NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Storing and Retrieving Data

Database design of an Online Store

Fábio Lopes (20200597)

Filipe Costa (20201041)

Jorge Pereira (20201085)

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

INDEX

1. INTRODUCTION

SuperDuperStore is an online store which requested an implementation of a database to keep track of all the information about their sales and customers.

After a meeting with the demand manager for the company, a sample target invoice was provided and, we can split the requirements in 4 major pillars:

- Products
  - Every product price change needs to be audited.
  - When a product is sold, its stock needs to be immediately reflected in the system.
  - Each product needs to be tracked individually per invoice.
- Clients
  - Each client needs to be segmented by the tax to be applied in each purchase for future easiness when/if the tax rates change.
  - Each client needs to be segmented by its spending category using the following logic:
    - If no purchase was made in the past 2 months, it is considered a "Lapsed Spender".
    - By comparing the average spend of the client versus the average spend for all clients in the past 2 months:
      - If below average, Low Spender.
      - If above average but below 2 * average, Medium Spender.
      - If above 2 * average, High Spender.
- Invoices
  - Every invoice needs to be analyzed both at the invoice level and the product per invoice level.
  - Each invoice needs to contain the information about its status and new status can be created in the future.
  - Easy access to pre-formatted data for the Invoice Generator.
- New acquisition migration
  - SuperDuperStore has just acquired a new company which tracked its users on an excel spreadsheet which contains several issues, which need to be dealt with before incorporating new users into the new system.
  - The company has ongoing promotional campaigns running for their users based on the category of each user and wants the new users to also be included in these campaigns. A logic was provided to migrate the old Spending category of the acquired store (with a range from 1-100) to the company standard.

With the information at hand, this document serves as a proposal for a database designed specifically with the SuperDuperStore necessities in mind.

We start by documenting the Entity Relationship Diagram proposed and a brief description of each entity, followed by a plan of migration for the new customers and the necessary steps to deploy a dummy database for POC.

2. DATABASE ERD

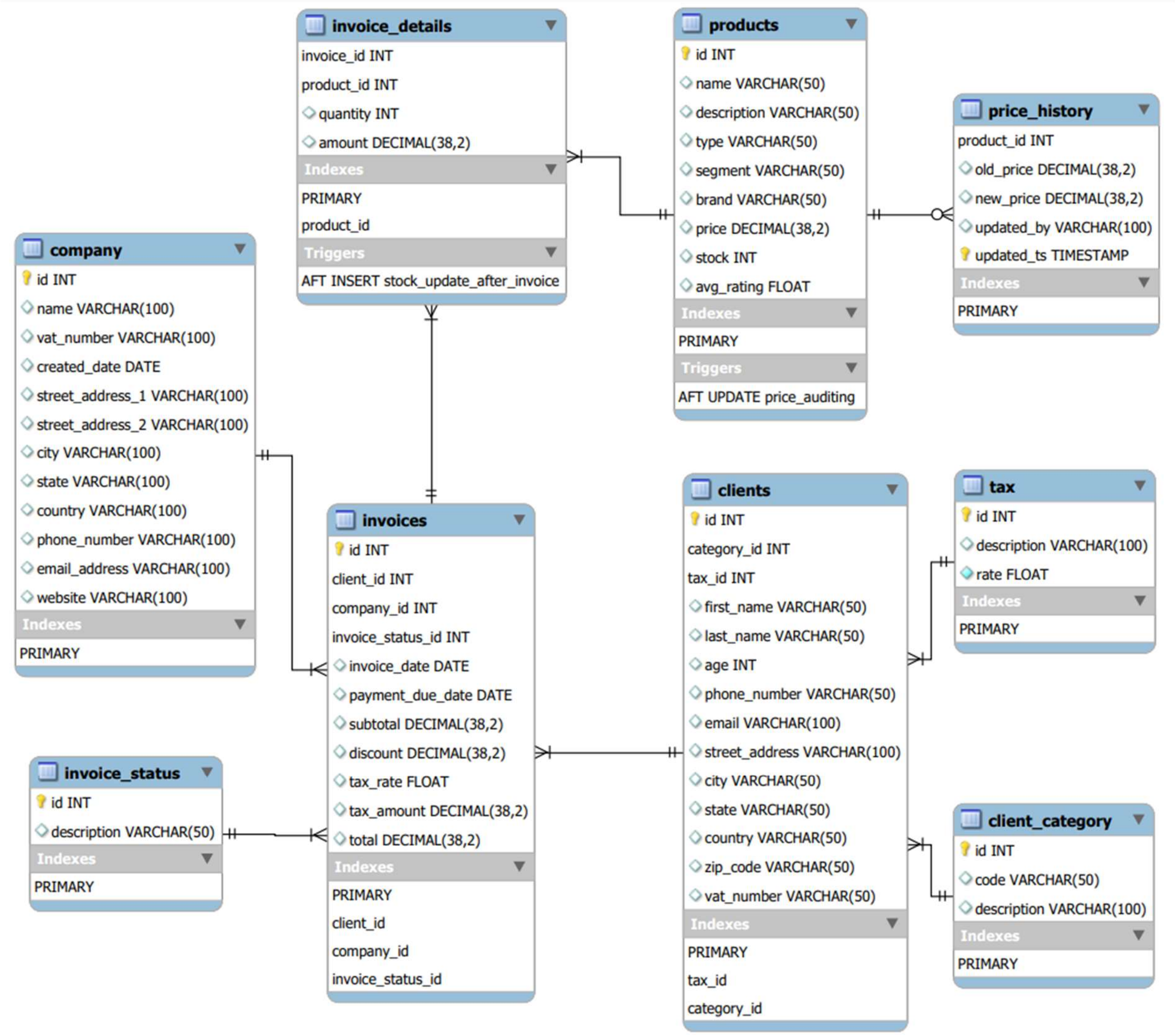


Figure 1 – Entity Relationship Diagram of the Database

Table 1 – Description of tables in the ERD

Table	Description
company	Contains all the information about the company.
invoice_status	Contains all the possible states about the invoices (PAID/PENDING/CANCELED)
price_history	Contains a log of the price history of all the products in the system. Whenever an update is executed on the products table, it will save the old price, the new price, the user which performed the operation and the timestamp. A unique row is identified by the product_id (Foreign Key) and the updated_ts column.
tax	Contains all the possible tax information. Currently includes only the tax information for Personal or Company instances.
client_category	Contains the different categories to which a customer can belong.
clients	Contains all the client information.
products	Contains all the product information.
invoices	Contains all the information about a specific invoice, namely, to which customer and company it belongs to, which date was the invoice created and is due, the status id of the status and all the numeric values associated with the invoice.
invoice_details	Contains all the products sold in each invoice and all the numeric values associated with each product, within each invoice.

3. MIGRATION OF NEW CUSTOMERS

As part of the database design, the store requires the integration of a subset of users from a newly acquired company into their own systems. This store managed their users on a excel spreadsheet containing customer_name, age and the spending category.

By maintaining users in an excel sheet, some issues are present in the data which need to be dealt with, mainly:

- If customer_name + age represents a single entity, there are duplicated users in the data.
- Usernames with 1 character are not recognized as valid users.
- Spending Category refers to the past system and is composed of a classification of users in the scale of 0-100.

To address these issues an ETL script was developed using *Pentaho*, an open source software distributed by *Hitachi Vantara*.

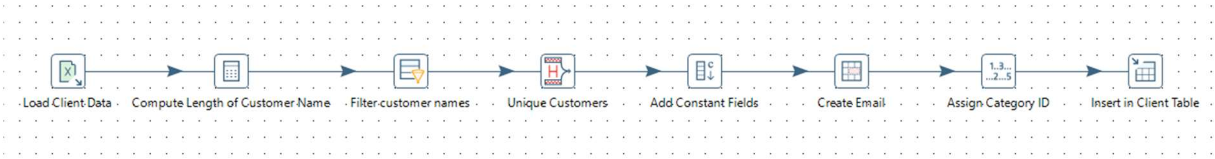


Figure 2 – Transformations in Pentaho

Table 2 – Description of the Activities in the ETL script

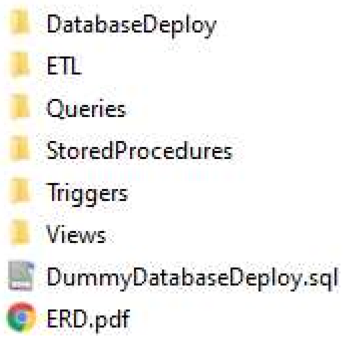
Activity Name	Activity Type	Details
Load Client Data	Microsoft Excel Input	Reads the excel file from a local path.
Compute Length of Customer Name	Calculator	Computes the Length of the customer name.
Filter Customer Names	Filter Rows	Based on the previous step, filter the rows where the customer name length is 1.
Unique Customers	Unique Rows (Hashset)	Based on the subset of columns, customer_name and age, drop the duplicates.
Add Constant Fields	Add Constants	Add the Tax ID column (we assume that all customers are Personal, and so tax_id = 2). Add the companies email suffix, "@SuperDuperStore.com".

Create Email	Concat Fields	Concatenate the customer_name, age and email suffix to build the email address of each user.
Assign Category ID	Number Range	Assign a category id per range of values of Spending Category: • 0-49: 1 (Low Spender) • 50-74: 2 (Medium Spender) • 75-100 (High Spender)
Insert in Client Table	Table Output	Inserts the new users into the Database.

After a successful run, a total of 4591 customers where introduced in the database.

4. DELIVERABLES

4.1. STRUCTURE



DatabaseDeploy
ETL
Queries
StoredProcedures
Triggers
Views
DummyDatabaseDeploy.sql
ERD.pdf

Figure 3 – Deliverables for this project

- **DatabaseDeploy**
  - Location of the Jupyter Notebook which generates dummy data for the database being designed.
- **ETL**
  - Location of the kettle file to be run in Pentaho.
- **Queries**
  - Location of sql file with the 5 Data Exploration Queries requested.
- **StoredProcedures**
  - Location of sql file with the Procedure Developed to segment the customers.
- **Triggers**
  - Location of the 2 sql files with the Triggers developed.
- **Views**
  - Location of the 2 sql files with the Views developed.
- **DummyDatabaseDeploy**
  - Sql file which deploys the database, including dummy data, procedures, triggers and views.
- **ERD**
  - PDF file with the Database Design

4.2. Dummy Database Deployment

To deploy this database, a sql script is provided (“DatabaseDeploy.sql”), which performs the following steps:

1. Creates the database;
2. Create the ETL User to be used in Pentaho;
3. Creates all the tables and all the relationships;
4. Inserts dummy data in the database;
5. Creates the 2 triggers;
6. Creates the 2 views to replicate the sample invoice;
7. Create a Stored Procedure to segment the customers in the categories specified;
8. Runs the Stored Procedure in step 7;

Also, a Jupyter Notebook (DatabaseDeploy/DatabaseGeneration.ipynb) was created to generate dummy data for this database. This script will pick up all the necessary subscripts (Views, Triggers, Stored Procedures, etc), generate brand new dummy data and produce a new “DabaseDeploy.sql” script.