**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MÔN HỌC: ĐIỆN TOÁN ĐÁM MÂY**

**ĐỀ TÀI:**

**TÌM HIỂU APACHE HIVE VÀ VIẾT ỨNG DỤNG DEMO**

**Giảng viên hướng dẫn:** ***TS. Huỳnh Xuân Phụng***

**Nhóm sinh viên thực hiện:**

1. Lê Phúc Hạo 17133018
2. Ngô Ngọc Hoài 17133021
3. Lâm Duy Khang 17133031

***TP. Hồ Chí Minh-1/2021***

**MỤC LỤC**

[CHƯƠNG 1: APACHE HIVE 1](#_Toc60843509)

[1. Tổng quan về Apache Hive 1](#_Toc60843510)

[**1.1.Vị trí, vai trò Hive trong hệ sinh thái Hadoop** 2](#_Toc60843511)

[**1.2.Đặc trưng của Hive** 2](#_Toc60843512)

[2.Hoạt động của Hive 3](#_Toc60843513)

[3.Kiến trúc Hive 4](#_Toc60843514)

[4.Tổ chức dữ liệu trong Hive 4](#_Toc60843515)

[**4.1.Tổ chức dữ liệu** 4](#_Toc60843516)

[**4.2.Kiểu dữ liệu** 5](#_Toc60843517)

[5. Ngôn ngữ truy vấn HiveQL 7](#_Toc60843518)

[CHƯƠNG 2: THIẾT KẾ VÀ XÂY DỰNG DATA WAREHOUSE 9](#_Toc60843519)

[1. Thiết kế Data warehouse 9](#_Toc60843520)

[**1.1.Thiết kế Business process và xây dựng Bus matrix** 9](#_Toc60843521)

[**1.2.Xây dựng Dimension và các bảng Fact** 9](#_Toc60843522)

[**1.2.1.Dimension Customers** 9](#_Toc60843523)

[**1.2.2.DimEmployees** 10](#_Toc60843524)

[**1.2.3.DimProducts** 10](#_Toc60843525)

[**1.2.4.Dimension Date** 11](#_Toc60843526)

[**1.2.5.Bảng Fact Oders** 11](#_Toc60843527)

[**1.3**.**Schema** 12](#_Toc60843528)

[2. Thực hiện xây dựng Data warehouse 12](#_Toc60843529)

[**2.1.Stagging dữ liệu** 12](#_Toc60843530)

[**2.2.Load dữ liệu vào DW** 15](#_Toc60843531)

[**2.2.1.Load FactOrder** 15](#_Toc60843532)

[**2.2.2.Load Dimentions** 16](#_Toc60843533)

[3.Truy vấn và báo cáo 18](#_Toc60843534)

[3.1.Số sản phẩm và số tiền bán được của từng nhân viên 18](#_Toc60843535)

[3.2.Số lượng bán được của từng sản phẩm 19](#_Toc60843536)

[3.3.Khách hàng mua nhiều sản phẩm nhất 19](#_Toc60843537)

[4.Kết luận 20](#_Toc60843538)

**CHƯƠNG 1: APACHE HIVE**

1. **Tổng quan về Apache Hive**

Apache Hive là 1 kho dữ liệu (data warehouse) hỗ trợ người sử dụng có thể dễ dàng hơn trong việc quản lý và truy vấn đối với các tập dữ liệu lớn được lưu trữ trên các hệ thống lưu trữ phân tán (distributed storage).

Hive được xây dựng dựa trên cơ sở của Apache Hadoop, nó cung cấp các tính năng chính sau:

* Công cụ cho phép dễ dàng thực hiện tác vụ như trích xuất, vận chuyển và lưu trữ dữ liệu.
* Cơ chế để xử lý cho nhiều định dạng dữ liệu khác nhau.
* Truy cập tới dữ liệu dạng files được lưu trữ trực tiếp ở trong Apache HDFS hoặc đối với nhiều hệ thống lưu trữ dữ liệu khác như Apache HBase.
* Thực hiện query thông qua MapReduce.

Hive định nghĩa ra một ngôn ngữ truy vấn đơn giản có cú pháp gần giống với SQL (SQL-like query language) được gọi là HiveQL, nó cho phép người sử dụng đã quen thuộc với các truy vấn SQL thực hiện việc truy vấn dữ liệu. Ngoài ra ngôn ngữ này còn cho phép các lập trình viên người đã quen thuộc với MapReduce framework có thể nhúng các mappers và reducers cho chính họ viết ra để thực thi nhiều hơn nữa các phân tích phức tập mà không được hỗ trợ bởi các hàm đã có sẵn trong ngôn ngữ HiveQL. HiveQL cung có thể được mở rộng với các custom scalar functions (UDF’s), aggregations (UDAF’s) và các table funtions (UDTF’s)

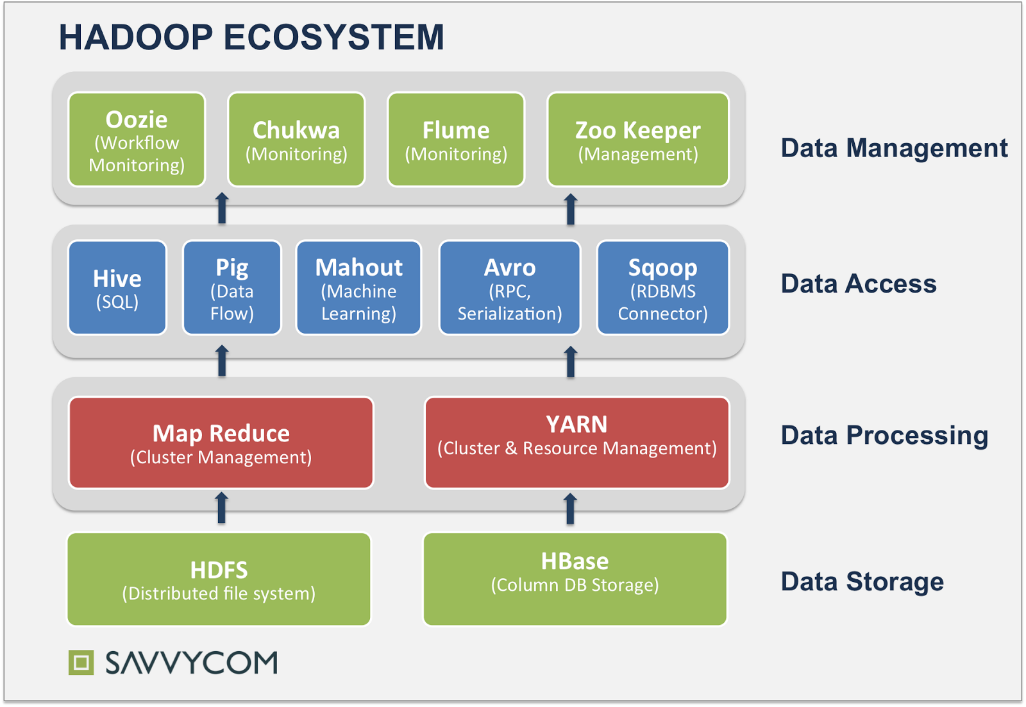
Hive không yêu cầu dữ liệu phải được đọc và ghi dưới một định dạng của riêng Hive (Hive format). Hive hoạt động tốt trên Thrift và các định dạng dữ liệu riêng của người sử dụng.

Hive không được thiết kế để cho các giao dịch online (OLTP workloads) và không nên dùng cho các real-time queries và các cập nhật trên từng dòng trong 1 table (row-level). Hive hoạt động tốt nhất cho các batch jobs trên các tập dữ liệu lớn, mà ở đó dữ liệu được thêm vào liên tục (append-only data) ví dụ như web logs. Hive có khả năng mở rộng theo chiều ngang tốt (thực thi tốt trên 1 hadoop cluster có số tượng máy biến đổi), có khả năng tích hợp với MapReduce framework và UDF, UDAF, UDTF; có khả năng chống chịu lỗi và mềm dẻo đối với các dữ liệu đầu vào của chính nó.

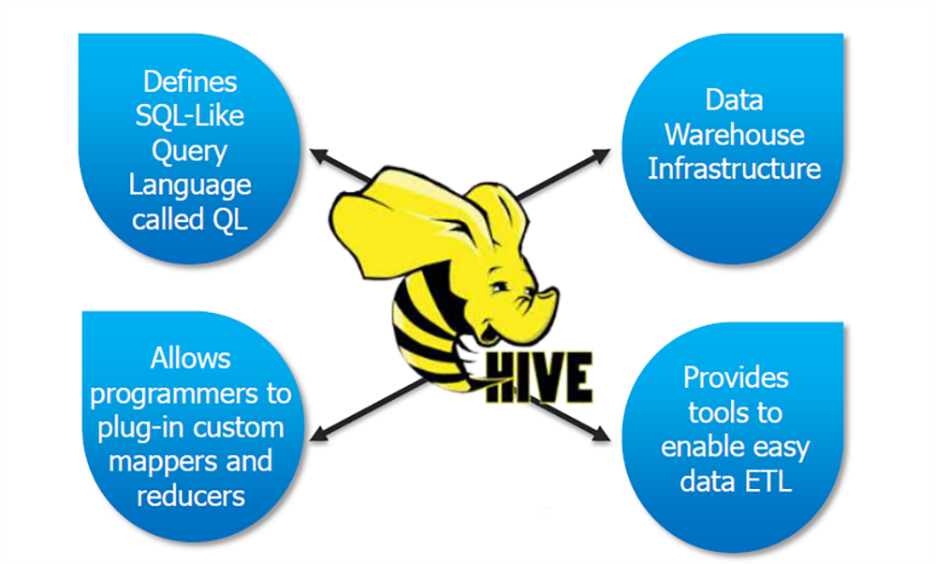
Các thành phần cấu hình Hive bao gồm HCatalog và WebHCat. HCatalog là một thành phần của Hive. Đây là lớp quản lý lưu trữ cho Hadoop (table and management layer), nó cho phép người dùng với các công cụ xử lý dữ liệu khác nhau bao gồm cả Pig và MapReduce thực thi hoạt động đọc, ghi một cách dễ dàng hơn. WebHCat cung cấp một dịch vụ cho phép bạn có thể thực thi Hadoop MapReduce (hoặc YARN), Pig, Hive.

* 1. **Vị trí, vai trò Hive trong hệ sinh thái Hadoop**

Hive là một công cụ cơ sở hạ tầng kho dữ liệu để xử lý dữ liệu có cấu trúc trong Hadoop. Nó nằm trên đỉnh Hadoop để tóm tắt Dữ liệu lớn và giúp truy vấn và phân tích dễ dàng.



* 1. **Đặc trưng của Hive**



Lưu trữ lược đồ trong cơ sở dữ liệu và xử lý dữ liệu vào HDFS.

Được thiết kế cho OLAP.

Cung cấp ngôn ngữ kiểu SQL để truy vấn được gọi là HiveQL hoặc HQL.

Quen thuộc, nhanh chóng, có khả năng mở rộng.

1. **Hoạt động của Hive**



Cách Hive tương tác với framework Hadoop:

1. Thực thi query: Giao diện Hive như Command line hoặc Giao diện người dùng web gửi truy vấn đến Trình điều khiển (bất kỳ trình điều khiển cơ sở dữ liệu nào như JDBC, ODBC, v.v.) để thực thi.
2. Nhận kế hoạch: Trình điều khiển có sự trợ giúp của trình biên dịch truy vấn để phân tích cú pháp truy vấn để kiểm tra cú pháp và kế hoạch truy vấn hoặc yêu cầu của truy vấn.
3. Nhận metadata: Trình biên dịch gửi yêu cầu metadata đến Metastore (bất kỳ cơ sở dữ liệu nào).
4. Gửi metadata: Metastore gửi metadata như một phản hồi cho trình biên dịch.
5. Gửi kế hoạch: Trình biên dịch kiểm tra yêu cầu và gửi lại kế hoạch cho trình điều khiển. Đến đây, việc phân tích cú pháp và biên dịch một truy vấn đã hoàn tất.
6. Kế hoạch thực hiện: Trình điều khiển gửi kế hoạch thực hiện đến công cụ thực thi.
7. Thực thi công việc: Trong nội bộ, quá trình thực thi công việc là một công việc MapReduce. Công cụ thực thi gửi công việc đến JobTracker, trong node Name và nó gán công việc này cho TaskTracker, trong node Data. Ở đây, truy vấn thực thi công việc MapReduce. Hoạt động metadata: Trong khi thực hiện, công cụ thực thi có thể thực thi các hoạt động metadata với Metastore.
8. Lấy kết quả: Công cụ thực thi nhận kết quả từ các node Data.
9. Gửi kết quả: Công cụ thực thi gửi các giá trị kết quả đó đến trình điều khiển.
10. Gửi kết quả: Trình điều khiển gửi kết quả đến Giao diện Hive.
11. **Kiến trúc Hive**



Sơ đồ thành phần này chứa các đơn vị khác nhau.

• User Interface: Hive là một phần mềm cơ sở hạ tầng kho dữ liệu có thể tạo ra sự tương tác giữa người dùng và HDFS. Các giao diện người dùng mà Hive hỗ trợ là Hive Web UI, Hive command line và Hive HD Insight (Trong máy chủ Windows).

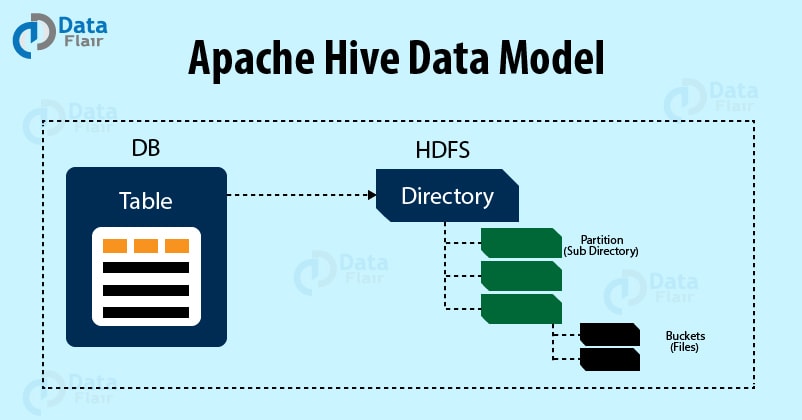
• Meta Store: Hive chọn các máy chủ cơ sở dữ liệu tương ứng để lưu trữ lược đồ hoặc metadata của các bảng, cơ sở dữ liệu, các cột trong một bảng, các loại dữ liệu của chúng và ánh xạ HDFS.

• HiveQL Process Engine: HiveQL tương tự như SQL để truy vấn thông tin lược đồ trên Metastore. Đây là một trong những thay thế của phương pháp truyền thống cho chương trình MapReduce. Thay vì viết chương trình MapReduce bằng Java, chúng ta có thể viết một truy vấn cho công việc MapReduce và xử lý nó.

• Execution Engine: Phần kết hợp của công cụ xử lý HiveQL và MapReduce là Công cụ thực thi Hive (Hive Execution Engine). Công cụ thực thi xử lý truy vấn và tạo kết quả giống như kết quả MapReduce.

• HDFS hoặc HBASE: Hệ thống tệp phân tán Hadoop hoặc HBASE là các kỹ thuật lưu trữ dữ liệu để lưu trữ dữ liệu vào hệ thống tệp.

1. **Tổ chức dữ liệu trong Hive**
   1. **Tổ chức dữ liệu**



Dữ liệu được tổ chức thành 3 định dạng trong HIVE.

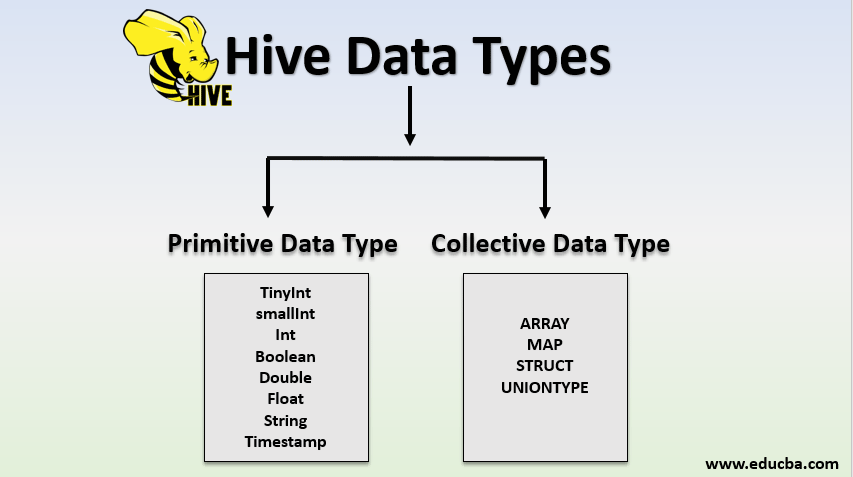
• Tables: Chúng rất tương tự như bảng (tables) trong RDBMS và chứa các dòng (rows). Hive chỉ được xếp lớp trên HDFS, do đó tables được ánh xạ trực tiếp vào các thư mục của hệ thống tập tin. Nó cũng hỗ trợ các tables được lưu trên các hệ thống tập tin khác.

• Partitions: Hive tables có thể có nhiều hơn 1 partition. Chúng được ánh xạ với các thư mục con và các hệ thống tập tin.

• Buckets: Trong Hive, dữ liệu có thể được chia thành các buckets. Buckets được lưu trữ như các tập tin trong partition trong hệ thống tập tin.

Hive cũng có metastore để lưu tất cả metadata. Nó là CSDL quan hệ chứa thông tin khác nhau liên quan đến Hive Schema (column types, owners, key-value data, statistics,…). Chúng ta có thể dùng MySQL cho việc này.

* 1. **Kiểu dữ liệu**



Hive là một script language được xây dựng theo cấu trúc gần giống với SQL với những khái niệm căn bản như bảng, cột, dòng và partition (giống như label cho các ngăn chứa). Đồng thời Hive hỗ trợ các kiểu dữ liệu gốc như intergers, floats, doubles và string cũng như các kiểu dữ liệu phức tạp như maps, list và structs.

Các kiểu dữ liệu gốc sau:

Integers

• TINYINT – 1 byte integer

• SMALLINT – 2 byte integer

• INT – 4 byte integer

• BIGINT – 8 byte integer

Boolean type

• BOOLEAN – TRUE/FALSE

Floating point numbers

• FLOAT – single precision

• DOUBLE – Double precision

String type

• STRING – sequence of characters in a specified character set

Hive cũng hỗ trợ các kiểu dữ liệu phức tạp như:

• Associative arrays – map<key-type, value-type>

• Lists – list

• Structs – struct

Các kiểu dữ liệu trên hoàn toàn có thể lồng vào nhau. Ví dụ:

list<map<string, struct<p1:int, p2:int>>

Ví dụ về tạo bảng t1:

CREATE TABLE t1(st string, fl float, li list<map<string, struct<p1:int, p2:int>>);

Ta truy xuất đến các trường trong bảng bằng cách sử dụng dấu “.” và dùng dấu “[ ]” để truy xuất vào các list và array.

Hạn chế

Hive có một vài hạn chế như không cho insert dữ liệu vào bảng. Các câu lệnh quen thuộc trong SQL như INSERT INTO, DELETE hay UPDATE đều được lược bỏ trong Hive.

Hive không hỗ trợ chèn vô bảng hay partition mà ta chỉ có thể overwrite lại toàn bộ dữ liệu INSERT OVERWRITE TABLE t1

1. **Ngôn ngữ truy vấn HiveQL**

Hive định nghĩa ra một ngôn ngữ truy vấn đơn giản có cú pháp gần giống với SQL (SQL-like query language) được gọi là HiveQL, nó cho phép người sử dụng đã quen thuộc với các truy vấn SQL thực hiện việc truy vấn dữ liệu. Ngoài ra ngôn ngữ này còn cho phép các lập trình viên người đã quen thuộc với MapReduce framework có thể nhúng các mappers và reducers cho chính họ viết ra để thực thi nhiều hơn nữa các phân tích phức tập mà không được hỗ trợ bởi các hàm đã có sẵn trong ngôn ngữ HiveQL. HiveQL cũng có thể được mở rộng với các custom scalar functions (UDF’s), aggregations (UDAF’s) và các table funtions (UDTF’s)

• Câu lệnh SELECT được sử dụng để lấy dữ liệu từ một bảng.

• Mệnh đề WHERE hoạt động tương tự như một điều kiện. Nó lọc dữ liệu bằng cách sử dụng điều kiện và cung cấp cho bạn một kết quả hữu hạn. Các toán tử và hàm tích hợp tạo ra một biểu thức, đáp ứng điều kiện.

• Mệnh đề ORDER BY được sử dụng để truy xuất các chi tiết dựa trên một cột và sắp xếp kết quả được đặt theo thứ tự tăng dần hoặc giảm dần.

• Mệnh đề GROUP BY được sử dụng để nhóm tất cả các bản ghi trong tập kết quả bằng cột thu thập cụ thể. Nó được sử dụng để truy vấn một nhóm các hồ sơ.

Cú pháp truy vấn:

SELECT [ALL | DISTINCT] select\_expr, select\_expr, ...

FROM table\_reference

[WHERE where\_condition]

[GROUP BY col\_list]

[HAVING having\_condition]

[ORDER BY col\_list]]

[LIMIT number];

**CHƯƠNG 2: THIẾT KẾ VÀ XÂY DỰNG DATA WAREHOUSE**

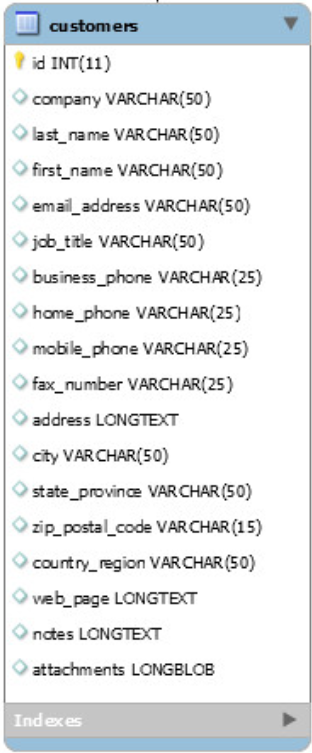
Để áp dụng những kiến thức nhóm đã tìm hiểu về Apache Hive một cách thực tế hơn, nhóm bắt tay vào việc chọn CSDL và tiến hành xây dựng một Data Warehouse đơn giản. Tập CSDL nhóm chọn là Northwind.

**1. Thiết kế Data warehouse**

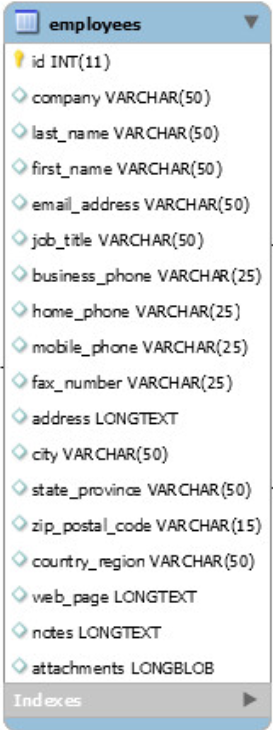
**1.1 Thiết kế Business process và xây dựng Bus matrix**

Từ CSDL Northwind nhóm muốn tạo data mart với các mục đích sau:

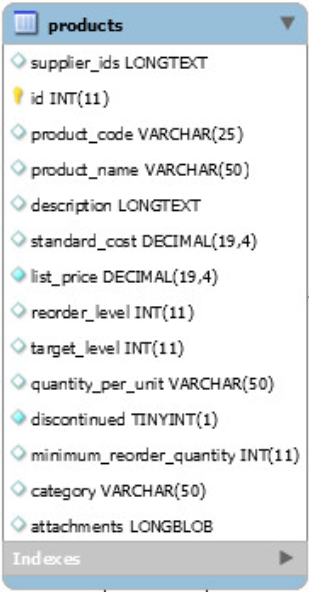
* Phân tích về việc đặt và giao hang, theo dõi thời gian dự kiến và thời gian thực của việc giao hàng.
  1. **Xây dựng Dimension và các bảng Fact**
     1. **Dimension Customers**

****

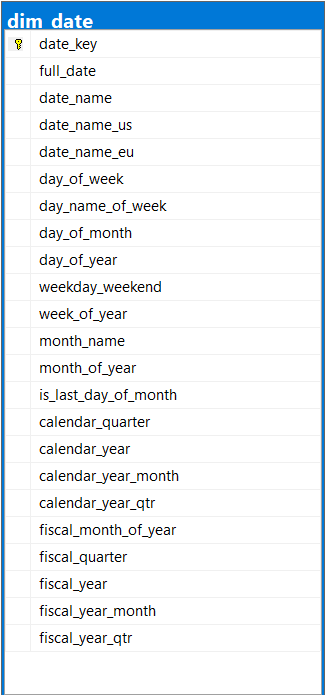
* + 1. **DimEmployees**

****

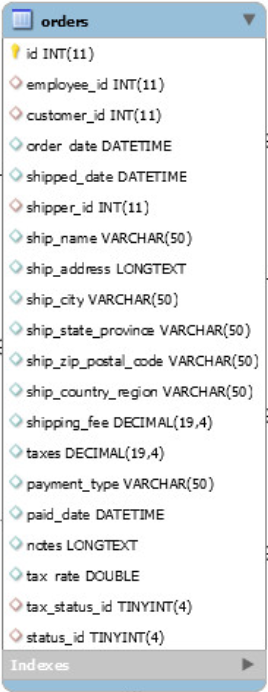
* + 1. **DimProducts**

****

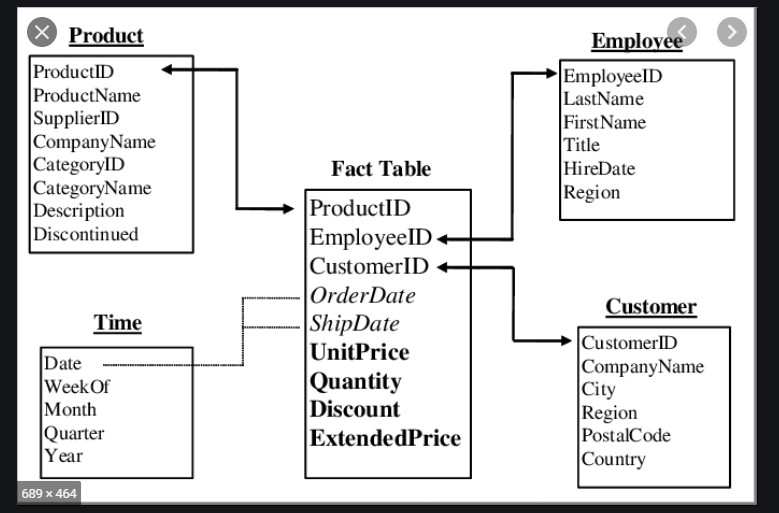
* + 1. **Dimension Date**

****

* + 1. **Bảng Fact Oders**

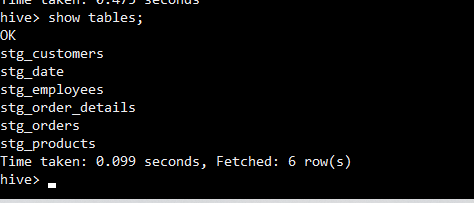
****

* 1. **Schema**



1. **Thực hiện xây dựng Data warehouse**
   1. **Stagging dữ liệu**

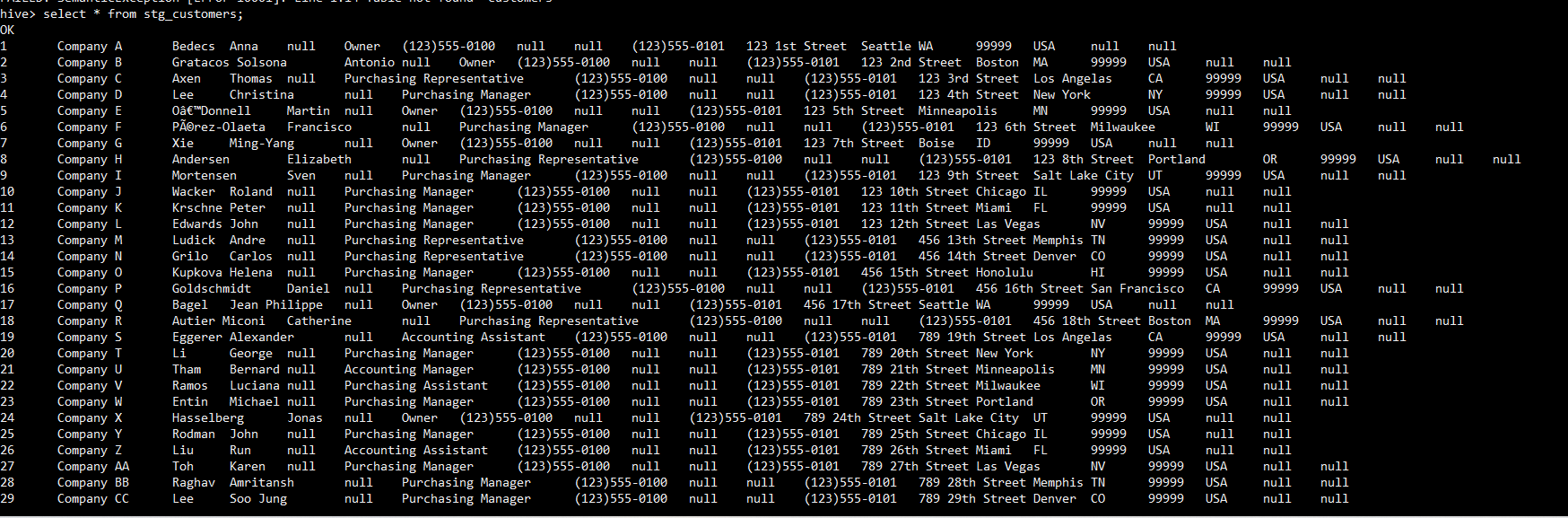
Các table cần thiết để xây dựng DataWarehouse. Những table này loại bỏ bad data ở database NorthWind ban đầu, sau đó sẽ load lên mô hình DataWarehouse.

****

**Dữ liệu sau khi đã stagging**

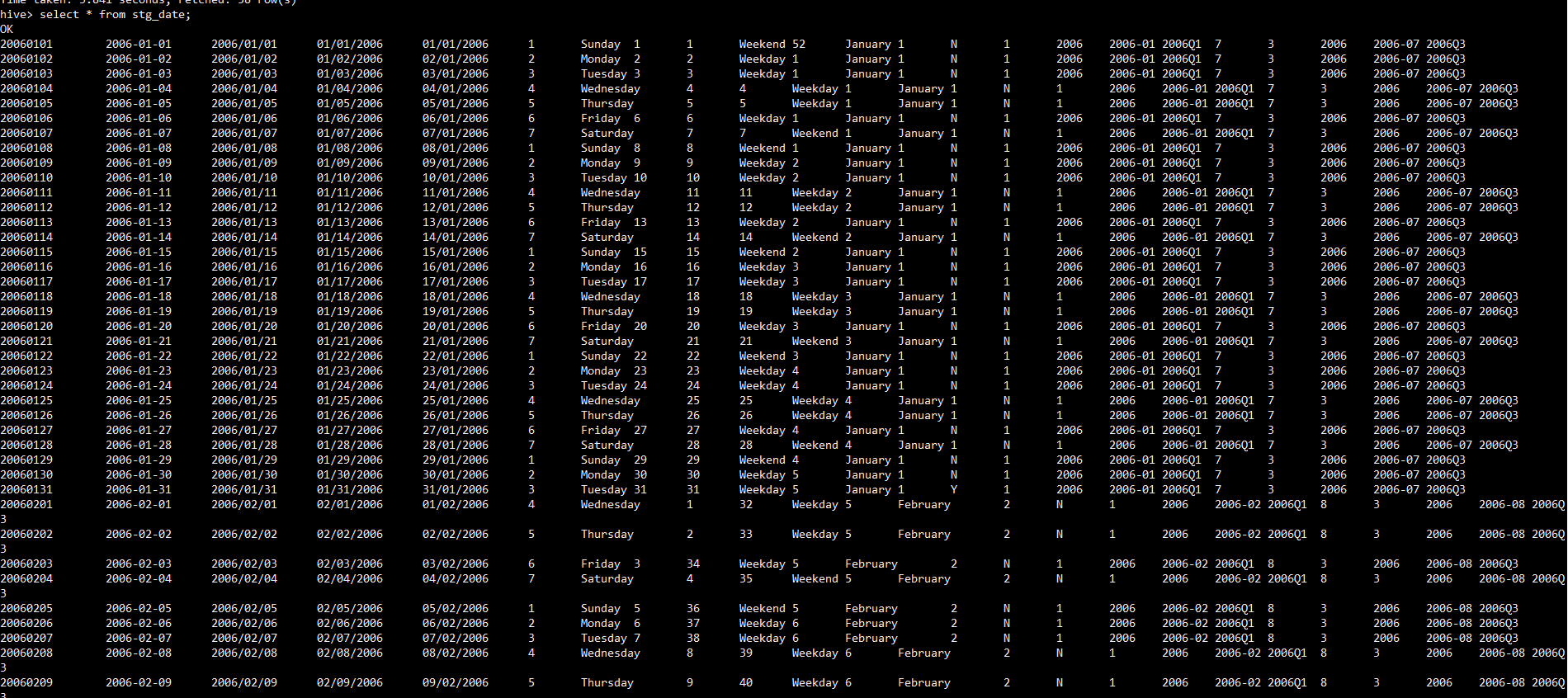
**Table Stg\_Customers**

Thực hiện truy vấn với HQL để insert data vào table stg\_customers từ database nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

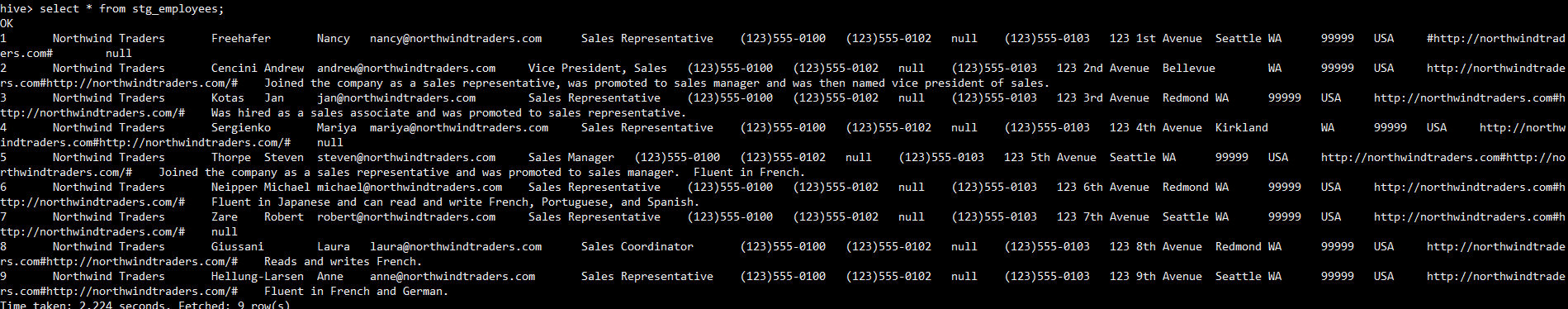
**Table stg\_date**

Thực hiện truy vấn với HQL để insert data vào table stg\_date từ database nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

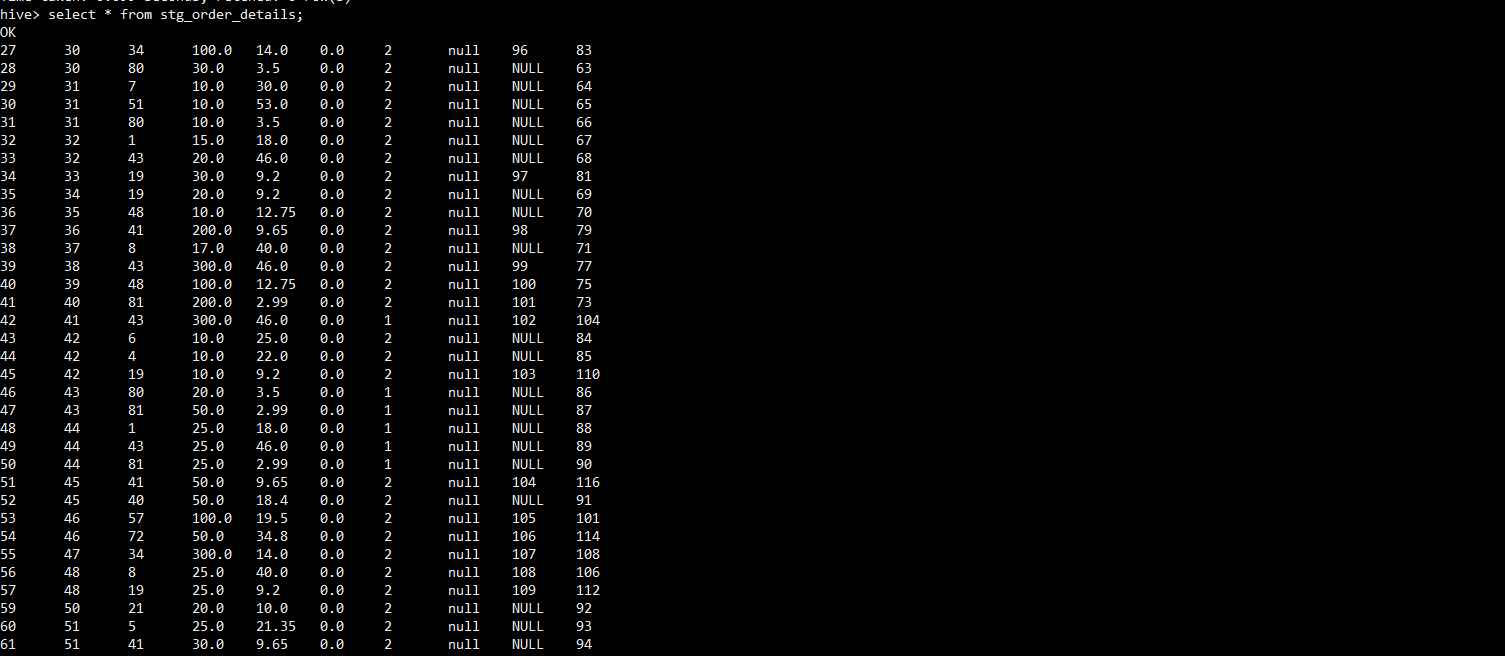
**Table stg\_employees**

Thực hiện truy vấn với HQL để insert data vào table stg\_employees từ database nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

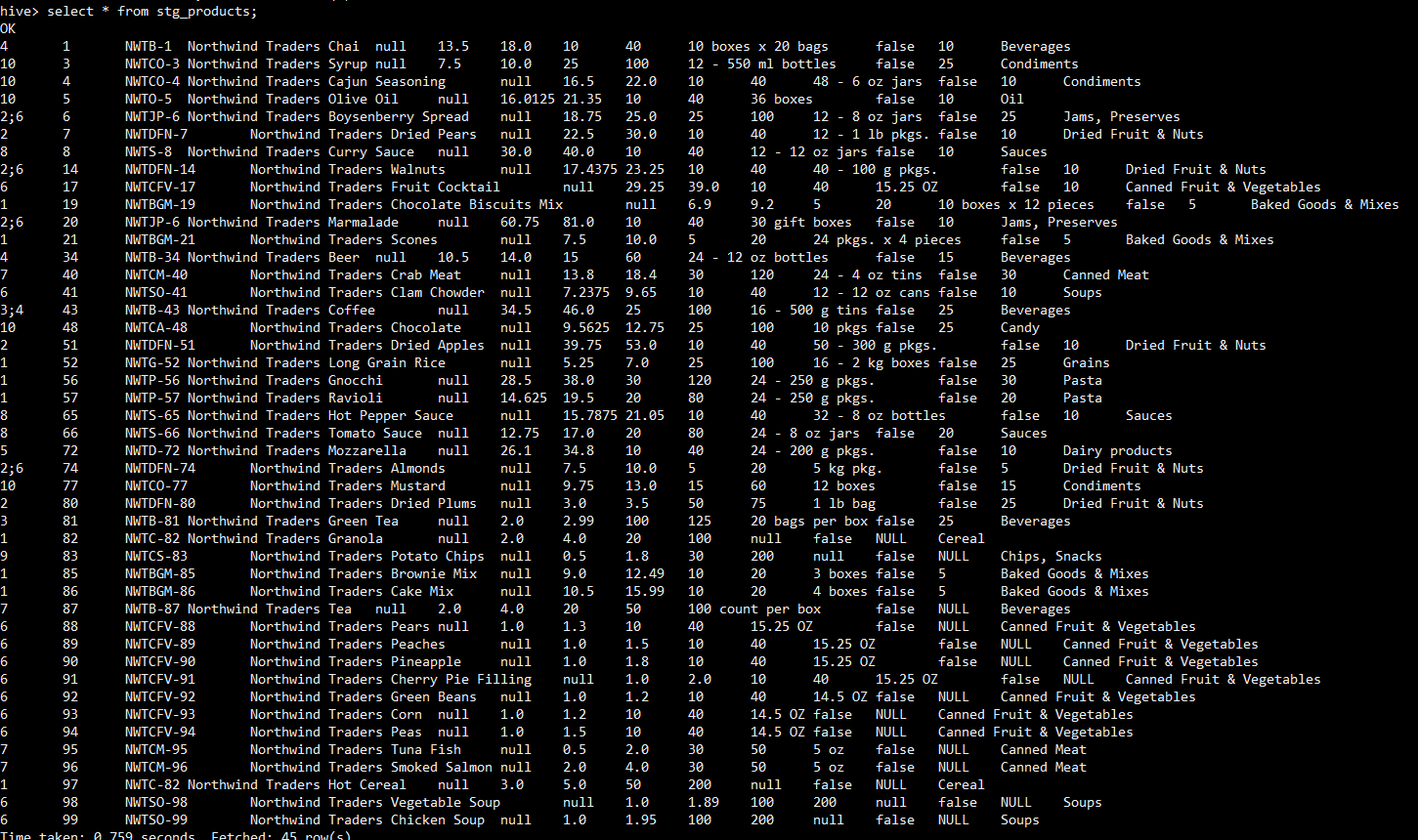
**Table Order\_details**

Thực hiện truy vấn với HQL để insert data vào table stg\_order\_details từ database nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

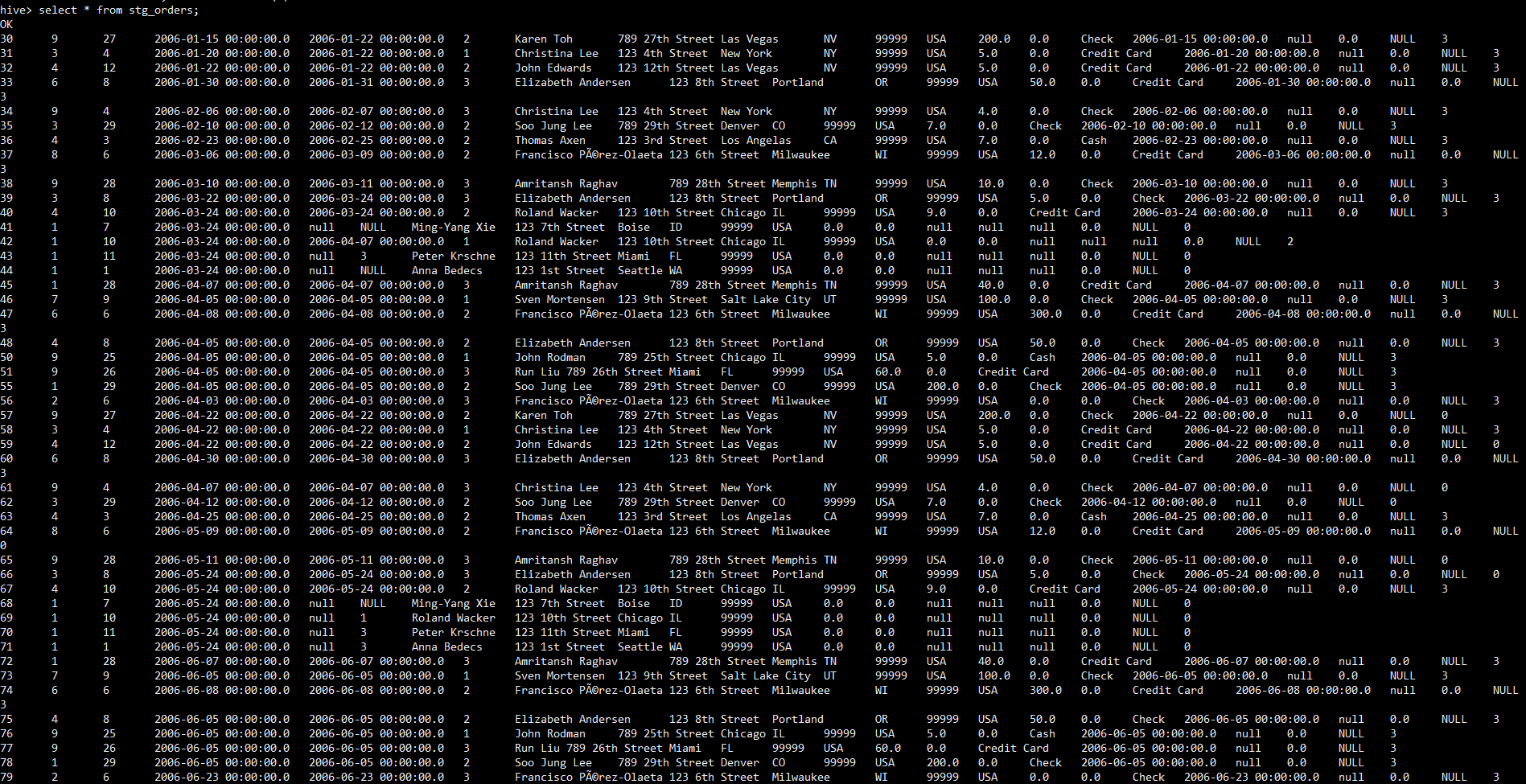
**Table Products**

Thực hiện truy vấn với HQL để insert data vào table stg\_product từ database nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

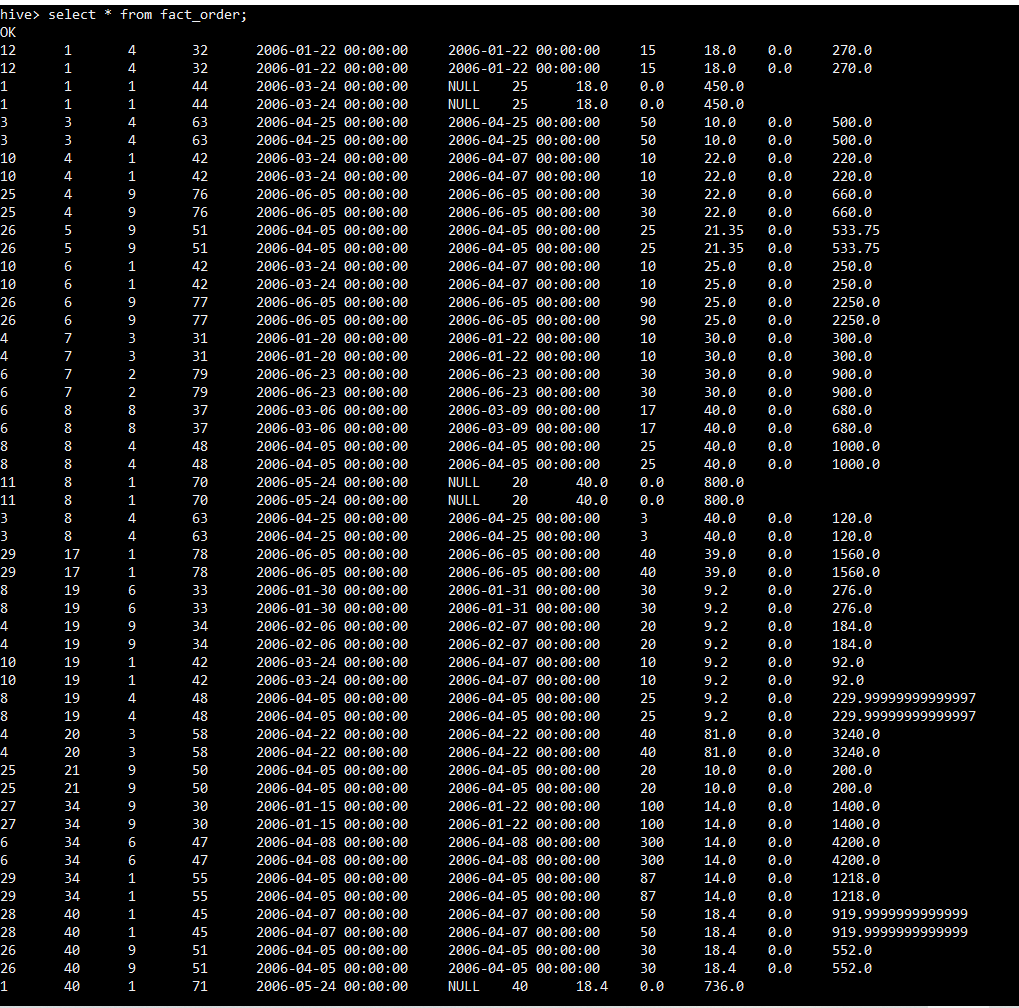
**Table Orders**

Thực hiện truy vấn với HQL để insert data vào table stg\_orders từ database nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

* 1. **Load dữ liệu vào DW**
     1. **Load FactOrder**

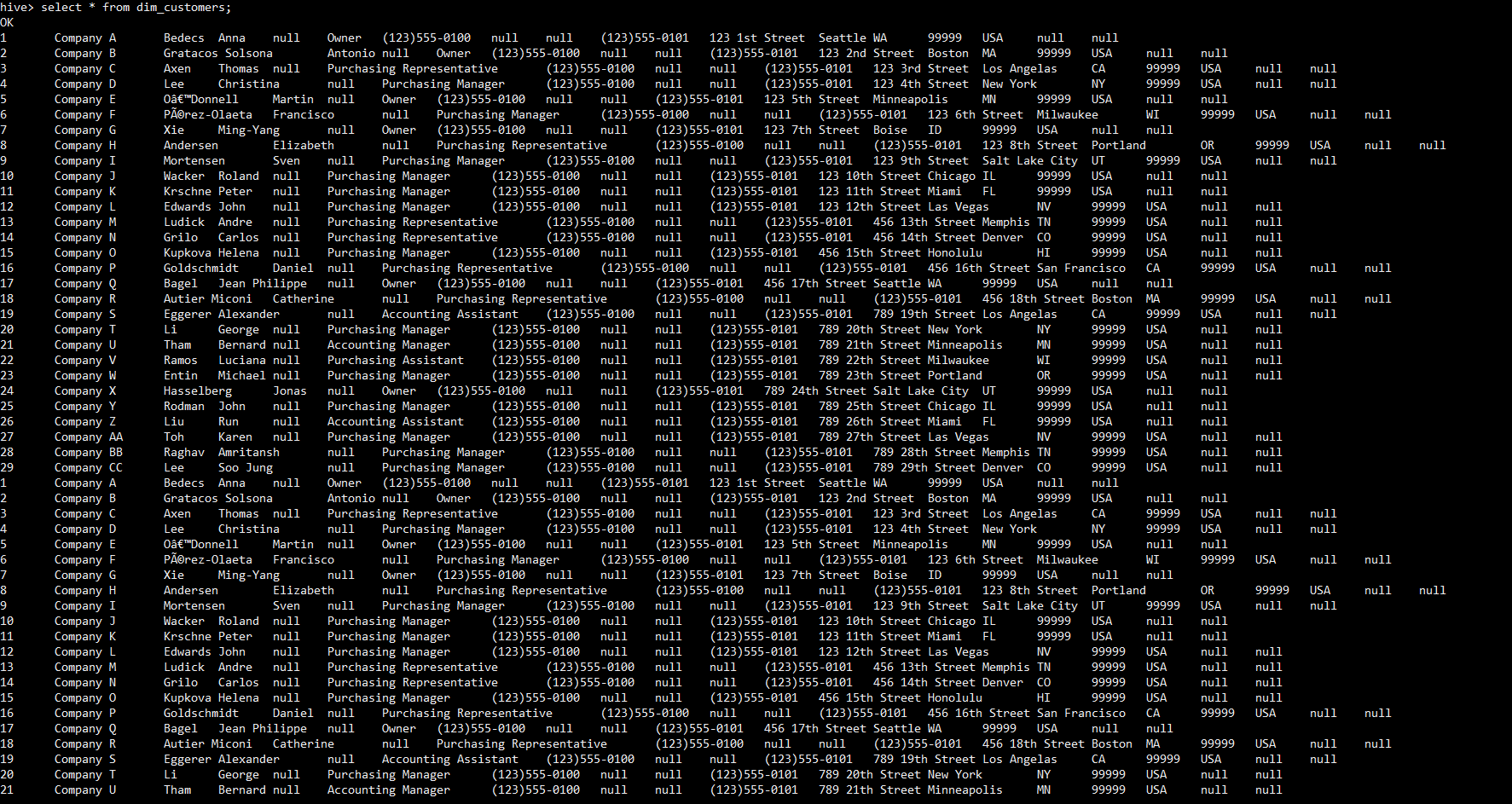
Thực hiện truy vấn với HQL để insert data vào table fact\_order từ database stg\_nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

* + 1. **Load Dimentions**

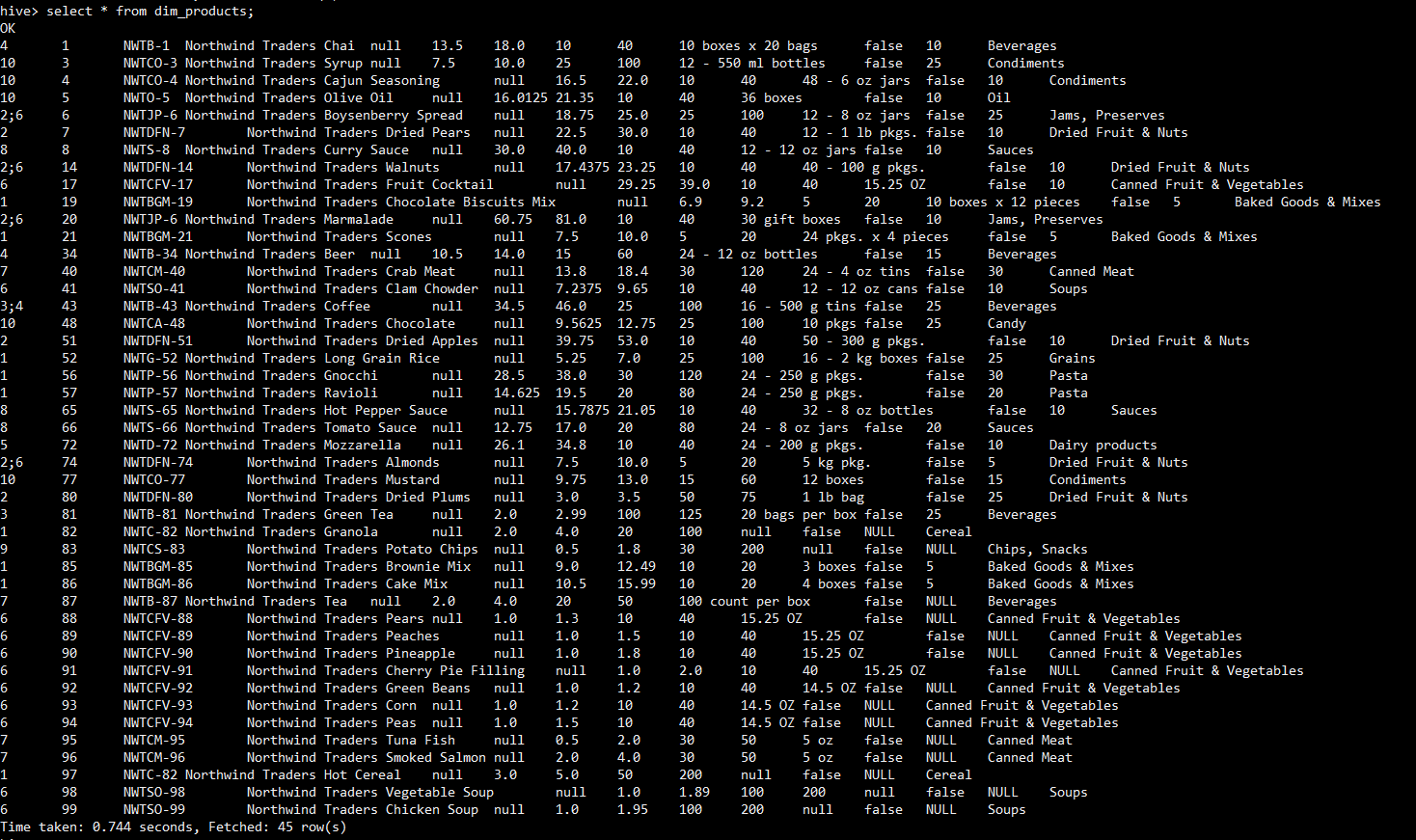
**Dim\_customers**

Thực hiện truy vấn với HQL để insert data vào table dim\_customers từ database stg\_nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

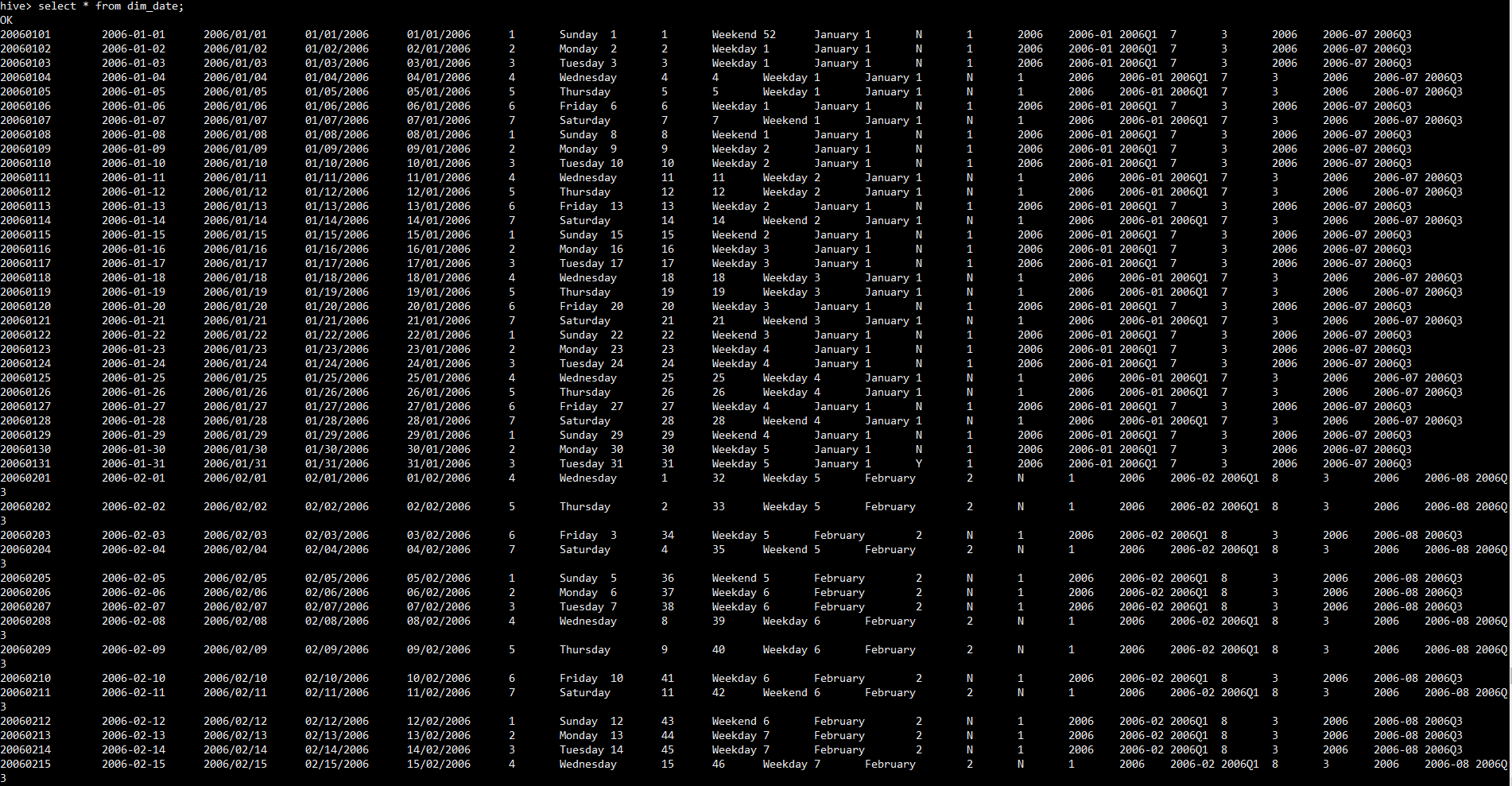
**Dim\_products**

Thực hiện truy vấn với HQL để insert data vào table dim\_products từ database stg\_nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

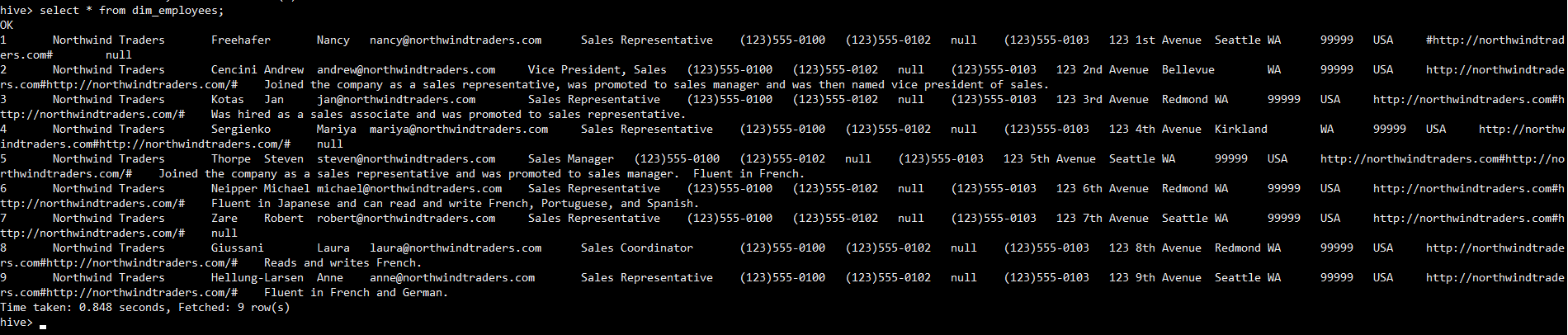
**Dim\_date**

Thực hiện truy vấn với HQL để insert data vào table dim\_date từ database stg\_nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

**Dim\_Employees**

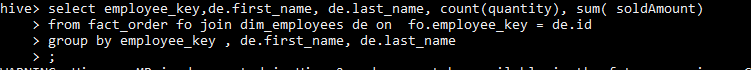
Thực hiện truy vấn với HQL để insert data vào table dim\_employees từ database stg\_nowi. Sau khi thực hiện truy vấn, ta được kết quả như sau:

****

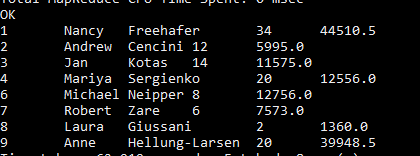
1. **Truy vấn và báo cáo**

3.1 Số sản phẩm và số tiền bán được của từng nhân viên

Thực hiện truy vấn với hql:



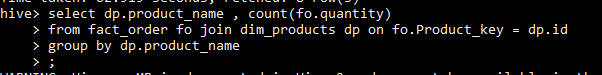
Sau khi thực hiện truy vấn, kết quả của từng nhân viên như sau:



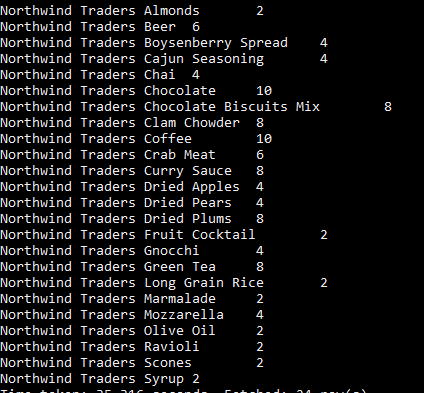
Cột đầu tiên là mã của từng nhân viên, 2 cột tiếp theo là tên nhân viên, cột thứ tư là số sản phẩm bán được và cột cuối là tổng số tiền bán được. Nhìn vào đâ ta có thể thấy nhân viên Nancy Freehafer là người bán được nhiều sản phẩm nhất, số tiền bán được cũng cao nhất. Nhân viên Laura Giusani bán được ít sản phẩm nhất. Chúng ta có thể xếp hạng cho từng nhân viên từ đó đưa ra khen thưởng hợp lý.

# 3.2 Số lượng bán được của từng sản phẩm

Thực hiện truy vấn với hql:



Sau khi thực hiện truy vấn, kết quả như sau:



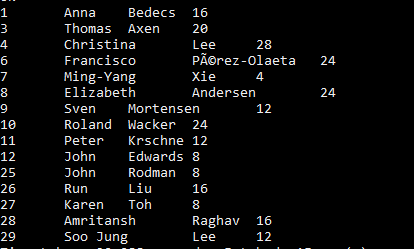
Cột đầu tiên là tên sản phẩm, cột tiếp theo là tổng số lượng sản phẩm đó bán được. Nhìn vào đây chúng ta có thể thấy, sản phẩm bán chạy nhất là Northwind Traders Coffee và Northwind Traders Chocolate với số lượng là 10. Rất nhiều sản phẩm bán ít chạy với số lượng là 2. Chúng ta có thể nhìn vào số liệu để thấy được sở thích, nhu cầu của khách hang, từ đó đưa ra chiến lược kinh doanh mặt hang hợp lý

3.3 Khách hàng mua nhiều sản phẩm nhất

Thực hiện truy vấn với hql:



Sau khi thực hiện truy vấn, kết quả như sau:



Cột đầu tiên là mã khách hàng, 2 cột tiếp theo là tên khách hàng, cột cuối là số lượng sản phẩm mà khách hàng đó mua. Chúng ta có thể thấy, những khách hàng top đầu mua sản phẩm số lượng lớn (24 đến 28 sản phẩm) chênh lệch với những khách hàng khách khá cao. Từ bảng thông tin khách hàng này, nhà kinh doanh có thể áp dụng các ưu đãi, hay thành viên VIP cho những khách hàng mua nhiều , đưa ra chiến lược kinh doanh đúng đắn.

1. **Kết luận**

Sau quá trình thực hiện Project, nhóm đã phần nào học được kiến thức về Apache Hive, truy vấn với HQL và áp dụng kiến thức để thiết kế và xây dựng một Data warehouse. Biết được cách Hive hoạt đông trên Hadoop, luồng dữ liệu của Hive, đặc trưng , kiến trúc, cách tổ chức dữ liệu trong Hive, thiết kế và xây dựng các bảng fact, các dimension, thực hiện tích hợp dữ liệu bằng ETL, HQL để tạo truy vấn.

Bên cạnh những việc nhóm đã làm thì không thể thiếu các sơ xuất xảy ra và những khó khăn mắc phải. Do phải đổi đề tài nên thời gian của nhóm còn hạn hẹp, chưa đủ để khai thác hết các business process cho kho dữ liệu NorthWind, mô hình datawarehouse còn nhỏ , chưa sử dụng được nhiều truy vấn với HQL.Nhóm mong muốn rằng trong thời gian sắp tới có thể cải thiện hơn về project, mong sự góp ý từ thầy và các bạn để nhóm đánh giá một cách khách quan hơn về những gì đạt và chưa đạt được của nhóm.

**TÀI LIỆU THAM KHẢO**

1. <https://www.tutorialspoint.com/hive/hive_introduction.htm>
2. <https://intellipaat.com/blog/what-is-apache-hive/>
3. <https://www.edureka.co/blog/introduction-to-apache-hive/>
4. Relational Database Support for Data Warehouses (University of Colorado System)