DISTRIBUTED SYSTEMS

Distributed Databases

Group 6

About Us

Group 6

Nguyễn Trấn Nguyên	BI10-128
Nguyễn Tương Quỳnh	BI10-154
Nguyễn Quang Vinh	BI10-195
Nguyễn Anh Quân	BI10-146
Mai Xuân Hiếu	BI10-064

CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1. What is Distributed Database?

1.2. The Types of Distributed Database

1.3. Distributed Data Storage

1.1. WHAT IS DISTRIBUTED DATABASE?

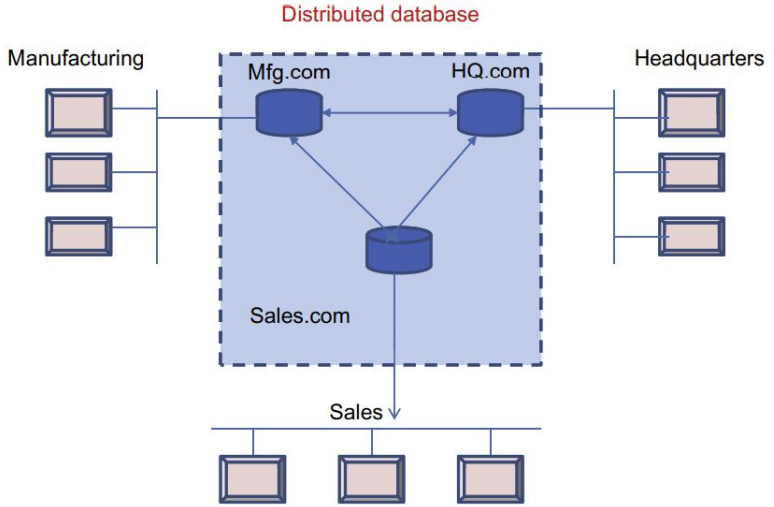
- A **distributed database** (DDB) is a collection of databases that are physically spread across multiple sites in a computer network.

- In DDB, the files must be **formatted**, logically **interconnected**, and physically **dispersed** over numerous sites to form a distributed database system. A common interface for accessing the distributed data is required.

- **Applications**: It is used in the Corporate Management Information System, multimedia applications, Military control systems, Hotel chains, also in the manufacturing control system, and many other systems.

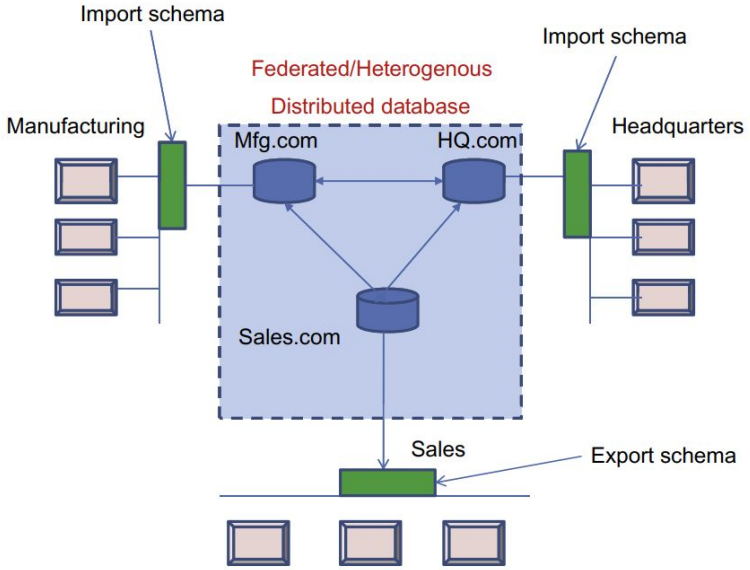
1.2. THE TYPES OF DISTRIBUTED DATABASE

- **Homogeneous Distributed Database:**
  - In a homogeneous database, all different sites store the database **identically**.
  - The operating system, database management system, and the data structures used – **all are the same** at all sites. Hence, they're **easy to manage**.
  - In our project, we would try to create a **homogeneous distributed database**



1.2. THE TYPES OF DISTRIBUTED DATABASE

- **Heterogeneous Distributed Database:**
  - In a heterogeneous distributed database, **different sites** can use **different schema** and **software**.
  - **Different computers** may use a **different operating system**, a **different database application**.
  - They may even use **different data models** for the database. Hence, **translations** are **required** for different sites to **communicate**.



1.3. DISTRIBUTED DATA STORAGE

- **Replication:**
  - In this approach, the entire relationship is **stored redundantly**. Hence, in replication, systems maintain **copies of data**.
  - This is advantageous as it increases the **availability** of data at different sites. Also, the query requests can be processed in parallel.
  - However, it has certain disadvantages as well.
    - **Data** needs to be **constantly updated**.
    - Also, **concurrency control** becomes way more complex as concurrent access now needs to be checked over a number of sites.

1.3. DISTRIBUTED DATA STORAGE

- **Fragmentation:**
  - In this approach, the **relations** are **fragmented** and each of the fragments is stored in different sites where they're required.
  - It must be made sure that the **fragments** are **able** to **reconstruct** the original relation (i.e, there isn't any loss of data).
  - Fragmentation is advantageous as it doesn't create copies of data, consistency is not a problem.
  - Fragmentation of relations can be done in **two ways:**
    - **Horizontal fragmentation – Splitting by rows:** The relation is fragmented into groups of tuples so that each tuple is assigned to at least one fragment.
    - **Vertical fragmentation – Splitting by columns:** The schema of the relation is divided into smaller schemas. Each fragment must contain a common candidate key so as to ensure a lossless join.
  - In certain cases, an approach that is a hybrid of fragmentation and replication is used.
- In our project, we choose **horizontal fragmentation** as our distributed data storage

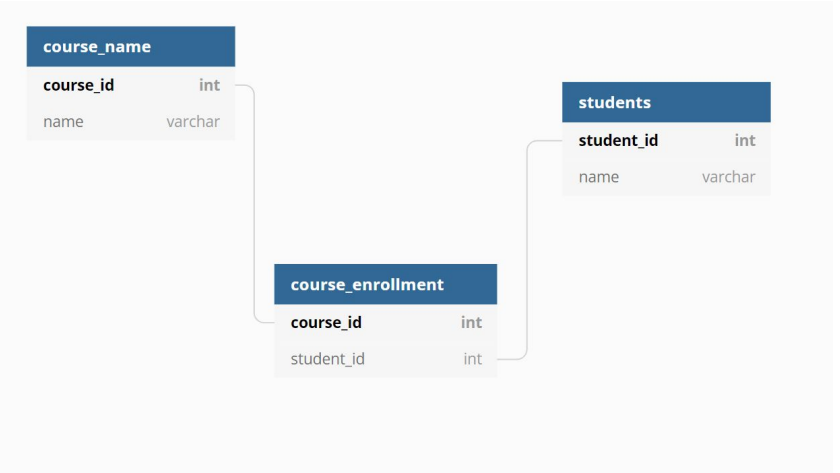
2. METHODOLOGY

2. METHODOLOGY

- We try to set up a distributed database to manage student course enrollment by **PostgreSQL** and **Citus**

+ **PostgreSQL** is a powerful, open source object-relational database system with a strong reputation for reliability and performance.

+ **Citus** is a PostgreSQL **extension** that allows commodity database servers (called *nodes*) to coordinate with one another in a “shared nothing” architecture. This architecture allows the database to scale by simply adding more nodes to the cluster.



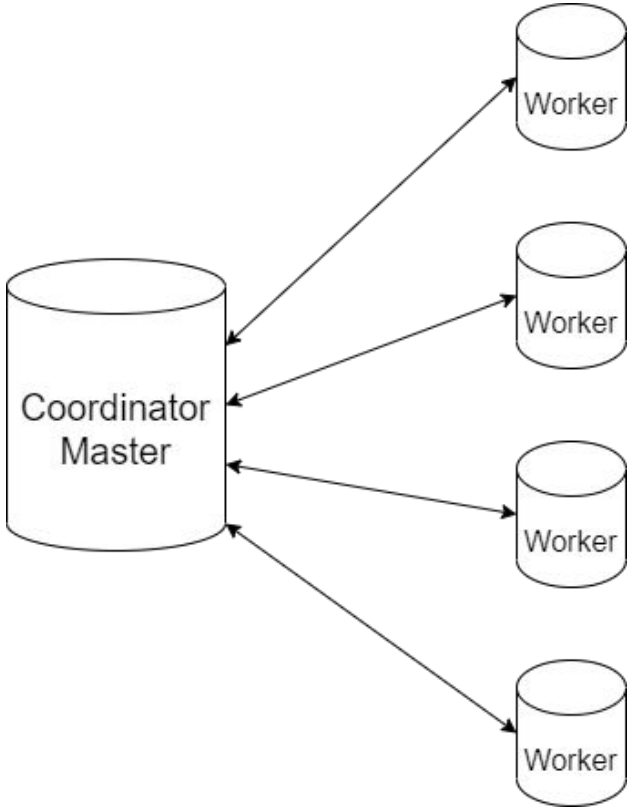
2. METHODOLOGY

- **Database Architecture:**

+ In total, we have 5 database.

+ These small database instances are easier to manage as most of the data exists in the separate database worker servers.

+ Coordinators hold smaller amounts of data, like some metadata and data that is not sensible to shard.



3. DEMO

4. CONCLUSION AND FUTURE WORKS

4. CONCLUSION AND FUTURE WORKS

- In conclusion, we have successfully set up a distributed database to manage student course enrollment by **PostgreSQL** and **Citus**
- However, there are a lot of **improvements** that can be done in the future, such as the larger database or more workers/replica in the architecture.

REFERENCE

- https://docs.citusdata.com/en/stable/get_started/concepts.html#table-types
- https://en.wikipedia.org/wiki/Distributed_database
- https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_2492
- https://www.tutorialspoint.com/DDBMS-Advantages-and-Disadvantages#:~:text=The%20database%20is%20easier%20to,be%20stored%20at%20different%20locations.
- https://www.youtube.com/watch?v=J-sj3GUrq9k&ab_channel=DatabasesDemystified
- https://www.geeksforgeeks.org/difference-between-centralized-database-and-distributed-database/
- Charles Tupper (2011) 'Distributed Databases' in *Data Architecture: From Zen to Reality*. Elsevier: p385-400
- https://www.smartly.io/blog/scaling-our-analytical-processing-service-sharding-a-postgresql-database-with-citus

THANK YOU FOR LISTENING!