



Divide and Conquer Multi-tenancy in Postgres

Karen Jex | Senior Solutions Architect @ Crunchy Data
CitusCon | April 2023



Agenda

- Introduction
- Setting the Scene
- Multi-tenant Options
- How do you Choose?
- Conclusions

Agenda

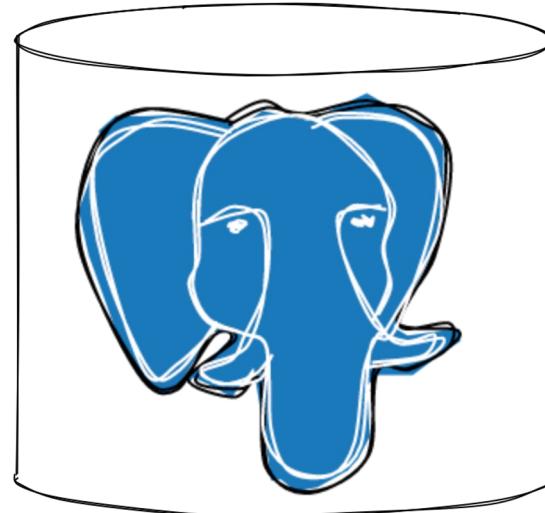
- **Introduction**
- Setting the Scene
- Multi-tenant Options
- How do you Choose?
- Conclusions

Introduction

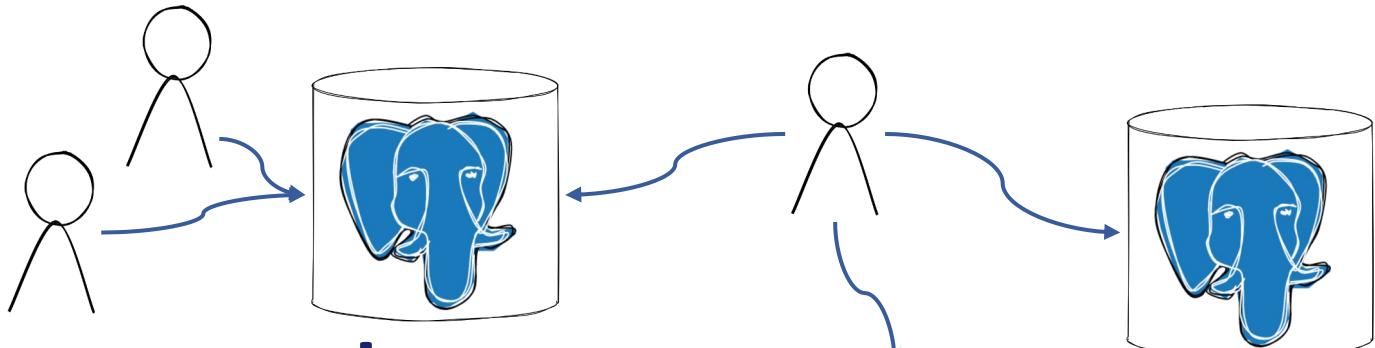
What is Multi-Tenancy?

Introduction

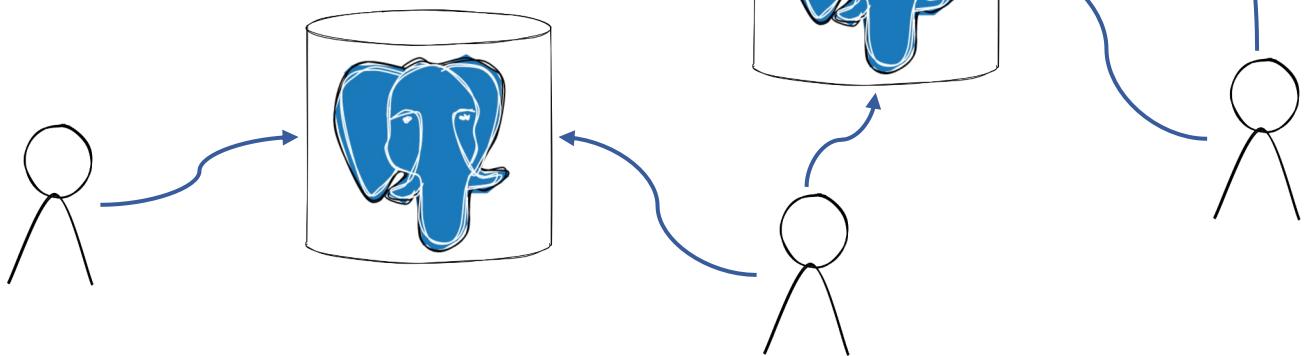
Why do you need Multi-tenancy?



Introduction



Why do you need Multi-tenancy?



Agenda

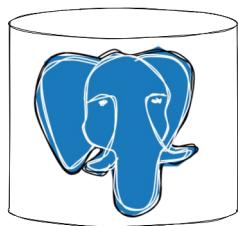
- Introduction
- **Setting the Scene**
- Multi-tenant Options
- How do you Choose?
- Conclusions

Setting the Scene Cake!



Setting the Scene

Hooray for Cake

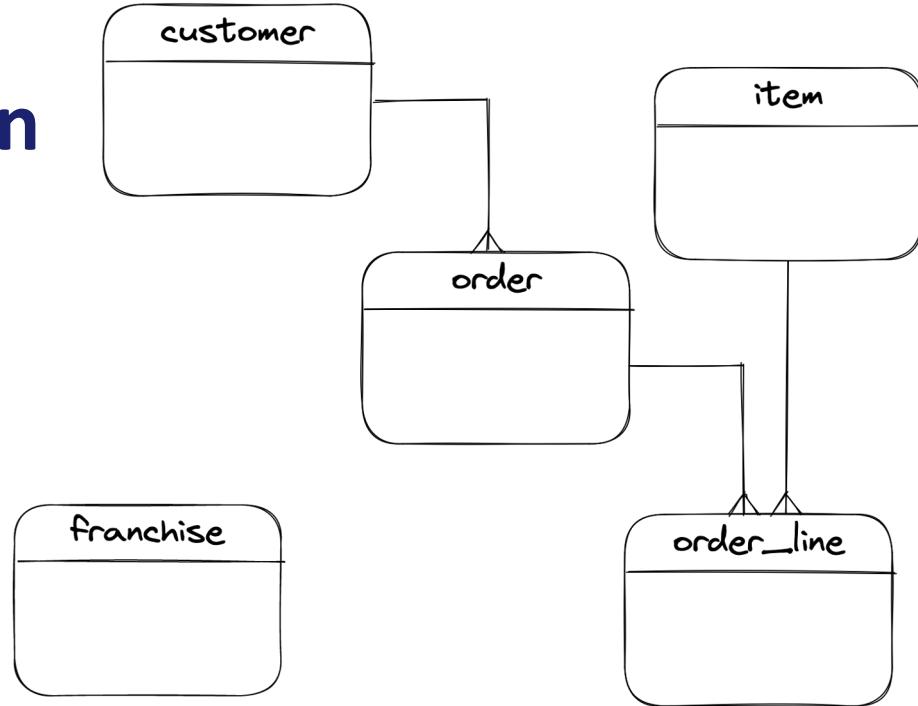
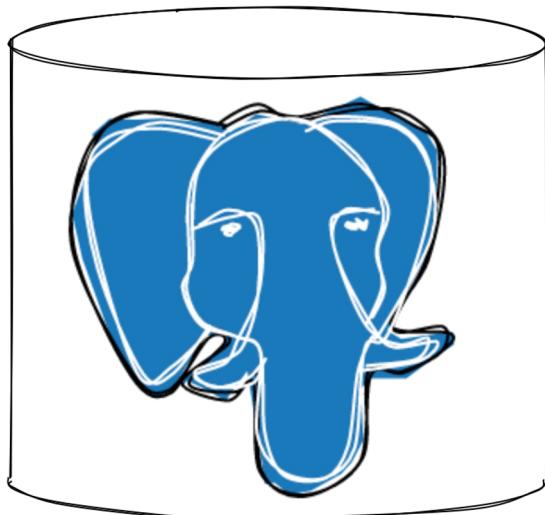


DBA for “HFC”



Setting the Scene

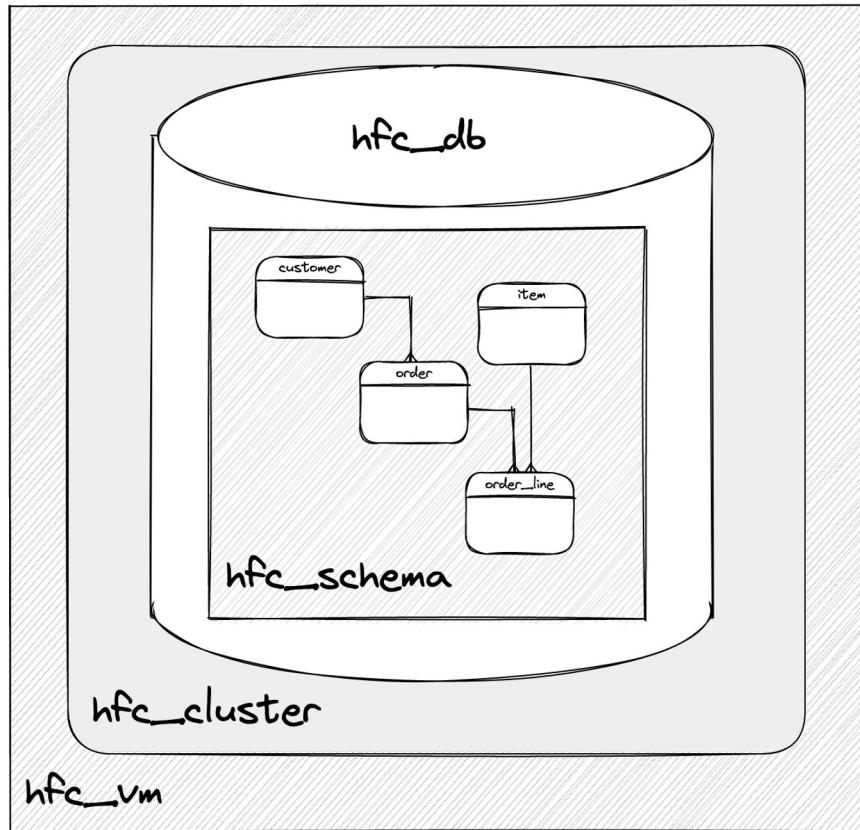
Database Preparation



Setting the Scene

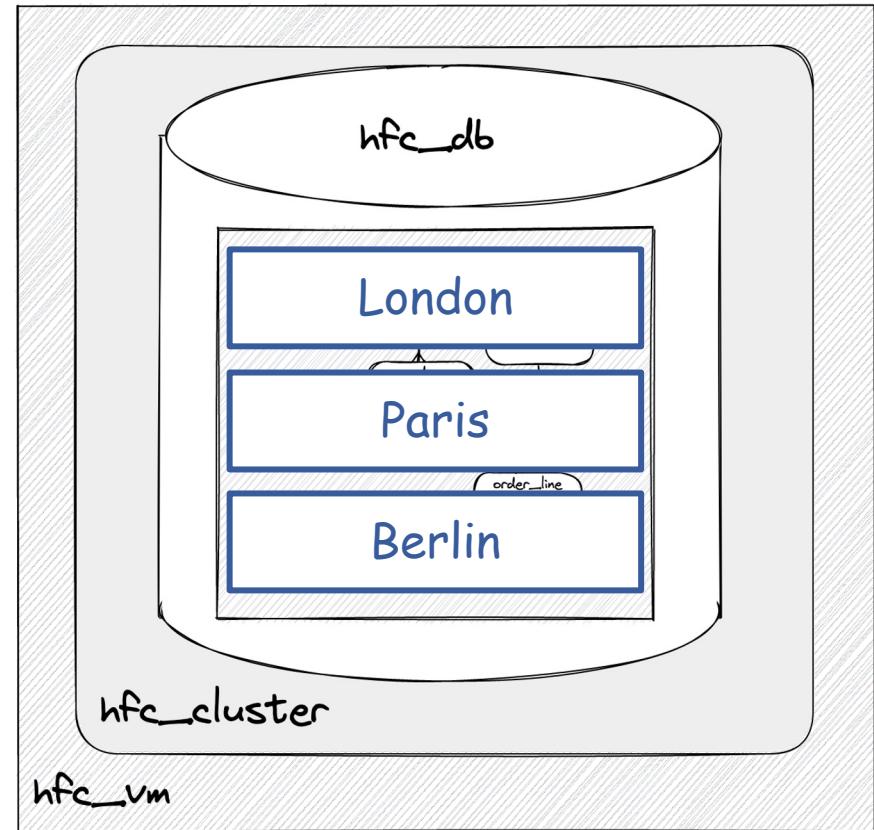
Initial Architecture

Host	hfc_vm
Postgres Cluster	hfc_cluster
Database	hfc_db
Schema	hfc_schema
Tables	



Setting the Scene

Adding Franchises



Setting the Scene

Adding Franchises



Agenda

- Introduction
- Setting the Scene
- **Multi-tenant Options**
- How do you Choose?
- Conclusions

Multi-tenant Options

What is the “right” architecture?

Multi-tenant Options



performance



isolation



cost

customer confidence



high availability

manageability



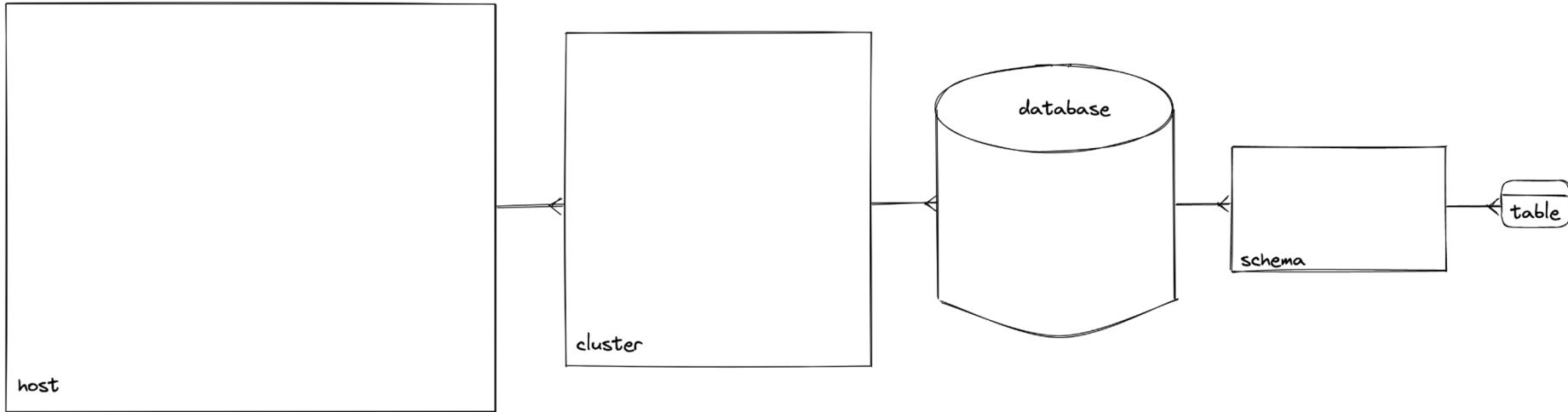
Multi-tenant Options

The Cake Hierarchy



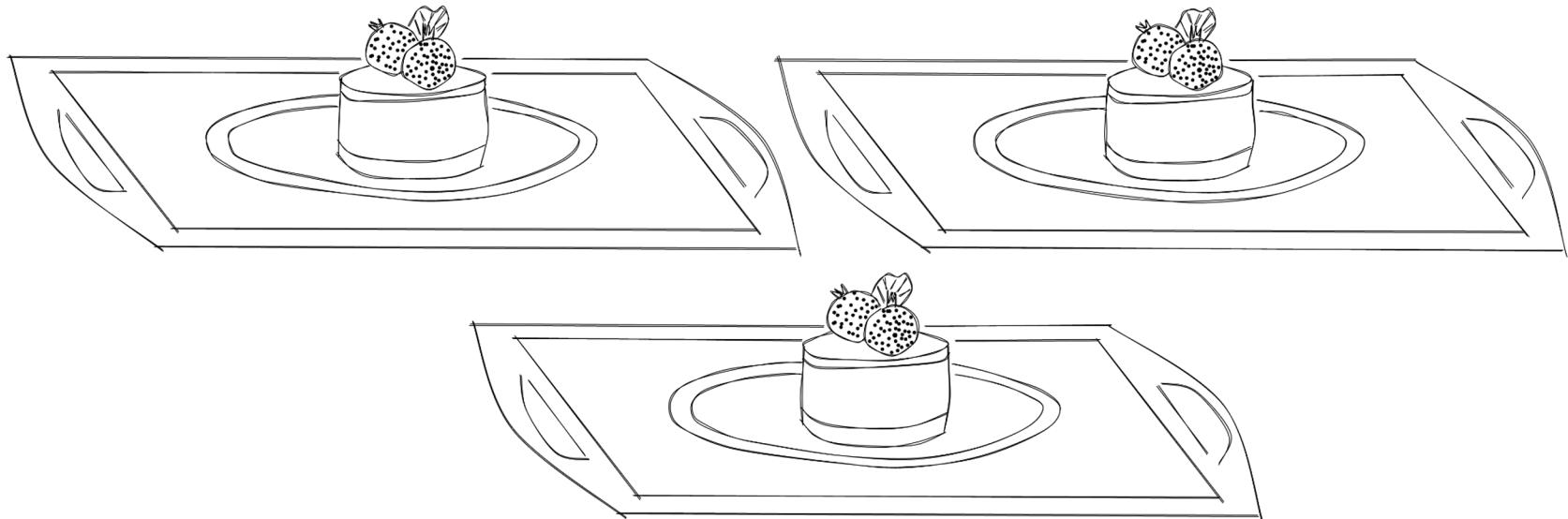
Multi-tenant Options

The Cake Architecture Hierarchy



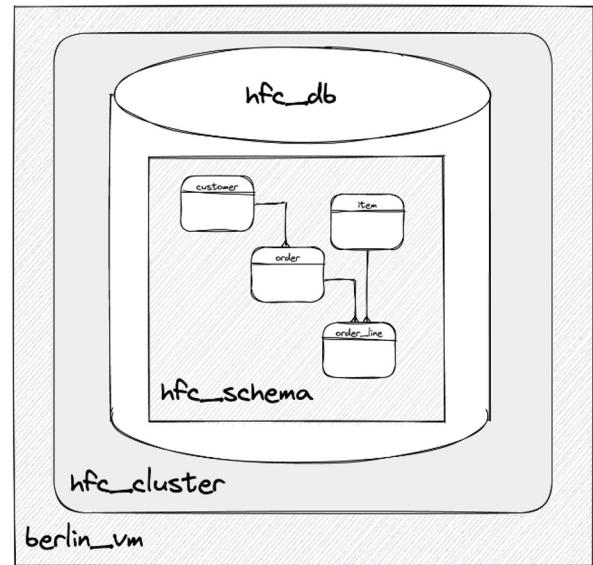
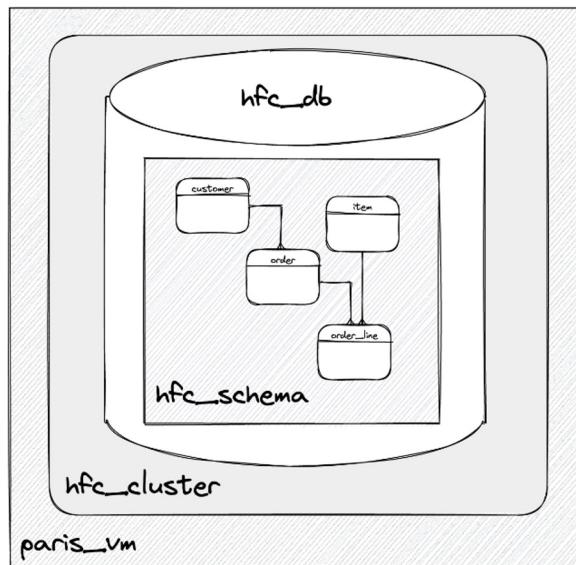
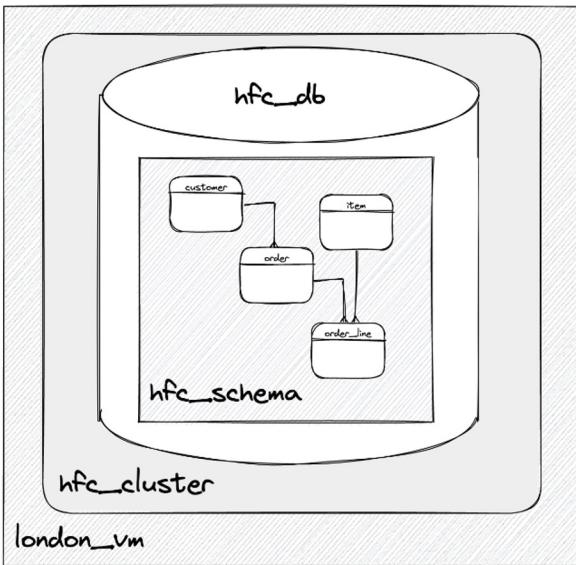
Multi-tenant Options

Cakes on Separate Trays



Multi-tenant Options

~~Cakes on Separate Trays~~ One Host per Tenant



Multi-tenant Options - One Host per Tenant

Advantages

- Flexibility
 - Parameters
- Separation
 - Upgrades
 - Backup and Restore
 - Performance
- Scalability
- Geographical Distribution

Multi-tenant Options - One Host per Tenant

Disadvantages

- Global Reporting
- Shared Reference Data
- High Overheads
 - Infrastructure
 - Setup
 - Monitoring
 - Maintenance

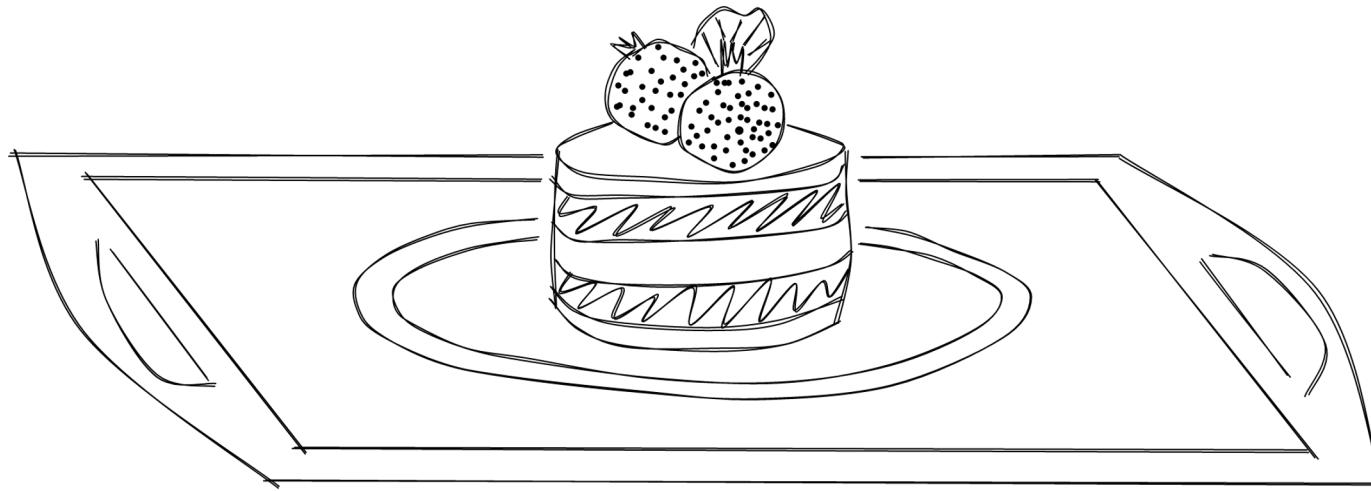
Multi-tenant Options - One Host per Tenant

Use Cases

- Tenants managed separately
- High isolation
- No shared data

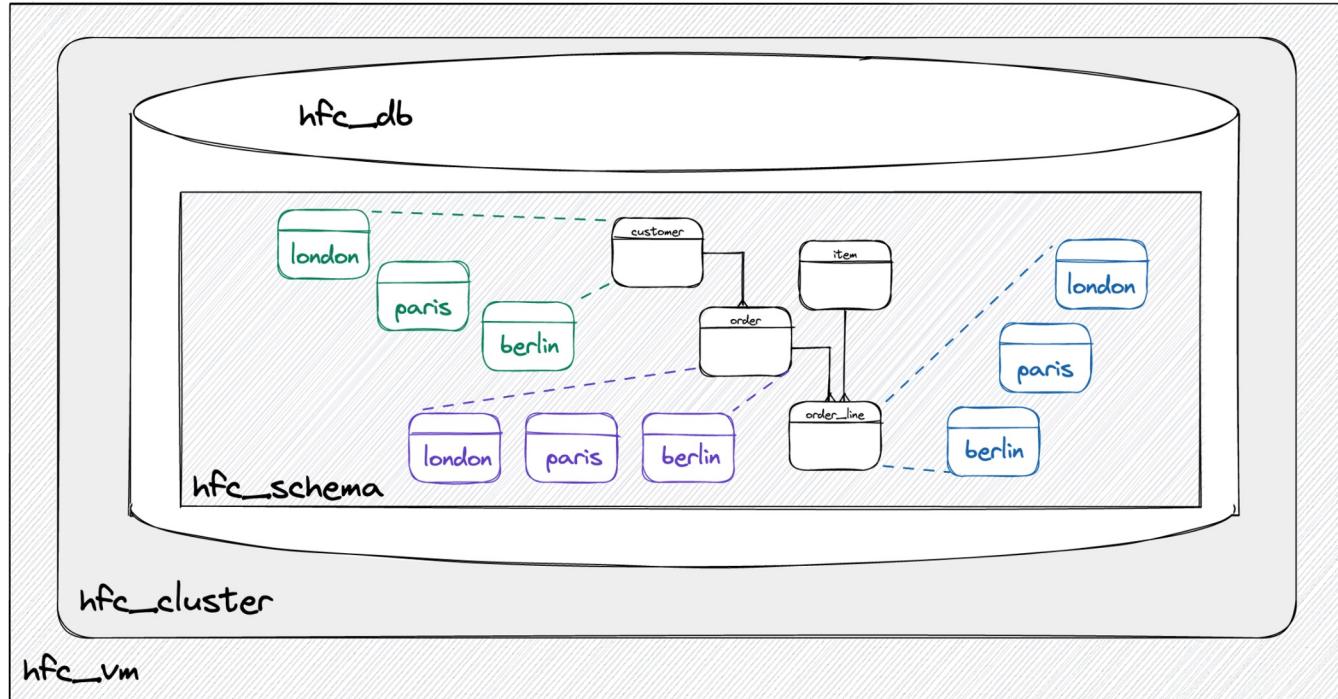
Multi-tenant Options

Multi-layered Cake



Multi-tenant Options

~~Multi-layered Cake~~ Partitioned Tables



Multi-tenant Options - Partitioned Tables

Schema Changes

- Create franchise table
- Add franchise_id to tables
- Partition tables by franchise_id
- Application: filter by franchise

Multi-tenant Options - Partitioned Tables

Row Level Security (RLS)

<https://www.postgresql.org/docs/current/ddl-rowsecurity.html>

- Enable RLS on Tables

```
ALTER TABLE my_table  
ENABLE ROW LEVEL SECURITY;
```

- Create security policies
 - Franchisees see only own data
 - Head office staff report on all data

Multi-tenant Options - Partitioned Tables

Advantages

- Shared Reference Data
- Global Reporting
- Lower Overheads
 - Setup
 - Monitoring
 - Maintenance

Multi-tenant Options - Partitioned Tables

Disadvantages

- Reduced Flexibility
 - Parameters
- Reduced Separation
 - Upgrades
 - Backup and Restore
 - Performance

Multi-tenant Options - Partitioned Tables

Use Cases

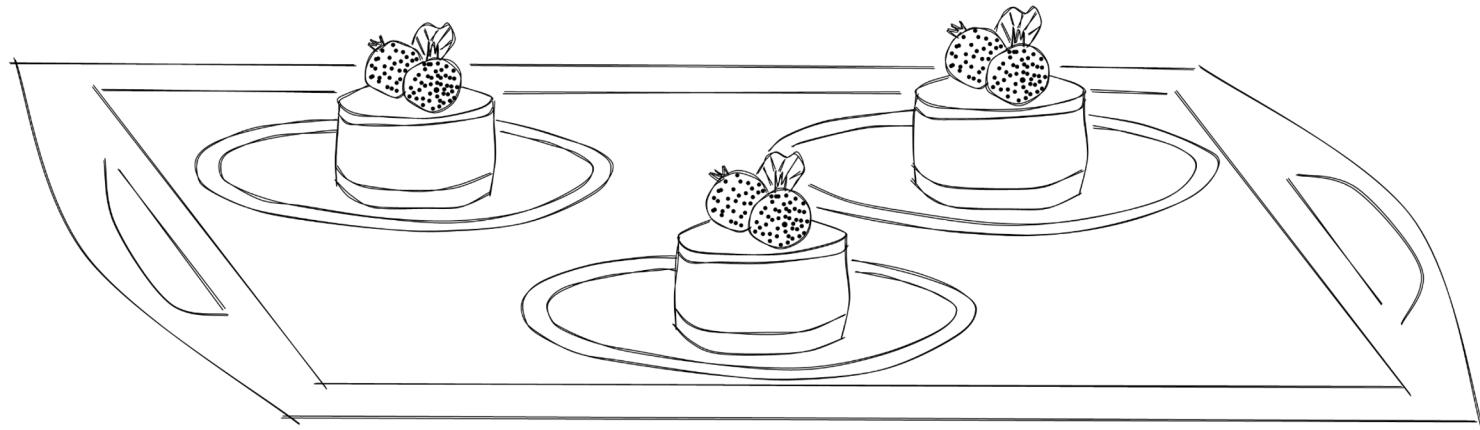
- Everything managed centrally
- Global Reporting
- Small number of tenants

Multi-tenant Options

Are there other options?

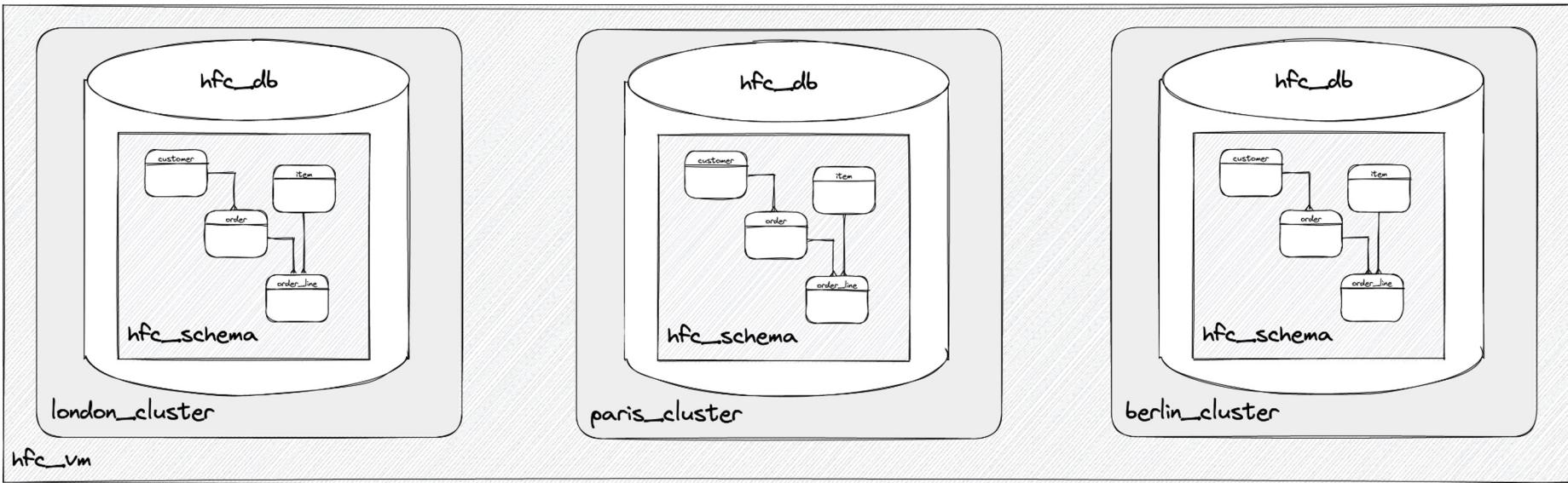
Multi-tenant options

Each cake on its own plate



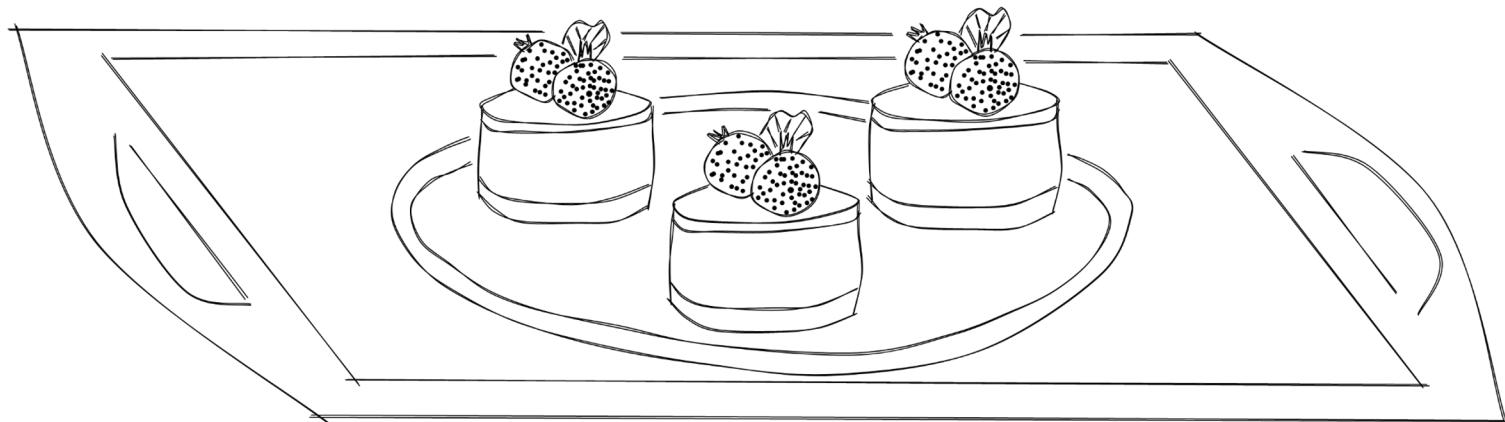
Multi-tenant options

~~Each cake on its own plate~~ Postgres Cluster per Tenant



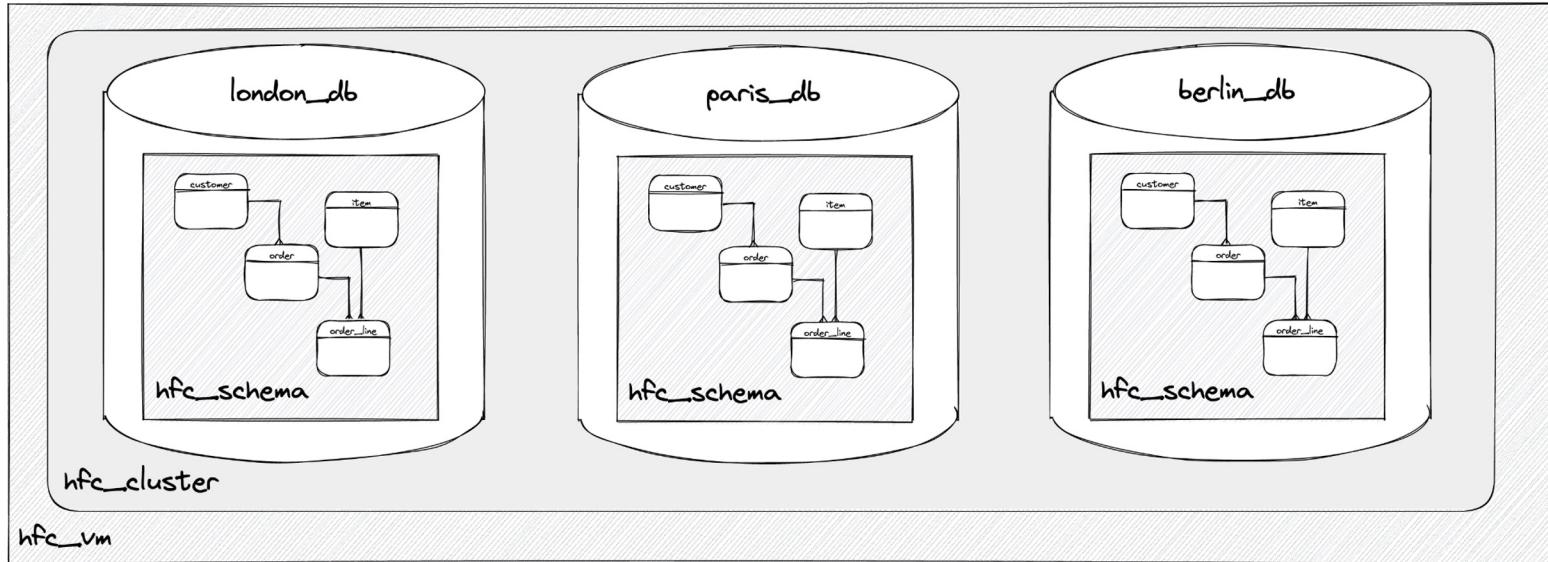
Multi-tenant options

All the cakes on one plate



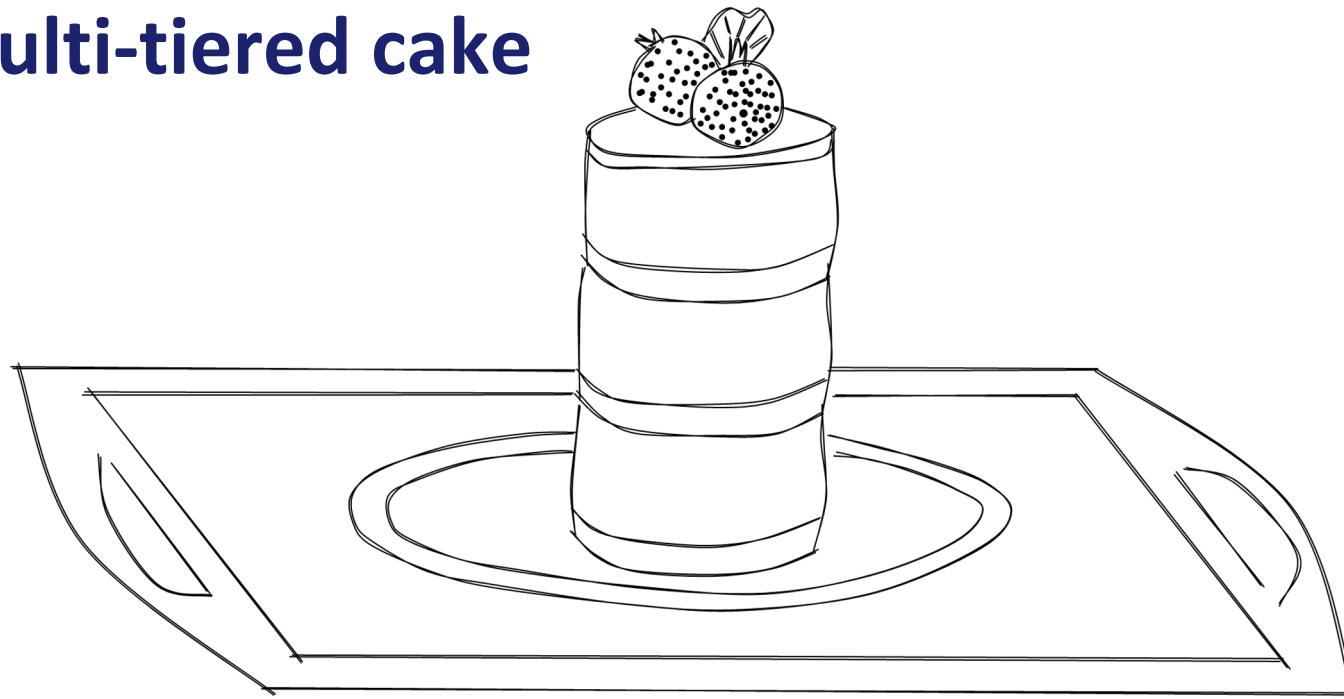
Multi-tenant options

All the cakes on one plate **One Database per Tenant**



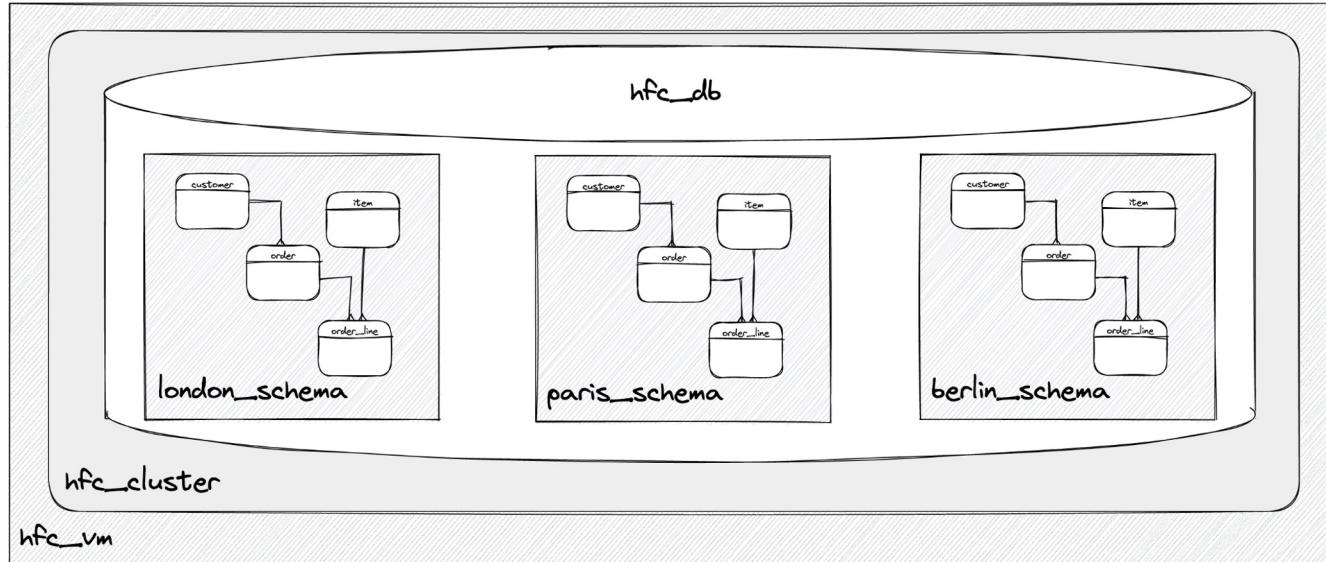
Multi-tenant options

A Multi-tiered cake



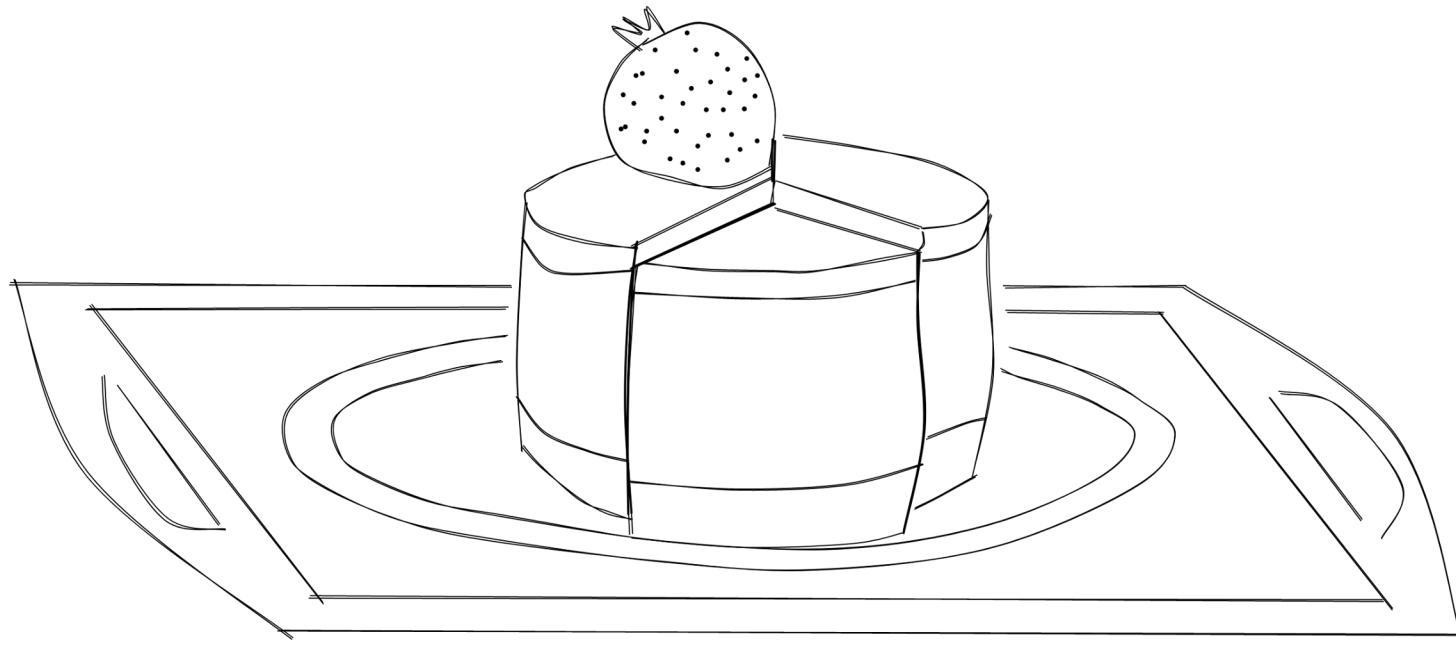
Multi-tenant options

A Multi-tiered cake A Schema per Tenant



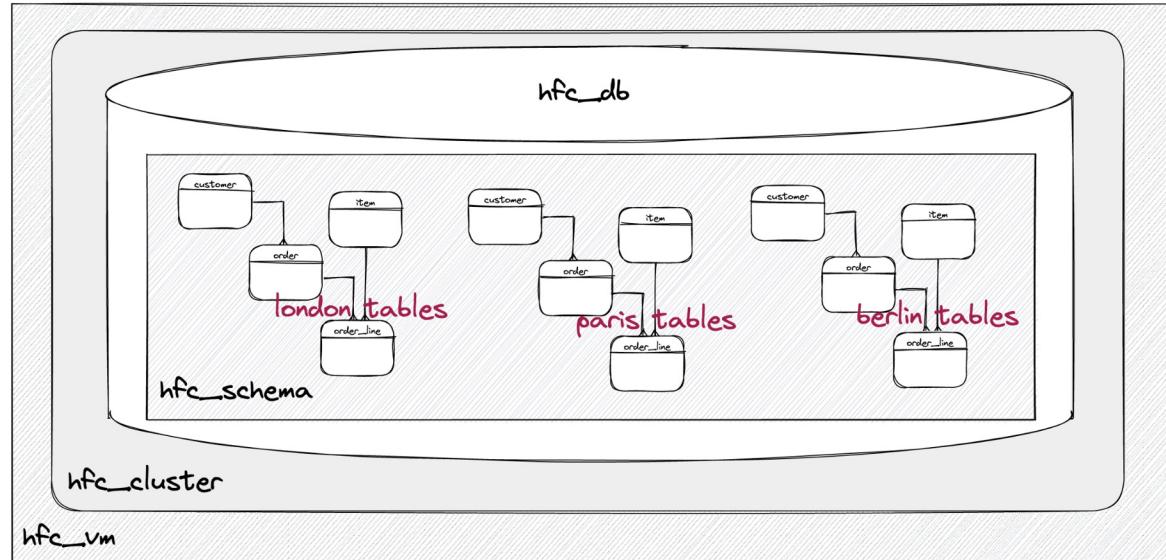
Multi-tenant options

One cake cut into slices



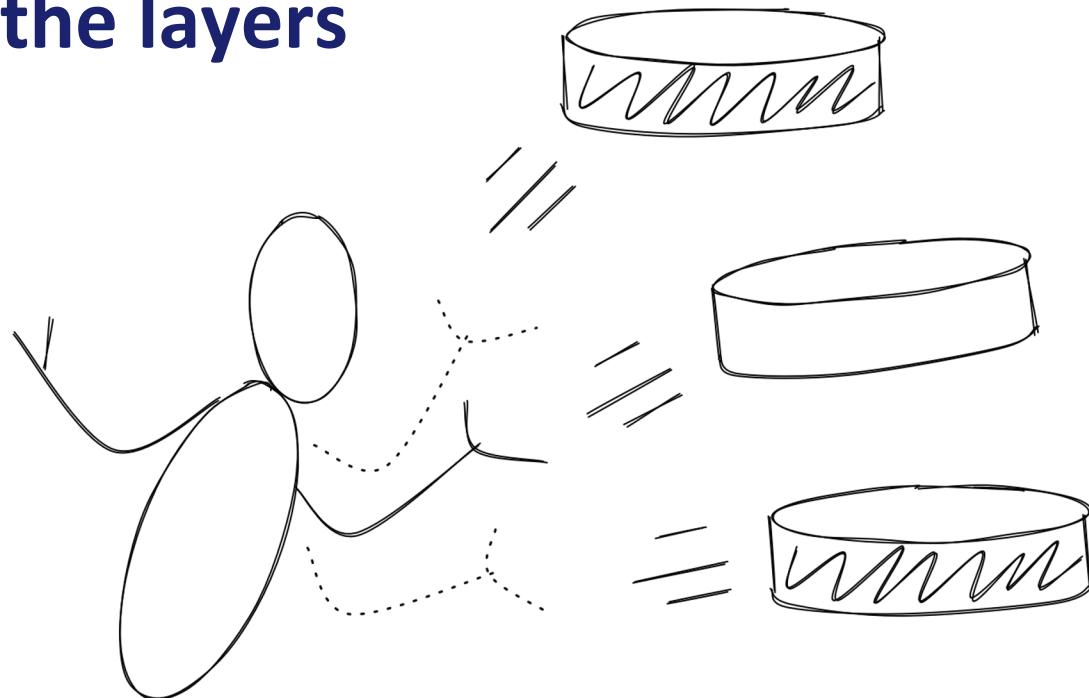
Multi-tenant options

~~One cake cut into slices~~ A set of tables per franchise



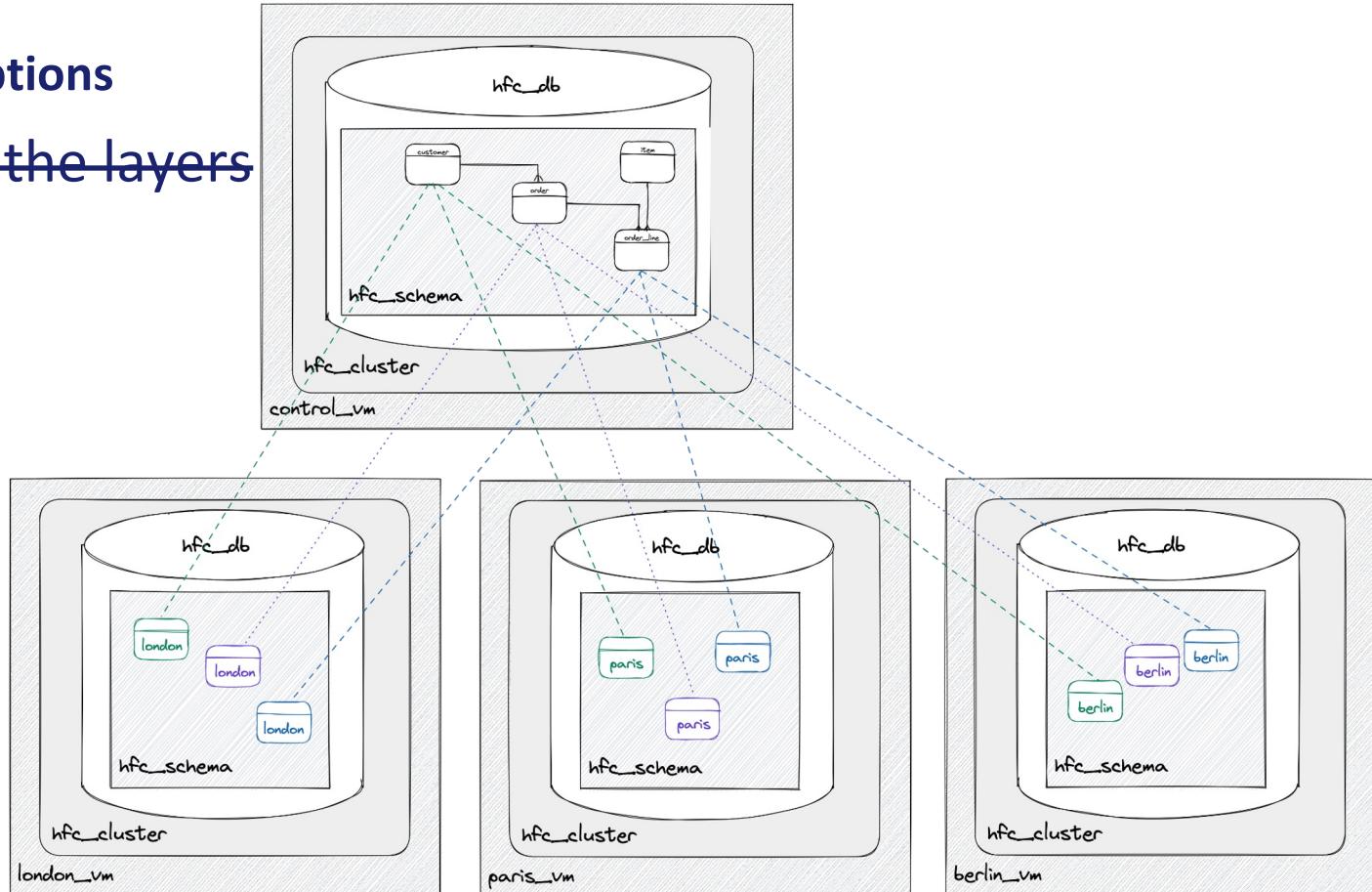
Multi-tenant options

Separate out the layers

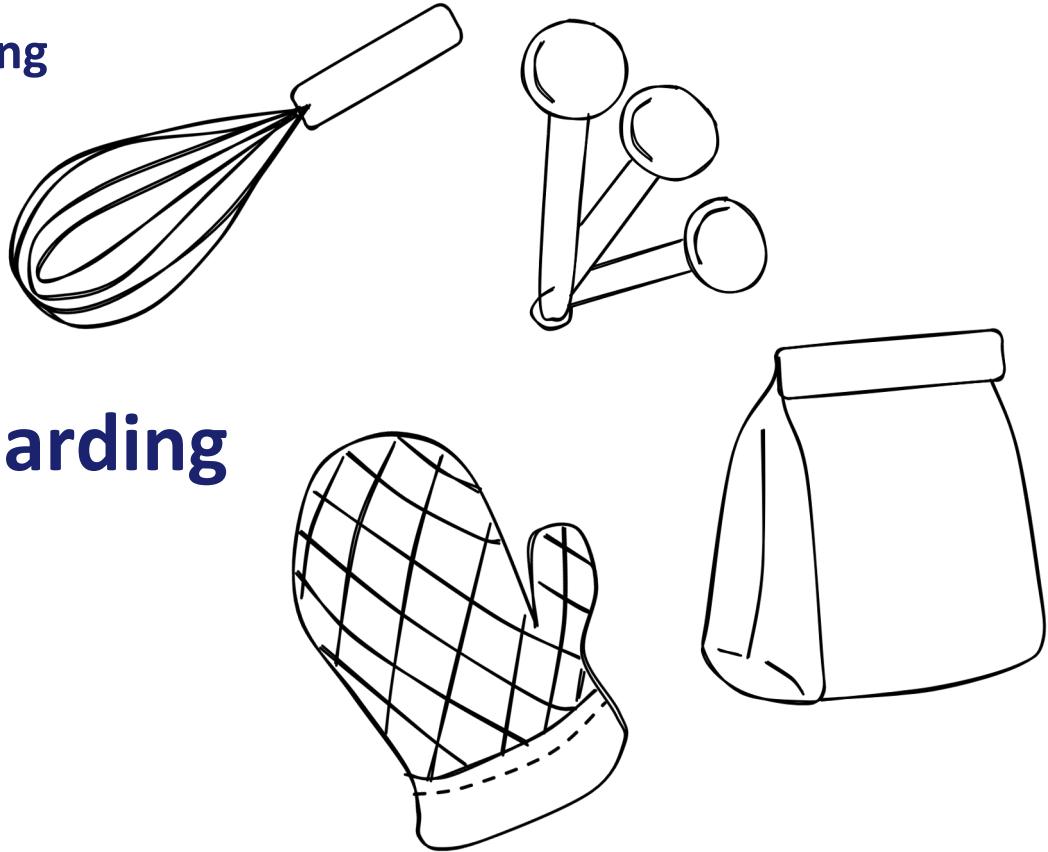


Multi-tenant options

~~Separate out the layers~~
Sharding



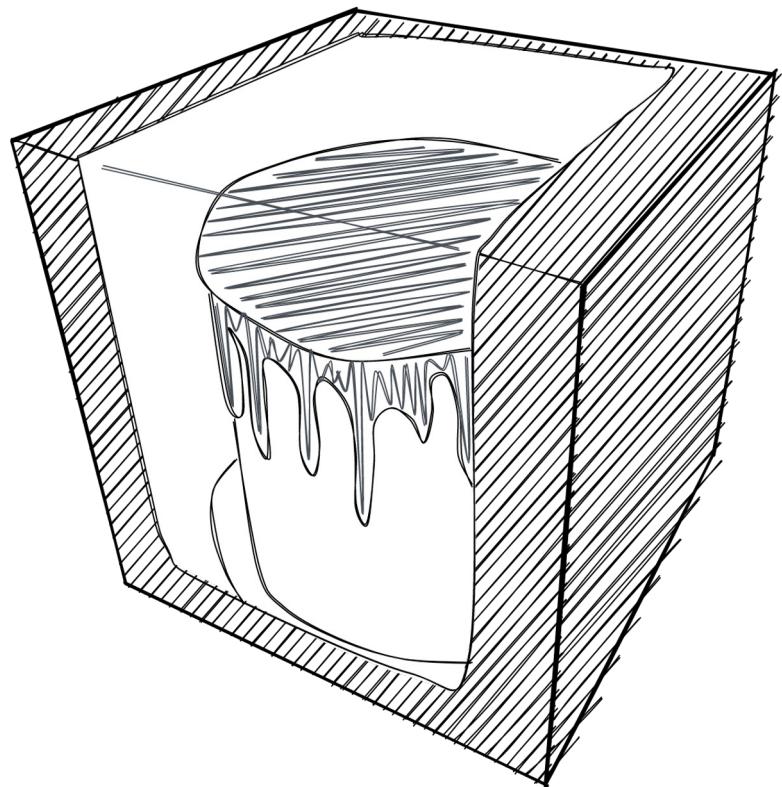
Multi-tenant Options - Sharding



Bake it Yourself Sharding

Multi-tenant Options - Implementing Sharding

Ready-prepared Sharding



Multi-tenant Options - Sharding

Use Cases

- Geographical Distribution
- Central Management
- Scaling

Multi-tenant options

Cupcakes Containers



Agenda

- Introduction
- Setting the Scene
- Multi-tenant Options
- **How do you Choose?**
- Conclusions

How do you Choose - Considerations 1/6

Responsibilities

- Infrastructure
- Application
- Schema
- Data

How do you Choose - Considerations 2/6

Shared Data

- Shared Reference Data
- Global Reporting

How do you Choose - Considerations 3/6

Costs

- Proprietary Software Licensing
- Hardware
- Cloud
- Management/Maintenance

How do you Choose - Considerations 4/6

Geographical Distribution

- Scaling
- Performance
- Privacy
- Compliance

How do you Choose - Considerations 5/6

Number of Tenants

- Performance
- Scaling
- Maintenance

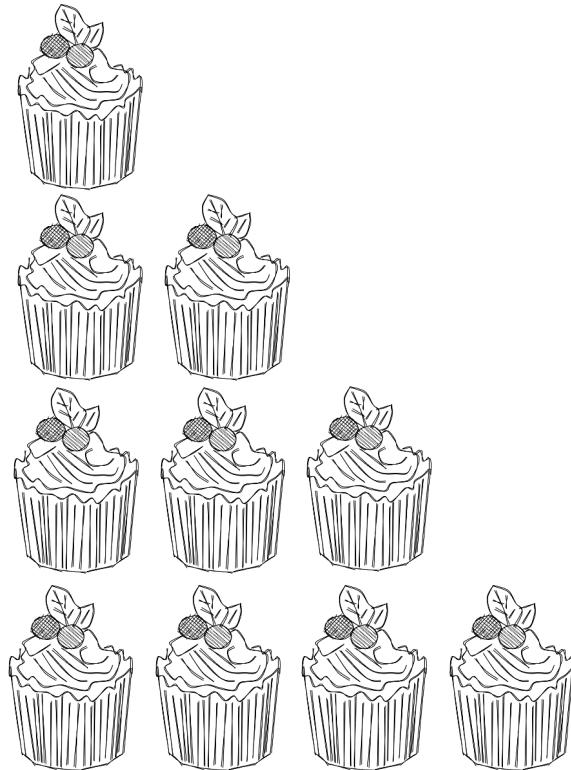
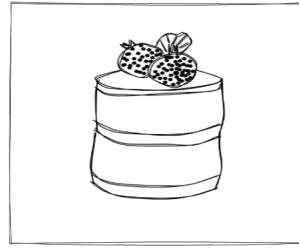
How do you Choose - Considerations 6/6

Level of Isolation

- Configuration
- Upgrades
- Backup/Restore
- Performance

How do you Choose?

Comparison



One tenant per...	Shared reference data?	Cross-tenant reporting?	Geographical distribution?	Level of isolation of tenants	Overheads/ Complexity
VM					
Cluster					
DB					
Schema					
Partition					
Shard					

One tenant per...	Shared reference data?	Cross-tenant reporting?	Geographical distribution?	Level of isolation of tenants	Overheads/ Complexity
VM					
Cluster					
DB					
Schema					
Partition					
Shard					

One tenant per...	Shared reference data?	Cross-tenant reporting?	Geographical distribution?	Level of isolation of tenants	Overheads/ Complexity
VM					
Cluster					
DB					
Schema					
Partition					
Shard	()	()			

One tenant per...	Manage parameters per tenant?		Upgrade individual tenant?			Backup/restore individual tenant?
	OS	PG	OS	PG	Schema	
VM						
Cluster						
DB						
Schema						
Partition						
Shard						

One tenant per...	Manage parameters per tenant?		Upgrade individual tenant?			Backup/restore individual tenant?
	OS	PG	OS	PG	Schema	
VM						
Cluster						
DB						
Schema						
Partition						
Shard						

How do you Choose

Keep it Super Simple!

Agenda

- Introduction
- Setting the Scene
- Multi-tenant Options
- How do you Choose?
- **Conclusions**

Conclusions

- Many multi-tenant options in Postgres
- No single “right” answer
- Carefully consider your use-case

Conclusions



performance



isolation



cost

customer confidence



high availability

manageability



Conclusions

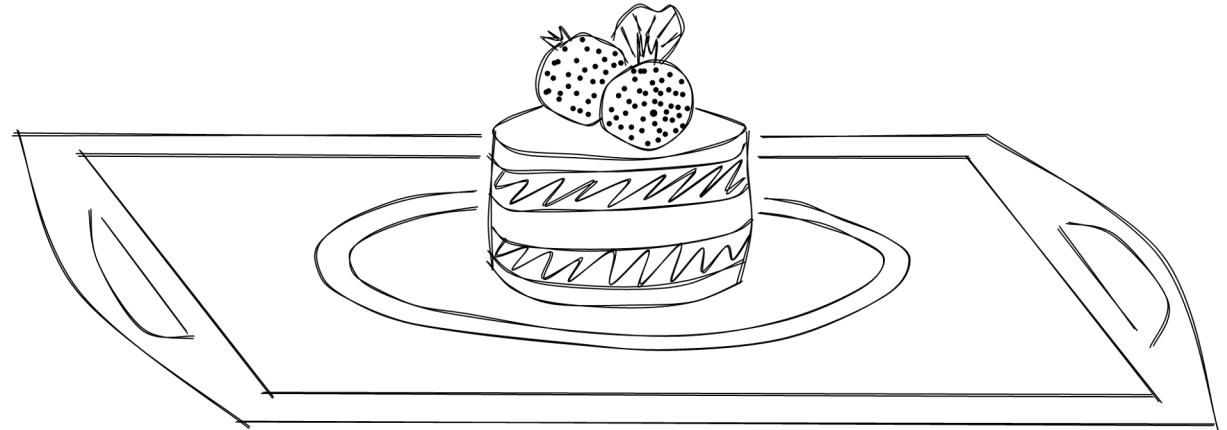
Hooray for Cake - Considerations

- Small number of tenants
- Centrally managed
- Shared reference data
- Global Reporting
- No Geographical Distribution

Conclusions

Hooray for Cake - Decisions

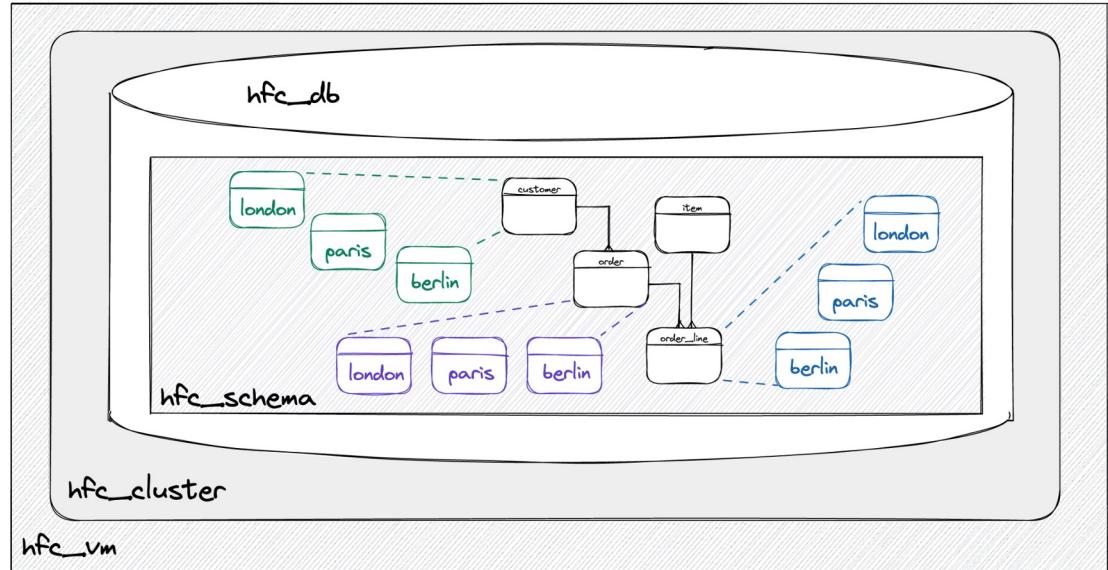
- Layered Cake



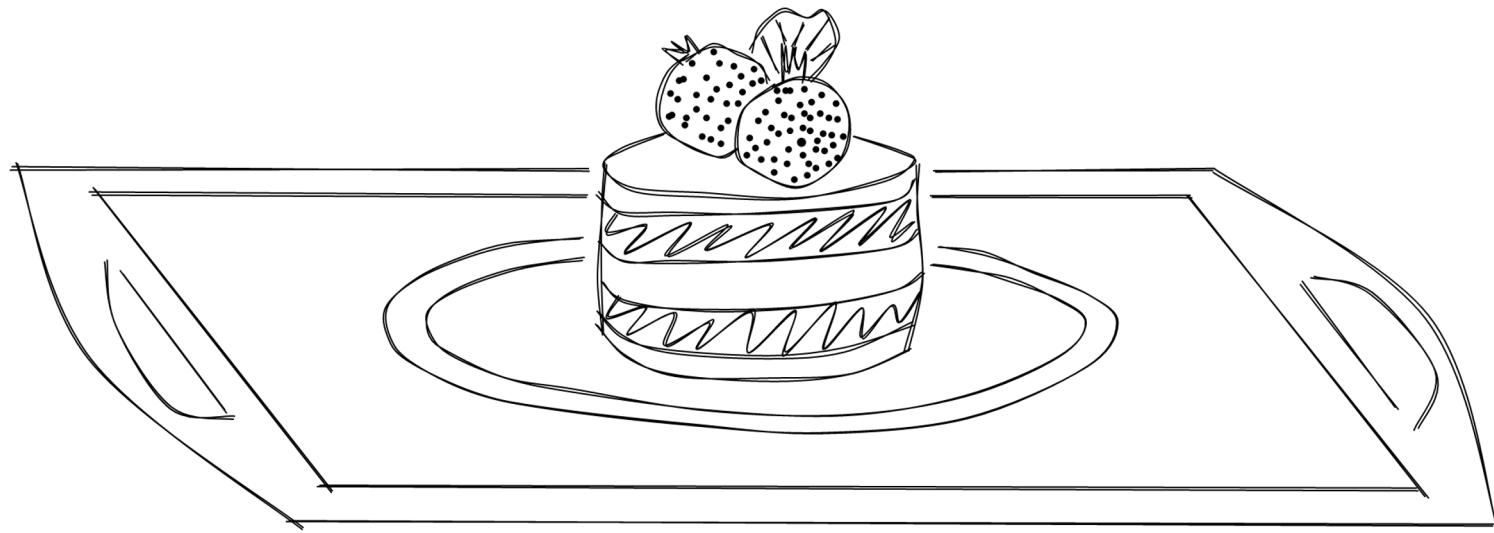
Conclusions

Hooray for Cake - Decisions

- Single database
- Partitioning
- Row Level Security



Conclusions



Thank You!



@karenhjex | @karenhjex@mastodon.online | karen.jex@crunchydata.com