UNIVERSIDAD LUTERANA SALVADOREÑA

FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN.

TÍTULO O TEMA:

SISTEMA CVI.

INTEGRANTES:

Álvaro Ernesto Castillo Cornejo CC01135826

José Elías Majano Lobato ML01135758

Miguel Alejandro Torres García TG01135889

Ángel Noé Rivera Rivas RR01135891

ASIGNATURA:

Base de datos II.

DOCENTE:

Pedro Noble.

CICLO/AÑO:

II/ 2021

FECHA:

4/12/2021

Video del proyecto:

https://www.youtube.com/watch?v=hLKUj2BgArw

INDICE

INTRODUCCION

En el siguiente proyecto se realizará una base de datos, esta será capaz de almacenar información de forma ordenada la cual otorgará accesibilidad de consultas que puede generar el usuario acerca de la disponibilidad de productos que se almacenen en ella.

La característica principal de esta base de datos es el manejo masivo de información de forma eficiente y eficaz para el que administra la base de datos como para el que la consulta.

Dicha base de datos será una herramienta muy útil ya que podrán llevar un mejor control tanto de sus compras, ventas e inventarios.

OBJETIVOS

OBJETIVO GENERAL

- Crear una base de datos local ordenada y precisa para diferentes almacenes.

OBJETIVOS ESPECÍFICOS

- Facilitar al usuario el manejo de sus datos, de una forma eficaz, fluida y veraz.
- Practicar las diferentes estructuras de una base de datos.
- Crear consultas multidimensionales efectivas para la utilización de la base de datos.

JUSTIFICACION

El presente proyecto se enfocará en elaborar un gestor de base de datos como estudiantes de la materia de Bases de Datos dos, estamos convencidos que la elaboración de diagrama entidad relación, modelo lógico relacional; así como también hacer consultas multidimensionales aprovechando la creación de cubos esto nos brindará la oportunidad de desarrollar todo tipo de soluciones con una sintaxis clara y visual. Nos proponemos entonces realizar un modelo de gestión de bases de datos para un sistema de compras y ventas para un almacén.

PLANTEAMIENTO DE LA SITUACIÓN

Diseñar y elaborar una base de datos, para brindar una solución efectiva para un sistema de compras y ventas para un almacén por la demanda en la adquisición y venta de productos, se hace muy difícil e ineficaz llevar un control de todos los productos en existencia y productos agotados, no cuenta con un registro digital, con este sistema se llevará un registro por categorías; así como también para que el gerente y vendedores del almacén se le facilite la búsqueda de productos y llevar un inventario ordenado; así mismo saber las ganancias en un periodo determinado y saber los productos con mayor demanda, obteniendo información actualizada para ser más eficiente y brindar una mejor atención a los clientes; Y así mismo obtener datos en tiempo real.

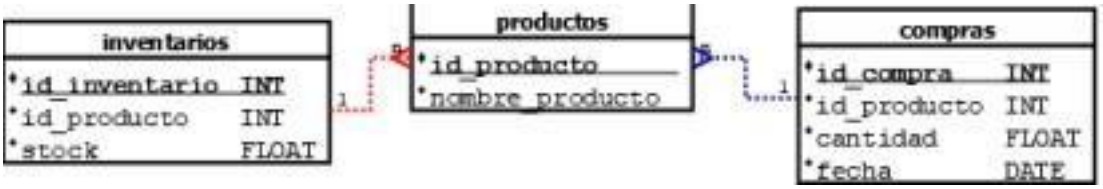
REQUERIMIENTOS

1. **Listar todas la compras y la cantidad con el stock disponbible de este año.**

SQL: SELECT productos.nombre_producto, inventarios.stock, sum(compras.cantidad) AS 'cantidad' FROM compras

INNER JOIN productos ON productos.id_producto = compras.id_producto

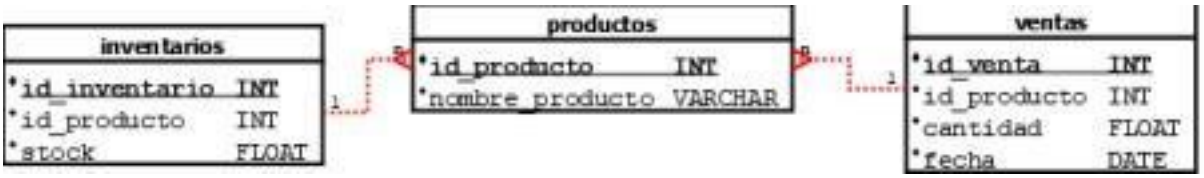
INNER JOIN inventarios ON productos.id_producto = inventarios.id_producto WHERE compras.fecha BETWEEN '2021-01-01' AND '2021-12-31';



2. **Listas todas las ventas y la cantidad vendida con el stock disponible en este año.**

SQL: SELECT productos.nombre_producto, inventarios.stock, sum(ventas.cantidad) AS 'cantidad' FROM ventas

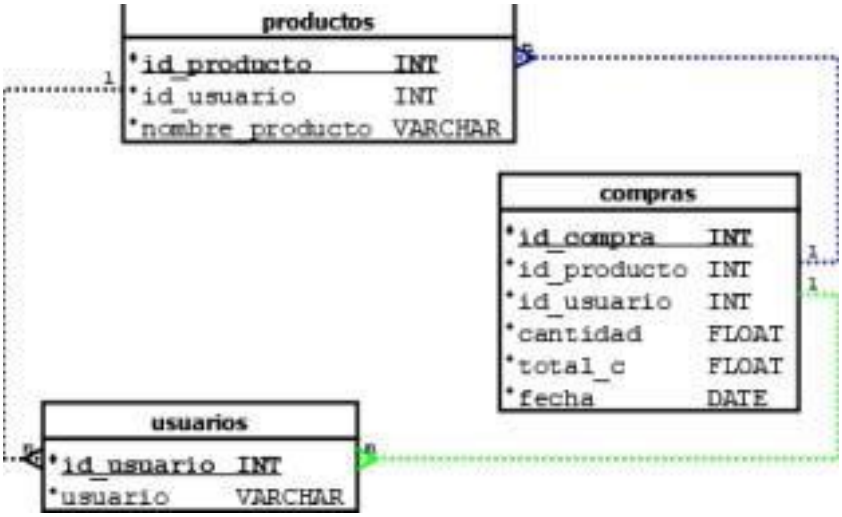
INNER JOIN productos ON productos.id_producto = ventas.id_producto INNER JOIN inventarios ON productos.id_producto = inventarios.id_producto WHERE ventas.fecha BETWEEN '2021-01-01' AND '2021-12-31';



3. **Listar todas las compras realizadas por un usuario en específico en este año.**

SQL: SELECT compras.id_compra, productos.nombre_producto, usuarios.usuario, compras.cantidad, compras.total_c, compras.fecha FROM compras INNER JOIN productos ON productos.id_producto = compras.id_producto INNER JOIN usuarios ON usuarios.id_usuario = compras.id_usuario

WHERE compras.fecha BETWEEN '2021-01-01' AND '2021-12-31' AND compras.id_usuario = '4';



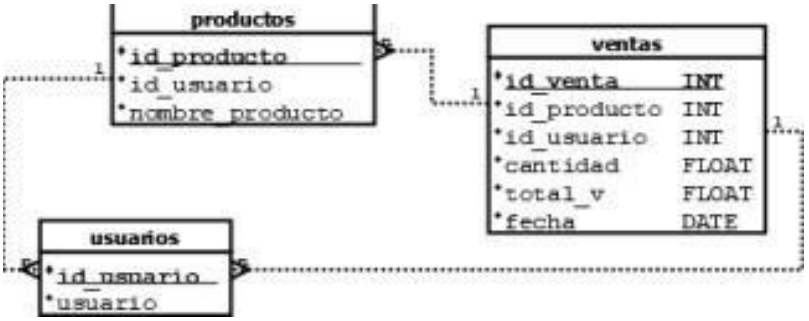
4. **Listar todas las ventas realizas de un usuario en específico en este año.**

SQL: SELECT ventas.id_venta, productos.nombre_producto, usuarios.usuario, ventas.cantidad, ventas.total_v, ventas.fecha FROM ventas

INNER JOIN productos ON productos.id_producto = ventas.id_producto INNER JOIN usuarios

ON usuarios.id_usuario = ventas.id_usuario

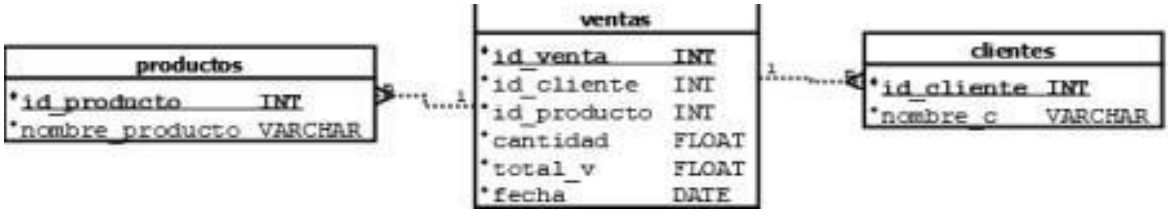
WHERE ventas.fecha BETWEEN '2021-01-01' AND '2021-12-31' AND ventas.id_usuario = '4';



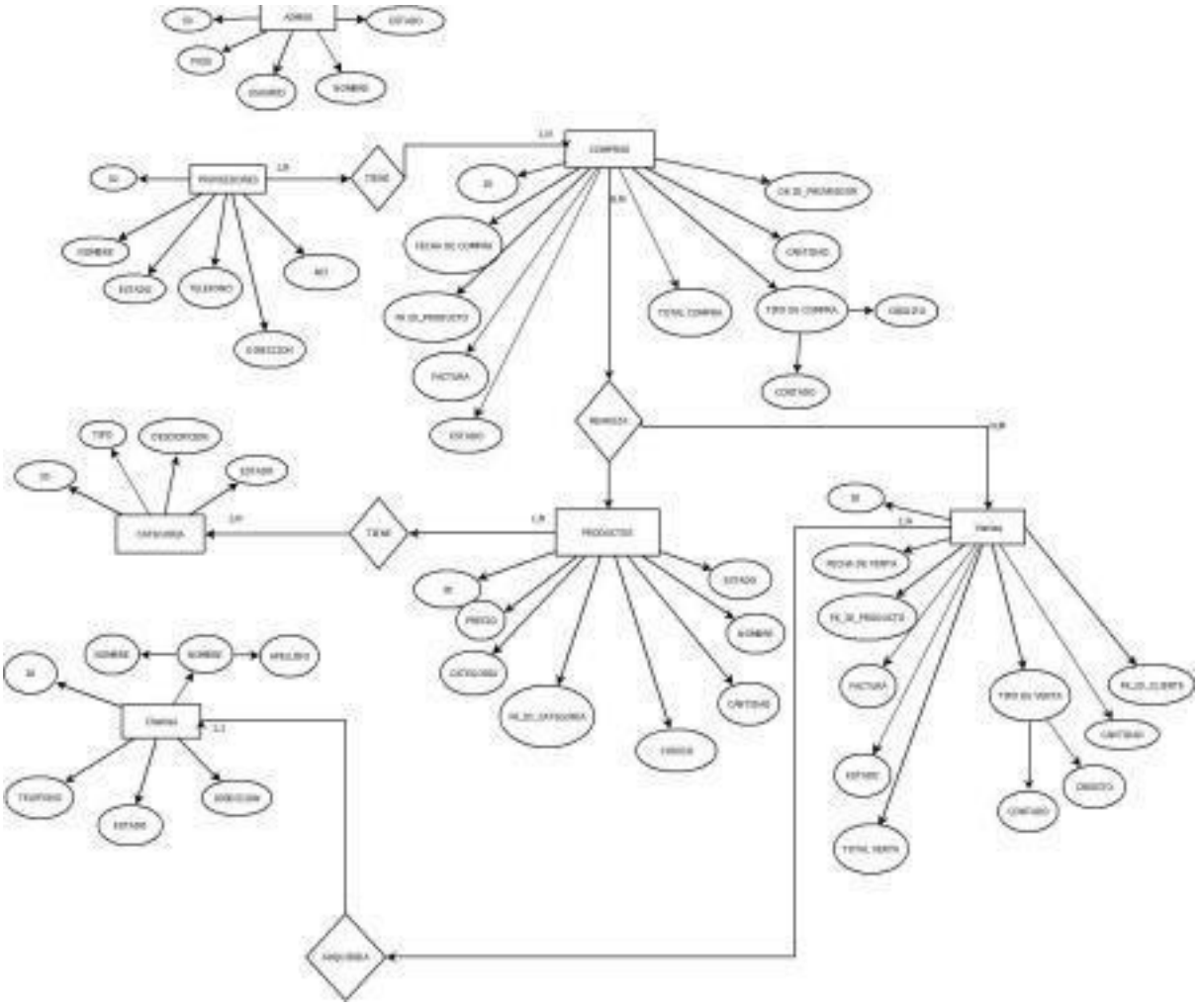
5. **Listar todas las ventas de un cliente en específico durante ese año.**

SQL: SELECT ventas.id_venta, productos.nombre_producto, clientes.nombre_c, ventas.cantidad, ventas.total_v, ventas.fecha FROM ventas

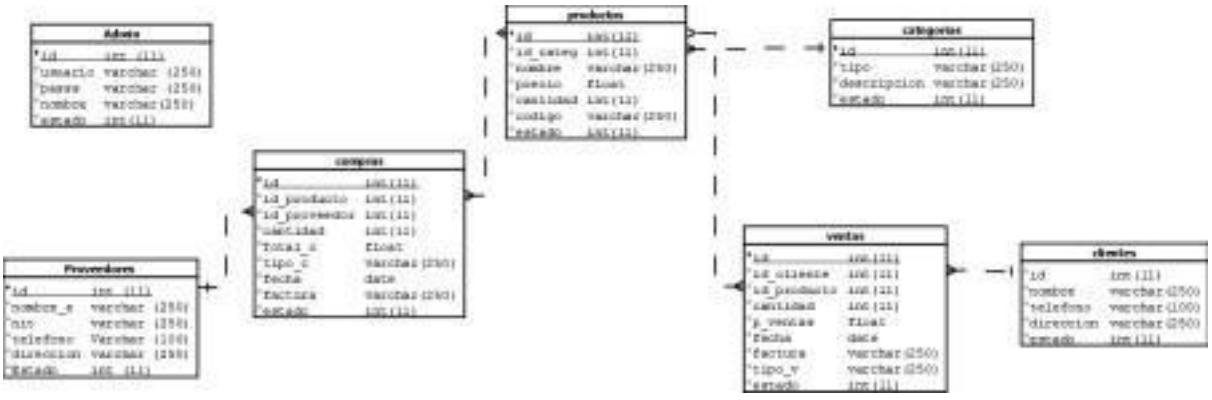
INNER JOIN productos ON productos.id_producto =
ventas.id_producto INNER JOIN clientes ON clientes.id_cliente = ventas.id_cliente
WHERE ventas.fecha BETWEEN '2021-01-01' AND '2021-12-31';



Diseño de base de datos (diagrama ER, Modelo lógico).



Modelo lógico base de datos



TECNOLOGIAS APLICADAS AL PROYECTO

MySQL.

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle.

Las versiones Enterprise, diseñadas para aquellas empresas que quieran incorporarlo en productos privativos, incluyen productos o servicios adicionales tales como herramientas de monitorización y asistencia técnica oficial. (Robledano, 2019, pág. 1)

Uso en el proyecto.

Esta herramienta es utilizada a la hora de crear nuesta base de datos y sus tablas.

XAMPP.

XAMPP es una herramienta de desarrollo que te permite probar tu desarrollo web basado en PHP en tu propio ordenador sin necesidad de tener acceso a internet. es una distribución de Apache que incluye diferentes softwares libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen: Linux, Apache, MySQL/MariaDB, PHP y Perl. (Garcia, 2020, pág. 22)

Uso en el proyecto.

Con XAMPP mostramos la parte grafica de nustra base de datos y sus consultas.

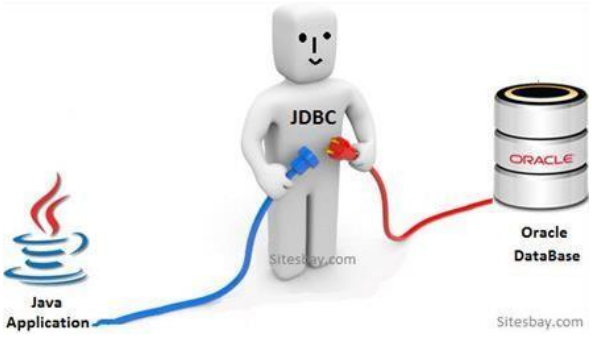
PHPMyAdmin.

PHPMyAdmin es un software de código abierto, diseñado para manejar la administración y gestión de bases de datos MySQL a través de una interfaz gráfica de usuario. Escrito en PHP, se ha convertido en una de las más populares herramientas basadas en web de gestión de Base de datos, viene con una documentación detallada y está siendo apoyado por un gran multi-idioma de la comunidad. (Cahuana, 2020, pág. 35)

Uso en el proyecto.

Hacemos uso de esta erramienta para administrar la base de datos ya creada.

Java connector.



Es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice. (Dorado, 2005).

Uso en el proyecto.

Con Java connector se hace posible la conexión de la base de datos.

Pentaho.

Pentaho BI Suite es una herramienta de Business Intelligence que extrae y analiza datos con técnicas ETL (extraer, transformar y cargar, en sus siglas en inglés). Estos datos se muestran a posteriori en Cuadros de Mandos, que serán muy útiles para crear informes y tener un seguimiento de la consecución de objetivos. En el caso de Pentaho, estamos ante un software open source que cuenta con herramientas de big data y IoT. (Brinquis, 2020) **Uso en el proyecto.**

Importaremos nuestra base de datos a la herramienta pentaho para darle estructura y crear los cubos necesarios.

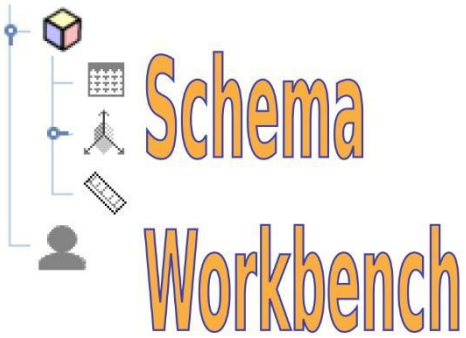
Mondrian.

Mondrian es un proyecto Open Source, licenciado bajo la Mozilla Public License (MPL). Esta licencia es una de las "Business Friendly" lo cual implica que es de las menos restrictivas para su uso desde la mayor parte de los puntos de vista (al igual que el resto de la suite de Pentaho), permitiendo Modificar, Embeber, Modularizar, el software sin restricciones; dejando al parecer de la organización el aporte o no de los cambios realizados al proyecto. (Rodríguez, 2005, pág. 59)

Uso en el proyecto.

Con este servidor creamos nuestros cubos OLAP.

Schema Workbench.



Es una herramienta desarrollada con java con altas capacidades de edición y customización, que facilita la creación y corrección de los ficheros XML que componen el esquema de Mondrian a desarrollar.

Schema Workbench es una herramienta Java que permite crear, modificar, testear y guardar esquemas OLAP de Mondrian. (Intelligence, 2017).

Uso en el proyecto.

Esta herramienta guarda nuestros cubos OLAP creados.

Transact-SQL ó T-SQL.

Es un lenguaje muy potente que permite definir casi cualquier tarea que se quiera efectuar sobre la base de datos, va más allá de un lenguaje SQL cualquiera ya que incluye características propias de cualquier lenguaje de programación, características que permiten definir la lógica necesaria para el tratamiento de la información.

(contributors, 2021).

Usaremos transact-SQL para hacer pruebas con nuestra base de datos y hacer consultas multitablas.

DBeaver.

Es un software que actúa como una herramienta de base de datos universal destinada a desarrolladores y administradores de bases de datos. DBeaver tiene una interfaz de usuario bien diseñada, la plataforma basada en un marco de código abierto y permite escribir múltiples extensiones, así como también es compatible con cualquier base de datos. (Darkcrizt, 2019)

Esta herramienta la usaremos para administrar nuestra base de datos y crear los disparadores

ANEXOS

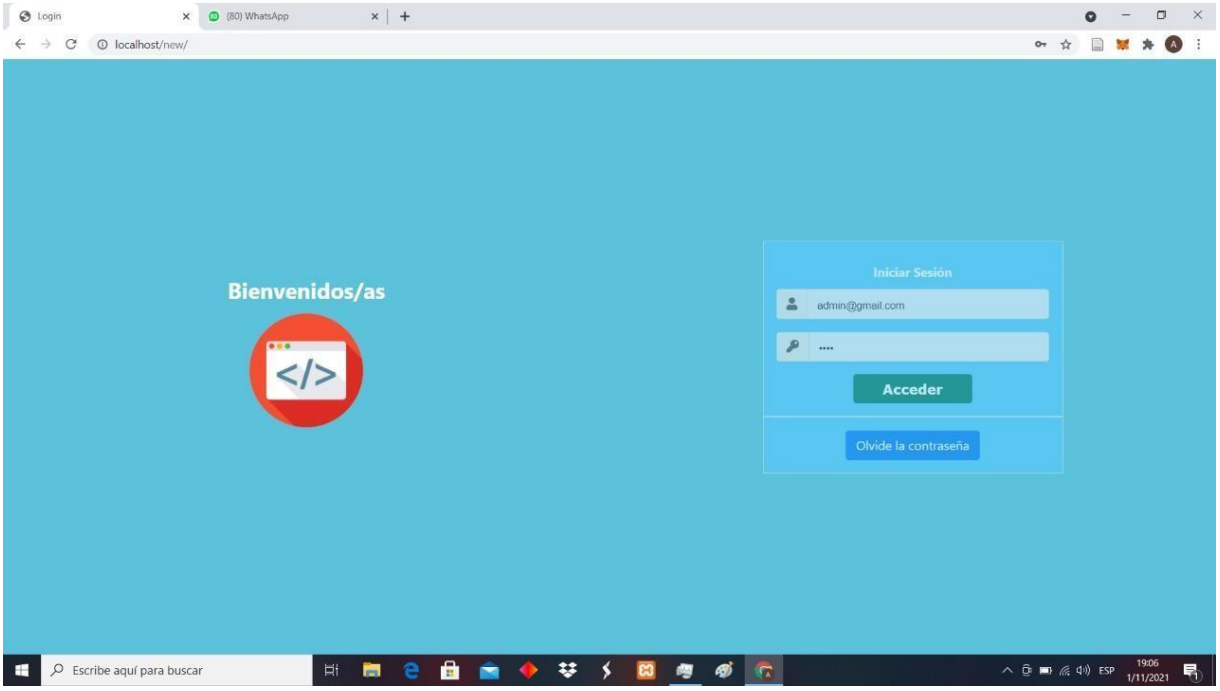
Video del proyecto:

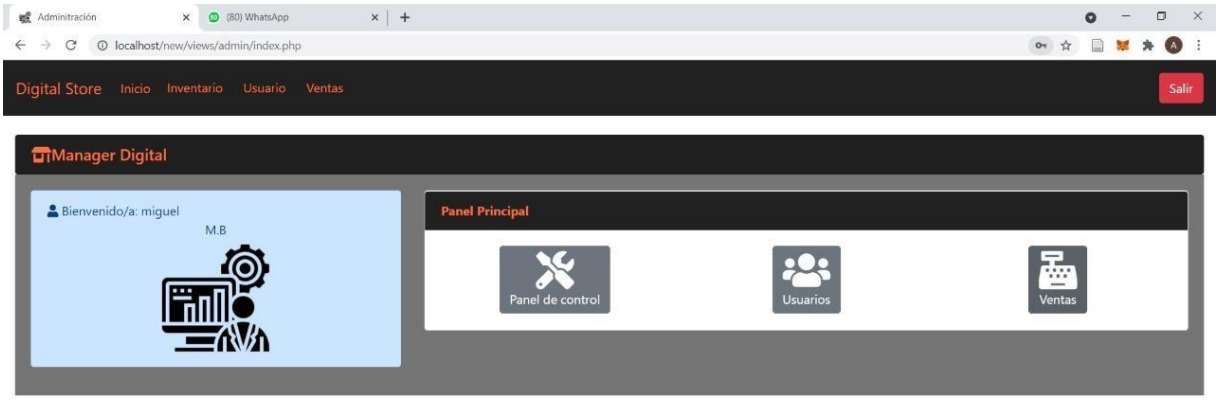
https://www.youtube.com/watch?v=hLKUj2BgArw

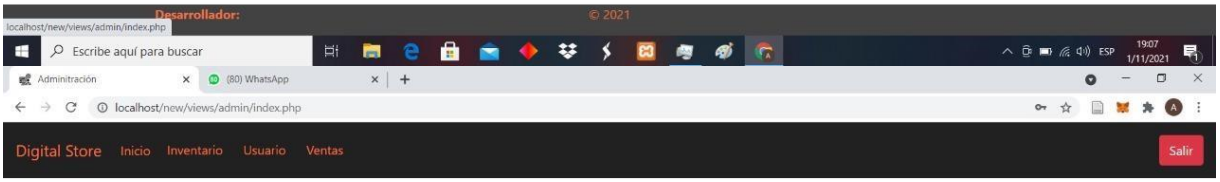
Enlace al proyecto:

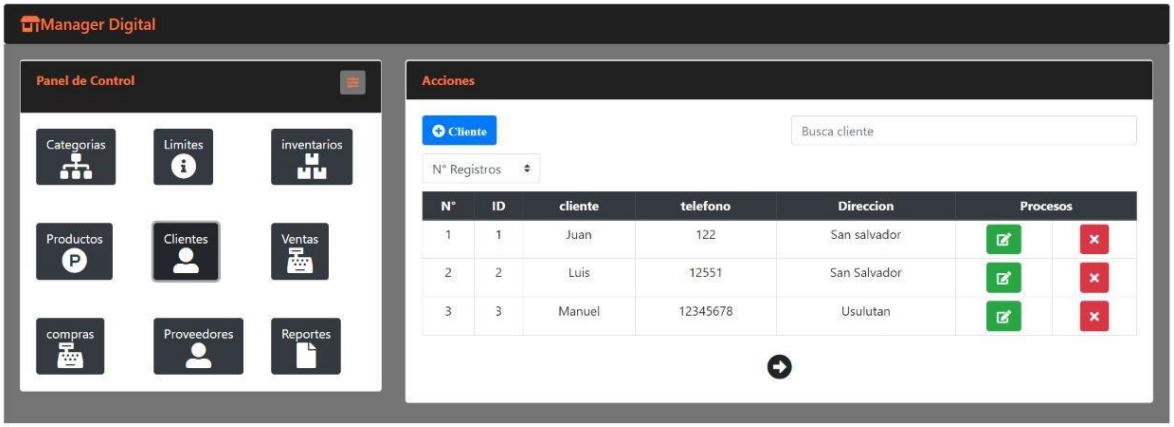
https://drive.google.com/file/d/1rTBAzjt637Iwn00wcNiQRZO24HMigVsN/view?usp=sharing

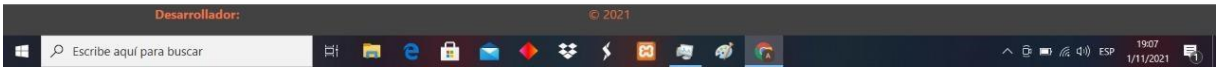
Sistema en funcionamiento

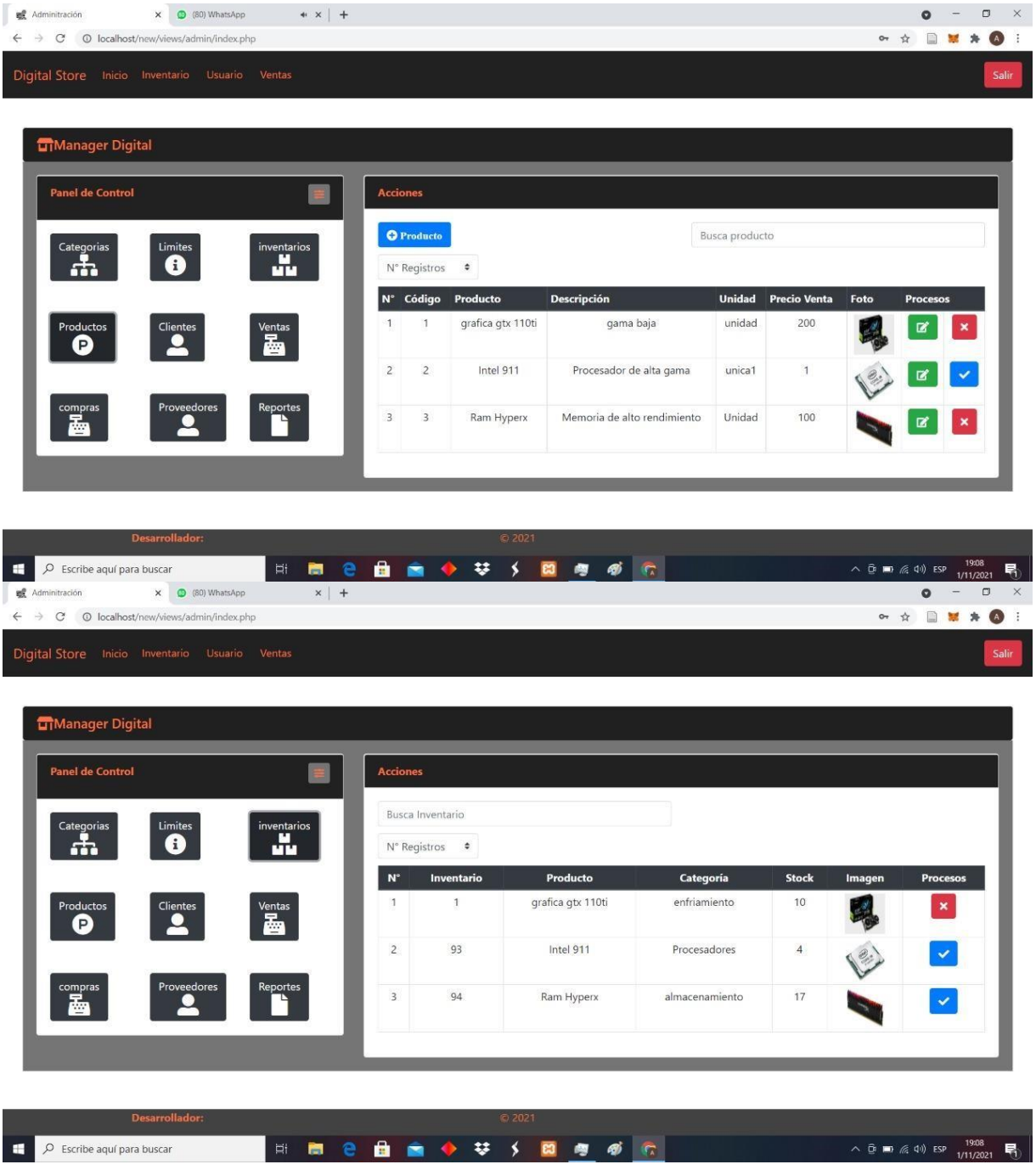



Adminitración
(80) WhatsApp
localhost/new/views/admin/index.php
Digital Store
Inicio
Inventario
Usuario
Ventas
Salir
Manager Digital
Bienvenido/a: miguel
M.B
Panel Principal
Panel de control
Usuarios
Ventas

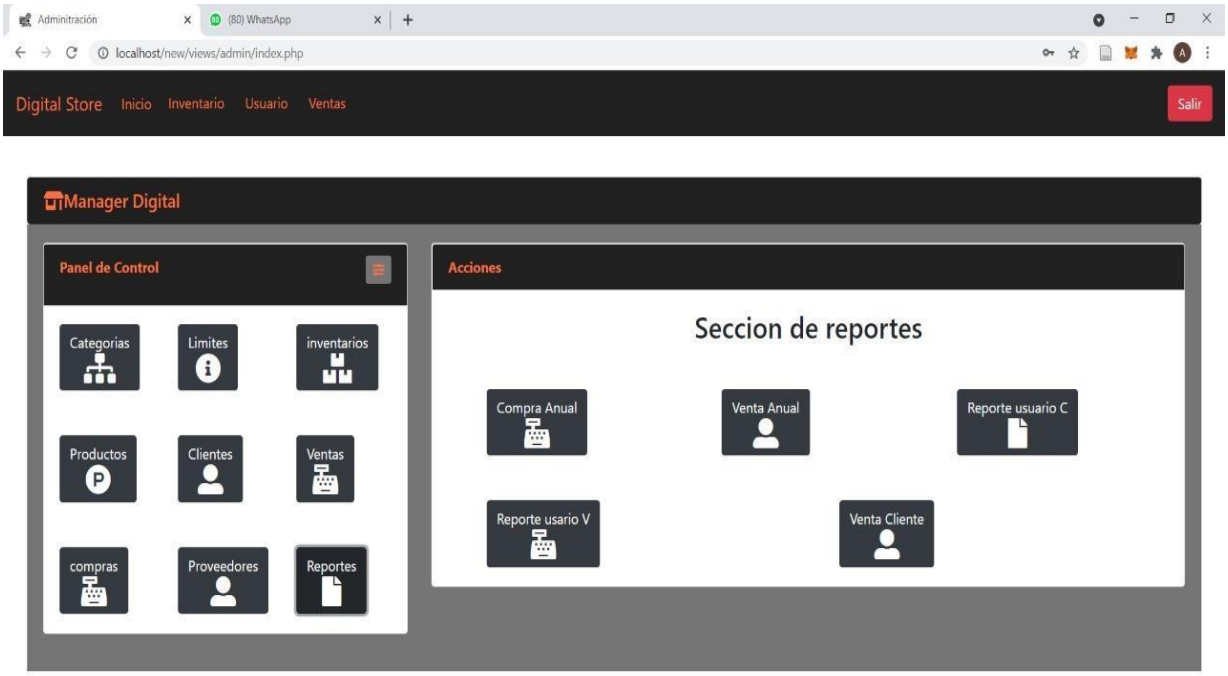

Desarrollador:
© 2021
localhost/new/views/admin/index.php
Escribe aquí para buscar
ESP
19:07
1/11/2021
Adminitración
(80) WhatsApp
localhost/new/views/admin/index.php
Digital Store
Inicio
Inventario
Usuario
Ventas
Salir

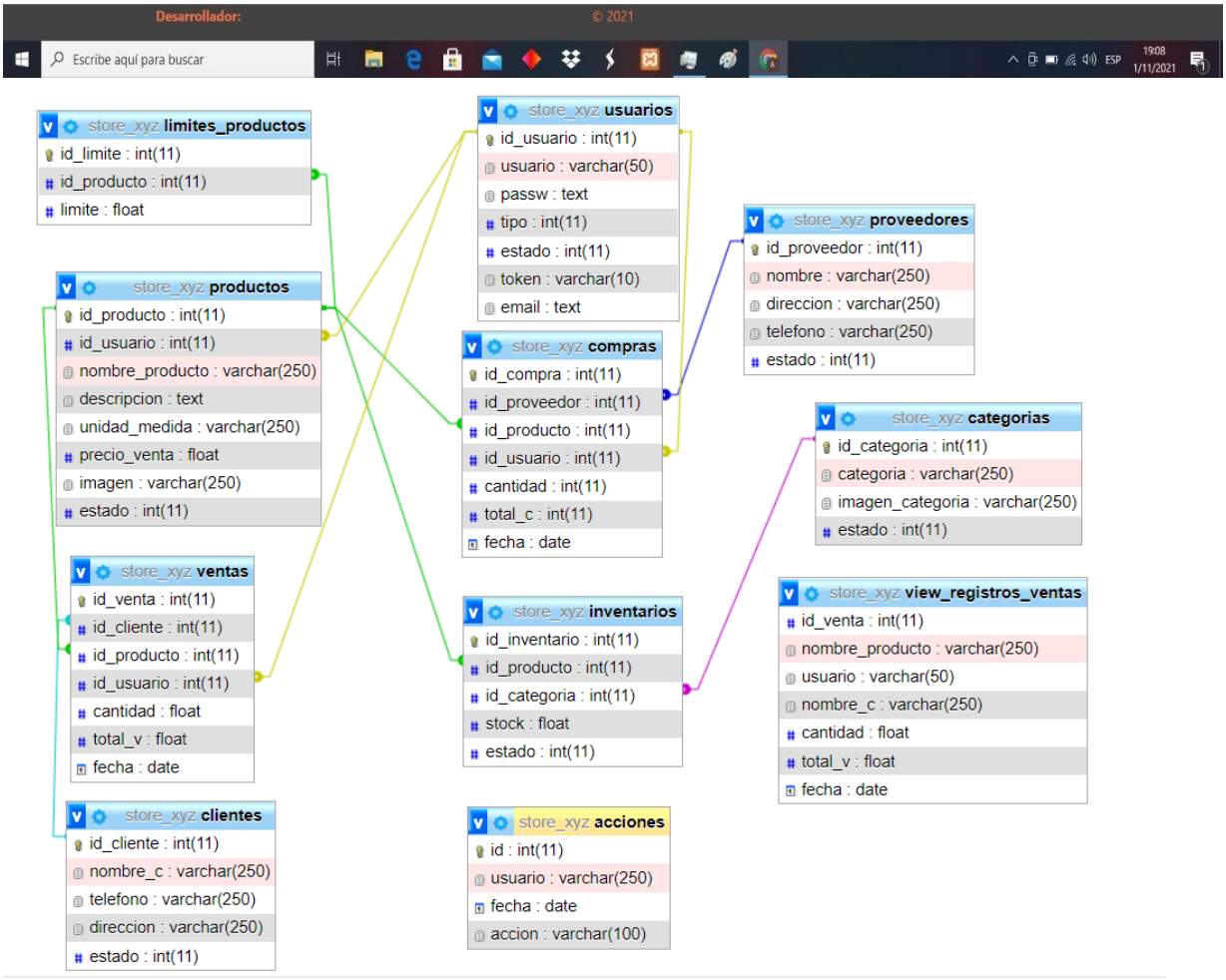

Manager Digital
Panel de Control
Categorias
Limites
inventarios
Productos
Clientes
Ventas
compras
Proveedores
Reportes
Acciones
Cliente
Busca cliente
N° Registros
N°
ID
cliente
telefono
Direccion
Procesos
1
1
Juan
122
San salvador
2
2
Luis
12551
San Salvador
3
3
Manuel
12345678
Usulutan


Desarrollador:
© 2021
Escribe aquí para buscar
ESP
19:07
1/11/2021


Adminitración
(80) WhatsApp
localhost/new/views/admin/index.php
Digital Store
Inicio
Inventario
Usuario
Ventas
Salir
Manager Digital
Panel de Control
Categorias
Limites
inventarios
Productos
Clientes
Ventas
compras
Proveedores
Reportes
Acciones
Producto
Busca producto
N° Registros
N°
Código
Producto
Descripción
Unidad
Precio Venta
Foto
Procesos
1
1
grafica gtx 110ti
gama baja
unidad
200
2
2
Intel 911
Procesador de alta gama
unica1
1
3
3
Ram Hyperx
Memoria de alto rendimiento
Unidad
100
Desarrollador:
© 2021
Escribe aquí para buscar
ESP
19:08
1/11/2021

Adminitración
(80) WhatsApp
localhost/new/views/admin/index.php
Digital Store
Inicio
Inventario
Usuario
Ventas
Salir
Manager Digital
Panel de Control
Categorias
Limites
inventarios
Productos
Clientes
Ventas
compras
Proveedores
Reportes
Acciones
Busca Inventario
N° Registros
N°
Inventario
Producto
Categoria
Stock
Imagen
Procesos
1
1
grafica gtx 110ti
enfriamiento
10
2
93
Intel 911
Procesadores
4
3
94
Ram Hyperx
almacenamiento
17
Desarrollador:
© 2021
Escribe aquí para buscar
ESP
19:08
1/11/2021


Administración
(80) WhatsApp
localhost/new/views/admin/index.php
Digital Store
Inicio
Inventario
Usuario
Ventas
Salir
Manager Digital
Panel de Control
Categorias
Limites
inventarios
Productos
Clientes
Ventas
compras
Proveedores
Reportes
Acciones
Seccion de reportes
Compra Anual
Venta Anual
Reporte usuario C
Reporte usuario V
Venta Cliente


Desarrollador:
© 2021
Escribe aquí para buscar
ESP
19:08
1/11/2021
store_xyz limites_productos
id_limite : int(11)
id_producto : int(11)
limite : float
store_xyz usuarios
id_usuario : int(11)
usuario : varchar(50)
passw : text
tipo : int(11)
estado : int(11)
token : varchar(10)
email : text
store_xyz proveedores
id_proveedor : int(11)
nombre : varchar(250)
direccion : varchar(250)
telefono : varchar(250)
estado : int(11)
store_xyz productos
id_producto : int(11)
id_usuario : int(11)
nombre_producto : varchar(250)
descripcion : text
unidad_medida : varchar(250)
precio_venta : float
imagen : varchar(250)
estado : int(11)
store_xyz compras
id_compra : int(11)
id_proveedor : int(11)
id_producto : int(11)
id_usuario : int(11)
cantidad : int(11)
total_c : int(11)
fecha : date
store_xyz categorias
id_categoria : int(11)
categoria : varchar(250)
imagen_categoria : varchar(250)
estado : int(11)
store_xyz ventas
id_venta : int(11)
id_cliente : int(11)
id_producto : int(11)
id_usuario : int(11)
cantidad : float
total_v : float
fecha : date
store_xyz inventarios
id_inventario : int(11)
id_producto : int(11)
id_categoria : int(11)
stock : float
estado : int(11)
store_xyz view_registros_ventas
id_venta : int(11)
nombre_producto : varchar(250)
usuario : varchar(50)
nombre_c : varchar(250)
cantidad : float
total_v : float
fecha : date
store_xyz clientes
id_cliente : int(11)
nombre_c : varchar(250)
telefono : varchar(250)
direccion : varchar(250)
estado : int(11)
store_xyz acciones
id : int(11)
usuario : varchar(250)
fecha : date
accion : varchar(100)

Despliegue de los cubos OLAP Pentaho

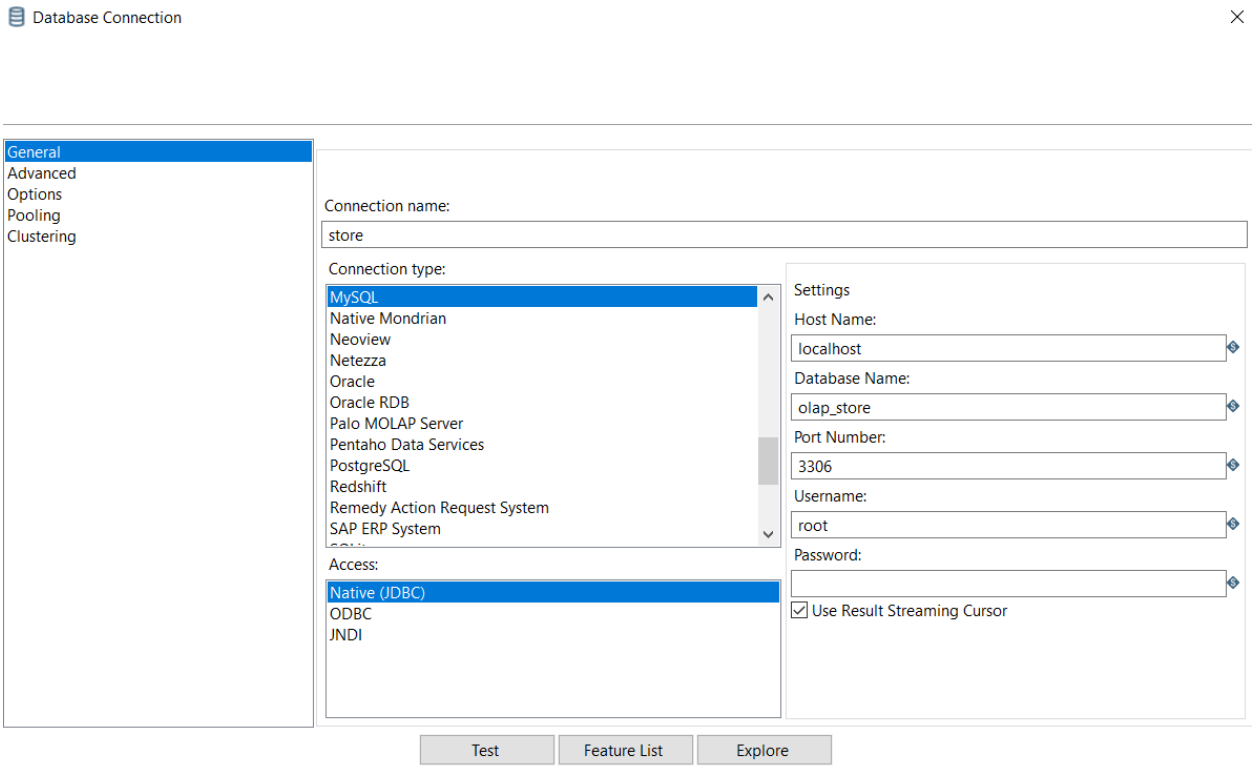
Requisitos:

Tener la base de datos donde se cargaran los datos. Ejemplo “Sistema OLAP”

Tener las tablas creadas para recibir los datos transformados del Pentaho Data Integration

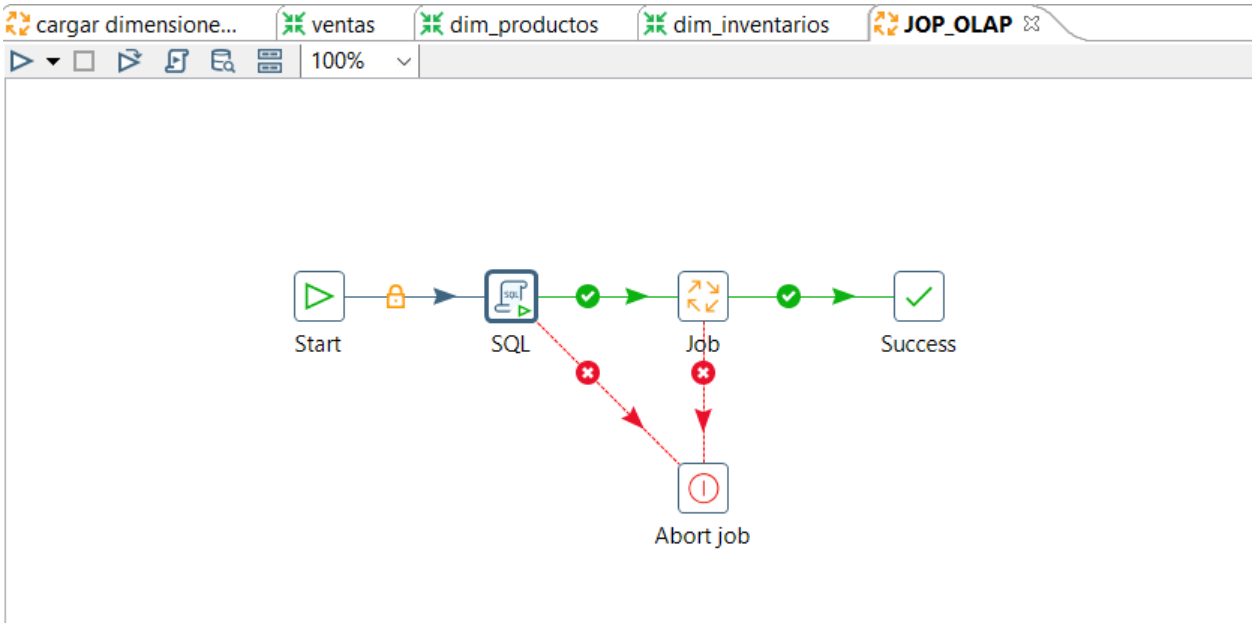
1. **Creación de los JOBS.**
   Se debe crear un JOB principal y el otro para cargar los datos.

   El JOB principal tendrá la función de realizar los pasos necesarios para realizar todo el proceso para eliminar registros y cargar los datos a la base de datos designados.

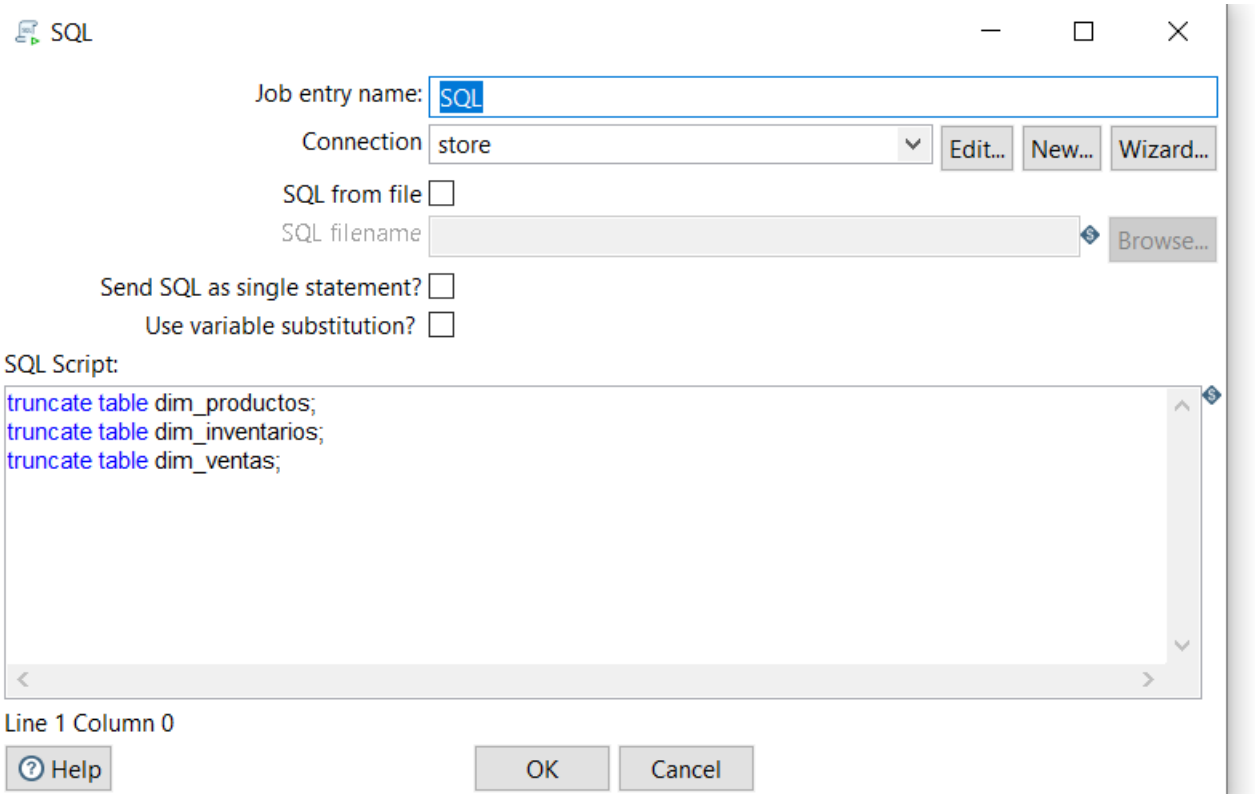


Antes que todo tenemos que tener conectado nuestro Pentaho conectado a nuestra base de datos a la cual vamos a utilizar.

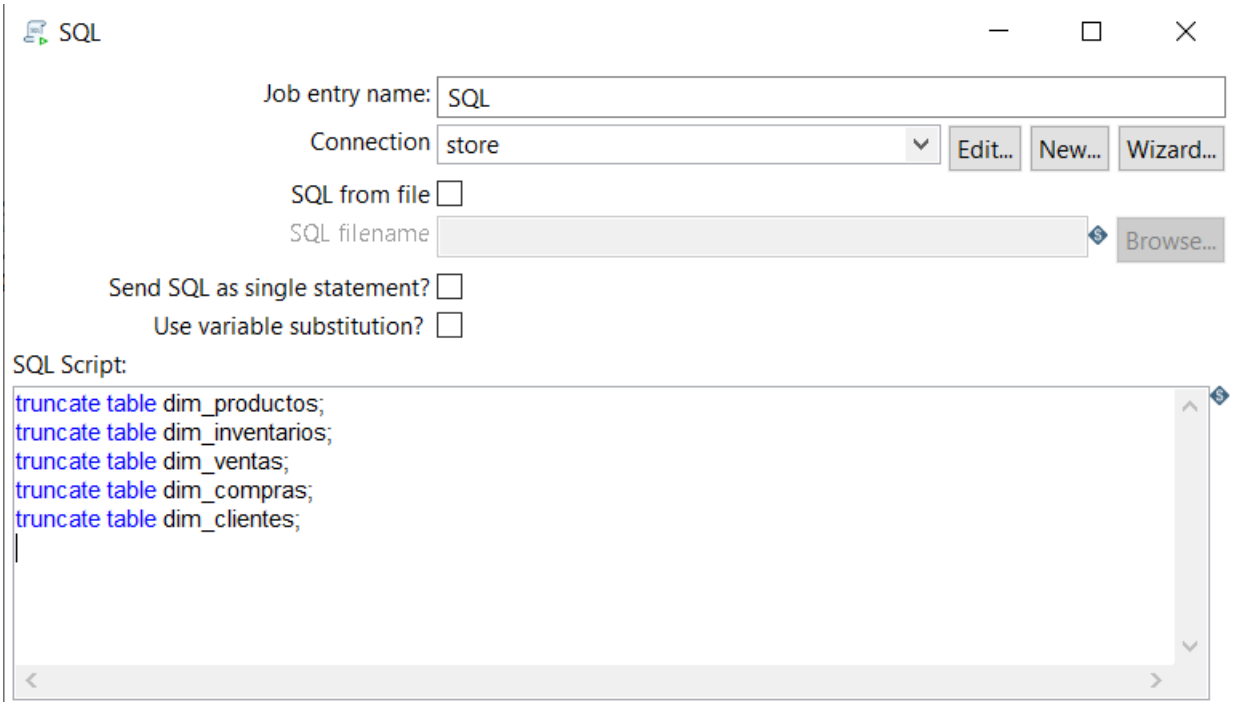
JOB principal:



En este JOB haremos la siguiente estructura para el proceso principal.

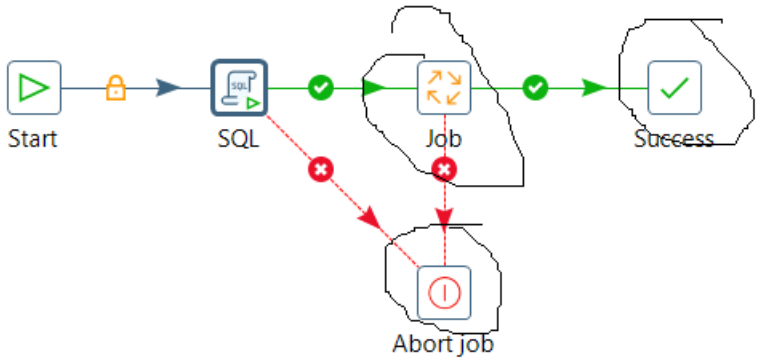


En el módulo SQL, vamos a realizar un truncate de las tablas donde insertaremos los datos la cual está conectada a nuestra base de datos específica.
En este caso voy a agregar un truncate a la tabla dim_compras y dim_clientes.

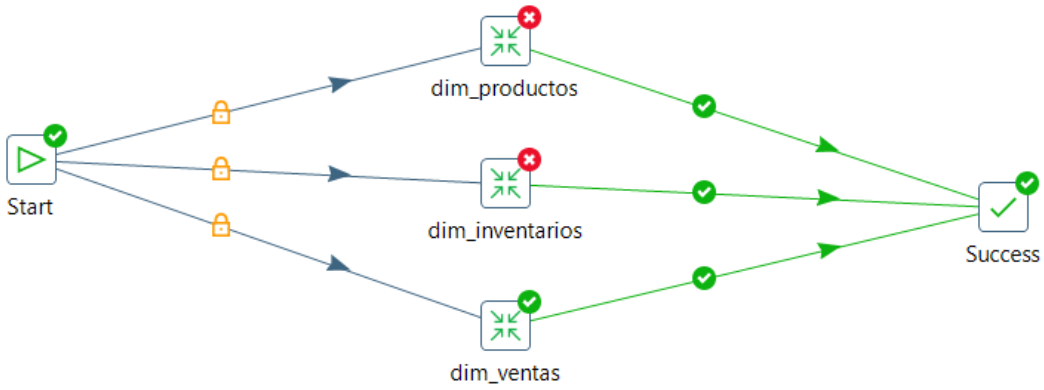


Luego de eso se coloca otro modulo para que cargue el JOB secundario, un abort Job y un módulo sucesse.

EL modulo JOB sirve para mandar a llamar otro JOB precreado parta ejecutar sus acciones.
El Abort Job es para abortar el JOB si encuentra un problema.
EL módulo Succes sirve para dar una operación exitosa si se ejecuta conexito lo establecido.

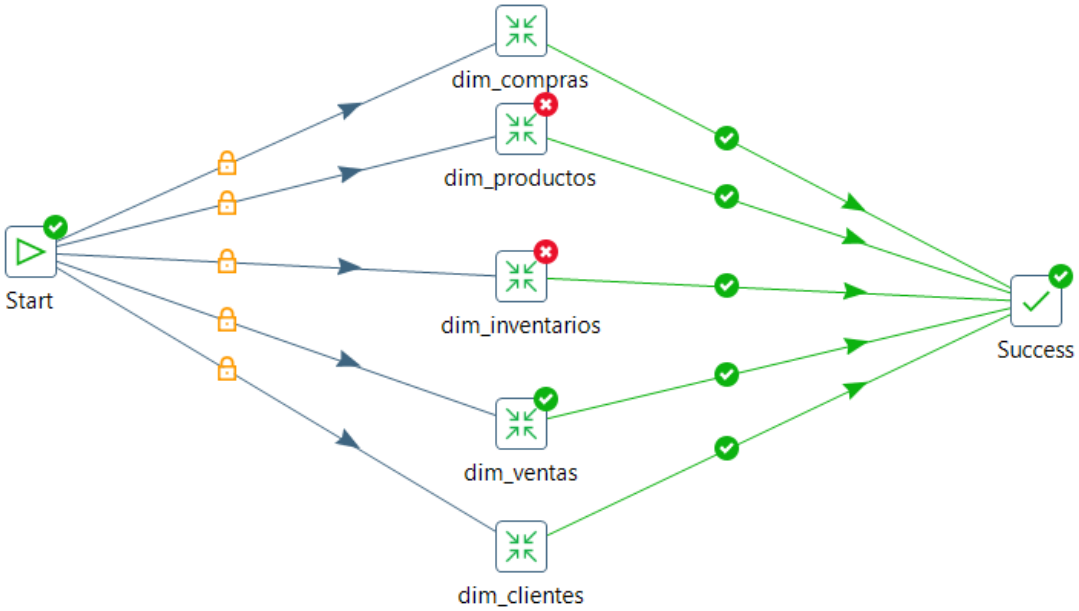


2. Creación de JOB de cargar de los datos.



Podemos usar el siguiente ejemplo para usar este JOB para hacer la carga de los de la base de datos originaria hacia la base de datos OLAP.

En este caso agregare las otras dos dimensiones (transformaciones) faltantes.

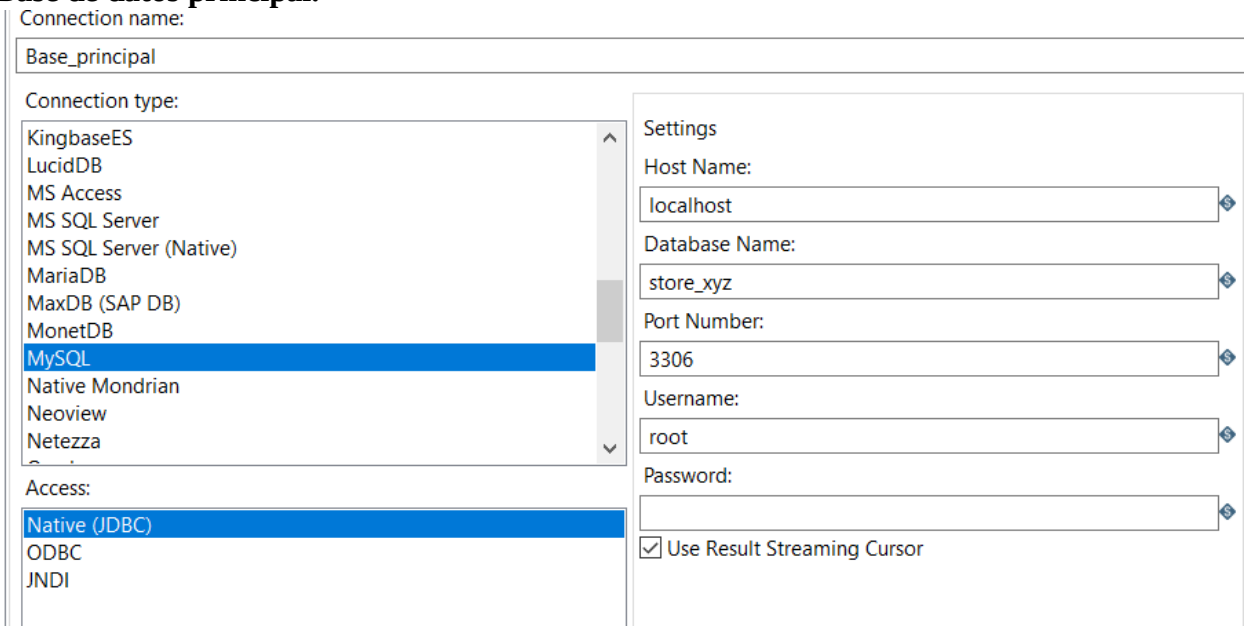


3. **Crear la transformación de la dimensión.**

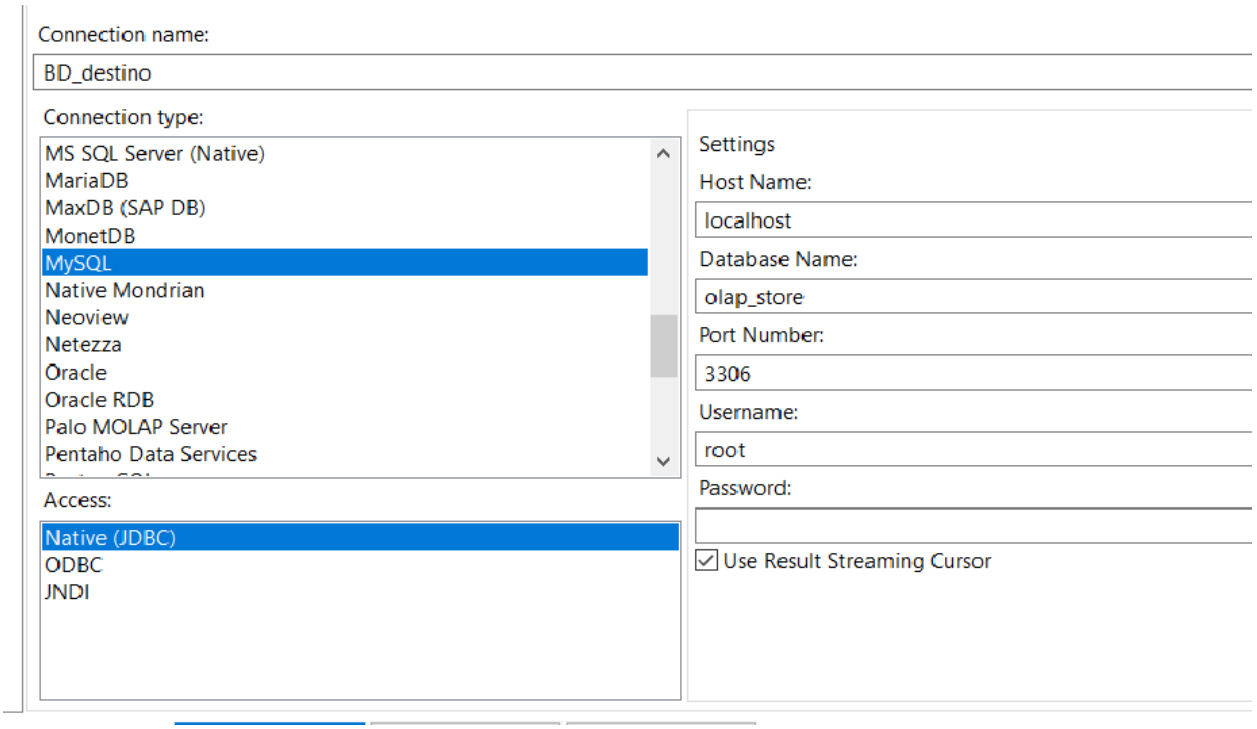
En las transformaciones haremos conexión del módulo input al módulo output

En cada transformación se debe colocar siempre la base de datos de donde se va a extraer la información y la base de datos de destino en nuestro caso es OLAP_store

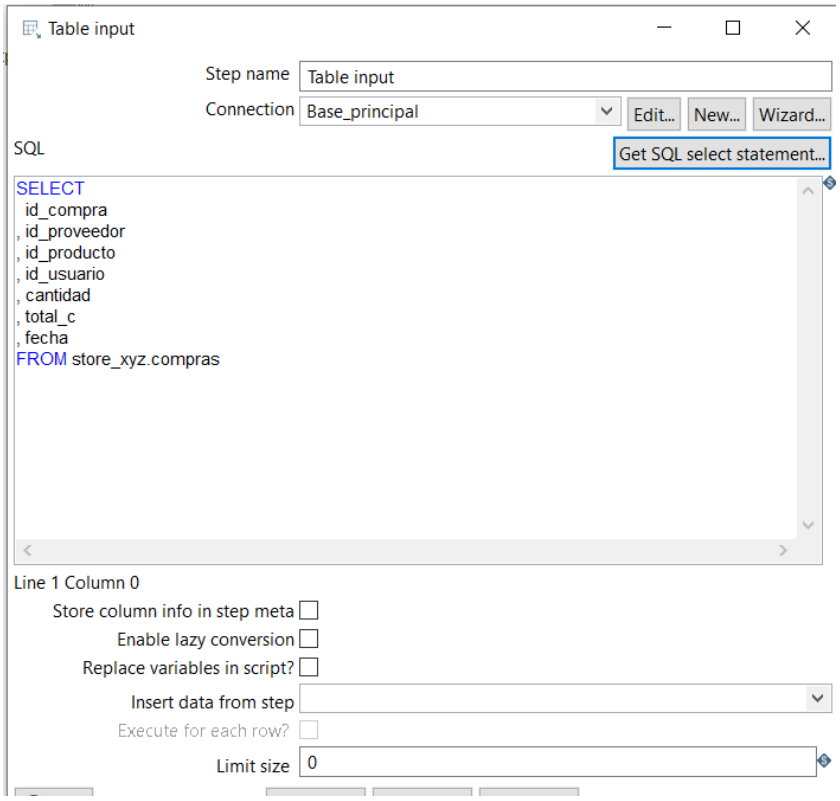
Base de datos principal.



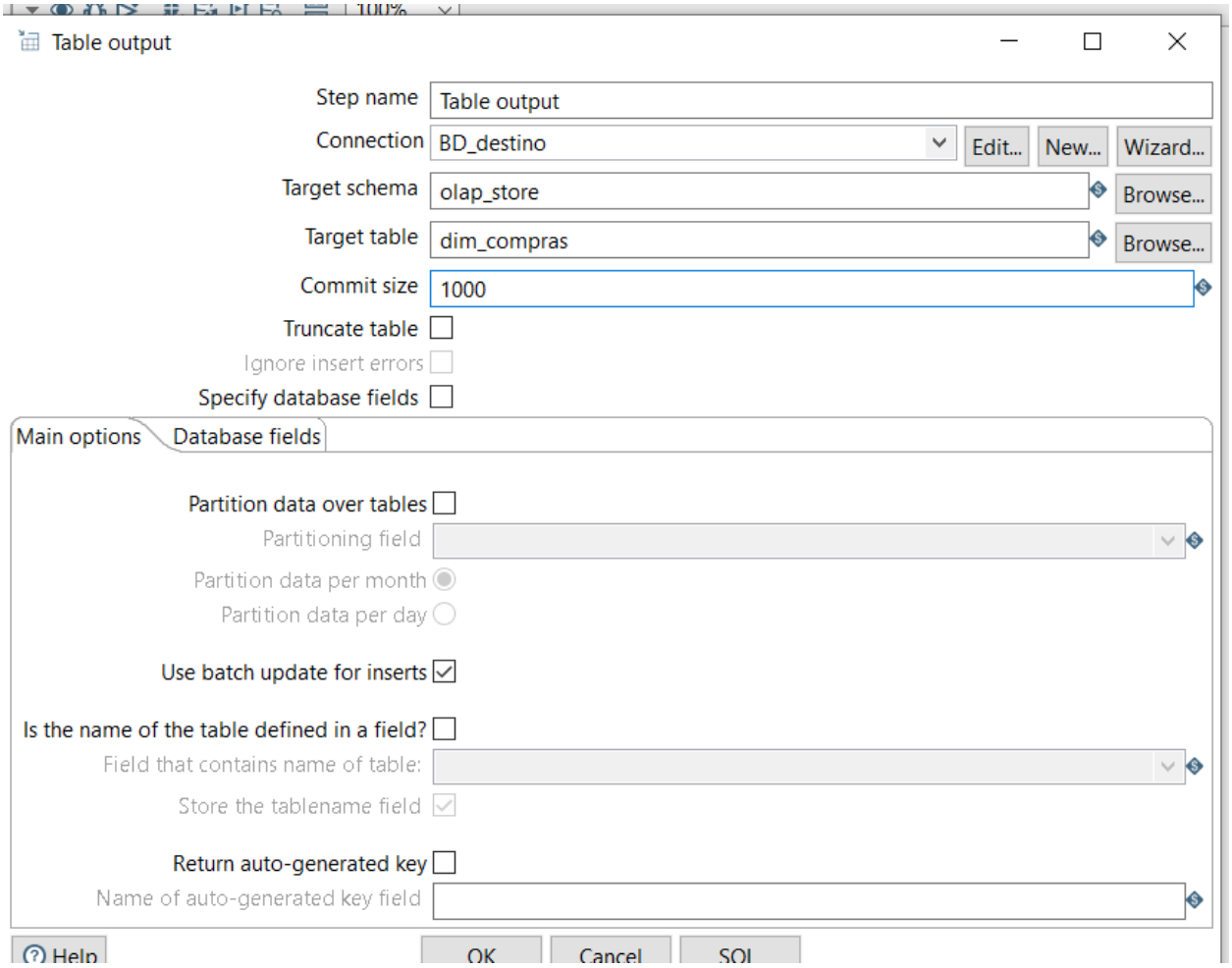
Base de datos destino



Completar la consulta SQL para extraer los datos de la tabla input.

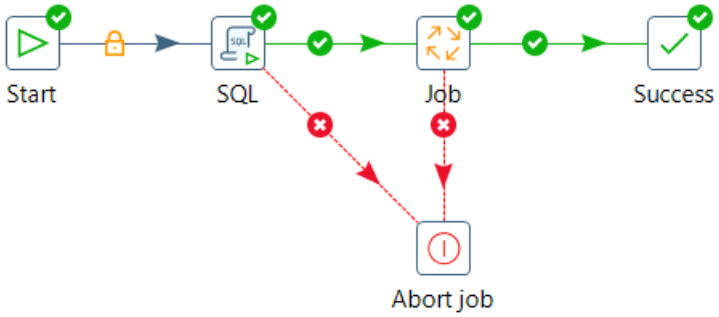


Configurar la el módulo Output para la inserción de datos a la base de destino.



Con ellos tendremos listo al transformación lista para crear las dimensiones

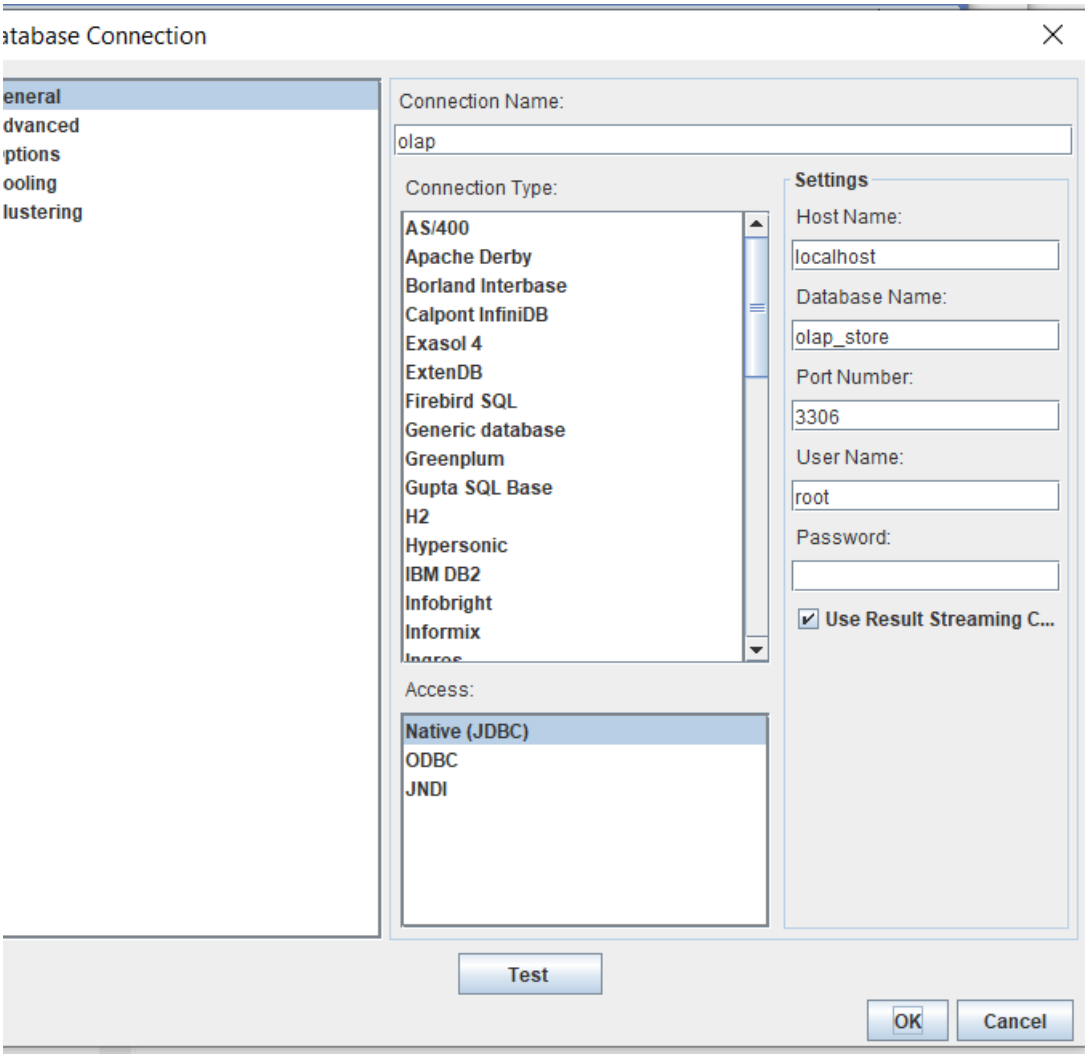
4. **Ejecutar el JOB el principal.**



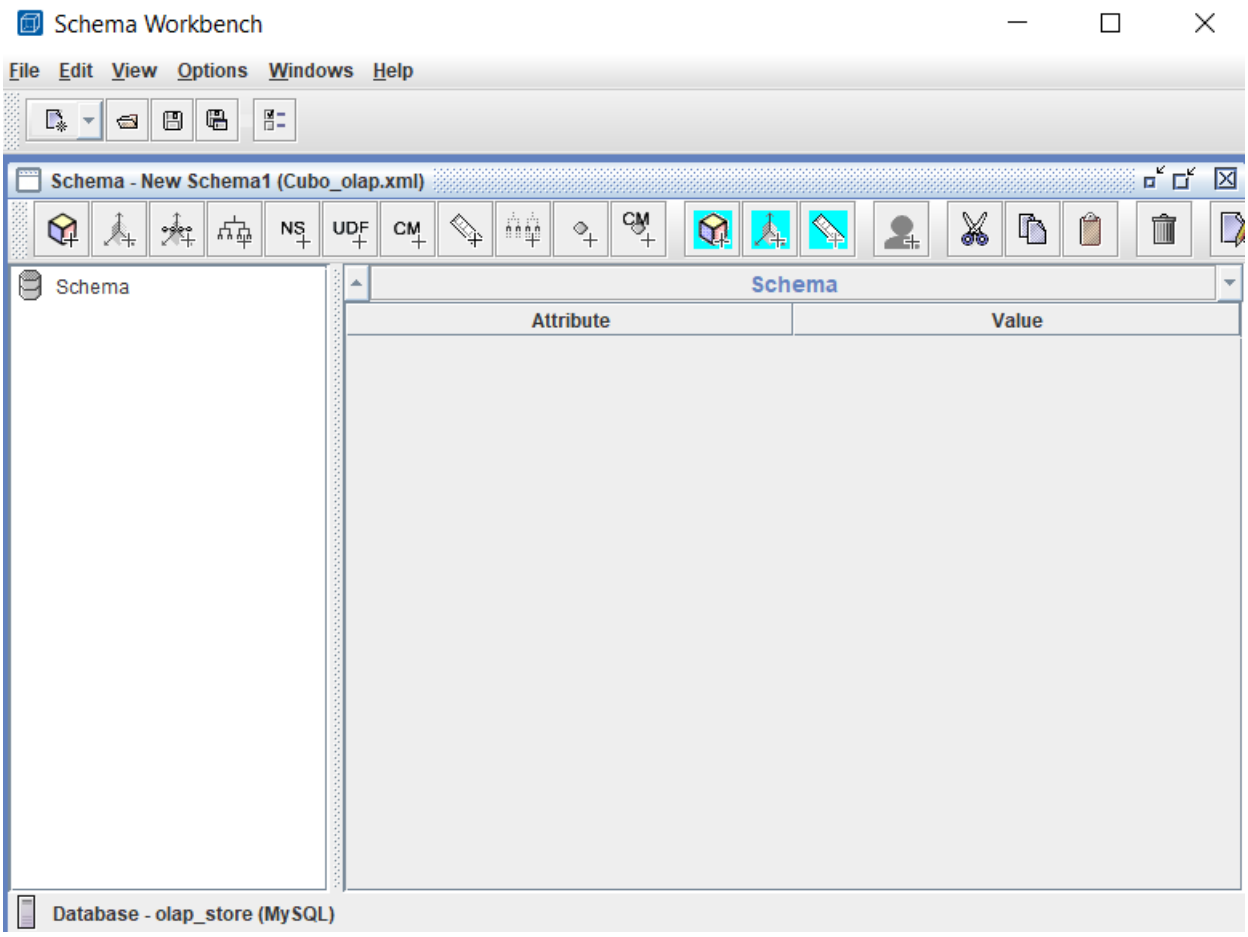
Si todo salió bien cada proceso marcara con check positivo de todos los procesos.

5. **Creación de Cubo en Schema Workbench.**

Antes de empezar tenemos que tener configurada la conexión a la base de datos donde extraeremos los datos.



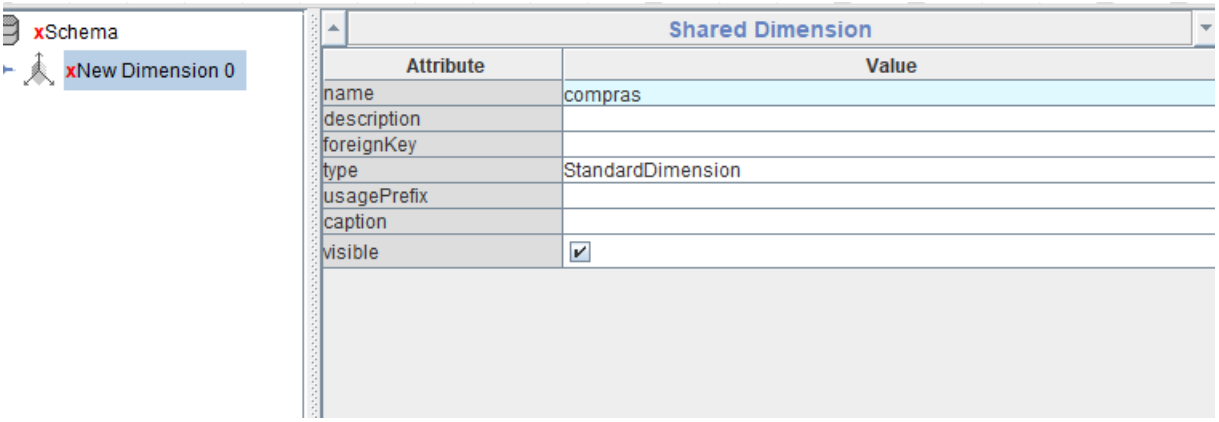
Luego crear un nuevo Schema.



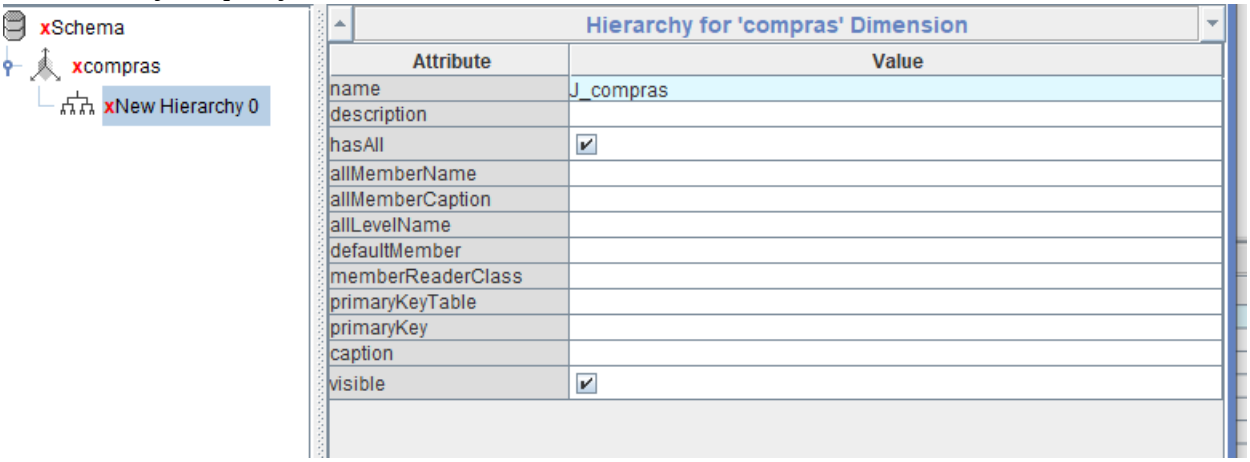
6. **Crear las dimensiones.**
   Vamos a crear las dimensiones del cubo a utilizar, en mi caso dim_compras y dim_proveedor.

   Pero vamos a configurar principal la dimensión compras.

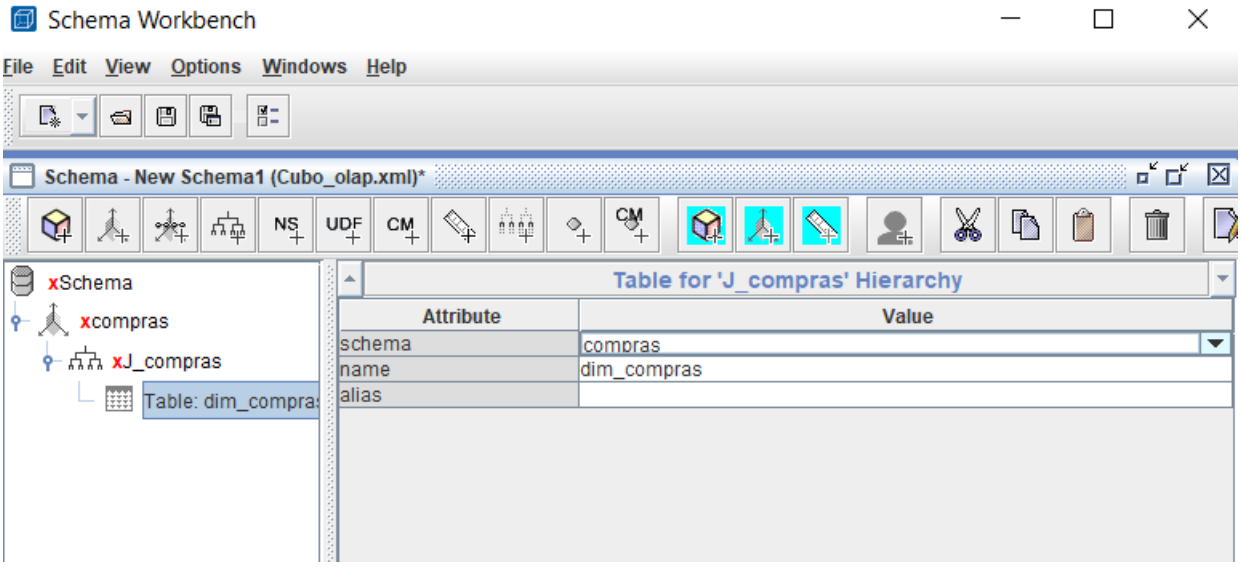
   Le damos nombre a la dimensión.



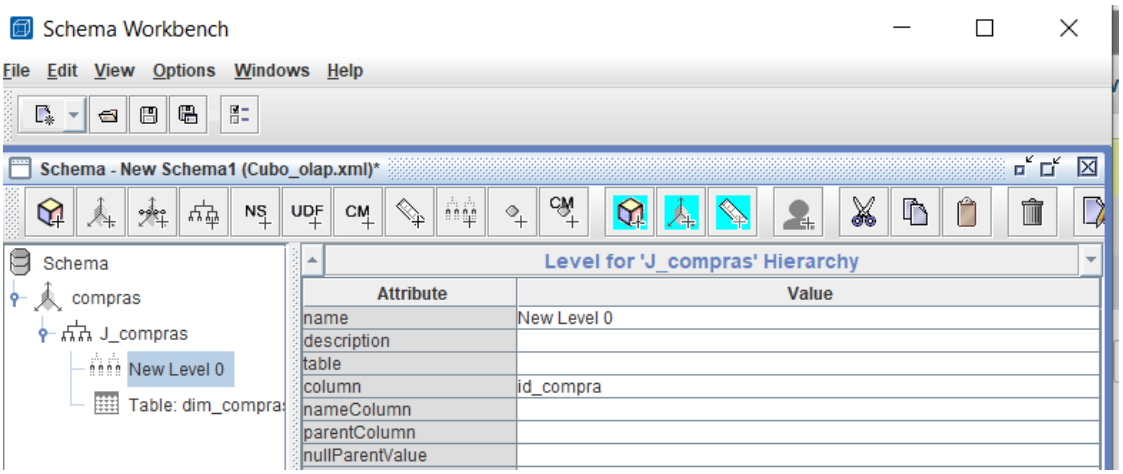
Creamos la jerarquía y le damos su nombre,



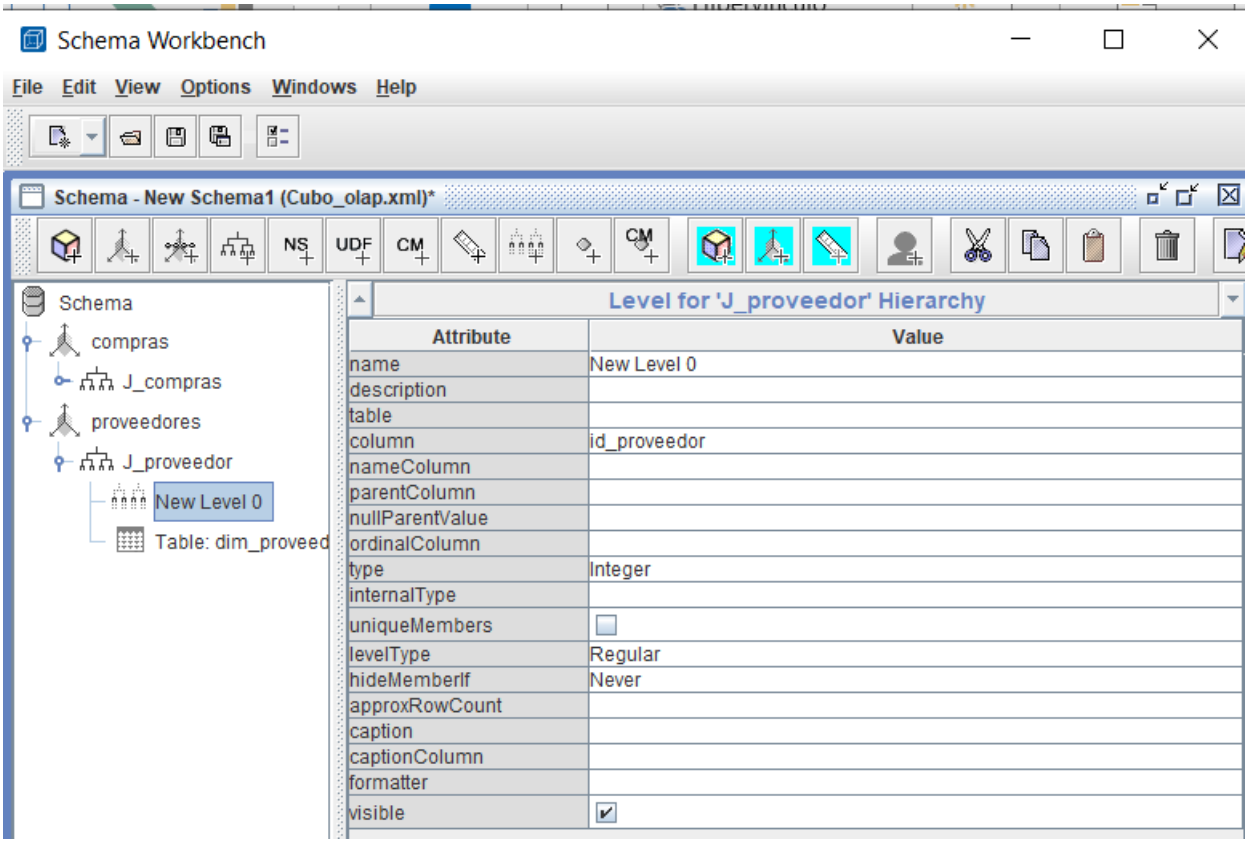
Creamos la tabla de la dimensión.



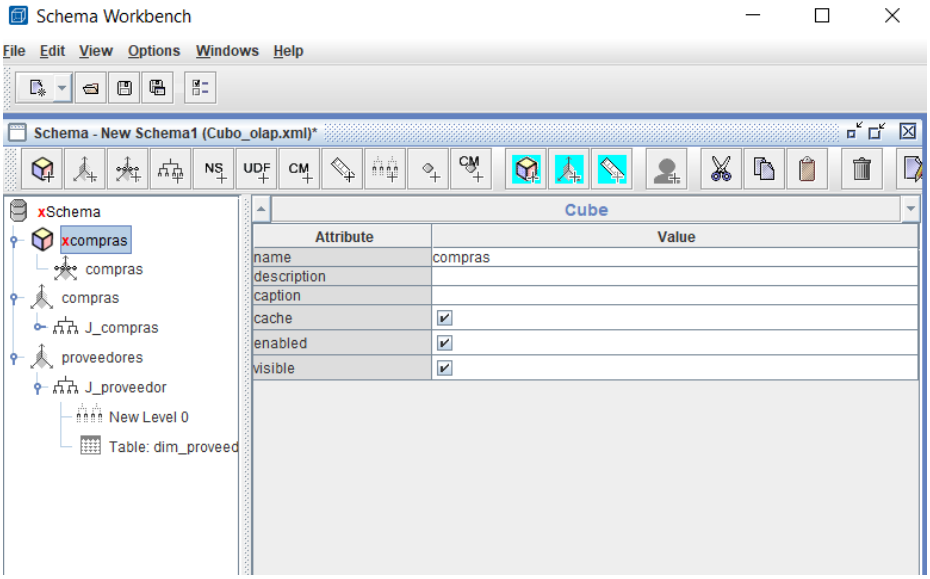
Creamos el nivel.



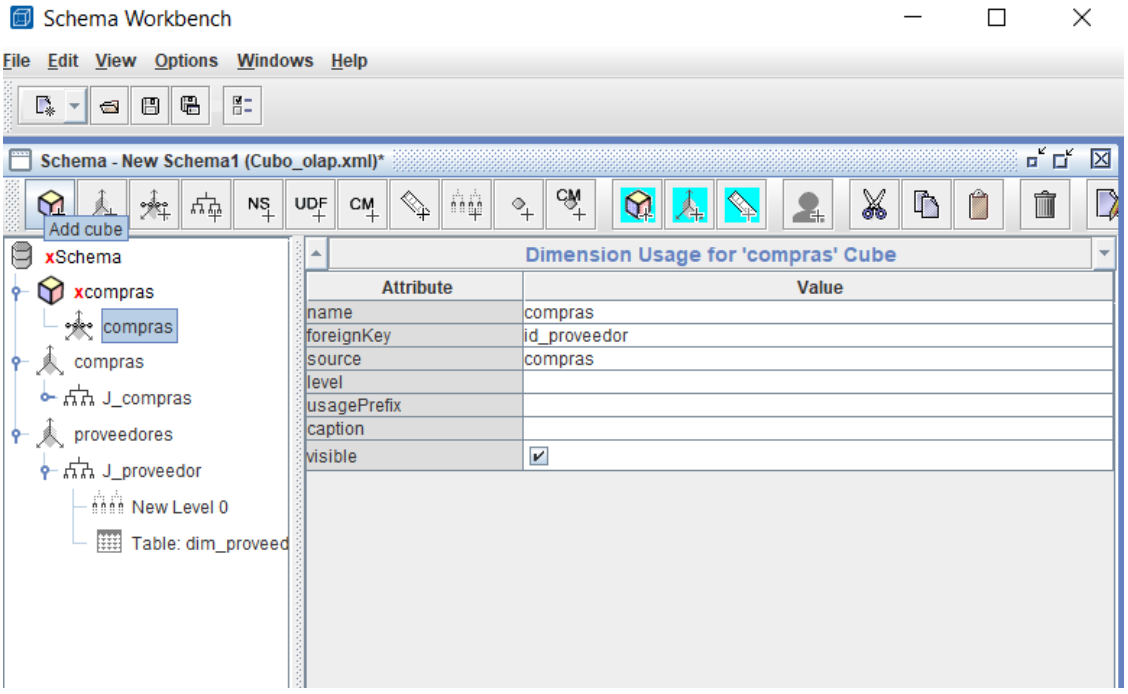
Con ello tendremos listo la dimensión compras para usarse.
Es de hacer las siguientes dimensiones para la creación del cubo.



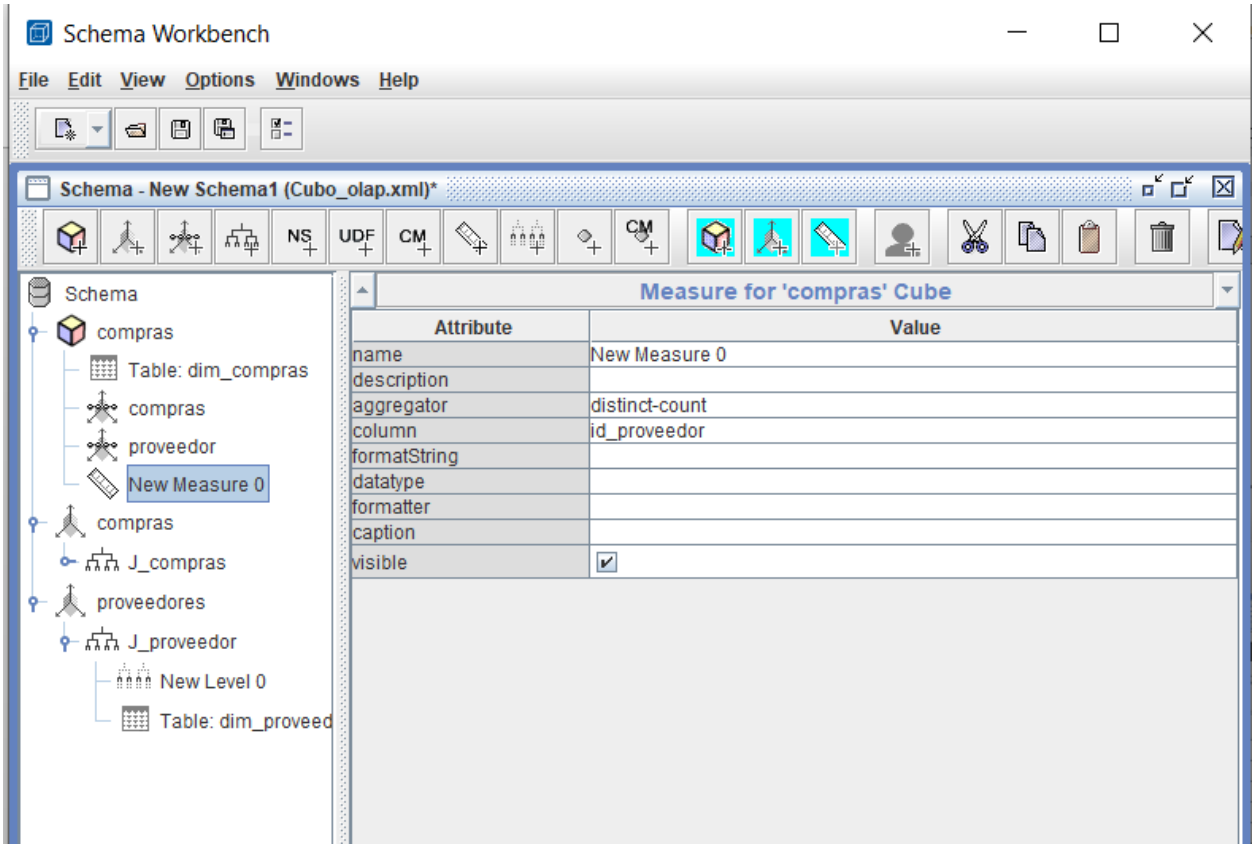
7. **Crear el cubo.**
Creamos el cubo y le damos un nombre.



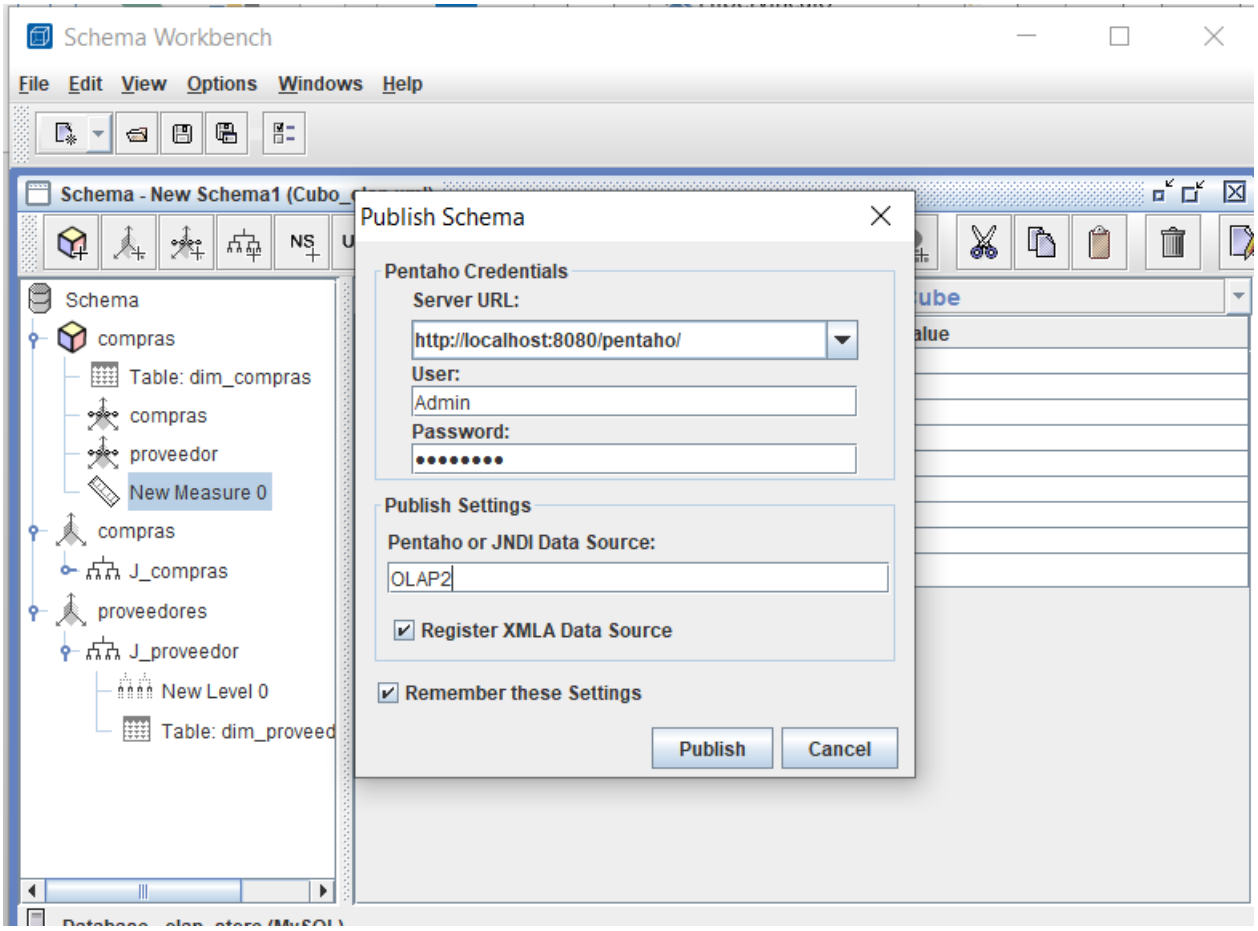
Ya que tenemos hechas las dimensiones de usarlas.

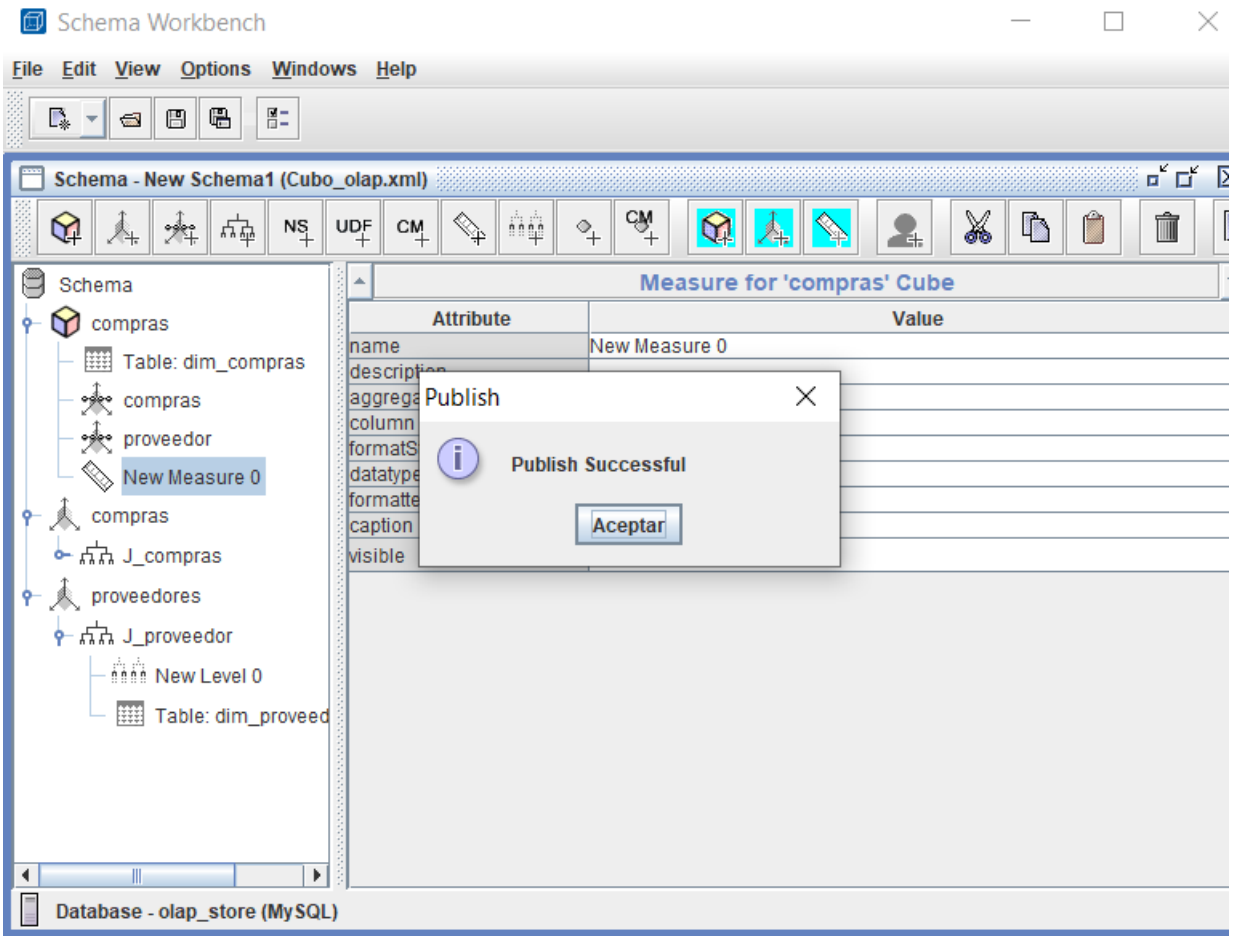


Creamos una medición.



8. **Publicar el cubo.**
   Ya una vez terminado el cubo, se procede a publicarlo en el Pentaho Server.




Schema Workbench
File Edit View Options Windows Help
Schema - New Schema1 (Cubo_olap.xml)
Schema
compras
Table: dim_compras
compras
proveedor
New Measure 0
compras
J_compras
proveedores
J_proveedor
New Level 0
Table: dim_proveed
Measure for 'compras' Cube
Attribute
Value
name
New Measure 0
description
aggrega
column
formatS
datatype
formatte
caption
visible
Publish
Publish Successful
Aceptar
Database - olap_store (MySQL)

Usamos este trigger para insertar datos a la tabla ventas

Nombre de disparador: venta_insert

Tiempo: BEFORE

Evento: INSERT

INSERT INTO acciones (usuario, fecha, accion) VALUES

(CURRENT_USER, CURRENT_DATE, 'Se inserto una venta ')

Usamos este trigger para actualizar los datos ingresados en la tabla ventas

Nombre de disparador: UPDATE_venta

Tiempo: BEFORE

Evento: UPDATE

INSERT INTO acciones (usuario, fecha, accion) VALUES

(CURRENT_USER, CURRENT_DATE, 'Se actulizo una venta ')

Usamos este trigger para eliminar datos ingresados en la tabla venta

Nombre de disparador: delete_venta

Tiempo: BEFORE

Evento: DELETE

INSERT INTO acciones (usuario, fecha, accion) VALUES

(CURRENT_USER, CURRENT_DATE, 'Se elimino una venta')

Los procedimientos se realizan para dejar una historia de algunos datos de la base de datos

Procedimiento 1.

Nombre de rutina: ESTADO

Tipo: PROCEDURE

Parámetros: Dirección IN, nombre dato, Tipo INT

```
BEGIN
 SELECT *
 FROM productos
 WHERE estado = dato;
END
```

Procedimiento 2.

Nombre de rutina: Hoy

Tipo: PROCEDURE

Parámetros: Dirección IN, nombre hoy, Tipo DATE

```
BEGIN
 SELECT *
 FROM productos
 WHE fecha= hoy;
 END
```