WEBO DAY 2021
04 MARZO • ONLINE CONFERENCE

DYNAMIC SCHEMAS FOR BACKEND API WITH AZURE SQL RELATIONAL AND POST-RELATIONAL UNITED FOR GOOD

DAVIDE MAURI
MICROSOFT MAURIDB
DAVIDE MAURI

THE DEVELOPER WHO LOVED DATABASES

Founder of SolidQ Italia (Now Lucient)

Microsoft Data Platform for 12 years

Azure SQL PM
Focus on Azure SQL Hyperscale & Developers

Heavy Metal Fan

Former Biker

http://davidemauri.it

#WEBDAY2021
A BOOK FOR YOU

SHAMEFUL PLUG

Chapter 1: A Database for the Modern Developer
Chapter 2: Azure SQL Kickstart
Chapter 3: Connecting and Querying Azure SQL
Chapter 4: Developing with Azure SQL – Foundations
Chapter 5: Developing with Azure SQL – Advanced
Chapter 6: Practical Use of Tables and Indexes
Chapter 7: Scalability, Consistency, and Performance
Chapter 8: Multi-model Capabilities
Chapter 9: More Than Tables
Chapter 10: Monitoring and Debugging
Chapter 11: DevOps with Azure SQL
MODERN DEVELOPMENT

WHAT’S THAT?

Supports multiple clients
Provides an API for accessing data and services
Data is available in a generic, consumable format
Can scale-out & Support automated deployment
Is Resilient and Fault-Tolerant
Is Designed to allow constant evolution and improvements
Secure from ground-up
MODERN DEVELOPMENT

WHAT’S THAT?

Supports multiple clients

Provides an API for accessing data and services

Data is available in a generic, consumable format

Can scale-out & Support automated deployment

Is Resilient and Fault-Tolerant

Is Designed to allow constant evolution and improvements

Secure from ground-up

#WEBDAY2021
SCHEMALESS

SCHEMWHAT?

There is no such thing as a “Schemaless” structure. (See: “Contradiction”)

https://www.martinfowler.com/articles/schemaless/

but schemaless structures still have an implicit schema

The discussion is “where do I prefer to manage my schema”?
- In the application (POCO/POJO/gRPC)
- In the database (Tables)
- I don't really care about the schema (my code will handle nicely any unknown property/field)
DYNAMIC SCHEMAS

THAT’S A DISCUSSION THAT REALLY COUNTS!

The real question is: “How to make it easy to change/evolve my schema, so that I don’t necessarily have to:

• Update Data or Database and/or
• Rebuild, Compile, Deploy my app

Both are potentially dangerous and user-impacting operations
Azure SQL natively support JSON

JSON is a great option to stabilize stored procedure, query or function signature

- Send back and forth an array of objects to Azure SQL
- OPENJSON + FOR JSON

JSON can also be used to store data into Azure SQL

- Can be indexed too
OPTIONS

PICK THE BEST FOR YOU

• Classic Table, use JSON as transport
• Hybrid Table: classic columns for well-known properties, “extension” column for anything you don’t know
• Document Table: Just JSON
WHAT ABOUT PERFORMANCES

RIGHT, WHAT ABOUT PERFORMANCES?

• JSON give you flexibility.

• Flexibility has some overhead (in fact gRPC is on the rise…)

• But…

LOCUST

Total Requests per Second
RESOURCES

CODE, CODE, CODE!

10K RPS REST API with Azure SQL, Dapper and JSON

https://github.com/Azure-Samples/azure-sql-db-dynamic-schema
THANKS!