Azure Open Source Day

Improve the development for
.NET application & Azure Cosmos DB for PostgreSQL

Who I am

Giorgio Desideri

Tech Lead of Cloud Solution

Seven Peaks Software

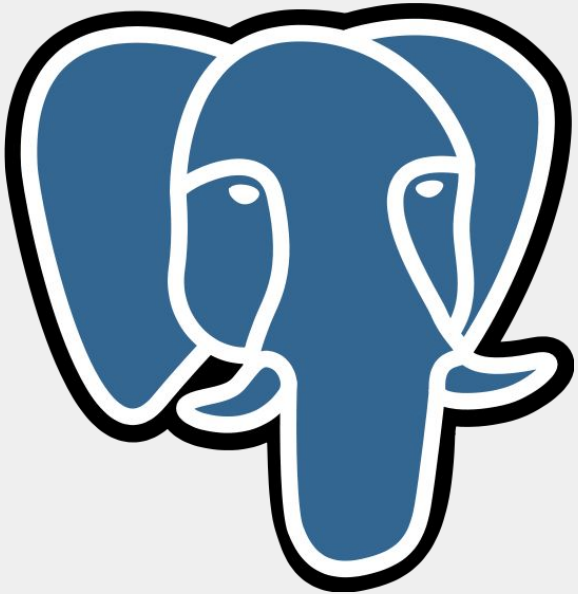
Agenda

Table of contents

1. Working with PostGreSQL Cluster
2. Development of the application
3. Application Use-cases

Working with a PostGreSQL Cluster

Who ever work with PGSQL Cluster or any other SQL-DB Cluster ?



Working with a PostGreSQL Cluster

Issues

- Understand the concept of “**Shard**” or “**Partitioning**”
- Performances and how to obtain them
- Monitoring of what the application is doing
  - and how the application does it

Working with a PostGreSQL Cluster

Issues

- Understand the concept of “**Shard**” or “**Partitioning**”
- Performances to retrieve and write data
  - … and how to obtain them
- Monitoring of what the application is doing
  - … and how the application does it

Do you have any other ?

Development of the application

Head scratch points

- Domain / Entities design
- Data Access
- Performances related to the Domain / Entities operations

"***Head Scratch***" = *think hard in order to find a solution to something.*

Entities design

Reasons

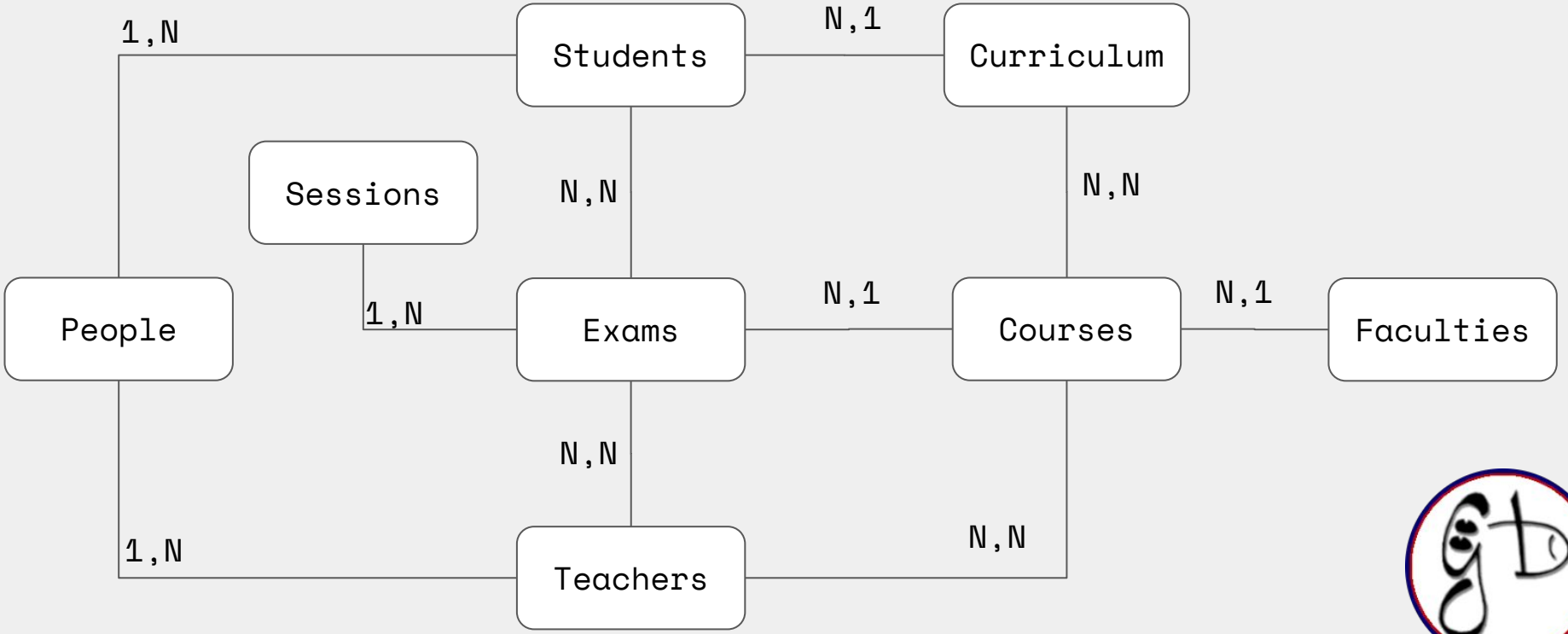
- Cosmos DB PostGreSQL has 3 different types of tables:
  - Distributed tables
  - Reference tables
  - Local tables

Entities design

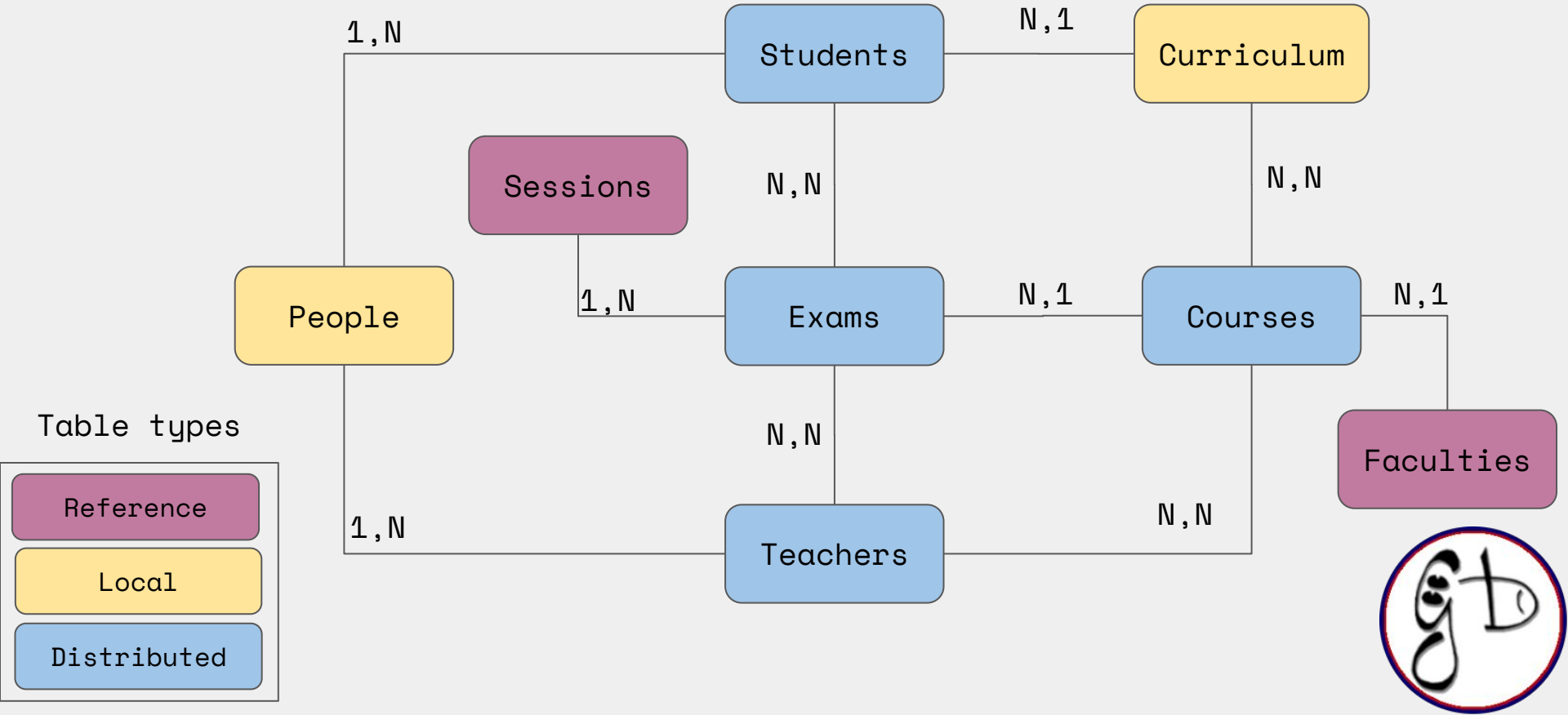
Challenge

- What (entities) can be “*distributed*” ?
- What (entities) can be “*reference*” ?
- What (entities) can be “*local*” ?

Entities design



Entities design



Entities design

Key concepts

- Understand the point of view for distribution
  - Sharding
  - Table colocation
  - Performance for read and write
  - Costs of Time complexity
    - Amount of Nodes
    - Total RUs

Entities design

Key Concepts

- SQL Normalization theory (1NF,2NF,3NF,…)
  - Difference between reference & not-reference entities
  - Using Entities ⇒ No good choice for cluster, because move semantic logic from data layer to business layer.
  - Cost of Space complexity
    - RUs used to retrieve or write
    - Table dimension (space allocation)

Entities design

Key Concepts

- Hierarchy analysis
  - TPH : Table-per-Hierarchy
  - TPT : Table-per-Type
  - TPC : Table-per-Concrete Type
  - Costs over space and time complexity

Data Access

Reason

- SQL vs ORM
- Connection Pool

Data Access

Challenge

- Usage of SQL brings:
  - More control over the queries
  - More flexibility over the DDL
  - Native support of the cluster commands
- Usage of ORM brings:
  - Code-first approach
  - Configuration over the ORM interfaces
  - Supervisioning over the migrations

Data Access

Challenge

- Connection Pools
  - It is a cluster, not a database !
  - Differences of implementation are performance-driven
    - Connection reuse
    - Fast connection acquisition
    - *Open()* or *Close()* automatic management

Data Access

Key Points

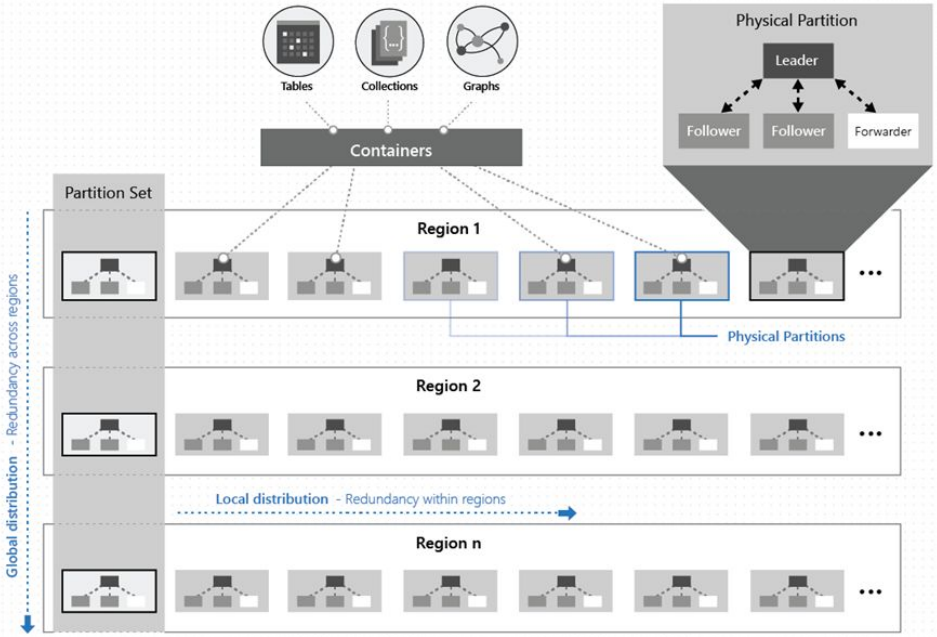
- All is performance-driven, so considering to study and analyse your queries
- SQL approach is the most “balanced” way
  - to use something that you “know”
  - reduce the risk to spend X amount of hours in customisation / overriding / work-arounding

Application Use-Cases

Reasons

- Multi-tenant SaaS
  - => Shard key identification
  - => RU understanding
- Real-Time application
  - => OLTP (On-Line Transaction Processing)
  - => Concurrency

Multi-Tenant SaaS



Shard Key Identification

- Logical partition
- Physical partition
- Partition Set
- Replication
- Consistency Level

Multi-Tenant SaaS

RUs understanding

- RU = Request Unit, 1 RU ≈ 1 Kb/s
- RU has impact over
  - all operations (Read / Write)
  - data model
- PoC-driven approach for development
  - Cannot be estimated OR Cannot be forecast
  - Must be implemented
- Concept of No-SQL DB usage

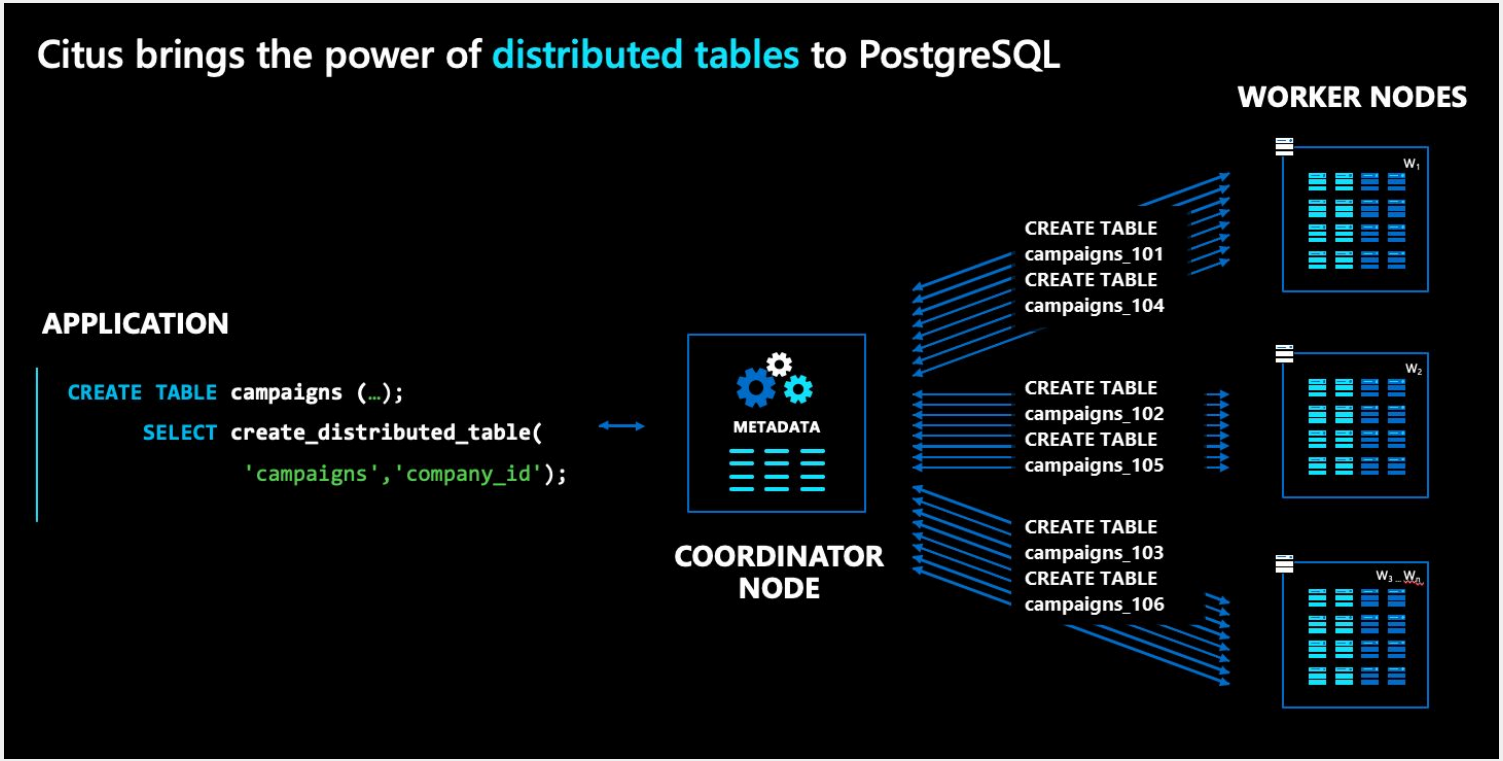
Real-Time application

Processing models

- OLTP : On-Line Transaction Processing
  - execute a number of transactions occurring concurrently
    - Transaction Lock
- OLAP : On-Line Analytical Processing
  - performing multidimensional analysis at high speeds on large volumes of data from a data warehouse, data mart
    - Roll-up / Drill-Down / Slice / Dice / Pivot

Real-Time application

Usage



Links

- https://learn.microsoft.com/en-us/azure/cosmos-db/postgresql/
- https://learn.microsoft.com/en-us/azure/cosmos-db/postgresql/concepts-cluster#nodes
- https://learn.microsoft.com/en-us/azure/cosmos-db/postgresql/concepts-nodes
- https://learn.microsoft.com/en-us/azure/cosmos-db/postgresql/quickstart-build-scalable-apps-overview
- https://learn.microsoft.com/en-us/azure/cosmos-db/postgresql/quickstart-build-scalable-apps-concepts
- https://learn.microsoft.com/en-us/azure/cosmos-db/postgresql/quickstart-build-scalable-apps-model-multi-tenant
- https://learn.microsoft.com/en-us/azure/cosmos-db/postgresql/resources-pricing

The End

Q / A