

# ISP\_Part5

---

## Overview

This part implements a distributed NoSQL database system using Apache Cassandra. It focuses on understanding data storage and retrieval processes within the context of inventory and supply chain management. The system is designed to manage products, warehouses, suppliers, orders, and related operations in a scalable and efficient manner.

## Problem Statement

The goal is to develop a NoSQL database system tailored to the specific needs of inventory and supply chain management. The system should efficiently handle various data types and relationships pertinent to this domain.

## Features

- **NoSQL Data Modeling:** Custom data schema and model designed for inventory and supply chain management.
- **CRUD Operations:** Implementation of Create, Read, Update, Delete operations for managing data.
- **Data Retrieval and Queries:** Sample queries to showcase database functionality and data retrieval capabilities.

## Components

1. **Cassandra Database:** A distributed NoSQL database used for storing and managing large amounts of data.
2. **Docker:** Containerization of the Cassandra database for easy deployment and scalability.
3. **CQL Scripts:** Cassandra Query Language scripts for creating schema, inserting data, updating, and querying.

## Installation and Setup

1. **Docker and Docker Compose:** Ensure Docker and Docker Compose are installed on your system.
2. **Build and Run Containers:**
  - Use `docker build -t cassandra .` to build the Cassandra image.
  - Use `docker-compose up -d` to start the containers.
3. **Database Initialization:**
  - Execute `docker exec -i inventory-supply-chain-management-cassandra-1-1 cqlsh -f /usr/local/bin/main_schema.cql` to create the keyspace and tables.
  - Insert data using the provided CQL scripts or directly run `commands.sh` to run the entire project.
4. **Database Access:**
  - Use `docker exec -it inventory-supply-chain-management-cassandra-1-1 cqlsh` to access the Cassandra shell.
  - Use `docker exec -it inventory-supply-chain-management-cassandra-1-1 bash` to access the container shell.

## Usage

- **CRUD Operations:** Use the provided CQL scripts to perform CRUD operations on the database.
- **Data Retrieval:** Execute specific queries to retrieve data related to orders, products, warehouses, etc.

## Project Structure

- **Dockerfile:** Configuration for building the Cassandra Docker image.
- **docker-compose.yml:** Defines services, networks, and volumes for the Cassandra containers.
- **cql\_directory/:** Contains all CQL scripts for schema creation, data insertion, updating, and deletion.

## Sample Queries

- Retrieve details of all products.
- Get information about specific warehouses.
- Fetch orders for a particular customer.
- Update product prices and order statuses.

## Deliverables

- **Code/Scripts:** Dockerfile, docker-compose file, and CQL scripts.
- **Documentation:** This README file.
- **Snapshots:** Screenshots or snapshots demonstrating the functionality.

## Tools

- **Cassandra:** For database management.
- **Docker:** For containerization and deployment.
- **Bash Script:** For automating the setup and initialization process.

## Outputs and Screenshots

- **Docker Containers:** Cassandra containers running on Docker.

```

Inventory-Supply-Chain-Management — bash commands.sh — bash — sleep · bash commands.sh — 137x37
➔ Inventory-Supply-Chain-Management git:(main) bash commands.sh
Starting the Cassandra containers
[+] Building 1.0s (15/15) FINISHED                                docker:desktop-linux
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                  0.0s
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 643B                             0.0s
=> [internal] load metadata for docker.io/library/cassandra:latest 0.9s
=> [auth] library/cassandra:pull token for registry-1.docker.io 0.0s
=> [1/9] FROM docker.io/library/cassandra:latest@sha256:e7410759812c49ebb71740e40237158f115b48e3ad9d956bb4e4224c68f15649 0.0s
=> => resolve docker.io/library/cassandra:latest@sha256:e7410759812c49ebb71740e40237158f115b48e3ad9d956bb4e4224c68f15649 0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 446B                                 0.0s
=> CACHED [2/9] COPY cql_directory/main_schema.cql /usr/local/bin/main_schema.cql 0.0s
=> CACHED [3/9] COPY cql_directory/insert_data.cql /usr/local/bin/insert_data.cql 0.0s
=> CACHED [4/9] COPY cql_directory/insert_order_details.cql /usr/local/bin/insert_order_details.cql 0.0s
=> CACHED [5/9] COPY cql_directory/retrieve_data.cql /usr/local/bin/retrieve_data.cql 0.0s
=> CACHED [6/9] COPY cql_directory/update_data.cql /usr/local/bin/update_data.cql 0.0s
=> [7/9] COPY cql_directory/customer_query.cql /usr/local/bin/customer_query.cql 0.0s
=> [8/9] COPY cql_directory/specific_category.cql /usr/local/bin/specific_category.cql 0.0s
=> [9/9] COPY cql_directory/delete_data.cql /usr/local/bin/delete_data.cql 0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                          0.0s
=> => writing image sha256:48f6011d57e25bf550cfb595014720764e0663d85c2082540651b3aad914163e 0.0s
=> => naming to docker.io/library/cassandra                    0.0s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
[+] Building 0.0s (0/0)                                docker:desktop-linux
[+] Running 2/2
✔ Network inventory-supply-chain-management_default          Created 0.0s
✔ Container inventory-supply-chain-management-cassandra-1-1  Starte... 0.0s
Waiting for 2 minutes for the Cassandra containers to start properly

```

- **CQL Scripts:** CQL scripts for creating schema, inserting data, updating.

```

Inventory-Supply-Chain-Management — avimehta@Avis-MacBook-Pro — .in-Management — zsh — 137x37
Creating keyspace and tables
Inserting data into the tables
/usr/local/bin/insert_data.cql:63:SyntaxException: line 2:17 no viable alternative at input 'SELECT' (...)VALUES (uuid(), [(SELECT...)
/usr/local/bin/insert_data.cql:67:SyntaxException: line 2:17 no viable alternative at input 'SELECT' (...)VALUES (uuid(), [(SELECT...)
[(SELECT...)insert_order_details.cql:13:SyntaxException: line 4:3 no viable alternative at input 'SELECT' (...)
[(SELECT...)insert_order_details.cql:24:SyntaxException: line 4:3 no viable alternative at input 'SELECT' (...)
Retrieving data from the tables

productid | category | price | productdescription | productname | supplierid
-----
cf3e80e4-d183-494c-bbe6-8eb0a5458dbe | Electronics | 1000.00 | High-performance laptop | Laptop | df639a3c-a3a9-4288-9846-2a3cb0ef
c9f9
71221389-ff09-429e-8ae1-9a15f23c169e | Furniture | 150.00 | Comfortable office chair | Chair | c5b02569-ad76-40c9-a6db-03e32239
6443
ae663e6b-21d0-4429-a418-618e81b6bdbd | Electronics | 500.00 | Latest smartphone model | Smartphone | 44fc4a32-8a67-4cec-bba5-4c55472e
8304

(3 rows)

warehouseid | capacity | contactinformation | location | warehousename
-----
8e06995b-97ba-4221-a595-f3962c01bae6 | 1200 | Contact C | City C | Warehouse C
fab3ebaa-99db-4c34-8cc8-ee84c79f98fc | 800 | Contact B | City B | Warehouse B
e383278e-fa84-4693-aa4b-7234599cce8 | 1000 | Contact A | City A | Warehouse A

(3 rows)

supplierid | contactinformation | productcategories | suppliername
-----
56c2108d-0a87-4e6e-ab2f-eb7cca9f3f72 | Supplier Z Contact | {'Furniture'} | Supplier Z
3422c68a-c59b-44a5-b983-434f22ae3ab3 | Supplier Y Contact | {'Electronics'} | Supplier Y
8e757254-6e20-4f13-bd8c-691a36d114c7 | Supplier X Contact | {'Electronics', 'Furniture'} | Supplier X

(3 rows)

orderid | customername | orderdate | orderstatus | shippingaddress | totalorderamount
-----

```

- **CQL Scripts:** CQL scripts for querying.

```
Inventory-Supply-Chain-Management -- avimehta@Avis-MacBook-Pro -- .ln-Management -- zsh -- 137x37
e383278e-fa84-4693-aa4b-7234599cce8 | 1000 | Contact A | City A | Warehouse A

(3 rows)
supplierid | contactinformation | productcategories | suppliername
-----
56c2108d-0a87-4e6e-ab2f-eb7cca9f3f72 | Supplier Z Contact | {'Furniture'} | Supplier Z
3422c68a-c59b-44a5-b983-434f22ae3ab3 | Supplier Y Contact | {'Electronics'} | Supplier Y
8e757254-6e20-4f13-bd8c-691a36d114c7 | Supplier X Contact | {'Electronics', 'Furniture'} | Supplier X

(3 rows)
orderid | customername | orderdate | orderstatus | shippingaddress | totalorderamount
-----
287289ea-22b6-4d98-9308-132d7dea65f3 | Customer C | 2023-11-25 | Pending | Address C | 1200.00
f9003717-4dc8-4df7-abe6-10ad5922cd0b | Customer B | 2023-11-24 | Processing | Address B | 750.00
0453dea9-701d-4d79-8748-9cf4a21f944b | Customer A | 2023-11-23 | Shipped | Address A | 1650.00

(3 rows)
Updating data in the tables
Retrieve the order history for a specific customer

supplierid | contactinformation | productcategories | suppliername
-----
3422c68a-c59b-44a5-b983-434f22ae3ab3 | Supplier Y Contact | {'Electronics'} | Supplier Y
8e757254-6e20-4f13-bd8c-691a36d114c7 | Supplier X Contact | {'Electronics', 'Furniture'} | Supplier X

(2 rows)
Retrieve the suppliers for a specific product category

orderid | customername | orderdate | orderstatus | shippingaddress | totalorderamount
-----
0453dea9-701d-4d79-8748-9cf4a21f944b | Customer A | 2023-11-23 | Shipped | Address A | 1650.00

(1 rows)
Deleting data from the tables
➔ Inventory-Supply-Chain-Management git:(main) |
```

# Conclusion

This project demonstrates the implementation of a NoSQL database system using Apache Cassandra, tailored for inventory and supply chain management. It highlights how Cassandra can be used to manage complex data relationships and large-scale data in a distributed environment.