Learning Azure Cosmos DB

INTRODUCTION AND OVERVIEW

Leonard Lobel
CTO, SLEEK TECHNOLOGIES

lennilobel.wordpress.com

What is NoSQL?

3Vs

- Volume
- Variety
- Velocity

What is a NoSQL database?

Distributed

Replicas ensure high throughput/availability, and low latency

Scale-out

Horizontal partitioning enables virtually limitless storage and throughput

Schema-free

Document, table, graph, and columnar data models

What is Cosmos DB?

Evolution of DocumentDB

Scalable NoSQL document database
Low latency (single-digit millisecond)

Virtually unlimited scale

Scale storage with server-side partitioning
Scale throughput with variable request units

Turnkey global distribution

Point-and-click control over where
your data gets geo-replicated

Multi-model / Multi-API

No longer exclusively a document database
Also supports tables, graph, and columnar

Brief Cosmos DB History

2010 - 2014

Office, OneNote, Xbox

Internal Microsoft DocumentDB service

Public preview

2015

Azure DocumentDB

General Availability (GA)

2017 - present

Azure Cosmos DB

Global distribution
Horizontal partitioning
99.999% SLAs
Multi-model / Multi-API

Getting Started

30-day Free Trial

http://azure.microsoft.com/try/cosmosdb

Microsoft Account
http://signup.live.com

Azure Portal

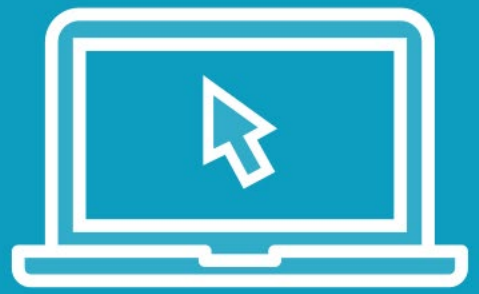
http://portal.azure.com

Azure Subscription
http://azure.microsoft.com

Local Emulator

http://aka.ms/cosmosdb-emulator

Demo



Creating a Cosmos DB account

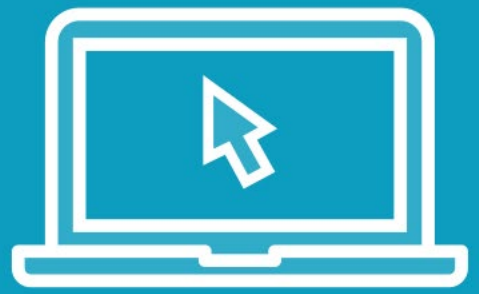
Introducing the Local Emulator

- Emulate Cosmos DB in a local development environment
  - Supports identical functionality as Azure Cosmos DB in the cloud

- No need for:
  - Azure subscription
  - Cosmos DB account
  - Internet connection

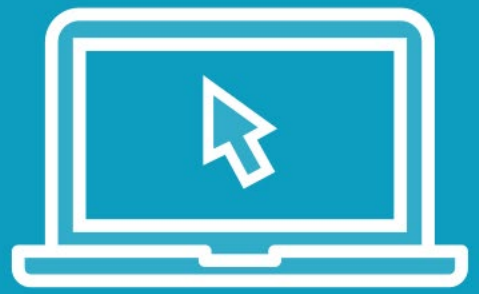
- Develop and test locally
  - Incur no costs
  - Deploy to the cloud when ready

Demo



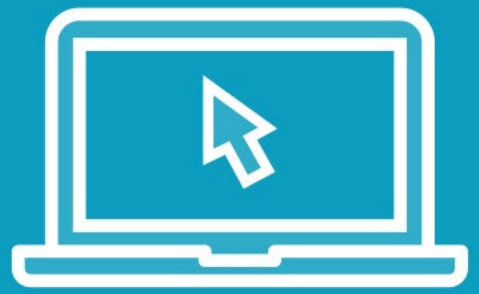
Installing the Cosmos DB local emulator

Demo



Creating a container

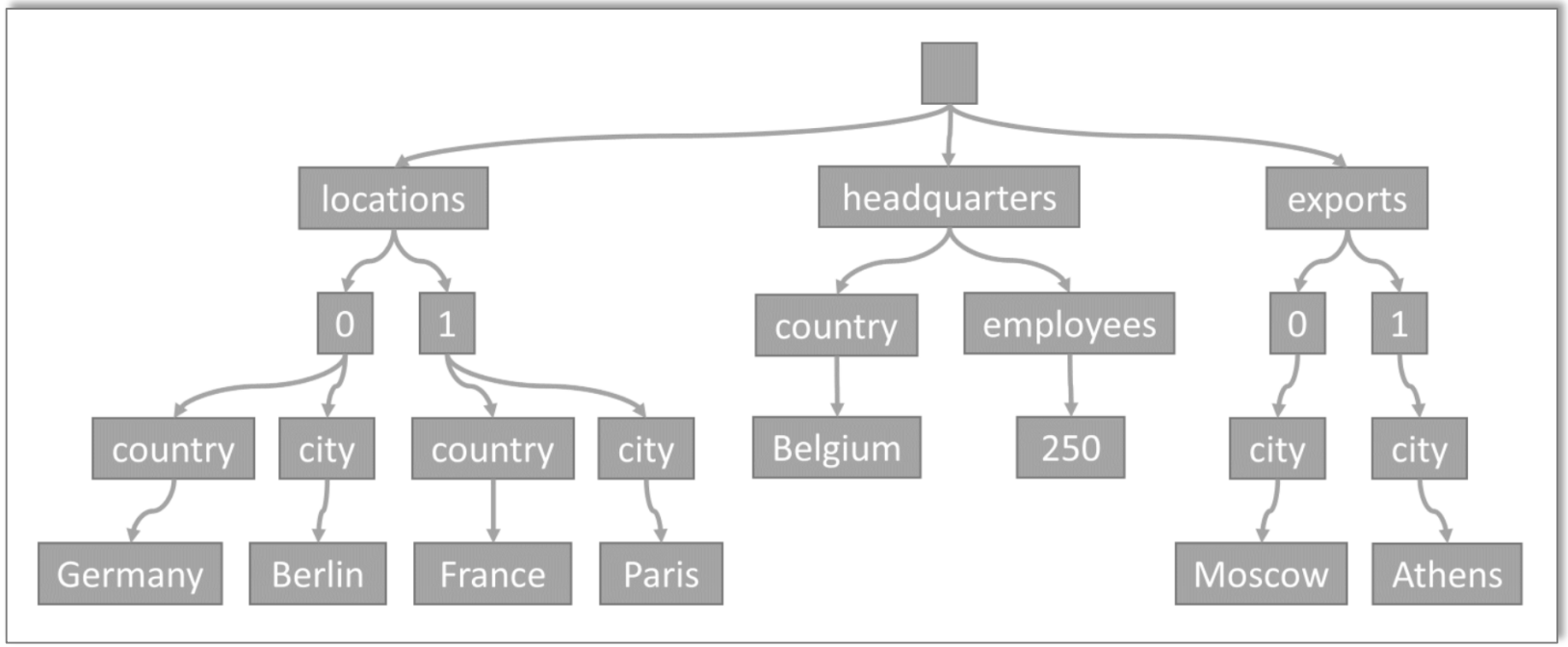
Demo



Creating documents

```
{
    "locations": [
        { "country": "Germany", "city": "Berlin" },
        { "country": "France", "city": "Paris" }
    ],
    "headquarters": { "country": "Belgium", "employees": 250 },
    "exports": [
        { "city": "Moscow" },
        { "city": "Athens" }
    ]
}
```

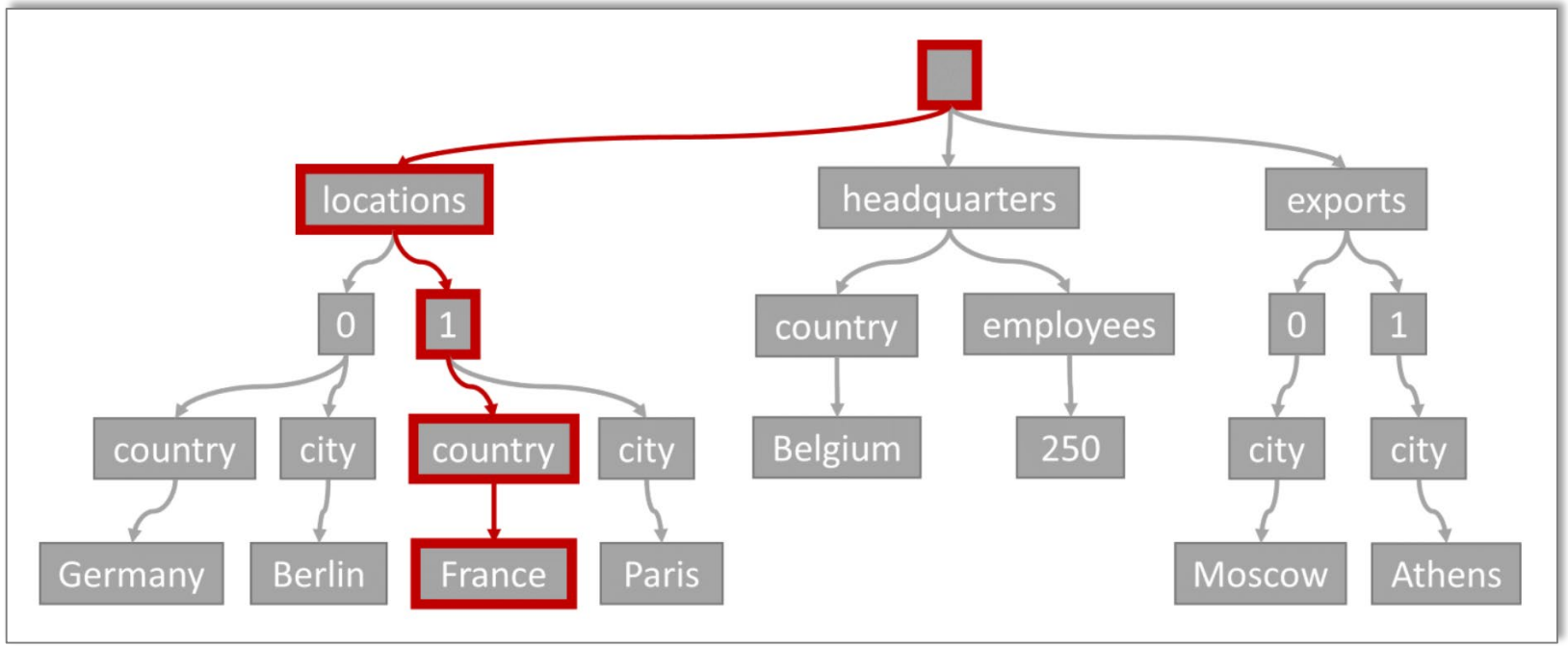
```
SELECT *
FROM l IN c.locations
WHERE l.country = 'France'
```




```
/locations/0/country: "Germany"
/locations/0/city: "Berlin"
/locations/1/country: "France"
/locations/1/city: "Paris"
/headquarters/country: "Belgium"
/headquarters/employees: 250
/exports/0/city: "Moscow"
/exports/1/city: "Athens"
```

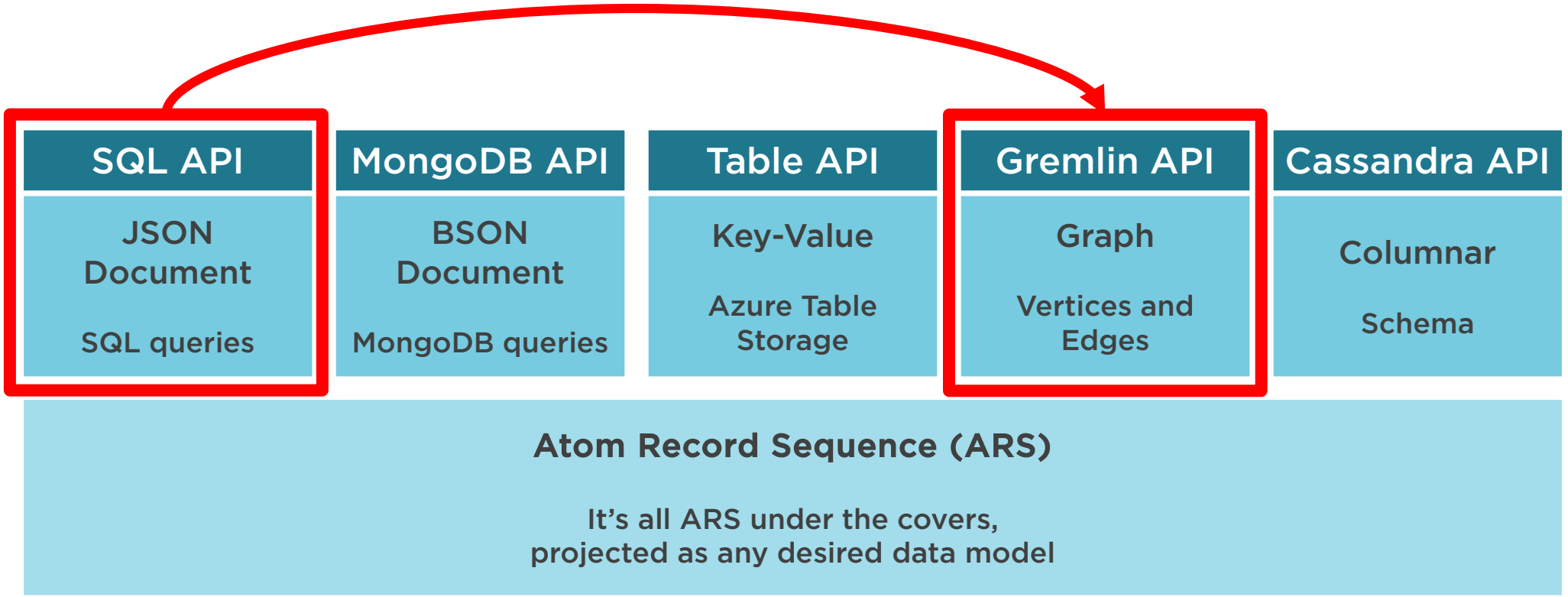
```
{
    "locations": [
        { "country": "Germany", "city": "Berlin" },
        { "country": "France", "city": "Paris" }
    ],
    "headquarters": { "country": "Belgium", "employees": 250 },
    "exports": [
        { "city": "Moscow" },
        { "city": "Athens" }
    ]
}
```

```
SELECT *
FROM l IN c.locations
WHERE l.country = 'France'
```




```
/locations/0/country: "Germany"
/locations/0/city: "Berlin"
/locations/1/country: "France"
/locations/1/city: "Paris"
/headquarters/country: "Belgium"
/headquarters/employees: 250
/exports/0/city: "Moscow"
/exports/1/city: "Athens"
```

Multiple APIs and Data Models



Automatic Indexing

Course Structure

Concepts

- Throughput
- Partitioning
- Global distribution

How-to

- SQL API
- Table API
- Gremlin API

Summary

- Introduction to NoSQL
- Introduction to Cosmos DB
- Getting started
- Local emulator
- Creating a container
- Creating and querying documents
- Multi-model APIs
- Automatic indexing
- Course structure