
AWS
re:Invent
NOV. 27 – DEC. 1, 2023 | LAS VEGAS, NV

BOA315

Dos and don’ts of NoSQL data modeling

Almas Shaikh
Solutions Architect
AWS

Mohammed Fazalullah Qudrath (Faz)
Sr. Developer Advocate
AWS

Any NoSQL DBAs or enthusiasts?

Agenda

SQL vs. NoSQL

Why NoSQL?

NoSQL database types

NoSQL data modeling best practices and anti-patterns

Takeaways

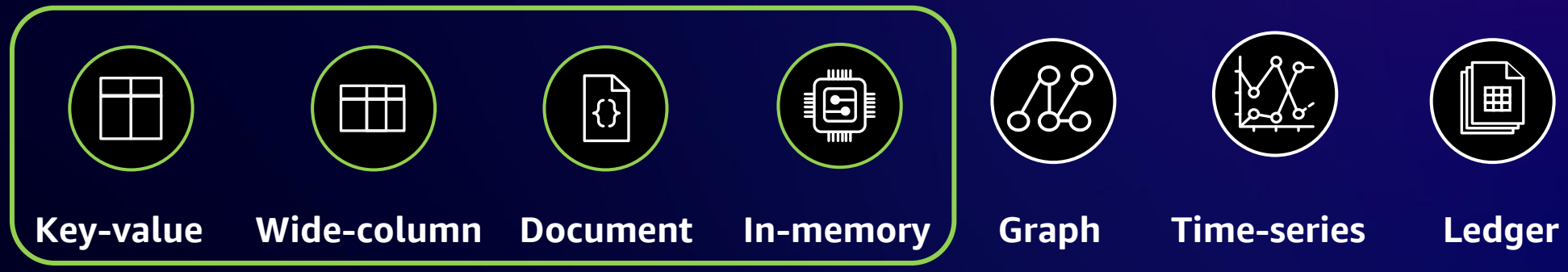
SQL and NoSQL side by side

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

* Most NoSQL databases are built for OLTP workloads

You've moved to NoSQL because . . .

NoSQL database types



Document databases

Designed for **rich JSON** data models

Supports **complex** access patterns, including indexing and querying nested fields, arrays

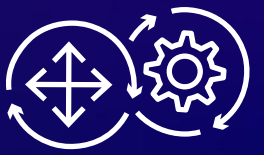
Supports **ad-hoc** queries and aggregations

Key-value databases

Designed for **simple** data models

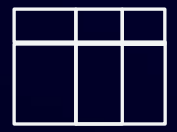
Key **uniquely** identifies a record

Joins, GROUP BY and ORDER BY clauses, and aggregations **not supported**



Provide **extreme** scale

Wide-column databases



Designed for **simple** data models

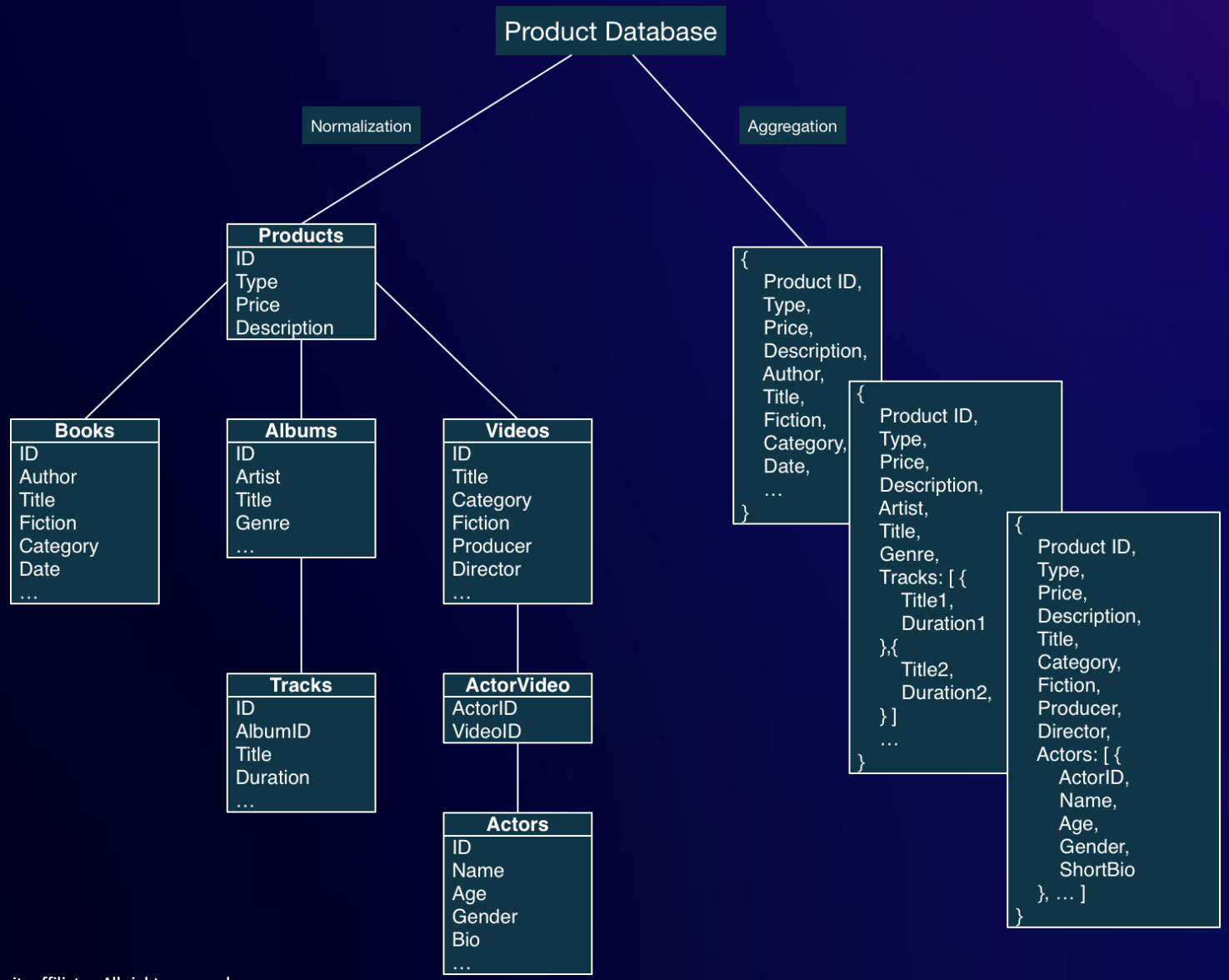
Key can be **multiple** columns

Joins **not supported**

GROUP BY and ORDER BY and aggregations **limited support**

Tables usually have **defined** schema

SQL vs. NoSQL design pattern

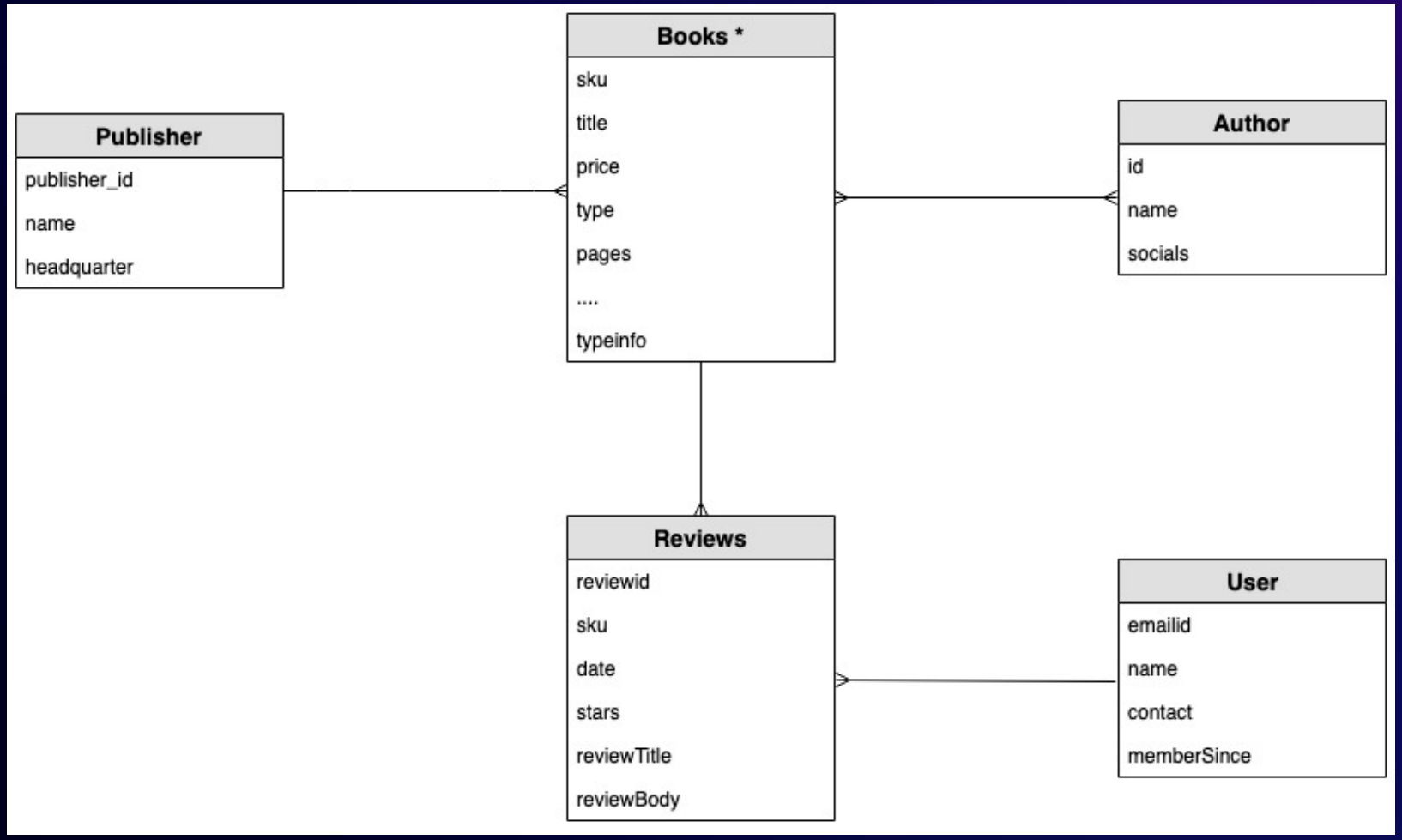


Data modeling steps

- Define the **use case, entities,** and **relations**
- Identify the **access** patterns
- Map it to a **model** specific to the database type

- Avoid relational design patterns (**think beyond joins!**)

Book review platform example



Book review platform: Access patterns

- Get **book** info (description and rating)
- Get top **reviews** for a **book**
- Get all **books** by an **author**
- Put/update **book** details
- Add/update **user** details
- Add/update a **review**
- Get **reviews** by **user**

Document database

Constraints:

- BSON/JSON documents
- 16 MB document limit

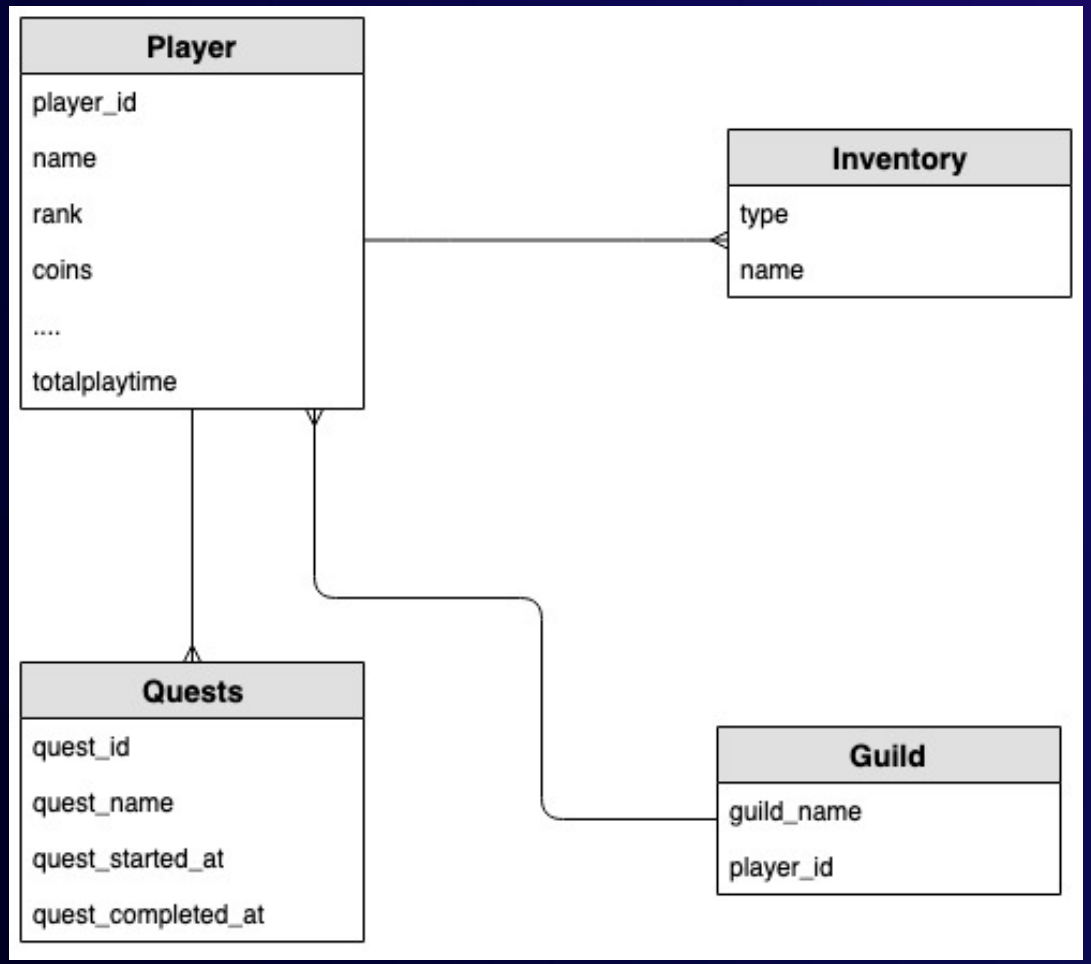
Avoid:

- Bloated documents
- Unbounded arrays
- Diluting the database into too many collections

Learnings:

- Embed unless compelling reason not to
- Polymorphic collections
- Aggregation pipelines – really powerful tool

Gaming example



Gaming: Access patterns

- Create **player**
- Increase coins for **player**
- Add **inventory** to **player**
- Remove **inventory** from **player**
- Add **player** to **guild**
- Get **player**
- Fetch **inventory** for **player**
- Get **players** by **guild**
- Get **quest** details for **player**
- Get **quests** for **player** for last week
- Get completed **quests** for **players**

Key-value database

Constraints:

- 400 KB item limit
- Single attribute partition key
- High cardinality partition key
- LSI limitations

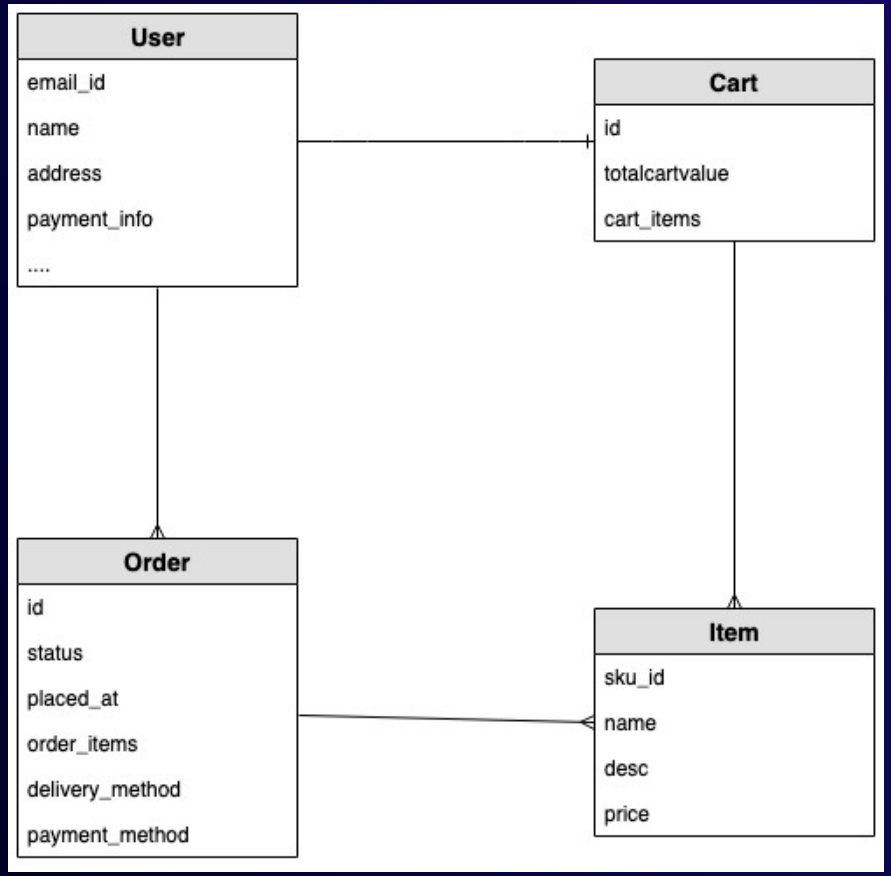
Avoid:

- Large items
- Reads on non-key attributes
- Scans

Learnings:

- Hierarchical items
- Overloading partition key and sort key
- GSIs for additional query capabilities
- Single-table design

Retail cart example



Retail cart: Access patterns

- Get historic details of an **order**
- Create an **order**
- Update **items** in **cart** (add/remove)
- Update status of an **order**
- Get all **orders** placed by **user**
- Display details of an **order**
- Get **orders** placed by a **user** containing an **item**

Wide-column database

Constraints:

- Multi-column partition key
- Multi-column clustering column

Avoid:

- Partitions > 100 MB
- Multi-partition reads/write
- High cardinality indexes

Learnings:

- Access pattern–oriented tables
- Multi-attribute primary key

AWS NoSQL purpose-built databases

CACHING

Amazon ElastiCache

MEMORY

Amazon MemoryDB

KEY-VALUE

Amazon DynamoDB

WIDE-COLUMN

Amazon Keyspaces

DOCUMENT

Amazon DocumentDB

TIME-SERIES

Amazon Timestream

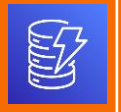
GRAPH

Amazon Neptune

LEDGER

Amazon QLDB

From bird’s eye view

	Amazon ElastiCache 	Amazon MemoryDB 	Amazon DynamoDB 	Amazon Keyspaces 	Amazon DocumentDB 
Category					
Manageability					
Scale					
API					

Takeaways

- NoSQL data modeling requires a shift in **mindset**
- **De-normalization** is a rule, not an option
- **Work backward** from your access patterns
- Keep re-iterating

Piqued your interest in data modeling?

ATTEND THESE SESSIONS

- **DAT410:** Advanced data modeling with Amazon DynamoDB
- **DAT304:** Data modeling in Amazon DocumentDB (with MongoDB compatibility)

Call to action!

NoSQL Workbench

DocumentDB workshop

DynamoDB workshop

Which NoSQL Database?

Keyspaces workshop

MemoryDB workshop

Thank you!

Please complete the session survey in the mobile app

Almas Shaikh
LinkedIn: almasshaikh1296

Mohammed Fazalullah Qudrath
LinkedIn: mohammedfazalullah