DAY 2 – 5 DOCUMENT MODEL VS. MONGODB

Emerging Data Modeling and Management Technology:

Key-Value and Document Databases

DS&AI : PROFESSIONAL TRAINING COURSE

1 – 3 APRIL 2021 | ASIAN INSTITUTE OF TECHNOLOGY

CHUTIPORN ANUTARIYA | CHUTIPORN AT AIT DOT AC DOT TH

Co-funded by the Erasmus+ Programme of the European Union

mongoDB

- Stores data in **JSON-like documents**.
- **Fields can vary** from document to document and data structure can be changed over time
- Distributed database, high availability, horizontal scaling, and geographic distribution
- Scalability
  - Performance Scale: Sustaining 100,000+ database read and writes per second while maintaining strict latency SLAs
  - Data Scale: Storing 1 billion+ documents in the database
  - Cluster Scale: Distributing the database across 100+ nodes, in multiple data centers

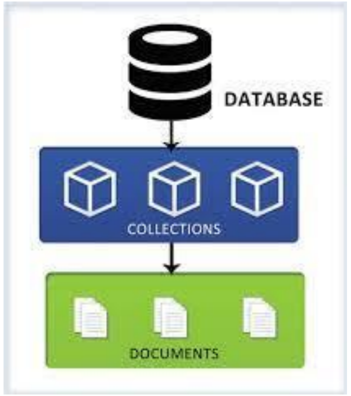
Getting familiarization with MongoDB on cloud (Atlas)

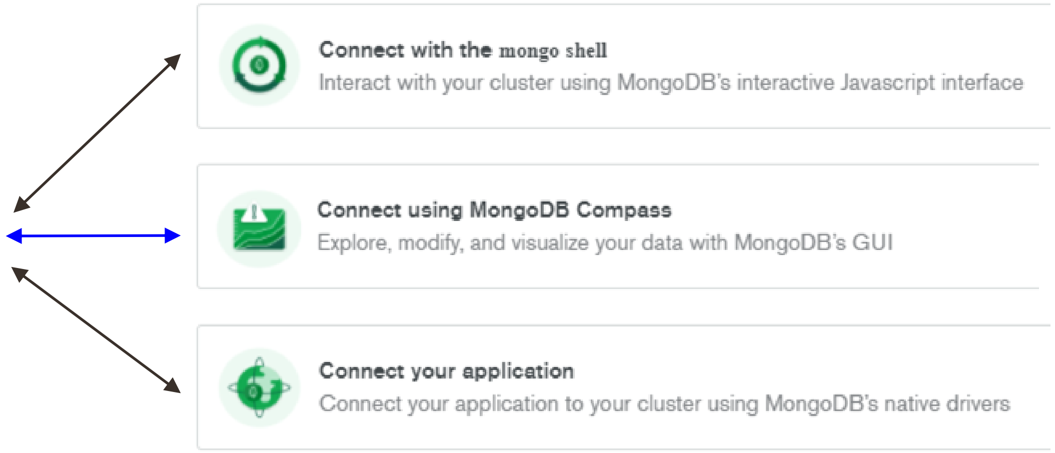
MongoDB Practice Architecture

MongoDB Atlas Cloud Database

MongoDB Clients

- Cluster0 (DBServer)
  - DSAIdb
    - sales
      - june@gmail.com buy 3 backpacks
    - customers
      - june@gmail.com
    - products
      - backpack



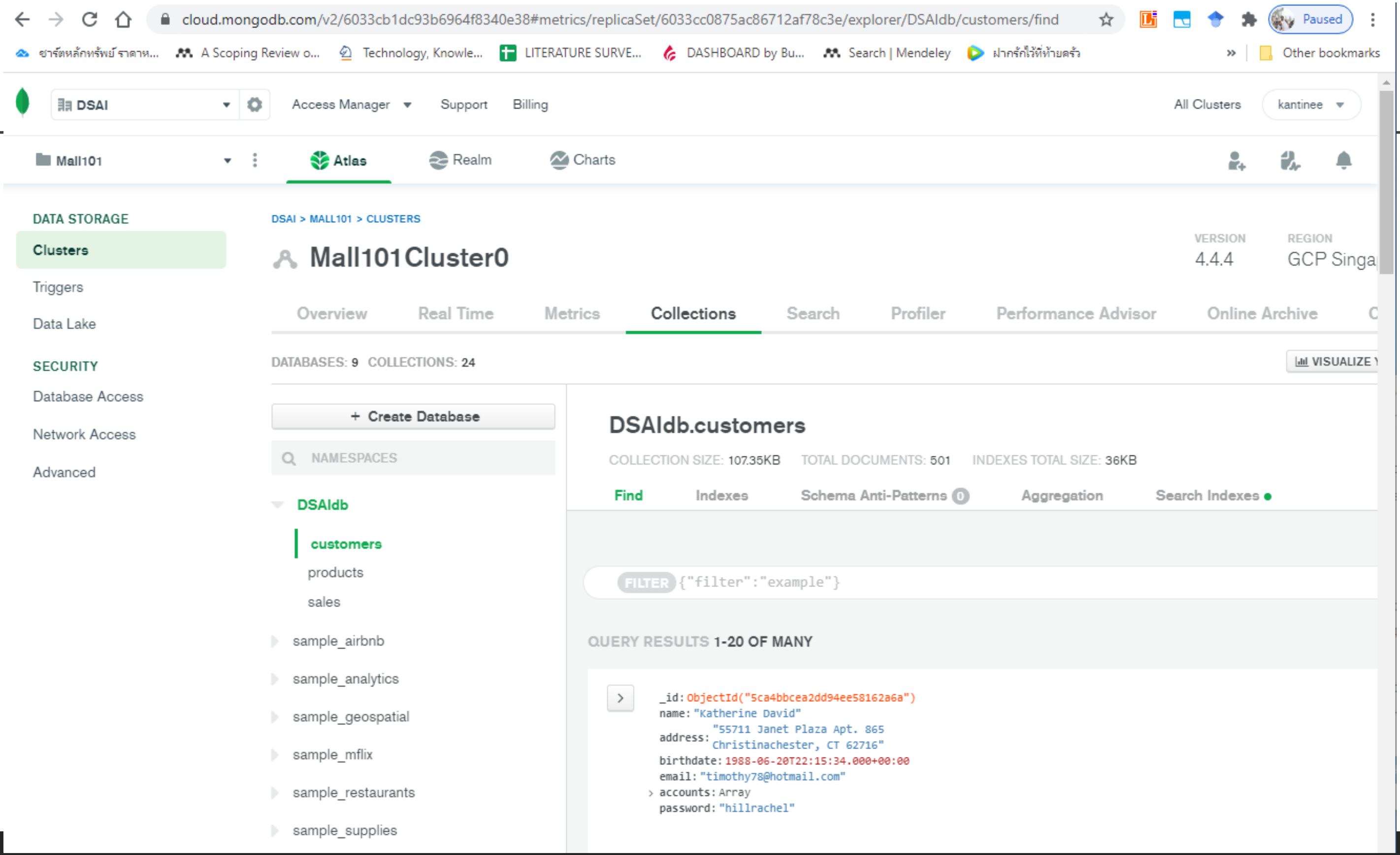


Connect with the mongo shell
Interact with your cluster using MongoDB's interactive Javascript interface

Connect using MongoDB Compass
Explore, modify, and visualize your data with MongoDB's GUI

Connect your application
Connect your application to your cluster using MongoDB's native drivers

MongoDB Atlas



cloud.mongodb.com/v2/6033cb1dc93b6964f8340e38#metrics/replicaSet/6033cc0875ac86712af78c3e/explorer/DSAIdb/customers/find

DSAI > MALL101 > CLUSTERS

Mall101Cluster0

VERSION 4.4.4 REGION GCP Singa

DATABASES: 9 COLLECTIONS: 24

DSAIdb.customers

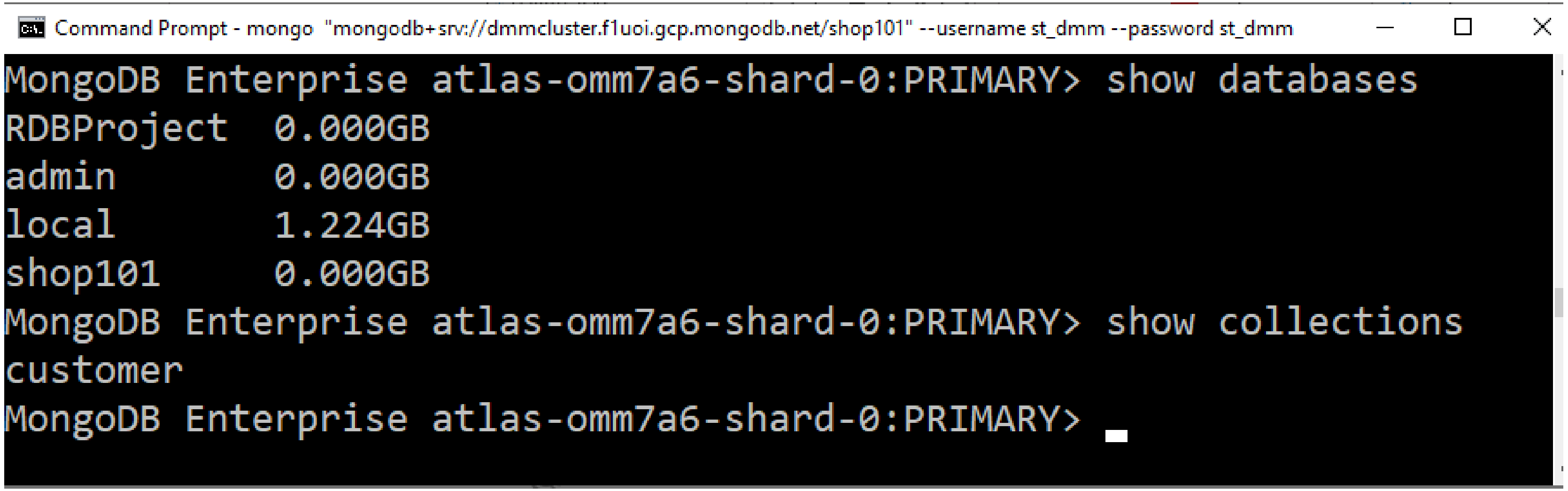
COLLECTION SIZE: 107.35KB TOTAL DOCUMENTS: 501 INDEXES TOTAL SIZE: 36KB

{"filter":"example"}

QUERY RESULTS 1-20 OF MANY

```
_id: ObjectId("5ca4bbcea2dd94ee58162a6a")
name: "Katherine David"
address: "55711 Janet Plaza Apt. 865
          Christinachester, CT 62716"
birthdate: 1988-06-20T22:15:34.000+00:00
email: "timothy78@hotmail.com"
> accounts: Array
password: "hillrachel"
```

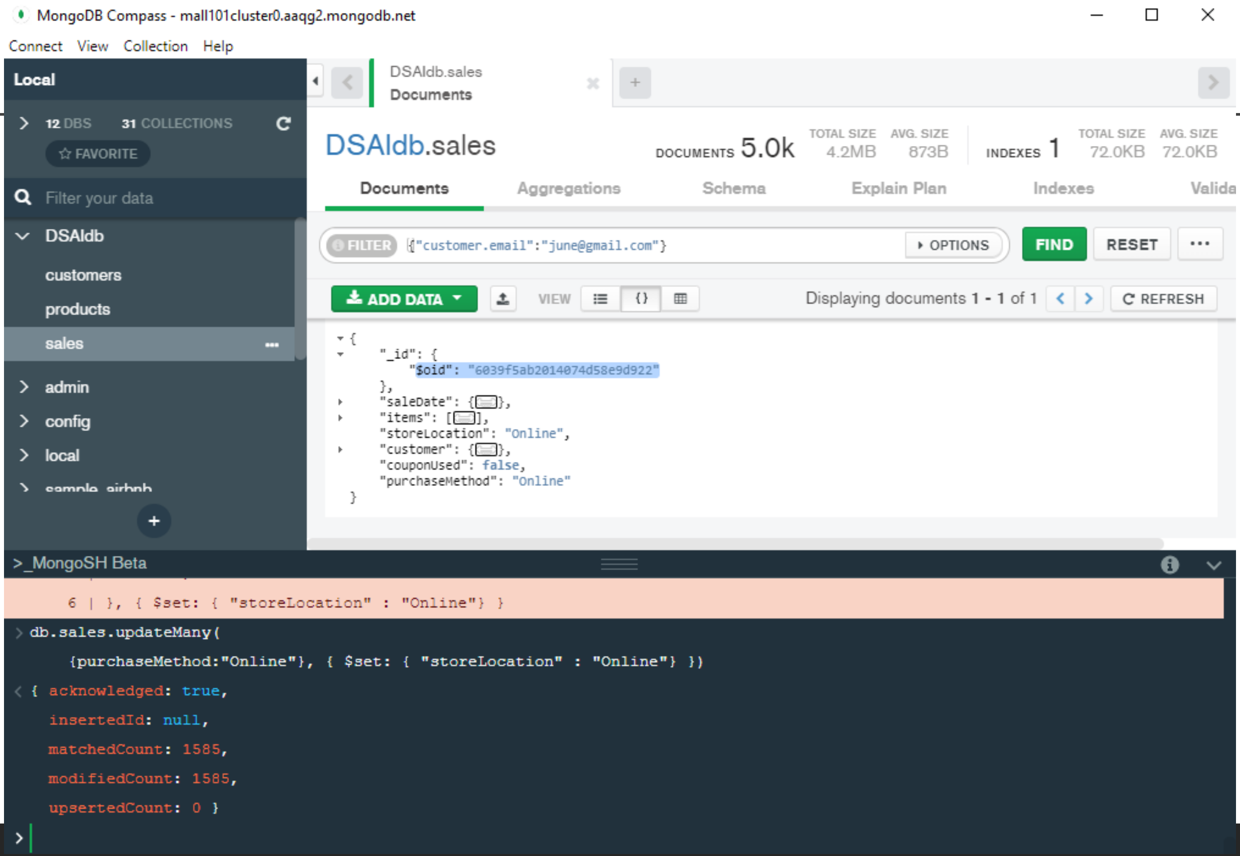
Mongo Shell



Command Prompt - mongo "mongodb+srv://dmmcluster.f1uoi.gcp.mongodb.net/shop101" --username st_dmm --password st_dmm

```
MongoDB Enterprise atlas-omm7a6-shard-0:PRIMARY> show databases
RDBProject  0.000GB
admin       0.000GB
local       1.224GB
shop101     0.000GB
MongoDB Enterprise atlas-omm7a6-shard-0:PRIMARY> show collections
customer
MongoDB Enterprise atlas-omm7a6-shard-0:PRIMARY> _
```

MongoDB Compass



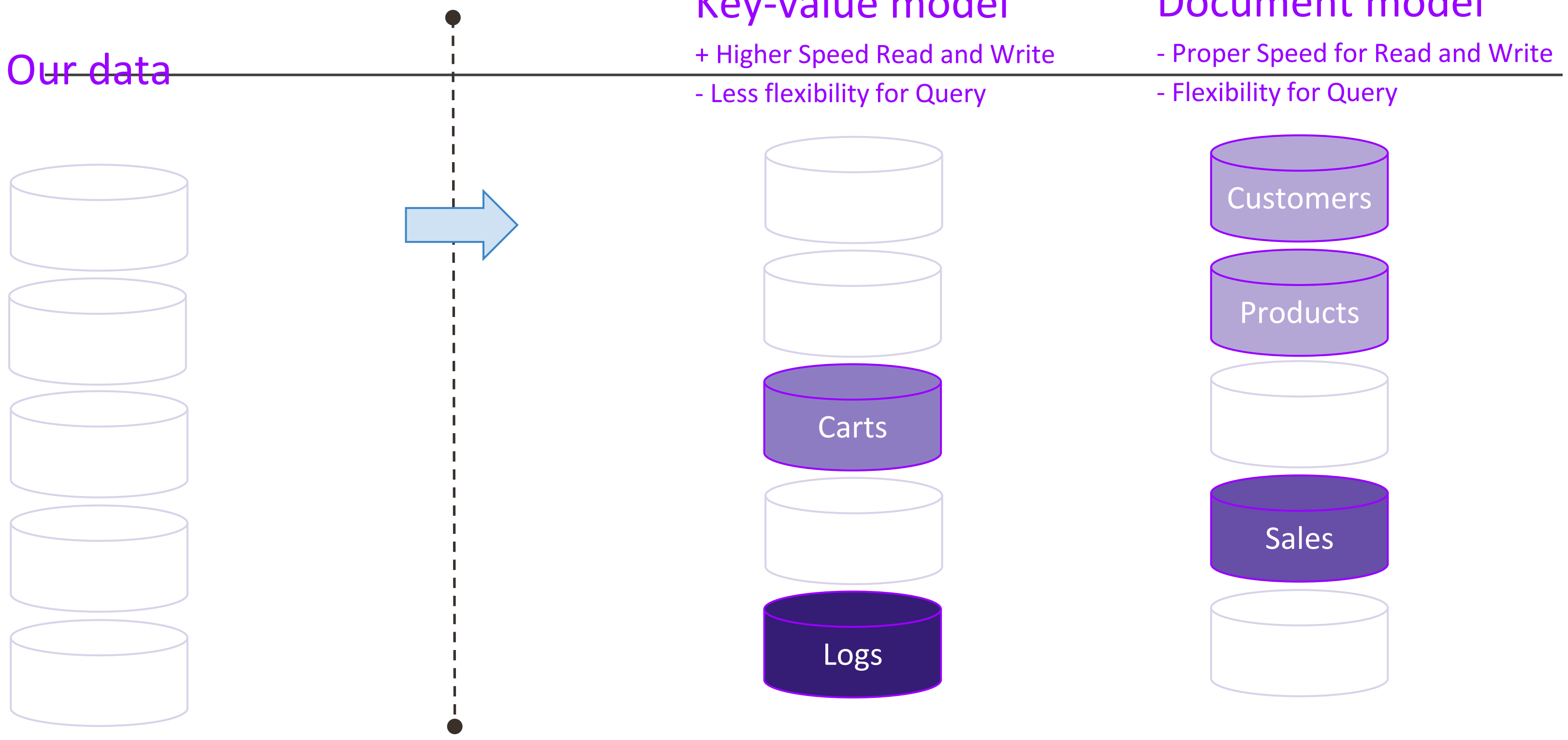
```
{"customer.email":"june@gmail.com"}
```

```
{
    "_id": {
        "$oid": "6039f5ab2014074d58e9d922"
    },
    "saleDate": {},
    "items": [],
    "storeLocation": "Online",
    "customer": {},
    "couponUsed": false,
    "purchaseMethod": "Online"
}
```

```
    6 | }, { $set: { "storeLocation" : "Online"} }
> db.sales.updateMany(
    {purchaseMethod:"Online"}, { $set: { "storeLocation" : "Online"} })
< { acknowledged: true,
    insertedId: null,
    matchedCount: 1585,
    modifiedCount: 1585,
    upsertedCount: 0 }
>
```

Modeling, managing and querying JSON documents

Data Modeling (Recall)



Key-value model

+ Higher Speed Read and Write

- Less flexibility for Query

Document model

- Proper Speed for Read and Write

- Flexibility for Query

TickTock data in RDB vs. MongoDB

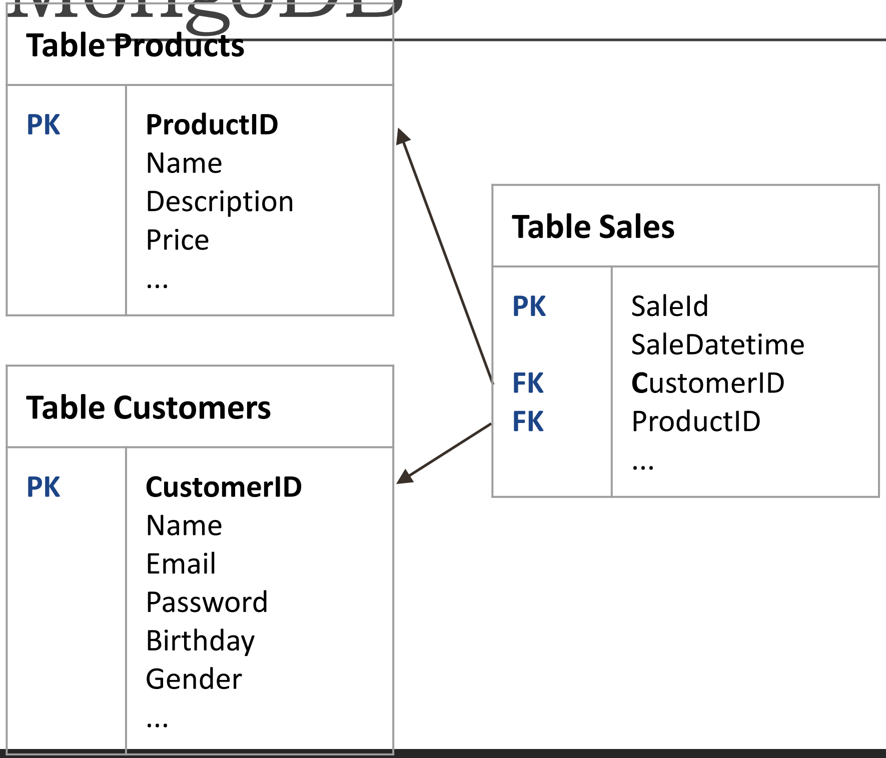


Table Products	
PK	**ProductID**
	Name
	Description
	Price
	...

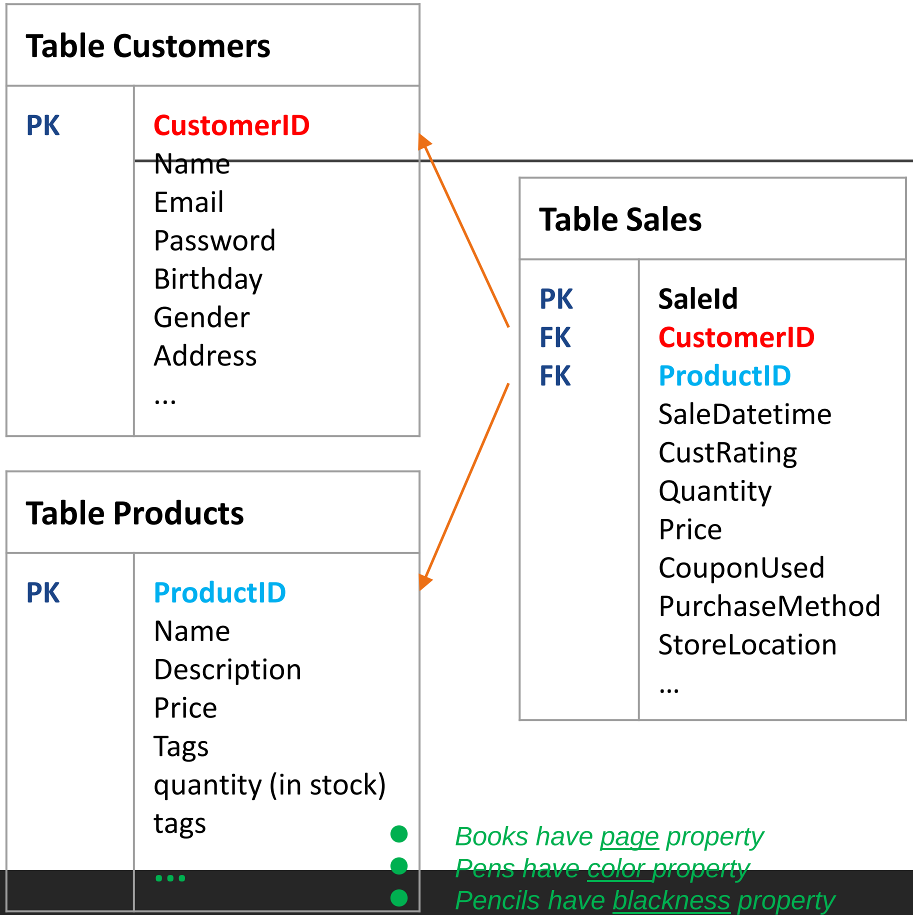
Table Customers	
PK	**CustomerID**
	Name
	Email
	Password
	Birthday
	Gender
	...

Table Sales	
PK	SaleId
	SaleDatetime
FK	CustomerID
FK	ProductID
	...

- What collections should we have?
- How can we model documents in the collections?
- Use reference or embedded model?

Data RDB

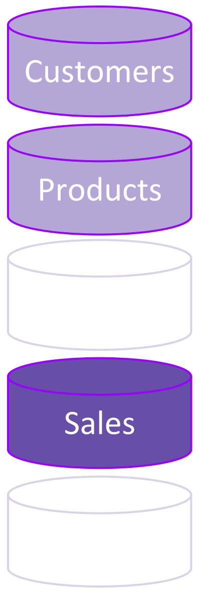
MongoDB



- *Books have <u>page</u> property*
- *Pens have <u>color</u> property*
- *Pencils have <u>blackness</u> property*

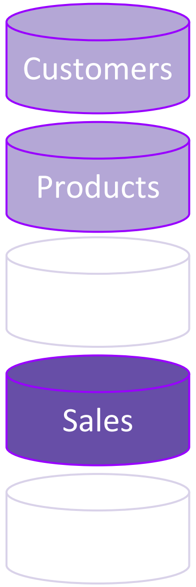
Modeling Solution

Document Data Modeling



Data	Field	Sample data	Datatype
Sales	SaleId	“5bd761dcae323e45a93c”	String
	SaleDatetime	2017-12-03 8:39:48	DateTime
	CustomerInfo		Object{}
	- CustID/Email	“june@gmail.com”	
	- Gender	“F”	
	- Age	40	
	- CustRating	4	
	ProductInfo		
	- Name	“backpack”	Object{}
	- Price	127	
	- Quantity	3	
	- Tags	“school”, “travel”, “kids”	
	CouponUsed	false	Boolean
	PurchaseMethod	“Online”	String
	StoreLocation	“Online Store”	String
	TotalPrice	700	Decimal

Document Data Modeling



Data	Field	Sample data	Datatype
Customer	custid	“june@gmail.com”	String
	cname	Junisa Jang	String
	address	111 Abc village Bond street, Bangkok,Thailand	Object{}
	birthdate	20/1/1988	DateTime
	password	“xxx”	String
	gender	“F”	String
Product	pid	“5bd761dcae323e45a93aaaa”	String
	pname	“Backpack”	String
	description	“A drawstring style crafted from sturdy black canvas”	String
	price	127.59	Decimal
	quantity (in stock)	200	Integer
	tags	“school”, “travel”, “kids”	Array

A sale

- a sale models as a document
- its properties model as a set of key-value pair.
- Value can be String, Number, DateTime, Boolean, Array, Object.

```
{
 "_id": {
     "$oid": "6039f5ab2014074d58e9d922"
 },
 "saleDate": {
     "$date": "2017-12-03T18:39:48.253Z"
 },
 "items": [{
     "name": "backpack",
     "tags": ["school", "travel", "kids"],
     "price": {
         "$numberDecimal": "127.59"
     },
     "quantity": 3
 }],
 "storeLocation": "Online",
 "customer": {
     "gender": "F",
     "age": 40,
     "email": "june@gmail.com",
     "customerRating": 4
 },
 "couponUsed": false,
 "purchaseMethod": "Online"
}
```

A product

- a product models as a document
- its properties model as a set of key-value pair.
- Value can be String, Number, DateTime, Boolean, Array, Object.

Example of common properties

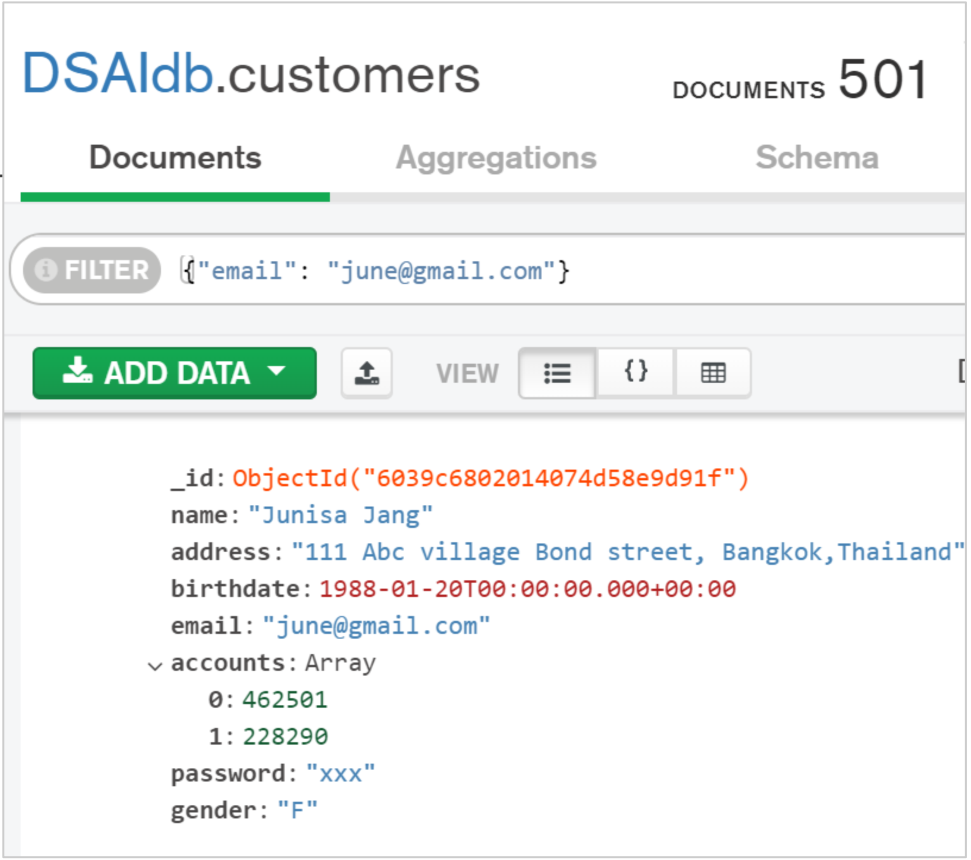
Specific properties
E.g.,
- *Books have page property*
- *Pens have color property*
- *Pencils have blackness property*

```
{
        "_id": {
         "$oid": "6039ccf72014074d58e9d921"
   },
        "name": "backpack",
        "description": "A drawstring style
         crafted from sturdy black canvas",
        "tags": ["school", "travel", "kids"],
        "price": {
                    "$numberDecimal": "127.59"
        },
        "quantity": 200
          ...
}
```

A customer

- a customer models as a document
- customer properties model as a set of key-value pair.
- Value can be String, Number, DateTime, Boolean, Array, Object.

```
 {
    "_id": {
        "$oid": "6039c6802014074d58e9d91f"
    },
    "name": "Junisa Jang",
    "address": "111 Abc village Bond
street, Bangkok,Thailand",
    "birthdate": {
        "$date": "1988-01-
20T00:00:00.000Z"
    },
    "email": "june@gmail.com",
    "accounts": [462501, 228290],
    "password": "xxx",
    "gender": "F"
}
```



```
{"email": "june@gmail.com"}
```

```
_id: ObjectId("6039c6802014074d58e9d91f")
name: "Junisa Jang"
address: "111 Abc village Bond street, Bangkok,Thailand"
birthdate: 1988-01-20T00:00:00.000+00:00
email: "june@gmail.com"
accounts: Array
    0: 462501
    1: 228290
password: "xxx"
gender: "F"
```

Mongodb data model

$$\begin{tikzcd} \text{Customer} & \text{Sale document} \arrow[l, "\text{email}"'] \arrow[r, "\text{name}"] & \text{Product} \end{tikzcd}$$

```
 {"_id":
    {"$oid": "60d58e9d91f"},
    "name": "Junisa Jang",
    "address": "111 Abc village
     Bond street, Bangkok,
    "birthdate": {
        "$date": "1988-01-20"
    },
    "email": "june@gmail.com",
    "accounts": [462501, 228290],
    "password": "xxx",
    "gender": "F"
}
```

```
{
 "_id": { "$oid": "6039f5ab20"},
 "saleDate": {
     "$date": "2017-12-03T18:39:48.253Z" },
 "items": [{
     "name": "backpack",
     "tags": ["school", "travel", "kids"],
     "price": {
         "$numberDecimal": "130" },
     "quantity": 3 }],
 "storeLocation": "Online",
 "customer": {
     "gender": "F",
     "age": 33,
     "email": "june@gmail.com",
     "customerRating": 4 },
 "couponUsed": false,
 "purchaseMethod": "Online",
 “totalPrice”:390
}
```

```
{
  "_id": {
      "$oid": "6039ccf72014...01"
  },
  "name": "backpack",
  "description": "A drawstring style
   crafted from sturdy black canvas",
  "tags": ["school","travel","kids"],
  "price": {
                 "$numberDecimal":
"127.59"
  },
  "quantity": 200
}
```

CRUD operations on MongoDB

CRUD Operations of MongoDB

- Read data
    - Simple query
    - Nested query
    - Aggregate function
    - Join/Lookup document
- Create/Insert, Update, Delete documents

$\left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\\a\end{array}}\right\}$ Through use cases

Use Cases & CRUD Operations

Customer

Use Case: U2

Customer searches products

Core Concepts

- Basic Query
- Query on Complex fields E.g., nested fields or array fields

Staff

Use case: U6

Staff manages Product information

6.1(1) add product information
6.1(2) update bestseller flag for product
6.1(3) delete rejected sale document from sales

Core Concepts

- INSERT Operations
- UPDATE Operations
- DELETE Operations
- Aggregate Operations

Use Case: U7

Staff views daily report

Store Owner

Use Case: U8

Owner views summary report

8.1(1) Get the top 3 popular product
8.1(2) Get total sales group by month

Core Concepts

Aggregate Operations

the REFERENCE

- P. Sadalage and M. Fowler: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, 2013
- Jan L. Harrington: Relational Database Design and Implementation, 4th edition, Morgan Kaufmann, 2016
- A. Makris, K. Tserpesa, V. Andronikou Dimosthenis Anagnostopoulos: A Classification of NoSQL Data Stores Based on Key Design Characteristics, Procedia Computer Science, Vol. 97, 2016, pp. 94-103.
- MongoDB Schema Design: Practical Applications and Implications [https://www.slideshare.net/mongodb/mongodb-schema-design-practical-applications-and-implications]