UNPSJB

LICENCIATURA EN SISTEMAS OPGCPI

BASES DE DATOS II

Trabajo Práctico 5

Datawarehouse

Cátedra
Lic. Gabriel Ingravallo
Lic. Cristian Parisse

Integrantes:
Luciano Serruya Aloisi
Pablo Toledo Margalef

2 de julio de 2018

Índice

1. Introducción

Para el siguiente trabajo de laboratorio se solicitaba implementar las bases de datos operacionales y un datawarehouse que satisfaga las necesidades de la empresa "PatSur". Los requerimientos son los siguientes:

> *La cooperativa frutihorticola "PatSur" lleva operando 5 años en la región patagónica. Ha ido creciendo y abriendo sucursales en otras ciudades y participa en la provisión de mercaderías a fabricas, otras cooperativas, pymes, cadenas de supermercados. A medida que fue ampliando su ámbito de compra y venta ha ido modifcando sus sistemas para manejar nuevos datos como producciones, vendedores y compradores, lugares de compra y venta, transportes, empleados, etc. De allí que cada Gerente de área posea distintos datos en diferentes formas. Por lo tanto la primer idea es tener un información de las ventas.*
>
> *La empresa cuenta con dos (2) sistemas distintos de facturación de productos (a partir del año 2010 se implementó un sistema desarrollado por una consultora, mientras que antes del 2010 se utilizaba un sistema desarrollado por el sobrino de uno de los socios de la cooperativa).*

Las implementaciones de todos los sistemas necesarios fueron hechas con contenedores de Docker® y orquestrados con Docker-Compose®; se deja a disposición con el presente informe y todos los *scripts* utilizados el archivo de configuración para levantar todos los contenedores necesarios.

Los cuatro servicios creados por contenedores (un contenedor por servicio) se encuentran dentro de la red **172.40.0.0/16**, cada uno con las siguientes características:

- Sistema de facturación viejo:
  - Imagen: PostgreSQL 9.6
  - Dirección IP: 172.40.0.10
  - Puerto de escucha: 5432
  - Nombre de la base de datos: facturacion
  - Usuario: admin
  - Contraseña: admin
- Sistema de facturación actual

    - Imagen: PostgreSQL 9.6
    - Dirección IP: 172.40.0.20
    - Puerto de escucha: 5432
    - Nombre de la base de datos: facturacion
    - Usuario: admin
    - Contraseña: admin

- Datawarehouse:

    - Imagen: PostgreSQL 9.6
    - Dirección IP: 172.40.0.30
    - Puerto de escucha: 5432
    - Nombre de la base de datos: datawarehouse
    - Usuario: admin
    - Contraseña: admin

- Pentaho *BI Server* :

    - Imagen: Pentaho BI Server
    - Dirección IP: 172.40.0.40
    - Puerto de escucha: 7070

2. Bases de datos operacionales

Los sistemas de facturación de "PatSur" presentan distintos modelados de datos, por lo cual la definición de las tablas de las bases de datos y su correspondiente cargado de datos serán procesos distintos.

Todos los *scipts* a los que se haga referencia se encontrarán en el directorio **"scripts"**. Los *scripts* correspondientes a la carga de datos tanto de los sistemas operacionales como del datawarehouse se encontrarán en el subdirectorio **"scripts/cargas"**

2.1. Sistema viejo

2.1.1. Definición

El primer sistema con el que dispuso la empresa tenía la siguiente estructura de tablas:

- Cliente(**nro_cliente**, nombre, tipo, direccion)
- Categoria(**nro_categ**, descripcion)
- Venta(**nro_factura**, fecha_venta, nro_cliente, nombre, forma_pago)
- Detalle_Venta(**nro_factura**, **nro_producto**, descripcion, unidad, precio)
- Producto(**nro_producto**, nombre, nro_categ, precio_actual)

El *script* correspondiente para la definición de las tablas previamente definidas tiene el nombre **tablas_sistema_viejo.sql**

2.1.2. Poblado

Los siguientes scripts *mencionados se encuentran en el subdirectorio "scripts/cargas/-sistema_viejo" y en "scripts/cargas/sistema_viejo/sql"*

A través de *scripts* escritos en Python (**productos.py** y **categorias.py**), se pudieron extraer datos de la *API* de MercadoLibre [2] para disponer de datos con cierto grado de veracidad. Estos *scripts* se encargan de consultar la fuente de datos, y de armar *scripts* en SQL (**sql/productos** y **categorias.sql**) para la inserción en la tabla correspondiente (adecuando los datos acordemente al modelo de la base de datos)

MercadoLibre maneja un árbol de **categorías** muy amplio y con muchas ramas, de las cuales solamente se exploraron y utilizaron unas pocas para este trabajo. También se obtuvieron **productos** de dichas categorías

```
1 facturacion=# SELECT COUNT(*) FROM categoria;
2  count
3 -------
4     30
5 (1 row)
6
7 facturacion=# SELECT COUNT(*) FROM producto;
8  count
9 -------
```

```
10 ||   1450
11 || (1 row)
```

Cantidad de categorías y productos disponibles en el sistema de facturación viejo

Con respecto a los **clientes**, se utilizó un sitio generador de datos [1] para crear registros de clientes ficticios (**sql/clientes.sql**).

Por último, los archivos **sql/venta.sql** y **sql/detalle_venta.sql** contienen las definiciones de las funciones **insertar_venta(int)** e **insertar_detalle_venta(int)**. Ambas funciones reciben un valor entero como parámetro, que indica la cantidad de registros que se quieren crear (tomando valores aleatorios de la base). Estas funciones son acumulativas, entonces se pueden correr cuantas veces se desee para poblar las tablas de ventas y detalles de ventas.

```
1 || facturacion=# SELECT insertar_ventas(50);
2 || NOTICE:  [1/50] INSERTANDO VENTA (2007-10-03, 862512, 4118, Linda
  ||    Acosta, EFECTIVO)
3 || NOTICE:  [2/50] INSERTANDO VENTA (2002-07-29, 143370, 3458, Jeremy
  ||    Pacheco, CREDITO)
4 || NOTICE:  [3/50] INSERTANDO VENTA (2006-02-15, 993978, 2414, Lisandra
  ||     Bernard, DEBITO)
5 || ...
```

Inserción de 50 ventas aleatorias en la BD

```
1 || facturacion=# SELECT insertar_detalle_venta(300);
2 || NOTICE:  [1/300] INSERTANDO DETALLE (333538, 639930615, 19, 169.99)
3 || NOTICE:  [2/300] INSERTANDO DETALLE (743695, 644739590, 41, 72)
4 || NOTICE:  [3/300] INSERTANDO DETALLE (124334, 712205536, 54, 419500)
5 || ...
```

Inserción de 300 detalles de ventas aleatorias en la BD

2.2. Sistema actual

2.2.1. Definición

El sistema desarrollado en el 2010 por la consultora cuenta con el siguiente esquema:

- Cliente(**cod_cliente**, nombre, cod_tipo, direccion)

- Tipo_Cliente(**cod_tipo**, descripcion)
- Producto(**cod_producto**, nombre, cod_categoria, cod_subcategoria, precio_actual)
- Categoria(**cod_categoria**, **cod_subcategoria**, descripcion)
- Venta(**id_factura**, fecha_venta, cod_cliente, nombre, cod_medio_pago)
- Detalle_Venta(**id_factura**, **cod_producto**, descripcion, unidad, precio)
- Medio_Pago(**cod_medio_pago**, descripcion, valor, unidad, tipo_operacion)

El *script* correspondiente para la definición de las tablas previamente definidas tiene el nombre **tablas_sistema_actual.sql**

2.2.2. Poblado

Los *scripts* realizados para la carga de datos de este sistema son idénticos a los del sistema anterior, adaptándolos a los distintas representaciones de los datos. Para los **productos** y **categorías**, también se extrajeron datos de MercadoLibre, agregando como valor de *subcategoría* un valor secuencial entre 1 y 5 por cada categoría (es decir, cada categoria tiene 5 subcategorías secuenciales)

```
facturacion=# SELECT COUNT(*) FROM categoria;
 count
-------
   150
(1 row)

facturacion=# SELECT COUNT(*) FROM producto;
 count
-------
  1454
(1 row)
```

Cantidad de categorías y productos disponibles en el sistema de facturación actual

La carga de **clientes** es igual (**sql/clientes.sql**), teniendo en cuenta cargar clientes repetidos para después figuren en las *tablas de equivalencia* del datawarehouse.

Varios **medios de pago** fueron registrados mediante el *script* **sql/medio_pago.sql**, al igual que **tipos de cliente** (**sql/tipo_cliente.sql**)

Las funciones para creaciones de **ventas** y **detalles de ventas** mantienen la misma firma que las implementadas en el sistema viejo (**sql/venta.sql** y **sql/detalle_venta.sql**, respectivamente).

```
facturacion=# SELECT insertar_ventas(15);
NOTICE:  [1/15] INSERTANDO VENTA (2015-07-25, 119357, 252269, Ifeoma
    G. Collins, Débito)
NOTICE:  [2/15] INSERTANDO VENTA (2014-07-31, 585847, 967234, Keane
   K. Hodge, Bitcoin)
NOTICE:  [3/15] INSERTANDO VENTA (2013-05-01, 501180, 639678, Edward
    T. Carey, Crédito)
...
```

Inserción de 15 ventas aleatorias en la BD

```
facturacion=# SELECT insertar_detalle_venta(100);
NOTICE:  [1/100] INSERTANDO DETALLE (462363, 678175295, 10, 2249)
NOTICE:  [2/100] INSERTANDO DETALLE (660157, 655324667, 77, 340)
NOTICE:  [3/100] INSERTANDO DETALLE (164479, 612842726, 38, 8499)
...
```

Inserción de 100 detalles de ventas aleatorias en la BD

3. Datawarehouse

3.1. Marco teórico

Un Datawarehouse o almacén de datos es definido por [3] como *una colección de datos orientada al sujeto, integrada, no volátil y de tiempo variable para el soporte de las decisiones de los directivos*

3.2. Definición

3.3. Poblado

3.3.1. *ETL*

3.4. Explotación

4. Herramienta de *BI*: *Pentaho*

Referencias

[1] Benjamin Keen. *Sitio generador de datos*. 2018. URL: http://www.generatedata.com/.

[2] MercadoLibre. *API REST de MercadoLibre*. Jul. de 2018. URL: http://developers.mercadolibre.com/categories-and-listings/.

[3] Ramez Elmasri; Shamkant B. Navathe. *Fundamentos de Sistemas de Bases de Datos*. Quinta edición. Pearson Education, 2007.