

# Storing Schemaless Data with Azure Table Storage

---



**Mark Heath**

CLOUD ARCHITECT

@mark\_heath [www.markheath.net](http://www.markheath.net)



# Overview



## Structuring data

- Relational database
- Schemaless document database

## Azure database options

- SQL Database
- Cosmos DB
- Table Storage

## “Who’s Playing” demo app

- Store events and responses in a DB



# Databases in Serverless Architectures



**We need somewhere to store our data**

- Serverless does not dictate database type

**Two approaches to structuring data**

- Relational database
- NoSQL or document database

# Relational vs Document Databases

## Relational Databases

- Well understood by most developers
- Tables and relationships
- Lots of mature tooling
- Support transactions
- Complex queries: joins and grouping
- Powerful query language: SQL
- Need for an Object Relational Mapper
- Schema changes require migrations

## Document Databases

- Fewer tables
- Keep related data in one place
- Don't require schema definition up front
- Can be cheaper to run
- Scaling via sharding
- Less flexibility around querying
- Eventual consistency



# Azure Database Offerings



## SQL Database

SQL Server

Tooling

Entity Framework

Relational Database



## Cosmos DB

Azure DocumentDB

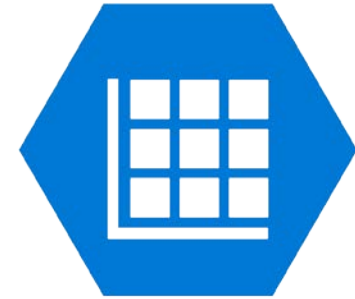
Query language

Document Database

Pricing: “Reserved Units”

MongoDB support

Azure Functions Binding



## Table Storage

Attribute Value Pair

Retrieve by key

Partition & row key

Schemaless

Serverless pricing

Azure Functions Binding



# Choosing a Database

---



# Table Design with SQL Database



```
SELECT * FROM Responses WHERE ResponseCode = "4H8ZDJW9"
```

```
SELECT * FROM Responses WHERE EventId = 1031
```



# Document Design with Cosmos DB



```
{
  "EventDate" : "2017-06-24T19:30:00+01:00",
  "Location" : "Wildern",
  "Responses" : [ {
    "Name" : "mark",
    "Email" : "mark@whosplaying.eu",
    "ResponseCode": "SH12gf87n2",
    "IsPlaying" : "yes"
  }, { "Name" : "jon",
    "Email" : "jon@whosplaying.eu",
    "ResponseCode": "wX6wP3aDp0",
    "IsPlaying" : "no"
  } ]
}
```



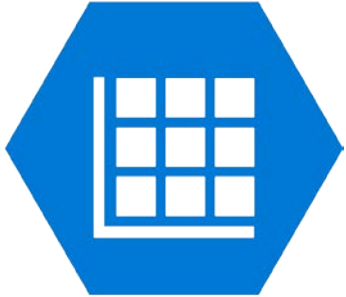
# Document Design with Cosmos DB



```
{
  "EventDate" : "2017-06-24T19:30:00+01:00",
  "Location" : "Wildern",
  "Responses" : [ {
    "Name" : "mark",
    "Email" : "mark@whosplaying.eu",
    "ResponseCode": "SH12gf87n2",
    "IsPlaying" : "yes"
  }, { "Name" : "jon",
    "Email" : "jon@whosplaying.eu",
    "ResponseCode": "wX6wP3aDp0",
    "IsPlaying" : "no"
  } ],
  "ResponseCodes" : ["SH12gf87n2",
    "wX6wP3aDp0" ]
}
```

```
SELECT * FROM events WHERE ARRAY_CONTAINS (events.ResponseCodes, "wX6wP3aDp0")
```





# Table Design with Table Storage

PartitionKey	"event"
RowKey	125712
DateTime	2017-06-24T19:30:00+01:00
Location	Wildern
ResponsesJson	[ { "Name" : "mark", "Email" : "mark@whosplaying.eu", "ResponseCode": "SH12gf87n2", "IsPlaying" : "yes" }, { "Name" : "jon", "Email" : "jon@whosplaying.eu", "ResponseCode": "wX6wP3aDp0", "IsPlaying" : "no" }]

<https://whosplaying.eu/index.html?responseCode=124718&eventId=125712>



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
--	--------------	-----------	---------------



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying			



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★		



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes			





# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★		



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost			



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★		



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration			





# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration	★		



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration	★	★ ★ ★	



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration	★	★ ★ ★	★ ★



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration	★	★ ★ ★	★ ★
Tooling			



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration	★	★ ★ ★	★ ★
Tooling	★ ★ ★		



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration	★	★ ★ ★	★ ★
Tooling	★ ★ ★	★ ★	



# Making the Choice

	SQL Database	Cosmos DB	Table Storage
Querying	★ ★ ★	★ ★	★
Schema Changes	★	★ ★ ★	★ ★
Cost	★ ★	★	★ ★ ★
Azure Functions Integration	★	★ ★ ★	★ ★
Tooling	★ ★ ★	★ ★	★ ★

**No clear winner!**

**Pick what suits your application**

**Factor in your own experience**



# Demo



## Updating the Create Event Function

- Add a new row to Table Storage
- Table Storage output binding





# Demo



## Updating the Get Event Function

- Look up the event in Table Storage
- Table Storage input binding



# Demo



## Update Response Function

- HTTP triggered
- Table Storage input binding
- Update Table Storage



# Demo



## Putting it all together

- Update response from web page



# Summary



## Azure database options

- SQL Database
- Cosmos DB
- Table Storage

**Choose what works best for your app**

## Table Storage bindings

- Output binding – create new rows
- Input binding – find existing rows
- Update rows in place

# Creating a secure event administration page

