

Introducing MongoDB: An Open-Source NoSQL Database

IBM

April 27, 2025



Skills
Network



Overview of MongoDB

Overview of MongoDB

© IBM Corporation. All rights reserved.

What you will learn



Explain what
MongoDB is



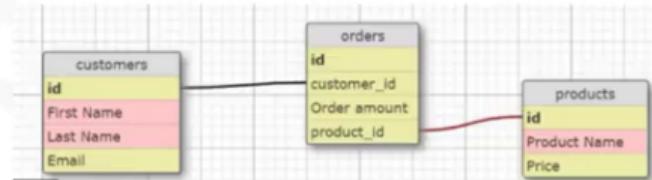
List different
components of
MongoDB



Describe why and
where to use
MongoDB

What is MongoDB database?

- A document and a NoSQL database



- Where data is structured in non-relational way



Order
document

What are documents?

- Associative arrays like JSON objects or Python dictionaries
- For example: A student document

```
{  
  "firstName": "John",  
  "lastName": "Doe",  
  "email": "john.doe@email.com",  
  "studentId": 20217484  
}
```

What is a collection?

- Is a group of stored documents
- For example, all student records in Students section (**collection**) and
- Staff records in Employees section (**collection**)



Students



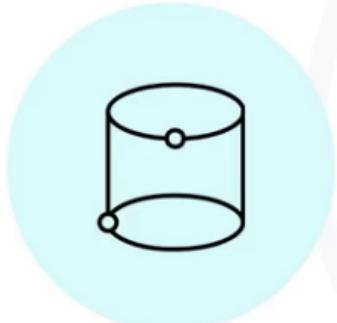
Employees



Employees

What is a database?

- A database stores collections



Students and Employees collections
stored in a database called
CampusManagementDB

Documents in detail - 1/2

See the following fields in the document:
firstName, lastName, email and studentId

```
{  
  "firstName": "John",  
  "lastName": "Doe",  
  "email": "john.doe@email.com",  
  "studentId": 20217484  
}
```

Documents in detail - 1/2

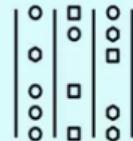
See the following fields in the document:
firstName, lastName, email and studentId

```
{  
  "firstName": "John",  
  "lastName": "Doe",  
  "email": "john.doe@email.com",  
  "studentId": 20217484  
}
```

Documents in detail - 2/2

MongoDB supports various data types:

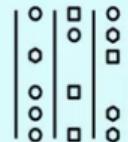
```
{  
  "name": "John",  
  "dateOfBirth": ISODate("2000-01-01T14:45:00.000Z"),  
  "studentId": 20217484,  
  "enrolled": true,  
  "balance": 20.01,  
  "address": {  
    "city": "Stonefield",  
    "country": "UK"  
  },  
  "interests": ["football", "skiing", "travelling"]  
}
```



Documents in detail - 2/2

MongoDB supports various data types:

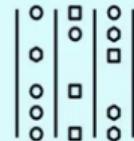
```
{  
  "name": "John",  
  "dateOfBirth": ISODate("2000-01-01T14:45:00.000Z"),  
  "studentId": 20217484,  
  "enrolled": true,  
  "balance": 20.01,  
  "address": {  
    "city": "Stonefield",  
    "country": "UK"  
  },  
  "interests": ["football", "skiing", "travelling"]  
}
```



Documents in detail - 2/2

MongoDB supports various data types:

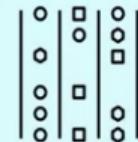
```
{  
  "name": "John",  
  "dateOfBirth": ISODate("2000-01-01T14:45:00.000Z"),  
  "studentId": 20217484,  
  "enrolled": true,  
  "balance": 20.01,  
  "address": {  
    "city": "Stonefield",  
    "country": "UK"  
  },  
  "interests": ["football", "skiing", "travelling"]  
}
```



Documents in detail - 2/2

MongoDB supports various data types:

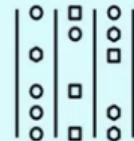
```
{  
    "name": "John",  
    "dateOfBirth": ISODate("2000-01-01T14:45:00.000Z"),  
    "studentId": 20217484,  
    "enrolled": true,  
    "balance": 20.01,  
    "address": {  
        "city": "Stonefield",  
        "country": "UK"  
    },  
    "interests": ["football", "skiing", "travelling"]  
}
```



Documents in detail - 2/2

MongoDB supports various data types:

```
{  
    "name": "John",  
    "dateOfBirth": ISODate("2000-01-01T14:45:00.000Z"),  
    "studentId": 20217484,  
    "enrolled": true,  
    "balance": 20.01,  
    "address": {  
        "city": "Stonefield",  
        "country": "UK"  
    },  
    "interests": ["football", "skiing", "travelling"]  
}
```



Why use MongoDB?

- Model data as you read/write, not the other way
 - Traditional relational databases: Create the schema first, then create the tables
 - To store another field, you have to alter tables

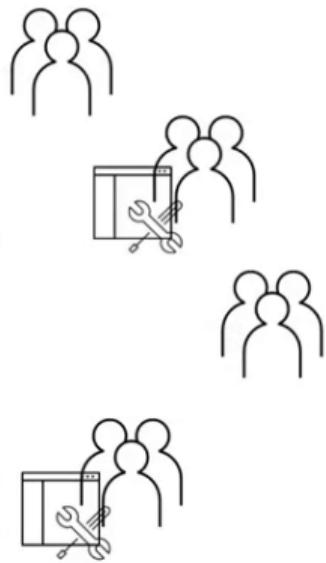
```
SELECT * FROM Orders INNER JOIN Customers...
```

```
ALTER TABLE Customers...
```



Why use MongoDB?

- Model data as you read/write, not the other way
- Bring structured/unstructured data



Why use MongoDB?

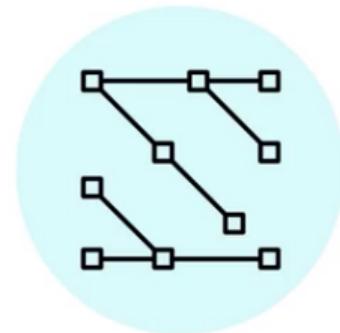
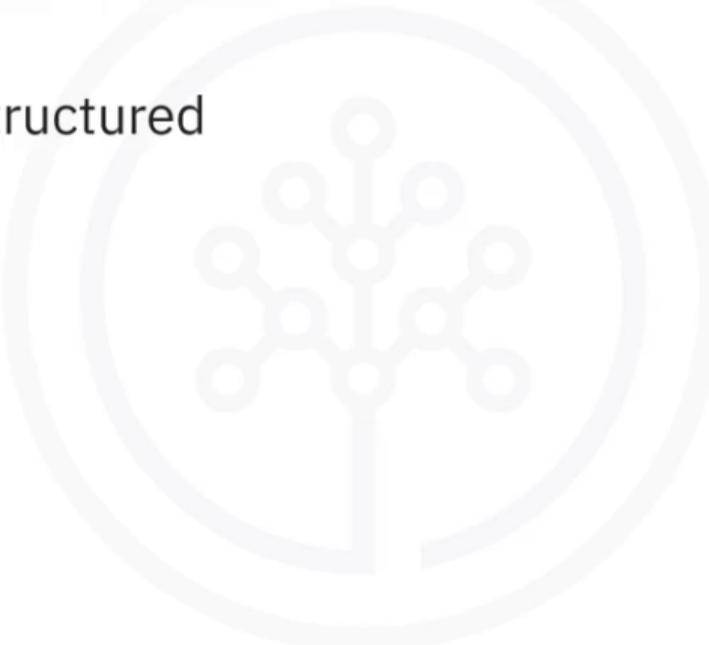
- Model data as you read/write, not the other way
- Bring structured/unstructured data
- High availability



Where to use MongoDB

MongoDB is a popular choice of database for

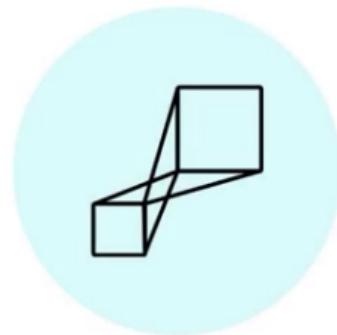
- Large and unstructured
- Complex
- Flexible



Where to use MongoDB

MongoDB is a popular choice of database for

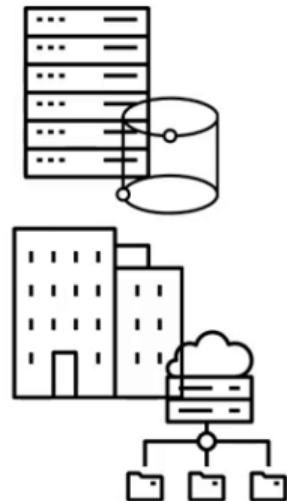
- Large and unstructured
- Complex
- Flexible
- Highly scalable applications



Where to use MongoDB

MongoDB is a popular choice of database for

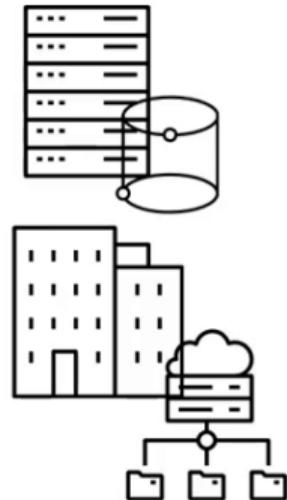
- Large and unstructured
- Complex
- Flexible
- Highly scalable applications
- Self-managed, hybrid, or cloud hosted



Where to use MongoDB

MongoDB is a popular choice of database for

- Large and unstructured
- Complex
- Flexible
- Highly scalable applications
- Self-managed, hybrid, or cloud hosted



<https://www.ibm.com/cloud/databases-for-mongodb>



Skills
Network

Advantages of MongoDB

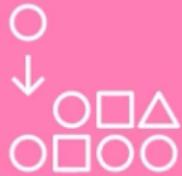
Advantages of MongoDB

© IBM Corporation. All rights reserved.

What you will learn



Identify the key benefits of using MongoDB



Explain why it suits your evolving data needs

Flexibility with schema

```
{  
  "street": "10 High St",  
  "city": "London",  
  "postcode": "W1 1SU"  
}
```

```
{  
  "street": "8717 West St",  
  "city": "New York",  
  "zip": "10940"  
}
```

Flexibility with schema

```
{  
  "street": "10 High St",  
  "city": "London",  
  "postcode": "W1 1SU"  
}
```

```
{  
  "street": "8717 West St",  
  "city": "New York",  
  "zip": "10940"  
}
```

Flexibility with schema

```
{  
  "street": "10 High St",  
  "city": "London",  
  "postcode": "W1 1SU"  
}
```

```
{  
  "street": "8717 West St",  
  "city": "New York",  
  "zip": "10940"  
}
```

MongoDB allows flexibility with the schema

Code-first approach

In relational databases

- Design
- Then code

```
CREATE TABLE Students (
    FirstName varchar(255),
    LastName varchar(255),
    Email varchar(255),
    StudentId int);
```

```
INSERT INTO Students
VALUES (
    "John",
    "Doe",
    "john.doe@email.com",
    20217484);
```

Code-first approach

In relational databases

- Design
- Then code

In MongoDB, code first!

- No complex table definitions
- Write as soon as you connect to DB

```
db.persons.insertOne({  
  "firstName": "John",  
  "lastName": "Doe",  
  "email": "john.doe@email.com",  
  "studentId": 20217484  
})
```

Evolving schema

The whole world changed in 2020!

```
//pre-covid schema
{
  "street": "10 High St",
  "city": "London",
  "postcode": "W1 1SU"
}
```

```
//evolved schema
{
  "street": "10 High St",
  "city": "London",
  "postcode": "W1 1SU",
  "contactlessDelivery": true
}
```

Evolving schema

The whole world changed in 2020!

```
//pre-covid schema
{
  "street": "10 High St",
  "city": "London",
  "postcode": "W1 1SU"
}
```

```
//evolved schema
{
  "street": "10 High St",
  "city": "London",
  "postcode": "W1 1SU",
  "contactlessDelivery": true
}
```

Unstructured data

Stock Market Aggregator

```
{  
  "symbol": "IBM",  
  "open": 235.9,  
  "high": 237.47,  
  "low": 233.17  
}
```

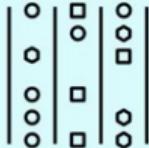
```
{  
  "stockName": "IBM",  
  "pricing": {  
    "o": 235.9,  
    "h": 237.47,  
    "l": 233.17  
  },  
  date: "2021-03-01T00:00:00+0000"  
}
```

Querying and analytics

MongoDB querying using MQL



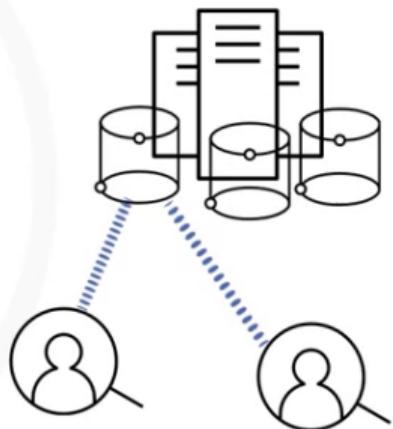
Has a wide range of operators



For complex analysis use aggregation pipelines

High availability

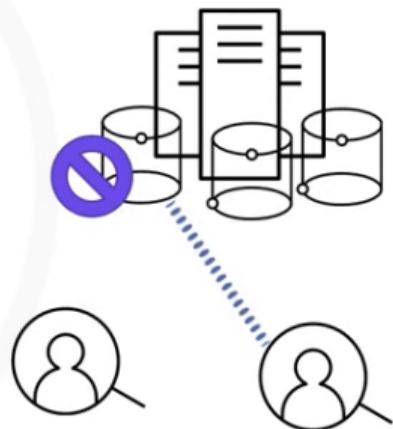
MongoDB is natively a highly available system:



High availability

MongoDB is natively a highly available system:

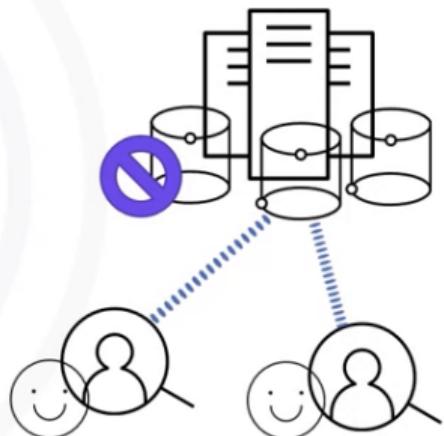
- Resilience through redundancy



High availability

MongoDB is natively a highly available system:

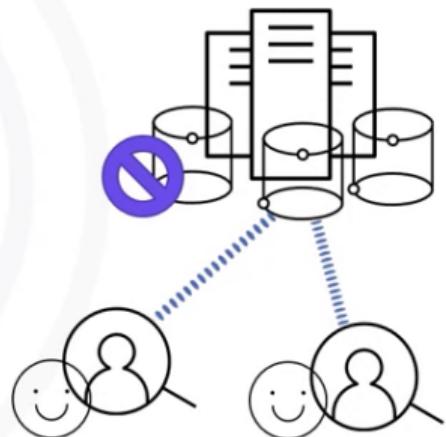
- Resilience through redundancy



High availability

MongoDB is natively a highly available system:

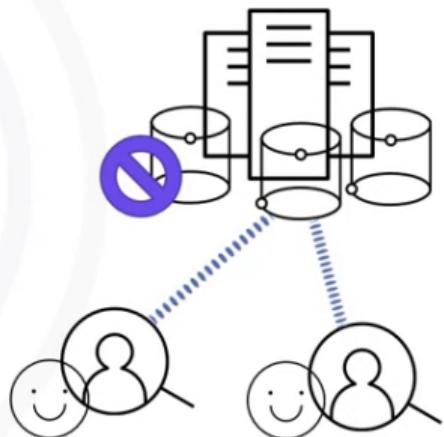
- Resilience through redundancy
- No system maintenance downtime



High availability

MongoDB is natively a highly available system:

- Resilience through redundancy
- No system maintenance downtime
- No upgrade downtime





Skills
Network

Use Cases for MongoDB

Use Cases for MongoDB

© IBM Corporation. All rights reserved.

What you will learn



List the most common use cases for MongoDB



Describe the Many Sources – One View use case



Describe the IoT use case

What you will learn



Describe the E-commerce use case



Describe the Real-time Analytics use case



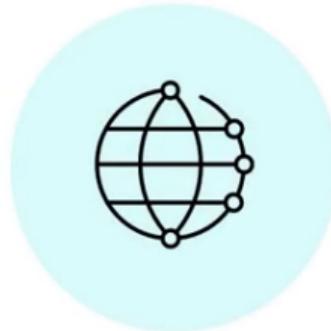
Describe the Gaming use case



Describe the Finance use case

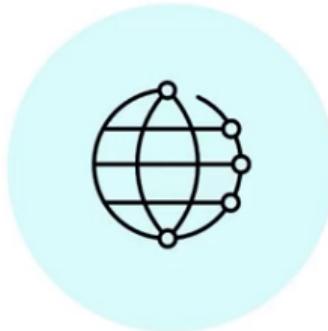
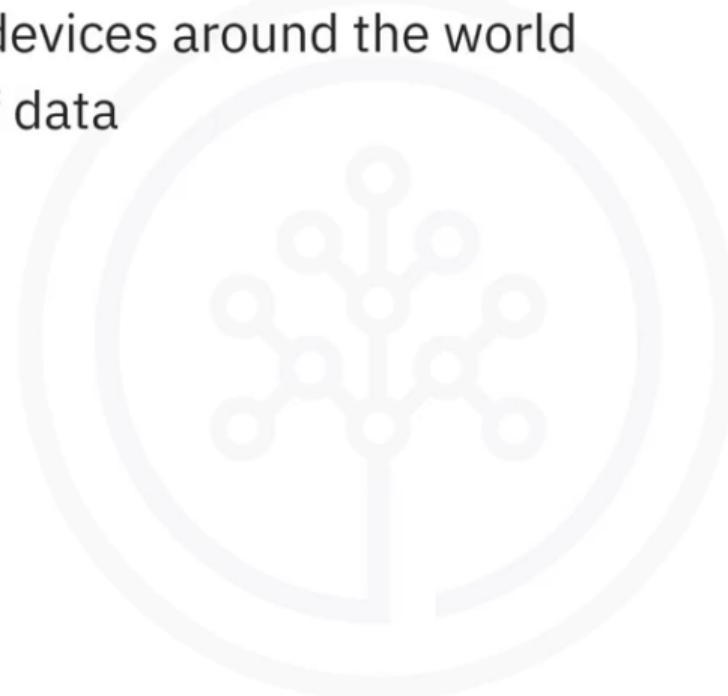
Internet of Things (IoT)

- Billions of IoT devices around the world



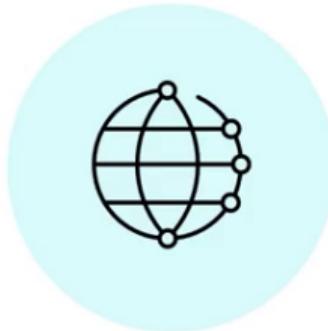
Internet of Things (IoT)

- Billions of IoT devices around the world
- Vast amount of data



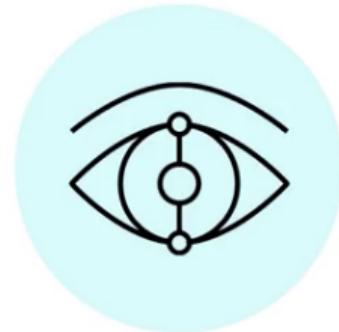
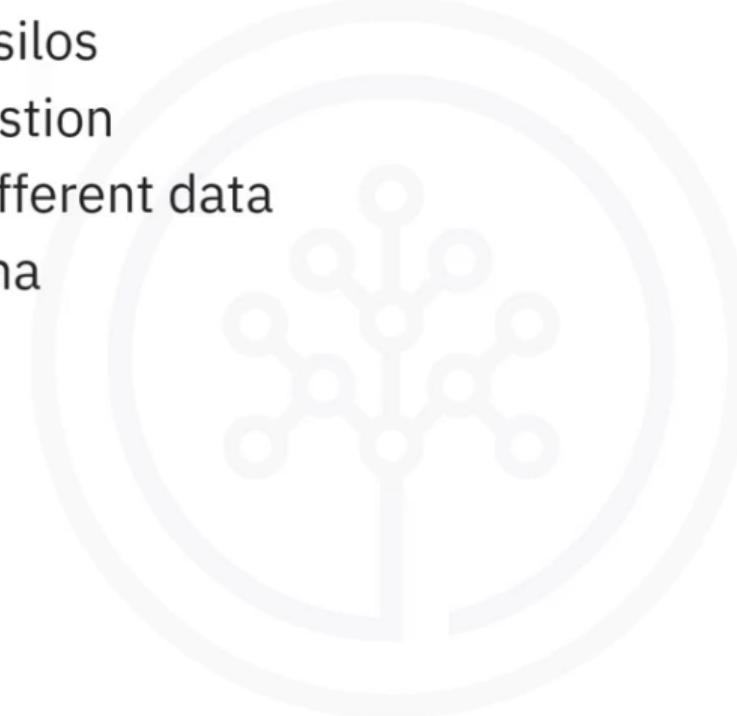
Internet of Things (IoT)

- Billions of IoT devices around the world
- Vast amount of data
- Scale



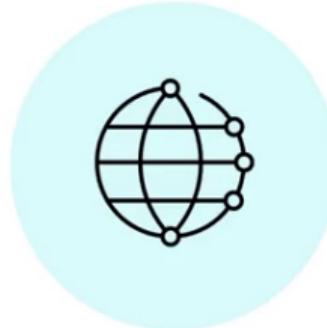
Many Sources – One View

- No more data silos
- Easy data ingestion
- Consolidate different data
- Flexible schema



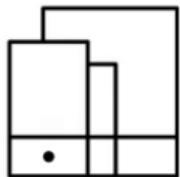
Internet of Things (IoT)

- Billions of IoT devices around the world
- Vast amount of data
- Scale
- Expressive querying

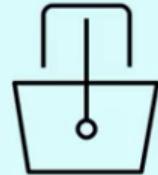


E-commerce

- Products with different attributes

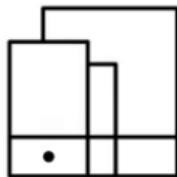


```
{ "storage": "64GB", "network": "5G", "color": "black" }
```



E-commerce

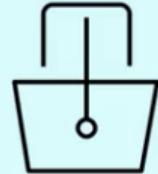
- Products with different attributes



```
{ "storage": "64GB", "network": "5G", "color": "black" }
```

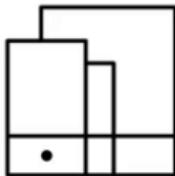


```
{ "publisher": "Oxford Press", "writer": "John Doe",  
"pages": 250 }
```



E-commerce

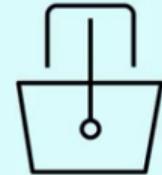
- Products with different attributes
- Optimized for read
- Dynamic schema



```
{ "storage": "64GB", "network": "5G", "color": "black" }
```

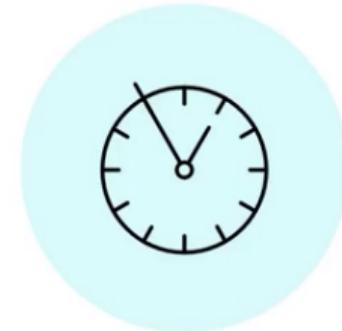
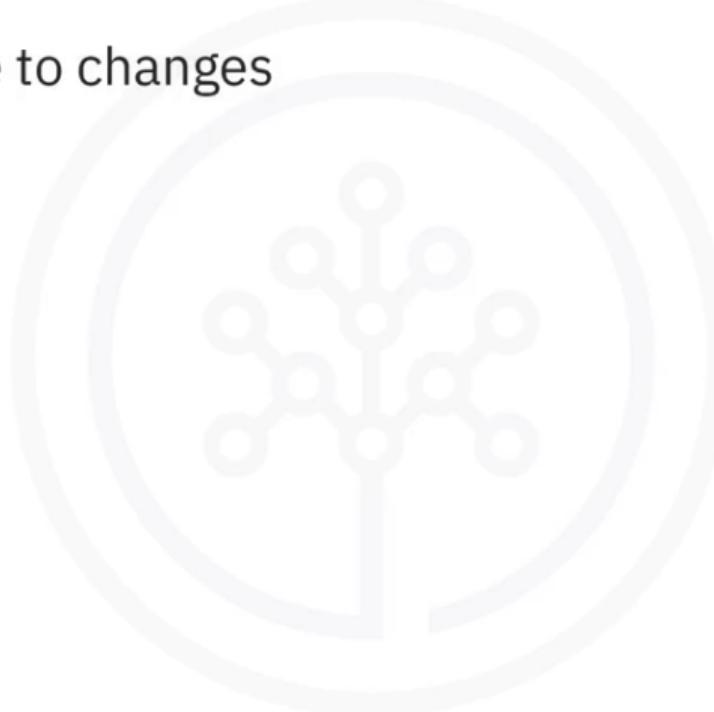


```
{ "publisher": "Oxford Press", "writer": "John Doe",  
"pages": 250 }
```



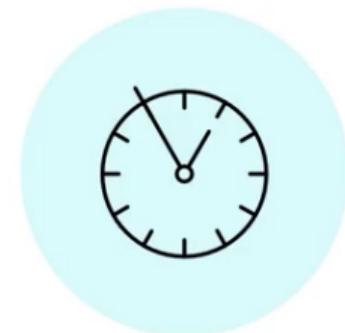
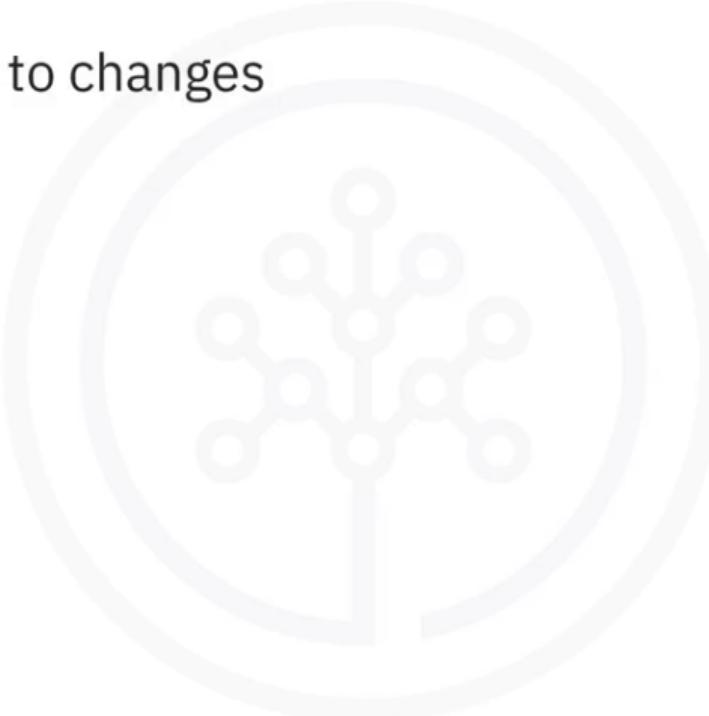
Real-time analytics

- Quick response to changes



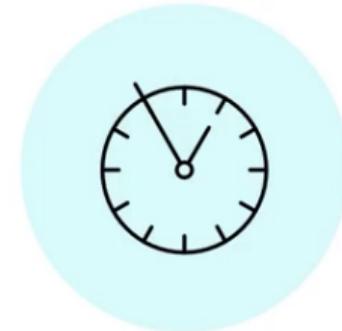
Real-time analytics

- Quick response to changes
- Simplified ETL



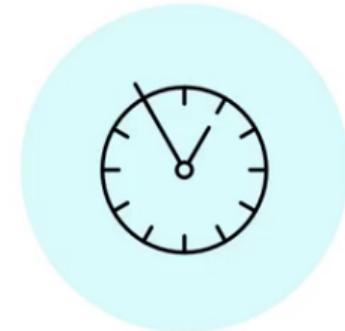
Real-time analytics

- Quick response to changes
- Simplified ETL



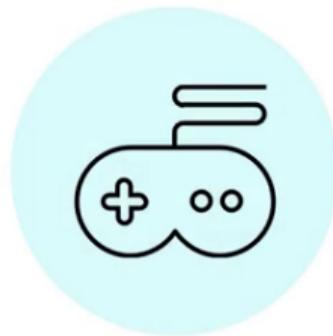
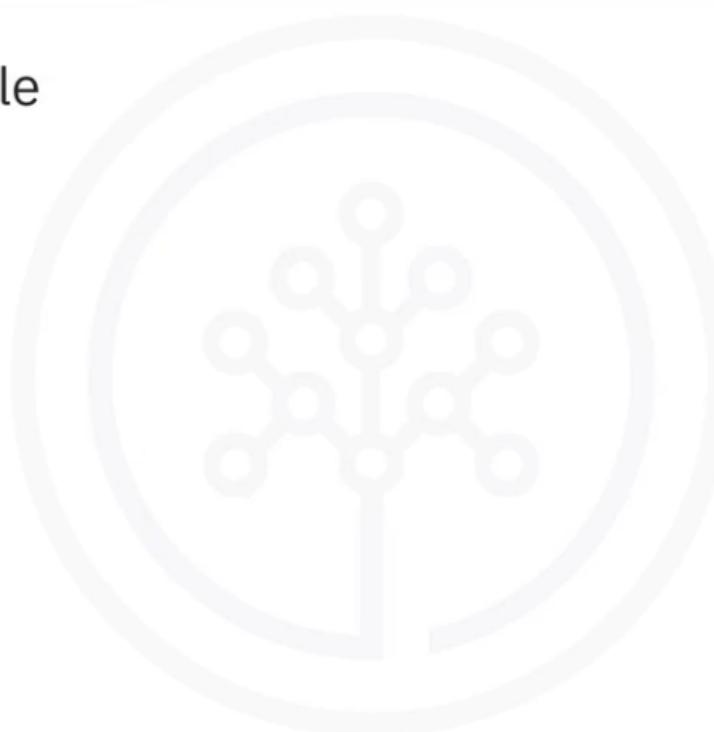
Real-time analytics

- Quick response to changes
- Simplified ETL
- Real time, along with operational data



Gaming

- Globally scalable



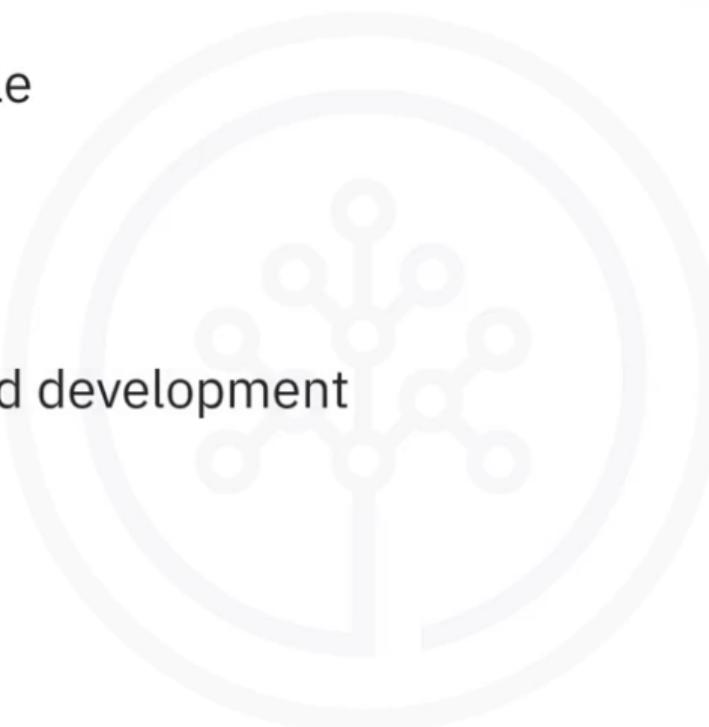
Gaming

- Globally scalable
- No downtime



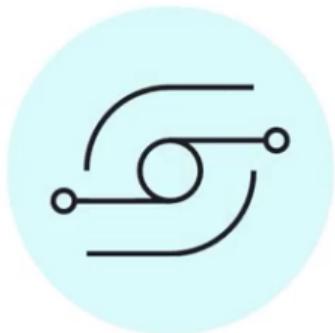
Gaming

- Globally scalable
- No downtime
- Supporting rapid development



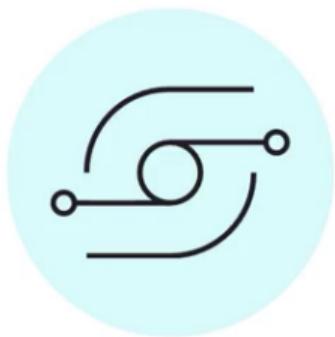
Finance

Speed



Finance

Speed



Security



Finance

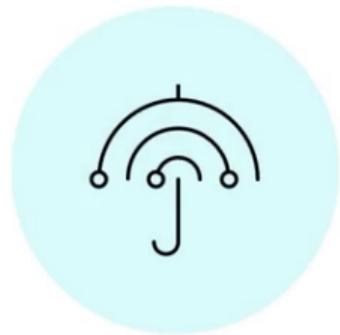
Speed



Security



Reliability





Skills
Network

IBM