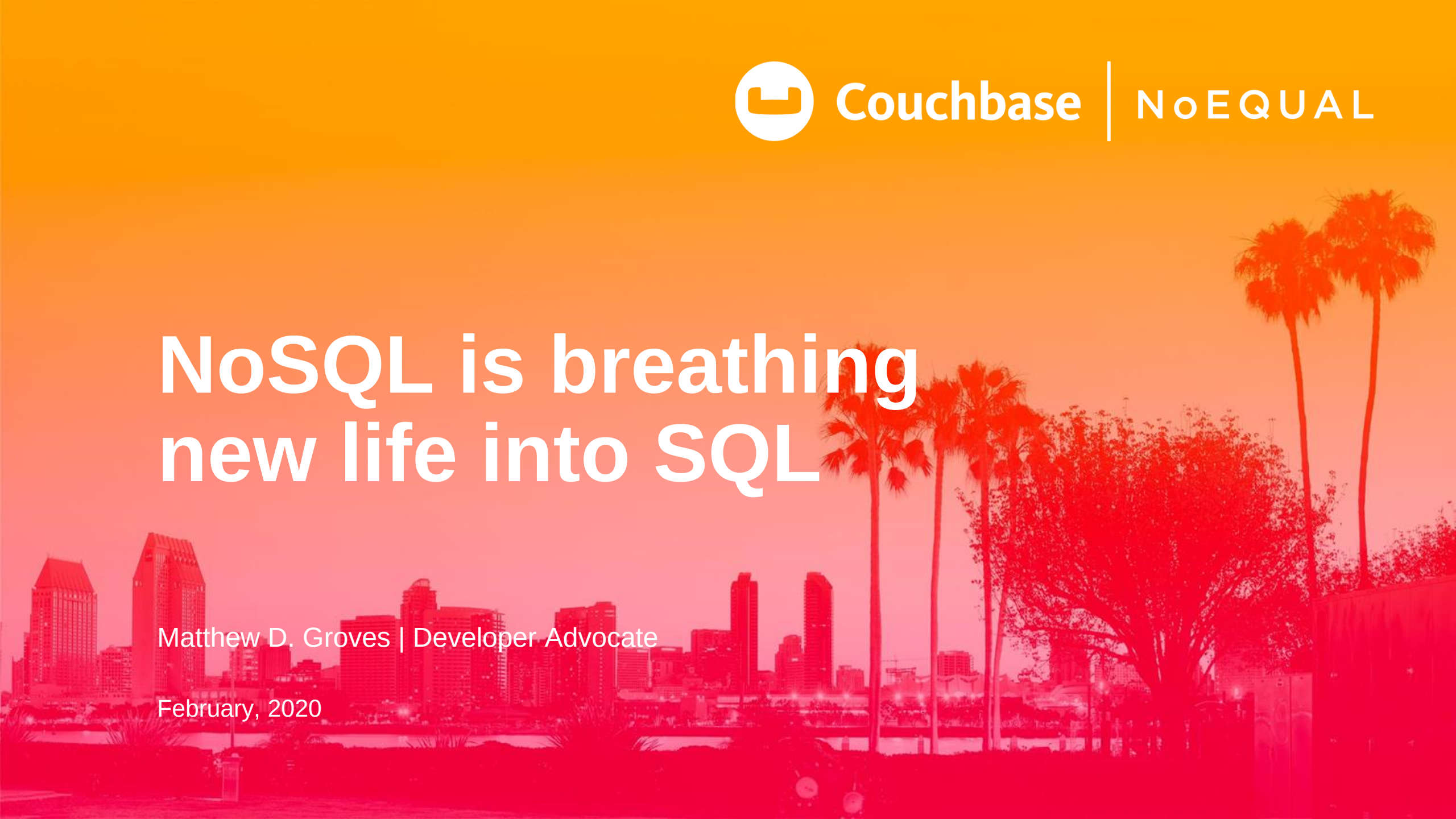


NoSQL is breathing new life into SQL

Matthew D. Groves | Developer Advocate

February, 2020

SQL, for the win

https://insights.stackoverflow.com/survey/2019

Then what's with this NoSQL stuff?

AGENDA

01/ SQL History

02/ Why NoSQL? Why JSON?

03/ Querying JSON

04/ Next Steps

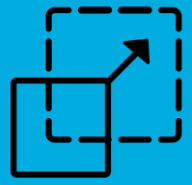
1 | SQL History

A History of SQL

- Created by Don Chamberlain & Raymond Boyce
- Designed to be an English-friendly way to query relational data
- "SQL" and "relational" are synonyms

2 Why NoSQL? Why JSON?

History: Criticisms / drawbacks of relational



Scaling

- Vertical scaling is "easy" but expensive and has a ceiling
- Horizontal scaling is difficult with traditional relational model.

Impedance Mismatch

- Object-oriented programming languages
- Data in memory is different than data in the database

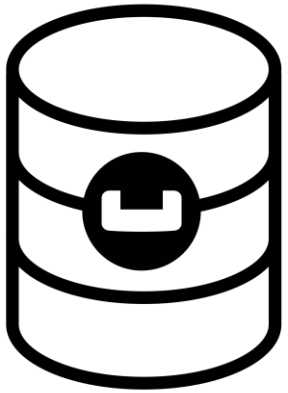
Inflexibility

- Relational requires strict schema and constraints
- Adjusting schemas can have major impact on operations

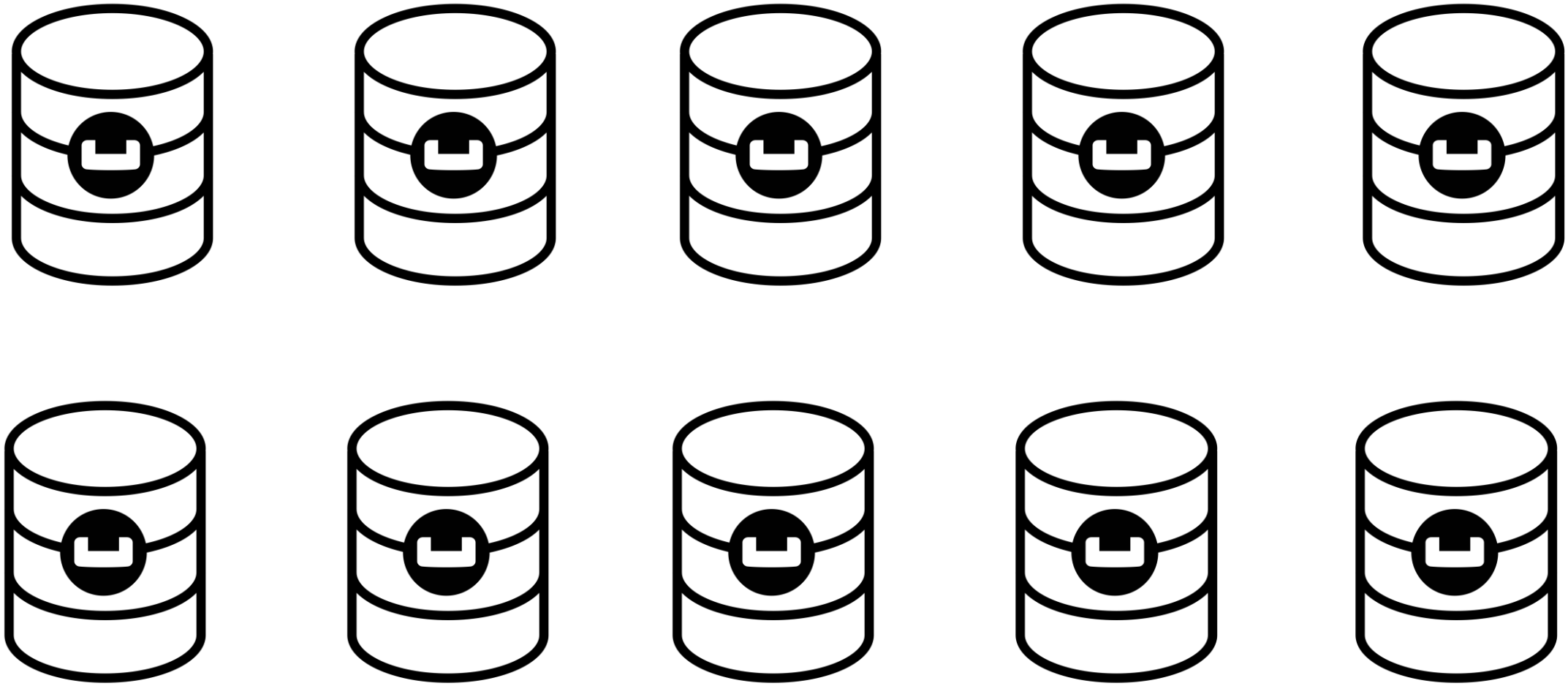
Performance

- Assembling / disassembling objects
- Limited to SQL-based access methods

History: Scaling with NoSQL



History: Scaling with NoSQL



History: JSON

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <name>Matt</name>
  <favoriteFoods>pizza</favoriteFoods>
  <favoriteFoods>cheesecake</favoriteFoods>
</root>
```

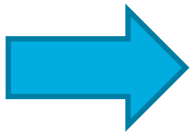
```json
{
  "name": "Matt",
  "favoriteFoods": [
    "pizza",
    "cheesecake"
  ]
}
```

3 | Querying JSON

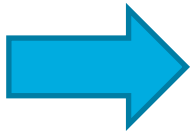
Querying Tables

person

ID	Name	Age
1	Matt	40
2	Emma	9
3	Ali	36



```
SELECT Name
FROM person
WHERE Age >= 21
```



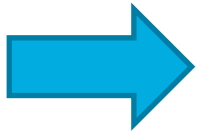
Name
Matt
Ali

Querying JSON

```
[
    { "<key>": 1, "Name": "Matt", "Age": 40  },

    { "<key>": 2, "Name": "Emma", "Age": 9  },

    { "<key>": 3, "Name": "Ali", "Age": 21 }
]
```

Querying JSON: MongoDb

```
[ {"_id": 1,
   "Name": "Matt",
   "Age": 40  },

  {"_id": 2,
  "Name": "Emma",
  "Age": 9  },

  {"_id": 3,
  "Name": "Ali",
  "Age": 21}
]
```

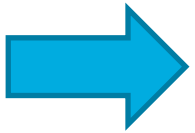


```
db.people.find( { Age: { $gte: 21 } }, { "Name" : 1 })
```

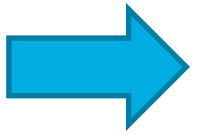
```
[
  { "_id": 1, "Name" : "Matt" },
  { "_id": 3, "Name" : "Ali" }
]
```

Querying JSON: N1QL

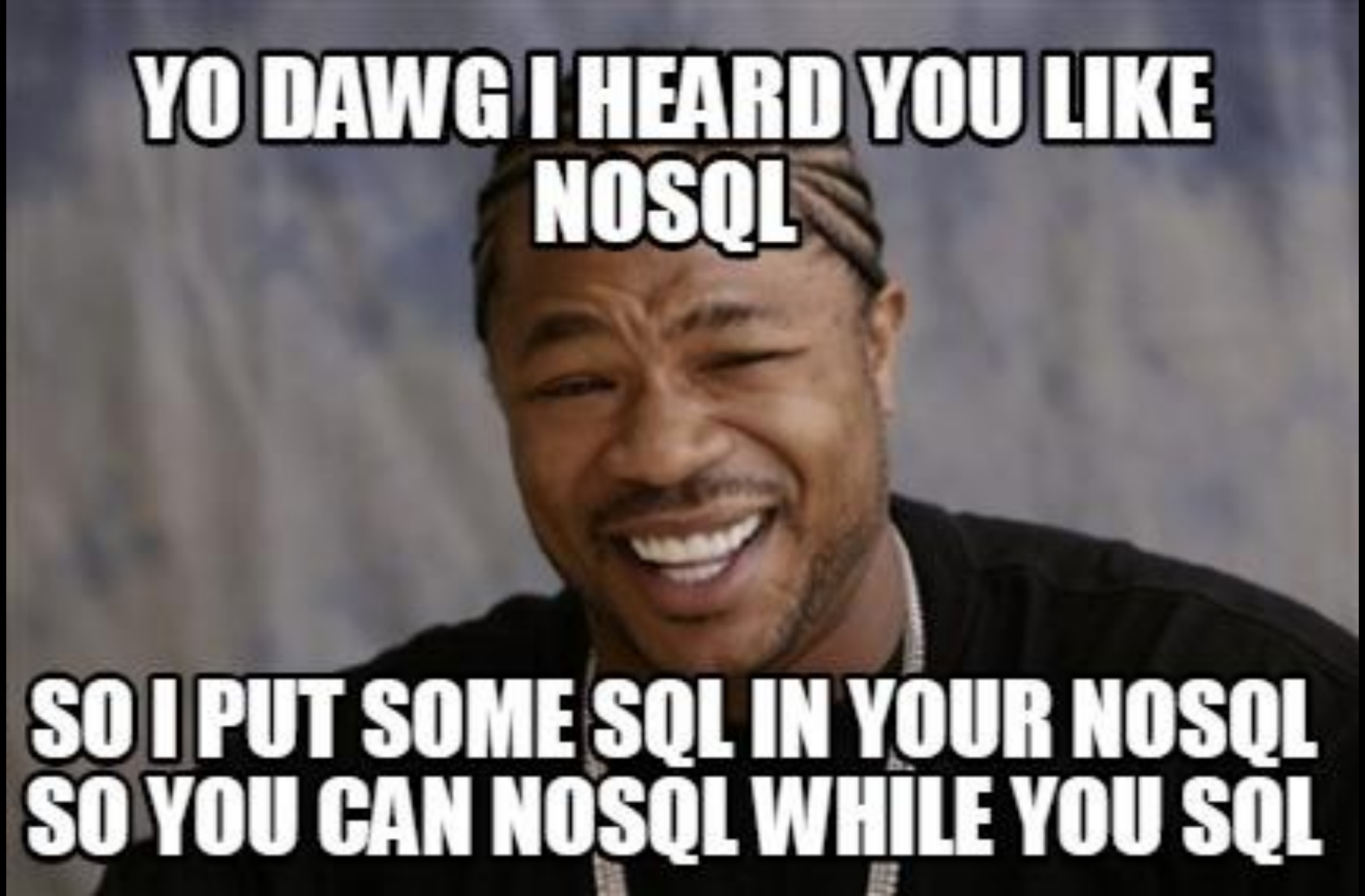
```
[ {"<key>": 1,
   "Name": "Matt",
   "Age": 40  },

  {"<key>": 2,
  "Name": "Emma",
  "Age": 9  },

  {"<key>": 3,
  "Name": "Ali",
  "Age": 21}
]
```



```
SELECT Name
FROM person
WHERE Age >= 21
```



```
[
  { "Name" : "Matt" },
  { "Name" : "Ali" }
]
```


YO DAWG I HEARD YOU LIKE NOSQL
SO I PUT SOME SQL IN YOUR NOSQL SO YOU CAN NOSQL WHILE YOU SQL

SQL for NoSQL Paradox

But it's not tables…?

- Yes, need to extend the language
- SQL++ (research paper)
- "JSON looks like tables if you squint" – Don Chamberlain, co-inventor of SQL

Missing fields?

- IS MISSING
- IS NOT MISSING

JOINs?

- Inter-document JOIN / INNER JOIN / LEFT JOIN
- Intra-document UNNEST

Performance?

- Indexing is still important
- Alternate APIs: key/value, full text search, etc

N1QL and Mongo side-by-side

```
SELECT Name
FROM person
WHERE Age >= 21
```

```
db.people
.find(
  { Age: { $gte: 21 } },
  { "Name" : 1 }
)
```

N1QL and Mongo side-by-side

```
SELECT DISTINCT
      r.destinationairport
FROM travel a
   JOIN travel r
      ON (
            a.faa = r.sourceairport
            AND r.type = "route"
      )
WHERE a.type = "airport"
   AND a.city = "San Francisco"
   AND a.country = "United States"
ORDER BY r.destinationairport
```

```
db.airport.aggregate([
  {
     $match: {
         $and: [
           {"type": "airport"},
           { city: "San Francisco"},
           { "country": "United States"}
         ]
     }
  },
  {
    $lookup:
     {
       from:"route",
       let: { rfaa : "$faa"},
       pipeline: [
         { $match:
             { $expr:
               { $and:
                 [
                   { $eq: ["$sourceairport", "$$rfaa"]} ,
                   { $eq: ["$type", "route"] }
                 ]

               }
             }
         ],
         as: "airline_docs"
       }
  },
  { $match: {"airline_docs": {$ne: []}} },
  {  $unwind: { path: "$airline_docs", preserveNullAndEmptyArrays: true }},
  { $project: { _id:0,  "airline_docs.destinationairport" : 1 }},
  { $group: {
     _id : "$airline_docs.destinationairport"
    }
  },
  { $sort: { _id : 1 }},
]);
```

4 | Next Steps

Next Steps

- Come talk to us at the Couchbase booth!
- Download and try Couchbase
  - https://couchbase.com/downloads
- Try N1QL right in your browser:
  - https://www.couchbase.com/n1ql
- Contact me:
  - @mgroves on Twitter
  - twitch.tv/matthewdgroves


THANK YOU
Couchbase
NoEQUAL