

# Divide and Conquer Multi-tenancy in Postgres

Karen Jex

March 2022



# whoami

Karen Jex | Senior Database Consultant EMEA @ EDB | @karenhjex



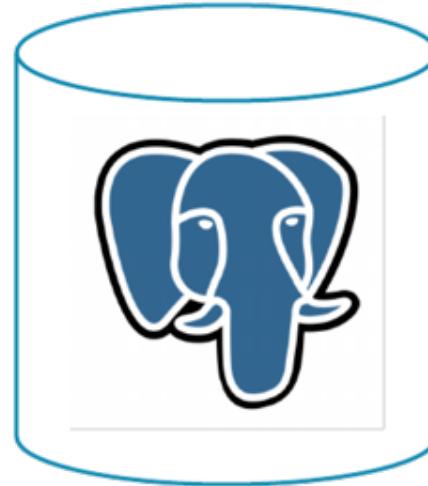
# whoami

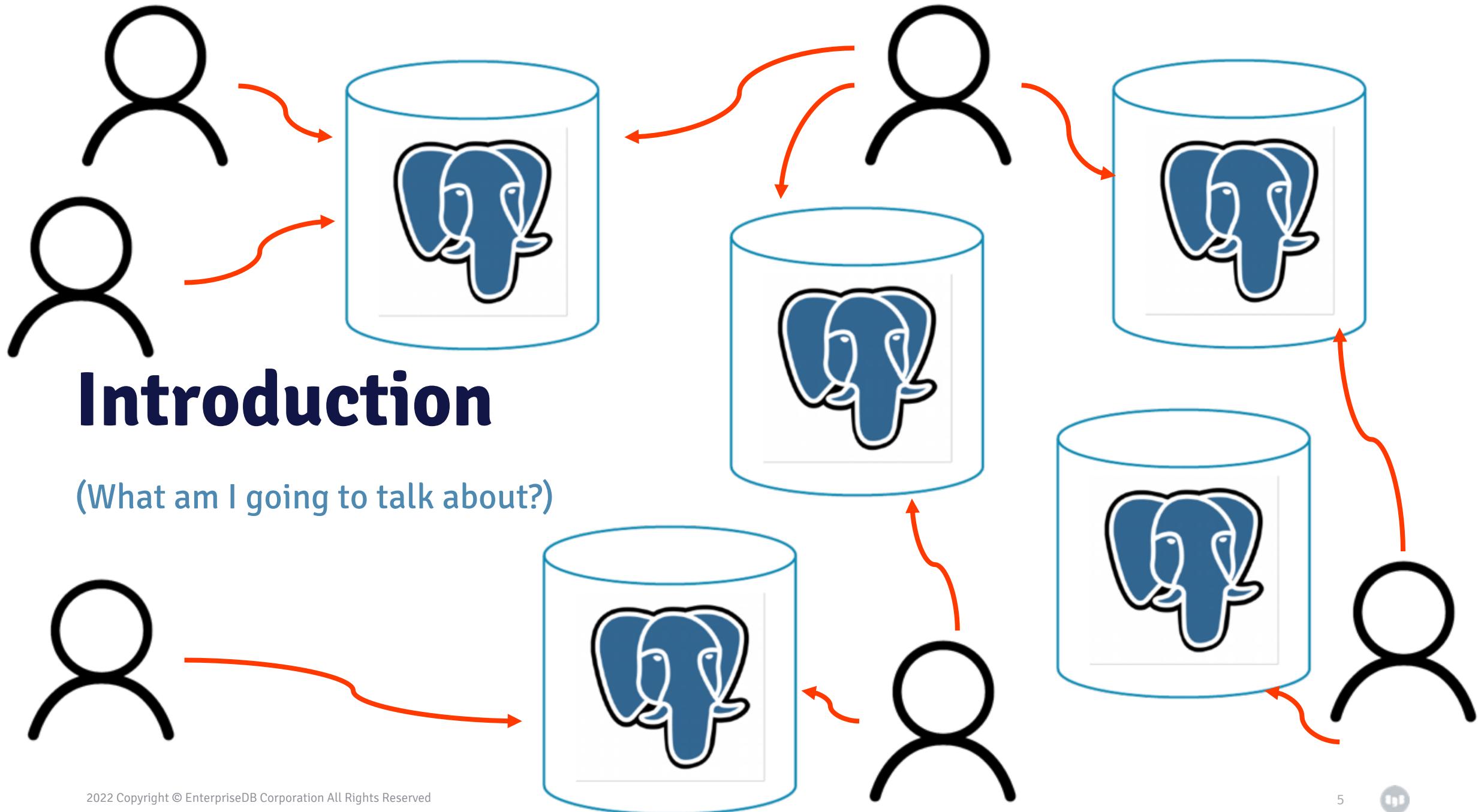
Comet and Shadow's human



# Introduction

(What am I going to talk about?)





# What is Multi-Tenancy?



multi-

1a : **many** : multiple : much *multivalent*

b : **more than two** *multilateral*

c : **more than one** *multiparous* *multibillion*

2 : many times over *multimillionaire*

<https://www.merriam-webster.com/dictionary>

# What is Multi-Tenancy?

multi-

**having many:**

a multi-coloured skirt

a multi-vitamin pill

<https://dictionary.cambridge.org/dictionary/english>



**tenant**

noun

2 : occupant, dweller

<https://www.merriam-webster.com/dictionary>

# What is Multi-Tenancy?

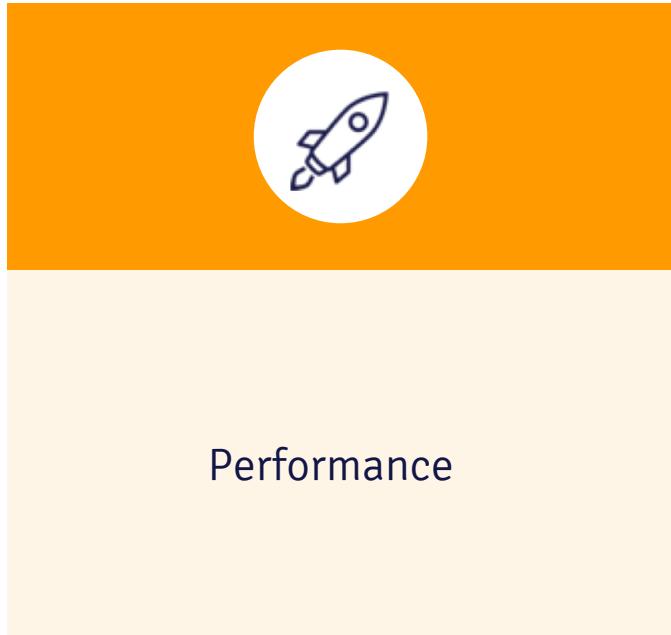


# What is Multi-Tenancy?



# Introduction

**Get the right balance between...**



# Introduction

Get the right balance between...



Performance



Isolation



# Introduction

Get the right balance between...



Performance



Isolation



Cost



# Introduction

**Get the right balance between...**



High Availability



# Introduction

Get the right balance between...



High Availability



Manageability



# Introduction

Get the right balance between...



High Availability



Manageability



Customer Confidence





# Background



You're the DBA for a popular local  
bakery Hooray for Cake  
<https://www.facebook.com/HoorayforCakeHarlow>



# Background



You're the DBA for a popular local bakery Hooray for Cake

<https://www.facebook.com/HoorayforCakeHarlow>



Hooray for Cake is growing fast and will soon have several franchises



# Background



You're the DBA for a popular local bakery Hooray for Cake  
<https://www.facebook.com/HoorayforCakeHarlow>



Hooray for Cake is growing fast and will soon have several franchises



Each franchise must access only its own data



# Background



You're the DBA for a popular local bakery Hooray for Cake  
<https://www.facebook.com/HoorayforCakeHarlow>



Hooray for Cake is growing fast and will soon have several franchises



Each franchise must access only its own data



You need to plan how to organise your Postgres databases



# Background

## Customers

Name	Address	Contact Name	Contact Email
The Little Coffee Shop	1 Main Road	Ms Manager	<a href="mailto:mgr@littlecoffeeshop.com">mgr@littlecoffeeshop.com</a>
Delicious Treats	4 High Street	Mr Owner	<a href="mailto:ownr@delicioustreats.com">ownr@delicioustreats.com</a>
Best Café	3 Central Square	Mx Bigboss	<a href="mailto:boss@bestcafe.com">boss@bestcafe.com</a>

## Orders

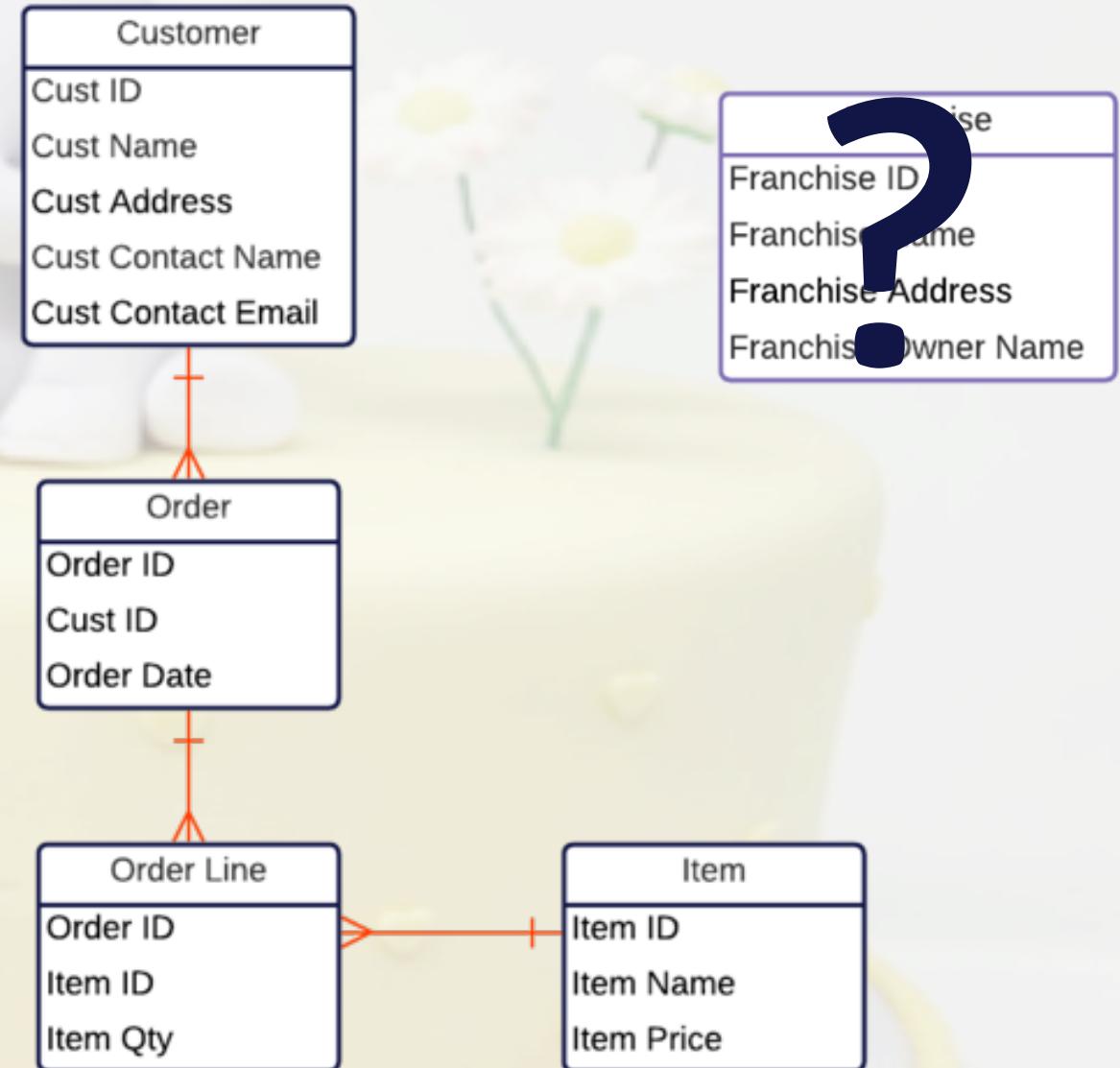
Order Date	Invoice#	Customer	Item	Quantity	Item Total	Order Total
01-Jan-22	1	Delicious Treats	Lemon drizzle loaf	2.00 €	20.00 €	20.00 €
15-Jan-22	2	The Little Coffee Shop	Salted caramel cupcake	15.00 €	37.50 €	61.50 €
			Chocolate chip cookie	24.00 €	24.00 €	
15-Feb-22	3	Delicious Treats	Chocolate chip cookie	12.00 €	12.00 €	42.00 €
			Lemon drizzle loaf	3.00 €	30.00 €	
10-Mar-22	4	Best Café	Salted caramel cupcake	20.00 €	50.00 €	50.00 €

## Price List

Item	Price
Chocolate chip cookie	1.00 €
Lemon drizzle loaf	10.00 €
Salted caramel cupcake	2.50 €
Birthday cake	25.00 €



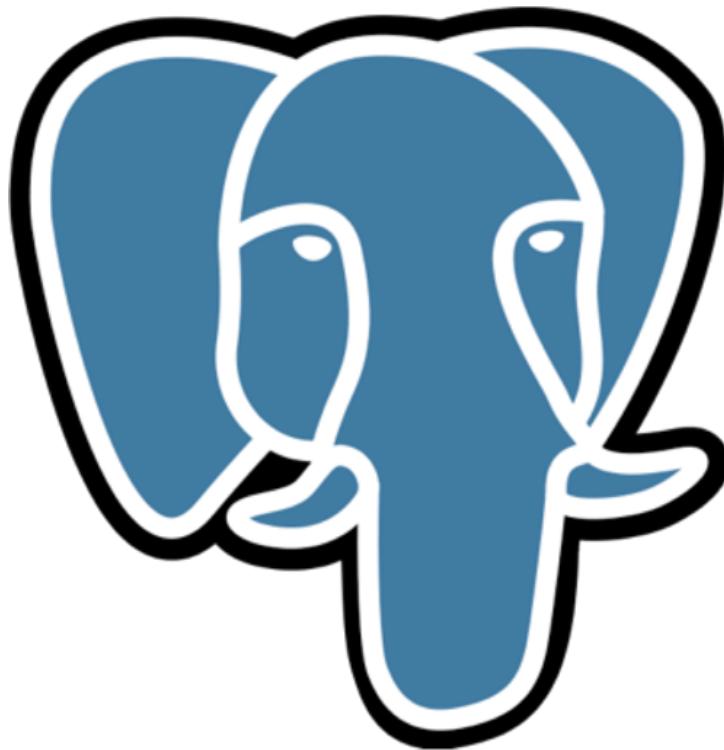
# Prepare to Create a Database

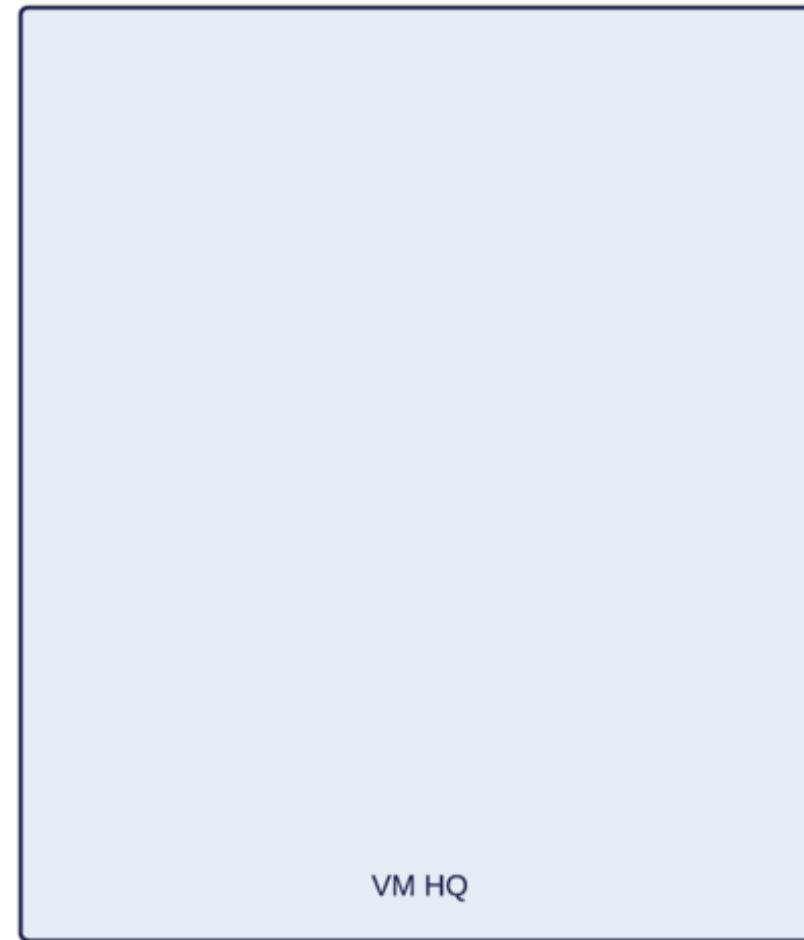


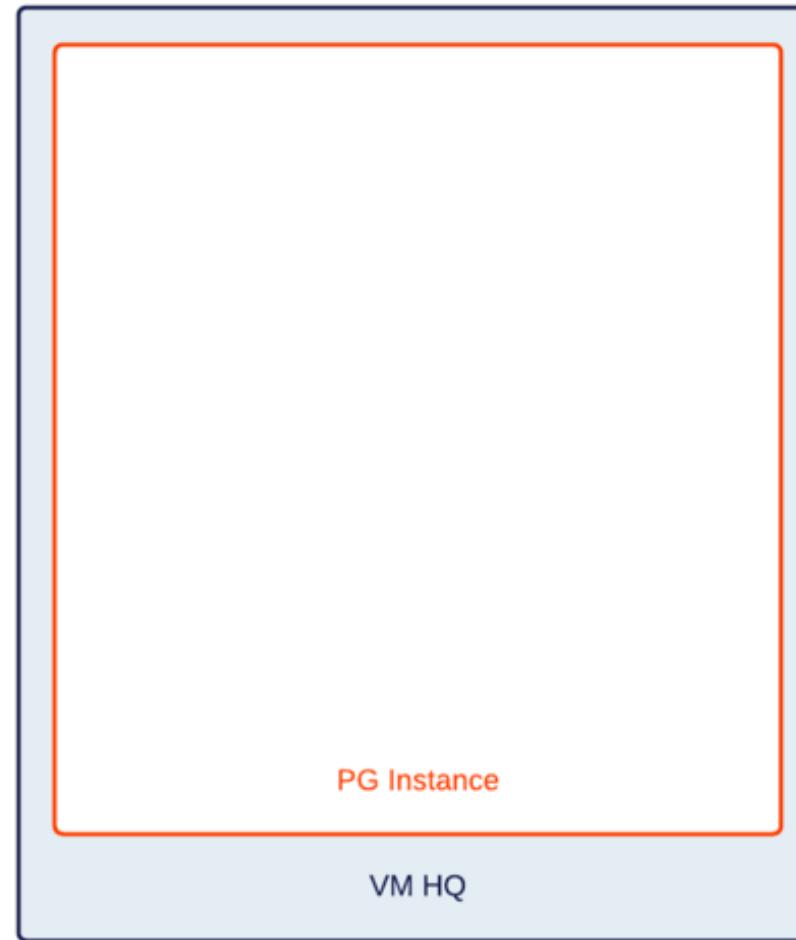
A soft-focus background image of a white dog sitting on a grassy hill. The dog is facing forward, looking slightly down. To its right is a small cluster of yellow flowers with green stems.

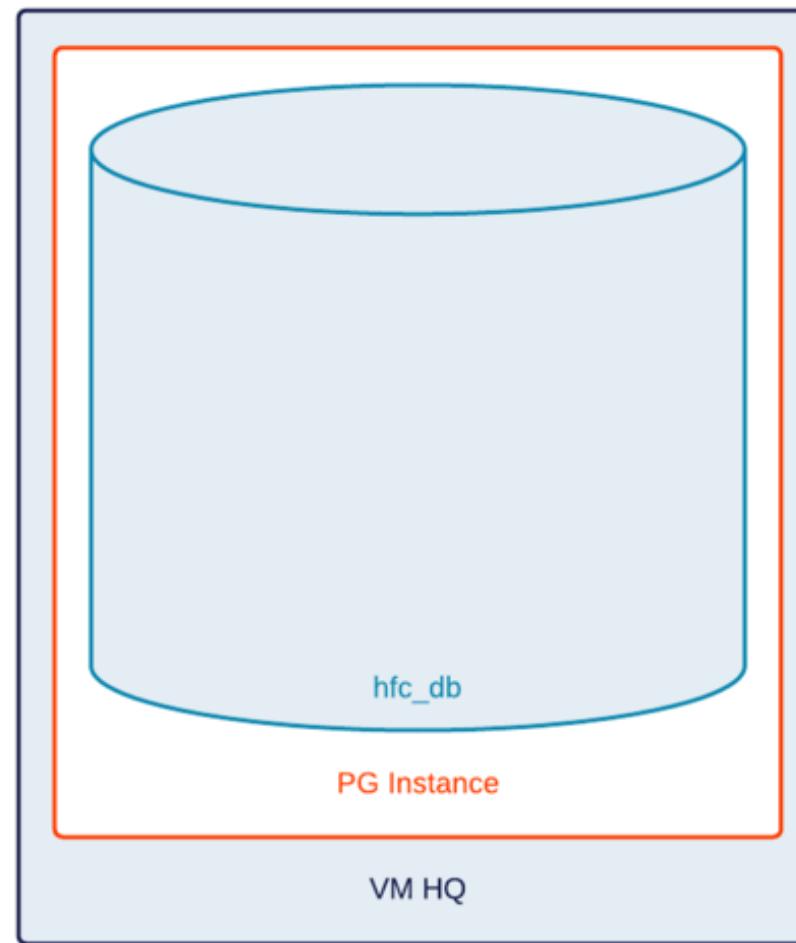
# Now what?

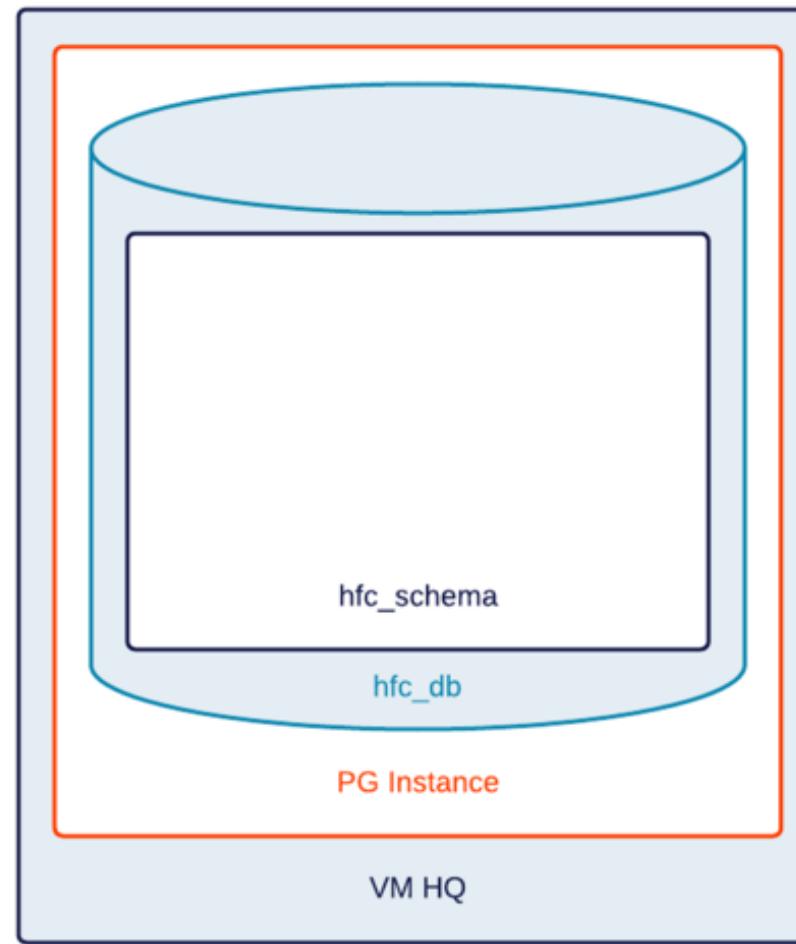


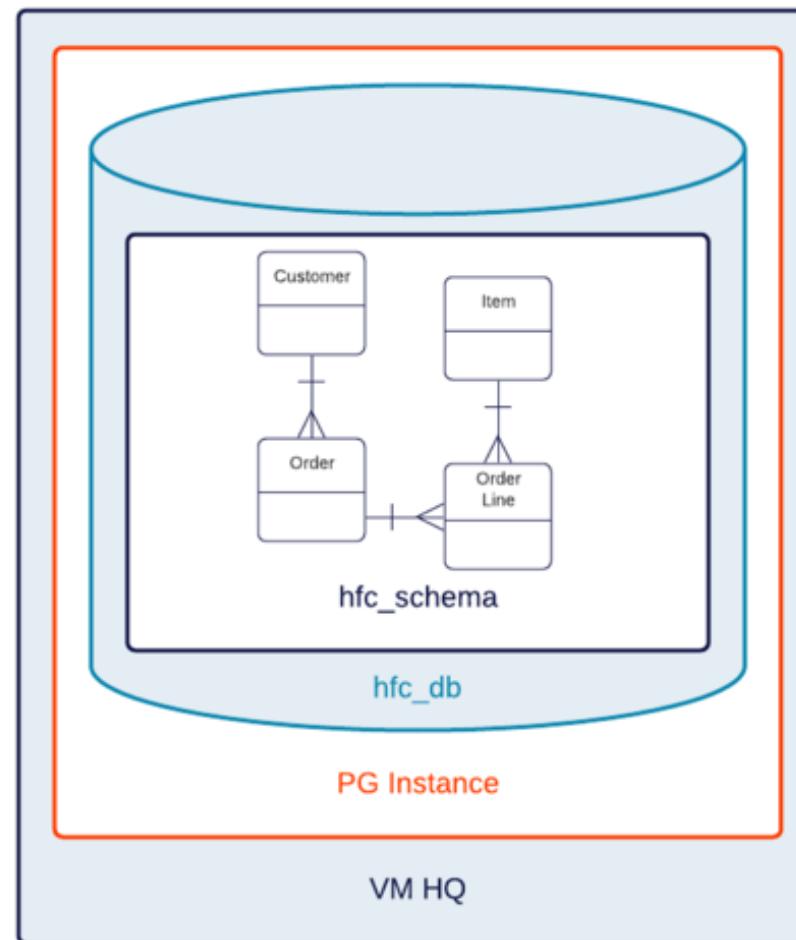












# What are the different options?



# What are the different options?



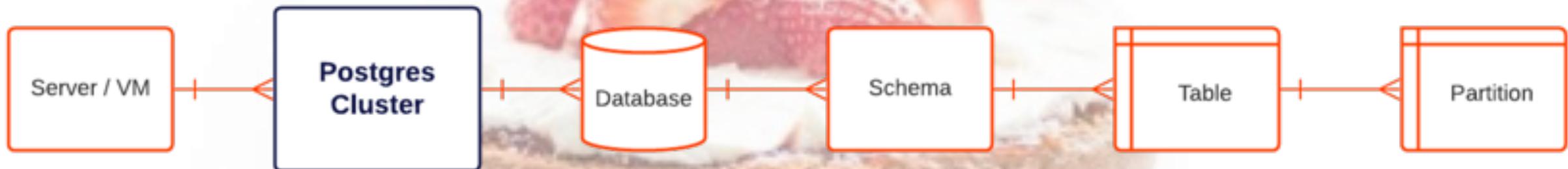
# What are the different options?



# What are the different options?



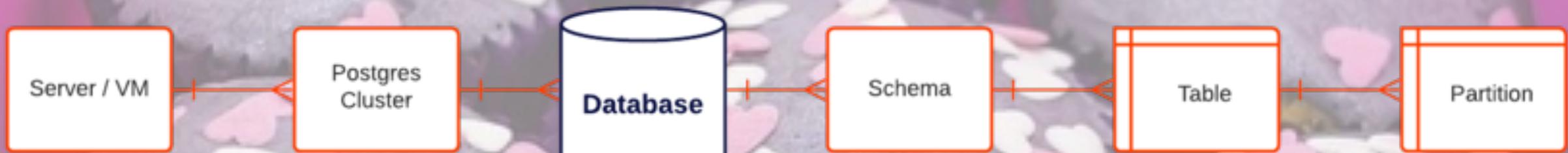
# What are the different options?



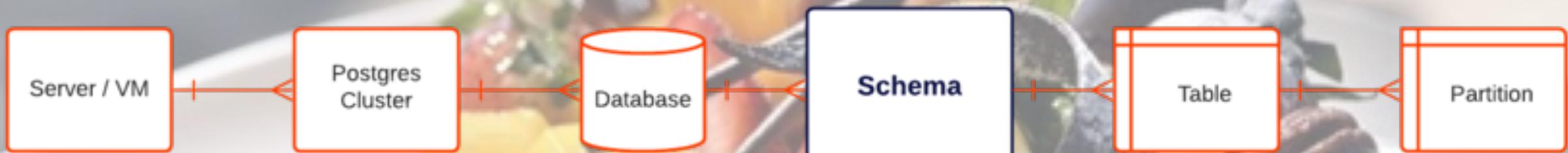
# What are the different options?



# What are the different options?



# What are the different options?



# What are the different options?



# What are the different options?



# What are the different options?



# What are the different options?



# What are the different options?



# What are the different options?

Table

Shard



# What are the different options?



# What are the different options?



# What are the different options?



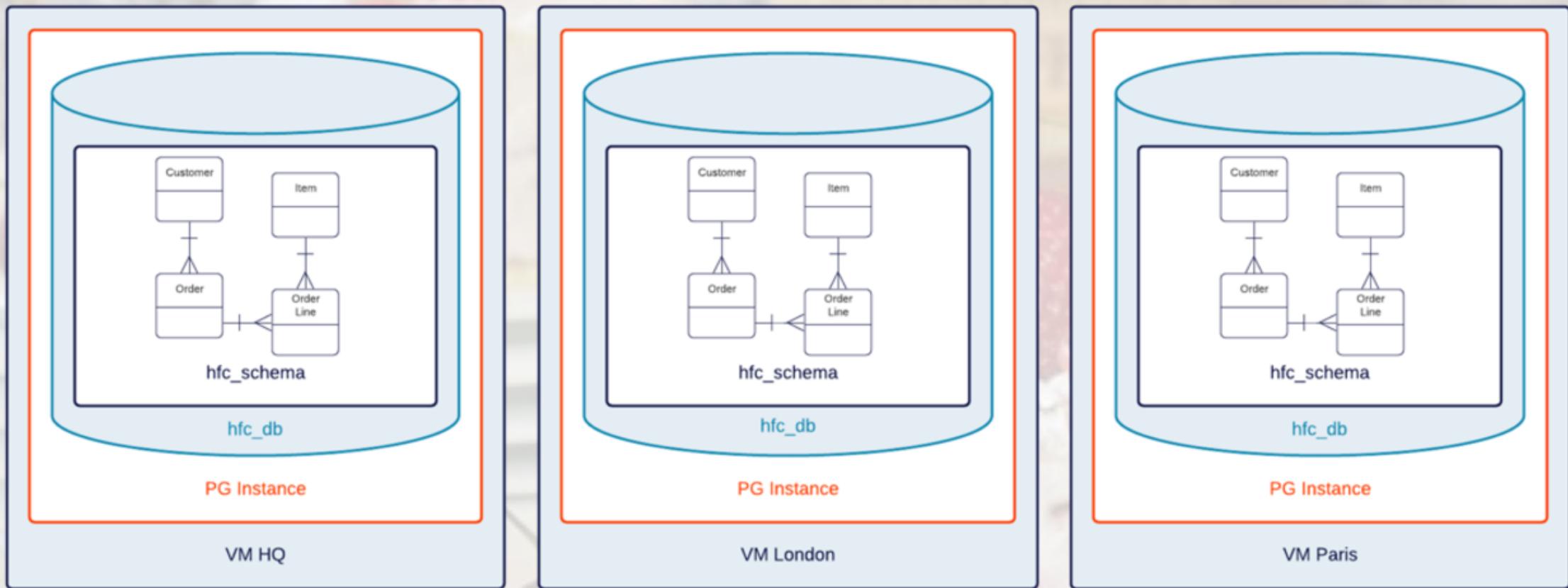
# **How do you decide?**



# One VM Per Franchise



# Architecture



# Setup

## Create VMs

- VM HQ
- VM London
- VM Paris

## Create Database

- In each cluster:
  - `CREATE DATABASE hfc_db;`

## Initialise Clusters

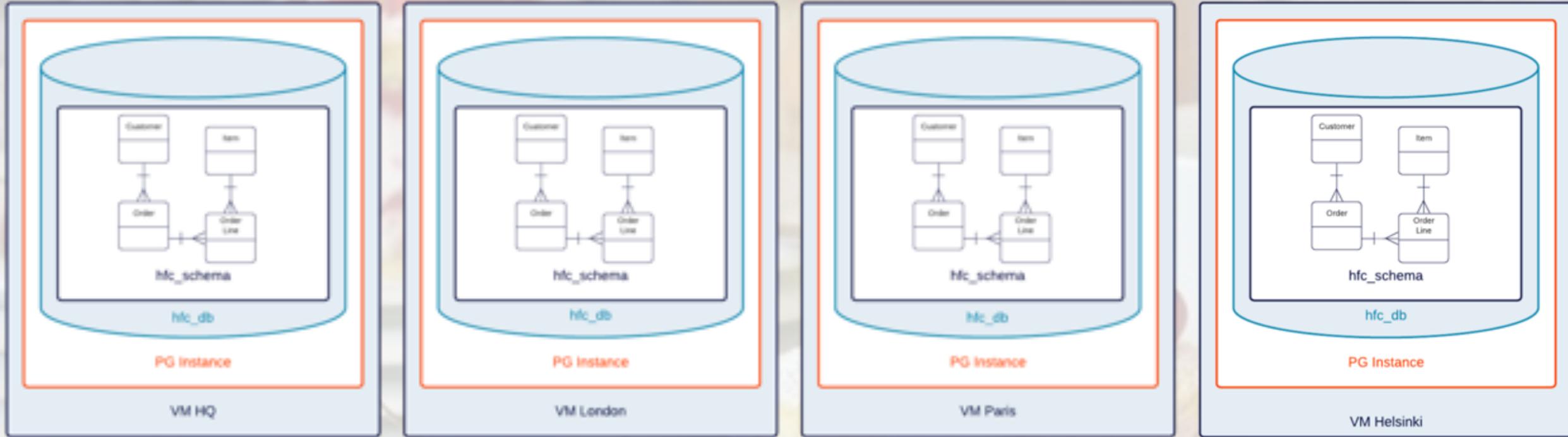
- `initdb` on each of the VMs to create PG Cluster
- Start Postgres

## Create Schema and Tables

- In `hfc_db` in each cluster:
  - `CREATE SCHEMA hfc_schema;`
  - `CREATE TABLE customer;`
  - `CREATE TABLE order...`



# Adding a Franchise



# Adding a Franchise

## Create VM

- VM Helsinki

## Create Database

- In new cluster:
  - CREATE DATABASE hfc\_db;

## Initialise Cluster

- initdb on new VM to create PG Cluster
- Start Postgres

## Create Schema and Tables

- In hfc\_db in new cluster:
  - CREATE SCHEMA hfc\_schema;
  - CREATE TABLE customer;
  - CREATE TABLE order...



# Advantages/Disadvantages

## Advantages

- Resources can be managed individually
- Upgrade one franchise
- Performance of one franchise does not impact the others
- Geographical Distribution
- Individual Backup/Restore

## Disadvantages

- Complex to report on all data
- Complex to manage shared reference data
- High Maintenance

## “It Depends”

- Spare capacity **can't** be shared between franchises





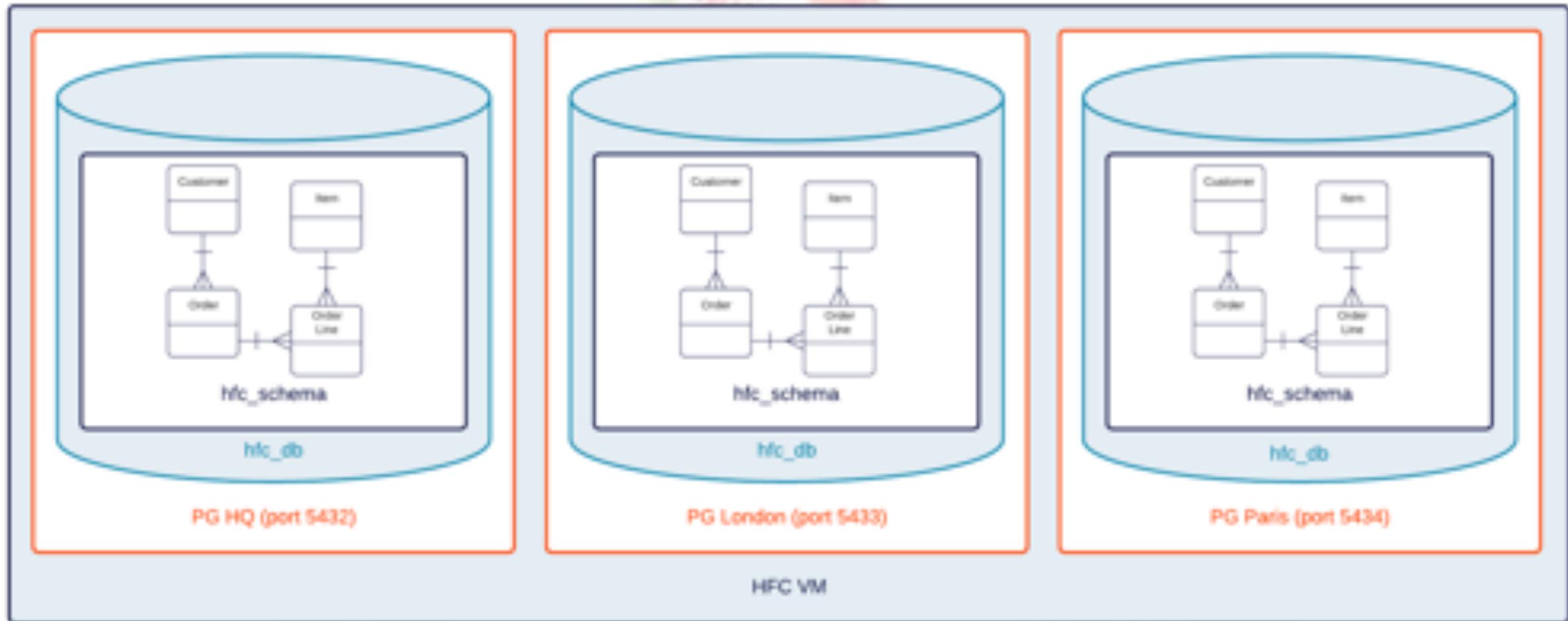
# When might this be useful?



# Single VM with a Postgres Cluster Per Franchise



# Architecture



# Setup

## Create VM

- HFC VM

## Create Database

- In each Cluster:
  - `CREATE DATABASE hfc_db;`

## Initialise Clusters

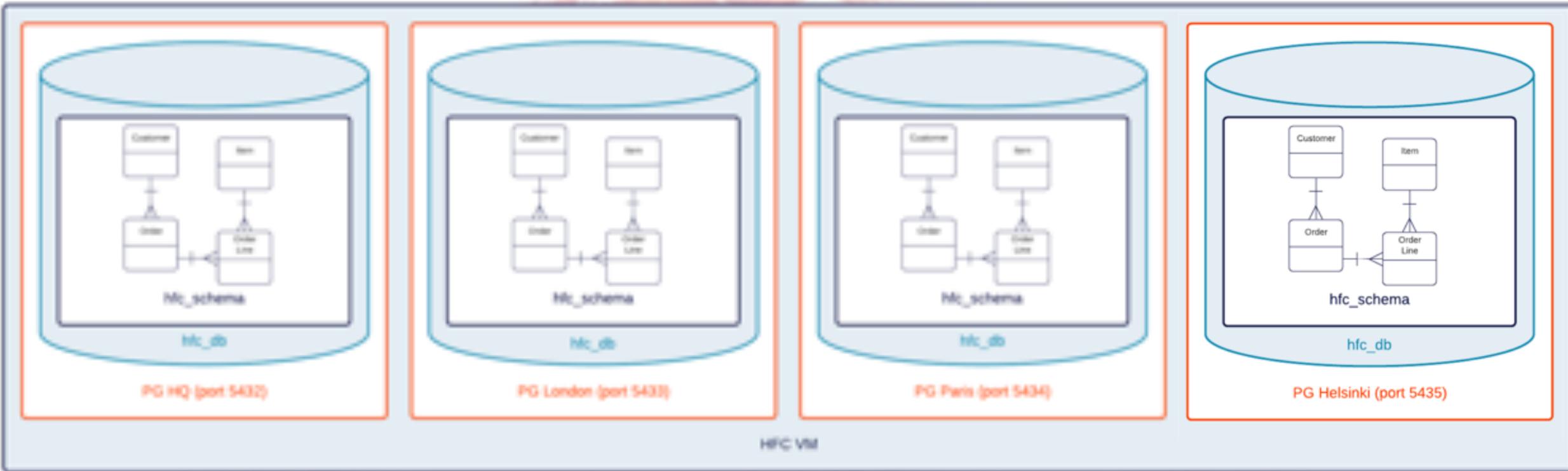
- For each franchise:
  - `initdb` to create named cluster on separate port
  - Start Postgres

## Create Schema and Tables

- In `hfc_db` on each cluster:
  - `CREATE SCHEMA hfc_schema;`
  - `CREATE TABLE customer;`
  - `CREATE TABLE order...`



# Adding a Franchise



# Adding a Franchise

## Create VM

- HFC VM

## Create Database

- On new instance:
  - `CREATE DATABASE hfc_db;`

## Initialise Cluster

- For new franchise:
  - `initdb` to create named cluster on separate port
  - Start Postgres

## Create Schema and Tables

- In `hfc_db` on new instance:
  - `CREATE SCHEMA hfc_schema;`
  - `CREATE TABLE customer;`
  - `CREATE TABLE order...`



# Advantages/Disadvantages

## Advantages

- Can manage Postgres parameters individually
- Postgres or schema upgrade for one franchise
- Just one VM to manage
- Individual Backup/Restore

## Disadvantages

- Performance of one franchise may impact others
- OS resources can't be managed individually
- OS upgrade affects all franchises
- Complex to report on all data
- Complex to manage shared reference data

## “It Depends”

- Spare capacity **can** be shared between franchises



A large, semi-transparent image of a strawberry shortcake dessert occupies the background. The dessert consists of several layers of light-colored sponge cake, white whipped cream, and fresh strawberries. Some strawberries have their green leafy tops still attached. The dessert is presented on a white plate.

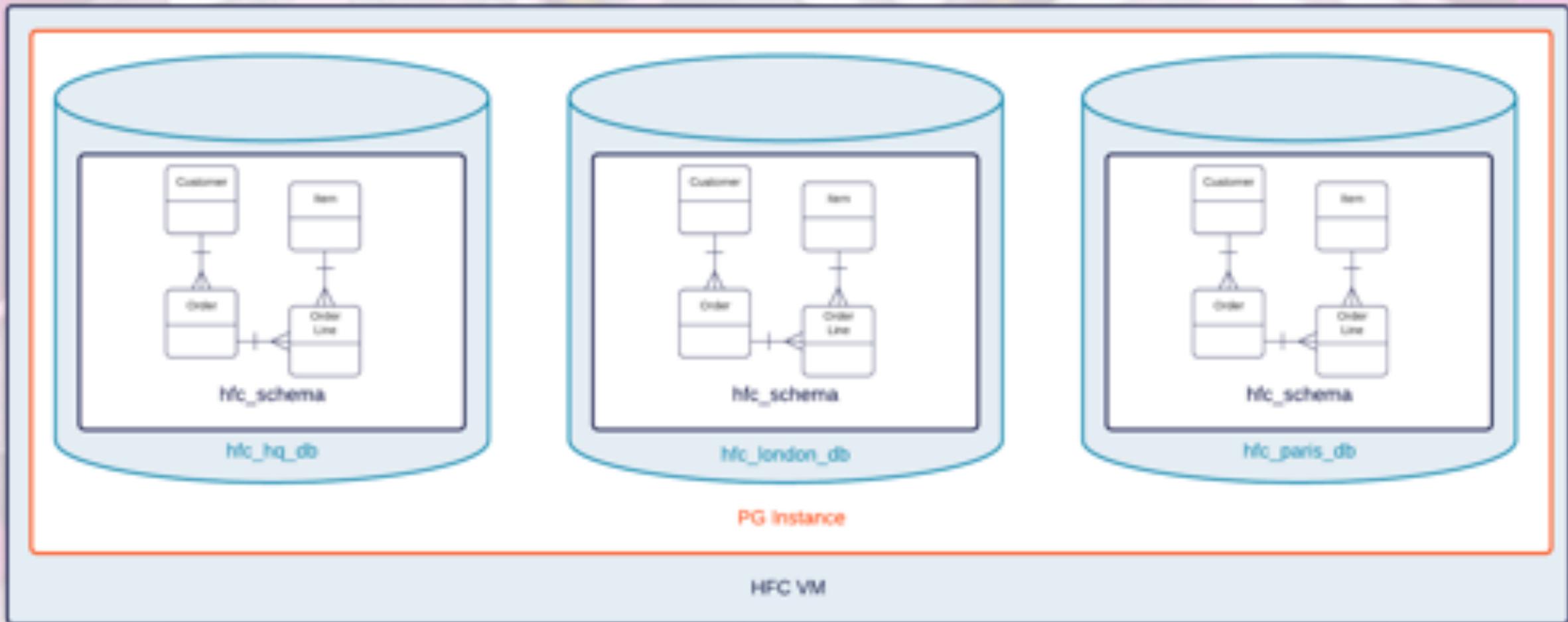
# When might this be useful?



# Single Cluster with a Database Per Franchise



# Architecture



# Setup

## Create VM

- HFC VM

## Create Databases

- CREATE DATABASE hfc\_hq\_db;
- CREATE DATABASE hfc\_london\_db;
- CREATE DATABASE hfc\_paris\_db;

## Initialise Cluster

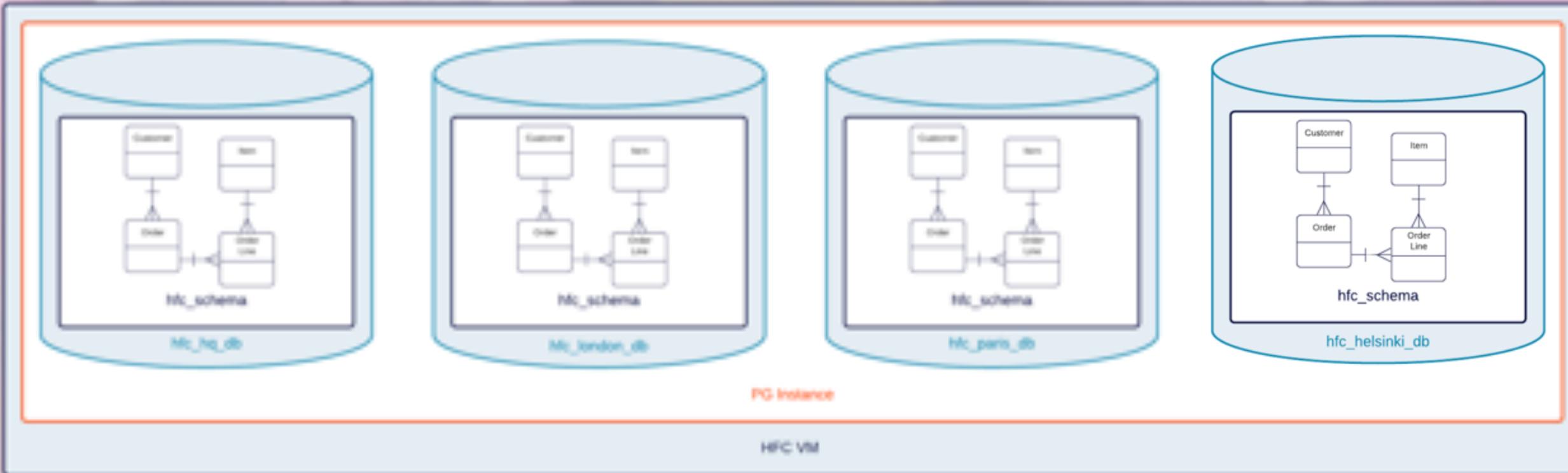
- initdb on the HFC VM to create PG Cluster

## Create Schema and Tables

- In each database:
  - CREATE SCHEMA hfc\_schema;
  - CREATE TABLE customer;
  - CREATE TABLE order...



# Adding a Franchise



# Adding a Franchise

## Create VM

- HFC VM

## Create Database

- `CREATE DATABASE hfc_helsinki_db;`

## Initialise Cluster

- `initdb` on the HFC VM to create PG Cluster
- Start Postgres

## Create Schema and Tables

- In new database:
  - `CREATE SCHEMA hfc_schema;`
  - `CREATE TABLE customer;`
  - `CREATE TABLE order...`



# Advantages/Disadvantages

## Advantages

- Just one VM to manage
- Individual schema upgrades

## Disadvantages

- OS or PG upgrade affects all franchises
- Performance of one franchise may impact others
- OS resources can't be managed individually
- PG configuration can't be managed individually
- Complex to report over all data
- Complex to manage shared reference data
- Backup/Restore can't be performed individually

## “It Depends”

- Spare capacity **can** be shared between franchises





# When might this be useful?



# Single Database with a Schema Per Franchise



# Architecture



# Setup

## Create VM

- HFC VM

## Initialise Cluster and Create Database

- initdb on the HFC VM to create PG Cluster
- Start Postgres
- CREATE DATABASE hfc\_db;

## Create Schemas

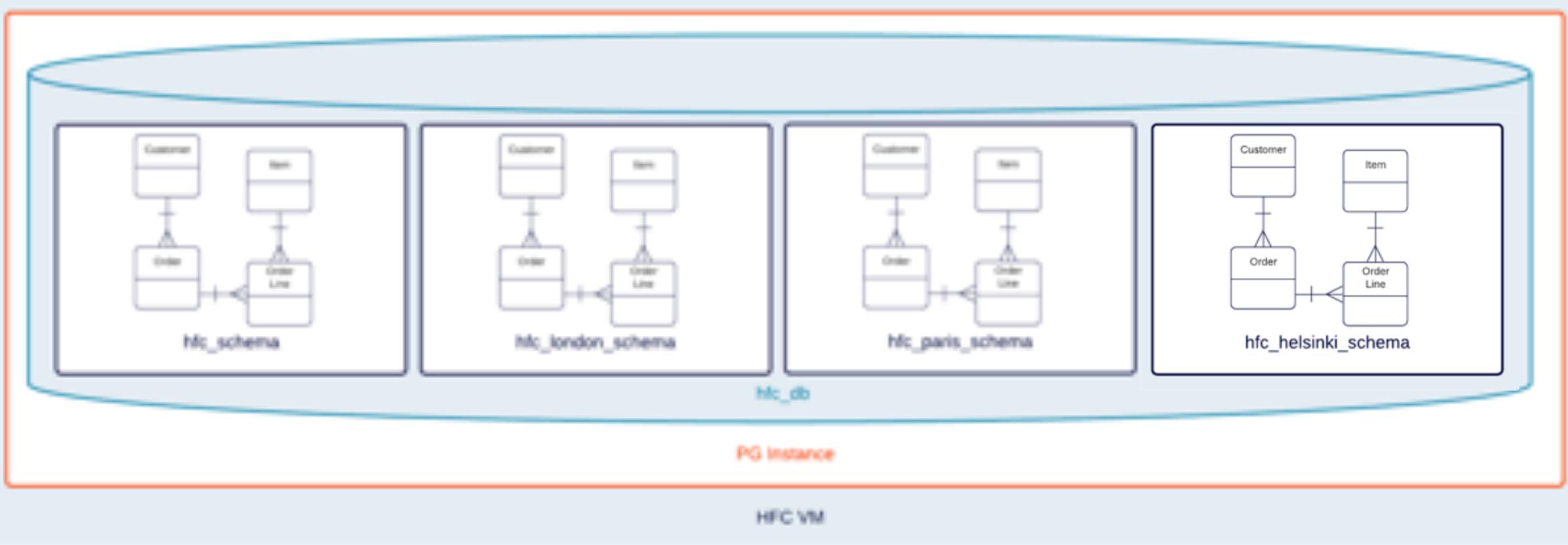
- CREATE SCHEMA hfc\_hq\_schema;
- CREATE SCHEMA hfc\_london\_schema;
- CREATE SCHEMA hfc\_paris\_schema;

## Create Tables

- In each schema:
  - CREATE TABLE schemaname.customer;
  - CREATE TABLE schemaname.order
  - CREATE TABLE schemaname.order\_line
  - CREATE TABLE schemaname.item



# Adding a Franchise



# Adding a Franchise

## Create VM

- HFC VM

## Create Schema

- CREATE SCHEMA hfc\_helsinki\_schema;

## Initialise Cluster and Create Database

- initdb on the HFC VM to create PG Cluster
- Start Postgres
- CREATE DATABASE hfc\_db;

## Create Tables

- CREATE TABLE hfc\_Helsinki\_schema.customer;
- CREATE TABLE hfc\_Helsinki\_schema.order;
- CREATE TABLE hfc\_Helsinki\_schema.order\_line;
- CREATE TABLE hfc\_Helsinki\_schema.item;



# Advantages/Disadvantages

## Advantages

- Just one VM to manage
- Individual schema upgrades
- Can report over all data
- Easy to manage shared reference data

## Disadvantages

- OS or PG upgrade affects all franchises
- Performance of one franchise may impact the others
- OS resources can't be managed individually
- PG configuration can't be customized individually
- Backup/Restore can't be performed individually

## “It Depends”

- Spare capacity **can** be shared between franchises



A blurred background image of a fruit salad in a white bowl. The salad contains various fruits like grapes, melon, and berries, with a fork resting on top.

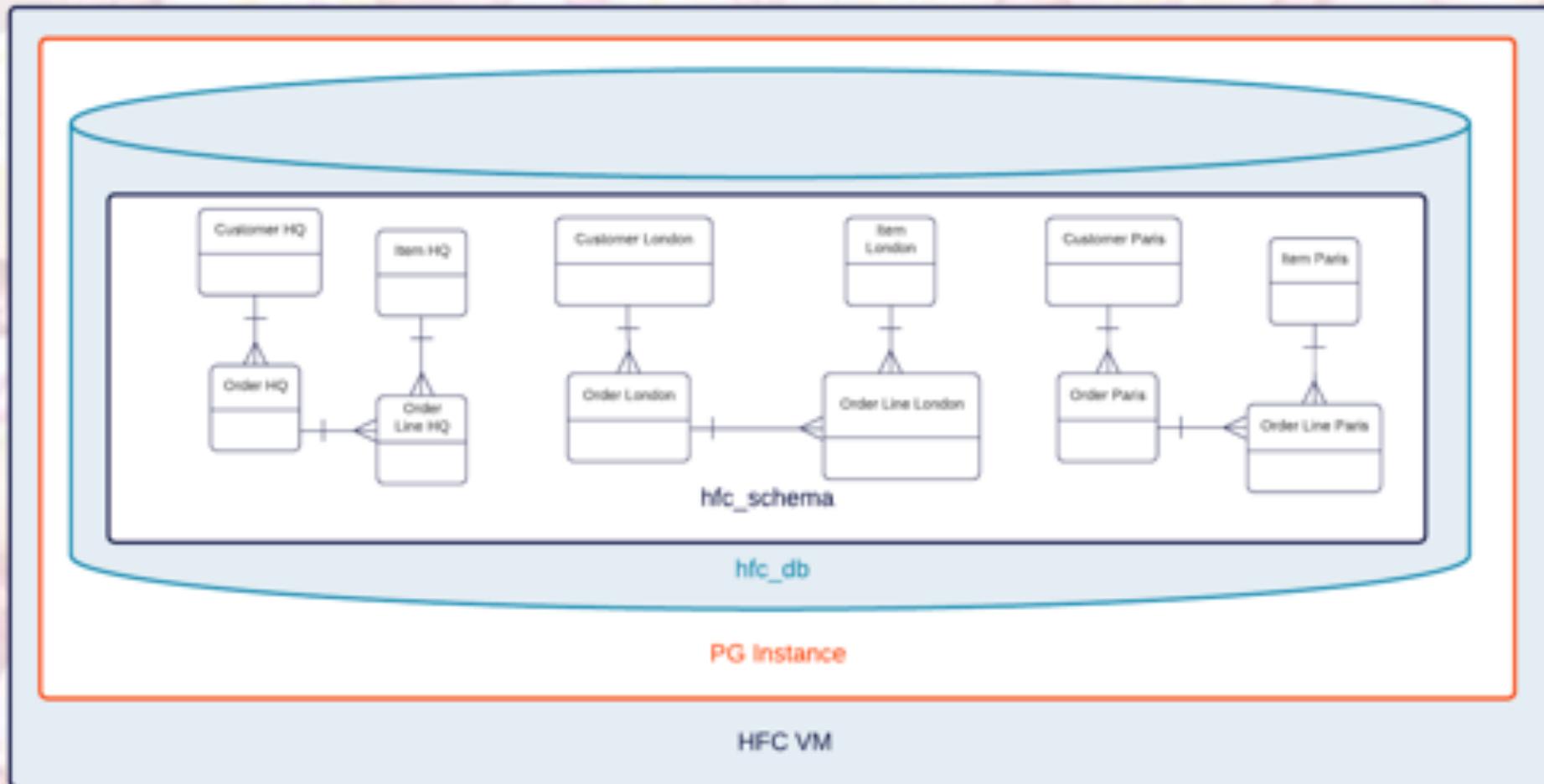
# When might this be useful?



# Single Database Schema with Tables Per Franchise



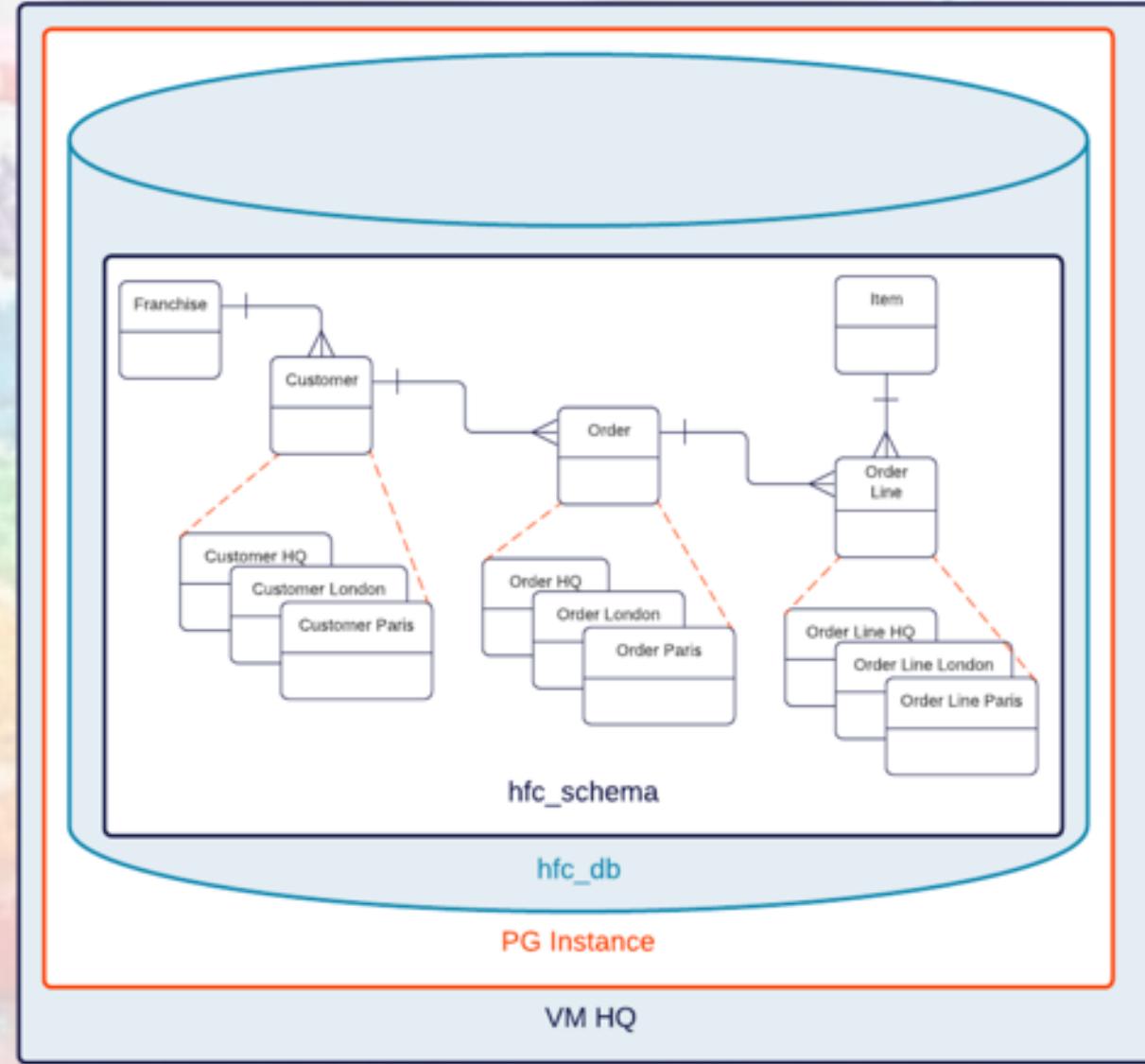
# Architecture



# Single Set of Tables, Partitioned by Franchise



# Architecture



# Setup

## Create VM and Initialise Cluster

- HFC VM
- initdb to create PG Cluster
- Start Postgres

## Create Database and Schema

- CREATE DATABASE hfc\_db;
- CREATE SCHEMA hfc\_schema;

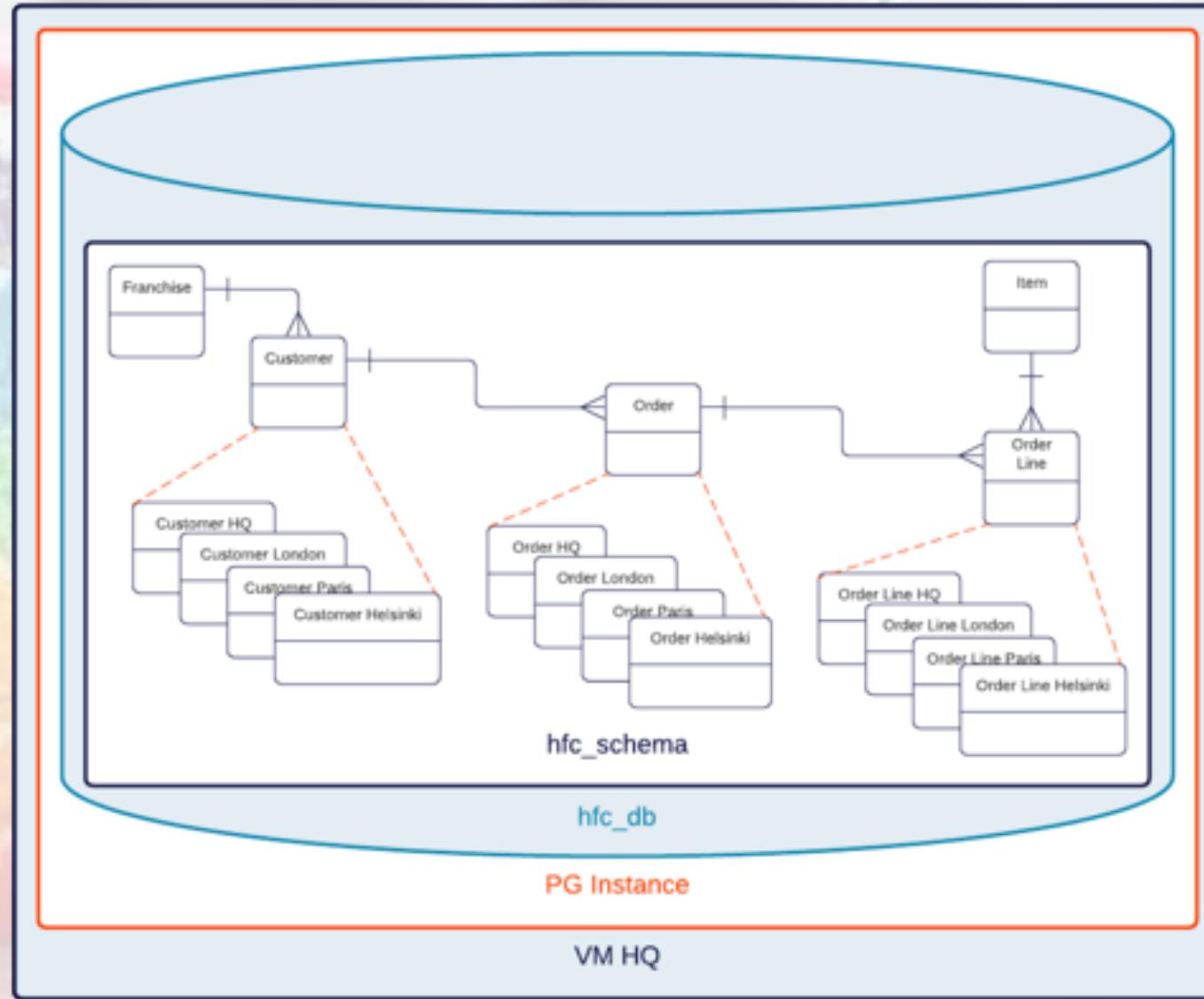
## Create Tables

- CREATE TABLE customer ... PARTITION BY LIST (franchise);
- CREATE TABLE order ... PARTITION BY LIST (franchise);
- CREATE TABLE order\_line ... PARTITION BY LIST (franchise);
- CREATE TABLE item

## Create Partitions

- CREATE TABLE customer\_hq  
PARTITION OF customer FOR VALUES IN ('HQ');
- CREATE TABLE customer\_london  
PARTITION OF customer FOR VALUES IN ('London');
- CREATE TABLE customer\_paris  
PARTITION OF customer FOR VALUES IN ('Paris');

# Adding a Franchise



# Adding a Franchise

## Create VM and Initialise Cluster

- HFC VM
- initdb to create PG Cluster
- Start Postgres

## Create Database and Schema

- CREATE DATABASE hfc\_db;
- CREATE SCHEMA hfc\_schema;

## Create Tables

- CREATE TABLE customer  
PARTITION BY LIST (franchise);
- CREATE TABLE order  
PARTITION BY LIST (franchise);
- ...

## Create Partitions

- CREATE TABLE customer\_helsinki  
PARTITION OF customer FOR VALUES IN ('Helsinki');
- CREATE TABLE order\_helsinki  
PARTITION OF customer FOR VALUES IN ('Helsinki');
- ...



# Advantages/Disadvantages

## Advantages

- Just one VM to manage
- Easy to report over all data
- Easy to manage shared reference data

## Disadvantages

- OS, PG or schema upgrade affects all franchises
- Performance of one franchise may impact the others
- OS resources can't be managed individually
- PG configuration can't be customized individually
- Backup/Restore can't be performed individually

## “It Depends”

- Spare capacity **can** be shared between franchises



# Row Level Security

ALTER TABLE . . . ENABLE ROW LEVEL SECURITY

<https://www.postgresql.org/docs/14/ddl-rowsecurity.html>





# When might this be useful?



# Sharding



# “Bake it Yourself” Implementation



# “Ready Baked” Implementation



Using a Postgres extension such as Citus

# Architecture



# Architecture



# Setup

## Create VMs

- VM Control
- VM HQ
- VM London
- VM Paris

## Initialise Clusters

- on each of the VMs:
  - initdb to create PG Cluster
  - Start Postgres

## Create Database and Schema

- In each cluster
  - CREATE DATABASE hfc\_db;
  - CREATE SCHEMA hfc\_schema;

## Create Tables on VM Control

- CREATE TABLE customer  
PARTITION BY LIST (franchise);
- CREATE TABLE order  
PARTITION BY LIST (franchise);
- ...



# Setup (continued)

## Create Shards

- In the database for each franchise (eg for London):
  - `CREATE TABLE customer_london  
PARTITION OF customer FOR VALUES IN ('London');`
  - `CREATE TABLE order_london  
PARTITION OF order FOR VALUES IN ('London');`
  - ...

## Create FDWs

- On the control hfc\_db database, for each franchise database (eg for London):
  - `CREATE SERVER hfc_london  
FOREIGN DATA WRAPPER postgres_fdw...;`
  - `CREATE USER MAPPING FOR ... SERVER hfc_London...;`

## Setup Control Database

- On the control hfc\_db database:
  - `CREATE EXTENSION postgres_fdw;`

## Create Foreign Tables

- For each franchise (eg for London):
  - `CREATE FOREIGN TABLE customer_london  
PARTITION OF customer FOR VALUES IN ('London')  
SERVER hfc_london;`
  - `CREATE FOREIGN TABLE order_london  
PARTITION OF order FOR VALUES IN ('London')  
SERVER hfc_london;`
  - ...

# Advantages/Disadvantages

## Advantages

- Resources can be managed individually
- Upgrade individual franchise (OS or Postgres)
- Performance of one franchise does not impact the others
- Geographical distribution
- Backup/Restore individual franchise
- Can easily report over all data

## Disadvantages

- Maintenance increases proportionally to the number of franchises
- Schema upgrade affects all franchises
- Complex to manage shared reference data

## “It Depends”

- Spare capacity **can't** be shared between franchises



# When might this be useful?

# Summary of Main Options

	Manage Resources Individually		Shared Reference Data	Cross-Franchise Reporting	Geographical Distribution	Upgrade Individual Franchise			Backup/Restore Individual Franchise	Level of Isolation of Franchises	Complexity of Adding Franchise
Level of Separation	OS	PG				OS	PG	Schema			
VM											
Cluster											
Database											
Schema											
Partition											
Shard											



# When to Use Multiple Clusters, Databases, or Schemas?

<https://www.enterprisedb.com/blog/when-use-multiple-clusters-databases-or-schemas>

Bruce Momjian 23/04/2012

Feature	Cluster	Database	Schema
Isolated Server Start/Stop	✓		
Connection Control	✓	✓	
Private System Tables	✓	✓	&nbsp;
Private Plug-Ins	✓	✓	
Isolated Administration	✓		
Shared Administration		✓	✓
Isolated Resource Usage	✓		
Shared Resource Usage(1)		✓	✓
Data Isolation(2)	✓	✓	
Cross-Container Queries			✓

(1) A large number of data containers increases the usefulness of resource sharing, e.g. shared\_buffers. Resource sharing includes log shipping and streaming replication sharing.

(2) User and database names, being global objects, are visible in all databases. It is impossible to query across databases, except via an external database session, e.g. dblink. Schema permissions allow data access control, but pg\_class still shows all tables defined in the database.



# Making the Decision



# Considerations

- Who manages the infrastructure?
- Who manages the schema?
- Who manages the data?
- Is there shared reference data?
- Do we need reporting across all data?
- If proprietary software is used, what is the licensing model?
- Is it important to have the data geographically distributed?
- How easy is it to create a new customer/remove an old one?
- What level of isolation is needed?
- Is it OK/desirable to perform OS/PG/Schema upgrades at the same time for all?
- Are backups/restores for individual franchises required?



# Final Decision



# Final Decision

**Single Set of Tables, Partitioned by Franchise**



# pgDay Paris Survey

Receive a gift card!

<https://bit.ly/PGDayParis>



Questions?



**Thank You!**

