IN-PERSON

SATURDAY | OCT 26 | 1400 AM - 1700 PM (ICT)

MongoDB Mini Meetup: MongoDB Data Modeling and Query Optimization

MongoDB®
User Group, Thailand

Piti Champeethong
Senior Consulting Engineer
MongoDB User Group Leader

Agenda

- Core Topologies
  - Replication
  - Sharding
- Core Data Modeling
  - Computed pattern
  - Inheritance pattern
  - Extended reference pattern
  - Schema versioning pattern
  - Subset pattern
  - Bucket pattern

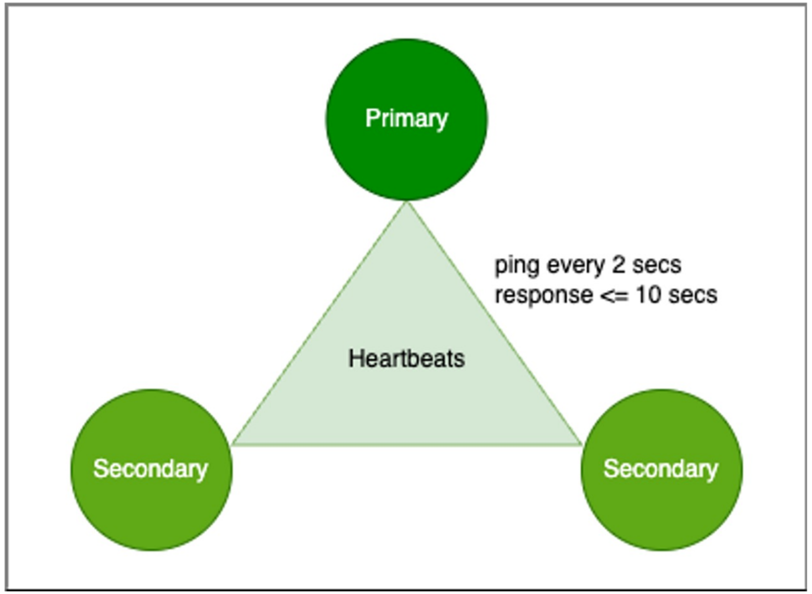
Preparation

- https://www.mongodb.com/docs/atlas/cli/current/install-atlas-cli/
- curl https://atlas-education.s3.amazonaws.com/sampledata.archive -o sampledata.archive
- https://www.mongodb.com/try/download/compass

MongoDB - Document database

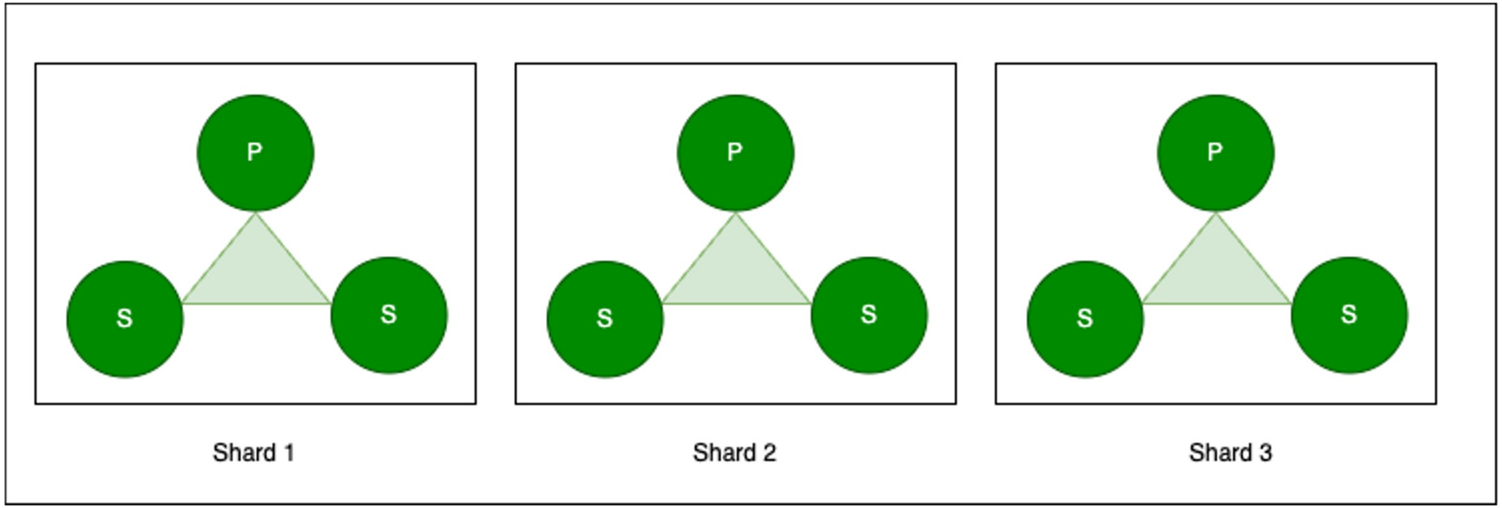
- Advantages
  - Full cloud-based developer data platform (Atlas)
  - Flexible document schemas
  - Widely supported and code-native data access
  - Change-friendly design
  - Powerful querying and analytics
  - Easy horizontal scale-out with sharding
  - Simple installation
- Disadvantages
  - Easy to get mistake by misunderstanding.

Replication



- Only primary node can accept write operations
- All nodes can accept read operations
- Maximum 7 nodes are vote member nodes.
- High Availability (HA)

Sharding



- Horizontal Scaling
- Collection sharding

Computed Pattern

```
{
   "_id": 1,
   "product_name": "Laptop",
   "reviews": [
      {
         "user": "Alice",
         "comment": "Great laptop",
         "stars": 5,
         "date": { "$date": "2021-01-01T12:30:00Z" }
      },
      {
         "user": "Bob",
         "comment": "Good laptop",
         "stars": 4,
         "date": { "$date": "2021-02-01T12:30:00Z" }
      }
   ],
   "computed_avg_rating": 4.5,
   "total_reviews": 2
}
```

Computed Pattern

Problem	• Documents are more similar than different • Need to query the documents on their similitudes.
Use cases	• Internet of Things • Event sourcing • E-commerce

Inheritance Pattern

```
{
    "_id": 1,
    "type": "parent",
    "name": "John Doe",
    "age": 40
},
{
    "_id": 2,
    "type": "child",
    "name": "Jane Doe",
    "age": 10,
    "parent": 1
}
```

```
{
    "_id": 1,
    "patient_id": "P0001",
    "type": "surgical",
    "surgery_name": "Appendectomy",
    "surgeon": "Dr. Smith",
    "date": { "$date": "2024-01-01T10:00:00Z" }
},
{
    "_id": 2,
    "patient_id": "P0001",
    "type": "dental",
    "dentist": "Appendectomy",
    "procedure": "Tooth Extraction",
    "date": { "$date": "2024-02-01T10:00:00Z" }
}
```

Inheritance Pattern

Problem	• Documents are more similar than different • Need to query the documents on their similitudes.
Use cases	• Single View • Product Catalog • Content Management • Mobile Application

Extended Reference Pattern

```
{
   "_id": {
      "$oid": "66027d48d57fb8ef29b4bb33"
   },
   "product_name": "Laptop ABC",
   "price": 34000,
   "specs": {
      "processor": "Intel Core i3 7th Gen",
      "ram": "4 GB DDR4",
      "storage": "1 TB HDD",
      "os": "Windows 10 Home"
   }
}
```

products collection

```
{
   "_id": { "$oid": "66027daed57fb8ef29b4bb34" },
   "customer_id": "C7990",
   "order_date": { "$date": "2024-02-01T10:30:00.000Z"
},
   "products": [
      {
         "product_id": {
            "$oid": "66027d48d57fb8ef29b4bb33"
         },
         "product_name": "Laptop ABC",
         "price": 34000,
         "quantity": 1
      }
   ]
}
```

orders collection

Extended Reference Pattern

Problem	• Too many joins in read operations • Embedding leads to document that are too big (16MB)
Use cases	• Catalog. • Real-time analytics. • Mobile Application. • E-commerce

Schema Versioning Pattern

```
{
    "_id": {
        "$oid": "66027d48d57fb8ef29b4bb33"
    },
    "username": "Solo",
    "email": "solo@company.mail",
    "schema_version": 1
}
```

```
{
    "_id": {
        "$oid": "66027daed57fb8ef29b4bb34"
    },
    "username": "Sanji",
    "email": "sanji@company.mail",
    "token": "66027daed57fb8ef29b4bb34",
    "token_expired": {
                "$date": "2029-02-01T10:30:00.000Z"
     },
    "schema_version": 2
}
```

Schema Versioning Pattern

Problem	• Doing a schema migration without downtime.
Use cases	• Any application that can’t sustain any downtimes

Subset Pattern

```
{
  "_id": {
    "$oid": "66027d48d57fb8ef29b4bb33"
  },
  "title": "MongoDB 101",
  "author": "jojo hakusho",
  "subset_reviews": [
    { "rev_id": 1, "user": "Bob", "rating": 10 },
    { "rev_id": 2, "user": "Alice", "rating": 9 },
    { "rev_id": 3, "user": "Mike", "rating": 8 },
    { "rev_id": 4, "user": "Koi", "rating": 8 },
    { "rev_id": 5, "user": "Henry", "rating": 8 }
  ]
}
```

books collection

```
[
    { "rev_id": 6, "user": "Mee", "rating": 10 },
    { "rev_id": 7, "user": "Maew", "rating": 9 },
    { "rev_id": 8, "user": "Kai", "rating": 8 },
    { "rev_id": 9, "user": "Pop", "rating": 8 },
    { "rev_id": 10, "user": "Pae", "rating": 8 },
    { "rev_id": 11, "user": "Bow", "rating": 10 },
    { "rev_id": 12, "user": "Row", "rating": 9 },
    { "rev_id": 13, "user": "Mai", "rating": 8 },
    { "rev_id": 14, "user": "Maa", "rating": 8 },
    { "rev_id": 15, "user": "Leo", "rating": 8 }
]
```

reviews collection

Subset Pattern

Problem	• Large documents are taking up a lot of space in memory.
Use cases	• List of reviews. • List of comments. • A long list of nearly anything kept in an array

Bucket Pattern

```
{
   "_id": {
      "$oid": "66027d48d57fb8ef29b4bb33"
   },
   "sensor_id": "S123",
   "date": 20240203,
   "readings": [
      { "value": 22, "ts": { "$date": "2024-02-03T08:00:01Z" },
      { "value": 20, "ts": { "$date": "2024-02-03T08:05:01Z" },
      { "value": 18, "ts": { "$date": "2024-02-03T08:10:00Z" },
      { "value": 23, "ts": { "$date": "2024-02-03T08:15:01Z" },
      { "value": 22, "ts": { "$date": "2024-02-03T08:20:01Z" },
   ],
  "count": 5
}
```

Bucket Pattern

Problem

- Avoiding too many documents or documents too big.
- A one-to-many relationship that can’t be embedded.

Use cases

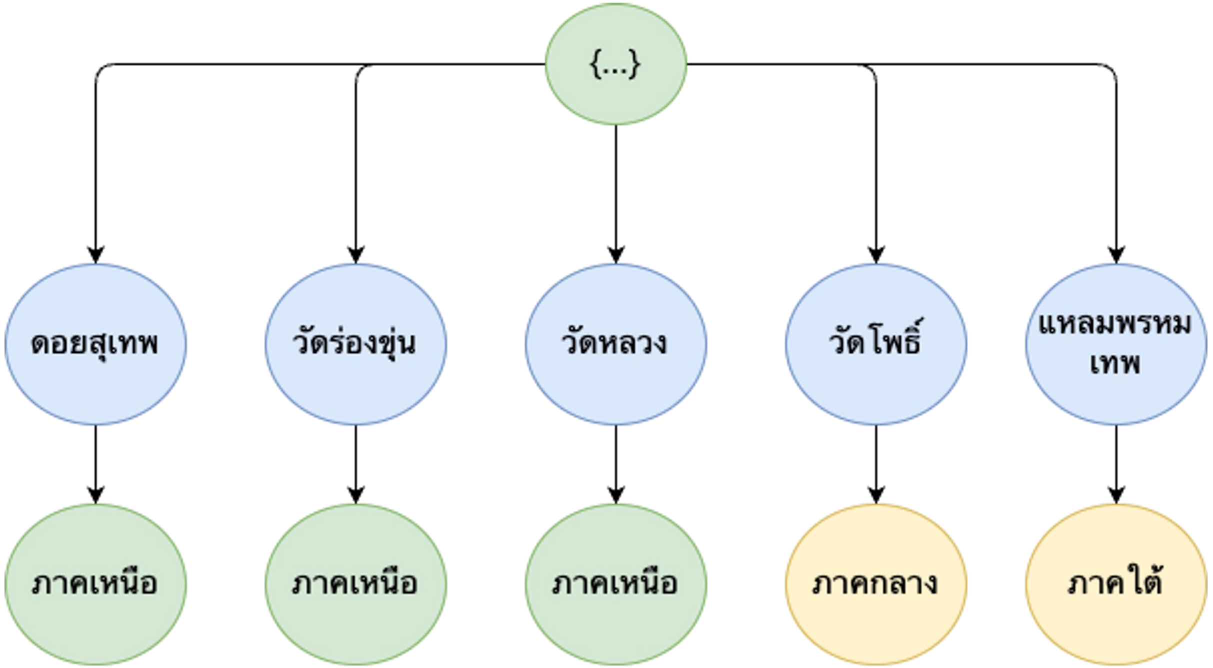
- Internet of Things.
- Data Warehouse.
- One-to-many relationships with high cardinality. (A large number of different values)

(E)quality
(S)ort
(R)ange

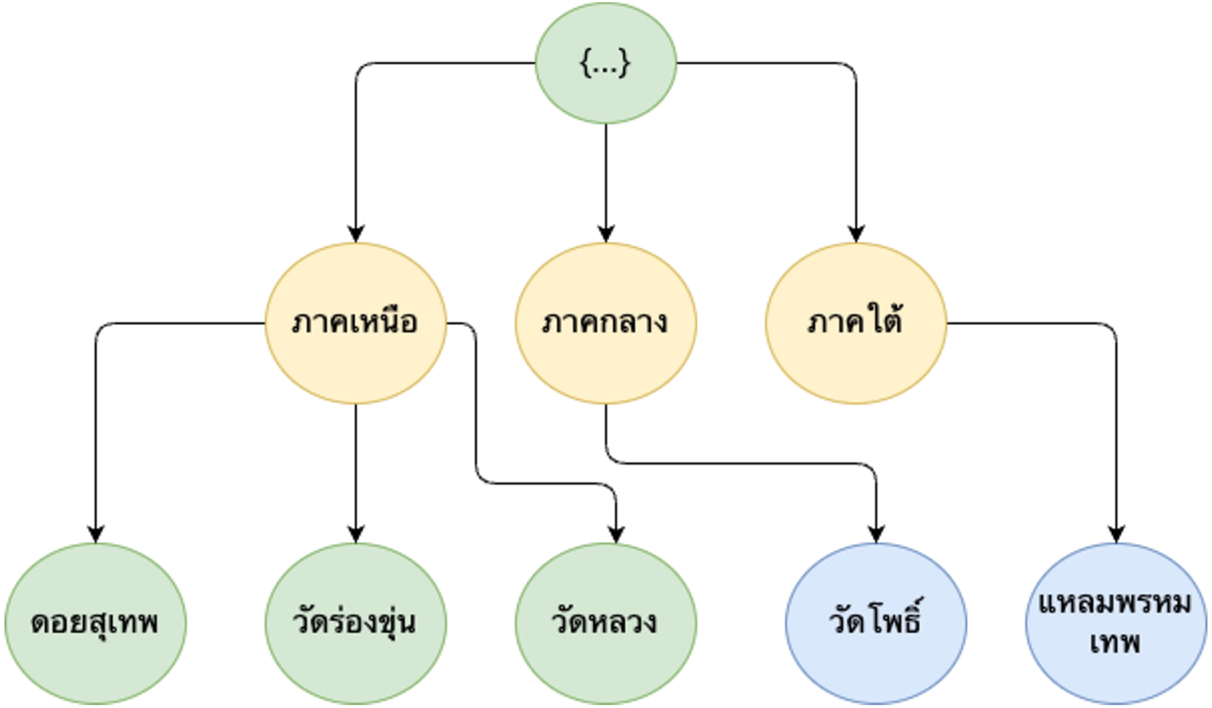
E-S-R

- Equality before Sort
- Equality before Range
- Sort before Range

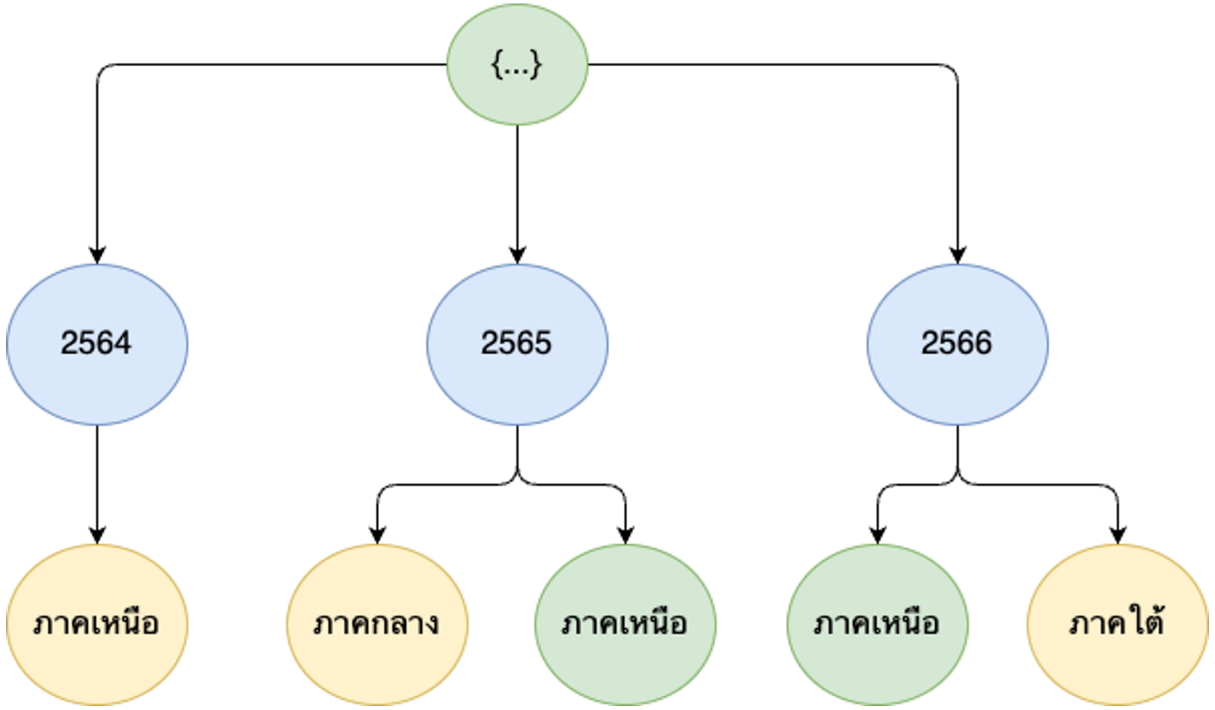
E-S-R - Good


{...}
ดอยสุเทพ
วัดร่องขุ่น
วัดหลวง
วัดโพธิ์
แหลมพรหมเทพ
ภาคเหนือ
ภาคเหนือ
ภาคเหนือ
ภาคกลาง
ภาคใต้

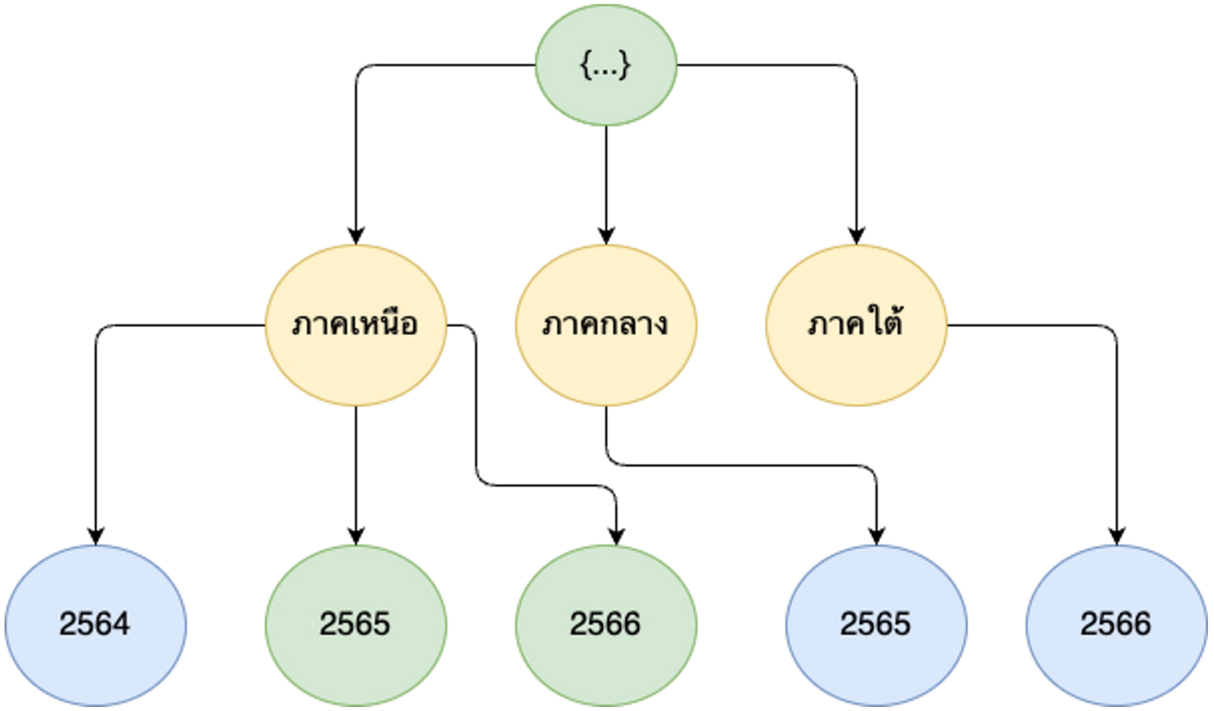
E-S-R - Better


{...}
ภาคเหนือ
ภาคกลาง
ภาคใต้
ดอยสุเทพ
วัดร่องขุ่น
วัดหลวง
วัดโพธิ์
แหลมพรหมเทพ

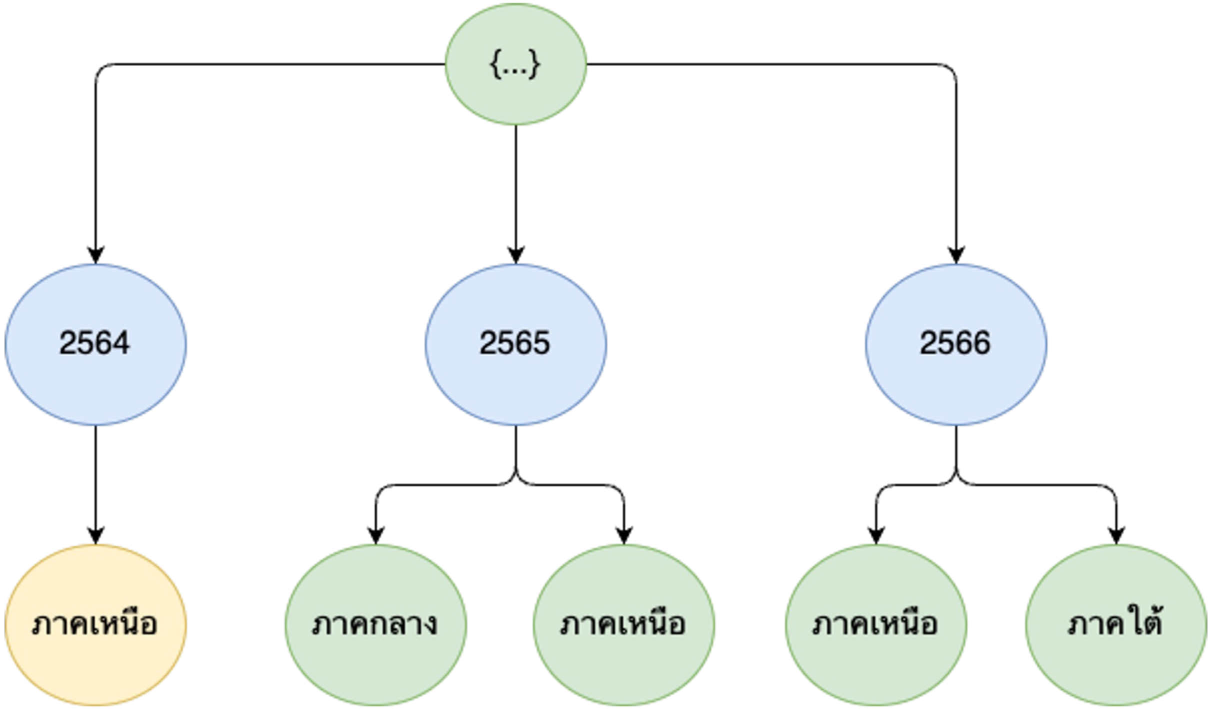
E before R – Good



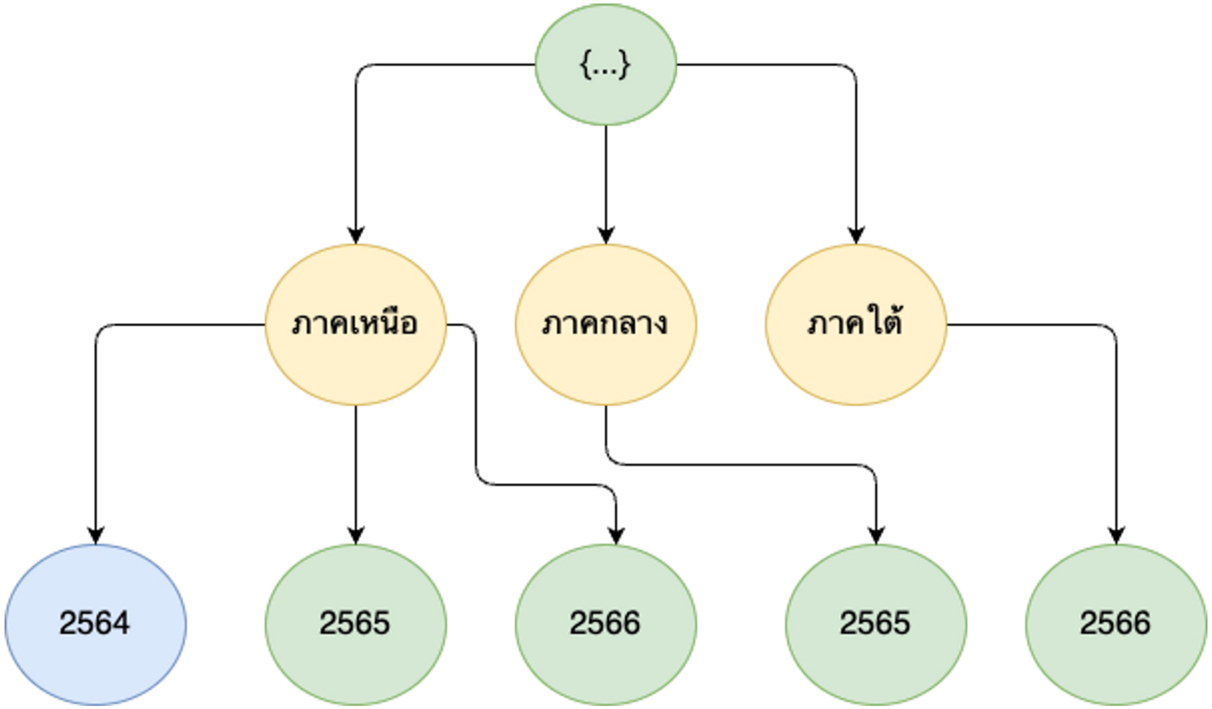
E before R - Better



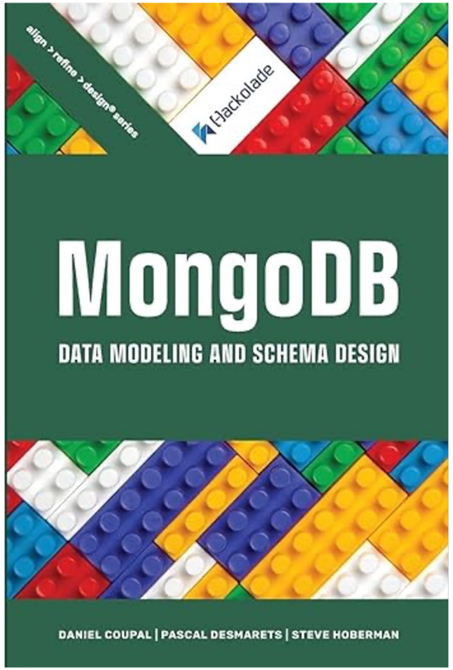
S before R – Good



S before R - Better



Book recommendations

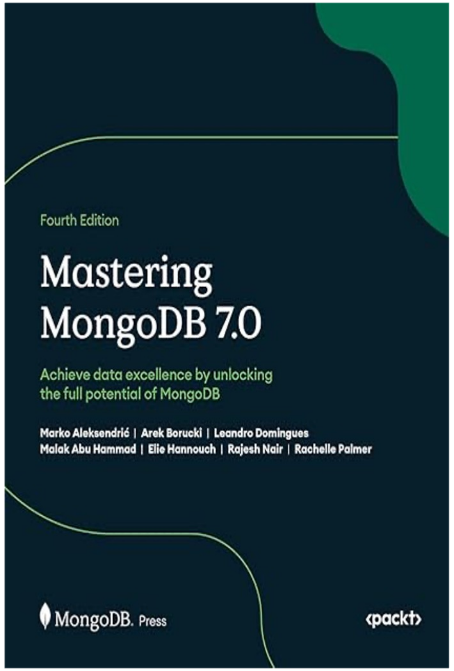

align > refine > design® series

Hackolade

MongoDB

DATA MODELING AND SCHEMA DESIGN

DANIEL COUPAL | PASCAL DESMARETS | STEVE HOBERMAN


Fourth Edition

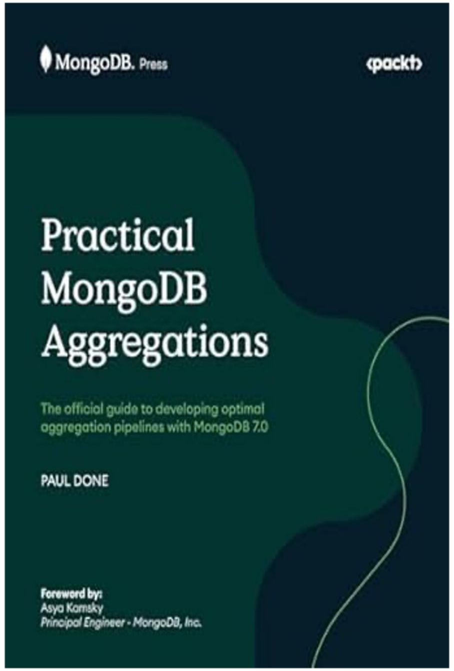
Mastering MongoDB 7.0

Achieve data excellence by unlocking the full potential of MongoDB

Marko Aleksendrić | Arek Borucki | Leandro Domingues
Malak Abu Hammad | Elie Hannouch | Rajesh Nair | Rachelle Palmer

MongoDB. Press

packt


MongoDB. Press

packt

Practical MongoDB Aggregations

The official guide to developing optimal aggregation pipelines with MongoDB 7.0

PAUL DONE

Foreword by:
Asya Kamsky
Principal Engineer - MongoDB, Inc.

References

- https://www.geopits.com/blog/mongodb-data-modeling-design-patterns.html
- https://www.mongodb.com/blog/post/new-data-modeling-learning-path-certification
- https://github.com/mongodbthailand/thmug-esr
- https://www.mongodb.com/developer/books