
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

PRT314

SPONSORED BY MONGODB

From RDBMS to NoSQL

Rick Houlihan

Director
Developer Relations, Strategic Accounts
MongoDB

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

The Great Migration - Why did Amazon choose NoSQL?

Blazing a New Trail - Finding the path to database freedom.

Making it Work - Modeling relationships in NoSQL.

Executing the Plan - Managing a technology transformation at enterprise scale.

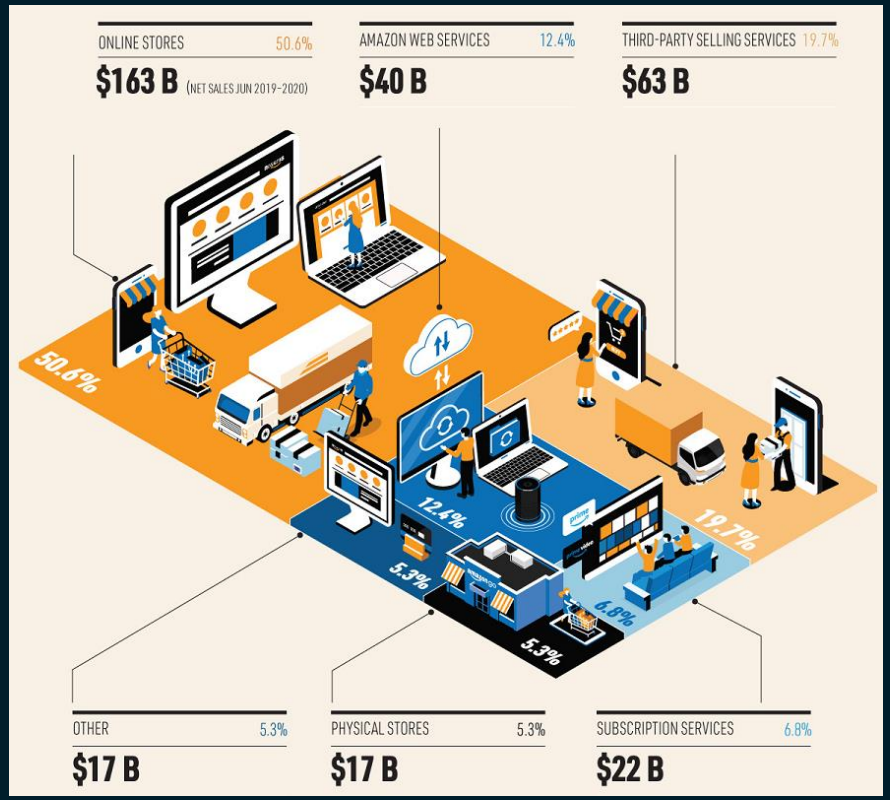
Thinking in Documents - Redefining the process for building applications.

The Great Migration - Why NoSQL?

“Change begins at the end of your comfort zone.”

- Roy T. Bennett

Not really “Born in the Cloud”…



Amazon was founded in 1995 and built on monolithic services.

$5B company when SOA became an internal standard in 2005.

By 2018 scale was breaking many things

- 3K Oracle instances
- 10K Application Services
- 25K Developers

Cost of RDBMS was a big line item

- Largest Oracle license in the world
- High infrastructure TCO

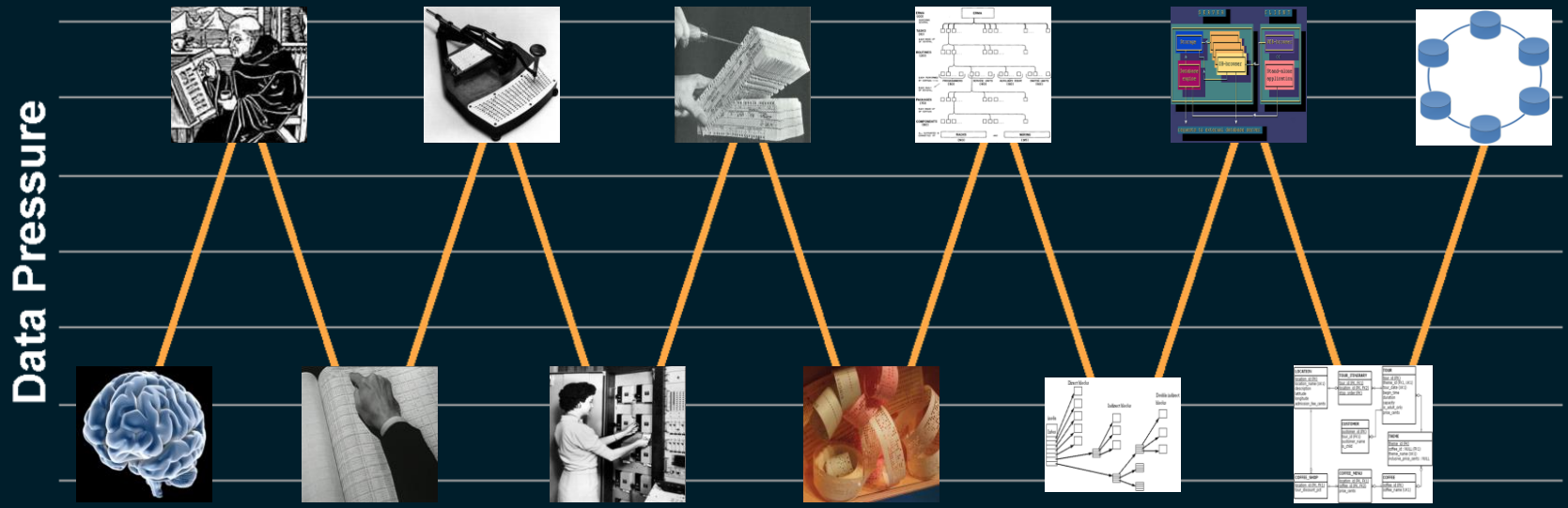
Maybe a better way?

Blazing a New Trail

“History repeats itself because nobody listens the first time.”

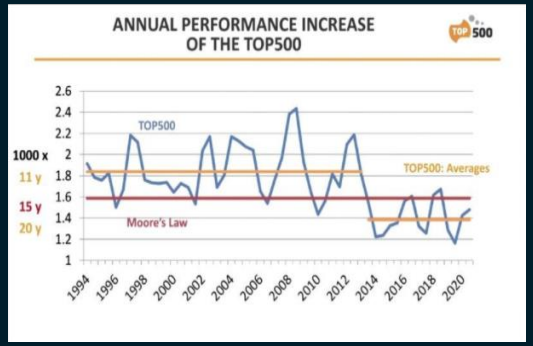
- Erik Qualman

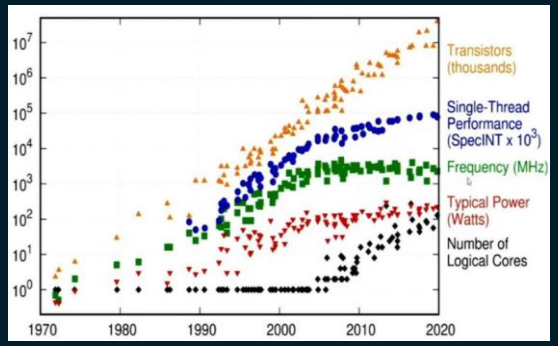
Timeline of Database Technology



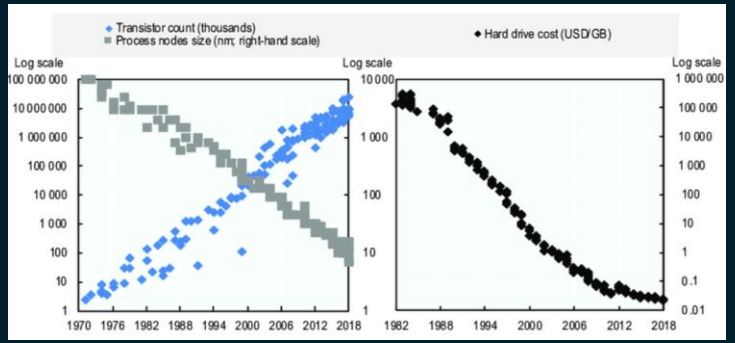
Moore's Law no longer an enabler

CPU performance is flattening





Cost of storage continues to fall



It’s not new thinking…

On Normalization:

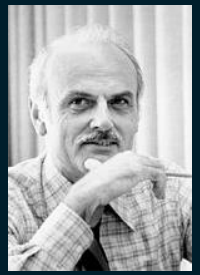
“The simplicity of the array representation which becomes feasible when all relations are cast in normal form is not only an **advantage for storage** purposes but also for **communication of bulk data** between systems which use widely different representations of the data.”

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

When NoSQL?



SQL	NoSQL
Optimized for storage	**Optimized for compute**
Normalized/relational	De-normalized/hierarchical
Ad hoc queries	Instantiated views
Scale vertically	Scale horizontally
Good for OLAP	Built for OLTP at scale



Making it Work

“Data is like garbage. You’d better know what you are going to do with it before you collect it.”

- Mark Twain

It’s all about relationships



Social network



Document management



Process control



IT monitoring



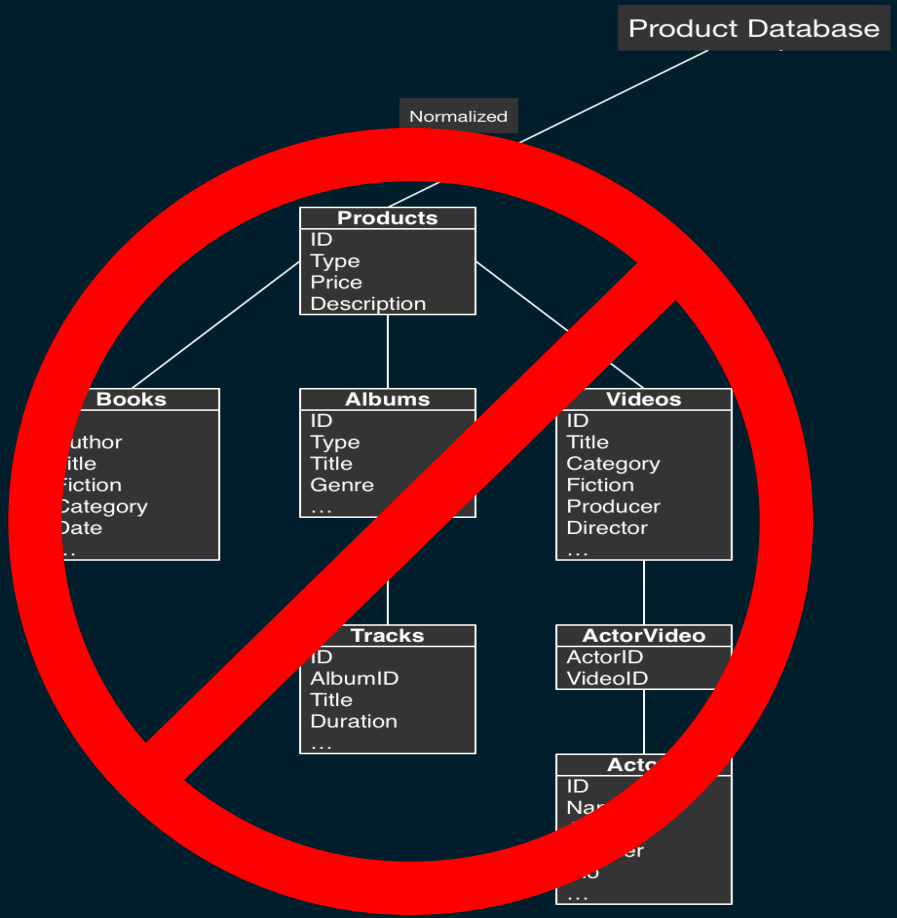
Data lake

But access patterns drive cost



- At Amazon in 2017, 70% of queries accessed a single row of data
- 20% more accessed a range of rows from a single table
- 50% of the infrastructure supported the last 10%

SQL vs. NoSQL design pattern



Ad hoc “joins” in SQL



```
SELECT * FROM PRODUCTS
INNER JOIN BOOKS ON
productId = productId
WHERE name = “Book Title”

SELECT * FROM PRODUCTS
INNER JOIN ALBUMS ON
productId = productId
INNER JOIN TRACKS ON
albumId = albumId
WHERE name = “Album Title”

SELECT * FROM PRODUCTS
INNER JOIN VIDEOS 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 Perki	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))

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))

Adding more dimensions...



```
{
    "_id": "Penny Marshall",
    "data": {
        "birthdate": "1943-10-15",
        "deceased": "12-17-2018"
    },
    "target": [
        {"id": "Penny Marshall"},
        {"id": "Big", "type": "movie"}
    ]
},
```

```
{
    "_id": "Tom Hanks",
    "data": {
        "birthdate": "1956-07-09"
    },
    "target": [
        {"id": "Tom Hanks"},
        {"id": "Big", "type": "movie"},
        {"id": "Rita Wilson", "type": "spouse"}
    ]
},
```

```
{
    "_id": "Mary Shelley",
    "data": {
        "birthdate": "1797-08-30",
        "deceased": "1851-02-01"
    },
    "target": [
        {"id": "Mary Shelley"},
        {"id": "Percy Bysshe Shelley", "type": "spouse"},
        {"id": "Frankenstein", "type": "author"}
    ]
},
```

```
{
    "_id": "Sultans of Swing",
    "data": {
        "released": {
            "UK": "1978-19-05",
            "US": "1979-01-01"
        },
        "length": "5:47"
    },
    "target": [
        {"id": "Sultans of Swing"},
        {"id": "Mark Knopfler", "type": "writer"},
        {"id": "Muff Winwood", "type": "producer"},
        {"id": "Basing Street", "type": "studio"},
        {"id": "Vertigo - Warner Bros.", "type": "label"},
        {"id": "Pub rock",  "type": "genre"},
        {"id": "Dire Straits",  "type": "album"},
        {"id": "Dire Straits", "type": "artist"}
    ]
},
```

Index “joins” in NoSQL



SELECT * WHERE target.id = “Mary Shelley”

SELECT * WHERE target.id = “Sultans of Swing”

SELECT * WHERE target.id = “Tom Hanks”

SELECT * WHERE target.id = “Penny Marshall”

SELECT * WHERE target.id = “musician”: “Dire Straits”

SELECT * WHERE target.id = “Big”

Time Complexity

O(log(N))

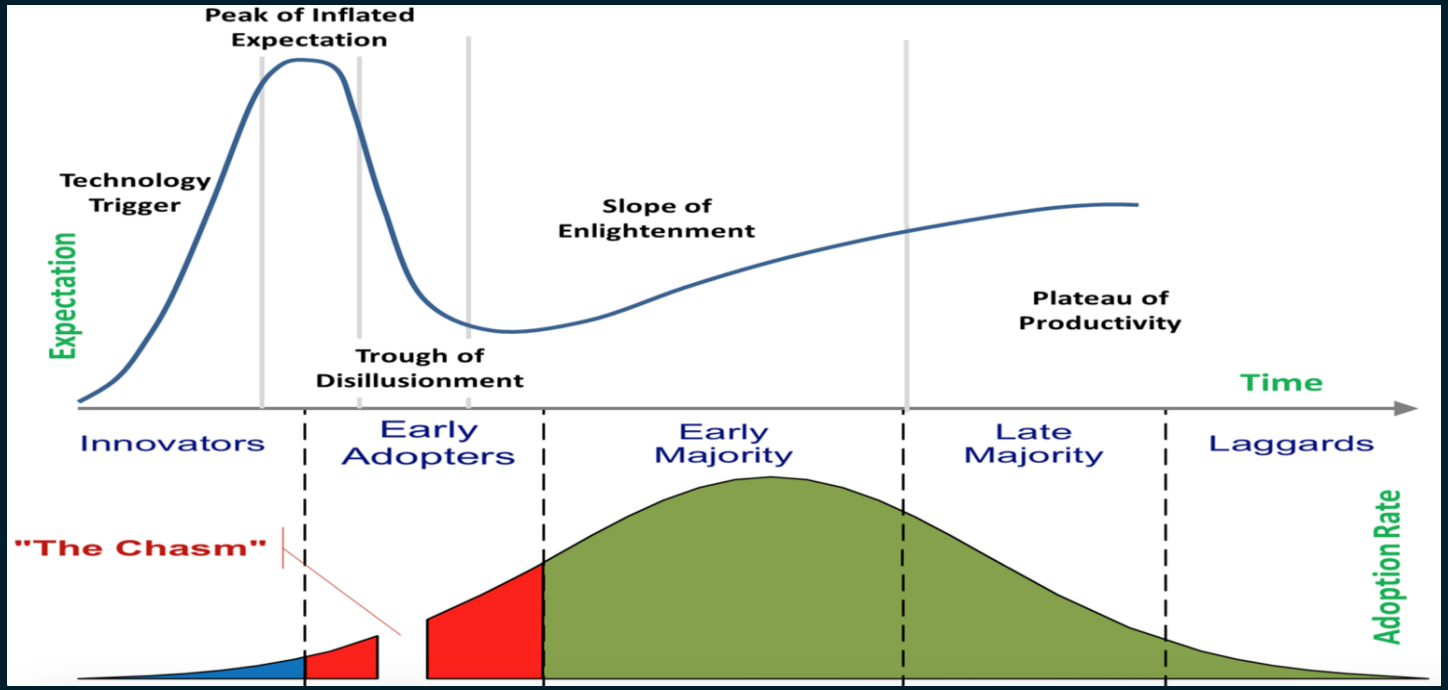
Target Index		
target.id	target.type	_id
"Mary Shelley"	"author"	"Frankenstein"
	""person	"Mary Shelley"
"Sultans of Swing"	"album"	"Dire Straits"
	"album"	"Best of Dire Straits"
	"artist"	"Dire Straits"
	"song"	"Sultans of Swing"
"Tom Hanks"	"actor"	"Big"
	"actor"	"Saving Private Ryan"
	"person"	"Tom Hanks"
"Penny Marshall"	"director"	"Big"
	"person"	"Penny Marshall"
"Dire Straits"	"album"	"Dire Straits"
	"album"	"Best of Dire Straits"
	"artist"	"Dire Straits"
	"artist"	"Sultans of Swing"
"Big"	"actor"	"Tom Hanks"
	"actor"	"Elizabeth Perkins"
	"actor"	"Robert Loggia"
	"director"	"Penny Marshall"
	"movie"	"Big"

Executing the Plan

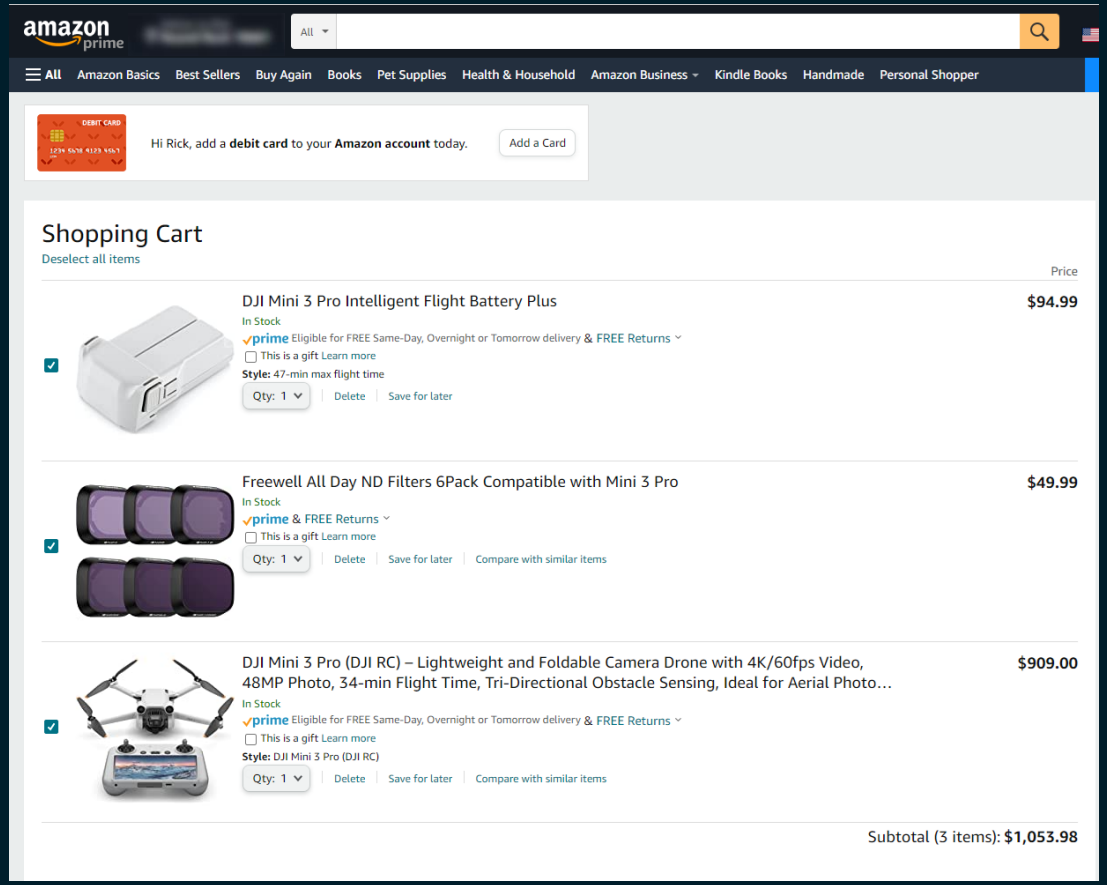
“Plans are only good intentions unless they immediately degenerate into hard work.”

- Peter Drucker

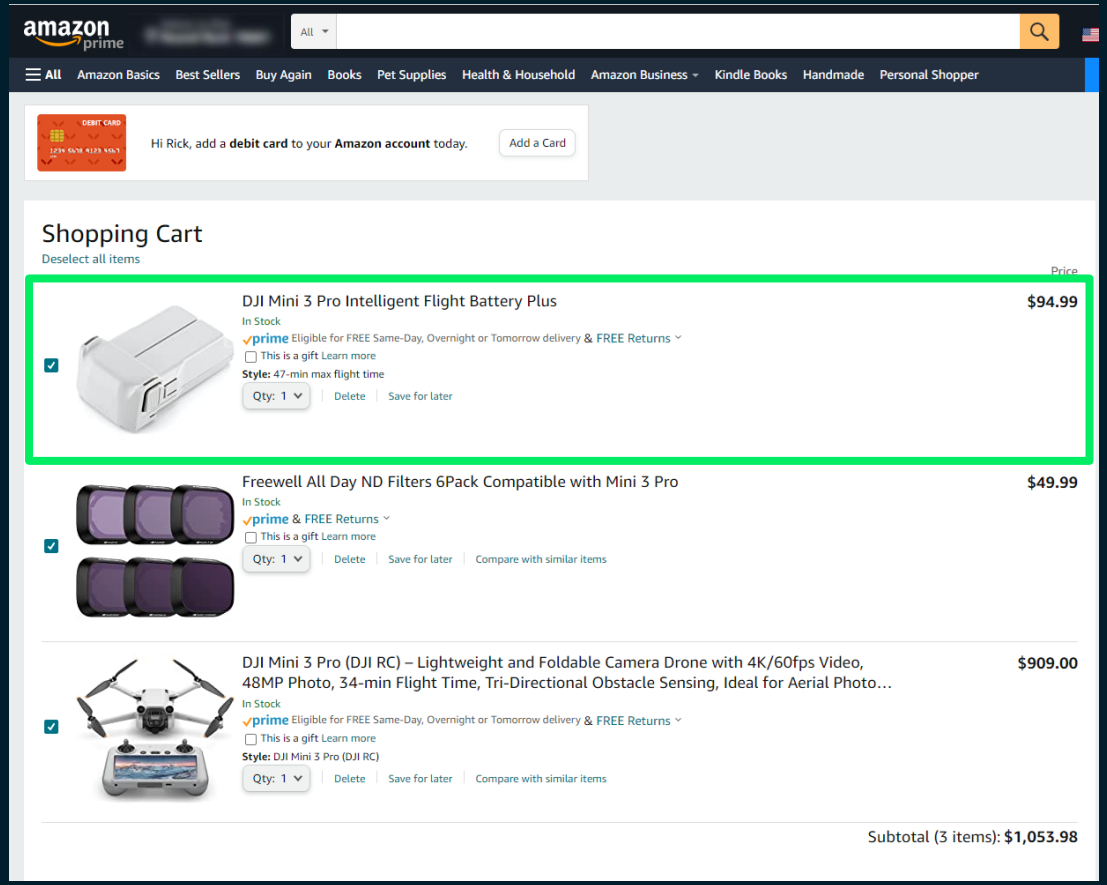
Technology Adoption and the Hype Curve



Making the Amazon shopping cart work

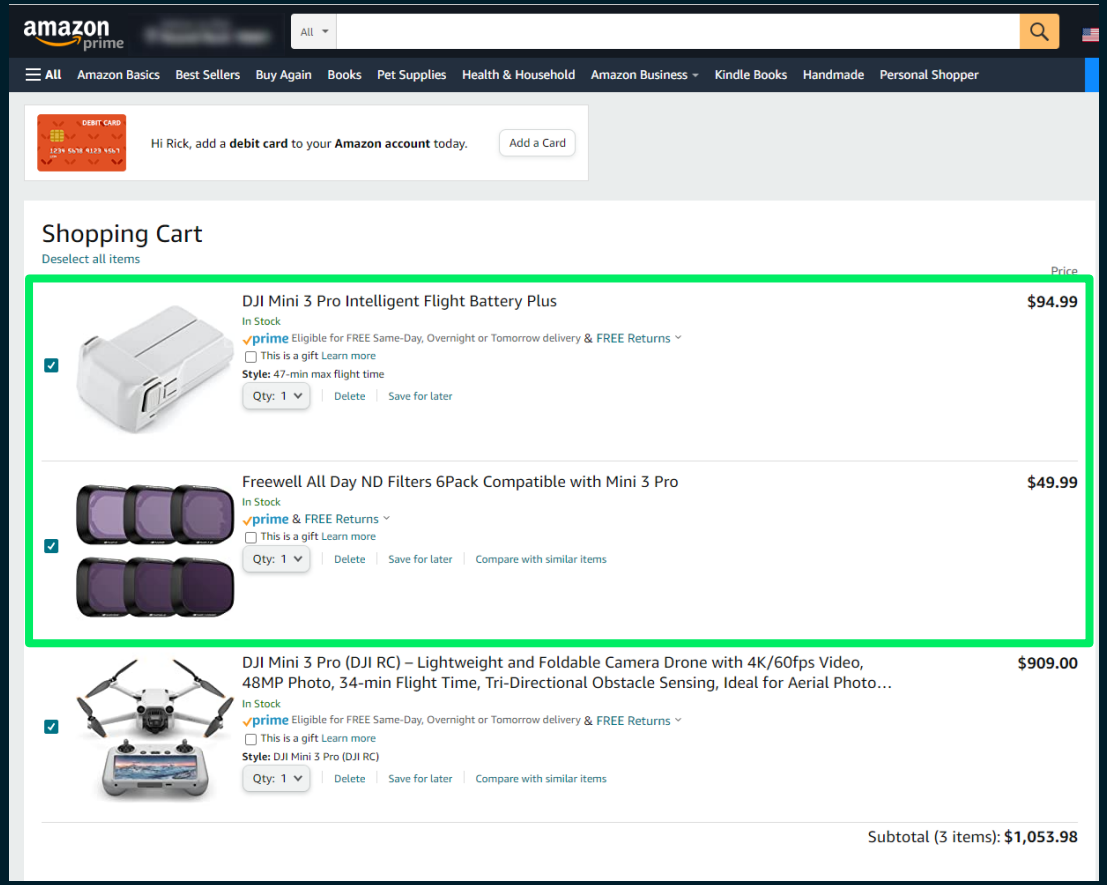


Making the Amazon shopping cart work



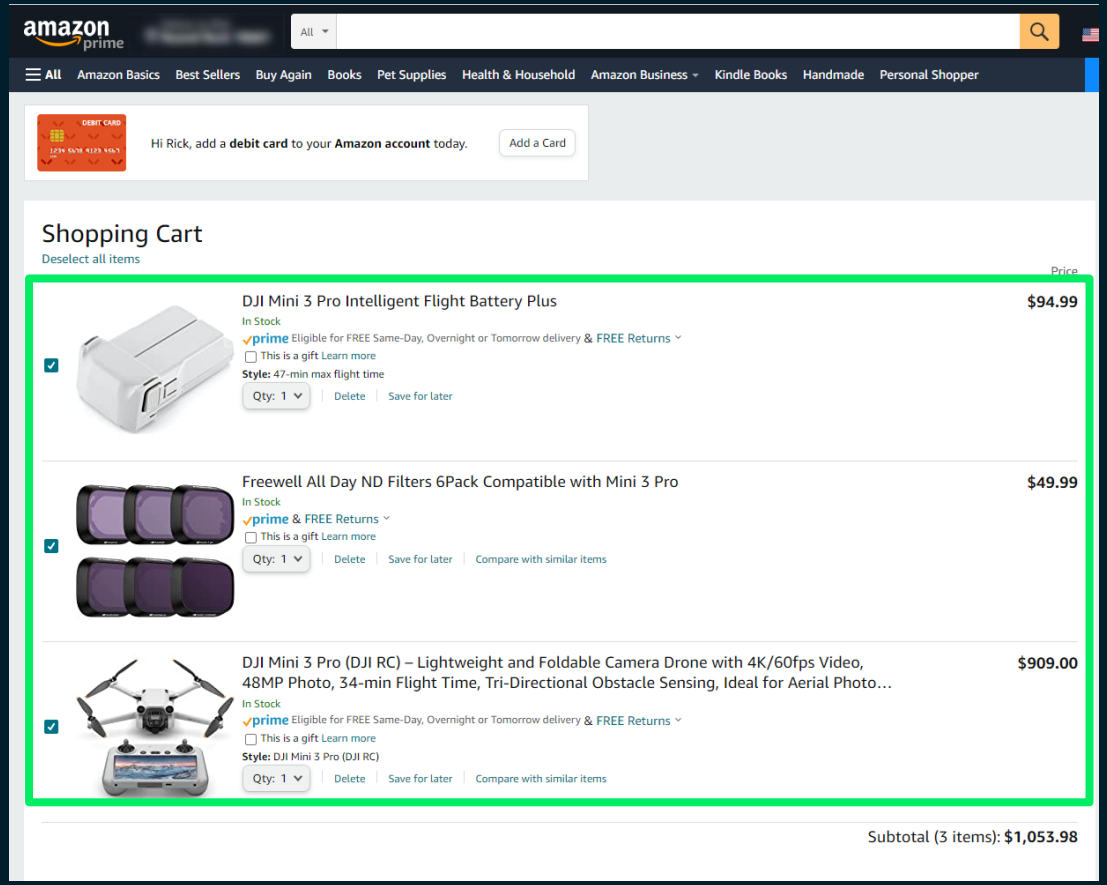
1 KB = 1 WCU

Making the Amazon shopping cart work



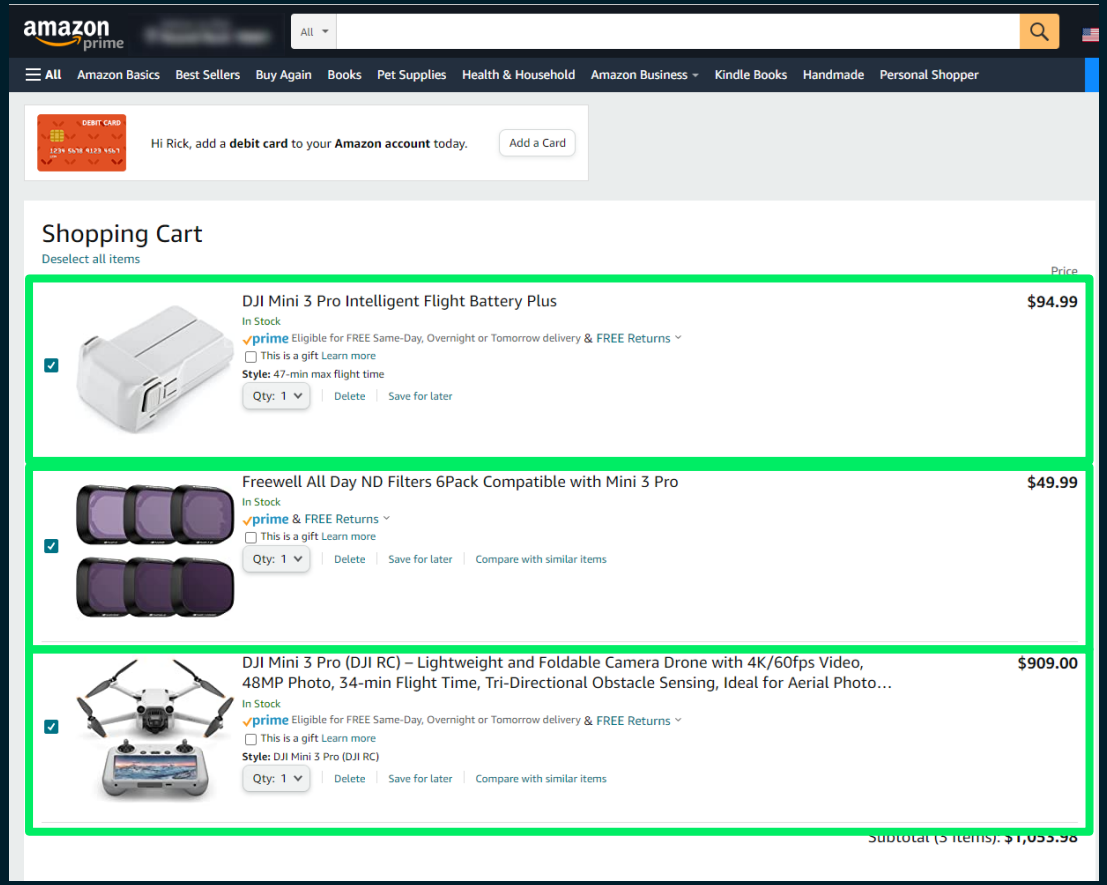
2 KB = 2 WCU

Making the Amazon shopping cart work



3 KB = 3 WCU

Making the Amazon shopping cart work

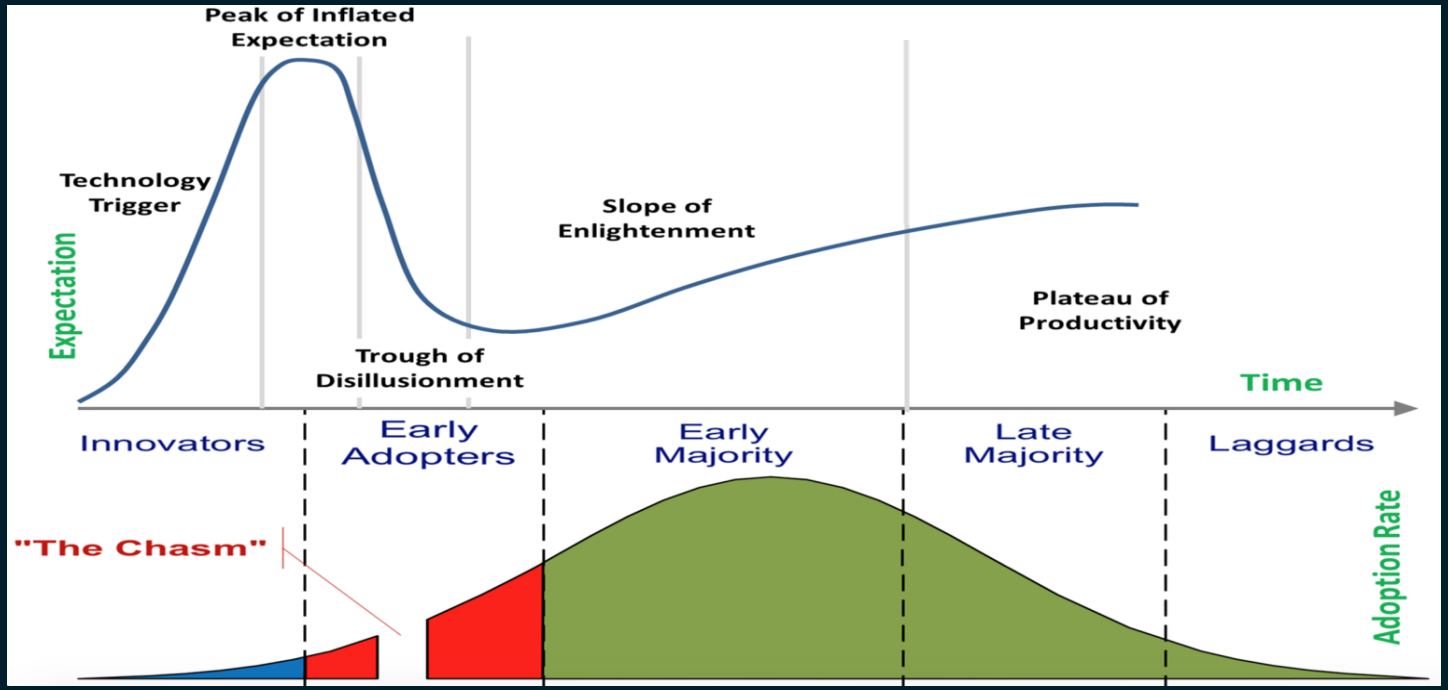


1 KB

1 KB

1 KB

Technology Adoption and the Hype Curve



Dive Deep on NoSQL at MongoDB Day

Tailored engagements to accelerate projects

- ❑ Detailed exploration of NoSQL Data Modeling
- ❑ Breakout content covering specific areas of interest
- ❑ Whiteboard sessions with subject matter experts
- ❑ Reach out to your Account Rep today

Thinking in Documents

“Plans are only good intentions unless they immediately degenerate into hard work.”

- Peter Drucker

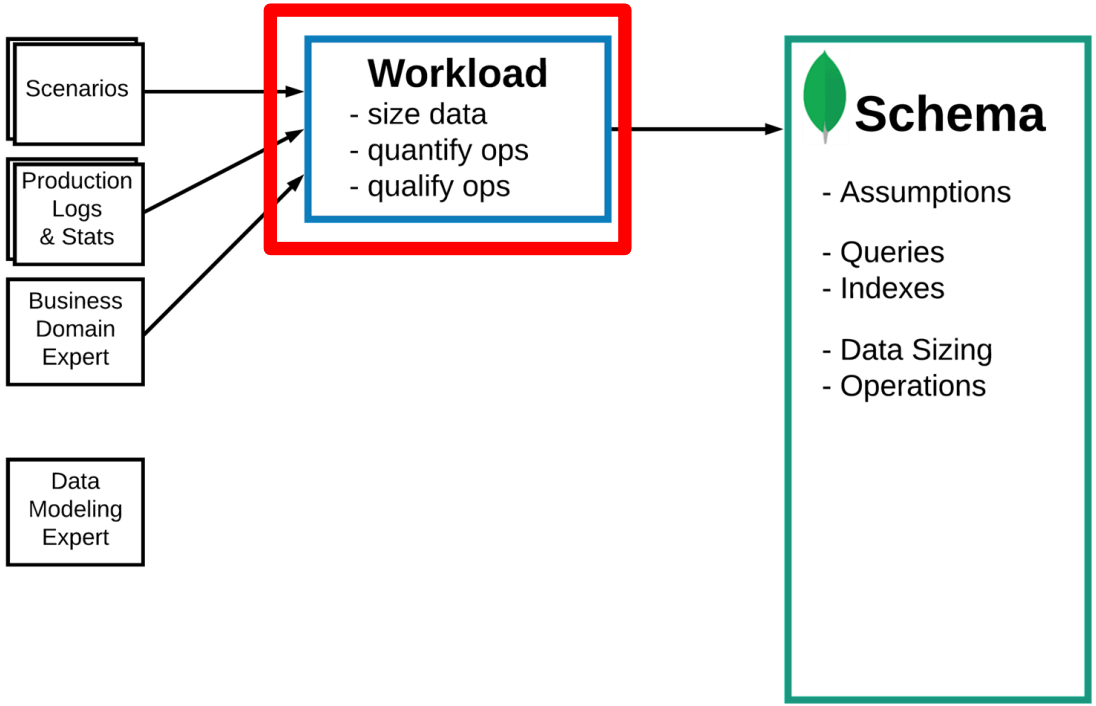
Methodology

A quick overview of NoSQL design methodology



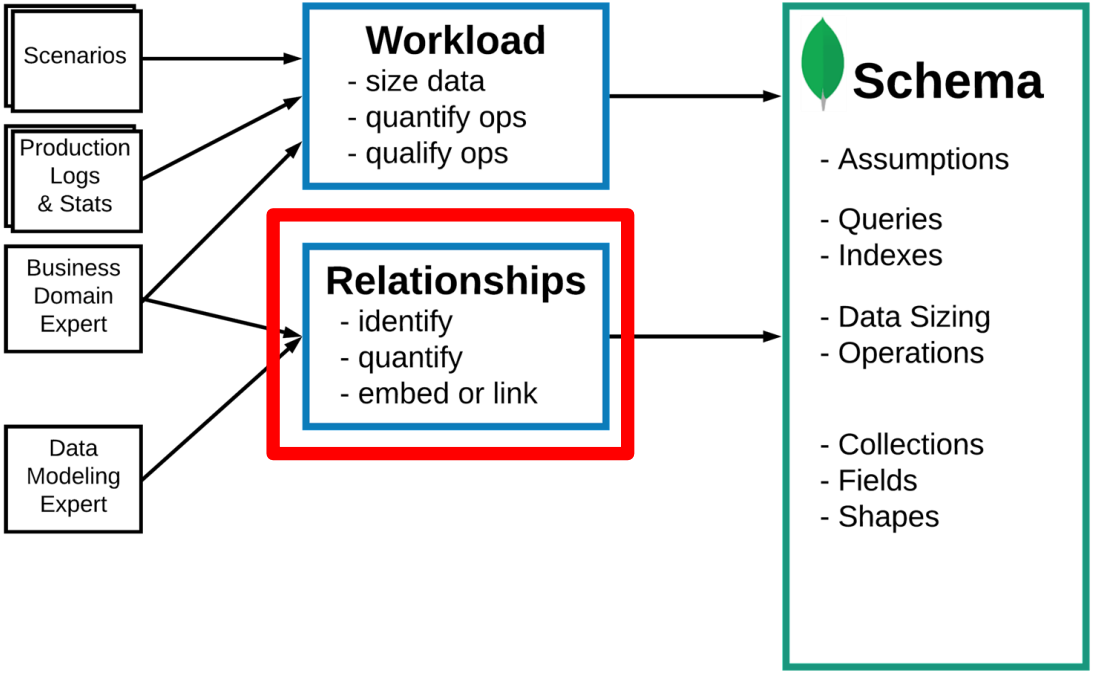
Methodology

1. **Describe the Workload**



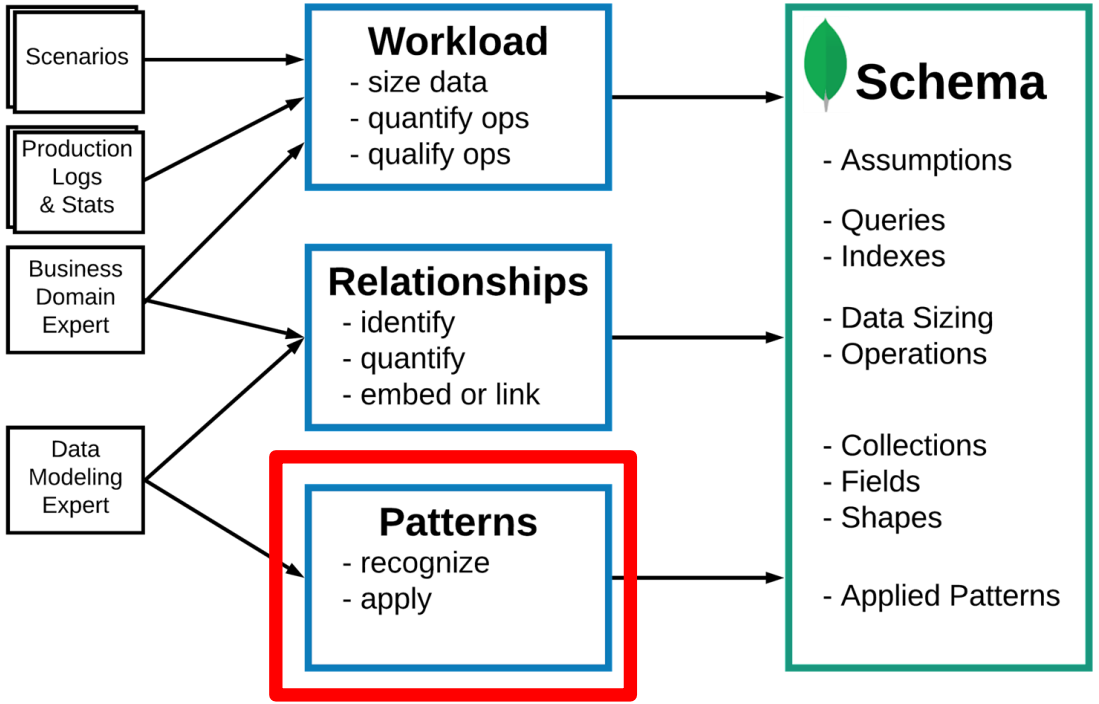
Methodology

1. Describe the Workload
2. **Identify and Model the Relationships**



Methodology

1. Describe the Workload
2. Identify and Model the Relationships
3. **Apply Patterns**



Flexible Methodology

Goal	*Simplicity*	*Performance*
1. Describe the **Workload**	Model for generic patterns	Optimize for common patterns while accommodating corner cases
2. Model the **Relationships**	Mostly embedding	Embedding and linking, use indexes to group related documents
3. Apply **Patterns**	Embedding	Referencing - Extended, Single Collection Bucketing Computed Aggregations …



Use Case: Single View

Mongo Insurance Corporation needs to unify legacy systems for a customer portal



Mongo Insurance Portal

Merging legacy systems

Superfast user interface

Analytics in batch

Future base for customer portal

1 – Workload: list the operations

Query	Operation	Description
1. Sync changes to MIP	write	Change stream from legacy databases to MIP's MongoDB database
2. Load user profile	read	Create a session with the user information
3. Load overview of account	read	Gather all information for the main page: policies, claims, payments, documents, and messages
4. Load detail of an artifact	read	Access a given policy, claim, payment, document, or message



1 – Workload: quantify/qualify the operations

Query	Quantification	Qualification
1. Sync changes to MIP	Streaming updates peak 2K tps	critical write
2. Load user profile	12 reads/sec (1M reads/day) < 2 ms SLA	no stale data
3. Load overview of account	12 reads/sec (10M reads/day) < 2 ms SLA	no stale data
4. Load detail of an artifact	115 reads/sec (100 reads/day) < 2 ms SLA	no stale data



Mongo Insurance Corporation

John Doe

Policies

- Home
- Car
- Boat

Open Claims 2

Billing

- Due: $800

Documents 8

Messages 25

Car Insurance Details

Car 1 - Supercar XLT (2020)
4 wheel drive, 8 cyl

Premium: $2000

Car 2 – Compact Car (2010)
2 wheel drive, 4 cyl
Premium: $800

Open claims
Collision
Damaged rear bumper
Trunk body damage

Find a partner repair center

54323 Views

Mongo Insurance Corporation

John Doe

Policies

- Home
- Car
- Earthquake

Open Claims 2

Billing

- Due: $800

Documents 8

Messages 25

Documents and Statements

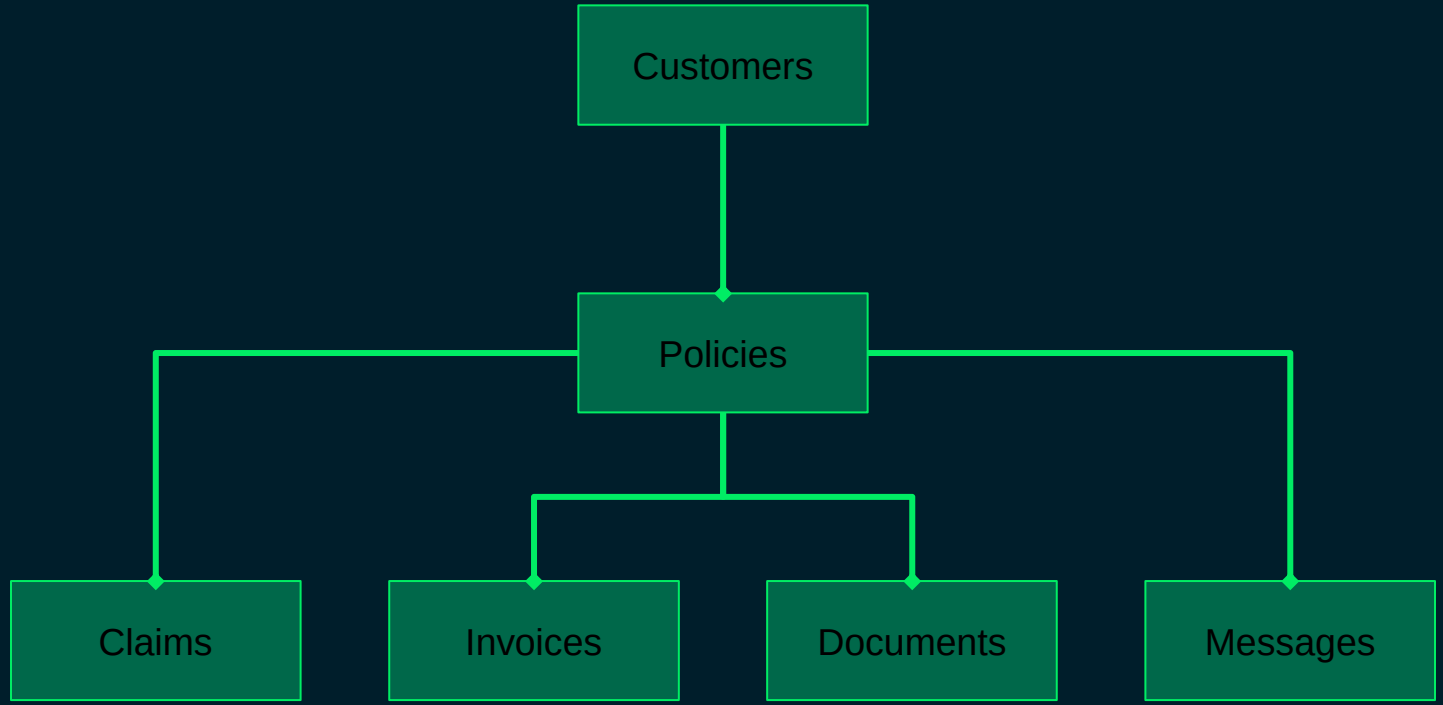
Total documents: 8

New policy inquiry	2020/02/14
Bill past due	2019/11/03
Bill past due (2nd notice)	2019/12/03
Your new claim	2019/04/20
Your claim status updated	2019/04/15
Your new claim	2019/04/13

next >>

54323 Views

2 - Entities for MongoDB Insurance Portal



Applying Patterns

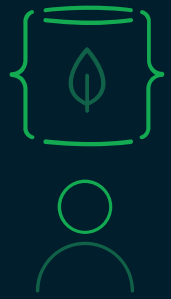
Bucket, Computed, Extended Reference, Outlier, Approximation

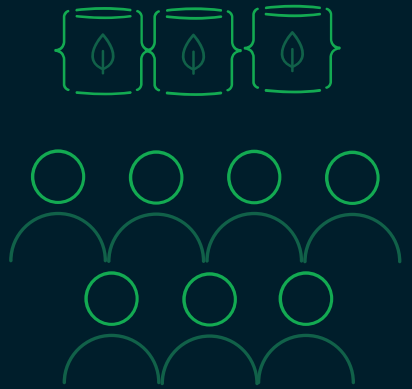


Main Tradeoff in NoSQL Modeling

Simplicity

Performance





Mongo Insurance Corporation

John Doe

Policies
- Home
- Car
- Earthquake

Open Claims 2

Billing
- Due: $800

Documents 8

Messages 25

Car Insurance Details

Car 1 - Supercar XLT (2020)
4 wheel drive, 8 cyl

Premium: $2000

Car 2 – Compact Car (2010)
2 wheel drive, 4 cyl
Premium: $800

Open claims
Collision
Damaged rear bumper
Trunk body damage

Simplicity versus Performance

One large document for all user data?

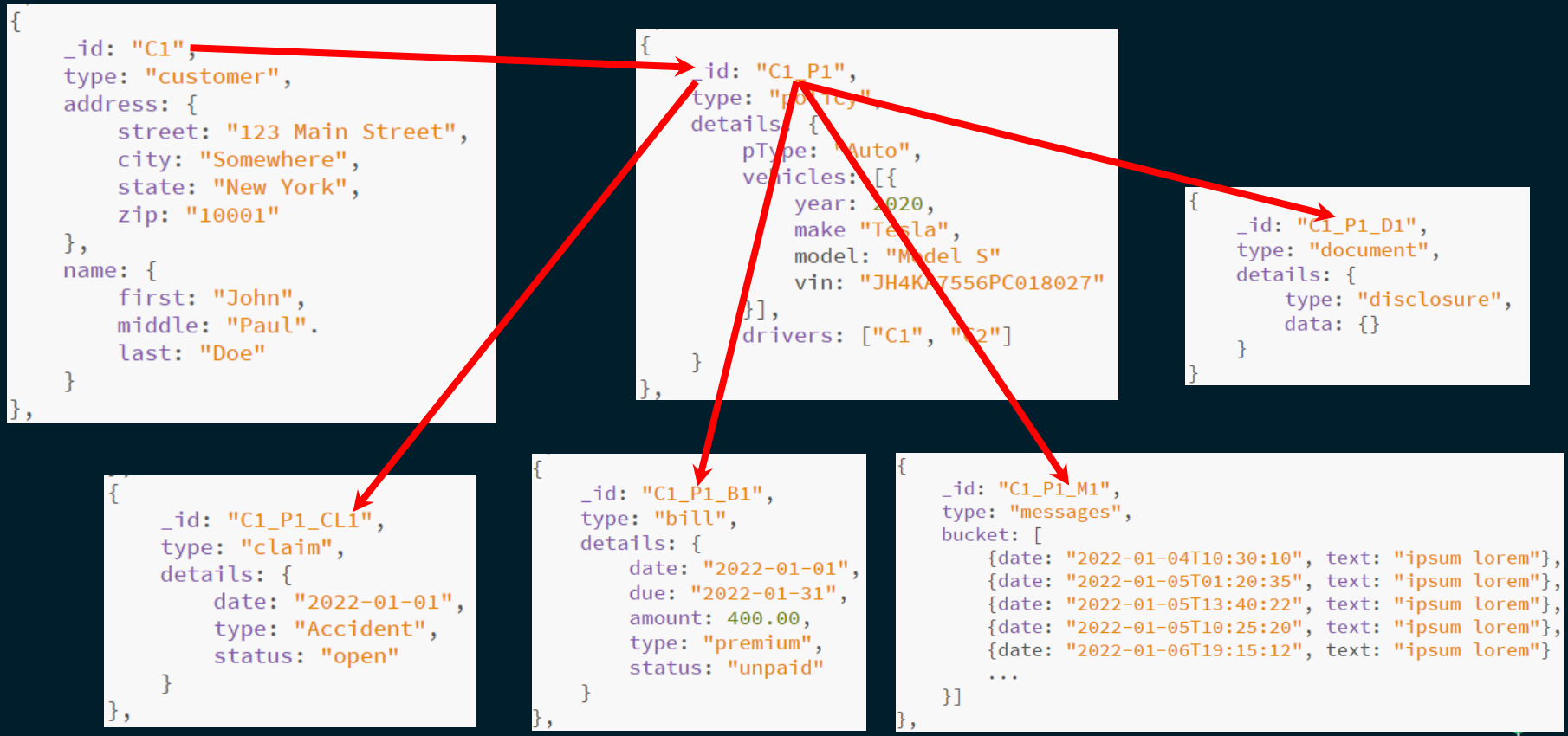
Separate documents for policies, claims, billing, etc?

54323 Views

Extended Reference pattern

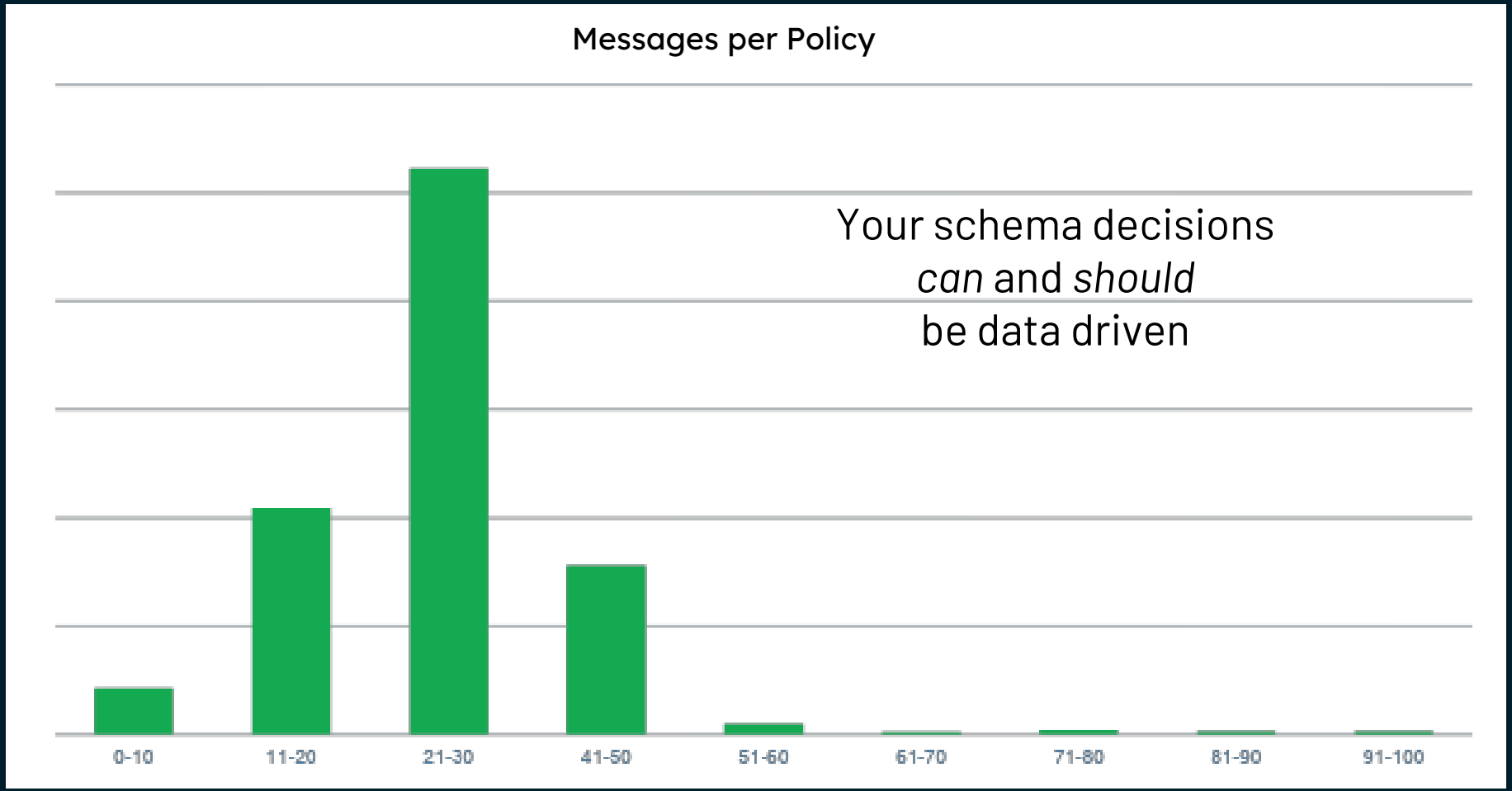
- Keep a “reference” to one document in another
- Efficiently query documents to support complex access patterns by indexing shared attributes
- Denormalize to increase read performance
- MongoDB has $lookup for JOIN, but use it sparingly
- One read is faster than many

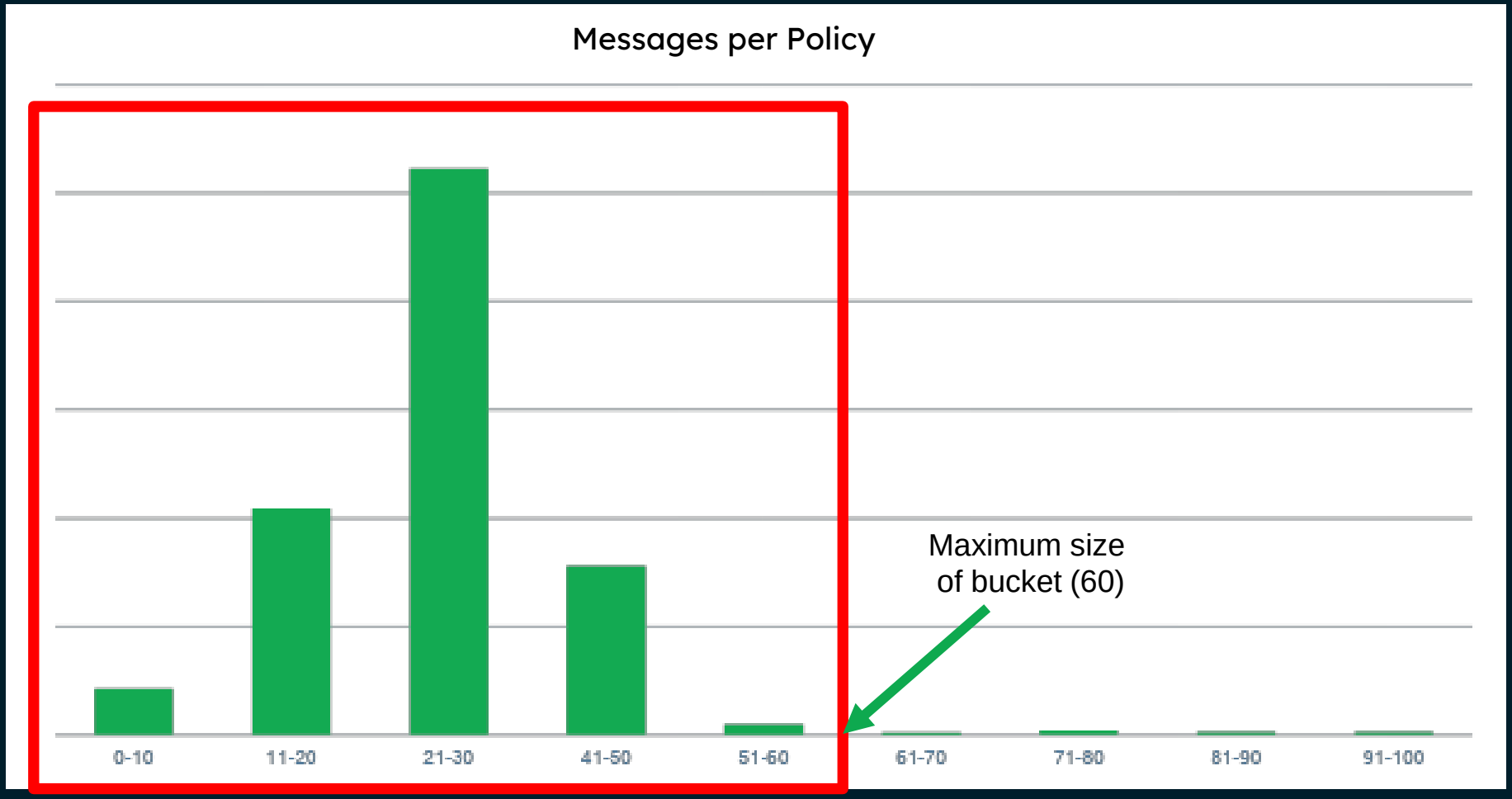
Extended References and Index Overloading

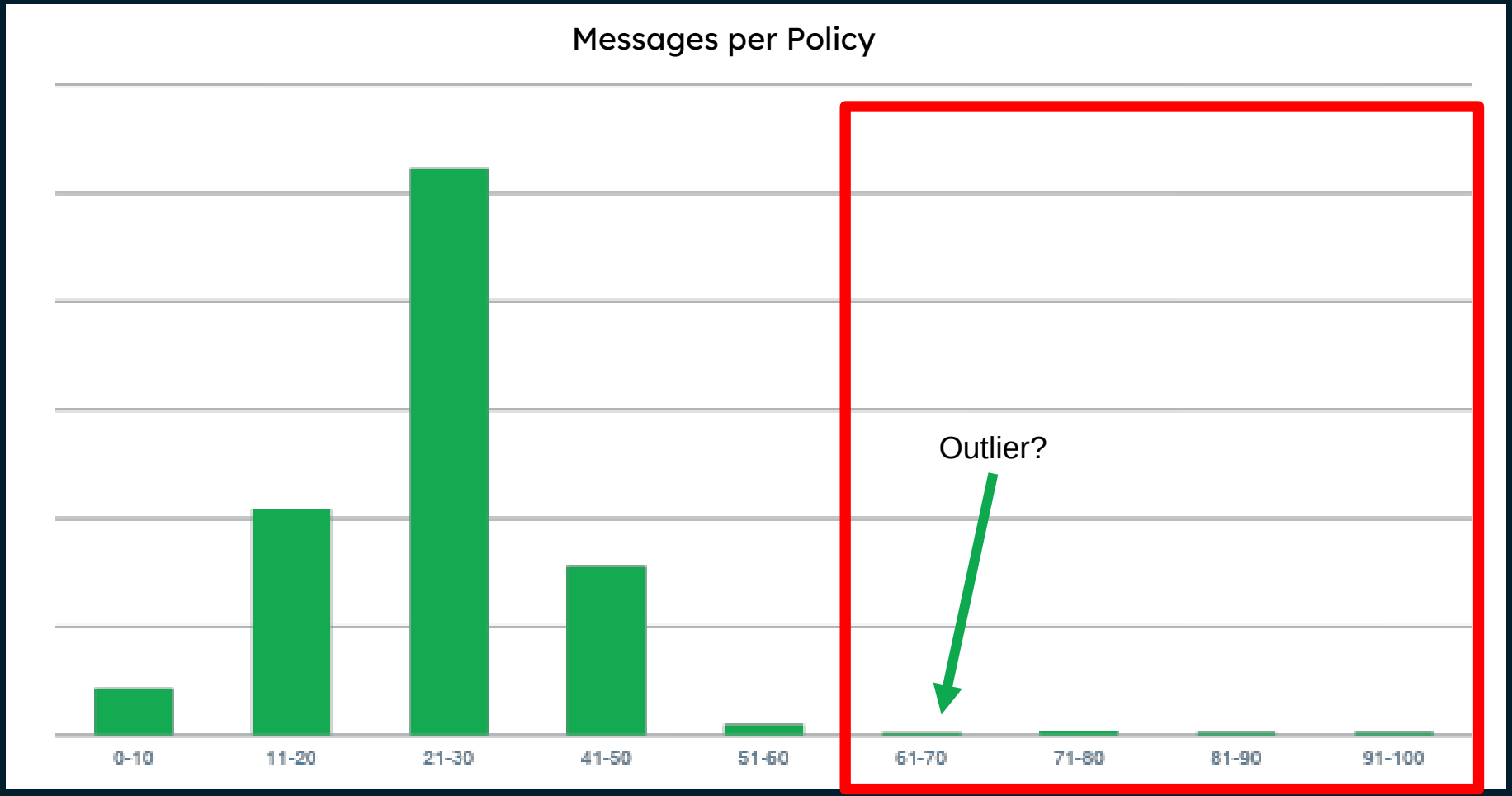


Outlier pattern

- Optimize for the most common access patterns
- Overflow large data across multiple documents when needed
- Use indexes to collect overflow documents efficiently when reading
- Keep documents reasonably sized


Messages per Policy
Your schema decisions
can and *should*
be data driven
0-10
11-20
21-30
41-50
51-60
61-70
71-80
81-90
91-100


Messages per Policy
Maximum size of bucket (60)
0-10
11-20
21-30
41-50
51-60
61-70
71-80
81-90
91-100


Messages per Policy
Outlier?
0-10
11-20
21-30
41-50
51-60
61-70
71-80
81-90
91-100

Bucket and Overflow Documents

Message Bucket

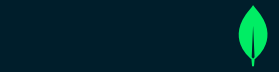
```
{
    _id: 'C1_P1_M1',
    type: "messages",
    bucket: [
        {date: "2022-01-04T10:30:10", text: "ipsum lorem"},
        {date: "2022-01-05T01:20:35", text: "ipsum lorem"},
        {date: "2022-01-05T13:40:22", text: "ipsum lorem"},
        {date: "2022-01-05T10:25:20", text: "ipsum lorem"},
        {date: "2022-01-06T19:15:12", text: "ipsum lorem"}
        ...
    }]
},
```

Message Bucket Overflow

```
{
    _id: "C1_P1_M2",
    type: "messages",
    bucket: [
        {date: "2022-01-04T10:30:10", text: "ipsum lorem"},
        {date: "2022-01-05T01:20:35", text: "ipsum lorem"},
        {date: "2022-01-05T13:40:22", text: "ipsum lorem"},
        {date: "2022-01-05T10:25:20", text: "ipsum lorem"},
        {date: "2022-01-06T19:15:12", text: "ipsum lorem"}
        ...
    }]
},
```



Conclusions



NoSQL does not mean non-relational

The ERD still matters

RDBMS efficiency is declining

NoSQL is a great fit for most OLTP workloads

The MongoDB API supports OLAP just as well

Thank you!



Please complete the session survey in the **mobile app**

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.