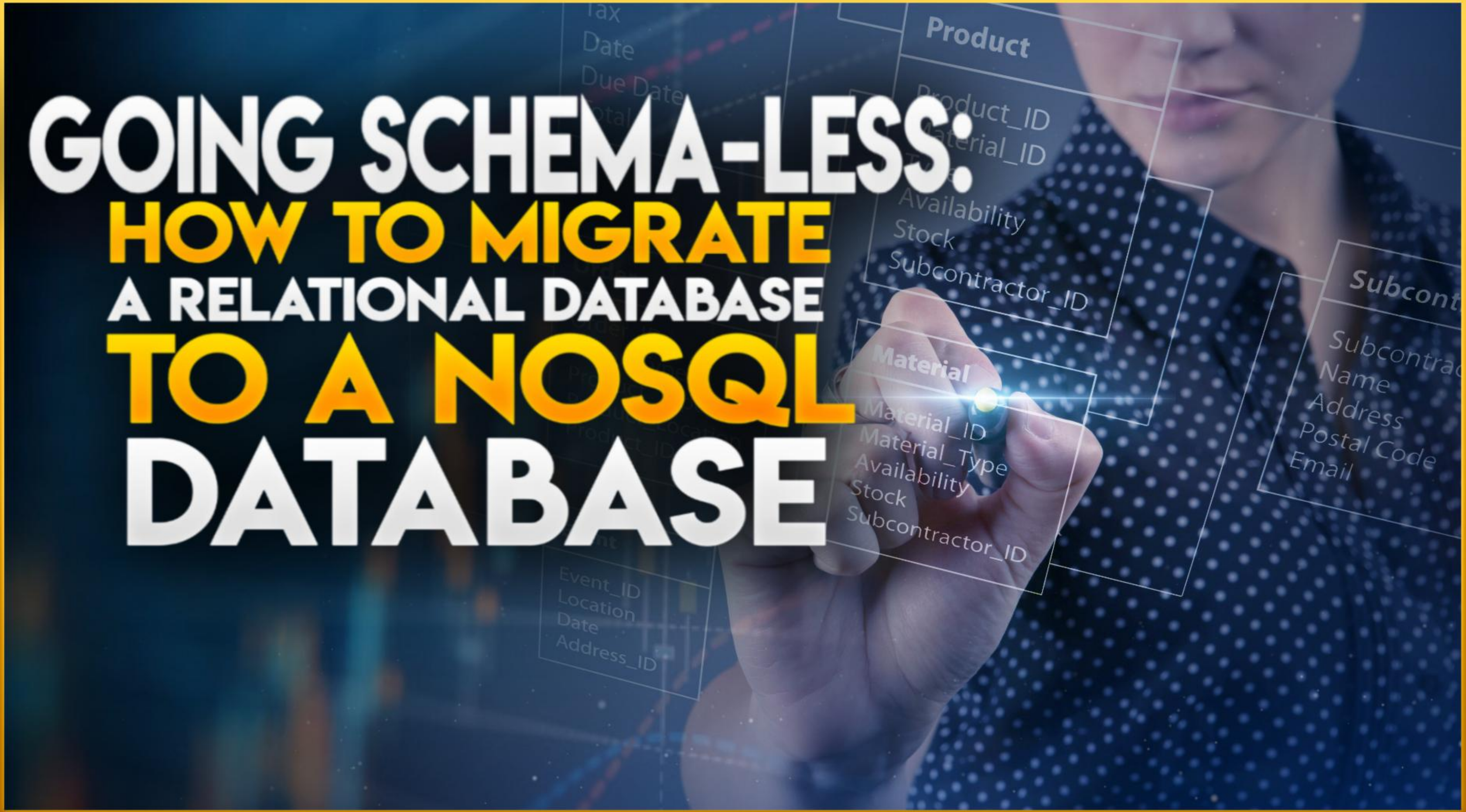
GOING SCHEMA-LESS: HOW TO MIGRATE A RELATIONAL DATABASE TO A NOSQL DATABASE



Who is Chad Green

chadgreen@chadgreen.com

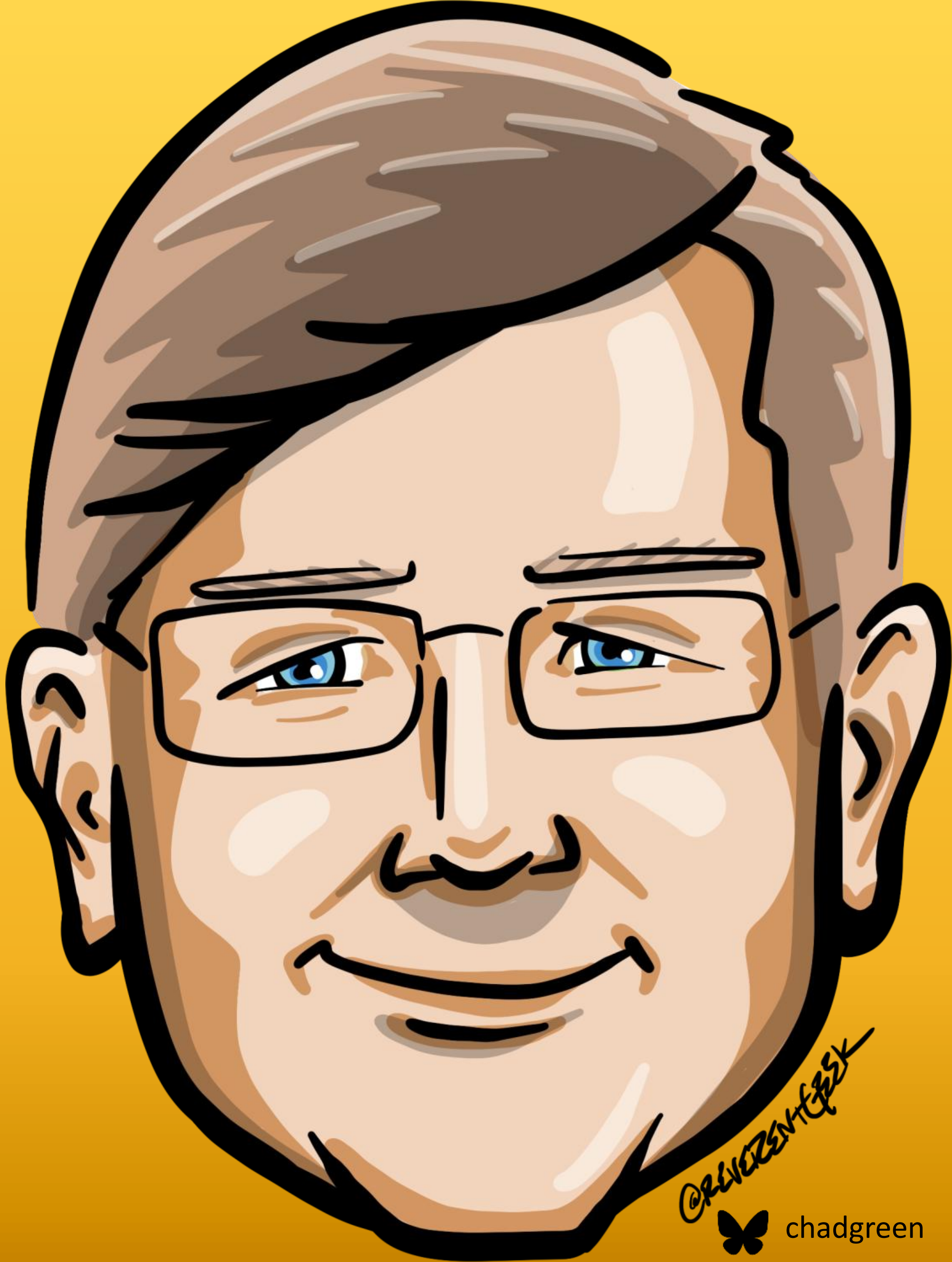
TaleLearnCode

ChadGreen.com

ChadGreen.bsky.social

ChadwickEGreen

Microsoft® MVP Most Valuable Professional



chadgreen

Who is experienced with relational databases?

Who has used NoSQL databases?

How did I get started with NoSQL databases?

chadgreen

What are Relational Databases

chadgreen

Relational Model

- First-order predicate logic
- Described by Edgar Codd in 1969
- Data represented in terms of tuples
- Purpose is to provide declarative method for specifying data and queries

Codd's 12 Rules

0: Foundation Rule

1: Information Rule

2: Guaranteed Access

3: Systematic treatment of NULL values

4: Active Online Catalog

5: Comprehensive data sublanguage

6: View Updating

7: Possible for high-level insert, update, and delete

8: Physical data independence

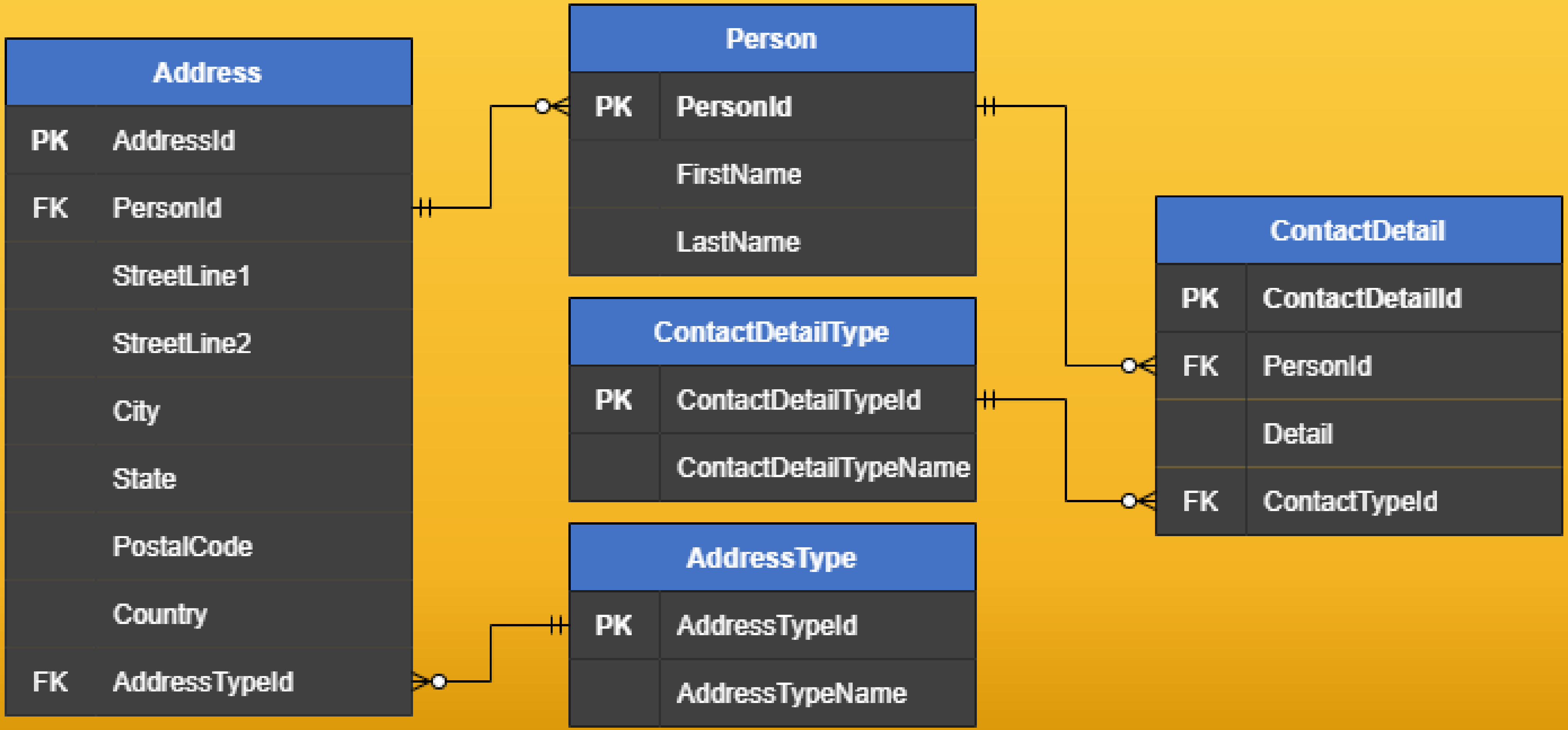
9: Logic data independence

10: Integrity Independence

11: Distribution Independence

12: Nonsubversion Rule

Typical Relational Model



chadgreen

True star of Relational Databases

SQL

Structured Query Language

SEQUEL

True star of Relational Databases

SC

Structured

By Saufhn - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=87255205

chadgreen

Big Names in Relational Databases

ORACLE®

MySQL™

Microsoft® SQL Server®

PostgreSQL

IBM DB2

SQLite

What are NoSQL Databases

chadgreen

What are NoSQL Databases

Modeled in means other than tabular relations

Existed since late 1960s

Increasingly used in big data and real-time web applications

NoSQL Motivations

Simplicity of Design

Simpler Horizontal Scaling

Finer Control over Availability

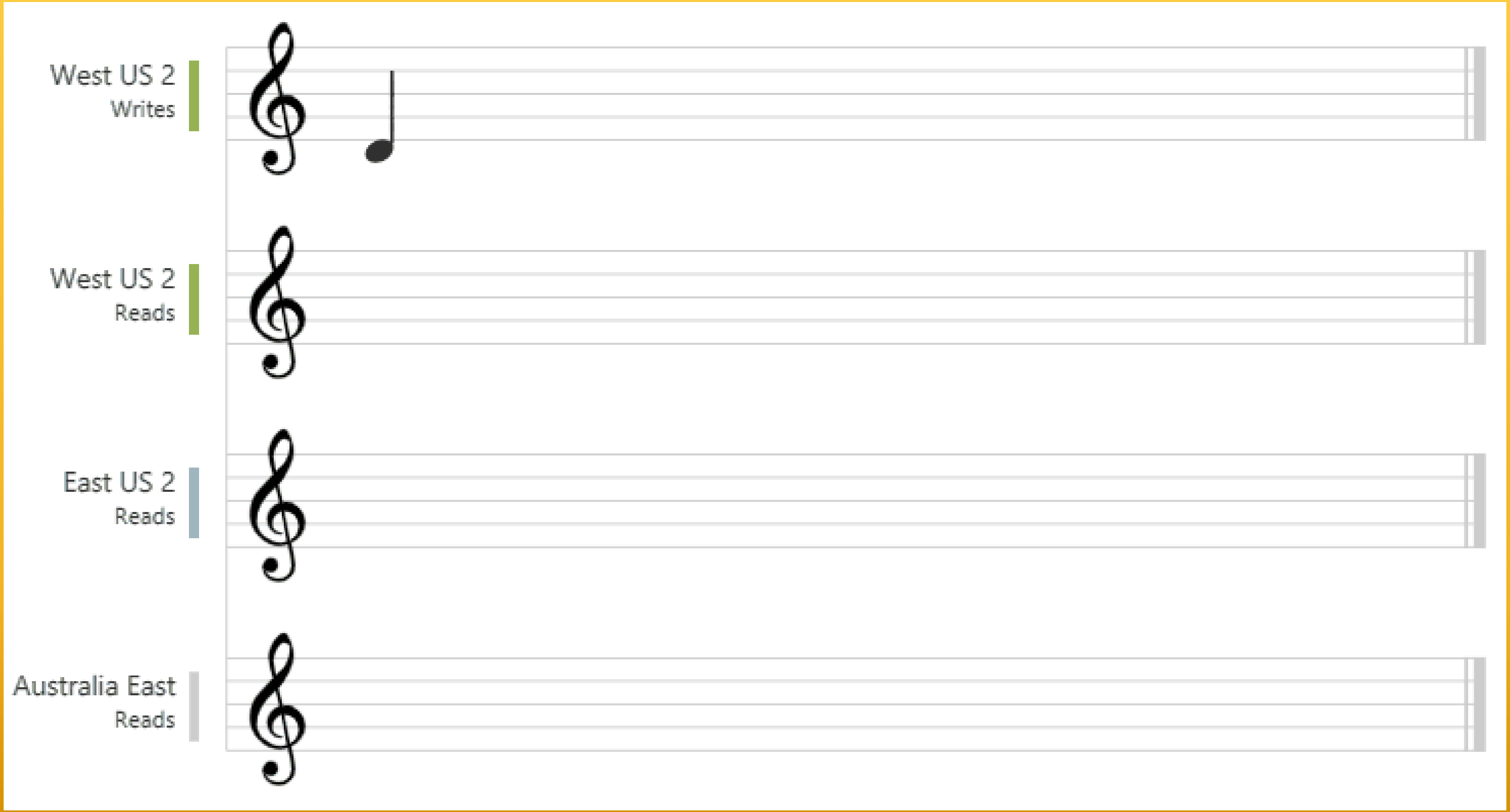
Limiting Object-Relational Impedance

Availability over Consistency

Relational
ACID Transactions

NoSQL
Eventual Consistency

Eventual Consistency



chadgreen

What's in a Name

NoSQL

What’s in a Name

NoSQL

What’s in a Name

Not only SQL

Non-SQL

Non-Relational

NoSQL

What’s in a Name

Not only SQL

Non-SQL

Non-Relational

NoSQL

What's in a Name

No-Schema

Not only SQL

Non-SQL

Non-Relational

NoSQL

Many types of NoSQL databases

mongoDB

Couchbase

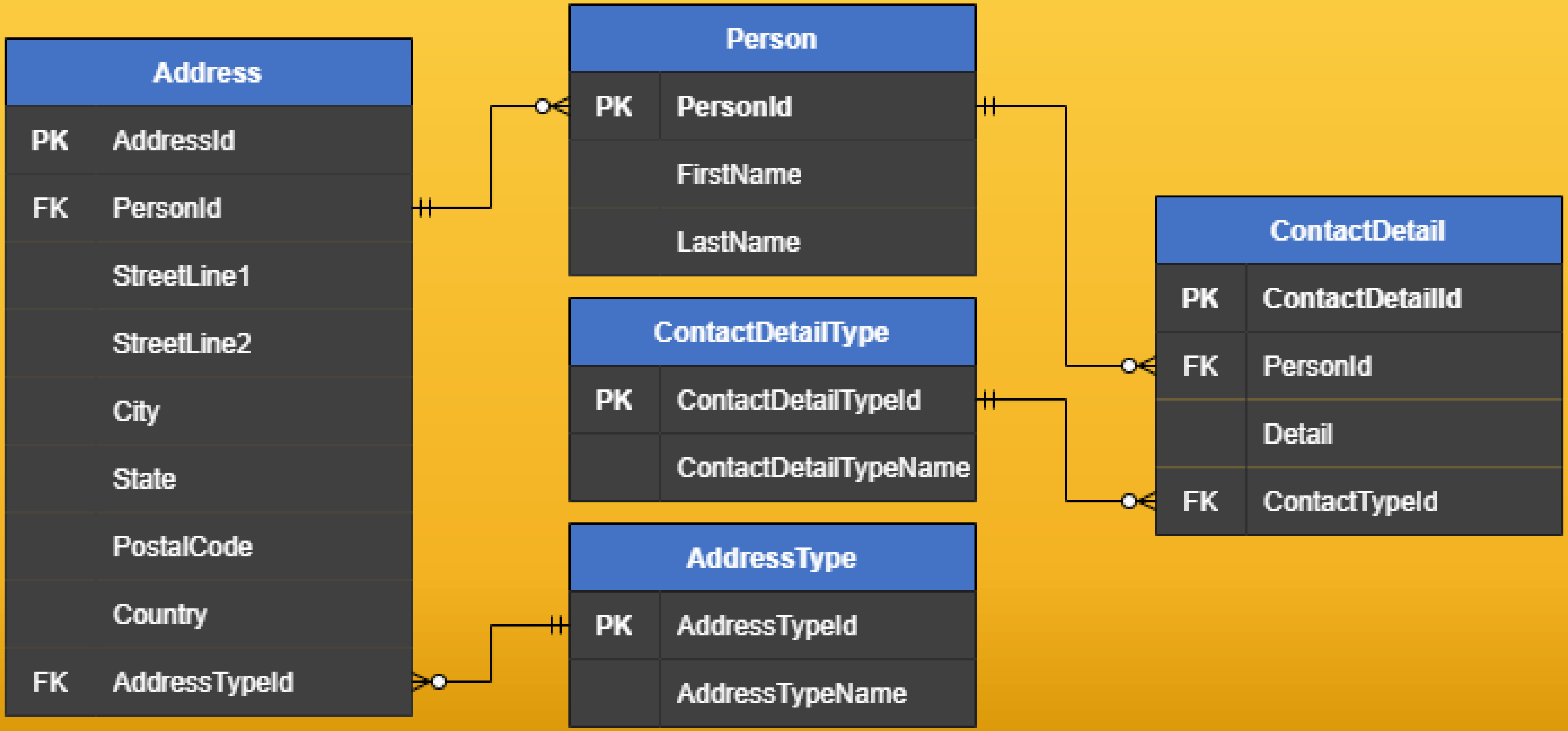
Document

elastic

CouchDB

ArangoDB

Typical Relational Model



chadgreen

Same but in a document database

```
{
  "id": "1",
  "firstName": "Thomas",
  "lastName": "Andersen",
  "addresses": [
    {
      "city": "Seattle",
      "state": "WA",
      "type": {
        "name": "Primary"
      }
    }
  ],
  "contactDetails": [
    {
      "detail": "First Detail",
      "type": {
        "name": "A detail type"
      }
    }
  ]
}
```

chadgreen

Many types of NoSQL databases

ArangoDB

Key-Value

Amazon DynamoDB

apache Ignite™

redis

Many types of NoSQL databases

Wide Column

APACHE HBASE

SCYLLA

cassandra

accumulo™

Many types of NoSQL databases

ArangoDB

Graph

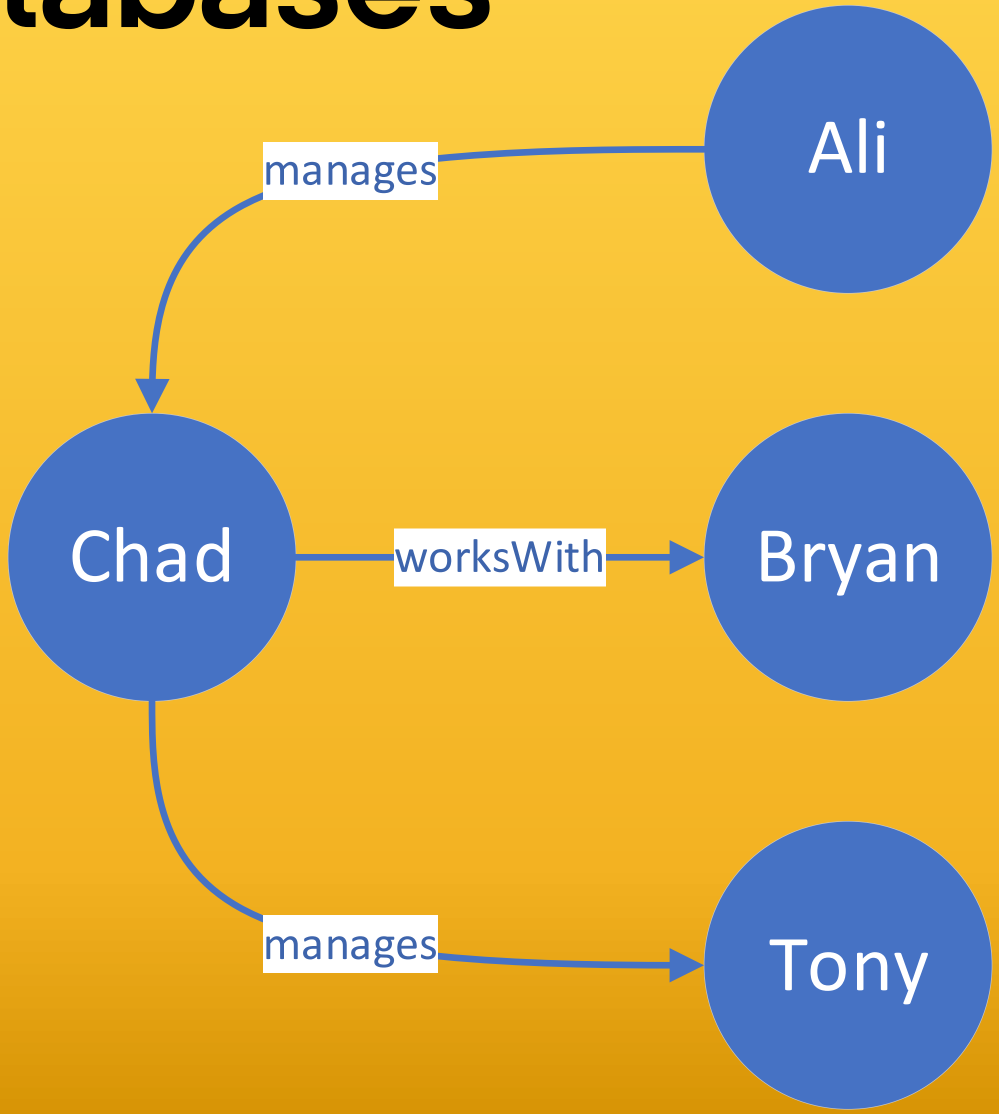
Amazon Neptune

neo4j

Apache TinkerPop

chadgreen

Graph Databases



Many types of NoSQL databases

- Document
- Key-Value
- Wide Column
- Graph
- Object
- Tabular
- Tuple Store
- Triple Store

chadgreen

Picking a Data Store

chadgreen

Data Model Comparison

Data Model	Performance	Scalability	Flexibility	Complexity	Functionality
Key-Value Store	High	High	High	None	Variable (None)
Column Store	High	High	Moderate	Low	Minimal
Document Store	High	Variable (High)	High	Low	Variable (Low)
Graph	Variable	Variable	High	High	Graph Theory
Relational	Variable	Variable	Low	Moderate	Relational Algebra

Ben Scofield – NoSQL presentation at CodeMash 2010

Things to think about

Skillset

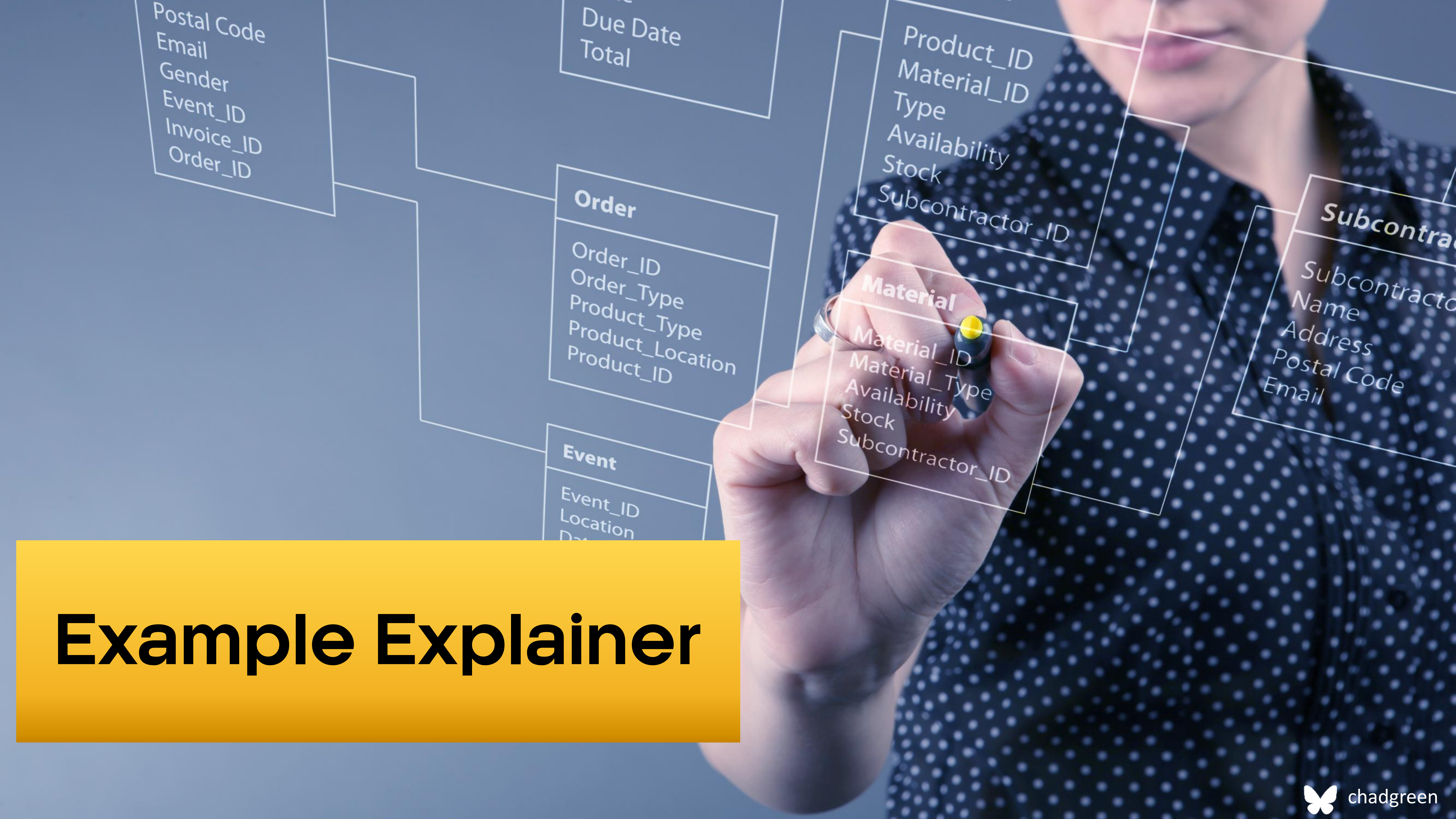
Time to Market

Known Data Structure

Scalability

Don’t forget

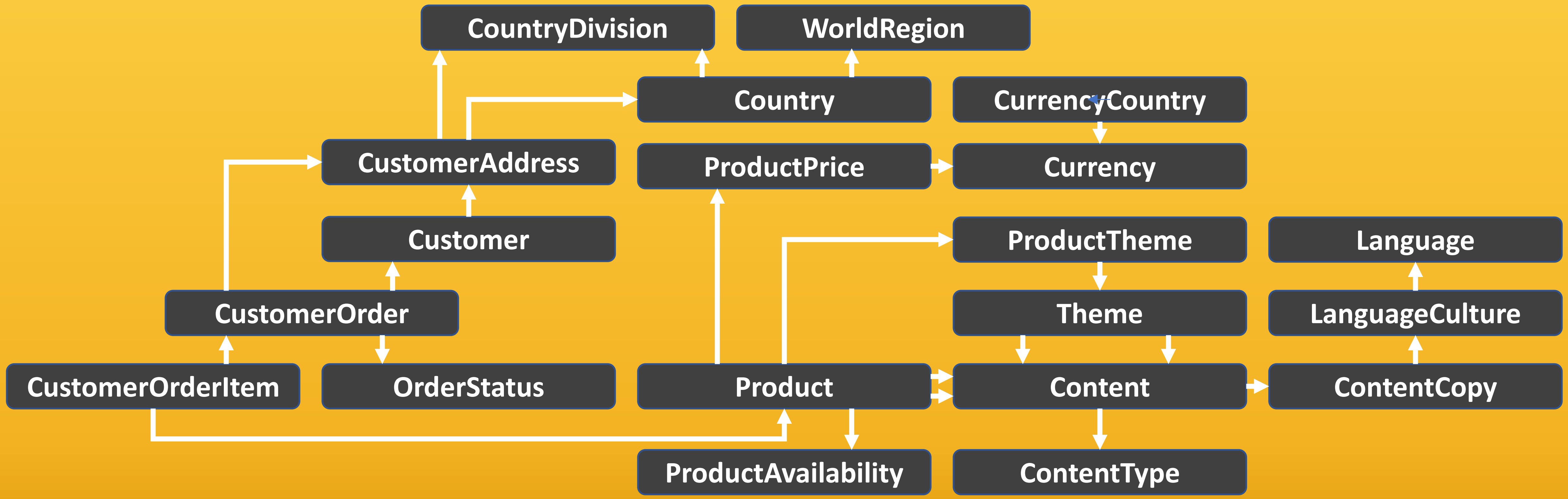
Hybrid



Example Explainer

chadgreen

Data Model



chadgreen

Very Quick Into to Cosmos DB

chadgreen

Azure Cosmos DB

etcd

Table API

cassandra

Core (SQL) API

Apache Spark

MongoDB

Gremlin

Key-value

Column-family

Document

Graph

Guaranteed low latency at the 99th percentile

Elastic scale out of storage & throughput

Five well-defined consistency models

Turnkey global distribution

Comprehensive SLAs

Which Azure Cosmos DB Data API?

Core
(SQL)
API

Core (SQL) API

chadgreen

Which Azure Cosmos DB Data API?

Core
(SQL)
API

mongoDB

MongoDB

chadgreen

Which Azure Cosmos DB Data API?

Core (SQL) API

mongoDB

Table Storage

chadgreen

Which Azure Cosmos DB Data API?

Core (SQL) API

Gremlin

chadgreen

Which Azure Cosmos DB Data API?

Core (SQL) API

mongoDB

Gremlin $G = (V, E)$

cassandra

Cassandra

chadgreen

Migrating to NoSQL

chadgreen

Database Considerations

- Data Model/API

Database Considerations

- Data Model/API
- Document Structure

Database Considerations

- Data Model/API
- Document Structure
- Partition Key

- Access Patterns
- Even Data Distributions
- Cardinality
- Query Isolation
- Write Patterns
- Data Growth
- Familiarity with Data
- Data Relationship
- Cost Considerations
- Immutable Properties
- Data Size
- Trial and Error

Database Considerations

- Data Model/API
- Document Structure
- Partition Key
- Indexing

Database Considerations

- Data Model/API
- Document Structure
- Partition Key
- Indexing
- Query Performance

Database Considerations

- Data Model/API
- Document Structure
- Partition Key
- Indexing
- Query Performance
- Consistency Level

Database Considerations

- Data Model/API
- Document Structure
- Partition Key
- Indexing
- Query Performance
- Consistency Level
- Time-to-Live (TTL)

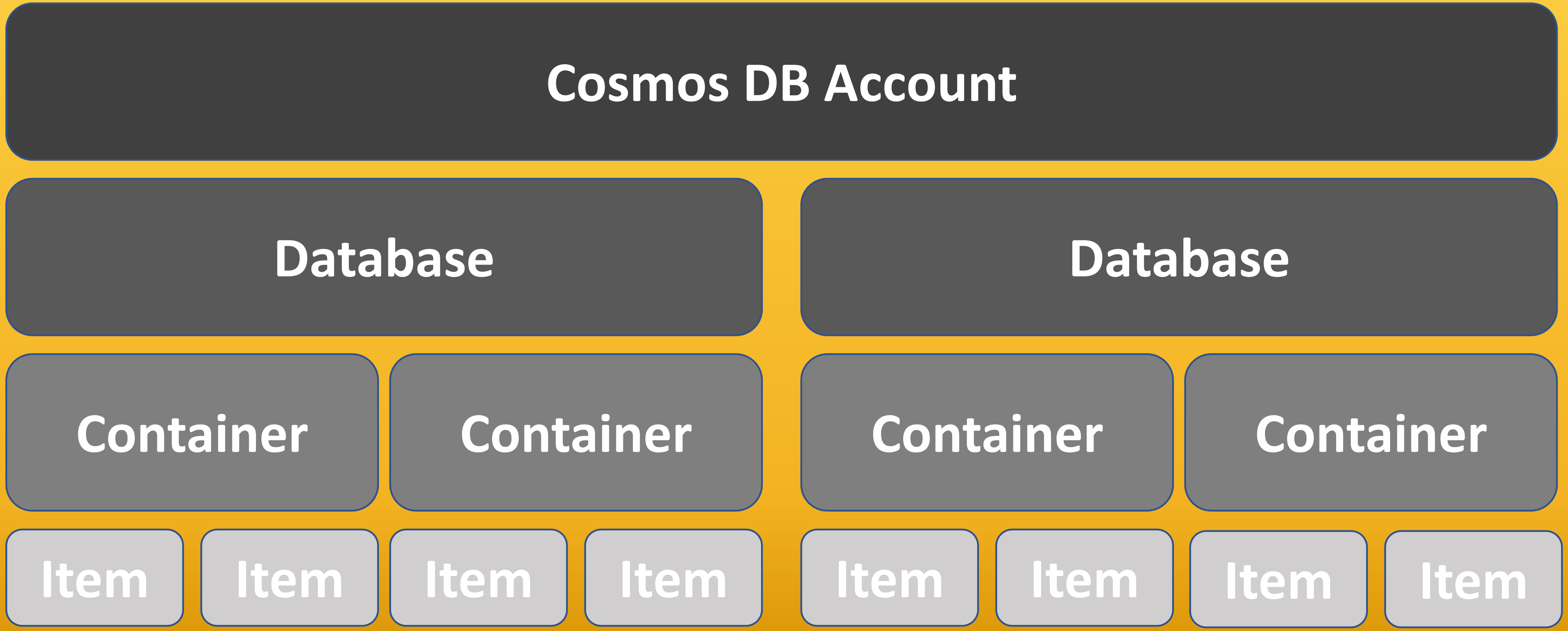
Database Considerations

- Data Model/API
- Document Structure
- Partition Key
- Indexing
- Query Performance
- Consistency Level
- Time-to-Live (TTL)
- Data Migration

Database Considerations

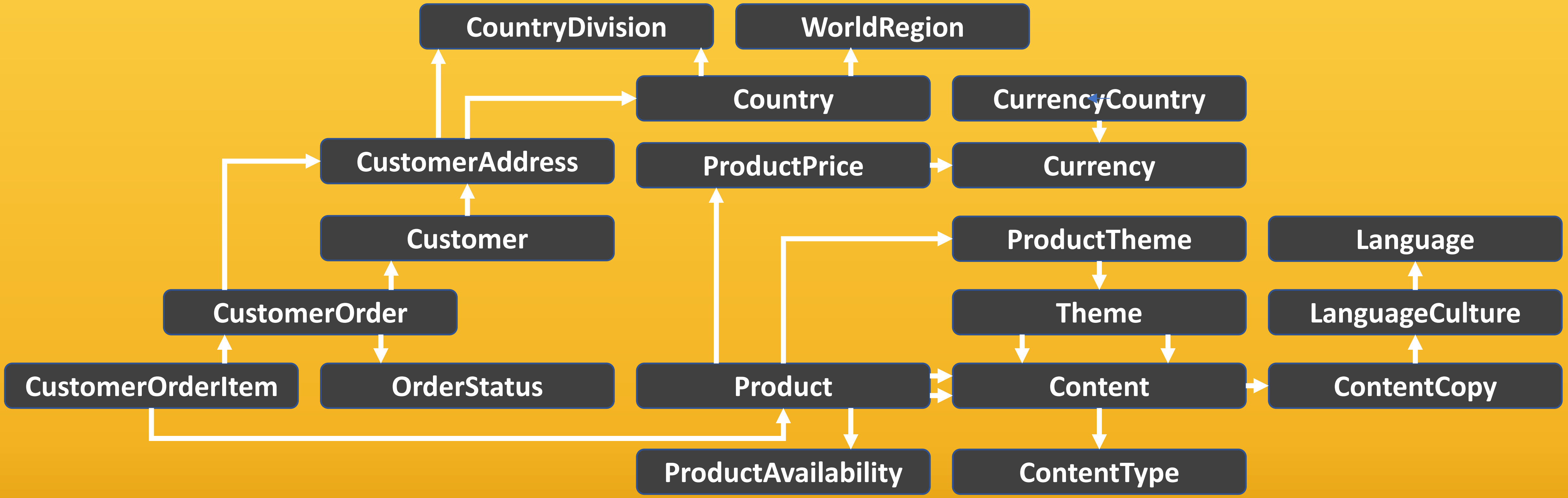
- Data Model/API
- Document Structure
- Partition Key
- Indexing
- Query Performance
- Consistency Level
- Time-to-Live (TTL)
- Data Migration
- Versioning and Evolution

Document Database Structure



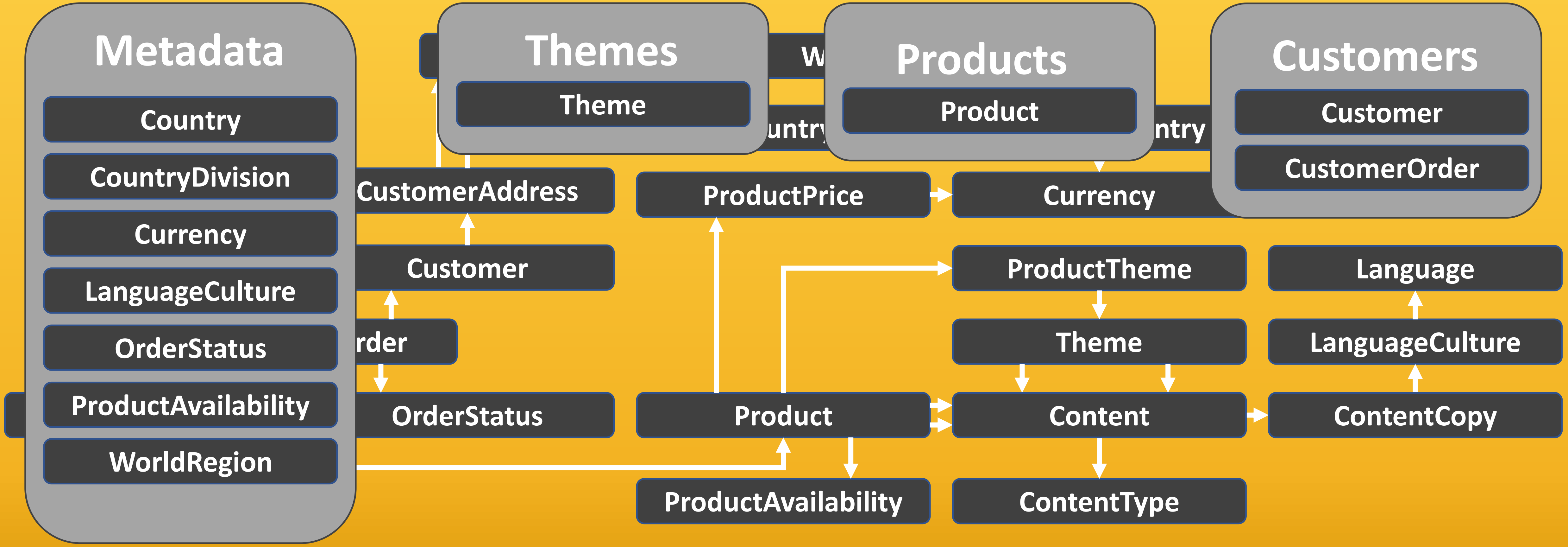
chadgreen

Data Model



chadgreen

Data Model



chadgreen

Data Model

Metadata

- Country
- CountryDivision
- Currency
- LanguageCulture
- OrderStatus
- ProductAvailability
- WorldRegion

Themes

- Theme

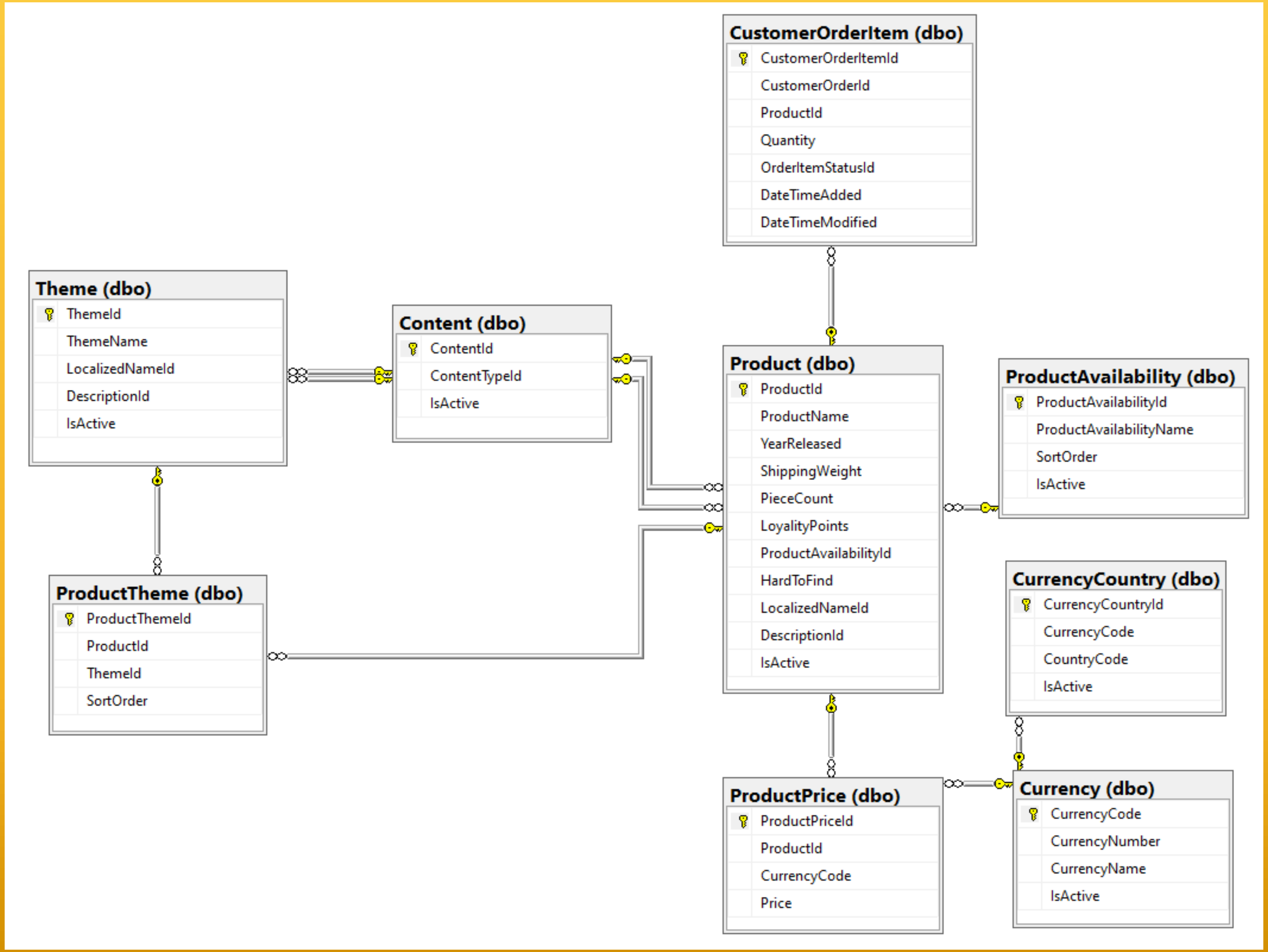
Products

- Product

Customers

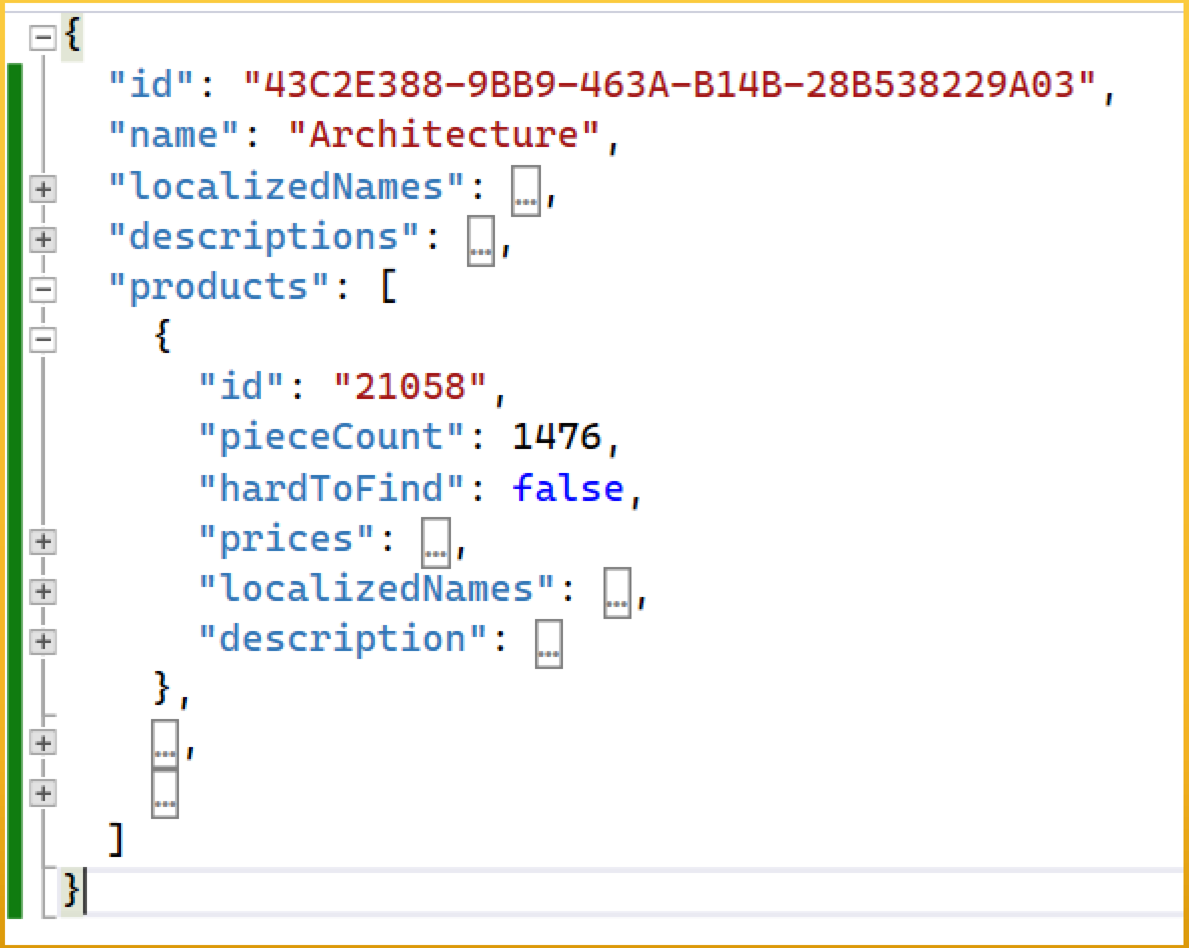
- Customer
- CustomerOrder

Data Model - Theme



chadgreen

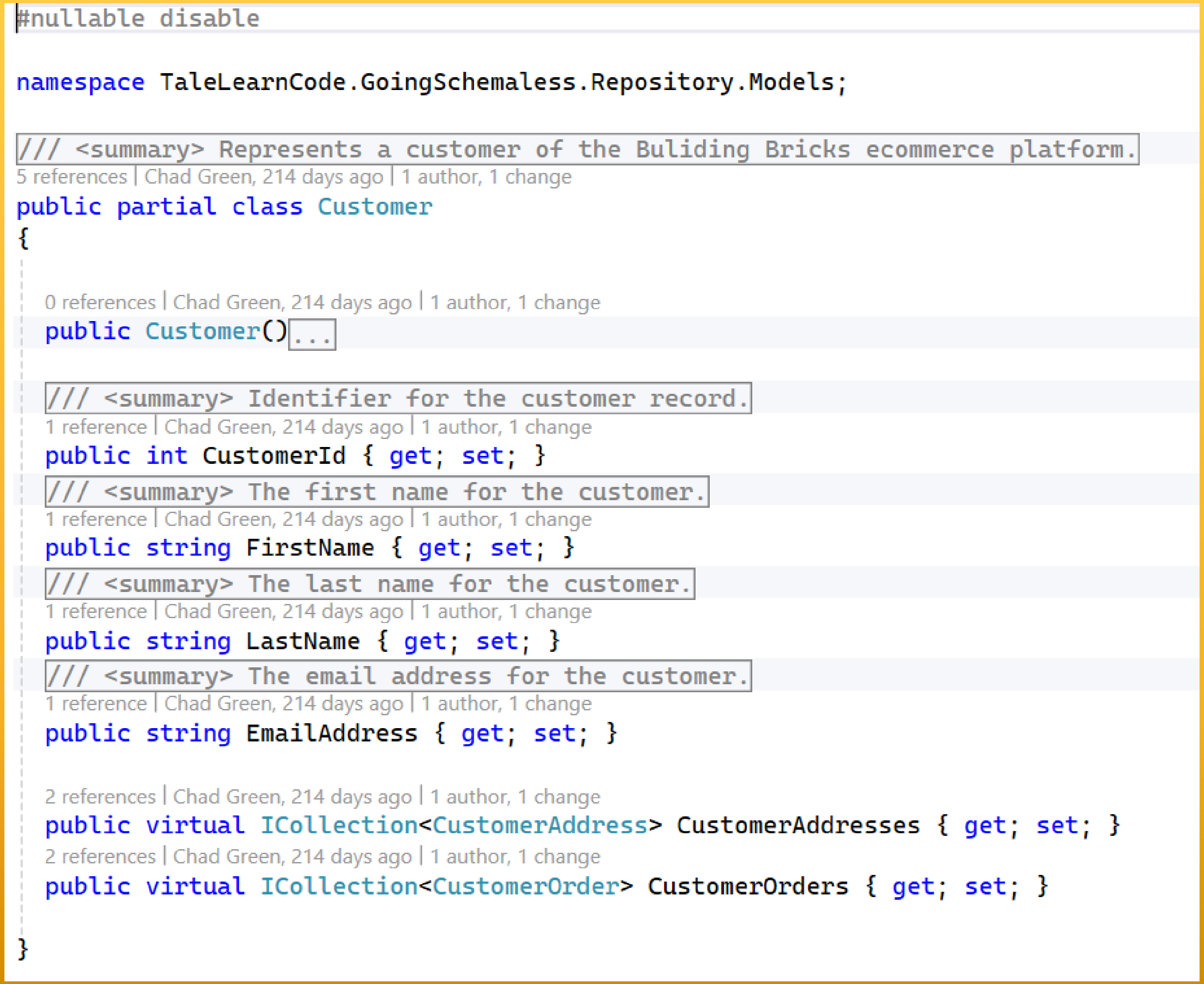
Data Model – Theme



```
{
  "id": "43C2E388-9BB9-463A-B14B-28B538229A03",
  "name": "Architecture",
  "localizedNames": ...,
  "descriptions": ...,
  "products": [
    {
      "id": "21058",
      "pieceCount": 1476,
      "hardToFind": false,
      "prices": ...,
      "localizedNames": ...,
      "description": ...
    },
    ...,
    ...
  ]
}
```

chadgreen

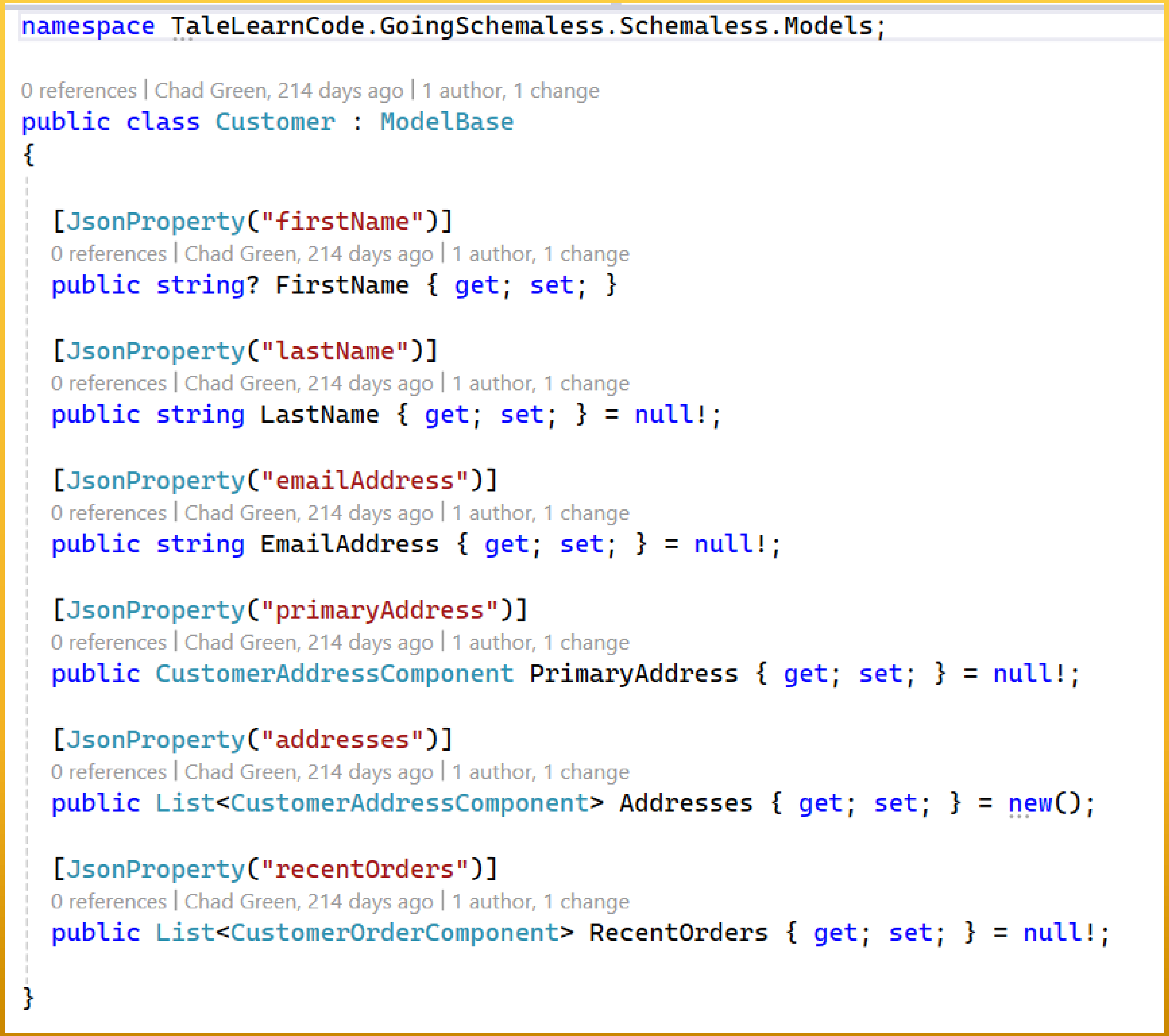
Relational Model



```
#nullable disable

namespace TaleLearnCode.GoingSchemaless.Repository.Models;

/// <summary> Represents a customer of the Buliding Bricks ecommerce platform.
5 references | Chad Green, 214 days ago | 1 author, 1 change
public partial class Customer
{

    0 references | Chad Green, 214 days ago | 1 author, 1 change
    public Customer()...

    /// <summary> Identifier for the customer record.
    1 reference | Chad Green, 214 days ago | 1 author, 1 change
    public int CustomerId { get; set; }
    /// <summary> The first name for the customer.
    1 reference | Chad Green, 214 days ago | 1 author, 1 change
    public string FirstName { get; set; }
    /// <summary> The last name for the customer.
    1 reference | Chad Green, 214 days ago | 1 author, 1 change
    public string LastName { get; set; }
    /// <summary> The email address for the customer.
    1 reference | Chad Green, 214 days ago | 1 author, 1 change
    public string EmailAddress { get; set; }

    2 references | Chad Green, 214 days ago | 1 author, 1 change
    public virtual ICollection<CustomerAddress> CustomerAddresses { get; set; }
    2 references | Chad Green, 214 days ago | 1 author, 1 change
    public virtual ICollection<CustomerOrder> CustomerOrders { get; set; }

}
```

chadgreen

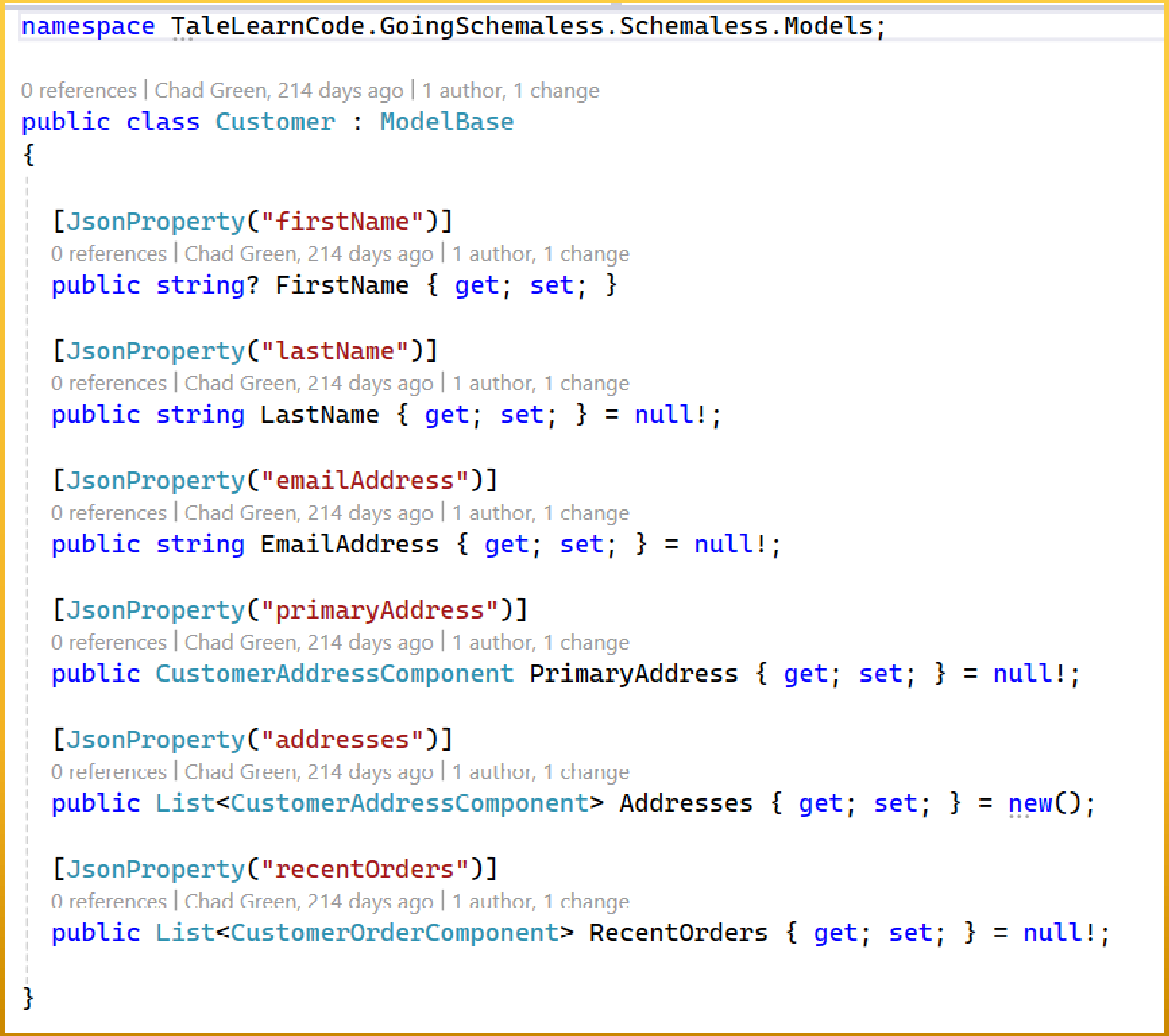
NoSQL Model (Code)



```
namespace TaleLearnCode.GoingSchemaless.Schemaless.Models;

0 references | Chad Green, 214 days ago | 1 author, 1 change
public class Customer : ModelBase
{

  [JsonProperty("firstName")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string? FirstName { get; set; }

  [JsonProperty("lastName")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string LastName { get; set; } = null!;

  [JsonProperty("emailAddress")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string EmailAddress { get; set; } = null!;

  [JsonProperty("primaryAddress")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public CustomerAddressComponent PrimaryAddress { get; set; } = null!;

  [JsonProperty("addresses")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public List<CustomerAddressComponent> Addresses { get; set; } = new();

  [JsonProperty("recentOrders")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public List<CustomerOrderComponent> RecentOrders { get; set; } = null!;

}
```

chadgreen

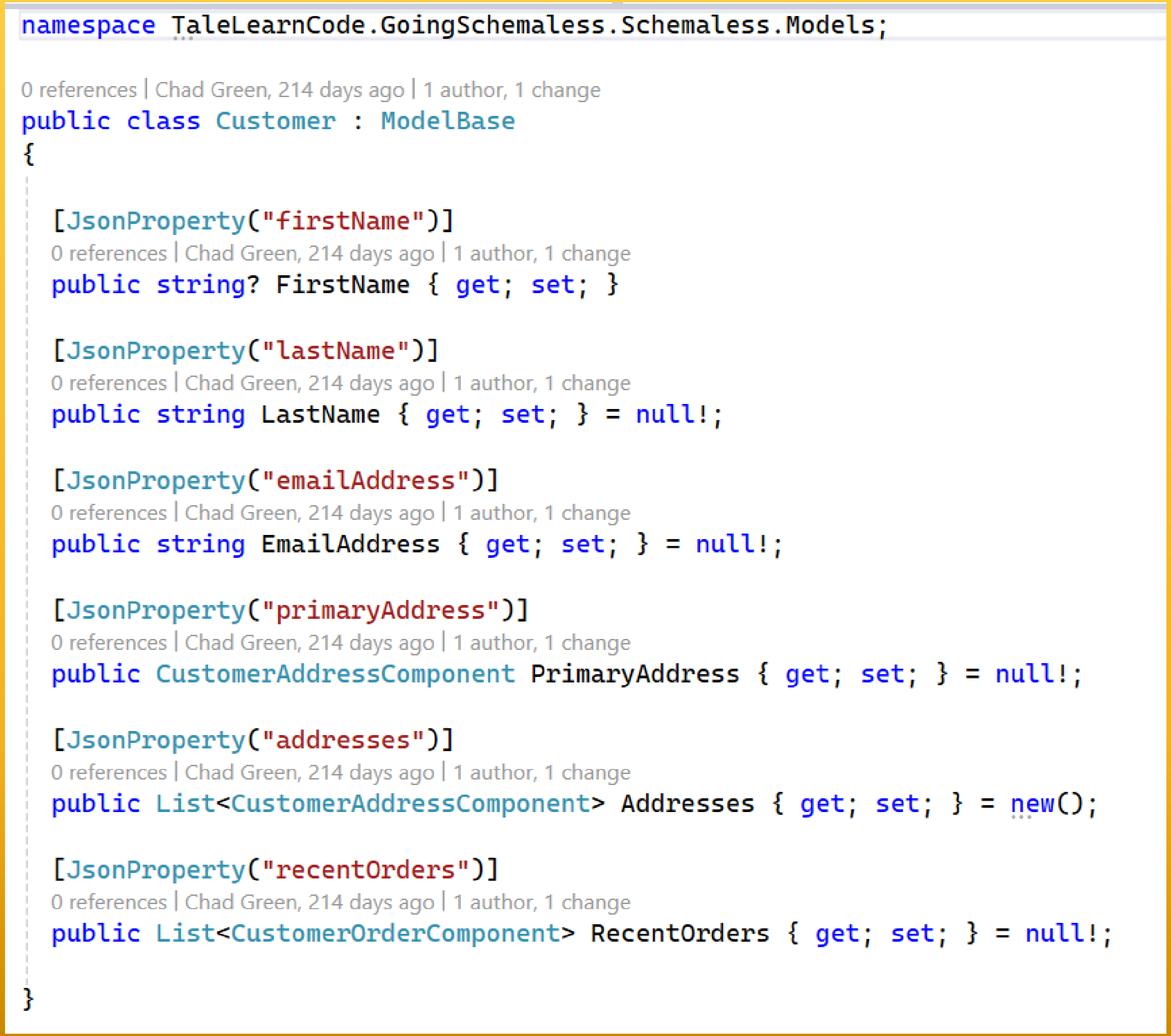
NoSQL Model (Code)



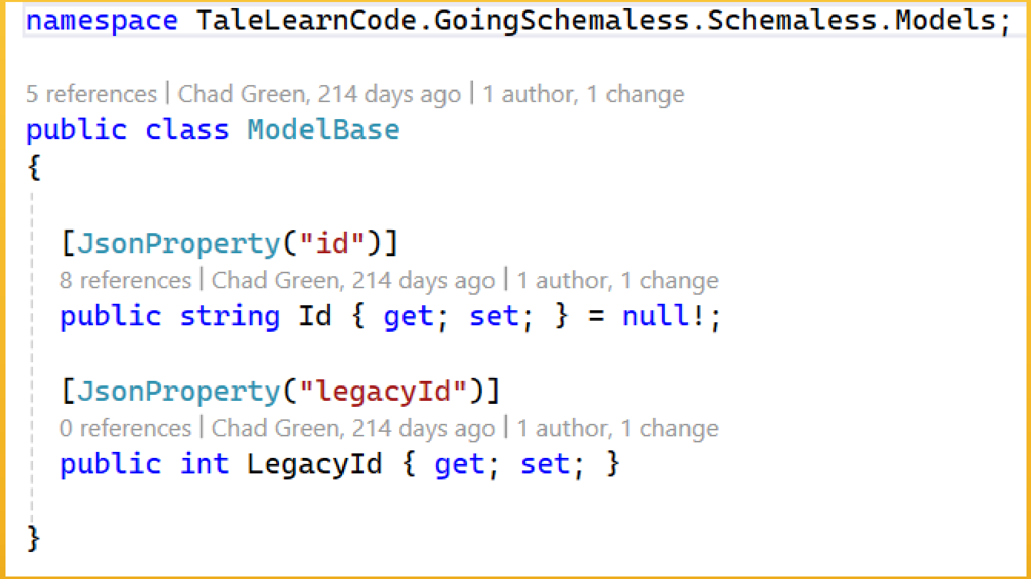
```
namespace TaleLearnCode.GoingSchemaless.Schemaless.Models;

0 references | Chad Green, 214 days ago | 1 author, 1 change
public class Customer : ModelBase
{

  [JsonProperty("firstName")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string? FirstName { get; set; }

  [JsonProperty("lastName")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string LastName { get; set; } = null!;

  [JsonProperty("emailAddress")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string EmailAddress { get; set; } = null!;

  [JsonProperty("primaryAddress")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public CustomerAddressComponent PrimaryAddress { get; set; } = null!;

  [JsonProperty("addresses")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public List<CustomerAddressComponent> Addresses { get; set; } = new();

  [JsonProperty("recentOrders")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public List<CustomerOrderComponent> RecentOrders { get; set; } = null!;

}
```

chadgreen

NoSQL Model (Code)



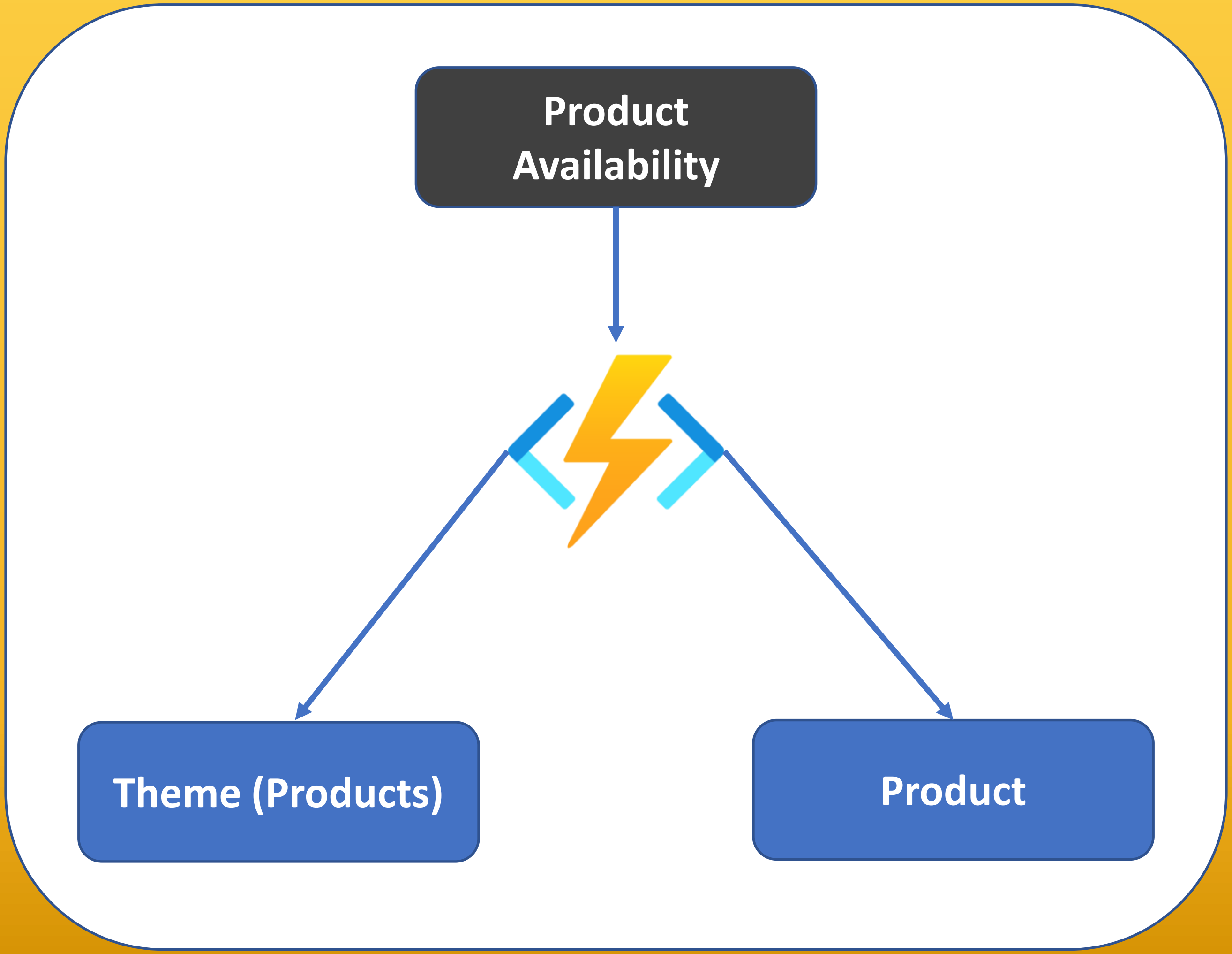
```
namespace TaleLearnCode.GoingSchemaless.Schemaless.Models;

0 references | Chad Green, 214 days ago | 1 author, 1 change
public class Customer : ModelBase
{

  [JsonProperty("firstName")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string? FirstName { get; set; }

  [JsonProperty("lastName")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string LastName { get; set; } = null!;

  [JsonProperty("emailAddress")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public string EmailAddress { get; set; } = null!;

  [JsonProperty("primaryAddress")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public CustomerAddressComponent PrimaryAddress { get; set; } = null!;

  [JsonProperty("addresses")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public List<CustomerAddressComponent> Addresses { get; set; } = new();

  [JsonProperty("recentOrders")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public List<CustomerOrderComponent> RecentOrders { get; set; } = null!;

}
```



```
namespace TaleLearnCode.GoingSchemaless.Schemaless.Models;

5 references | Chad Green, 214 days ago | 1 author, 1 change
public class ModelBase
{

  [JsonProperty("id")]
  8 references | Chad Green, 214 days ago | 1 author, 1 change
  public string Id { get; set; } = null!;

  [JsonProperty("legacyId")]
  0 references | Chad Green, 214 days ago | 1 author, 1 change
  public int LegacyId { get; set; }

}
```

chadgreen

Reference Types



What about data migration?

chadgreen

Best Tool(s) for the Job

Thank You

chadgreen@chadgreen.com

TaleLearnCode

ChadGreen.com

ChadGreen.bsky.social

ChadwickEGreen