Thank You to the Code PaLOUsa Sponsors

Progress

WAYSTAR

Bastian SOLUTIONS
a TOYOTA ADVANCED LOGISTICS company

<code> LOUISVILLE

eblu SOLUTIONS

twilio

Scout

aws

cbts

<prosoft>

CGI

Friends of Code PaLOUsa

Asperitas

Couchbase

DATASTAX

mongoDB

redis

ROCKET Mortgage

Couchbase

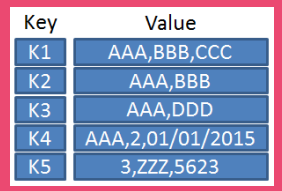
Legacy Relational to Modern NoSQL in Three Steps

Presented by Matthew D. Groves

Are These Your Goals?

- Identify a use case that can benefit from a modern database
- Get some/all of data on to a modern database
- Switch your application / Build a new application
- Optimize your data model / access

Couchbase: The Modern NoSQL Database



Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

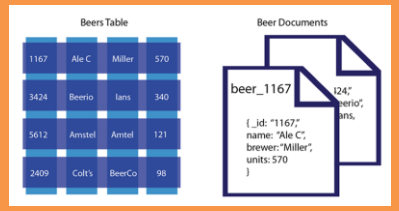
Key-Value Store

Sub-millisecond performance

Simplicity & Scalability

UC: Caching, Cart, IOT, User Profiles & Sessions

Compare with: Redis, Elasticache, Aerospike



Document Database

Flexible JSON schema maps to RDBMS

SQL++ is SQL for JSON

UC: Content Management, Catalog, Metadata, Customer 360

Compare with: DynamoDB, MongoDB, CosmosDB

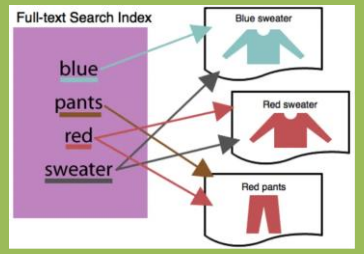
Edge Database

Cloud to Edge Sync

Peer to Peer Sync

UC: Field Apps, Airplanes, Ships, Retail Stores, Restaurants, IOT Apps

Compare with: Realm, SQLite, Berkeley DB



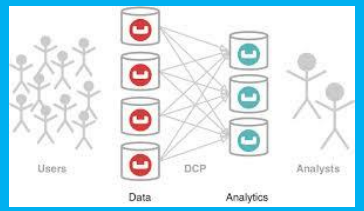
Search Engine

Full-text Search

Inverted Index

UC: Log analysis, Collaboration, Website Search

Compare with: Elasticsearch, Solr, Lucene



Analytics HTAP/HOAP

No ETL, SQL++, MPP

Workload Isolation

UC: Real-time Insights, Fraud detection, AI, Recommendation engines

Compare with: MemSQL, CockroachDB, Hana

Who am I?

- Matthew D. Groves
- Microsoft MVP, author at Pluralsight, Manning, Apress
- Product Marketing Manager at Couchbase
- https://twitter.com/mgroves
- https://www.linkedin.com/in/mgroves/
- C# Advent: https://csadvent.christmas/

Agenda

01/ Mapping from relational

02/ Shift Application Code

03/ Optimize

1 Mapping from relational

Why now?

- Couchbase Capella
- Scopes / collections
- ACID Transactions
- SQL++ (née N1QL)

So, is NoSQL just relational database now?

- Distributed
- Highly available
- Flexible
- Memory-first

SQL-to-Couchbase Dictionary

Legacy (relational)	Modern (Couchbase)	Notes
Server	Cluster	+ Scalability / High Availability
Database	Bucket	+ Replication / Caching
Schema	Scope	"dbo" and "_default"
Table	Collection	+ Flexibility
Row	Document	+ JSON
tSQL (SQL)	SQL++	Full SQL implementation with: SELECT, JOIN, COMMIT, ROLLBACK, GROUP BY, INSERT, etc…
Primary Key	Document Key	
Index	Index	

SqlToCouchbase Library

Search or jump to... /

Pull requests Issues Marketplace Explore

mgroves / SqlServerToCouchbase

Unwatch

Code | Issues 2 | Pull requests | Actions | Projects | Wiki | Security | Insights

master | 1 branch | 0 tags | Go to file | Add file | Code

mgroves Merge pull request #10 from mgroves/denormalize ... 9193904 on Jul 16 38 commits

SqlServerToCouchbase.Console	refactoring to reduce same sql query in multiple places	2 months ago
SqlServerToCouchbase.Tests	comments cleanup, upgrading to latest SDK releases	4 months ago
SqlServerToCouchbase	added more logging	last month
WebApiExample.Couchbase	upgrading couchbase SDK, some other minor tweaks/improvements	3 months ago

Other Options/Tools

- Dataworkz - https://dataworkz.com/
  - Real-time sync and model refactoring
- GlueSync - https://gluesync.com/
  - Real-time sync between NoSQL and SQL Server / Oracle
  - Especially good for mobile
- Couchgres - https://github.com/metonymic-smokey/couchgres
  - Open source community project
  - Migrate from Postgres to Couchbase
- Python script - https://bit.ly/mysqlPython
  - Bare-bones, script-based approach
  - Import CSV via UI

Demo: Mapping

2 Shift Application Code

What do I need to shift?

- SDK – connecting to the database
- SQL – querying the database
- ACID Transactions – updating data reliably

SQL: Reuse

Official Microsoft example of tSQL

```
SELECT RTRIM(p.FirstName) + ' ' + LTRIM(p.LastName) AS Name, d.City
FROM AdventureWorks2016.Person.Person AS p
INNER JOIN AdventureWorks2016.HumanResources.Employee e
   ON p.BusinessEntityID = e.BusinessEntityID
INNER JOIN
   (SELECT bea.BusinessEntityID, a.City
    FROM AdventureWorks2016.Person.Address AS a
    INNER JOIN AdventureWorks2016.Person.BusinessEntityAddress AS bea
    ON a.AddressID = bea.AddressID) AS d
ON p.BusinessEntityID = d.BusinessEntityID
ORDER BY p.LastName, p.FirstName;
```

SQL: Reuse

SQL++ version of the same query (after converting data into Couchbase)

```
SELECT RTRIM(p.FirstName) || ' ' || LTRIM(p.LastName) AS Name, d.City
FROM AdventureWorks2016.Person.Person AS p
INNER JOIN AdventureWorks2016.HumanResources.Employee e
   ON p.BusinessEntityID = e.BusinessEntityID
INNER JOIN
   (SELECT bea.BusinessEntityID, a.City
    FROM AdventureWorks2016.Person.Address AS a
    INNER JOIN AdventureWorks2016.Person.BusinessEntityAddress AS bea
    ON a.AddressID = bea.AddressID) AS d
ON p.BusinessEntityID = d.BusinessEntityID
ORDER BY p.LastName, p.FirstName;
```

ACID Transactions: relational

C# example – SQL Server

```
public async Task<IActionResult> UpdatePurchaseOrderAsync(PersonUpdateApi personUpdateApi)
{
        var transaction = await _context.Database.BeginTransactionAsync();

        try
        {
                // update one or more rows of data, save changes
                await _context.SaveChangesAsync();

                // commit transaction
                await transaction.CommitAsync();

                return Ok($"Person {personUpdateApi.PersonId} name and email updated.");
        }
        catch (Exception ex)
        {
                // rollback transaction
                await transaction.RollbackAsync();
                return BadRequest("Something went wrong, transaction rolled back");
        }
}
```

ACID Transactions: Couchbase

C# Example – Couchbase Server

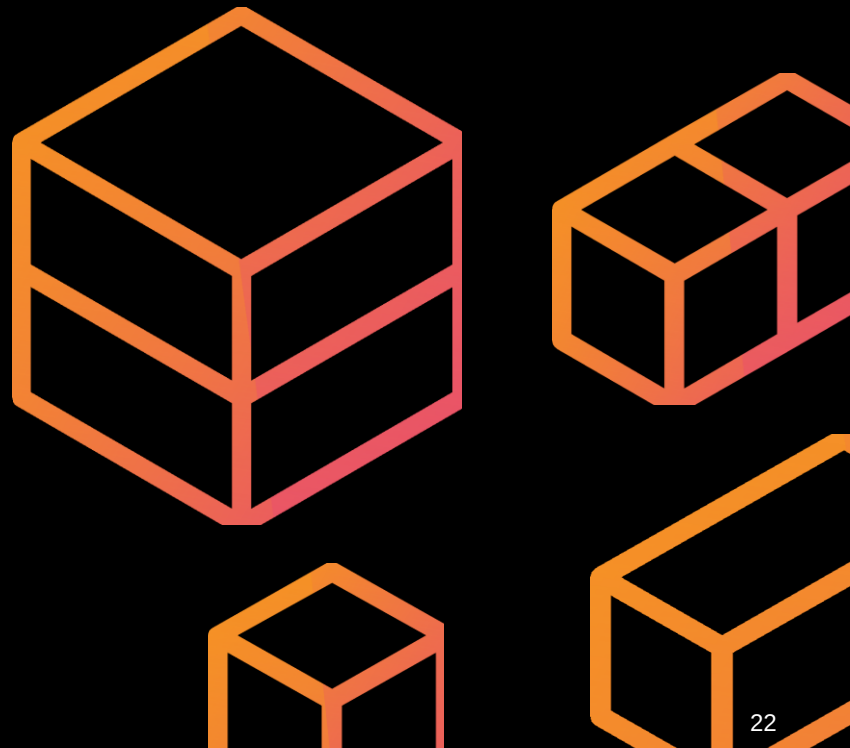
```
public async Task<IActionResult> UpdatePurchaseOrderAsync(PersonUpdateApi personUpdateApi)
{
        // create transaction
        var transaction = Transactions.Create(cluster,
                TransactionConfigBuilder.Create().DurabilityLevel(DurabilityLevel.None).Build());
        try
        {
                await transaction.RunAsync(async (context) =>
                {
                        // update one or more documents, save changes

                        // commit (implied)
                });
                return Ok($"Person {personUpdateApi.PersonId} name and email updated.");
        }
        catch (Exception ex)
        {
                // rollback (implied)
                return BadRequest("Something went wrong, transaction rolled back.");
        }
}
```

Demo: Shifting App

3 Optimize

Consolidate

- Relational-style data is spread out
- Requires JOINs and/or ACID transactions (overhead)
- De-normalize



Many-to-One consolidation

Person has 1 or more EmailAddress

```
{
  "PersonID": 123,
  "FirstName": "Ken",
  "MiddleName": "J",
  "LastName": "Sánchez",
  "ModifiedDate": "2009-01-07T00:00:00"
}

{
  "PersonID": 123,
  "EmailAddress": "ken0@adventureworks.com",
  "ModifiedDate": "2009-01-18T04:23:07"
}
```

```
{
  "PersonID": 123,
  "FirstName": "Ken",
  "MiddleName": "J",
  "LastName": "Sánchez",
  "ModifiedDate": "2009-01-07T00:00:00",
  "EmailAddresses": [
   {
    "PersonID": 123,
    "EmailAddress": "ken0@adventureworks.com",
    "ModifiedDate": "2009-01-18T04:23:07"
   }
  ]
}
```

One-to-One consolidation

PhoneNumber has a PhoneNumberType

```
{
  "PhoneNumberID": 10000,
  "PhoneNumber": "388-555-0157",
  "PhoneNumberTypeID": 1,
  "ModifiedDate": "2013-11-09T00:00:00"
}

{
  "PhoneNumberTypeID": 1,
  "Name": "Cell",
  "ModifiedDate": "2017-12-13T13:19:22.273"
}
```

Use the key-value API when possible

The key-value API is always faster

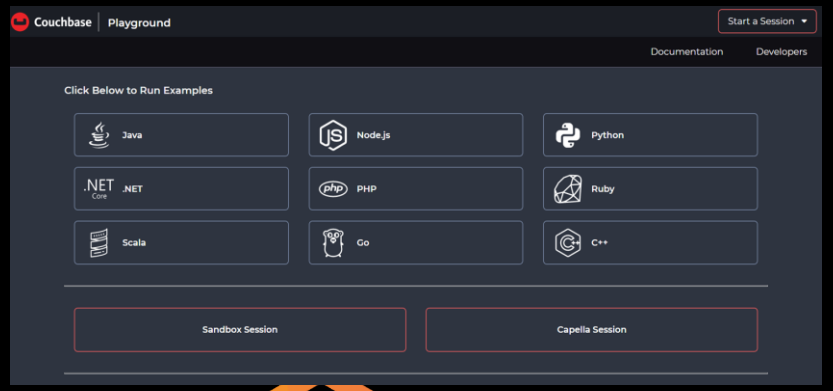
```sql
SELECT p.*

FROM AdventureWorks2016.Person.Person p

WHERE p.PersonID = 123
```

```
var person = await coll.GetAsync("123");
```

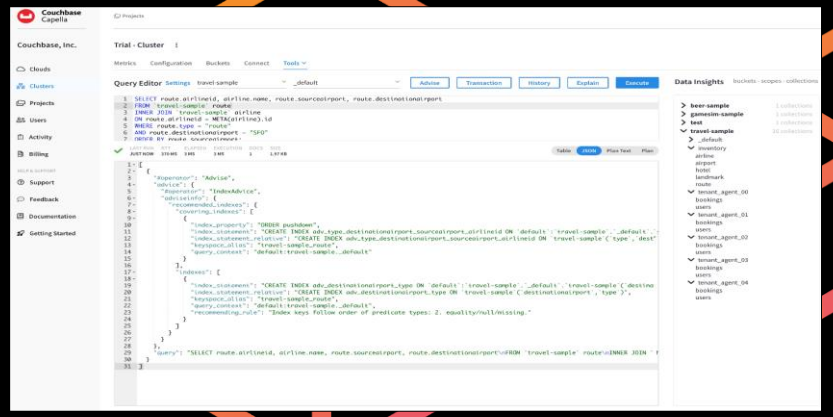
Demo: Refactoring

Next Steps

- Couchbase Playground: 30 min session
  - https://couchbase.live/
  - Examples and learning path
  - Connect to the Capella trial



- Couchbase Capella Trial: 30 days
  - https://cloud.couchbase.com/sign-up
  - Self service tutorial
  - Sample data / import data



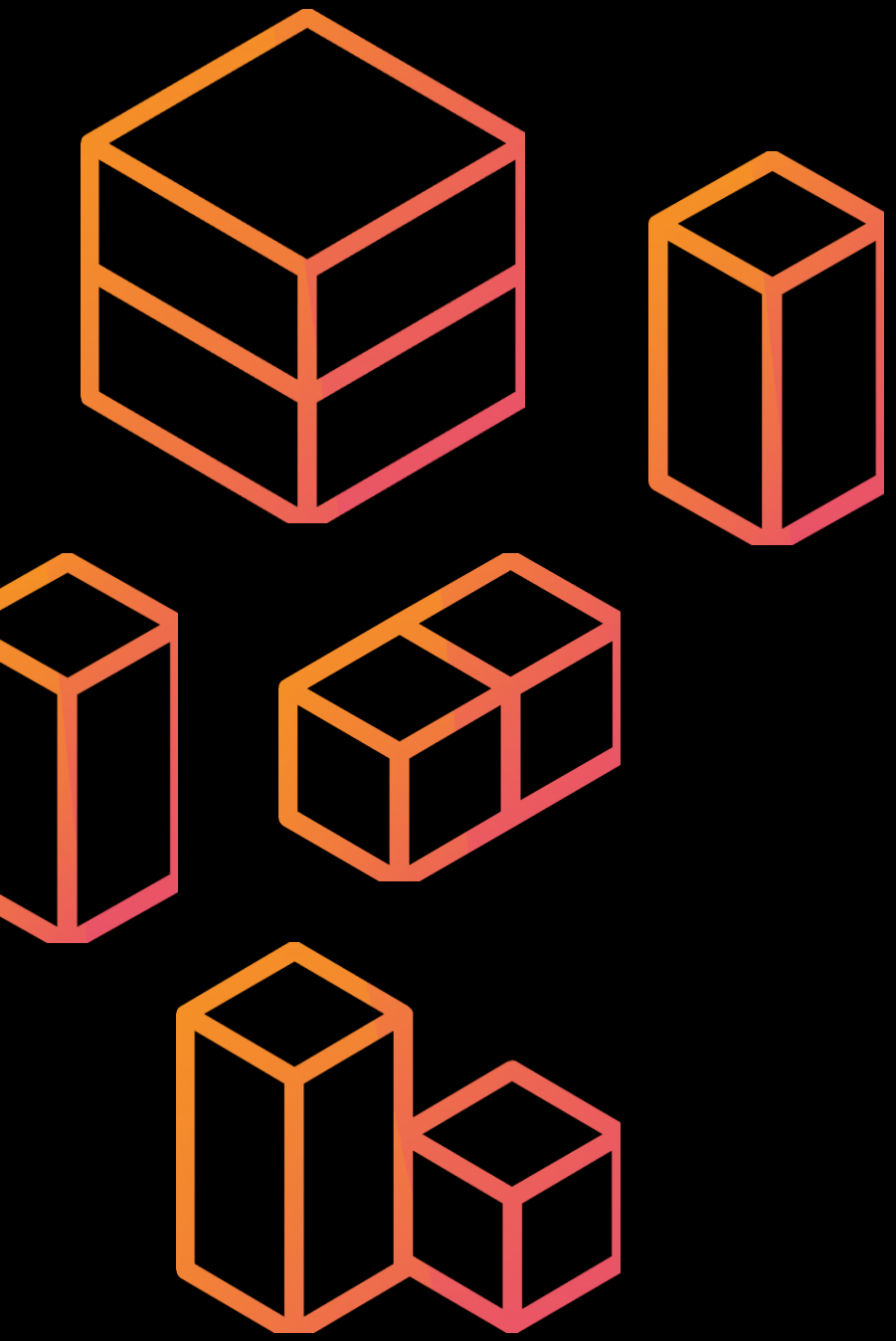
- Post trial options
  - Sign up with pay-as-you-go model
  - Starter Kits: Discounted credits & professional services

CONTACT

Matthew Groves

https://github.com/mgroves/SqlServerToCouchbase

Web	couchbase.com
Twitter	twitter.com/couchbase
Facebook	facebook.com/couchbase



THANK YOU

Couchbase