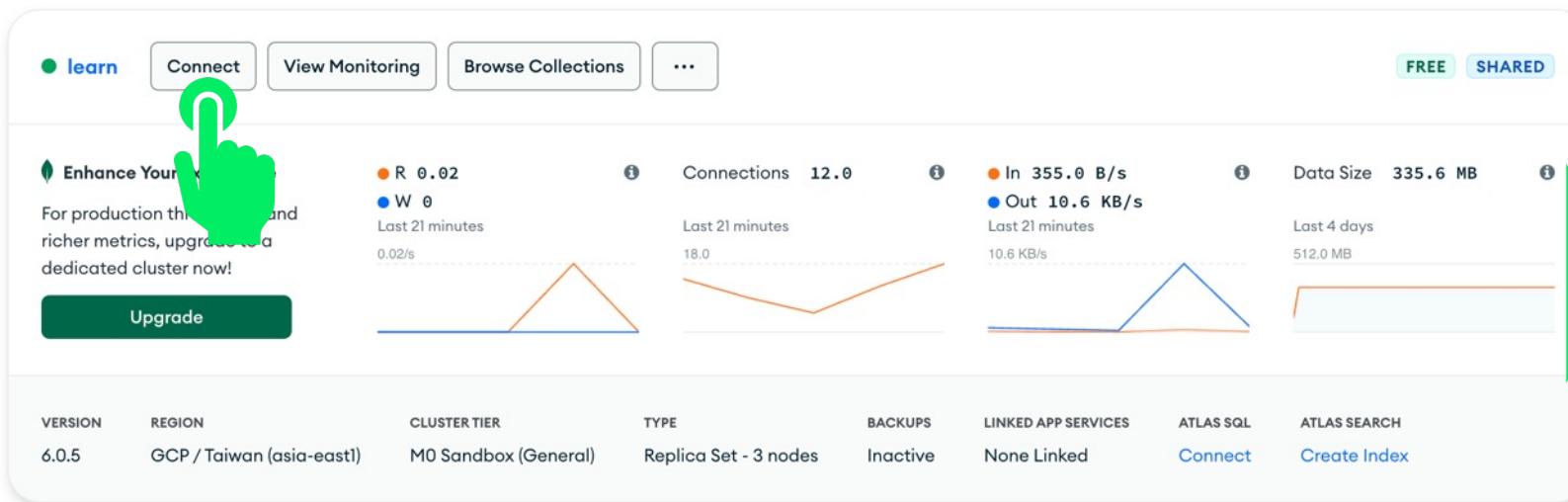
Sign up with Google



Exercise 1: 運用 MongoDB 工具連結 Atlas

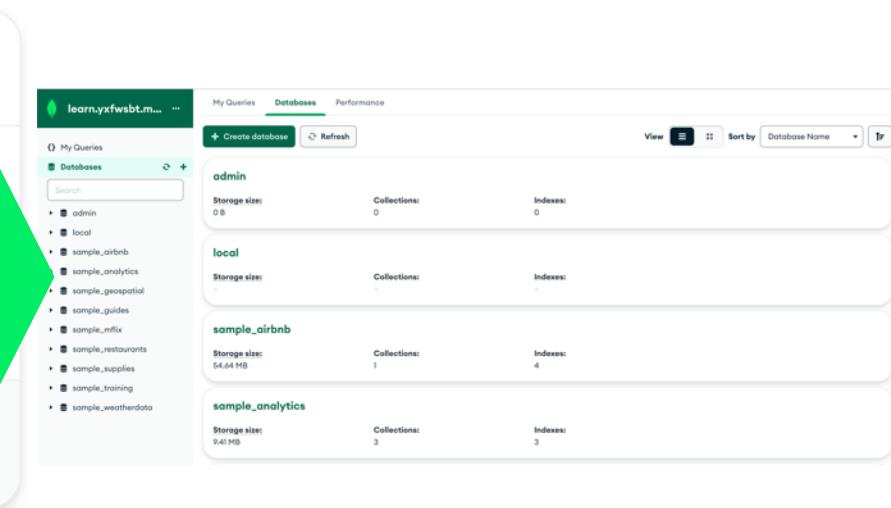
從 Deployment 下的 Connect 完成：

- Compass 連結



The screenshot shows the MongoDB Atlas dashboard for a cluster named 'learn.yxfwsbt.m'. The 'Connect' button is highlighted with a large green hand icon. The dashboard displays various metrics and configuration details:

- Metrics:** R 0.02 (Read operations per second), W 0 (Write operations per second), Connections 12.0 (Last 21 minutes), In 355.0 B/s (Input bandwidth), Out 10.6 KB/s (Output bandwidth), Data Size 335.6 MB (Last 4 days).
- Cluster Configuration:** VERSION 6.0.5, REGION GCP / Taiwan (asia-east1), CLUSTER TIER M0 Sandbox (General), TYPE Replica Set - 3 nodes, BACKUPS Inactive, LINKED APP SERVICES None Linked, ATLAS SQL Connect, ATLAS SEARCH Create Index.



The right side of the screenshot shows the MongoDB Compass interface. A green arrow points from the 'Connect' button on the left to the 'Databases' tab in the Compass interface. The Compass interface lists the following databases and their details:

- admin**: Storage size: 0 B, Collections: 0, Indexes: 0.
- local**: Storage size: 0 B, Collections: 0, Indexes: 0.
- sample_airbnb**: Storage size: 54.64 MB, Collections: 1, Indexes: 4.
- sample_analytics**: Storage size: 9.41 MB, Collections: 3, Indexes: 3.
- sample_geospatial**: Storage size: 0 B, Collections: 0, Indexes: 0.
- sample_guides**: Storage size: 0 B, Collections: 0, Indexes: 0.
- sample_mflix**: Storage size: 0 B, Collections: 0, Indexes: 0.
- sample_restaurants**: Storage size: 0 B, Collections: 0, Indexes: 0.
- sample_supplies**: Storage size: 0 B, Collections: 0, Indexes: 0.
- sample_training**: Storage size: 0 B, Collections: 0, Indexes: 0.
- sample_weatherdata**: Storage size: 0 B, Collections: 0, Indexes: 0.



CRUD - Create/Insert Data

- Inserts document(s) into a collection
- Can optionally define the writeConcern of the operation
- _id field is automatically generated with an ObjectId

```
db.mytab.insertOne(  
  { item: "envelopes", qty : 100, type: "Clasp" },  
  { writeConcern: { w: "majority", wtimeout: 5000 } }  
)
```

```
INSERT INTO mytab(item, qty, type)  
VALUES ("envelopes", 100, "Clasp" )
```



CRUD - Read Data

- Get documents from the collection (table)
 - Can filter by content (query) and by what is returned (project)
 - `findOne()` returns just the first document
 - `findMany()` return all of result
-

```
db.mytab.find( {x:1, y:2}, {a:1, b:1, c:1} )
```

```
SELECT a, b, c
```

```
FROM mytab
```

```
WHERE x = 1 and y = 2
```



CRUD - Update Data

- Modifies existing document(s) in a collection
- Use with a field update operator (MQL)
- Query for the documents you want to update
- Can optionally upsert, multi-update, define writeConcern, etc.

```
db.books.updateOne(  
  { author: "Tom Gleitsmann" },  
  { $set: { author: "Thomas Gleitsmann" }})
```

```
UPDATE books  
SET author = "Thomas Gleitsmann"  
WHERE author = "Tom Gleitsmann"
```



CRUD - Delete Data

- Deletes a document from the collection
- Takes an optional writeConcern argument

```
db.products.deleteMany(  
  { qty: { $gt: 20 } },  
  { writeConcern: { w: "majority", wtimeout: 5000 } })
```

DELETE **books**

WHERE **qty > 20**



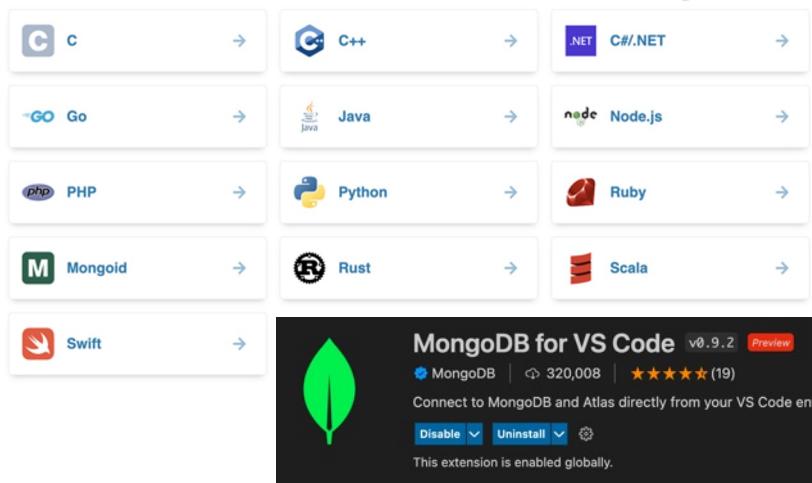
MongoDB Developer Ecosystem



Build Applications

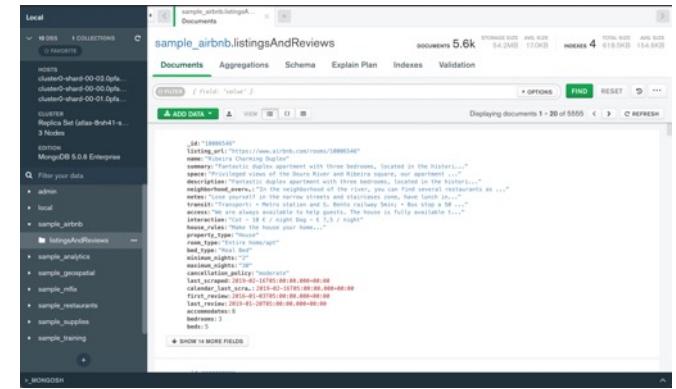
IDE Integrations

/S Code for MongoDB



Navigate & explore your data

MongoDB Compass



Interface with MongoDB

MongoDB Shell

<https://docs.mongodb.com/mongodb-shell/install#mdb-shell-install>

<https://www.mongodb.com/try/download/compass>



MongoDB Compass 好的工具帶你上天堂

The screenshot shows the MongoDB Compass interface for the database 'TAG' and collection 'raw.dw_cscust'. The 'Aggregations' tab is selected. The pipeline consists of the following stages:

```
1 { $temp: { tx_products: { $size: { $setUnion: ["$tx.PDAC_CD", "$tx.PC"] } } }, card_products: { $size: { $setUnion: [ "$card.CC_ACCT_TYP", "$card.CC_ACCT_TYP" ] } } },
2 { $visit_count: { $sum: "$track.VISITCOUNT" } },
3 { $visit_avg: { $avg: "$track.VISITDURATION" } },
4 { $last_visit: { $max: "$track.LASTACTIONDATETIME" } },
5 { $tx_in: { $filter: { input: "$tx", as: "each", cond: { $eq: ["$$each.DB_CR", "C"] } } } }
```

The preview stage shows sample documents with fields like CUST_KEY, CUST_TYP, DOB, tx, track, card, acct, cardinfo, and temp.

- Compass 是 MongoDB 官方提供**免費的 Developer 工具**
- 透過內建 Aggregation Builder 能大幅**增加 Aggregation Pipeline 開發的效率**
- 提供 Preview, Code Assistant, Hiding Pipeline, Export to Programming Language 功能



運用 Compass 汇出語言功能加速開發效率

The screenshot shows the Compass MongoDB interface. On the left, the collection 'raw.dm_04cc_ascrdinfo' is selected. The main area displays document statistics: 3.0m DOCUMENTS and 2 INDEXES. Below this, there are tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. A 'Filter' dropdown is set to '{CUST_KEY: "V258472112"}'. The 'Find' button is highlighted with a green box. To its right is a 'More Options' dropdown. At the bottom, there are 'ADD DATA' and 'EXPORT DATA' buttons, along with navigation controls (1-1 of 1, back, forward).

A green line connects the 'Find' button to a modal window titled 'Export Query To Language'.

Export Query To Language

My Query

Shell

```
{  
  CUST_KEY: "V258472112"  
}
```

Exported Query

Java

```
import org.bson.Document;  
import com.mongodb.client.FindIterable;  
  
/*  
 * Requires the MongoDB Java Driver.  
 * https://mongodb.github.io/mongo-java-driver  
 */  
  
Bson filter = eq("CUST_KEY", "V258472112");  
  
MongoClient mongoClient = new MongoClient(  
  MongoClientOptions.builder()  
    .build());  
  
FindIterable<Document> findIterable = mongoClient.getDatabase("test").getCollection("test").find(filter);  
  
List<Document> documents = findIterable.into(new ArrayList<>());  
  
System.out.println(documents);
```

Include Import Statements
 Include Driver Syntax
 Use Builders

Close



今日將會採用
sample_mflix.movies

作為測試

sample_mflix.movies

STORAGE SIZE: 20.79MB LOGICAL DATA SIZE: 36.06MB TOTAL DOCUMENTS: 23530 INDEXES TOTAL SIZE: 21.7MB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes •

INSERT DOCUMENT

FILTER

{ field: 'value' }

▶ OPTIONS

Apply

Reset

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('573a1390f29313caabcd4135')
plot: "Three men hammer on an anvil and pass a bottle of beer around."
> genres: Array
  runtime: 1
> cast: Array
  num_mflix_comments: 0
  title: "Blacksmith Scene"
  fullplot: "A stationary camera looks at a large anvil with a blacksmith behind it..."
> countries: Array
  released: 1893-05-09T00:00:00.000+00:00
> directors: Array
  rated: "UNRATED"
> awards: Object
  lastupdated: "2015-08-26 00:03:50.133000000"
  year: 1893
> imdb: Object
  type: "movie"
> tomatoes: Object
```

```
_id: ObjectId('573a1390f29313caabcd42e8')
plot: "A group of bandits stage a brazen train hold-up. Only to find a determ..."
```

```
use sample_mflix
db.movies.insertOne(
{
  "title": "Ghostbusters",
  "Year": 1984,
  "Rated": "PG",
  "Runtime": 105,
  "Type": "movie",
  "Genres": ["comedy", "action"] ,
  "Director": "Ivan Reitman",
  "Writers": ["Dan Aykroyd", "Harold Ramis"]
})
```

PREVIOUS

1-20 of many results

NEXT >



Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes •

INSERT DOCUMENT

FILTER {title: "Ghostbusters"} ▾ OPTIONS Apply Reset

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('573a1398f29313caabce912c')
plot: "Three unemployed parapsychology professors set up shop as a unique gho..."
> genres: Array
  runtime: 105
  metacritic: 67
  rated: "PG"
> cast: Array
  num_mflix_comments: 1
  poster: "https://m.media-amazon.com/images/M/MV5BMTkxMjYyNzgwMl5BMl5BanBnXkFtZT...
  title: "Ghostbusters"
  fullplot: "Three odd-ball scientists get kicked out of their cushy positions at a...
> languages: Array
  released: 1984-06-08T00:00:00.000+00:00
> directors: Array
> writers: Array
> awards: Object
  lastupdated: "2015-09-04 00:15:08.537000000"
  year: 1984
> imdb: Object
> countries: Array
  type: "movie"
```

```
use sample_mflix
db.movies.find(
  {"title": "Ghostbusters"}
)
```



FILTER {title: "Ghostbusters"} ▶ OPTIONS Apply Reset

QUERY RESULTS: 1-1 OF 1

▶ _id: ObjectId('573a1398f29313caabce912c')
plot: "Three unemployed parapsychology professors set up shop as a unique gho..."
> genres: Array
runtime: 105
metacritic: 67
rated: "PG"
> cast: Array
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BMTkxMjYyNzgwMl5BMl5BanBnXkFtZT..."
title: "Ghostbusters"
fullplot: "Three odd-ball scientists get kicked out of their cushy positions at a..."
> languages: Array
released: 1984-06-08T00:00:00.000+00:00
> directors: Array
> writers: Array
> awards: Object
lastupdated: "2015-09-04 00:15:08.537000000"
year: 1984
> imdb: Object

```
use sample_mflix
db.movies.updateOne(
  {title: "Ghostbusters"}, {
    $set: {
      imdb: {
        id: "tt0087332",
        rating: 7.8,
        votes: 312798
      }
    }
  }
)
```



Exercise 2: CRUD 操作

Query	What's the Question?
<code>db.movies.find({year: 1987})</code>	... from 1987
<code>db.movies.find({genres: "Comedy"})</code>	"Comedy" as one of their genres
<code>db.movies.find({genres: ["Comedy"]})</code>	"Comedy" as the <i>only</i> genre
<code>db.movies.find({genres: {\$in:["Comedy", "Drama"]}})</code>	"Comedy" or "Drama" as the genre
<code>db.movies.find({revenues: {\$exists: true}})</code>	Documents that have a revenue field
<code>db.movies.find({imdb.rating: {\$gt: 8.0}, rated: "PG"})</code>	IMDB Rating > 8.0 and PG Rating
<code>db.movies.find({title: {\$regex: '^Dr. Strangelove'}})</code>	Title starting with "Dr. Strangelove"

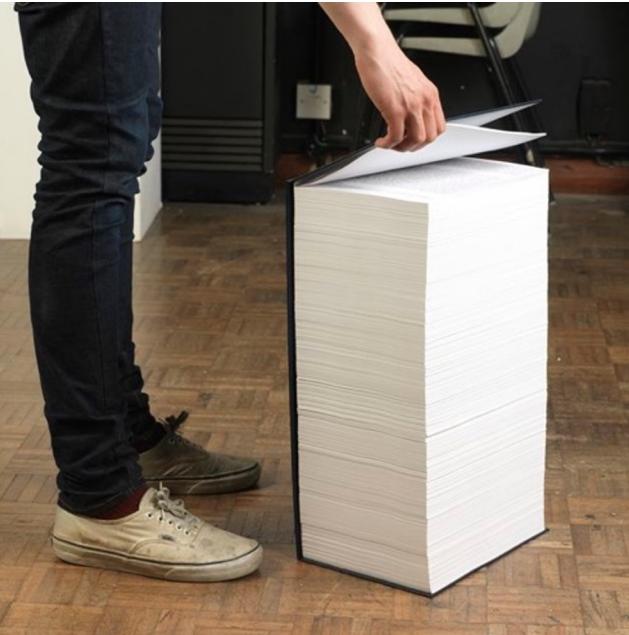


{ 2: “Index & Shard Key” }



有沒有 Index 對查詢效能差很多

COLLSCAN: No Index for Query



IXSCAN: Use Index for Query

Alphabetical Dictionary Index

(Words given in **bold type** are the names of contents or sections.
Numbers in **bold** are the context numbers.)

A
abacus 116 74
about 116 74
above 122 86
accelerator 103 5
accelerator pedal
107 52
about about to happen 7
accident, it was a 52 3
according 112 25
accept 127 17
accuser 127 15
accusing 44
adult 4 13
across (from) 128 15
ACTIONS related pp.
40–49; in sequence
pp. 40–49
Adam's apple 110 7
addition 116 46
address 99 55
addressing 99 66
adult 103 3; 105 103
adventure the film 102 8
adversaries 127 27
adversity 137 7
affection,
expressing 55
afraid 14 21
after that 94 2
afternoon 2 1, 2; 12 4
against 128 21
alone 63 1
air cleaner 107 29
air pollution 137 60
air traffic controller
138 8
airliner 138 1
airplane 138 1
airport 138
airship 138 55
alarm clock 134 39
album cover 13138
All right! 28
already 37 52
ally/ally 137 53
along 128 24
alone 114 21
amongst/fol 132 39
ambulance 138 30
ambulance attendant
138 30
anchovy 128 5
anchovy 121 21
anemometer 135 55
angry 114 7
animal 104 2
animal 110 45
ant 126 21
ants 138 6, 35
antihill 132 36
aperture 102 4
apple 125 18
appointment book
136 9
approach the runway
128 21
apricot 125 16
April 129 18
apress 132 57; 138 16
arachnid 138 21
architect 63 111
air cleaner 107 29
air pollution 137 60
air traffic controller
138 8
airliner 138 1
airplane 138 1
airport 138
airship 138 55
alarm clock 134 39
album cover 13138
All right! 28
already 37 52
ally/ally 137 53
along 128 24
alone 114 21
amongst/fol 132 39
ambulance 138 30
ambulance attendant
138 30
anchovy 128 5
anchovy 121 21
anemometer 135 55
angry 114 7
animal 104 2
animal 110 45
ant 126 21
ants 138 6, 35
antihill 132 36
aperture 102 4
apple 125 18
appointment book
136 9
approach the runway
128 21
apricot 125 16
April 129 18
apress 132 57; 138 16
arachnid 138 21
architect 63 111
air cleaner 107 29
air pollution 137 60
air traffic controller
138 8
airliner 138 1
airplane 138 1
airport 138
airship 138 55
alarm clock 134 39
arrange 123 34
arranging 66
arrangement 63
arrives 115 41
artichoke 124 27
artist 121 13
ascend 121 13
ascetic 138 44
ascetic bucket 13143
ashamed of you, I'm
52 2
ashen 122 37
ashes 131 47
ashley 106 36
asking for a price 26
asked 126 21
answers 35 26
asking for help 12, 14;
20; 27, 41
asleep 106 36
assembler 138 21
asking for the time 12
asleep 82 3
aspargus 124 7
asthma 138 21
at the bottom (of)
128 16
at the top (of) 128 17
at the top 29
attaching 89



MongoDB 支援多種類型的 Index 規格

Fully featured secondary indexes—document optimized—extended beyond RDBMS experiences

Index Types

Primary Index

Every Collection has a primary key index

Compound Index

Index against multiple keys in the document

MultiKey Index

Index into arrays

Wildcard Index

Auto-index all matching fields, sub-documents & arrays

Text Indexes

Atlas Search offers Lucene-based inverted indexes

GeoSpatial Indexes

2d & 2dSphere indexes for spatial geometries

Clustered Indexes

For time series collections, pre-sorted by timestamp for low latency queries

Index Features

TTL Indexes

Single Field indexes, when expired delete the document

Unique Indexes

Ensures value is not duplicated

Partial Indexes

Expression based indexes, allowing indexes on subsets of data

Sparse Indexes

Only index documents which have the given field

Hidden Indexes

Hide indexes from the query planner without removing them to assess impact of removing it



用 Compass 查詢沒有 index

查詢 : 1987 年的所有電影清單

Documents Aggregations Schema Explain Plan Indexes Validation

Filter {year: 1984}

如果我們用 Compass 的 Explain Plan 查詢

sample_mflix.movies 23.5k 2 DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter {year: 1984} Reset Explain More Options ▾

VIEW VISUAL TREE RAW JSON

Query Performance Summary Learn more

Documents Returned: 211 Actual Query Execution Time (ms): 22
Index Keys Examined: 0 Sorted in Memory: no
Documents Examined: 23530 ▲ No index available for this query.

COLLSCAN

nReturned 211 Execution Time 3 ms
Documents Examined: 23530

DETAILS

```
{  
  "stage": "COLLSCAN",  
  "filter": {  
    "year": {  
      "$eq": 1984  
    }  
  },  
  "nReturned": 211,  
  "executionTimeMillisEstimate": 23320,  
  "works": 23532,  
  "advanced": 211,  
  "needTime": 23320,  
  "needYield": 0,  
  "saveState": 23,  
  "restoreState": 23,  
  "isEOF": 1  
}
```



Index 建立口訣：ESR 原則

The ordering of index keys in a compound index is critically important.

E-S-R provides guidance that is useful in **most** cases:

- **Equality** first
- **Sort** next
- **Range** last



ESR 的定義

```
// Equality Queries  
coll.find({x: 123})  
coll.find({x: {$eq: 123}})  
coll.find({"x.y": 123})  
  
// Bounds  
"[123.0, 123.0]"
```

Equality Fields

An **exact** match on a single value.

```
// Sorting  
curs.sort({a:1})  
curs.sort({a:1, b:-1})  
  
// Bounds  
"[MinKey, MaxKey]",  
"[MinKey, MaxKey]"
```

Sort Fields

The (entire) request sort.

```
// Range Queries  
coll.find({z:{$gt:900}})  
coll.find({foo:{$lte:100}})  
  
// Bounds  
"(900.0, inf.0]", "[-  
inf.0, 100]"
```

Range Predicates

Any predicates that are **not** exact matches



ESR Query and Index Example

```
> db.grades.find({  
  'scores.type': 'exam', ← Equality → 'scores.type':1,  
  'scores.score': {$lt:60} ← Range ← 'student_id':1,  
}).sort({student_id:1}) ← Sort → 'scores.score':1 })
```



Queries that Index Supports

```
Create Index: { "item": 1, "location": 1, "stock": 1 }
```

```
db.collection.find({})
```

The index supports the following query:

```
{ item }
{ item, location }
{ location, item }
{ location, item, stock }
```



Queries that Index NOT Supports

```
Create Index: { "item": 1, "location": 1, "stock": 1 }
```

```
db.collection.find({})
```

The index NOT supports the following query:

```
{ location, stock }
{ stock }
```



Index Prefixes Supports Sorts

Create Index: { "item": 1, "location": 1, "stock": 1 }

```
db.collection.find().sort({})
```

The index support the following sorts:

```
{ item: 1 }
{ item: 1, location: 1 }
{ item: 1, location: 1, stock: 1 }
{ item: -1 }
{ item: -1, location: -1 }
{ item: -1, location: -1, stock: -1 }
```



Index Prefixes NOT Supports Sorts

Create Index: { "item": 1, "location": 1, "stock": 1 }

```
db.collection.find().sort({})
```

The index NOT support the following sorts:

```
{ item: 1, location: 1, stock: -1 }
{ item: 1, location: -1, stock: 1 }
{ item: -1, location: -1, stock: 1 }
{ item: -1, location: 1, stock: -1 }
```

Quick Quiz



Consider the following index in the **computers** collection:

```
{ processor: 1, price: 1, memoryGB: -1 }
```

Which of the following queries could use this index for **filtering** and **sorting**?

Check all that apply:

1

```
db.computers.find( { processor: "i7" } )  
    .sort( { price: 1 } )
```

2

```
db.computers.find( { price: 2000 } )  
    .sort( { memoryGB: -1 } )
```

3

```
db.computers.find( { processor: "i9" } )  
    .sort( { memoryGB: -1 } )
```



Exercise 3: 利用 Compass 建立 Index

The screenshot illustrates the process of creating an index for the `sample_mflix.movies` collection in the Compass interface.

Left Panel (Initial State):

- Collection:** `sample_mflix.movies`
- Documents:** 23.5k
- Indexes:** 2
- Explain Plan:** Selected tab. Shows a query performance summary:
 - Documents Returned: 211
 - Actual Query Execution Time (ms): 22
 - Index Keys Examined: 0
 - Sorted in Memory: no
 - Documents Examined: 23530
 - No index available for this query.
- COLLSCAN:** Details of the execution plan stage:
 - nReturned: 211
 - Execution Time: 3 ms
 - Documents Examined: 23530
 - Details JSON:

```
{  "stage": "COLLSCAN",  "filter": {    "year": {      "$eq": 1984    }  },  "nReturned": 211,  "executionTimeMillisEstimate": 23532,  "works": 23320,  "advanced": 211,  "needTime": 23320,  "needYield": 0,  "saveState": 23,  "restoreState": 23,  "isEOF": 1}
```

Middle Panel (Action):

- Indexes Tab:** Selected tab. Shows the `sample_mflix.movies` collection details:
 - Documents: 23.5k
 - Indexes: 2
 - Explain Plan: Selected tab
 - Query Performance Summary: Shows results for the query `{year: 1984}`.
- Create Index Dialog:** Opened over the collection details. It shows the `sample_mflix.movies` collection and the `_id` field selected for indexing.
 - Name and Definition:** `sample_mflix.movies`
 - Type:** `REGULAR`
 - Size:** 766.0 KB
 - Usage:** 0 (since Thu Jan 01 00:00:00 1970)
 - Properties:** `UNIQUE`
 - Fields:** `year` (selected), `1(asc)`

Right Panel (Final State):

- Indexes Tab:** Selected tab. Shows the `sample_mflix.movies` collection details:
 - Documents: 23.5k
 - Indexes: 2
 - Explain Plan: Selected tab
 - Query Performance Summary: Shows results for the query `{year: 1984}`. The execution time is now 0 ms.
- COLLSCAN:** Details of the execution plan stage:
 - nReturned: 211
 - Execution Time: 0 ms
 - Documents Examined: 23530
 - Details JSON:

```
{  "stage": "COLLSCAN",  "filter": {    "year": {      "$eq": 1984    }  },  "nReturned": 211,  "executionTimeMillisEstimate": 0,  "works": 211,  "advanced": 211,  "needTime": 0,  "needYield": 0,  "saveState": 0,  "restoreState": 0,  "isEOF": 1}
```

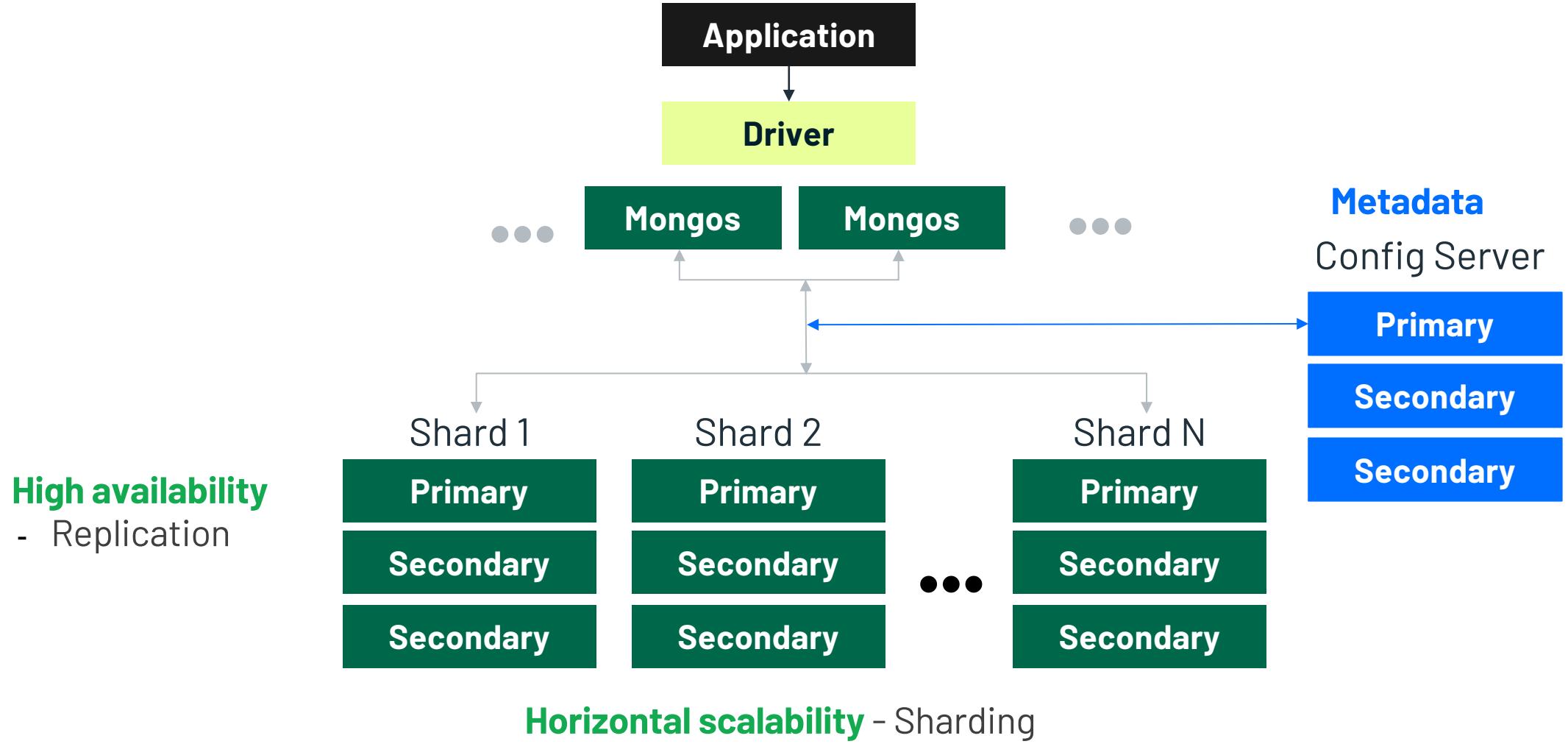
- FETCH:** Details of the execution plan stage:
- nReturned: 211
- Execution Time: 0 ms
- Documents Examined: 23530
- Details JSON:

```
{  "stage": "FETCH",  "nReturned": 211,  "executionTime": 0,  "details": [    {      "field": "year",      "value": 1984,      "type": "NUMBER",      "isKey": true,      "sortOrder": 1,      "isLast": true    }  ]}
```
- IXSCAN:** Details of the execution plan stage:
- nReturned: 211
- Execution Time: 0 ms
- Index Name: `year_1`
- Multi Key Index: no
- Details JSON:

```
{  "stage": "IXSCAN",  "nReturned": 211,  "executionTime": 0,  "details": [    {      "index": "year_1",      "isKey": true,      "sortOrder": 1,      "isLast": true    }  ]}
```

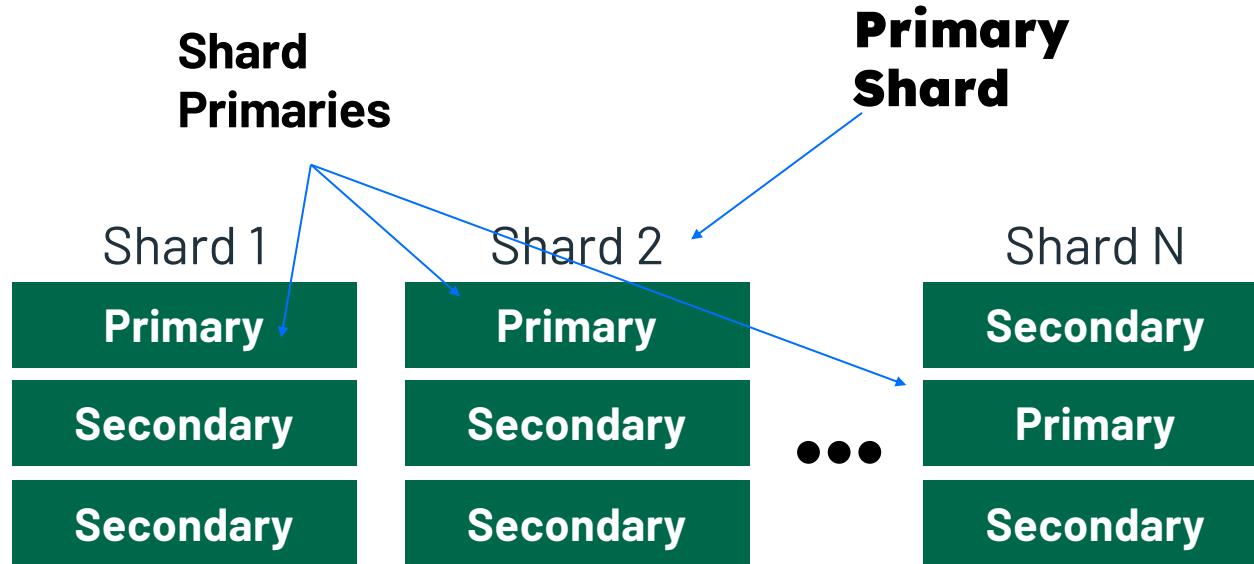


横向擴充第一步：Sharding





Primary Shards vs. Shard Primaries



Collections

Orders

Collections

Customers
Orders
Products
Suppliers
Prices

Collections

Orders



Shard keys

Shard key means two things

- Fields chosen to **partition data**
- Values on those fields in a given document

Rules for choosing a good shard key

- Included in **most of your queries**
- Reasonably **high cardinality**
- Ideally no more than 64MB of data to share a shard key
- Co-locates data you wish to retrieve together



Choosing shard keys in practice

In large collections, users work with subsets of the data

- User data - shard by a user, for example: bank accounts or games
- Departmental data - shard by department or branch
- Where there are no clear subsets, shard for parallelism
 - For instance, in an analytic data store, add a random value for the shard key



Steps to Shard a collection

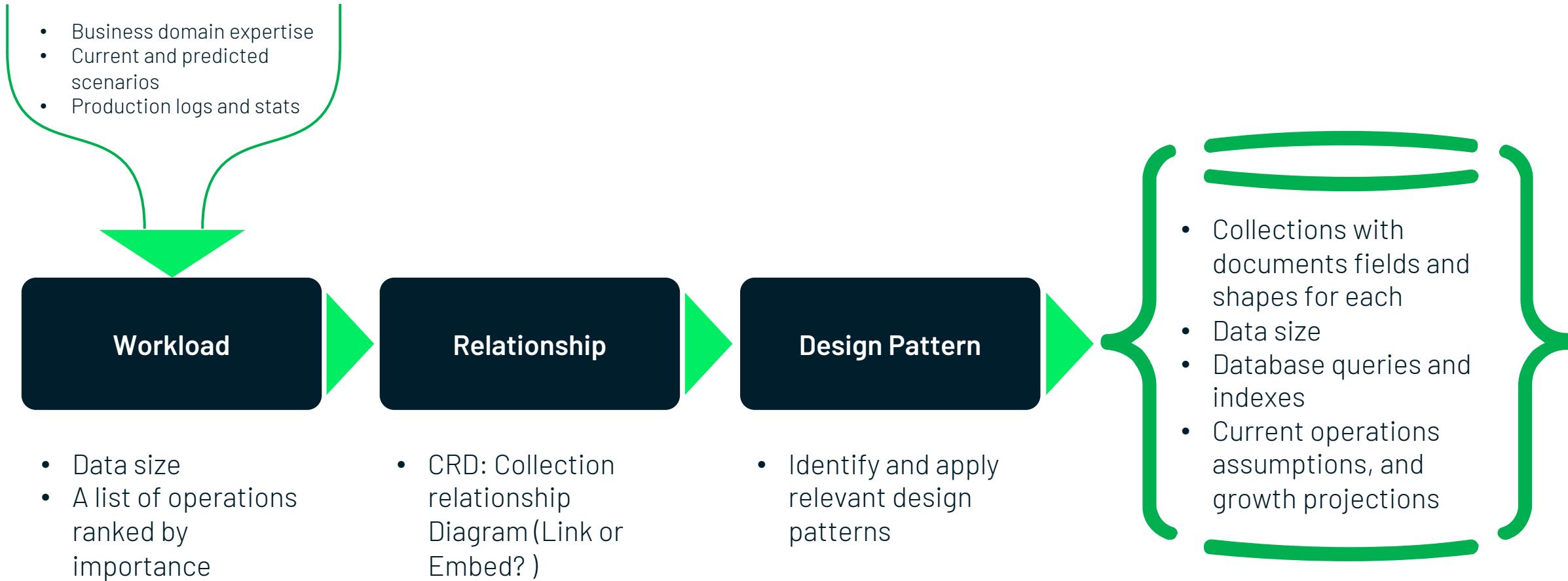
1. Ensure the cluster is a sharded cluster - Check using `sh.status()`
2. Decide on the shard key
3. Configure the database for sharding:
`sh.enableSharding("MyGameDB")`
4. Specify the shard key for the collection:
`sh.shardCollection("MyGameDB.players",
 {playerid:1, gametime:1 })`
5. Check status again to verify chunks are moving - `sh.status()`



{ 3: “Schema Design” }



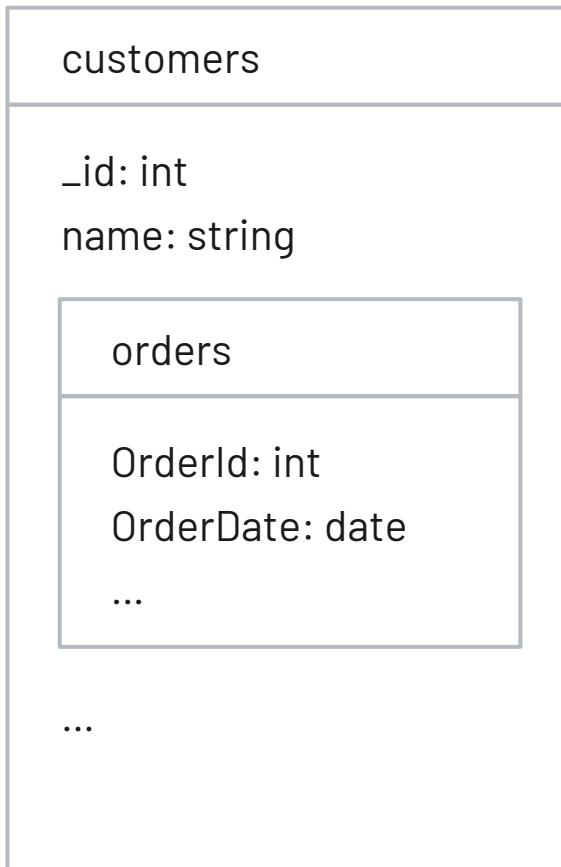
JSON Data Model Design Methodology



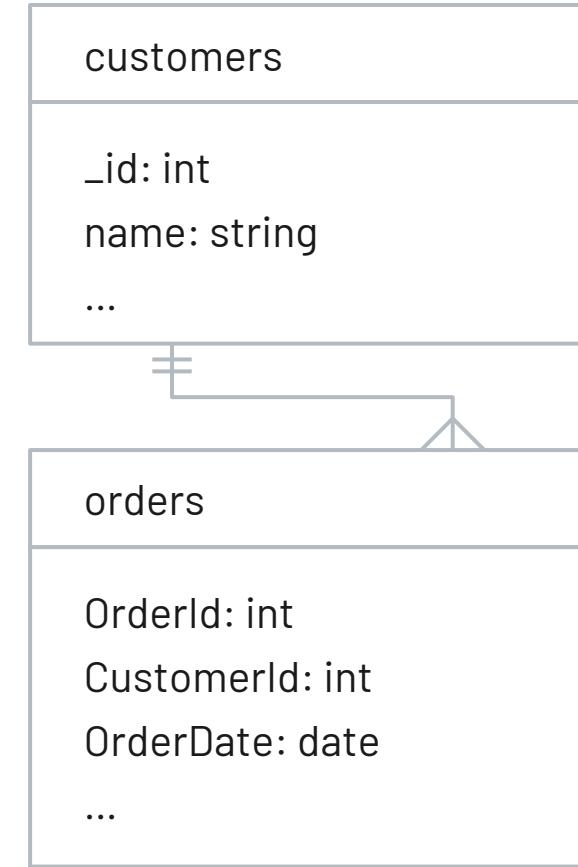


Embedding vs. Linking

Embedding



Referencing/Linking





Relationships

One-to-One (1-1)

One-to-Many (1-N)

Many-to-Many (N-N)

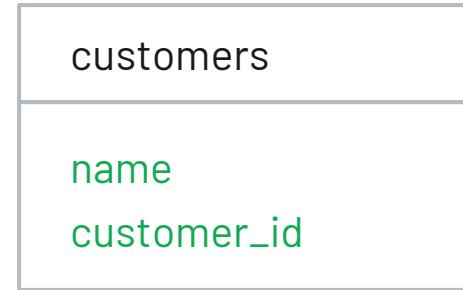


Relationships

One-to-One (1-1)

One-to-Many (1-N)

Many-to-Many (N-N)



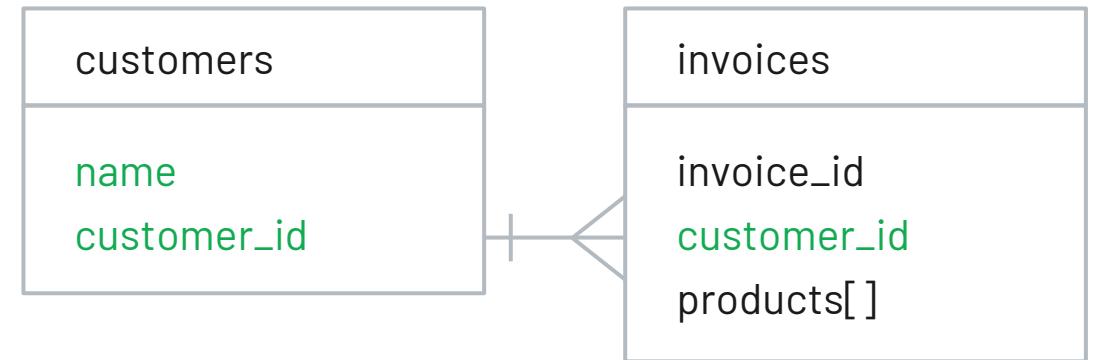


Relationships

One-to-One (1-1)

One-to-Many (1-N)

Many-to-Many (N-N)



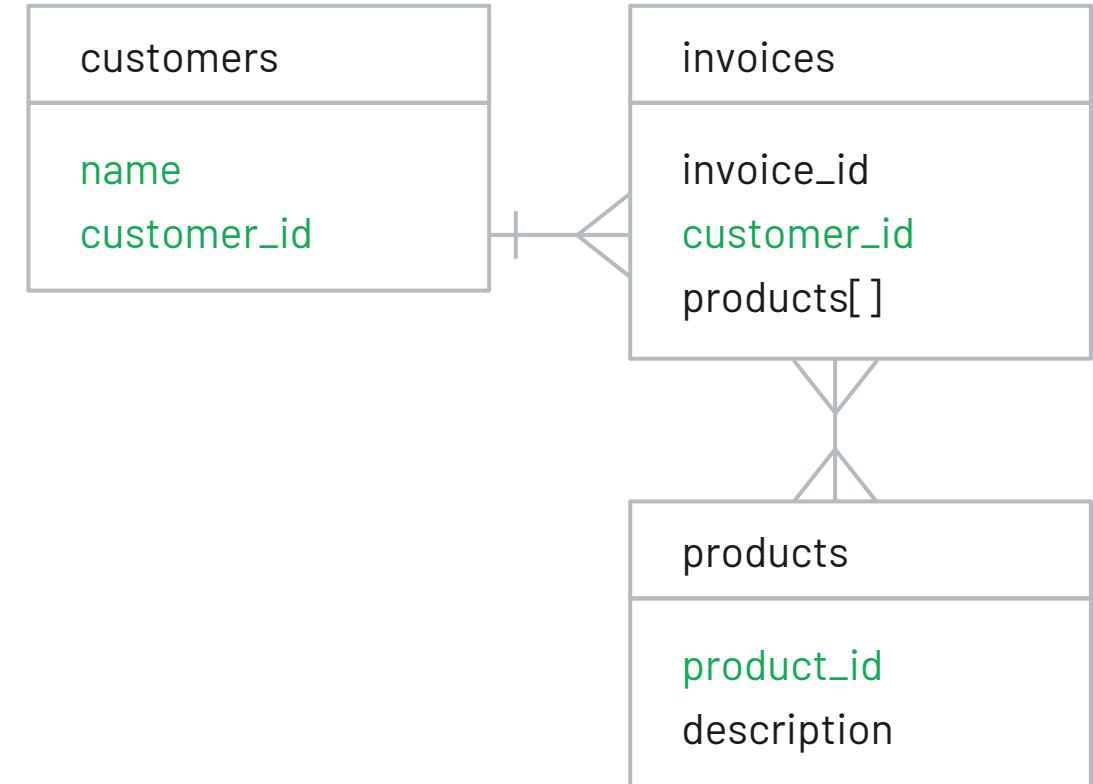


Relationships

One-to-One (1-1)

One-to-Many (1-N)

Many-to-Many (N-N)





Embed

For integrity with read operations

For integrity with write operations

On one-to-one and one-to-many

For data that is deleted together by default

Link

When the "many" side is a huge number

For integrity on write operations on many-to-many

When a piece is frequently used, but not the other and memory is an issue

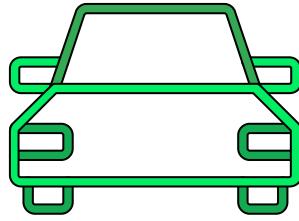


Patterns Overview

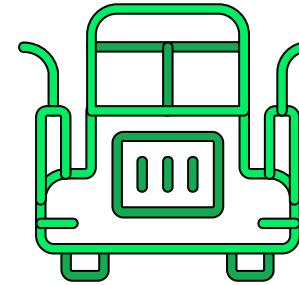
Use Cases	Patterns										Related Examples	
	Approximation	Attribute	Bucket	Computed	Document Versioning	Extended Reference	Outlier	Preallocated	Polymorphic	Schema Versioning	Subset	
Catalog	✓	✓		✓	✓	✓		✓	✓	✓	✓	Inventory Management
Content Management		✓			✓			✓	✓	✓	✓	
Internet Of Things	✓		✓	✓			✓	✓	✓			Log data, Time Series, Block Chain
Mobile	✓			✓		✓	✓	✓	✓	✓		
Personalization				✓	✓	✓	✓		✓	✓		Point of sale, User Management
Real-Time Analytics	✓		✓	✓		✓	✓		✓	✓		Data Warehouse
Single View		✓		✓	✓	✓		✓	✓			



Polymorphic Pattern



```
{  
  "vehicle_type": "car",  
  "owner": "Roland",  
  "taxes": "200",  
  "wheels": 4  
}
```



```
{  
  "vehicle_type": "truck",  
  "owner": "Daniel",  
  "taxes": "800",  
  "wheels": 10,  
  "axles": 3,  
}
```



Polymorphism Pattern

Problem

Objects more similar than different

Want to keep objects in same collection

Solution

Field tracks the type of document or sub-document

Application has different code paths per document type, or has subclasses

Use Case Examples

Single View

Product Catalog

Content Management

Benefits and Trade-offs

Easier to implement

Allow to query across a single collection



Attribute Pattern

```
{  
    title: "Star Wars",  
    director: "George Lucas",  
    ...  
    release_US:  
        ISODate ("1977-05-20T01:00:00+01:00"),  
    release_France:  
        ISODate ("1977-10-19T01:00:00+01:00"),  
    ...  
}  
  
{  
    title: "Star Wars",  
    director: "George Lucas",  
    ...  
    releases:[  
        {  
            location: "US",  
            Date: ISODate ("1977-05-20T01:00:00+01:00")  
        }, {  
            location: "France",  
            Date: ISODate ("1977-05-20T01:00:00+01:00")  
        },  
    ]...  
}
```



Attribute Pattern

Problem

Lots of similar fields

Want to search across many fields at once

Fields present in only a small subset of documents

Solution

Break the field/value into a sub-document

```
{ "color": "blue", "size": "large" }  
[ { "k": "color", "v": "blue" },  
{ "k": "size", "v": "large" } ]
```

Use Case Examples

Characteristics of a product

Set of fields all having same value type

List of dates

Benefits and Trade-offs

Easier to index

Allow for non-deterministic field names

Ability to qualify the relationship of the original field and value

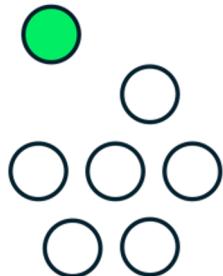


Subset Pattern: When and Why it is Needed?

Used when **a large portion of data** inside a document is **rarely needed**.

- Examples of when data is not part of the majority of the queries include product reviews, article comments, or cast in a movie.

The Subset Pattern is the solution to refactoring schemas with this characteristic.





Subset Pattern

```
{  
  "name": "MongoDB- The Definitive Guide",  
  "edition": "3",  
  "description": "Learn all about MongoDB",  
  "price": { "value":NumberDecimal(29.99), "currency": "USD" },  
  "reviews": [ { rid: 767, username: "eliot_h", review: "Great introduction to  
    MongoDB", date: ISODate("2020-09-09")},  
    { rid: 766, username: "dwight_m", review: "Fantastic overview of  
    MongoDB", date: ISODate("2020-09-08")}, ...  
    { rid: 1, username: "kevin_r", review: "Nice introduction to  
    MongoDB", date: ISODate("2020-01-06")}  
}
```



Subset Pattern: Split the Data

```
{ review_id: 740, product_id:  
  ObjectId("5f2aefa8fde88235b959f0b1e") ,  
  review_author: "ken_a", review_date: ISODate("2020-08-08") ,  
  review_text: "Nice book, great topics!"} ,  
  
{ review_id: 739, product_id:  
  ObjectId("5f2aefa8fde88235b959f0b1e") ,  
  review_author: "matt_j", review_date: ISODate("2020-08-06") ,  
  review_text: "Fantastic book, learnt lots."} ,  
  
{ review_id: 738, product_id:  
  ObjectId("5f2aefa8fde88235b959f0b1e") ,  
  review_author: "sonalim", review_date: ISODate("2020-07-06") ,  
  review_text: "Comprehensive MongoDB coverage"}
```



Subset Pattern

Problem

Working set is too big

Lot of pages are evicted from memory

A large part of documents is rarely needed

Solution

Split the collection in 2 collections

Most used part of documents

Less used part of documents

Duplicate part of a 1-N or N-N relationship that is often used in the most used side

Use Case Examples

List of reviews for a product

List of comments on an article

List of cast in a movie

Benefits and Trade-offs

Smaller working set, used documents are smaller

Shorter disk access for bringing in additional documents from the most used collection

Can add more round trips to the server

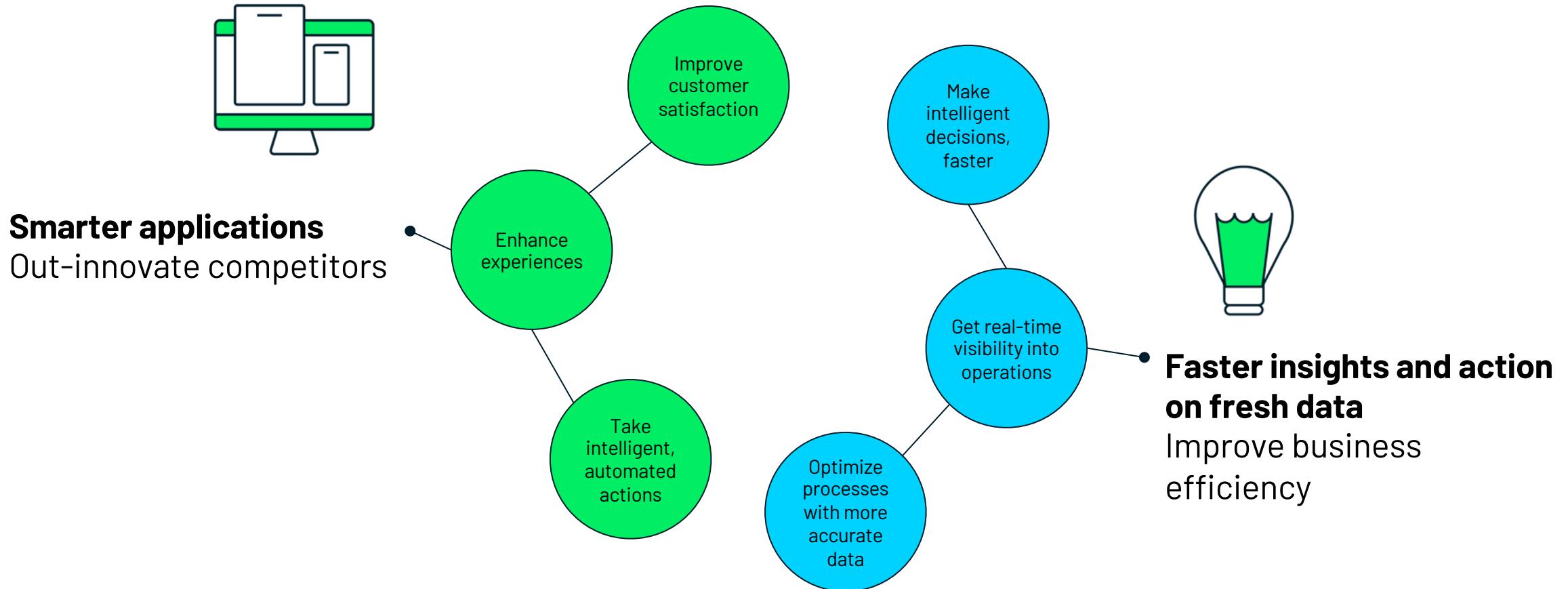
A little more space used on disk

{ 4: “Event-driven Architecture Design” }



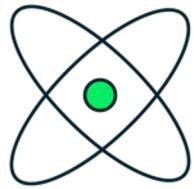


數位轉型帶來的競爭優勢



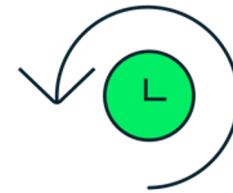


即時分析的應用特性



Live app data

<5 seconds old
Constantly updated



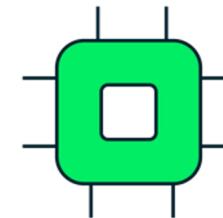
In the app flow

<1 second response
latency



At app scale

Supports thousands+
concurrent users

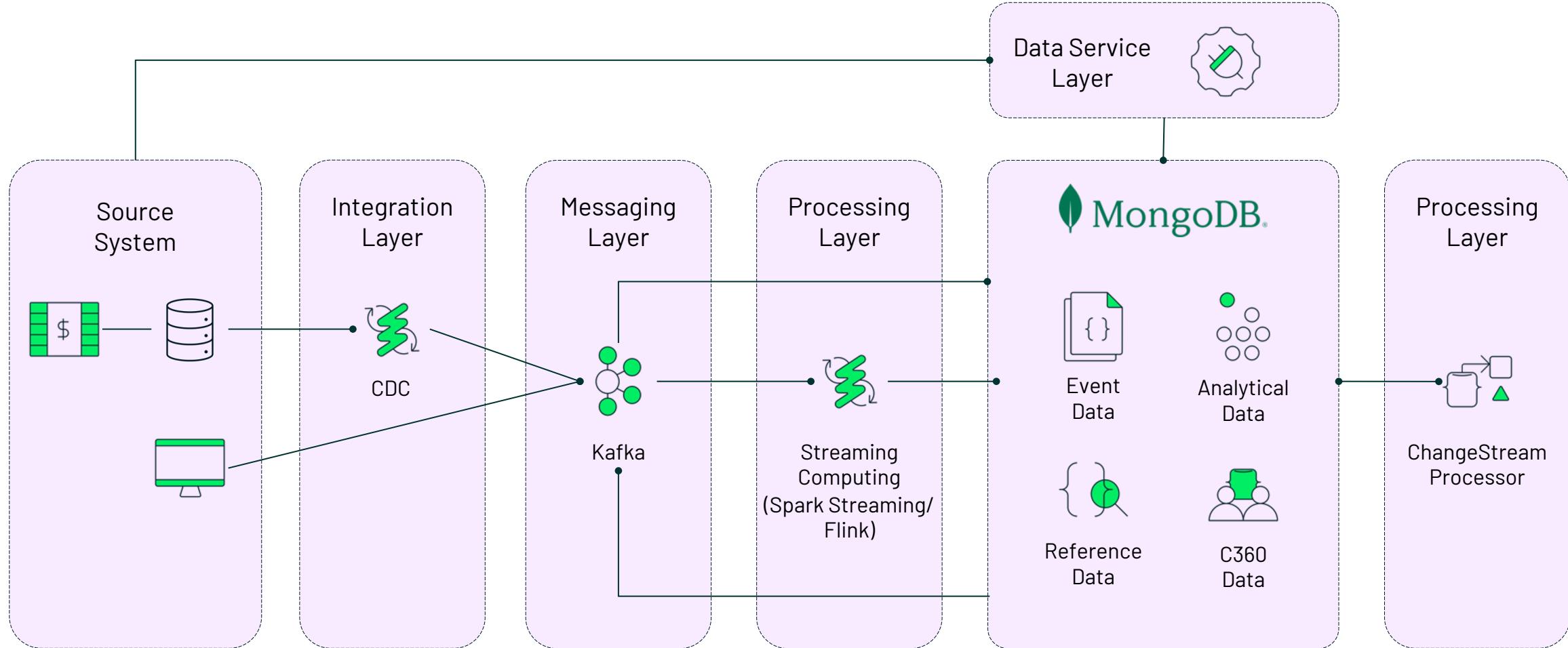


App automation

Analytics consumed by
software



常見的 Event-driven Architecture





Messaging Layer 的關鍵元件 - Kafka

Cluster

One or more servers (brokers) that run Kafka

Topic

Category/feed name to which messages are stored and published

Message

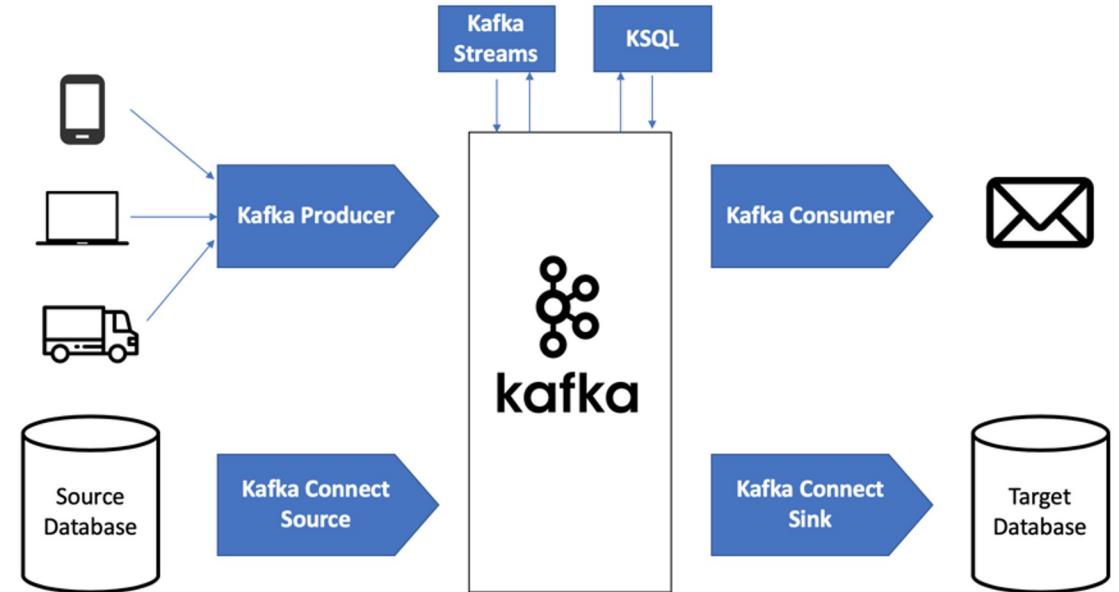
Byte arrays that can store any object in any format

Producer

Writes messages to Kafka topic(s)

Consumer

Reads messages from Kafka topic(s)



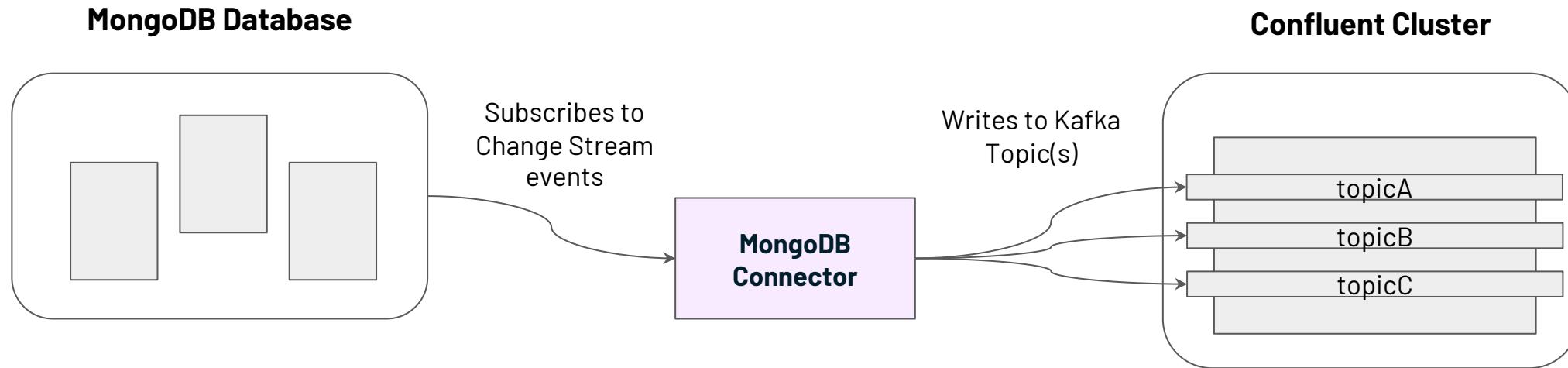


MongoDB Connector for Apache Kafka

- Enables users to easily integrate MongoDB with Kafka
- Users can configure **MongoDB as a source** to publish data changes from MongoDB into Kafka topics for streaming to consuming applications
- Users can configure **MongoDB as a sink** to easily persist events from Kafka topics directly to MongoDB collections
- Available from [Confluent Hub](#) and Verified Gold
- Certified against **Apache Kafka 2.3** and **Confluent Platform 5.3 (or later)**



MongoDB Source



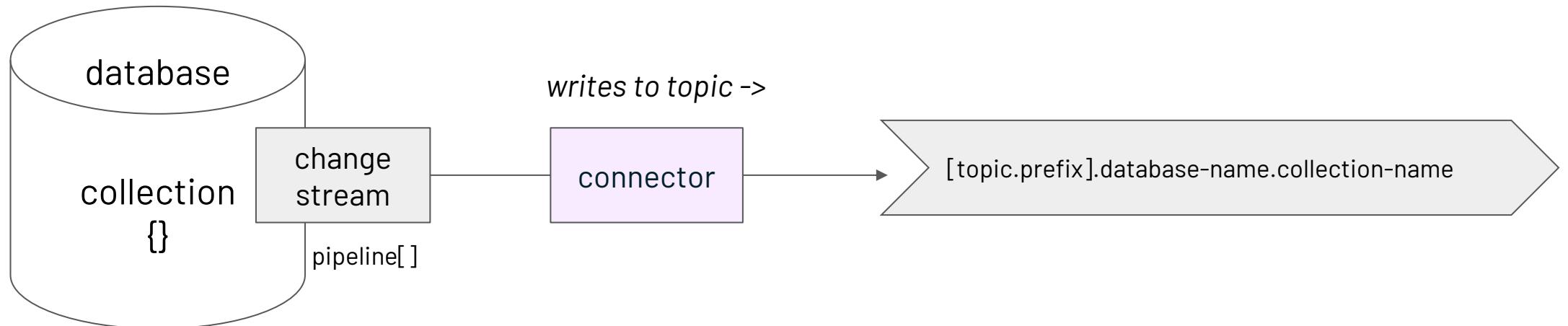
When MongoDB is a Source...

Subscribes to change stream events and writes to Kafka topic(s)



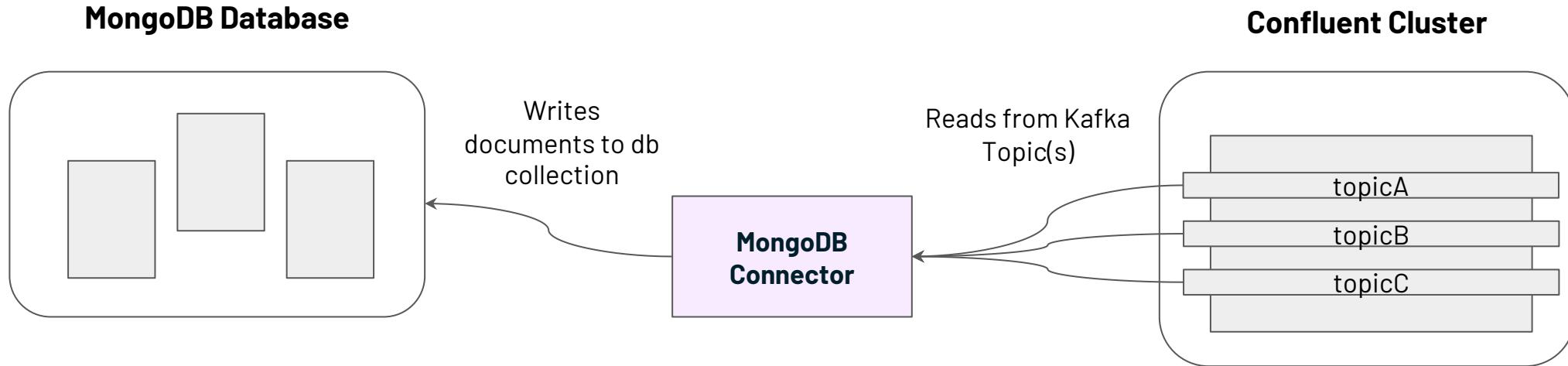
MongoDB Source: Writing a Topic

- Writes to topic based on database and collection name
- Optionally specify pipeline to manage change stream events to watch
- Optionally set a topic.prefix in the connector configuration





MongoDB Sink



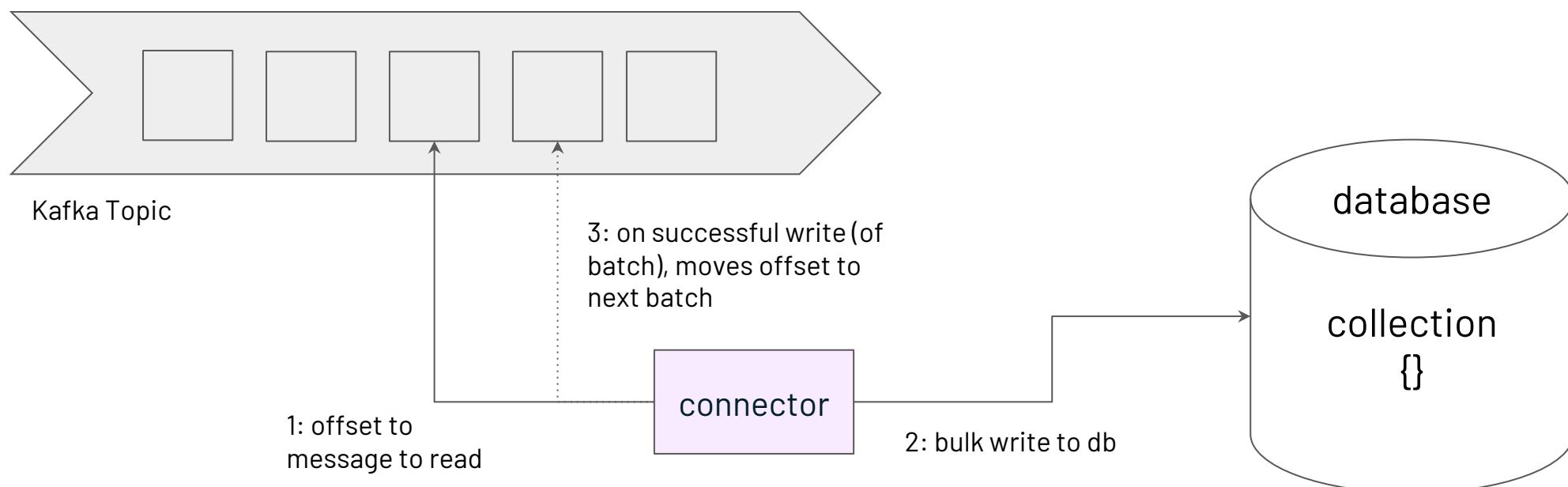
When MongoDB is a Kafka Sink...

Reads from Kafka topic(s) and writes documents to collection



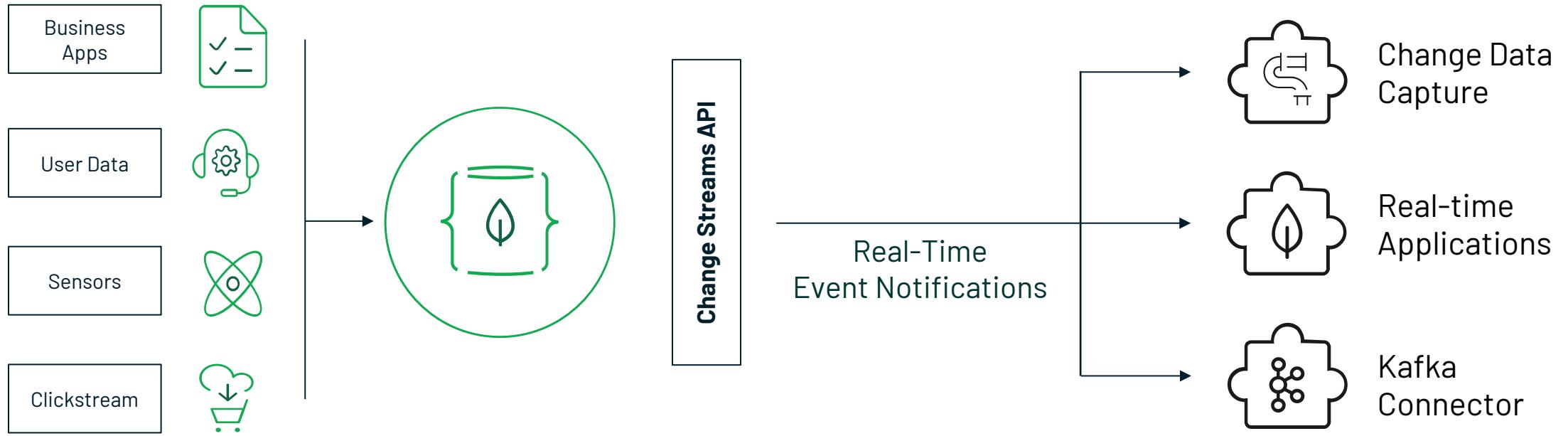
MongoDB Sink: Reading Messages from Topics

- Reads messages from topic (based on pointer to message in topic)
- Writes message into MongoDB database collection
- Moves pointer to next message based on write to database





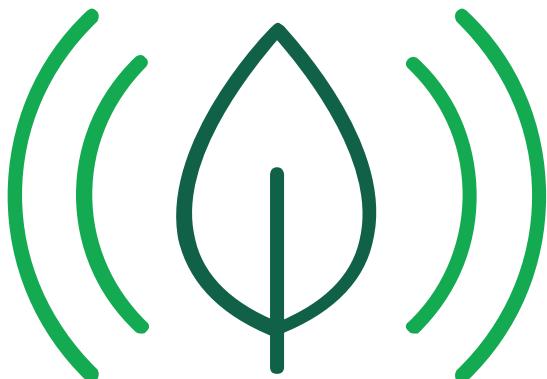
MongoDB Change Streams



Enabling developers to build
reactive, real-time services
in a scalable, testable way



Characteristics of Change Streams



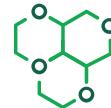
Targeted Changes - Filter change events to specific event types, and fields



Resumability - Automatically and safely resume a change stream using resume tokens



Total Ordering - Guaranteed ordering of all changes (even for a sharded cluster)



Durability - Only see changes relating to events which have been durably (majority) committed to the database across



Security - Ensure change streams fit into the existing enterprise grade role-based access control model



Ease of use - Change streams are familiar and take advantage of the existing MongoDB drivers and query language



Idempotence - Change events are structured so that they are safe to apply multiple times while still reaching the same consistent state



Okay, but how do I
actually do it?

I bet setting up Change
Streams is really hard...

Opening a Change Stream (Python)

```
# Define the MongoDB Connection
conn=pymongo.MongoClient("mongodb://localhost:27017/demo?replicaSet=rs")
db = conn["demo-db"]
collection = db["demo-coll"]

# Define the change_stream object
change_stream = collection.watch()

# Execute logic for every change in the stream
for change in change_stream:
    # Do business logic here (print is just an example)
    print(change)
```



Opening a Targeted Change Stream (Python)

```
# Define the MongoDB Connection
conn=pymongo.MongoClient("mongodb://localhost:27017/demo?replicaSet=rs")
db = conn["demo-db"]
collection = db["demo-coll"]

# Define the change_stream object
change_stream = collection.watch([{"$match":
                                    {"fullDocument.payment_value": {"$gt": 10000}}])

# Execute logic for every change in the stream
for change in change_stream:
    # Do business logic here (print is just an example)
    print(change)
```





考試時間！！！





Survey

Thank you for
your time.

