UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

ANÁLISIS COMPARATIVO DE HERRAMIENTAS PARA ALMACENES DE DATOS. UNA EVALUACIÓN A TRAVÉS DE LA EXPERIENCIA DE USUARIO

TESIS

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

CRISTIAN HERNÁNDEZ GARCÍA

DIRECTOR DE TESIS:

M. en I. GERARDO AVILÉS ROSAS

Ciudad Universitaria, Cd. Mx., 2018

Agradecimientos

A Dios, por permitirme existir. Por haberme dado la oportunidad de ocupar un lugar en este magnífico universo. Por darme la oportunidad de aprender cada día. Por haber puesto en mi camino a las personas que me han guiado y apoyado. Porque en cada uno de mis pasos siempre esta ahí.

A mis padres, por haberme traído al mundo. Porque gracias a sus enseñanzas me he convertido en lo que soy.

A mi familia, por trasmitirme su experiencia y sus consejos. Por acompañarme en este camino de la vida.

A Marco Aurelio, por haber sido un excelente amigo y compañero. Por haber sido un gran ejemplo a seguir durante la carrera.

A mis amigos, por haberme brindado su amistad y su cariño. Su apoyo incondicional formó parte de mi personalidad.

A mis profesores, por haberme dado más que el conocimiento de sus clases. Por haber sido parte de mi camino y mi formación académica.

A mi tutor, Gerardo Avilés, por enseñarme a mí y a todos sus alumnos como ser un profesor ejemplar, un gran modelo a seguir.

A la Facultad de Ciencias, por haber sido mi segundo hogar. Por mostrarme lo hermoso que es la ciencia y el conocimiento.

A la Universidad Nacional Autónoma de México, por permitirme el honor de ser parte de una de las mejores universidades del mundo.

Dedicatoria

Dedico la presente tesis, primeramente, a Dios, por haberme permitido vivir y concluir una etapa maravillosa en mi vida.

A mis padres, y a toda mi familia. Su cariño, motivación y esfuerzo me han impulsado día con día a lograr mis metas.

A todos mis profesores. Este es el resultado de su labor, gracias a su arduo trabajo me han proporcionado su conocimiento y sus diferentes formas de aprender.

No se vive una vez. El cuerpo muere, pero el conocimiento trasciende.
Aquello que hacemos en vida, no se lo puede llevar la muerte.
Infinito es aquel conocimiento que llena de vida el universo.
Todos nuestras memorias se las lleva el viento.
Sin embargo alguien más seguirá en el camino.
Imitando nuestro ejemplo, nuestras enseñanzas y nuestras acciones.
Rescribiendo nuestros éxitos, o nuestros fracasos.
Conmemorando cada acto que en vida hayamos generado.

Índice

Índice de figuras

Página

Índice de tablas

Convenciones

* Las palabras en inglés incluidas en el contenido de este trabajo (sin contar los anexos) tendrán formato de letra cursiva.

* Algunos de los términos abreviados serán usados a partir del momento en que son definidos. (Como BI, de Business Intelligence)

* Las letras negritas serán utilizadas para títulos, definiciones y en el caso de los anexos, para describir las herramientas (y su plataforma donde se usaron) y los elementos más representativos para demostrar a los usuarios.

* En el caso de los anexos, las palabras en inglés ya no tendrán formato cursivo, debido al mayor uso de contenido (en inglés).

INTRODUCCION

El presente trabajo tiene como objetivo mostrar el resultado de una comparación sobre algunas de las principales herramientas para Inteligencia de Negocios, en particular las que se refieren a los Almacenes de Datos, y proporcionar una opinión respecto a mi experiencia personal a través de los puntos más importantes del ciclo de vida de un almacén de datos.

La utilización de herramientas se ha vuelto importante debido al creciente aumento en la generación de datos y por otro lado, las empresas pretenden hacer uso de los datos para crear modelos que les generen conocimiento. Los datos tratados deben representar una ganancia en conocimiento que permita a las empresas mejorar la efectividad de la organización.

Tras las dificultades de los sistemas tradicionales para satisfacer las necesidades de procesamiento y almacenamiento de la información, surge el concepto de Almacenes de Datos (*Data Warehouse, término acuñado por Bill Inmon)*, como solución a las necesidades informáticas globales de la empresa. La ventaja principal de este tipo de soluciones se basa en su concepto fundamental, la estructura en que se almacenan los datos.

Este concepto significa el almacenamiento de datos homogéneos, en una estructura basada en la consulta y el tratamiento jerarquizado de la misma, en un entorno diferenciado de los sistemas operacionales. Las empresas de todos los tamaños y en diferentes industrias, así como gobierno, están descubriendo que pueden obtener beneficios significativos mediante la aplicación de un almacén de datos. Es generalmente aceptado que esta tecnología proporciona un excelente enfoque para la transformación de las grandes cantidades de datos que existen en información útil y confiable, para obtener respuestas a sus preguntas y para apoyar el proceso de toma de decisiones.

Un almacén de datos proporciona la base para la aplicación exitosa de técnicas de análisis de datos como la minería de datos y el análisis multidimensional, así como consultas tradicionales y presentación de informes; esto permite que los almacenes de datos pueden dar lugar a un acceso más fácil a la información que se necesita, para obtener decisiones informadas y a tiempo.

Este trabajo está conformado por cuatro capítulos. En el primero se presenta el marco teórico de los almacenes de datos. En el segundo capítulo se ilustran los procesos por los que un almacén debe de pasar para su creación. En el tercer capítulo se presenta una demostración de algunas de las principales herramientas utilizadas para el almacenamiento de datos. En el capítulo cuatro se realiza un resumen sobre la experiencia personal vista por las herramientas implementadas, así como la conclusión sobre la comparación entre estas.

CAPÍTULO 1. ALMACENES DE DATOS

1. TEORÍA DE ALMACENES DE DATOS

1.1 Almacén de datos. Definición

Un almacén de datos es una colección de datos orientado hacia un tema relevante para una organización, el cual debe cumplir con la característica de no ser volátil, tener sus datos integrados y ser variable en el tiempo. Su principal función es permitir la integración de datos provenientes de fuentes funcionalmente distintas (fuentes heterogéneas), para proporcionar una vista integrada, esto permite construir sistemas que soportan la toma de decisiones en una organización. [1]

Un almacén de datos mantiene de forma integrada todas los datos resultantes de las operaciones diarias de la organización: movimientos que modifican el estado de la organización, interacciones con clientes y/o proveedores, etc., de manera que se pueda obtener información que no suele estar a simple vista, a través de la aplicación de técnicas de análisis de datos. Derivado de la naturaleza heterogénea de los datos, es de suma importancia que se haga un aseguramiento de calidad de los datos, el cual puede lograrse en primera instancia, a partir de técnicas de limpieza e integración de datos, de manera que se mantengan estructuras homogéneas y consistentes, durante un largo período de tiempo. [13]

1.2 Características de un almacén de datos

Las características de un almacén de datos, se describen a continuación:

• **Orientado a un tema relevante para organización.** En este sentido, se busca construir un entorno que permita hacer consultas eficientes para información relacionada con actividades básicas de la organización: compras, ventas, producción, por mencionar algunos. No es de interés en este punto que se encargue de manejar la información transaccional (para eso existen las fuentes de datos).

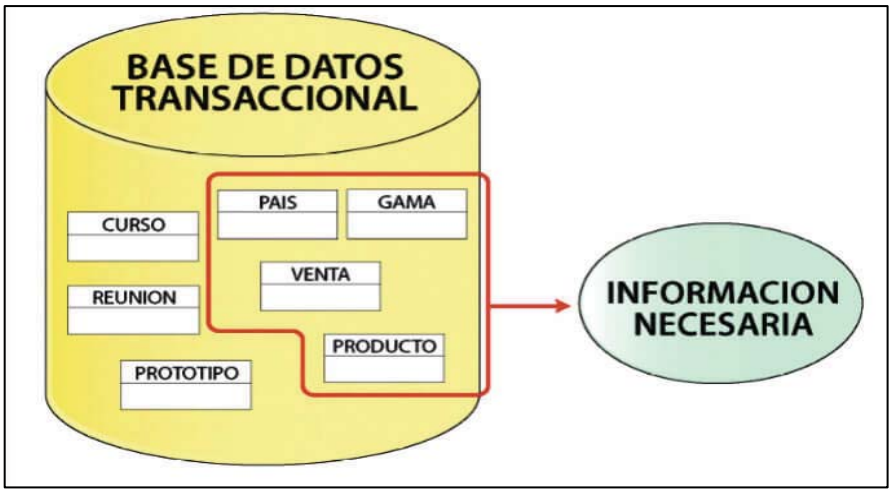


Figura 1. Almacén orientado hacia los datos relevantes [10]

• **Datos integrados**. Reúne en un solo repositorio consolidado los datos recolectados de diferentes sistemas operacionales de la organización y/o fuentes externas. Se suelen aplicar distintas técnicas (limpieza e integración) que permitan asegurar la consistencia de los datos.

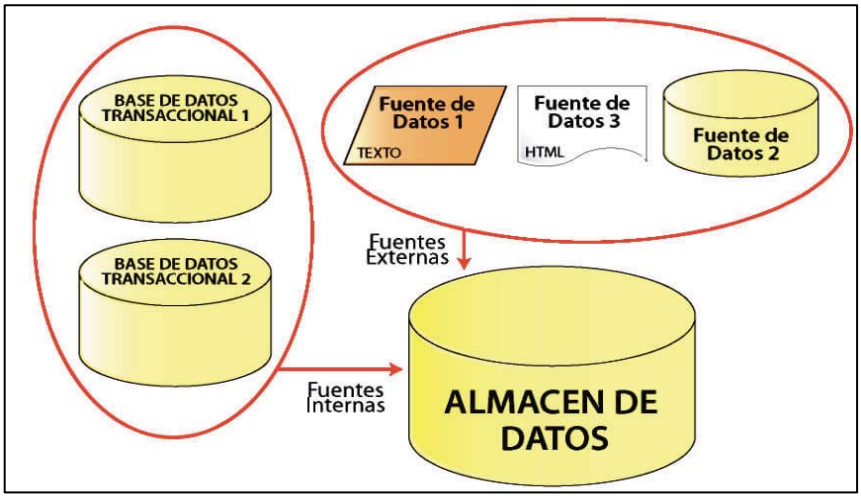


Figura 2. Datos integrados [10]

Como se muestra en la figura 2, pueden integrarse los datos desde fuentes como bases de datos, archivos de texto, html, entre otras fuentes de datos. Esto permite a los usuarios concentrar datos para obtener más información.

• **Variante en el tiempo.** Permite que los datos se estudien desde una perspectiva histórica, ya que los datos son relativos a un periodo de tiempo. Por esta razón es que los nuevos datos deben ser integrados periódicamente.

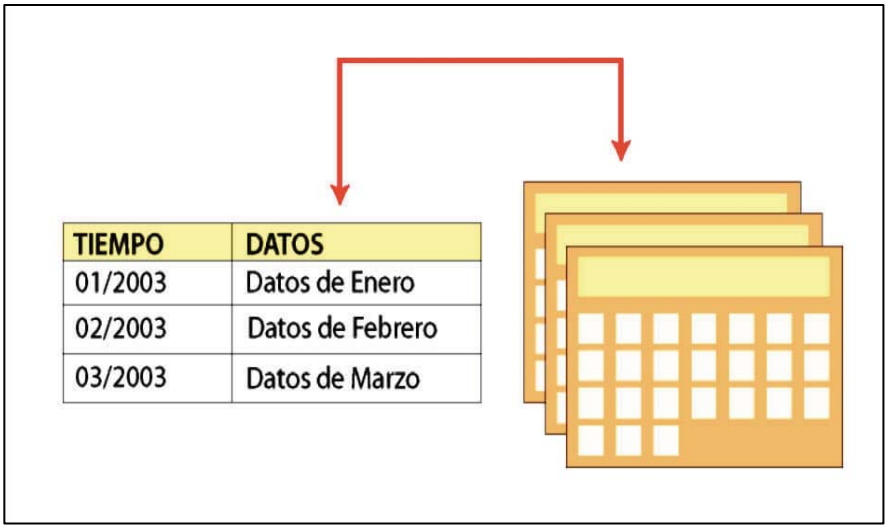


Figura 3. Almacén variable en el tiempo [10]

• **No volátil.** Bajo esta estructura no se requieren mecanismos que permitan controlar transacciones, ni mecanismos para el control de concurrencia, ya que se espera que los datos almacenados no sufran de ninguna actualización, más bien agregar nuevos datos (incrementales) y operaciones para consulta. [10]

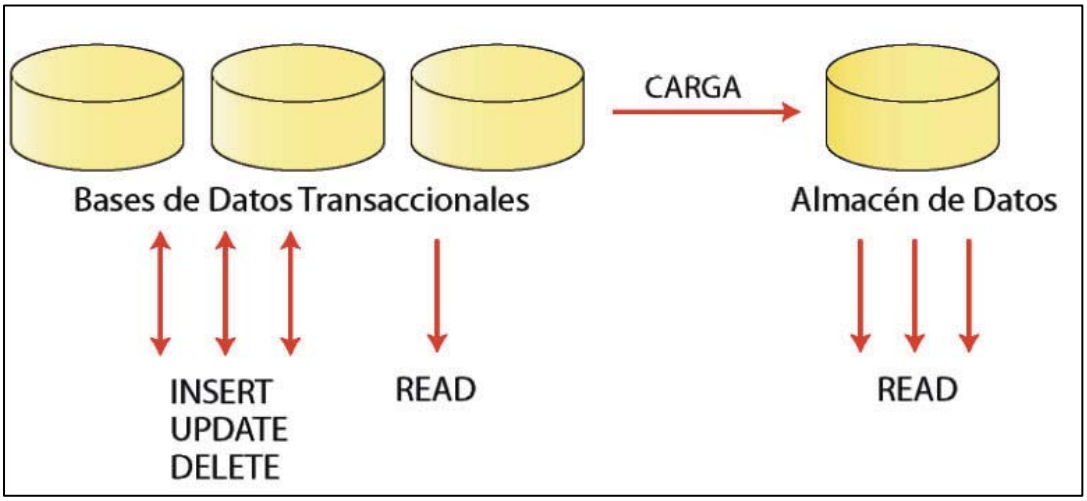


Figura 4. Datos no volátiles [10]

1.3 Arquitectura de un almacén de datos

Los componentes que permiten la existencia de un almacén de datos son los siguientes:

• **Fuentes de datos heterogéneas**. Se trata de un componente que normalmente está presente en las organizaciones, se trata de sistemas que mantienen el día a día de la empresa, se suele referir a ellos como sistemas de procesamiento de transacciones en línea (OLTP, por sus siglas en inglés) aunque no necesariamente todos los datos se encuentran dentro de estos sistemas, ya que con frecuencia muchos sistemas críticos de la organización se basan en hojas de cálculo e incluso, archivos de texto plano. A partir de estos sistemas se realiza la captura de datos que formarán parte del almacén. Estas fuentes de datos pueden ser: sistemas operacionales corporativos, sistemas operacionales departamentales, fuentes externas, etc.

• **Extracción y transformación.** Este componente es el encargado de que la información pueda tomarse de las fuentes heterogéneas y llevarla al nuevo repositorio consolidado. Este paso no es trivial, ya que se tiene un esquema de almacenamiento diferente, por lo que es necesario que los datos puedan ser transformados para ajustarlos al nuevo entorno.

• **Repositorio de Metadatos:** Los metadatos son básicamente datos acerca de los datos que se encuentra presentes en el almacén. A través de este componente los usuarios pueden saber qué datos se encuentran en el almacén y de esta forma buscar una manera para acceder a sus requerimientos de información. Entre sus funcionalidades más representativas se encuentran: catálogos y descripción de la información disponible; especificación del propósito de la misma; especificación de las relaciones entre los distintos datos; relación con los datos operacionales, entre otras.

• **Área de preparación de datos:** Es un espacio orientado a almacenar los datos provenientes de las fuentes de datos heterogéneas de forma temporal. Se utiliza como punto de partida de los procesos de depuración, limpieza, integración, transformación y carga. Se trata de una área intermedia en la arquitectura del almacén de datos, esto permite que sea más fácil extraer los datos desde las fuentes de origen, además de realizar un pre-procesamiento, permite acceder en detalle a información no contenida (aún) en el almacén. Derivado de las tareas que debe realizar, es altamente recomendable que se realicen en un servidor (generalmente relacional) que esté separado (física y lógicamente) de las fuentes de datos y del almacén.

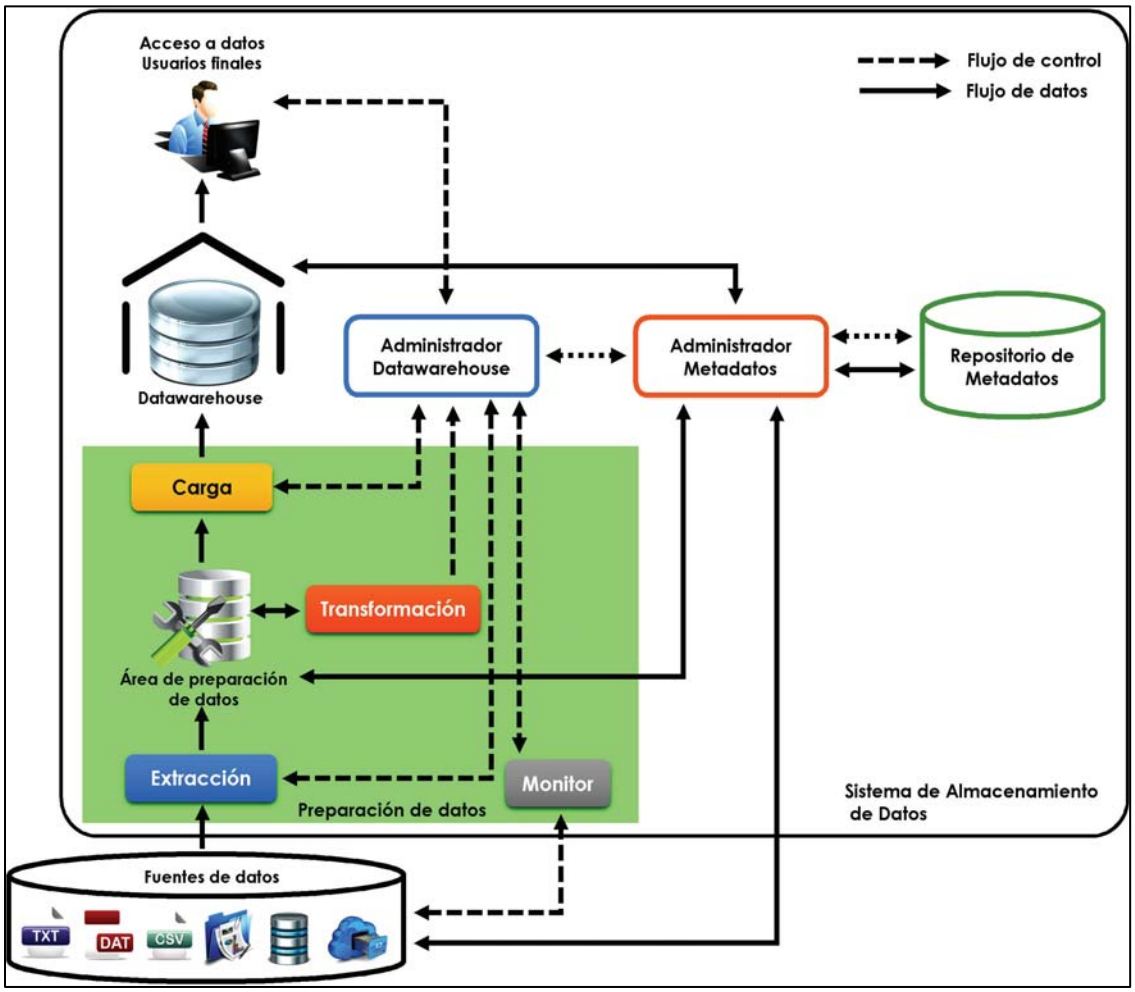


Figura 5. Componentes de un almacén de datos

Se trata de un área muy importante dentro de la arquitectura del almacén de datos, por esta razón solo debe estar administrada por el equipo encargado de la construcción del almacén de datos, ningún usuario debe tener acceso a este espacio, ya que es un sitio de construcción, no de consulta.

Fuera de la arquitectura se encuentran algunos conceptos importantes a mencionar:

• **Consolidación:** Es el proceso que se encarga de reunir las entidades y atributos en un almacén de datos, es importante mencionar que en este punto, las entidades mantienen sus valores, pero aún no tienen un significado en común hasta que se apliquen técnicas de integración.

• ***Middleware*:** Es un software que se encuentra entre un cliente y un servidor. Funciona como una capa intermedia que permite la conectividad entre los componentes del almacén de datos.

• **Mercado de datos (*Data mart*):** Es un subconjunto de los datos de un almacén de datos. Pueden ser utilizados para ciertas áreas específicas o departamentos. Los datos de cada mercado se encuentran integrados con los del almacén.

• **OLAP (*On-Line Analytical Process*):** Es una tecnología utilizada para administrar grandes bases de datos. Consiste de una base de datos multidimensional en la cual, los elementos se almacenan de forma vectorial (cubos). De aquí se origina el termino de cubos OLAP. La estructura de cubos permite a los usuarios mayor transparencia de los datos. Los datos pueden almacenarse de manera histórica, añadido al hecho de que OLAP esta creada para usar solo consultas. Calcular una consulta en una base de datos multidimensional es más rápida y eficiente [13].

1.4 Tecnología OLAP

El procesamiento analítico en línea (OLAP) es una tecnología que permite analizar datos de negocio para generar información estratégica que brinde soporte a los usuarios en la toma de decisiones. Edgar F. Codd propone una serie de directrices para medir las herramientas OLAP [18]:

• **Vista multidimensional conceptual.** Provee un modelo de datos multidimensional. Este modelo debe ajustarse a las necesidades de negocio en la mejor medida posible tal como la perciben los usuarios.

• **Transparencia.** Tanto la arquitectura del almacén (resaltando repositorios de datos) como los datos de origen deben ser totalmente comprensibles para los usuarios.

• **Arquitectura cliente/servidor.** Genera un sistema basado en la arquitectura cliente/servidor. De esta manera el sistema permite que varios clientes puedan adaptarse sin problemas, y el sistema pueda responder de forma óptima.

• **Accesibilidad.** Permite acceder a los datos requeridos para llevar a cabo un análisis. El sistema también se encarga de presentar una vista consistente para los usuarios, haciendo uso de las transformaciones que sean necesarias para obtener datos heterogéneos.

• **Soporte multiusuario.** Permite el trabajo simultáneo entre más de un usuario. También apoya en la creación de diferentes modelos usando los mismos datos.

• **Operaciones sin restricciones de dimensiones cruzadas.** Permite la capacidad para realizar operaciones de enrollar (*roll-up*) y profundizar (*drill-down*) dentro de cualquier dimensión.

• **Dimensiones ilimitadas y niveles de agregación.** El usuario puede libremente definir el número de dimensiones y niveles de agregación que sean requeridos.

• **Reportes consistentes.** El sistema se encarga de que los reportes no presenten alguna alteración en caso de aumentar el número de dimensiones.

• **Manipulación de datos intuitivos.** Los procesos de profundizar (*drill-down)* y enrollar (*roll-up*) son realizados de manera simple e intuitiva gracias a las interfaces de arrastrar y soltar (*drag and drop*).

• **Dimensionalidad genérica.** El sistema asegura que cada dimensión de datos sea equivalente en estructura y capacidad operativa.

• **Reportes flexibles.** Permite a los usuarios obtener de manera sencilla cualquier análisis de la información.

1.5 Formas de configuración OLAP

Entre las formas de configuración OLAP más conocidas se encuentran [11]:

• **OLAP Relacional (ROLAP):** Esta configuración requiere de sistemas manejadores de bases de datos relacionales, los cuales son colocados entre el servidor y las herramientas cliente (figura 6). Algunas ventajas que pueden considerarse son: el uso de datos y estructuras dinámicas; el sistema aprovecha la seguridad e integridad de la base de datos; es escalable y sus datos pueden ser compartidos con otras aplicaciones.

También presenta algunas desventajas significativas: sus consultas son más lentas; la construcción es más costosa y tiene algunas limitaciones en cuanto a las funciones de la base de datos.

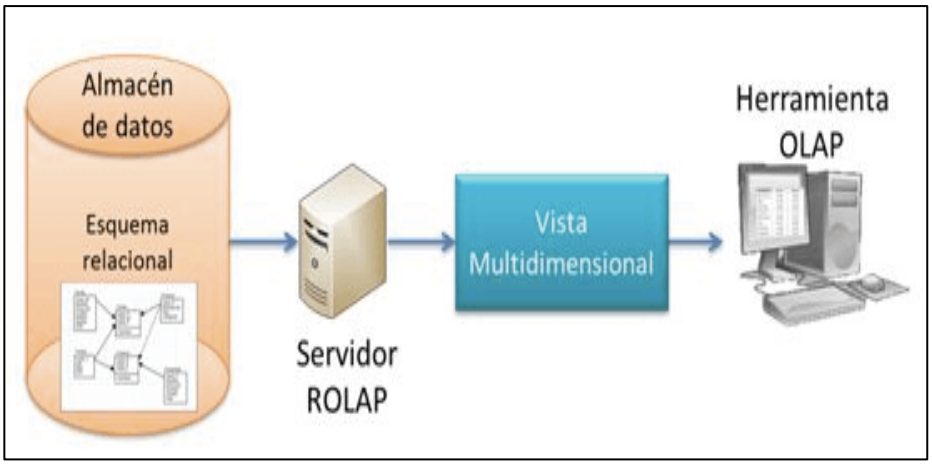


Figura 6. Configuración ROLAP [20]

• **Multidimensional OLAP (MOLAP):** Esta configuración utiliza motores de almacenamiento multidimensional. Presenta algunas ventajas importantes, tales como: escribir sobre la base de datos; ejecutar un mejor desempeño en el procesamiento de consultas; generar funciones más complejas y realizar los cálculos de estas funciones en menor tiempo. Una desventaja importante es que presenta un tamaño limitado para la arquitectura del cubo.

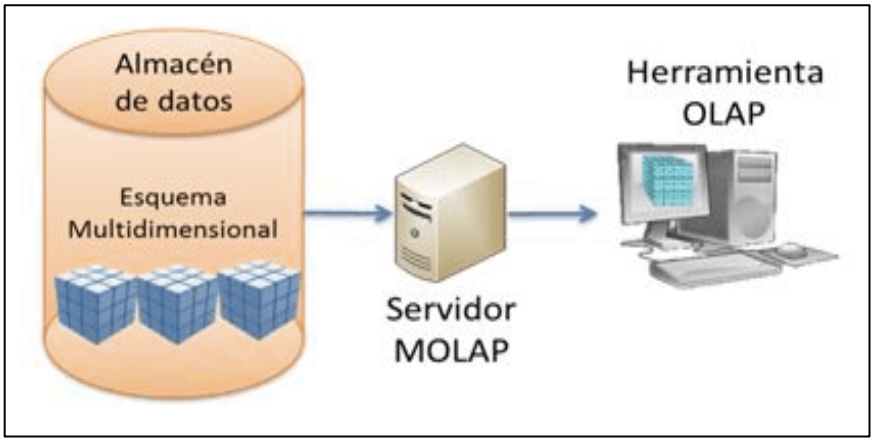


Figura 7. Configuración MOLAP [20]

• **OLAP híbrido (HOLAP):** OLAP híbrido es una combinación entre ROLAP y MOLAP. Esta configuración permite mayor escalabilidad de ROLAP y un cálculo más rapido que MOLAP. También puede almacenar grandes volúmenes de datos.

• **Servidores SQL especializados:** Estos servidores aportan un soporte particular para el procesamiento de consultas SQL y lenguaje de consulta avanzada MDX (más adelante se define el lenguaje MDX).

1.6 Procesos ETL

El sistema ETL (extracción, transformación y carga, de sus siglas en inglés) es el proceso intermedio entre los datos de origen y el área de presentación. Primero se extraen los datos, lo cual implica leer los datos de origen y posteriormente copiar los datos requeridos hacia el área de preparación. A continuación, los datos pueden requerir algún proceso de transformación para crear uniformidad.

Existen diferentes transformaciones que pueden aplicarse al contenido de los datos, principalmente cuando los datos provienen de múltiples fuentes. Algunas de las más comunes pueden ser: retirar los errores ortográficos, tratar con elementos que faltan o elementos que sobran, estandarizar algún tipo de formato, quitar elementos duplicados, etc. Una vez que se han aplicado tareas de limpieza a los datos, el resultado obtenido será un nuevo conjunto de datos que ha sido optimizado. En esta parte, el sistema ETL le ha añadido valor a los datos de origen porque ha establecido homogeneidad a los datos transformados.

Para finalizar el sistema lleva a cabo el proceso de carga, en el que transfiere los datos hacia el modelo dimensional destino (zona de presentación). [5]

1.7 Ciclo de vida de un almacén de datos

La metodología propuesta por Ralph Kimball (considerado uno de los autores con mayor aporte en el área de almacenes de datos), está compuesta por [4]:

1.- Planificación del proyecto: Establecer los objetivos y los alcances del proyecto. Definir el plan de desarrollo del proyecto así como el seguimiento.

2.- Definición de los requerimientos de negocio: Localizar los factores clave que determinarán el diseño apropiado.

3.- Diseño de la arquitectura técnica: Considerar los requerimientos de negocio, los entornos técnicos y la posibilidad de escalar el desarrollo del almacén.

4.- Modelado dimensional: Determinar las dimensiones y el grado de detalle de cada factor clave del negocio.

5.- Diseño físico: Localizar las estructuras que serán necesarias para dar soporte al diseño lógico.

6.- Diseño y desarrollo de la presentación de datos: Orientado a los procesos ETL. Se busca definir las actividades necesarias de manipulación de los datos.

7.- Selección de productos e instalación: Seleccionar los componentes que se utilizarán para construir la arquitectura (hardware, motor de base de datos, etc).

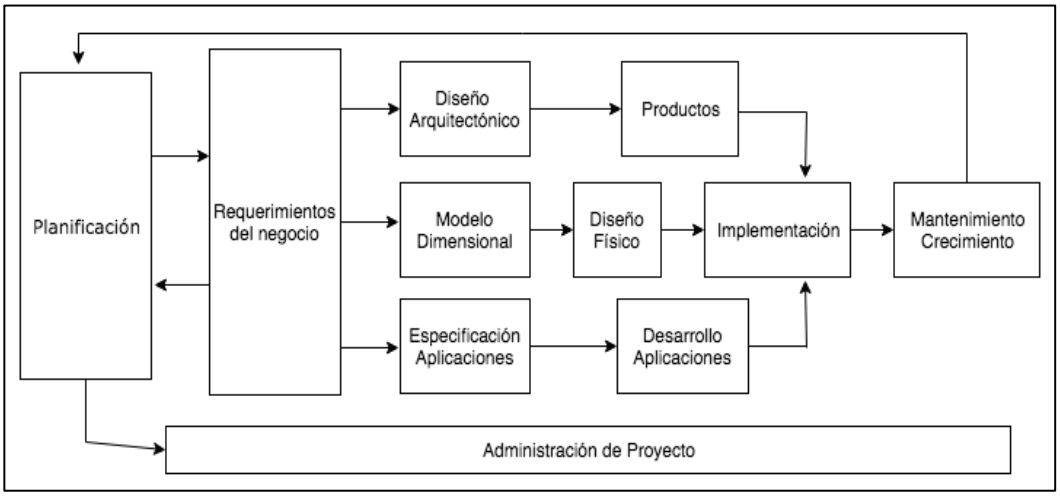


Figura 8. Ciclo de vida

1.8 Opciones de implementación de un Almacén de Datos

Para comprender la necesidad de un almacén de datos, es necesario mencionar la diferencia entre los sistemas orientados al procesamiento de transacciones (OLTP) y los sistemas OLAP. En los sistemas OLTP se llevan a cabo

transacciones diariamente, por lo que están hechos para cumplir con tareas de lectura y escritura. El diseño de su base de datos suele ser altamente normalizado (principalmente con tercer forma normal). Los sistemas OLAP tienen como objetivo almacenar grandes cantidades de datos a lo largo de mucho tiempo, por lo que su principal función es la consulta. Su diseño de base de datos suele no estar normalizado. La finalidad de estos sistemas es brindar soporte analítico para la toma de decisiones.

“La elección de un enfoque de implementación se ve influenciada por factores tales como la actual infraestructura, los recursos disponibles, la arquitectura seleccionada, el alcance de la aplicación, la necesidad de acceso a los datos más globales en toda la organización, los requisitos de retorno de la inversión y la velocidad de implementación.” [1, p. 18]

• **Implementación *top down.*** Se basa en la planificación previa a la implementación. Cada una de las áreas involucradas planeará cuales son los factores que deben desarrollar para integrar la implementación. Se definen las variables y las reglas de negocio de toda la organización. Como resultado se espera que el costo de planificación pueda impactar en los beneficios y el retorno de inversión, ya que el desarrollo de un modelo de datos global es una tarea compleja.

• **Implementación *bottom up*.** Se basa en el desarrollo de la planificación y la implementación incremental. Implica comenzar con pequeños diseños de mercados de datos sin contar con una infraestructura global. Podría considerarse como una inversión más segura cuando se trata de un negocio en crecimiento. Se ejecuta una evaluación de las necesidades y se van escalando las infraestructuras. Suele reducir el costo a corto plazo.

• **Un enfoque combinado.** De acuerdo a las necesidades de un negocio, es posible ajustar un enfoque que equilibre ambos tipos de implementación. Puede

lograrse con un alto grado de planificación que se lleve a cabo en una estructura de almacenamiento accesible por todos los mercados. Se debe de considerar como desventaja, el espacio de almacenamiento y la posibilidad de redundancia de datos. Otro detalle a considerar es mantener los datos de los múltiples mercados de datos al mismo nivel de consistencia. [1]

1.9 Modelos de datos

En los sistemas OLTP es tradicional utilizar bases de datos normalizadas con tercera forma normal. Los modelos en tercera forma normal son muy útiles cuando se desea eliminar la redundancia de datos. Sin embargo, al momento de construir consultas se vuelve un problema muy complejo. Debido a esto, no es factible utilizar tercera forma normal en un almacén de datos, en su lugar se usa el modelo dimensional. Este modelo incluye los mismos datos que un modelo en tercera forma normal, la diferencia es que presenta redundancia de datos debido a su organización. A cambio se busca un mejor desempeño en las consultas.

Un modelo dimensional se compone de tablas de hecho y tablas de dimensión. Las tablas de hecho contienen los datos centrales del almacén (los datos que explotarán la información principal del negocio, por ejemplo, ventas, costos, ganancias, etc.), mientras que las tablas de dimensión contienen una serie de atributos o características relacionadas con los hechos. También incluyen otros elementos que se explicarán con detalle en el siguiente capítulo. Existen dos representaciones básicas utilizadas en el modelo dimensional:

• **Modelo estrella.** El modelo estrella es la estructura básica de un modelo dimensional. Cuenta con una tabla de hechos en el centro y un conjunto de tablas de dimensiones a su alrededor. La tabla de hechos contiene los valores de las medidas principales que se desean conocer. Las tablas de dimensiones contienen atributos sobre las medidas de la tabla de hechos. En la figura 9, la tabla ventas representa la tabla de hechos.

En un negocio tradicional, un usuario de negocio comúnmente busca conocer información sobre las ventas de su empresa. Por otro lado, las tablas producto, sucursal, fecha y cliente, forman las tablas de dimensiones. Aquí se incluyen los datos que conforman una venta. El modelo toma el nombre de estrella debido a que solo hay una tabla que conecta con todas las dimensiones.

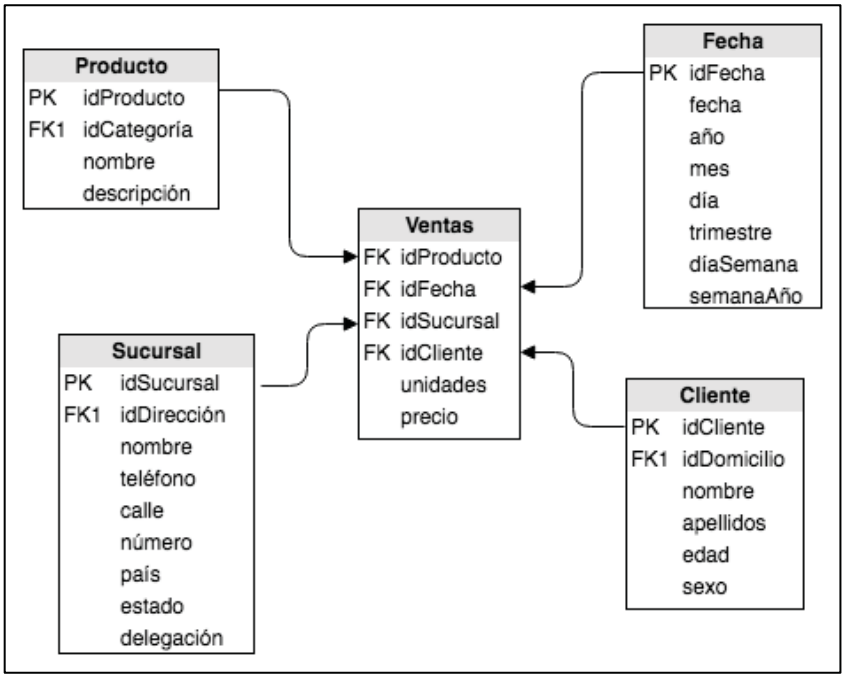


Figura 9. Modelo estrella

• **Modelo copo de nieve.** El modelo copo de nieve inicia con la forma de modelo estrella, pero a su vez, algunas de sus dimensiones se encuentran normalizadas creando otras dimensiones a su alrededor. La finalidad de tener más dimensiones consiste en evitar la redundancia de datos, lo que reduce también el espacio de almacenamiento. Como resultado, se obtiene un modelo estrella pero con dimensiones expandidas.

Sin embargo, es importante considerar que este modelo conlleva un mantenimiento más difícil además de que la extracción de datos se vuelve más compleja. En este caso se requiere un mejor diseño para acomodar los requisitos de negocio de forma óptima.

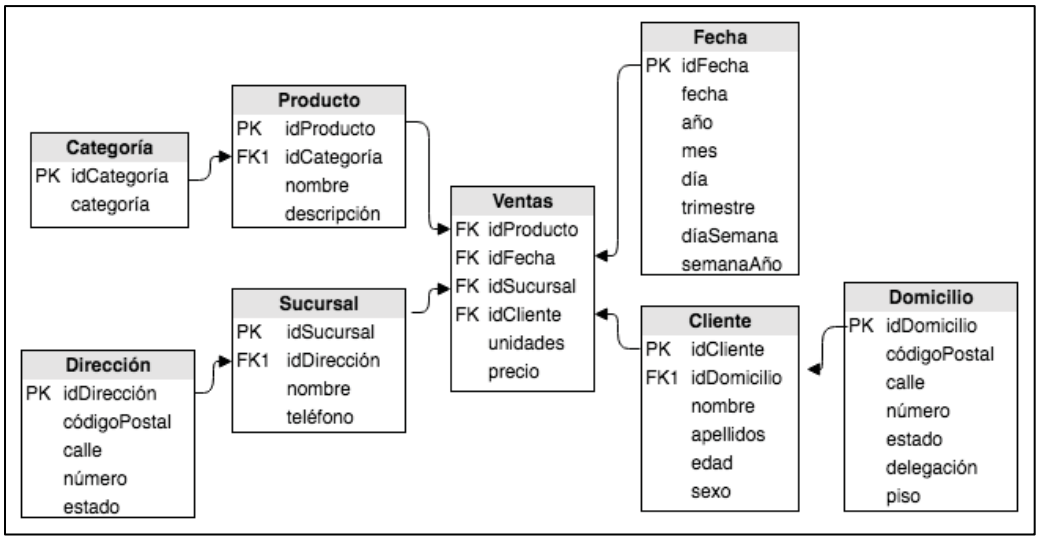


Figura 10. Modelo copo de nieve

1.10 Beneficios del Procesamiento Analítico en Línea

Entre los beneficios a destacar se encuentran los siguientes [18]:

• Incrementar la productividad.

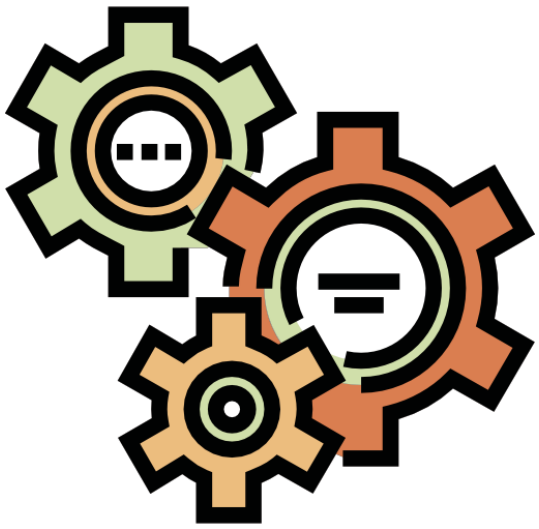
• Permitir independencia de los usuarios con respecto al departamento de tecnologías de la información.

• Mejor desempeño, gracias a la velocidad de las consultas.

• Los usuarios pueden modelar sus requerimientos de negocio de acuerdo a sus propias métricas y dimensiones.

• A diferencia de las bases de datos OLTP, el historial de datos se acumula a largo plazo.

Los conceptos previamente presentados apoyan a los usuarios de negocio a comprender la teoría básica de los almacenes de datos. A continuación se presenta un breve resumen sobre las herramientas con las que es posible llevar a cabo un proceso de almacenamiento de datos.



CAPÍTULO 2. HERRAMIENTAS PARA ALMACENAMIENTO DE DATOS

2. TECNOLOGÍAS PARA INTELIGENCIA DE NEGOCIOS

La inteligencia de negocios, mejor conocida por su traducción en inglés *business intelligence* (BI), se ha vuelto tendencia en los últimos años a consecuencia del incremento exponencial en el almacenamiento de datos. A este término se le atribuyen los procesos, las herramientas y las estrategias de obtención de datos para brindar información de utilidad a una organización. Esta nueva tendencia ha generado un nuevo mercado y la oportunidad de crear productos para una amplia gama de proveedores. [12]

2.1 Herramientas de código abierto

La flexibilidad en la construcción de software ha establecido un mejor equilibrio de costos contra eficiencia. Los usuarios pueden ajustar sus presupuestos económicos o adaptar el software de acuerdo a sus requerimientos de negocio. [2]

2.1.1 Modelo de licencias

En el mercado de software existen dos maneras de publicar un producto: el software desarrollado como código abierto o el software comercial tradicional. El software de código abierto puede ser compartido y modificado. Es mantenido por una comunidad de desarrolladores, sin motivos de lucro. Este software está disponible para los usuarios de manera gratuita.

En el caso del software comercial, se ofrecen productos cuyo código fuente no está alterado. La finalidad es obtener beneficios monetarios y posteriormente, obtener más ganancias por soporte, servicios, etc. En contraste, los proveedores de software de código abierto ofrecen productos con más flexibilidad y menor costo, generando una mejor interacción entre clientes y desarrolladores.

2.2 Cuadrante mágico de Gartner

La compañía Gartner es una empresa dedicada al análisis de empresas enfocadas a diferentes ámbitos empresariales. Uno de estos ámbitos es la inteligencia de negocios. Gartner presenta anualmente un análisis de investigación (conocido como cuadrante mágico) en el que cataloga a los proveedores de herramientas (en este caso de BI) en dos categorías y cuatro criterios. [12]

En el **eje x**, se define la categoría integridad de visión y representa el conocimiento de los proveedores sobre cómo se puede aprovechar el momento actual del mercado para generar valor. En el **eje y** se define la capacidad de ejecutar, donde mide la habilidad de los proveedores para ejecutar con éxito su particular visión del mercado. El cuadrante tiene un criterio por sector en función de las tendencias actuales de las compañías y la de sus productos.

Figura 11. Cuadrante mágico de plataformas [12]

• **Líderes**: Capacidad de visión del mercado y la habilidad para ejecutar.

• **Retadores o aspirantes**: Buenas funcionalidades pero menor variedad de productos.

• **Visionarios**: Capacidad para anticiparse a las necesidades del mercado pero sin medios suficientes para lograr implantaciones globales.

• **Jugadores de nicho**: Sin puntuación suficiente en ninguna de las dos categorías.

En este trabajo se han incluido tres de las herramientas presentadas en el cuadro mágico de Gartner: Microsoft (ubicada en el cuadrante de líderes), Pentaho y Oracle (ambas herramientas se encuentran en el cuadrante de jugadores de nicho). La selección fue realizada debido a dos factores, el primero fue que estas empresas tienen una gran trayectoria y ofrecen soluciones de BI. El segundo factor, es que ofrecen las herramientas para cada parte del proceso, para ser utilizadas por los usuarios. Algunas de las compañías que se encuentran en el cuadro mágico ofrecen soluciones incompletas (es decir, no cubren las 4 partes del proceso) o simplemente ofrecen soluciones como un servicio; el cliente entrega sus datos y las compañías procesan la solución.

2.2.1 Proveedores de herramientas de almacenamiento de datos

• **Pentaho**: Ofrece una amplia gama de capacidades comerciales de código abierto, incluyendo reportes, análisis, paneles, minería de datos, integración de datos y una plataforma que se ha convertido en la suite de código abierto más popular del mundo. Pentaho es un software de inteligencia empresarial de código abierto con software de aplicación para reportes de empresas, análisis, tableros de mando, minería de datos, flujo de trabajo y capacidades ETL. Posee una versión de comunidad y una versión comercial que puede ser obtenida por medio de una suscripción anual. La versión comercial incluye características extra añadiendo servicios de soporte. [10]

• **Oracle:** Desarrolla, fabrica, comercializa, distribuye y suministra software de base de datos y *middleware*, así como software de aplicaciones. Su negocio se divide en software y servicios. Provee sistemas completos, abiertos e integrados de software y hardware comercial. Oracle se encuentra en ediciones de comunidad con recursos limitados, y una edición comercial que incluye características avanzadas de almacenamiento y minería de datos. [8]

• **Microsoft**: Microsoft ofrece una amplia gama de capacidades de inteligencia de negocios y análisis. Cuenta con una gran variedad de herramientas, entre las que se encuentran las soluciones para integración de datos, OLAP y creación de reportes. Recientemente, Microsoft ha añadido Power BI para cumplir con la creación de tableros de mando y así poder competir correctamente contra otros proveedores de herramientas de almacenamiento de datos. Las herramientas ofrecidas por Microsoft se integran muy bien con otras herramientas propias de la compañía, tales como Word o Excel. [12]

2.3 Herramientas para procesos de integración de datos

La integración de datos comprende el conjunto de aplicaciones, productos, técnicas y tecnologías que permiten una visión única consistente de los datos. Entre las funciones de estas herramientas se encuentran: gestión y administración de servicios, extracción, transformación, carga y gestión de datos. La integración puede darse en cuatro grandes áreas [3]:

• **Integración de datos:** Da una visión única de todos los datos de negocio.

• **Integración de aplicaciones:** Da una visión unificada de todas las aplicaciones, pueden ser internas o externas a la empresa.

• **Integración de procesos de negocio:** Da una visión unificada de todos los procesos de negocio.

• **Integración de la interacción de los usuarios:** Proporciona una interfaz personalizada para datos, aplicaciones y procesos de negocio.

2.3.1 Técnicas de integración de datos

• **Consolidación de datos:** Los cambios realizados se capturan desde múltiples orígenes y luego se propagan a un entorno destino.

• **Datos federados:** Proporciona a las aplicaciones una visión lógica virtual común de una o más bases de datos. De este modo, es posible acceder a diferentes orígenes de datos y crear una visión de este conjunto como si fuera una base de datos única e integrada.

• **Captura de cambios en los datos:** Se obtienen los cambios realizados en las fuentes de origen para ser propagados a los entornos destino. De este modo se busca mantener la consistencia con las fuentes de origen.

• **Propagación de datos:** Los datos de un lugar de origen se copian hacia un destino (local o remoto).

• **Técnicas híbridas:** Un conjunto de varias técnicas de integración. [3]

2.3.2 Tecnologías para integración de datos

• **Integración de información empresarial:** Consiste en acceder por medio de las aplicaciones a los datos dispersos de diferentes fuentes (mercados de datos, archivos de texto, servicios web, etc) como si todos estuvieran en una base de datos común.

• **Extracción, transformación y limpieza:** Consiste en obtener los datos de los orígenes, transformarlos (darles formato de acuerdo a las reglas de negocio) y luego cargarlos en los entornos destino. Se basa en técnicas de consolidación.

• **Replicación de datos empresariales:** Detecta los cambios en las fuentes de origen. El concepto es casi igual a las técnicas de integración de propagación y de captura de cambios en los datos. [3]

2.3.3 Integración de datos en el contexto de Pentaho

Pentaho Data Integration (PDI), anteriormente llamado Kettle, es una solución de integración de datos programada en Java, orientada completamente al usuario y basada en un enfoque de metadatos. Cada uno de los procesos se encapsulan en metadatos para ejecutarse a través del motor ETL. Esta herramienta permite obtener datos de múltiples fuentes de origen, despues se cargan en un almacén de datos para que posteriormente la información consolidada sea de utilidad. [3]

2.3.4 Integración de datos en el contexto de Oracle

Oracle utiliza su herramienta Oracle Warehouse Builder. Es una herramienta que proporciona una solución integrada para diseñar e implementar almacenes de datos, mercados de datos y aplicaciones de inteligencia empresarial. Warehouse Builder también apoya el ciclo completo de gestión de la información. [7]

2.3.5 Integración de datos en el contexto de SQL Server

Microsoft ofrece una potente suite que brinda diferentes soluciones. Al momento de descargar, el usuario selecciona los componentes que desea incluir en su kit de herramientas. Para integración de datos, Microsoft cuenta con SQL Server Integration Services (SSIS), una herramienta que se utiliza para importar, limpiar y validar datos. Facilita las complejas cargas de datos que son comunes a las soluciones de almacenamiento de datos. [6]

2.4 Herramientas para soporte OLAP

Un proceso analítico en línea (OLAP) es un método para organizar datos (especialmente metadatos) sobre un sistema multidimensional, cuyo objetivo es recuperar y manipular datos y combinaciones de los mismos a través de consultas o reportes.

Una herramienta OLAP está formada por un motor y un visualizador. El motor es el concepto que se acaba de definir. El visualizador OLAP es la interfaz que permite consultar, manipular, reordenar y filtrar datos existentes. Las estructuras OLAP permiten realizar consultas que serían sumamente complejas mediante SQL. [3]

2.4.1 Elementos OLAP

OLAP permite el análisis multidimensional. Existen diferentes elementos comunes a las diferentes tipologías OLAP [3]:

• **Esquema:** Colección de cubos, dimensiones, tablas de hecho y roles.

• **Roles:** Permisos asociados a determinados usuarios.

• **Tabla de hecho:** Tabla que contiene los datos centrales del negocio.

• **Dimensión:** Contiene datos asociados a la tabla de hechos (atributos).

• **Métricas:** Indicadores numéricos establecidos por los usuarios.

• **Cubo:** Colección de dimensiones asociadas a una tabla de hecho.

• **Miembro:** Punto en la dimensión de un cubo que pertenece a un determinado nivel de una jerarquía.

• **MDX:** Acrónimo de *multidimensional expressions*. Es el lenguaje de consulta de estructuras OLAP.

• **Nivel:** Grupo de miembros en una jerarquía con los mismos atributos y el mismo nivel de profundidad en la jerarquía.

• **Jerarquía:** Conjunto de miembros organizados en niveles.

2.4.2 Reglas OLAP de E. F. Codd

La definición de OLAP presentada anteriormente se basa en las 12 leyes que acuñó Edgar F. Codd en 1993. Los fabricantes de software intentan, en la medida posible, cumplir con estas reglas [3]:

• **Vista conceptual multidimensional:** Trabajar a partir de métricas de negocio y dimensiones.

• **Transparencia:** Formar parte de un sistema abierto que soporta fuentes de datos heterogéneas.

• **Niveles de dimensiones y de agregación ilimitados:** No hay límite con el tamaño del cubo.

• **Manejo dinámico de matrices dispersas:** Poder diferenciar valores vacíos de valores nulos y además poder ignorar las celdas sin datos.

• **Operaciones cruzadas entre dimensiones sin restricciones:** Las operaciones entre dimensiones no deben restringir las relaciones entre celdas.

• **Accesibilidad:** Presentar el servicio con un único esquema lógico de datos.

• **Rendimiento de reportes consistente:** Los reportes no deben degradarse cuando el número de dimensiones incrementa.

• **Manipulación de datos intuitiva:** Debe aplicarse la máxima usabilidad de los usuarios (vista desde la usabilidad de un usuario nuevo).

• **Arquitectura cliente/servidor:** Facilidad de interacción y la colaboración.

• **Reportes flexibles:** Simplicidad para crear reportes. Los cambios en el modelo de datos deben reflejarse automáticamente en esos reportes.

• **Dimensionalidad genérica:** Mismas funcionalidades aplicables entre una dimensión y otra.

2.4.3 OLAP en el contexto de Pentaho

Mondrian es el motor/servidor OLAP integrado en Pentaho y ha sido renombrado como Pentaho Analysis Services. Este motor utiliza un visualizador OLAP y con dos herramientas de desarrollo. Los visualizadores para la versión de comunidad son los siguientes [3]:

• **JPivot:** Cliente OLAP basado en JSP, permite realizar consultas tanto MDX como a partir de elementos gráficos que se visualizan en un navegador web. Es un visualizador analítico que permite la posibilidad de extenderse mediante desarrollo.

• **PAT:** Es el acrónimo de Pentaho Analysis Tool. Las características de PAT son: arrastrar y soltar (*drag and drop*); uso de temas; uso de olap4j como visualizador OLAP de múltiples motores; uso de todas las funcionalidades de Jpivot y la extensión de sus funcionalidades.

Para la versión Enterprise se cuenta con:

• **Pentaho Analyser:** Es una solución que ofrece: arrastrar y soltar (*drag and drop*); creación de nuevas medidas calculadas; buscador de objetos y funcionalidades encapsuladas.

Las herramientas de desarrollo utilizadas:

• **Pentaho Schema Workbench:** Permite crear todos los objetos que soporta Mondrian: esquema, cubo, dimensiones, métricas, etc.

• **Pentaho Aggregation Designer:** Permite analizar la estructura del esquema de Mondrian contra la cantidad de datos a recuperar, a partir de dicho análisis permite recomendar la creación de tablas agregadas.

2.4.4 OLAP en el contexto de Oracle

Oracle BI Answers es la solución OLAP de parte de Oracle. Permite a los usuarios explorar e interactuar con los datos, y presentar y visualizar información mediante gráficos, tablas dinámicas e informes. Es posible configurar un informe para actualizar los resultados en tiempo real. Los informes creados con Oracle BI Answers se pueden guardar en Oracle BI Presentation Catalog y se integran en cualquier página de Oracle BI o tablero de mandos. [16]

Oracle BI Answers cuenta con una vista lógica de los datos, de modo que los usuarios tienen oculta la complejidad de la estructura de datos. Esto evita que los usuarios hagan consultas que puedan generar desbordamiento. Oracle BI Answers también permite crear fácilmente gráficos, tablas dinámicas, informes y paneles visualmente atractivos que pueden ser guardados, compartidos, modificados, formateados o incrustados en los tableros de mando. [7]

2.4.5 OLAP en el contexto de SQL Server

SQL Server Analysis Services (SSAS) es el motor de datos analíticos en línea que Microsoft utiliza en la toma de decisiones y en el análisis empresarial. Proporciona los datos analíticos para informes empresariales y aplicaciones cliente como Power BI, Excel, informes de Reporting Services y otras herramientas de visualización de datos. [14]

SSAS proporciona los mecanismos de almacenamiento y consulta para los datos utilizados en cubos OLAP para el almacén de datos. Incluye un componente que le permite crear estructuras de minería de datos. Los modelos de minería de datos son objetos que contienen datos de origen que se han procesado utilizando uno o más algoritmos. [6]

2.5 Herramientas para reportes

Las empresas acumulan cada día, más y más datos. En el pasado era común utilizar archivos de texto para poder gestionar información de valor. Con el paso del tiempo llegaron las bases de datos para poder solucionar el problema del exceso de datos. Posteriormente, se generó la necesidad de obtener físicamente esta información para facilitar la toma de decisiones. Actualmente, los usuarios necesitan generar y distribuir reportes para conocer el estado del negocio y poder tomar decisiones a todos los niveles (operativo, táctico y estratégico).

Se define como un reporte al documento a través del cual se presentan los resultados de uno o varios procesos de negocio. Contiene elementos como tablas o gráficos para comprender la información presentada. Por otro lado, se define como plataforma de *reporting,* aquellas soluciones que permiten diseñar y gestionar reportes. [3]

2.5.1 Tipos de reportes

Existen diferentes tipos de reportes en función de la interacción [3]:

• **Estáticos:** Reportes con formato preestablecido inamovible (por ejemplo, PDF).

• **Parametrizados:** Reportes con parámetros de entrada y permiten múltiples consultas (por ejemplo, hojas de calculo con Excel).

• **Ad-hoc:** Reportes creados a partir de la capa de metadatos que permite usar el lenguaje de negocio propio (desarrollados por omisión desde la aplicación).

2.5.2 Elementos de un reporte

Un reporte puede estar formado por [3]:

• **Tablas:** Estructura en forma de matriz para mostrar información.

• **Texto:** Proporciona descripciones necesarias para entender el contenido de los elementos del reporte.

• **Alertas visuales y automáticas:** Elementos gráficos automatizados para mostrar cambios en el estado de la información.

• **Gráficos:** Elementos visuales para representar información de forma simple.

• **Mapas:** Elementos para mostrar información geolocalizada.

• **Métricas:** Indicadores para conocer el estado de un proceso de negocio.

2.5.3 Tipos de métricas

Los reportes incluyen métricas de negocio definidas por los usuarios. A continuación se muestran las medidas existentes [3]:

• **Métricas:** Indicadores que miden el proceso de una actividad. Se consideran dos tipos de métricas:

a) Métricas de realización de actividad: Miden la realización de una actividad (por ejemplo, la participación de una persona en un evento).

b) Métricas de resultado de una actividad: Miden los resultados de una actividad (por ejemplo, la cantidad de puntos de un jugador en un partido).

• **Indicadores clave:** Son medidas del desempeño de una organización. Todos los indicadores son métricas (aunque no todas las métricas son indicadores). Un indicador clave puede ser la suma de varias métricas, para mostrar un resultado corporativo a alcanzar. Entre los indicadores principales se encuentran:

a) *Key Performance Indicator*: Indicador de desempeño en un proceso. Por ejemplo, si se desea incrementar la venta de un producto de 800 a 1000 unidades, el indicador significa incrementar un 25% la venta de ese producto.

b) *Key Goal Indicator*: Indicador de metas. Por ejemplo, en el contexto de ventas, lograr vender 1000 unidades de un producto.

2.5.4 Reportes en el contexto de Pentaho

Pentaho Reporting es el motor de reportes de Pentaho. Permite procesar reportes generados con la herramienta de diseño. Con Pentaho Reporting es posible ejecutar un proceso ETL para recoger información, pasarla al reporte, ejecutar el reporte y enviarlo vía correo. Existen tres herramientas de desarrollo [3]:

• **Pentaho Report Designer:** Editor programado en Java, encapsula la lógica de un reporte en un archivo XML.

• **Pentaho Metadata:** Herramienta de diseño para crear vistas de negocio. Permite crear una capa de metadatos basada en la información consolidada en el almacén de datos.

• **WAQR (Web Ad-hoc Query Reporting):** Editor web para reportes basado en templates creados con Pentaho Report Designer.

2.5.5 Reportes en el contexto de Oracle

Oracle BI Publisher es la solución de reportes de Oracle. Los formatos de reporte pueden ser diseñados utilizando Microsoft Word o Adobe Acrobat. Permite obtener datos desde varias fuentes de datos en un solo documento de salida. Puede enviar reportes por impresora, correo electrónico o fax.

Esta herramienta puede permitir a los usuarios editar y administrar reportes de forma colaborativa en servidores web WebDAV (*Distributed Authoring and Versioning*) basados en Web. También aprovecha los servicios comunes de metadatos, seguridad, cálculo, almacenamiento en caché y generación de peticiones inteligentes. [7]

2.5.6 Reportes en el contexto de SQL Server

La solución de reportes de SQL Server es SQL Server Reporting Services (SSRS). Esta herramienta incluye un diseñador de consultas visuales para cubos SSAS, lo que facilita la creación rápida de reportes. Incluye otro componente de generación de reportes para ser utilizado por los analistas, en lugar de los desarrolladores. También cuenta con varias herramientas de cliente: una interfaz web, Web Parts para Microsoft Office SharePoint Server y componentes de cliente para aplicaciones de Windows Forms. [6]

2.6 Herramientas para tableros de mando

Con las herramientas previamente vistas (cubos OLAP y reportes) es posible proporcionar información a los usuarios. Sin embargo, debido a la gran cantidad de información que contiene (comúnmente) un almacén, estas herramientas pueden ser inadecuadas para que los usuarios analicen y tomen decisiones de forma rápida. Como solución, surge el concepto de tablero de mando.

Se define como un tablero de mando o *dashboard* al sistema que proporciona información consolidada de alto nivel sobre los indicadores fundamentales de negocio de una organización. Permite monitorear los procesos de negocio dado que muestra información crítica a través de elementos gráficos de fácil comprensión. El tiempo de actualización suele ser cercana al tiempo real. [3]

2.6.1 Elementos de un tablero de mando

Un tablero de mando puede contener diversos elementos combinados [3]:

• **Tabla:** Estructura en forma de matriz para mostrar información, puede ser estática, dinámica, o incluso un análisis OLAP.

• **Métricas:** Indicadores sobre proceso de actividades, principalmente KPI.

• **Alertas visuales y automáticas:** Elementos gráficos automatizados para mostrar cambios en el estado de la información.

• **Listas:** Elementos ordenados de texto, formadas por KPI.

• **Menús de navegación**: Elementos para facilitar al usuario la interacción con los elementos del tablero de mando.

• **Gráficos:** Elementos visuales para representar información de forma simple.

• **Mapas:** Elementos para mostrar información geolocalizada.

2.6.2 Proceso de creación de un tablero de mando

Crear un tablero de mando conlleva diversos pasos [3]:

1.- Seleccionar los datos a mostrar (indicadores requeridos).

2.- Seleccionar el formato de presentación de los elementos convenientes.

3.- Combinar los datos y presentarlos en conjunto. Organizar los elementos para dar una vista coherente a los datos.

4.- Planificar la interactividad del usuario.

5.- Implementar el tablero de mando.

2.6.3 Tableros de mando en el contexto de Pentaho

• **Community Dashboard Framework:** Es un *plugin* (complemento) que permite construir tableros basados en CSS y plantillas. Se encuentra incluido por omisión en Pentaho Server. Requiere conocimientos de desarrollo web. [3]

• **Pentaho Dashboard Designer:** Es un *plugin* para la versión profesional que permite crear un tablero de mando de forma sencilla. Entre algunas de sus mejores, se destaca el uso de elementos preexistentes (reportes, gráficos, OLAP) y el uso de plantillas predefinidas.

2.6.4 Tableros de mando en el contexto de Oracle

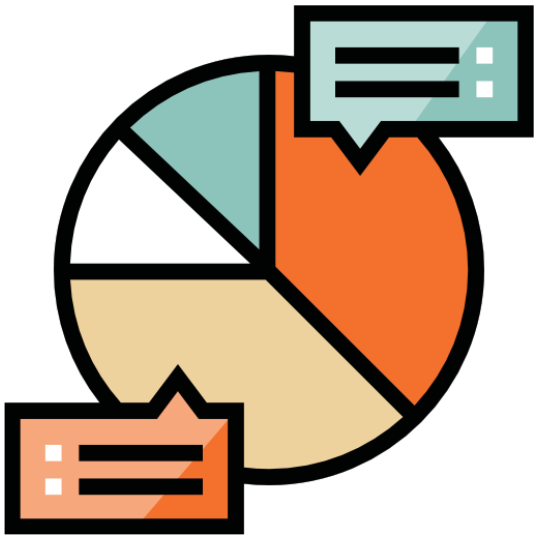
Oracle BI Interactive Dashboards ofrece un acceso intuitivo a la información. Permite trabajar con reportes en vivo, avisos, gráficos, tablas, tablas dinámicas y gráficos en una arquitectura web pura. Los usuarios cuentan con capacidad para perforar, navegar, modificar e interactuar con estos resultados. Además puede agregar contenido de una amplia variedad de fuentes (incluyendo internet, servidores de archivos compartidos y repositorios de documentos). [7]

2.6.5 Tableros de mando en el contexto de SQL Server

La suite de BI de Microsoft no incluye por omisión una herramienta de tableros de mando. Recientemente Microsoft relleno esta brecha con una poderosa herramienta, Power BI. Esta herramienta sirve para analizar datos y compartir conocimientos. Permite mostrar información en un solo lugar, en tiempo real.

Power BI unifica los datos de la organización, ya sea en la nube o en el entorno local. Puede conectar bases de datos de SQL Server, modelos de Analysis Services y muchas otras fuentes de datos, destacando una gran variedad de herramientas propias de Microsoft. [15]

A continuación se presentarán ejemplos prácticos sobre el uso de estas herramientas, ilustrando brevemente las características más destacadas sobre cada una de estas.



CAPÍTULO 3. PROBANDO HERRAMIENTAS PARA ALMACENES DE DATOS

3. PROBANDO HERRAMIENTAS PARA ALMACENES DE DATOS

Para realizar una comparativa entre las herramientas de código abierto expuestas en este trabajo, se llevará a cabo un ejemplo del proceso de almacenamiento de datos con cada una de las herramientas. Este proceso contempla las herramientas de ETL, OLAP, reportes y tableros de mando con cada uno de los proveedores seleccionados (Oracle, Pentaho y Microsoft). La compañía Gartner propone una evaluación de las herramientas de acuerdo con las siguientes secciones [12]:

a) Infraestructura: Contempla el rendimiento de plataforma, administrar aplicaciones analíticas en la nube, la conectividad de los orígenes de datos, la seguridad de la plataforma y la administración de usuarios.

b) Gestión de datos: Permitir a los usuarios compartir el mismo modelo semántico y metadatos. Acceder, integrar, transformar y cargar datos en una capa de almacenamiento. Combinación de datos de diferentes fuentes, creación de modelos analíticos tales como conjuntos, grupos y jerarquías.

c) Análisis y creación de contenido: Acceder fácilmente a capacidades de análisis avanzadas (dentro y fuera de la plataforma). Desarrollar y entregar contenido a dispositivos móviles en un modo de publicación interactivo. Exploración de datos mediante la manipulación de gráficos. Exploración de datos mediante la manipulación de gráficos.

d) Compartir resultados: Crear y modificar contenido analítico, visualizaciones y aplicaciones. Publicar e implementar el contenido analítico a través de diversos tipos de salida y métodos de distribución. Compartir y discutir información, análisis, contenido analítico y decisiones.

Esta propuesta será tomada como base para la evaluación realizada en el siguiente capítulo. Sin embargo, los aspectos más importantes a evaluar serán aquellos que sean más visibles para los usuarios.

A continuación se presenta una demostración sobre el procedimiento para realizar un almacenamiento de datos. Los pasos correspondientes a la carga de datos (los datos iniciales de las fuentes de datos) no serán mostrados ya que no son el objetivo de este trabajo. Se mostrará la teoría de todos los requerimientos iniciales, posteriormente, la parte del proceso donde los usuarios deben procesar sus datos (ETL) será el punto de partida para comenzar la evaluación.

3.1 Definición de requerimientos

A nivel de usuario, se requiere conocer una o más métricas relacionadas con un determinado modelo de negocio. Por ejemplo, en los sitios web, es deseable conocer el tráfico de usuarios, que tipo de páginas visitan, o que tipo de productos suelen ser de mayor interés. En un modelo de negocio basado en ventas, se requieren conocer a detalle los atributos (fechas, productos, edades, etc) que permitan introducir estrategias para optimizar los beneficios.

Comúnmente los negocios que ofertan productos intentan vender más de manera constante, pero al no obtener el resultado esperado es conveniente tomar decisiones que limiten o anulen el riesgo financiero. Algunas métricas pueden ser los clientes o el rubro poblacional que más solicita un producto (por edad, por genero, etc.). Aquí aplica para clientes locales o internacionales.

3.2 Modelo dimensional

El diseño del modelo dimensional debe ser extensible de acuerdo a posibles necesidades futuras. Comenzando con este principio, las dimensiones y la tabla de hecho deben contener valores que puedan ser medidos y al mismo tiempo ser independientes, preferentemente por medio de una normalización de las tablas. En la siguiente figura, se muestra un ejemplo de modelo de almacén de datos aplicable a diferentes modelos de negocio.

La tabla de hechos Venta conecta con cada una de sus dimensiones, de modo que obteniendo un registro de venta se obtiene la llave primaria de todas las dimensiones existentes. Aquí el modelo estrella juega un papel importante debido a la facilidad de escalar un almacén de datos.

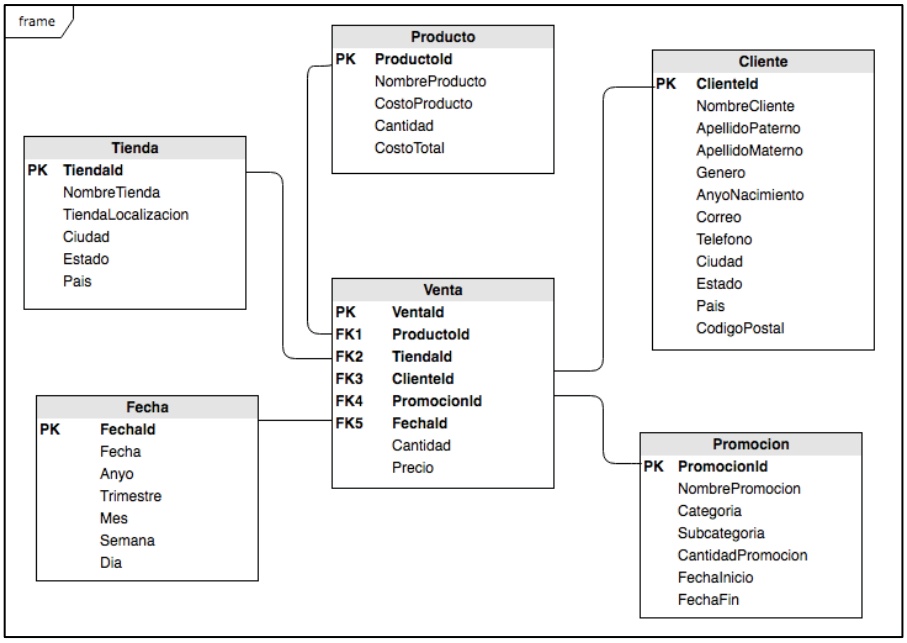


Figura 12. Modelo dimensional de ventas

3.3 Diseño físico

El modelo dimensional anterior se considera un modelo genérico debido a la sencillez para aplicarse a distintos tipos de negocios, por ejemplo, las ventas de un supermercado o las ventas de una tienda de ropa. Por la forma de su modelo estrella, es posible extender más tablas, añadir nuevos campos, o modificar la estructura para adaptarlo a un modelo de negocio en particular.

El diseño físico se compone de los motores de bases de datos y todas las herramientas que sirven para procesar y llevar a cabo el flujo de almacenamiento de datos. Cada herramienta tiene diferentes plataformas.

En el caso de Pentaho, es una herramienta multiplataforma, por lo que es posible usarla en equipos con sistemas operativos Linux, Windows y Mac. Anteriormente, Oracle contaba con ciertos componentes compatibles con el sistema operativo Mac. En la actualidad, solo ofrece sus productos para Linux y Windows. SQL Server es exclusivo de Windows, debido a que sus productos son comerciales y se ofrecen bajo su propio sistema operativo. En cuanto al diseño físico de las herramientas, se presentan los componentes utilizados en este trabajo:

Para Pentaho:

- Ubuntu 16.10
- Mondrian 3.14.0
- Pentaho Data Integration 7
- Pentaho Reporting 7.1
- Community Dashboard Editor

Para Oracle:

- Windows 7 Ultimate x64
- Oracle Database 11g Release 2
- Oracle Web Logic Server
- Oracle Business Intelligence 12c

Para SQL Server:

- Windows 7 Ultimate x64
- Microsoft .NET Framework 4.0
- Microsoft SQL Server 2012 Express (con herramientas avanzadas)
- Microsoft SQL Management Studio
- Visual Studio 2010
- Power BI

Para todas las plataformas:

- Java SE Development Kit 8u144

3.4 Restricciones de software

Los componentes de Pentaho son de código abierto, por lo que los usuarios son capaces de utilizarlos según sean sus necesidades, aunque debe utilizarse como usuario final. No se permite obtener ganancias de parte de la instalación o mantenimiento de las herramientas de Pentaho. Las ediciones comerciales incluyen mejoras que no distan mucho en cuanto a su versión de comunidad. En lo que corresponde a Oracle, es posible utilizar versiones de sus productos comerciales sin fines de lucro, es decir, pueden usarse para uso personal mientras no se monetice por el uso de este software, aun si estas son versiones empresariales.

Windows tiene software comercial que permite utilizar de manera gratuita durante un determinado tiempo de prueba. Posteriormente los usuarios pueden elegir comprar la licencia o cancelar la suscripción. Sin embargo, algunos de sus productos pueden ser descargados en modo desarrollador, con la finalidad de recibir retroalimentación de parte de los usuarios, únicamente cumpliendo con la misma condición de usarse sin fines de lucro.

3.5 Procesamiento de datos

En este trabajo no se llevará a cabo la carga de bases de datos debido a que no es el objetivo de este material. Se sobreentiende que un usuario de almacenes de datos comprende y aplica conocimientos básicos sobre bases de datos. Cuando una base de datos está poblada, existen factores que alteran la toma de decisiones al obtener reportes. Pueden ser datos duplicados, datos faltantes, con valores erróneos, etc.

Los procesos de extracción, transformación y carga, consisten en obtener datos de una o diferentes fuentes, transformar o “limpiar” los datos, y posteriormente, cargarlos en un documento o base de datos que permita su análisis.

Se considera necesario retirar los datos “sucios” o incompletos, para poder enfocarse en los datos que representen un valor a un usuario de negocio. Asimismo, la transformación de datos podría no necesariamente tener que limpiar datos, sino simplemente, filtrar por medio de diferentes fuentes los datos que se van a usar en el proceso de negocio.

3.5.1 Proceso ETL en Pentaho

Pentaho realiza procesos ETL mediante su herramienta Spoon. Esta herramienta está incluida por medio del paquete de Pentaho Data Integration junto con su motor Mondrian. Aunado a esto, se ejecuta utilizando una terminal. En principio esto suele no ser tan atractivo para los usuarios, ya que hay más tendencia por las aplicaciones de escritorio que suelen tener accesos directos. A cambio de esto, Pentaho ofrece una plataforma completamente versátil, capaz de integrarse fácilmente con otras herramientas sin necesidad de utilizar todo el kit de inteligencia de negocios. Para comenzar, una vez realizado el modelo dimensional, se procede a conectar la base de datos.

Pentaho ofrece múltiples conectores a bases de datos. También ofrece conectores a diferentes archivos de salida. Las operaciones utilizadas para realizar transformaciones son explícitamente llamadas de ese modo. Por otro lado, al proceso de gestionar y administrar procesos ETL de alto nivel se le conoce como trabajo. Como se muestra en la figura 13, Spoon ofrece conexión a las bases de datos más conocidas, como Oracle y SQL Server. El primer paso consiste en acceder a los datos de una base ya poblada.

Para generar un proceso ETL, se utiliza una combinación entre la interfaz gráfica, y los datos de las fuentes conectadas. Gráficamente consiste en arrastrar y soltar elementos del panel de componentes, y posteriormente extraer datos mediante las propiedades de cada elemento. Cada uno de estos elementos tiene una acción asociada que realizara una transformación hacia los datos.

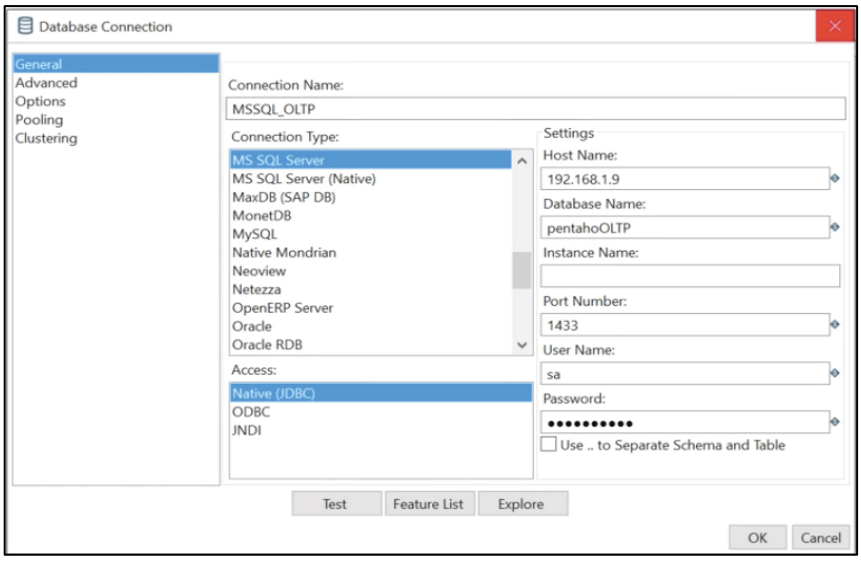


Figura 13. Conectores de bases de datos

En la siguiente figura se muestra un trabajo (*job*). Las líneas se conocen como saltos. Los saltos conectan pasos de transformaciones o entradas de trabajos con otros. Un trabajo es un conjunto de transformaciones que se llevan a cabo según el orden en que los saltos se encuentren ordenados Estos saltos podrían estar habilitados o no, dependiendo de si se aplican ciertos casos de prueba. Cada transformación realizada conlleva al siguiente salto. Si la ejecución de las transformaciones es exitosa, el proceso finalizara. En resumen, el proceso ETL realiza consultas para transformar los datos de manera simple para el usuario.

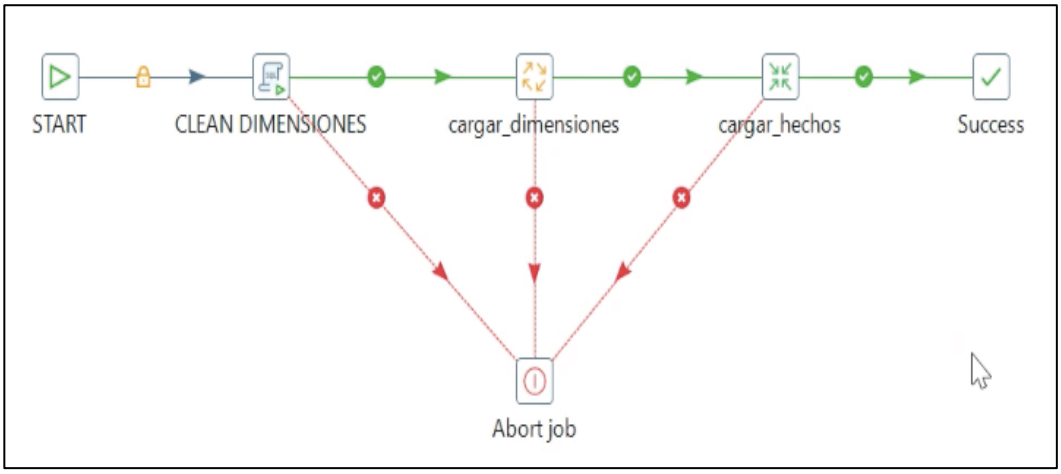


Figura 14. Diagrama de un trabajo en Pentaho

Cada que el usuario genera un trabajo, la interfaz de Spoon “sobreescribe” una consulta, es decir, el usuario puede aplicar alguna transformación sin tener que haber realizado la consulta con código. La plataforma lleva el proceso de manera automatizada, de modo que el beneficio se ve principalmente en el tiempo y en la eficiencia. En un proceso más complejo, existe la posibilidad de mezclar orígenes de datos o de tener múltiples salidas. Por ejemplo, combinar datos desde bases de datos diferentes, para posteriormente, transferir datos a uno o más archivos de diferente tipo. (Más detalles en el Anexo A, página 115)

Como se muestra en la siguiente figura, el trabajo ejecuta una transformación de mapeo de datos desde una base MYSQL y una base SQL Server mezclando los valores obtenidos en un punto destino, desde el cual serán transferidos a tres salidas, una dimensión, un archivo de texto plano y un archivo de hoja de cálculo (Excel). En un principio, la instalación y la ejecución no son simples, pero su usabilidad retribuye al usuario con su interfaz intuitiva y accesible.

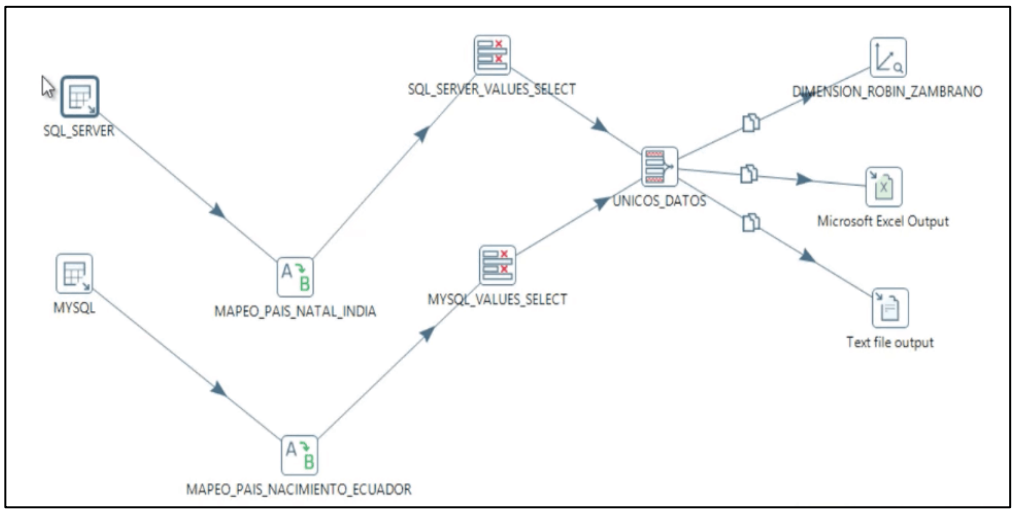


Figura 15. Mapeo de datos hacia diferentes salidas

3.5.2 Proceso ETL en Oracle

A diferencia de Pentaho, las herramientas de Oracle pueden ser descargadas por medio de un instalador. Al descargar una versión de Oracle Business Intelligence,

el usuario tiene la posibilidad de seleccionar que otros productos complementarios requiere según su necesidad de negocio. Esto puede tener ventajas para los usuarios, ya que al conocer otras herramientas que complementen sus necesidades, pueden elegir integrarlas con las que Oracle provee.

Oracle tiene una interfaz más formal, orientada a usuarios con conocimiento de bases de datos. Comenzando por la seguridad, Oracle solicita información al usuario para notificar sobre fallas en los servicios de Oracle que puedan presentarse. Al momento de ejecutar Warehouse Builder, es necesario que los usuarios ingresen sus datos de inicio de sesión que previamente hayan establecido al cargar su base de datos.

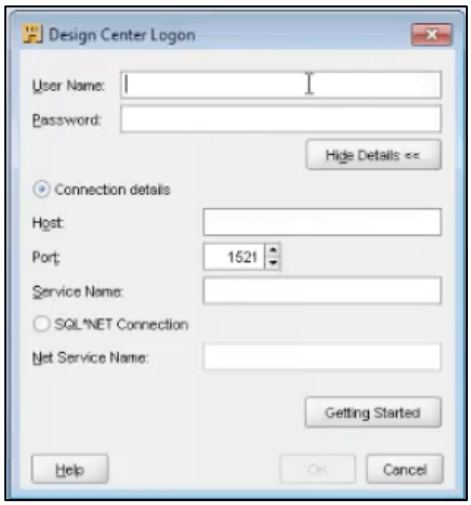


Figura 16. Inicio de sesión en Warehouse Builder

Posteriormente, realizar un proceso ETL es muy similar al que se lleva a cabo en Pentaho. Sin embargo, hay diferencias muy notables. Tiene un panel que ofrece muchas operaciones que pueden usarse. La tipografía y la parte gráfica también son más complejas para los usuarios. A simple vista, se puede observar que el entorno de Oracle es más robusto y cuenta con una interfaz no tan amigable para los usuarios. A cambio de eso, Oracle provee poderosas opciones que pueden usarse al momento de realizar un proceso ETL. Algunas de estas opciones se muestran en una paleta de componentes, como en la siguiente figura.

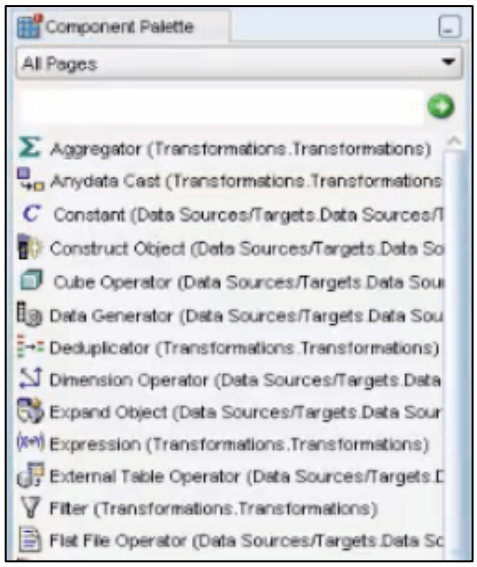


Figura 17. Componentes de transformación

Las operaciones de Oracle son una especie de atajo hacia las operaciones conocidas de bases de datos. En la siguiente figura se pueden observar las operaciones “join” y “agregación”, las cuales transformarán los datos de origen hasta guardarlos en una tabla de “staging”. Se pueden adaptar los parámetros que servirán como valores de entrada, limitando únicamente los campos que se requieren transformar.

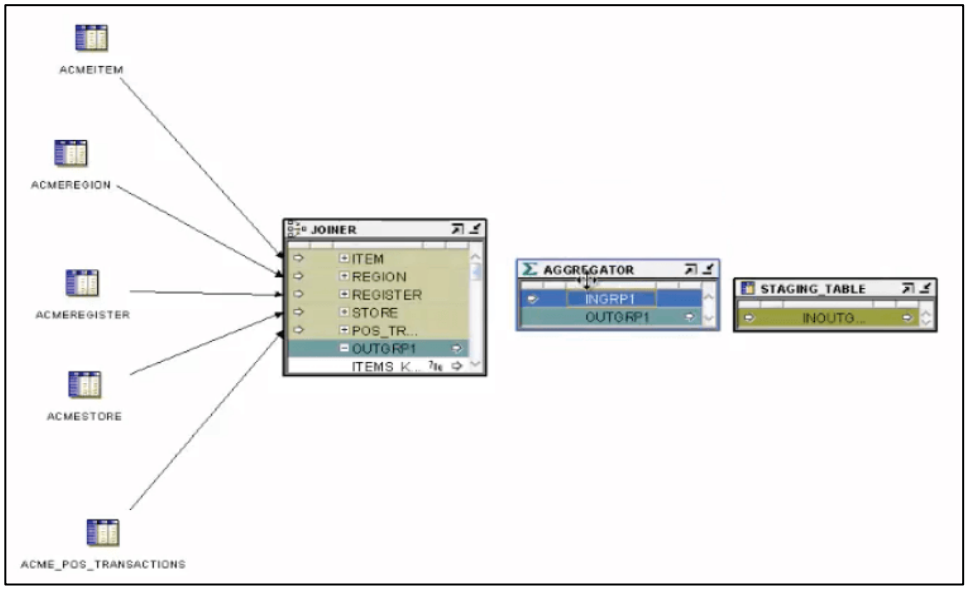


Figura 18. Operación de transformación en Warehouse Builder

Como se muestra en la figura 19, al crear una transformación se muestra un panel, en el lado izquierdo se encuentran todas las dimensiones del origen de datos y en el lado derecho superior se agrupan las dimensiones requeridas de las cuales se obtendrán sus datos. Es decir, en el lado derecho se construye la transformación con los parámetros de la columna izquierda. Esta interfaz no es tan intuitiva como la de Pentaho.

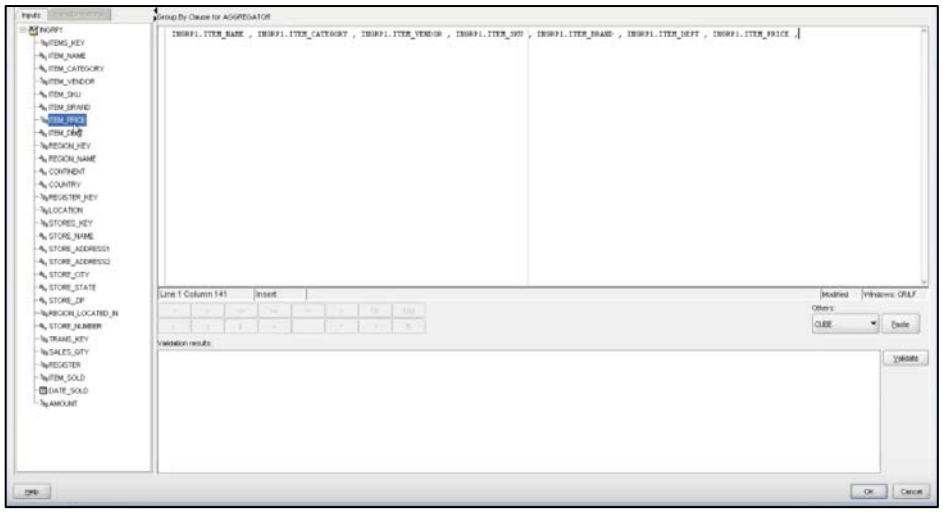


Figura 19. Selección de parámetros en transformación

Debido a su gran potencial de transformaciones se ve debilitada su usabilidad por la complejidad que requiere conocer el proceso que lleva a cabo cada uno los componentes. Una desventaja inicial a considerar, es el hecho de que la aplicación solicita a los usuarios instalar previamente una base de datos Oracle. Esto no implica que posteriormente no sea posible usar otras fuentes de datos, ya que Oracle también tiene servicios para usar bases de datos como SQL Server.

Otro punto a considerar, es que hay un cambio entre las herramientas de inteligencia de Oracle 11c y 12c. La versión 11c gestiona sus repositorios por medio de Repository Creation Utility. En la versión 12c, se requiere cambiar esta herramienta por el middleware de Oracle Web Logic Server. (Más detalles en el Anexo A, página 120)

3.5.3 Proceso ETL en SQL Server

La instalación en los componentes de SQL Server es relativamente más sencilla, ya que debido a que solo puede ser instalados en Windows, algunos ya están instalados por omisión. Al igual que Oracle, el usuario puede elegir que versión de herramientas utilizar, así como los componentes que quiera descargar. Estas dos herramientas cuentan con un amplio catálogo de productos. Los productos correspondientes a Pentaho pueden ser descargados desde la página de Pentaho, aunque no todos se incluyen aquí, algunos se encuentran en repositorios independentes.

SQL Server utiliza su herramienta SQL Server Management Studio para llevar a cabo los procesos de creación de bases de datos. La generación del proceso ETL conlleva el uso de Visual Studio, el cual puede estar instalado por omisión en los equipos con Windows. Con Visual Studio se integra la base de datos para realizar el diseño del cubo y las transformaciones a aplicar. En la figura 20 se muestra un script de tablas de una base hecha en SQL Server Management Studio. Este es uno de los múltiples orígenes de datos que pueden implementarse. Entre los más significativos se encuentran los archivos Excel, las conexiones FTP, HTTP, SMTP, ODBC, objetos propios de la plataforma, etc.

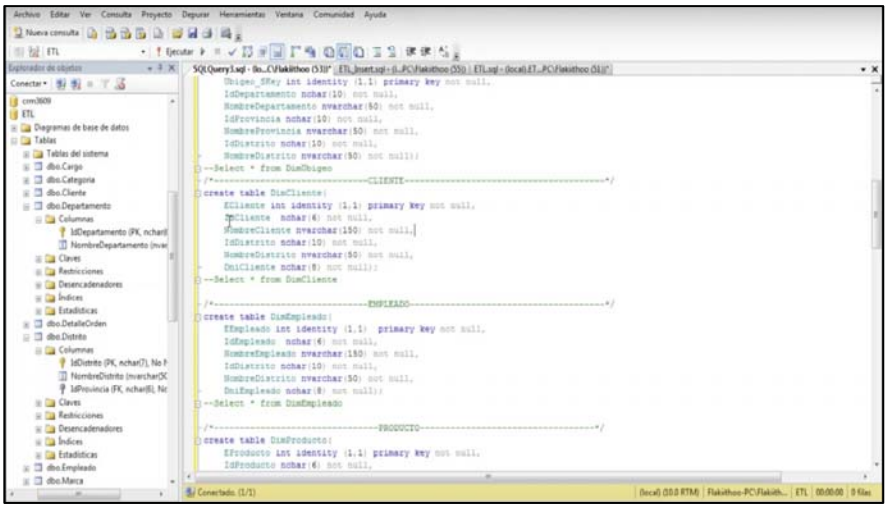


Figura 20. Tablas de base de datos en SQL Server

A diferencia de Oracle y Pentaho, SQL Server tiene un editor de origen de datos cuando estos son proporcionados por una base. Esta funcionalidad permite a los usuarios escribir consultas manualmente. En principio no sería muy diferente a un motor de base de datos, pero puede ser utilizado para filtrar datos dependiendo de las necesidades propias del usuario.

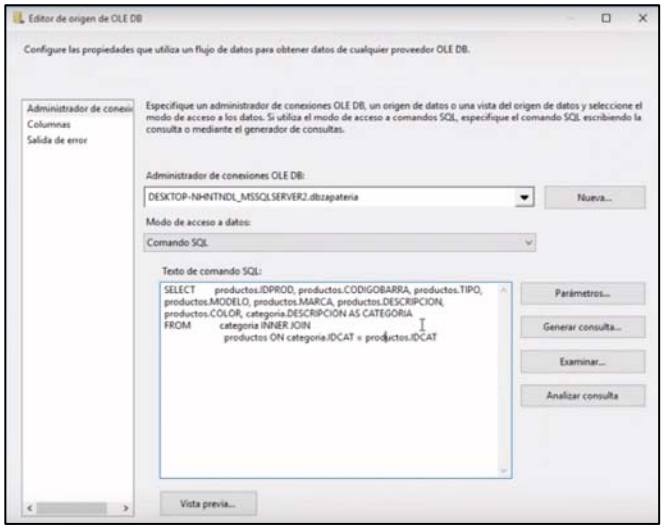


Figura 21. Editor de origen de OLE DB

Visual Studio integra la herramienta de integración de datos de SQL Server (SSIS) y permite realizar operaciones en procesos que son llamados “paquetes” en vez de los “trabajos” de Pentaho y Oracle.

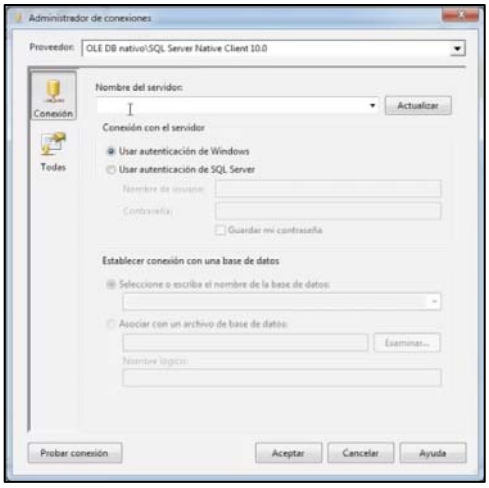


Figura 22. Integración de base de datos en Visual Studio

Esto no implica mucha diferencia en su funcionalidad. Al igual que Oracle, tiene un panel que incluye las operaciones de transformación. Tiene una interfaz muy accesible e incluso intuitiva para aquellos usuarios que no tienen tanto conocimiento de la herramienta. En la figura 23 se muestra un ejemplo de transformación ETL. Consiste en tomar un origen de datos para copiar una determinada columna hacia un nuevo destino. En lo que a SQL Server corresponde, cuenta con mucha facilidad de instalación. El ambiente se ve favorecido por la descripción de texto y los iconos en cada función.

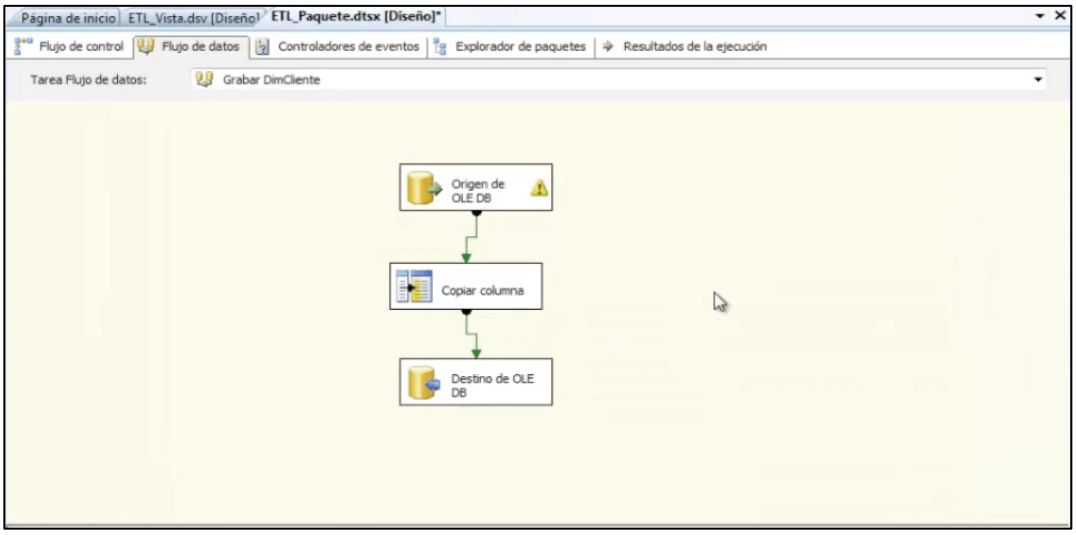


Figura 23. Transformaciones en Visual Studio

Se vuelve una experiencia simple utilizar esta herramienta debido a que Microsoft aporta muchos tutoriales para la mayor parte de las utilidades de sus componentes. Las funciones de transformación pueden ser relativamente complejas considerando que se pueden realizar diferentes mezclas y “joins”. Aquí la diferencia es la posibilidad de ingresar consultas mediante el editor. Las funciones en Pentaho son simples pero eficientes. En cambio, Oracle tiene una gran desventaja, al ofrecer mucho potencial de transformación se pierde la accesibilidad para el usuario, debido a que sus iconos y sus descripciones no son fáciles de entender. (Más detalles en el Anexo A, página 132)

3.6 Análisis de datos

Como se vió en el capítulo 2, el objetivo de utilizar OLAP es tener un alto desempeño al realizar una consulta cuando se tienen grandes volúmenes de datos. Una de sus características, es acumular datos durante grandes periodos de tiempo. En contraposición a su alta velocidad en consulta, los cubos OLAP no son tan efectivos en acciones de eliminar, insertar y actualizar.

3.6.1 Cubos OLAP en Pentaho

Una vez configurado Mondrian, el procedimiento para crear un cubo es similar al del proceso ETL. Primero se debe generar una conexión de la base de datos sobre el servidor, en este caso localhost, para poder trabajar sobre las tablas y datos poblados. Crear un cubo es simple, ya que el modelo dimensional previamente elaborado permite construir el cubo de forma manual, es decir, unicamente indicando cual es la tabla de hecho y sus dimensiones.

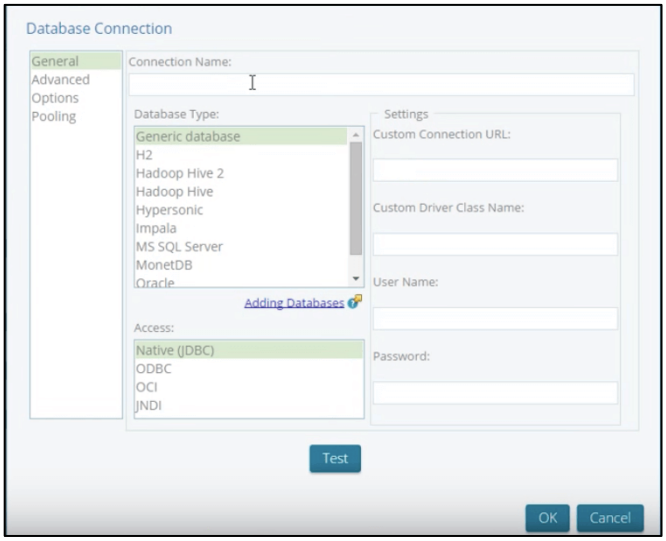


Figura 24. Conexión de base de datos en servidor de Pentaho

Desde Schema Workbench se construye un nuevo cubo, con un origen de datos previamente poblado. También es posible crearlo desde el servidor, pero su construcción no es tan explicita.

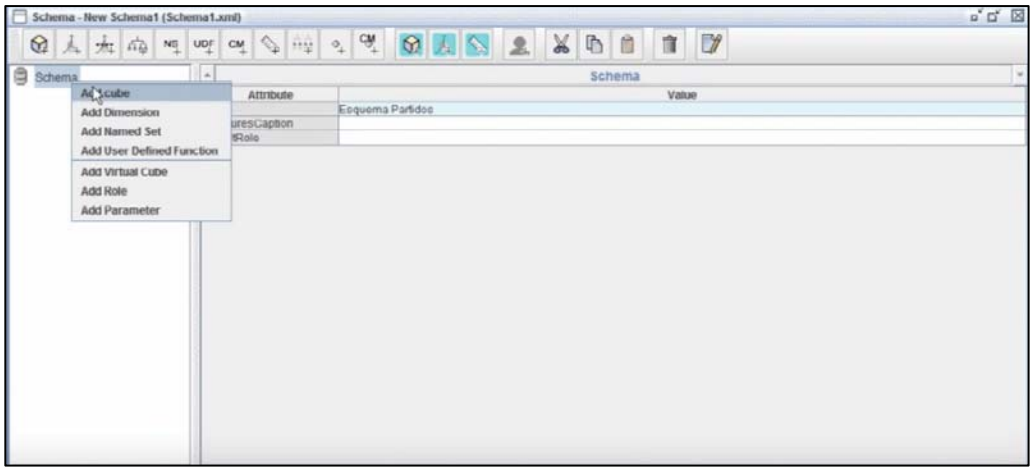


Figura 25. Creación de nuevo cubo

Como se muestra en la figura 25, en un esquema vacío podemos añadir un nuevo cubo, y posteriormente añadir dimensiones, tablas, métricas entre otras propiedades. En la figura 26 se muestra en el lado izquierdo la vista previa de los elementos que componen el cubo, y en el lado derecho un panel para personalizar cada uno de estos. El resultado final sera un archivo xml que puede ser modificado, para luego ser desplegado en el servidor.

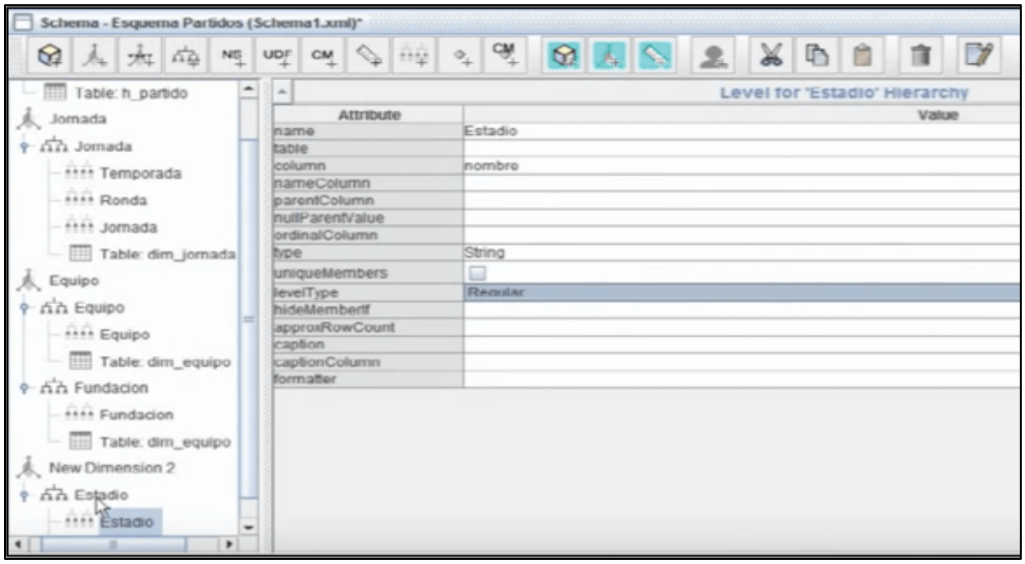


Figura 26. Construcción de cubo

Un detalle importante a considerar, es que es posible modificar el xml para filtrar datos. Por ejemplo, restringir rangos de fecha, limitar cantidades o datos.

Como se muestra en la figura 27, una vez creado el cubo, podrá ser visualizado desde el servidor donde la información consultada será desplegada. En la tabla central, “rows” y “columns” refieren a los renglones y columnas de las tablas que se usarán, mientras que “measures” refiere al dato que queremos medir (o consultar). (Más detalles en el Anexo B, página 139)

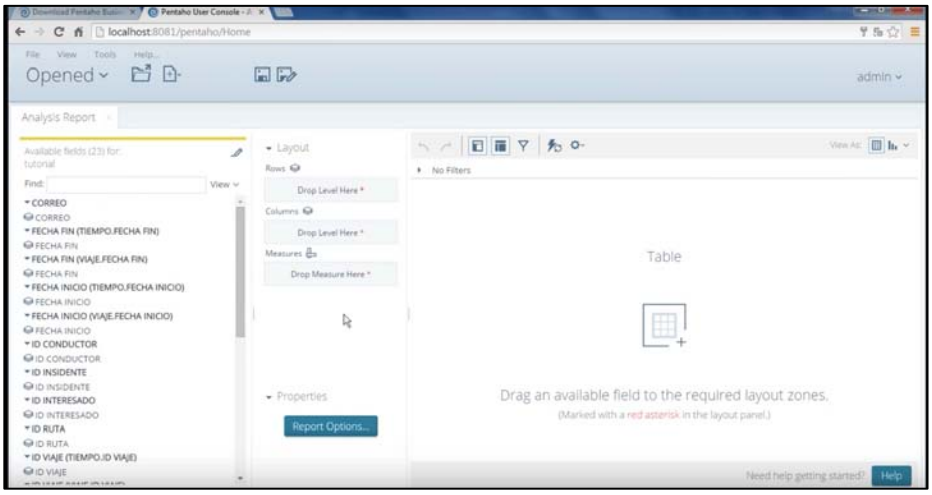


Figura 27. Base de datos cargada en el servidor

Crear una consulta consiste en arrastrar las columnas y renglones de la base de datos que queremos analizar, y posicionarlos en el panel central que le corresponda. También es posible añadir filtros o etiquetas en la sección del panel derecho. Como se muestra en la figura 28, 29 y 30, los datos de la consulta son desplegados en el panel derecho mediante presentaciones diferentes.

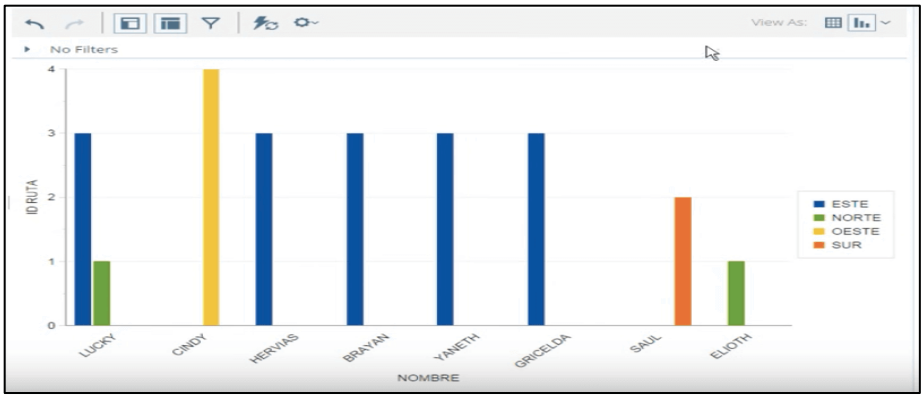


Figura 28. Visualización de datos en gráfica de barras

Figura 29. Visualización de datos en gráfica lineal

Pentaho ofrece gráficos simples pero funcionales. Aunque cuenta con pocas presentaciones predefinidas, permiten al usuario analizar el comportamiento de su información correctamente. Algunas de estas son la gráfica de barras (figura 28), gráfica lineal (figura 29) y la gráfica circular (figura 30).

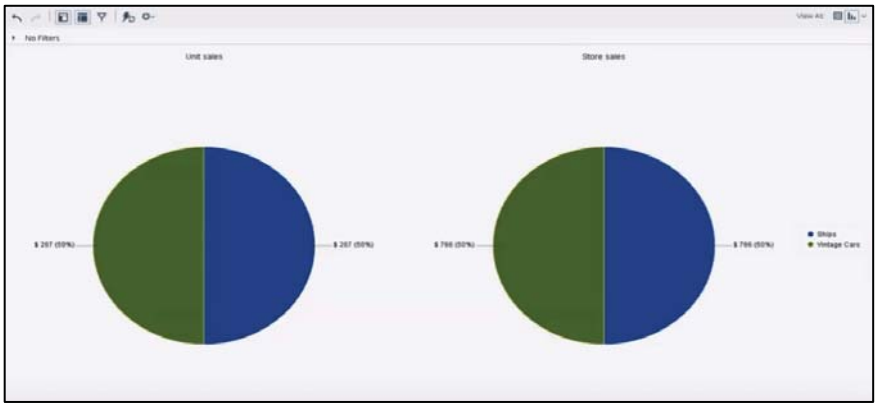


Figura 30. Visualización de datos en gráfica circular

3.6.2 Cubos OLAP en Oracle

Oracle requiere una conexión a Web Logic Server para utilizar el servidor. En este caso, el inicio de sesión no está basado en el usuario del motor de la base de datos de Oracle, sino en el utilizado en el middleware. A diferencia de Pentaho y SQL Server, Oracle requiere de Web Logic Server como un servicio intermedio entre su servidor y las bases de datos.

Figura 31. Conexión al servidor de Oracle

Para construir un cubo es necesario generar las dimensiones y el cubo por separado. Primero, se accede a Warehouse Builder para crear los nuevos elementos.

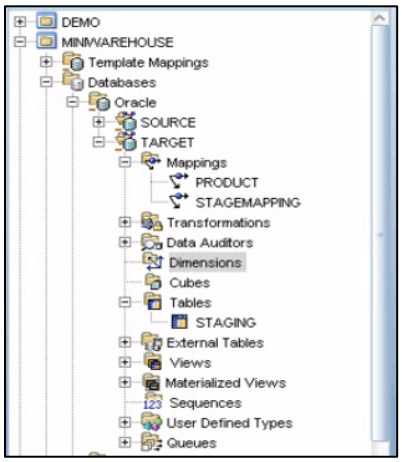


Figura 32. Carpetas en Warehouse Builder

En la izquierda del panel se muestran las carpetas (figura 32) en las que se generan los elementos a utilizar. Aquí hay una pequeña diferencia con respecto a Pentaho. Oracle genera dimensiones y cubos como si fueran elementos independientes. Primero, se añaden las dimensiones, y posteriormente “se arman” dentro del cubo. En la figura 35 se muestra la configuración del cubo, una vez que ya se han concluido sus propiedades. Posteriormente, se genera el archivo xml que se agregará al servidor.

Figura 33. Asistente de creación de dimensiones

Figura 34. Asistente de creación de cubos

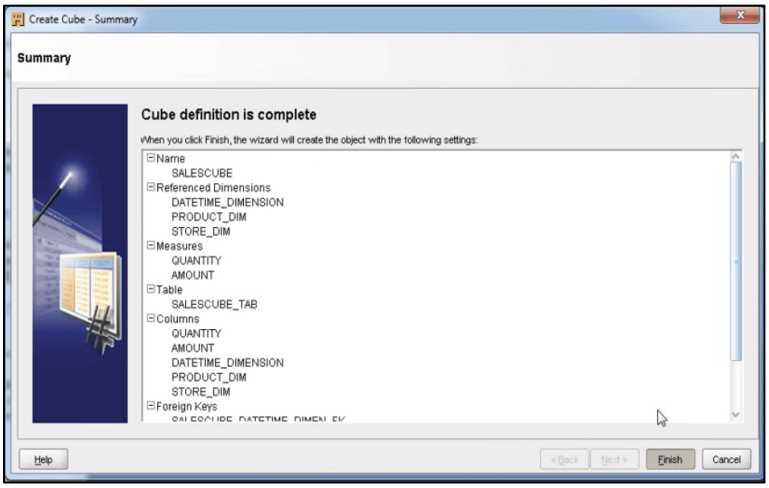


Figura 35. Configuración de cubo

A partir de que el usuario inicia sesión, se puede acceder a un esquema de base de datos, o acceder a un esquema de ejemplo ya cargado. También es posible crear un esquema desde la aplicación. Oracle y SQL Server proveen bases de datos de ejemplo con la intención de que los usuarios interactúen con las capacidades de sus productos. Al seleccionar una base, se despliegan las dimensiones y la tabla de hecho.

Figura 36. Selección de esquema de base de datos

Como se observa en la figura 37, el panel derecho muestra dos recuadros, la parte de arriba es de las columnas, y la parte de abajo es de los filtros.

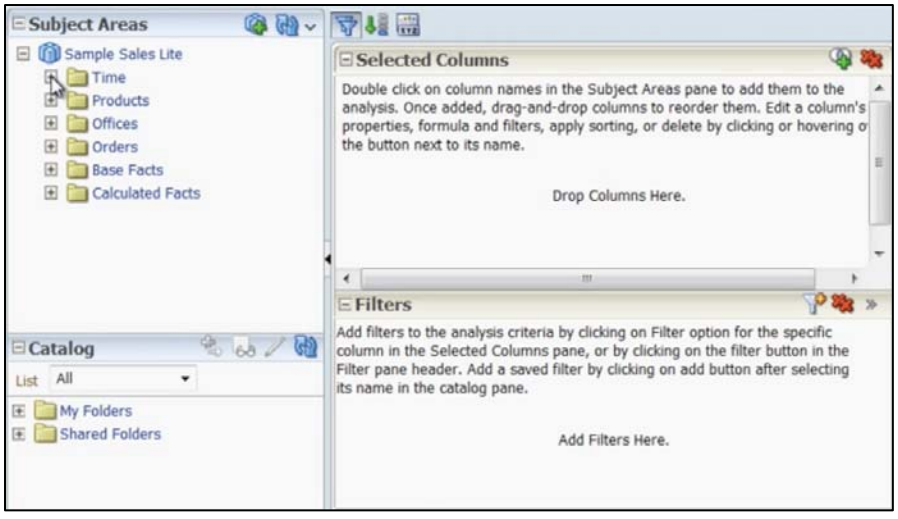


Figura 37. Cubo cargado en el servidor

Su mecanismo también es soltar y arrastrar las columnas que se desean consultar, y posteriormente agregar los filtros necesarios. Una salida de estos datos se muestra en la figura 38. (Más detalles en el Anexo B, página 146)

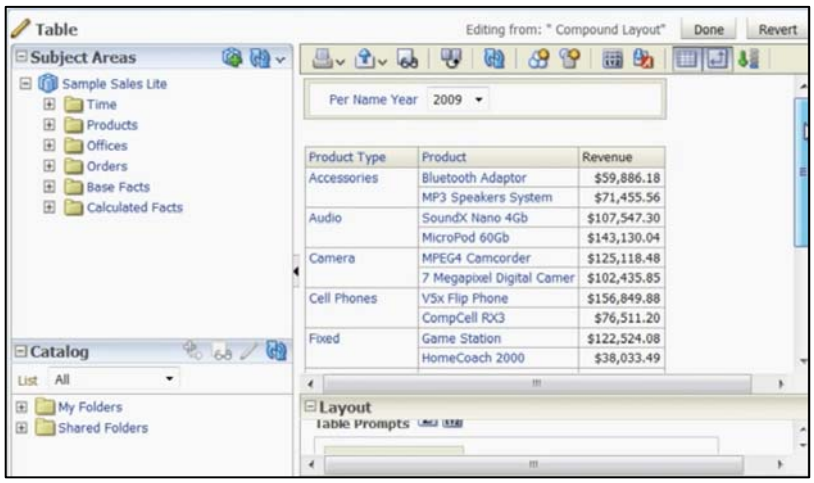


Figura 38. Despliegue de información

3.6.3 Cubos OLAP en SQL Server

Los cubos en SQL Server se diseñan con SQL Server Analysis Services. Esta herramienta se descarga en conjunto con el kit de SQL Server y posteriormente se añade dentro de Visual Studio.

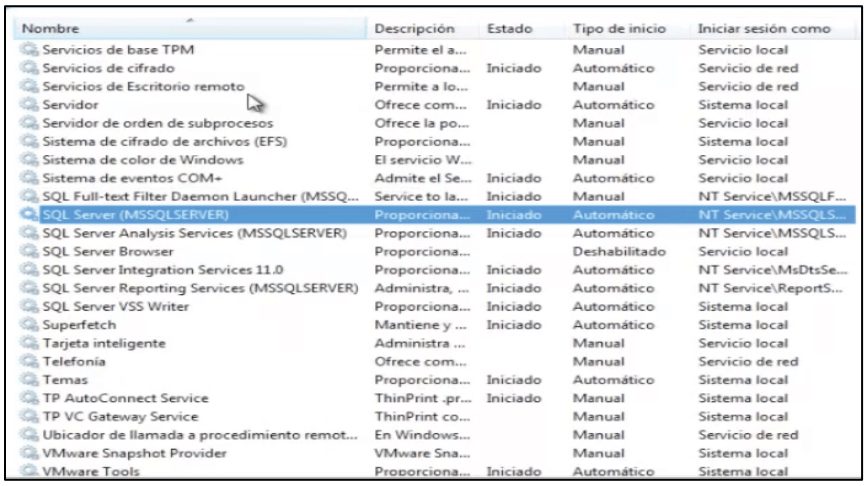


Figura 39. Servicios de Windows

Al momento de descargar el kit, el usuario obtendrá Visual Studio si previamente no lo tenía, aunque debe configurar manualmente las opciones de Analysis Services. Aquí debe resaltar el hecho de que, aunque esté instalado podría no funcionar si el servicio no está activo. Para resolver este detalle, el usuario debe dirigirse a los servicios de Windows y activar manualmente la opción de Analysis Services, como se muestra en la figura 39. Una vez activado, el usuario puede comenzar Visual Studio. En las opciones de nuevo proyecto, se debe elegir la opción de “Proyecto multidimensional y de minería de datos” para accesar a las opciones de OLAP.

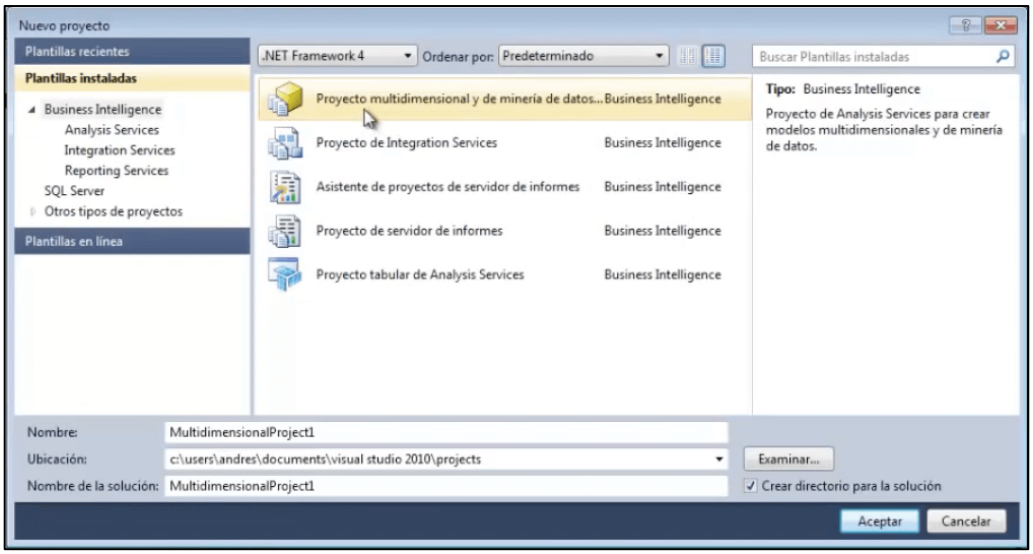


Figura 40. Selección de proyecto

Posteriormente, se le pedirá al usuario un origen de datos, un caso parecido al del proceso ETL donde se debe ingresar la información de la base. Luego un asistente para la creación de tablas ayudará al usuario a seleccionar tablas y vistas, dando un resultado como se muestra en la figura 41. Este proceso únicamente almacena la información como tablas. Para llevar a cabo un análisis, es necesario acceder al asistente para cubos, y llevar a cabo un proceso similar al del asistente de tablas. Luego de seleccionar las tablas, las medidas y ajustar las dimensiones necesarias, se procede a construir un cubo, dando como resultado la muestra de la figura 42. (Más detalles en el Anexo B, página 156)

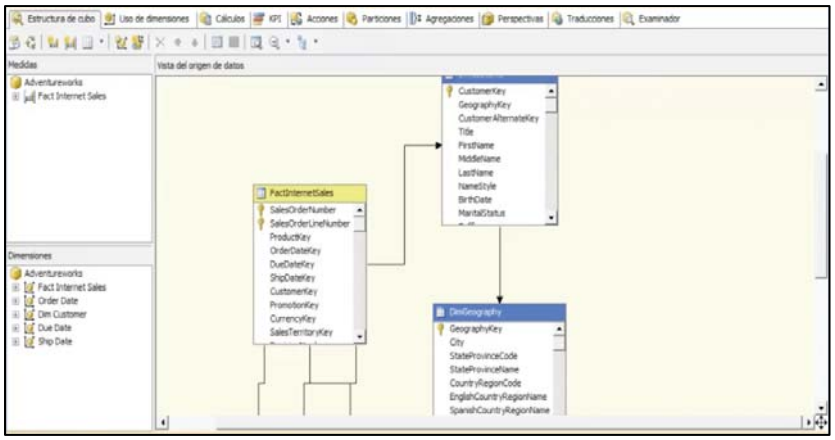


Figura 41. Construcción de tablas en SQL Server

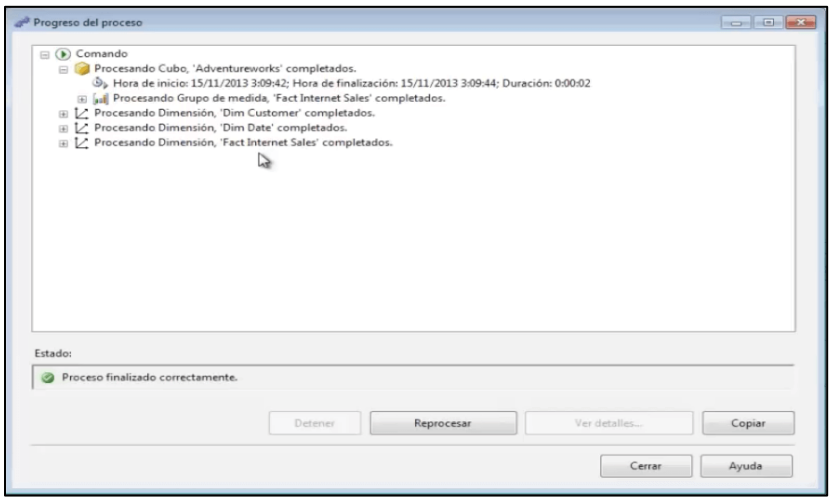


Figura 42. Construcción de cubo en SQL Server

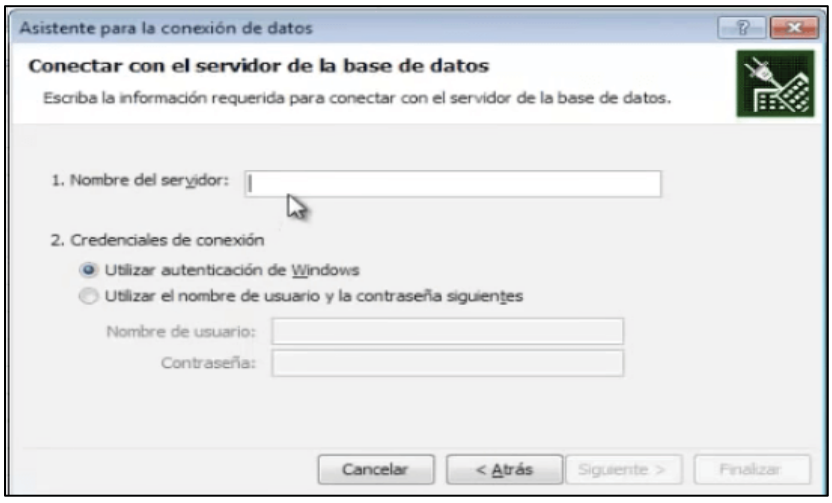


Figura 43. Conexión de base de datos en Excel

Una opción para visualizar datos desde el cubo es Excel. Se debe seleccionar la opción “De otras fuentes” para poder iniciar el asistente de conexión de datos, como se muestra en la figura 43. Nuevamente, se ingresan los datos del servidor. Una vez ingresados los datos correspondientes, es posible utilizar las dimensiones creadas desde el panel derecho, para importar los datos hacia el archivo excel, usando formatos de texto o gráficos, como se muestran en la figura 44.

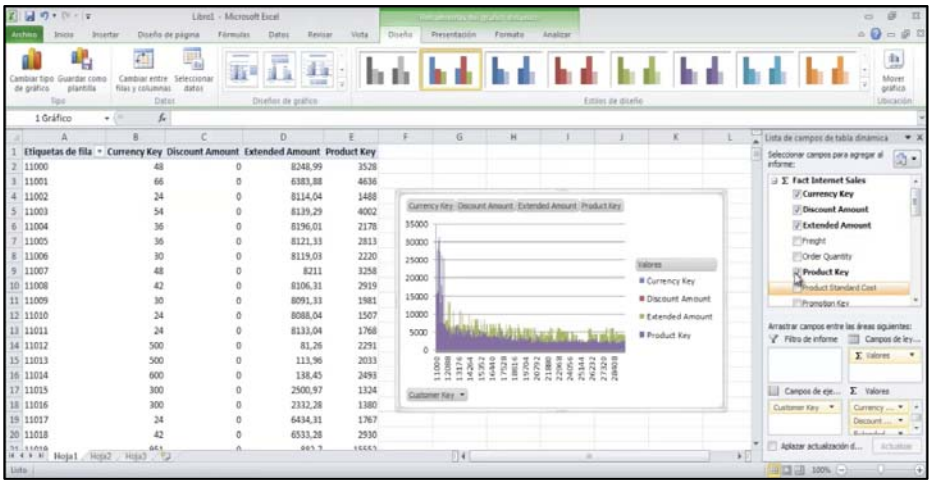


Figura 44. Análisis de datos desde el cubo

3.7 Creación de reportes

Aunque los datos obtenidos en el proceso OLAP retribuyen información de negocio de alto valor, estos datos deben ser mostrados a usuarios que no podrían no estar involucrados en el área de TI. Para resolver esto se crean reportes, información integrada de las consultas realizadas en OLAP para ser mostradas de un modo más legible para los usuarios.

3.7.1 Reportes en Pentaho

Los reportes en Pentaho se realizan con Report Designer. Esta herramienta no está incluida en el servidor de Mondrian, por lo que debe ser descargada de manera independiente. El modo de ejecución dependerá de la plataforma en la que sea descargado. En Mac se generá un icono de acceso directo.

Debe ser ejecutado desde la terminal por medio del archivo .sh si se ejecuta en Linux, si se ejecuta en Windows sería un archivo .bat. Al ejecutarse, el usuario podrá elegir crear un reporte con o sin ayuda del asistente.

Figura 45. Ejecución de Pentaho Report Designer

Al seleccionar “New report” nos despliega un documento dividido en secciones. Las secciones apoyan al usuario a dar un formato más limpio al momento de ordenar sus consultas (como un archivo de Word, divide encabezados, detalles y pies de página), solo son sugerentes para el usuario ya que los elementos se acomodan a sus propias preferencias.

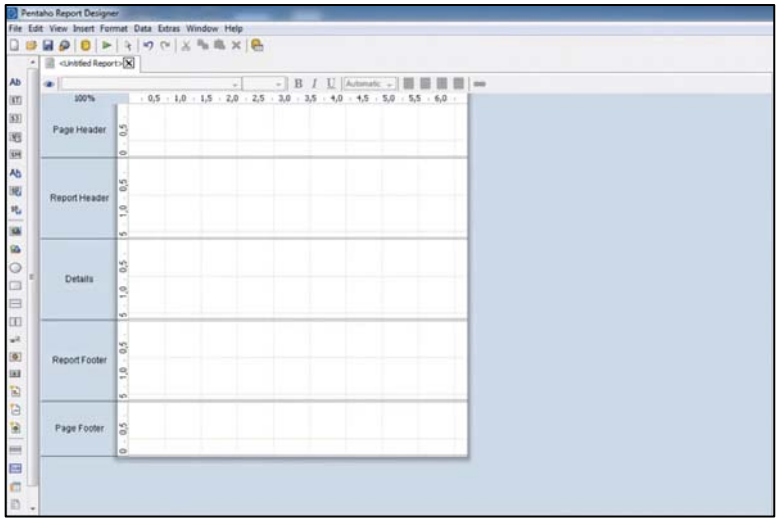


Figura 46. Documento en blanco

En el lado izquierdo, hay un panel con opciones de arrastrar y soltar que permiten añadir elementos como etiquetas o imágenes. En el lado derecho se encuentra un panel de selección de fuentes de datos. Si el usuario desea extraer datos desde una base, debe ingresar los datos de su conexión.

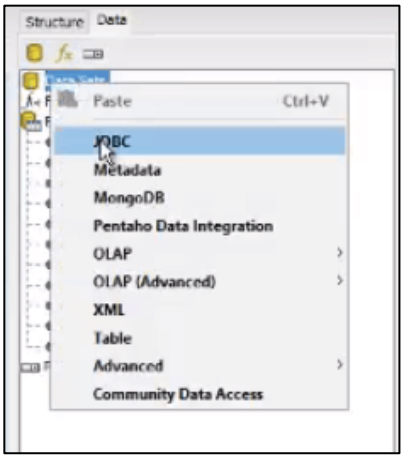


Figura 47. Selección de conjunto de datos

Una vez que se ha realizado la conexión, el usuario puede construir manualmente las consultas que arrastrará a su reporte. Es decir, el usuario ingresará manualmente una consulta y acumulará esa función en una etiqueta que utilizará para arrastrar y soltar en algún espacio del reporte.

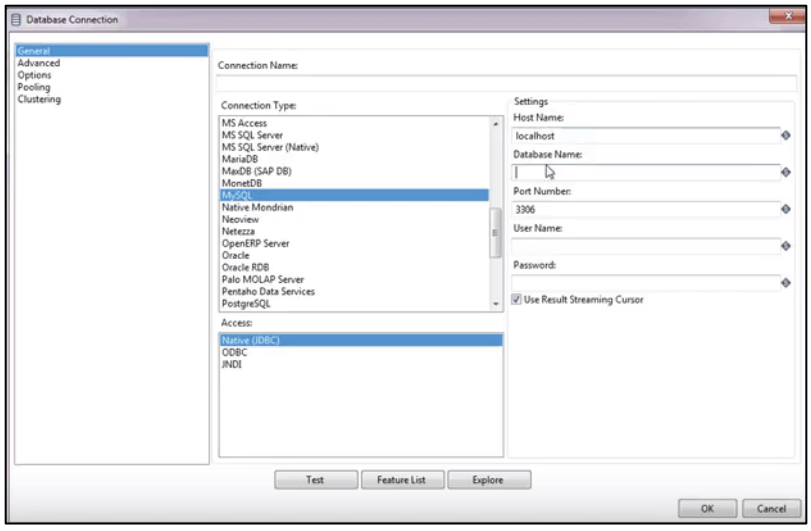


Figura 48. Selección de conexión a una base de datos

Como se muestra en la figura 49, la consulta será creada. En el panel derecho estará disponible para arrastrarse al documento y desplegar el resultado de esta consulta.

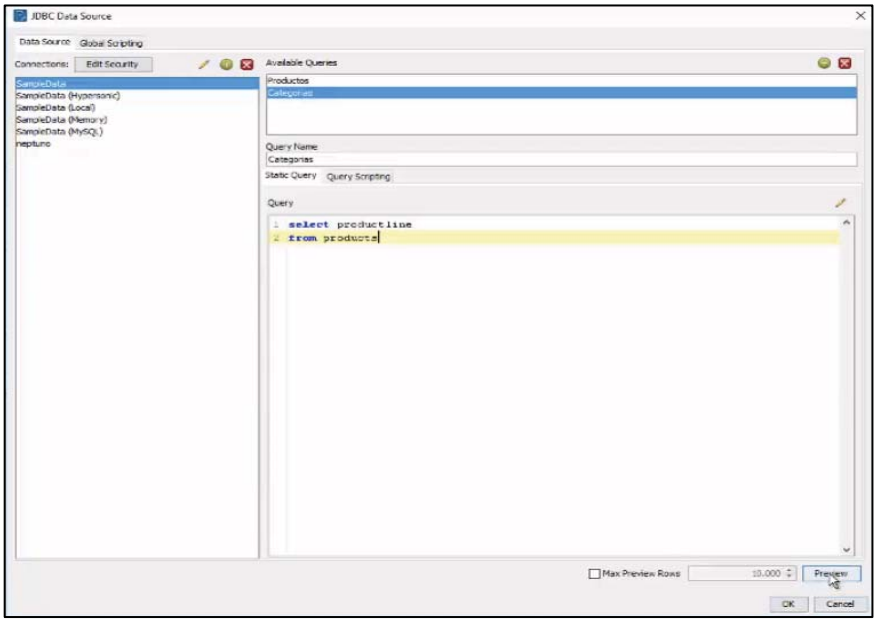


Figura 49. Construcción de consulta

Sobre el lenguaje MDX, en el capítulo 2 se mencionó sobre su definición. En la práctica, es necesario mencionar los detalles más relevantes sobre este lenguaje. MDX está diseñado para navegar por las bases de datos multidimensionales y definir consultas en todos sus objetos (dimensiones, jerarquías, niveles, miembros y celdas) para obtener (simplemente) una representación de tablas dinámicas. MDX usa muchas palabras claves parecidas a las de SQL, como SELECT, FROM, WHERE. La diferencia es que SQL produce vistas relacionales mientras que MDX produce vistas multidimensionales de datos. La diferencia también se ve en la estructura general de los dos lenguajes:

a) Consulta SQL: SELECT columna1, columna2, ..., columnan FROM tabla
b) Consulta MDX: SELECT eje1 ON COLUMNS, eje2 ON ROWS FROM cubo

Este lenguaje es aplicado para realizar consultas sobre cubos multidimensionales.

Cuando el usuario selecciona las dimensiones desde la interfaz, la aplicación construye automáticamente la consulta que está por ejecutarse. Las consultas MDX pueden ser utilizadas para generar reportes dinámicos. En la sección “Data” (figura 47) existe una sección donde se añaden parámetros. Aquí es donde el usuario ingresa sus consultas MDX para calcular valores desde el servidor. El tipo de reporte que se ofrece por omisión es de tipo lienzo. Su estructura permite acomodar los elementos libremente. Si el usuario requiere un tipo más formal, se ofrecen los tipos de reporte de bloque, en fila y en línea.

Sobre la presentación de los datos, con respecto al texto, Pentaho ofrece un panel donde se pueden cambiar las fuentes y los estilos. No es muy diferente a los estilos de CSS. Con respecto a las gráficas se ofrece otro panel con una cantidad de opciones predefinidas bastante útiles. En la siguiente figura se muestran los tipos de gráficas que se pueden utilizar, así como sus opciones configurables. En la parte de arriba estan los tipos de gráficas más conocidos (lineal, barra, circular, etc). Son pocas opciones pero son útiles. En la parte de abajo se encuentran dos paneles. Cuando un usuario selecciona, por ejemplo, una gráfica de barras, en los paneles de abajo se modifican las opciones pertenecientes a cada tipo de gráfica, por ejemplo, barras, columnas, textos, numero de círculos, etc.

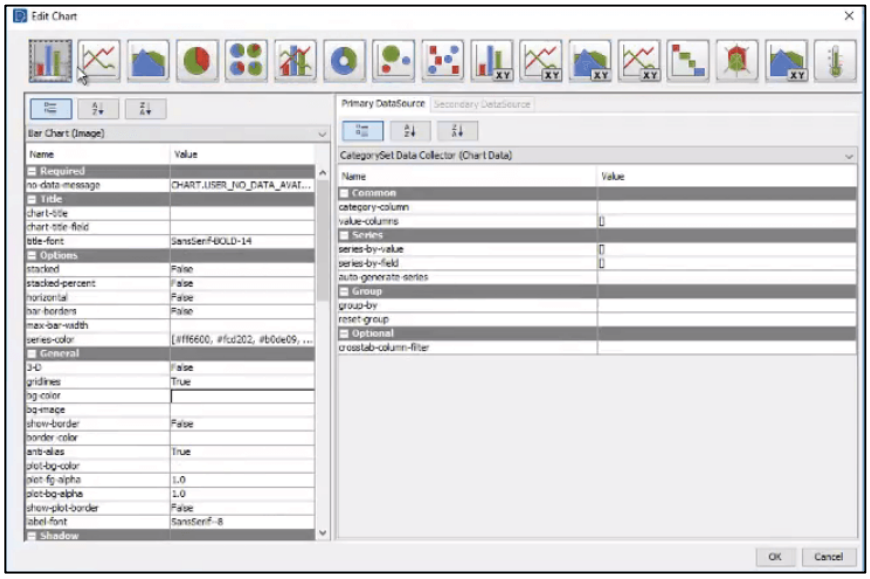


Figura 50. Panel de configuración de gráficas

Para finalizar, una vez que se ha poblado el documento con consultas o elementos, se procede a ejecutar la visualización del documento, que posteriormente podrá ser impreso o exportado en diferentes formatos. (Más detalles en el Anexo C, página 166)

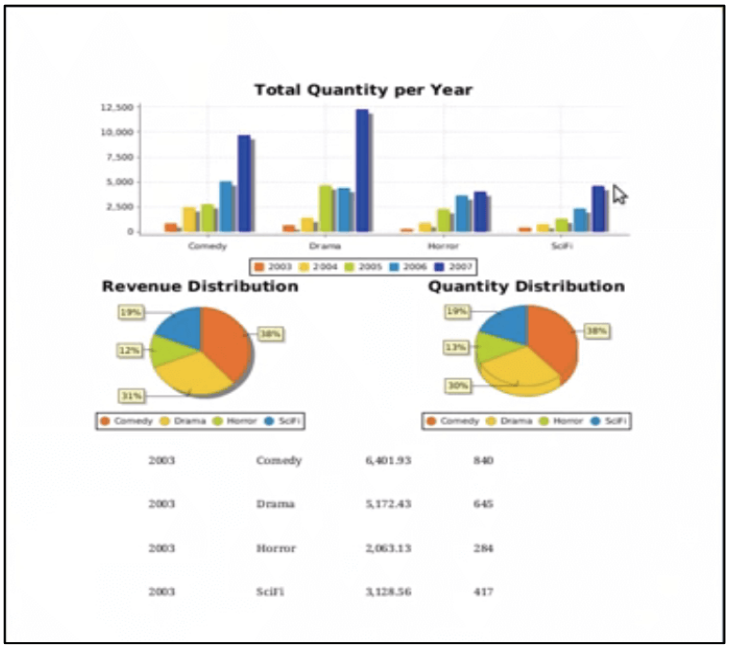


Figura 51. Reporte concluido

Una vez que el reporte ha sido concluido, el usuario puede publicar su reporte para ser visualizado desde el servidor. Para publicarlo es necesario apuntar a la dirección en la que será alojado (en este caso, localhost). Cuando los reportes ya se encuentran en el servidor, Pentaho ofrece la exportación en los siguientes formatos:

- PDF
- HTML
- Excel
- CSV
- RTF
- XML
- Texto plano

3.7.2 Reportes en Oracle

Oracle diseña sus reportes desde “Published Reporting”, seleccionando la opción “Report”, como se muestra en el panel izquierdo de la figura 52.

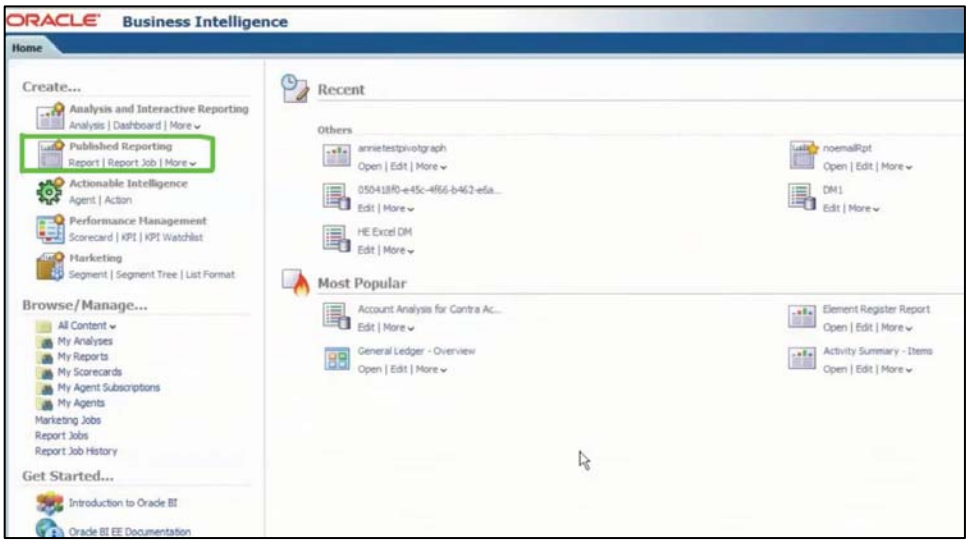


Figura 52. Interfaz de selección de reportes en Oracle

Posteriormente el usuario debe añadir las propiedades de su reporte, y en el lado izquierdo, debe seleccionar un conjunto de datos de donde provendrá la información de sus reportes.

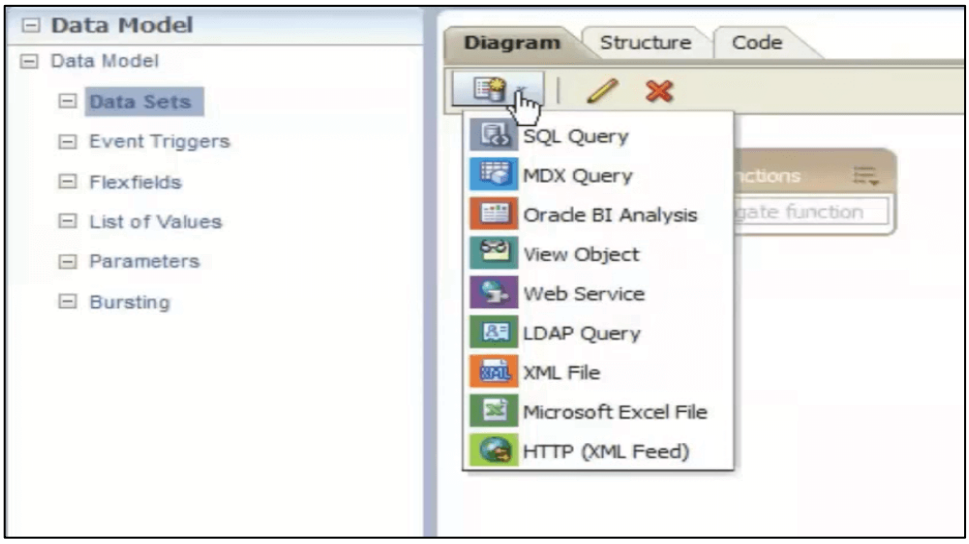


Figura 53. Selección de conjunto de datos

Una vez seleccionada una fuente de datos, al igual que en Pentaho, el usuario puede generar una consulta para añadir al reporte. Después se agregará la consulta (en sql) sobre el reporte, aunque no como una etiqueta como en Pentaho, se almacena como un elemento con opciones de configuración.

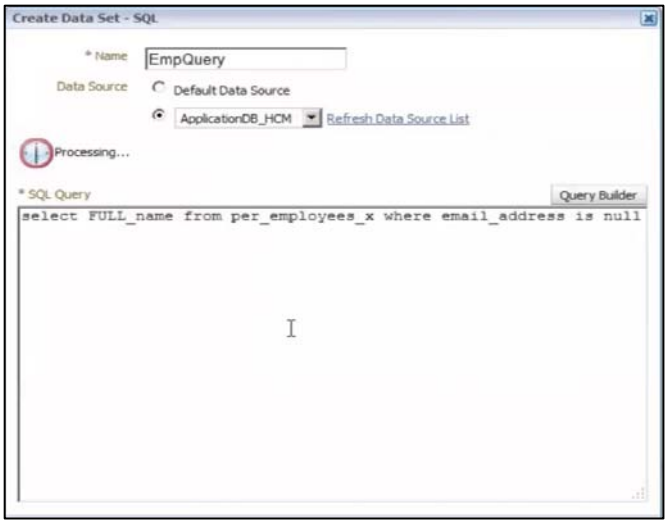


Figura 54. Creación de consulta

Al visualizar el documento, se pueden agregar etiquetas y elementos de una forma muy similar a la de Pentaho, aunque en Oracle la herramienta no es tan intuitiva, ya que el diseño en su panel de opciones es más parecido a un editor de textos (se parece un poco a la interfaz de un documento de Word).

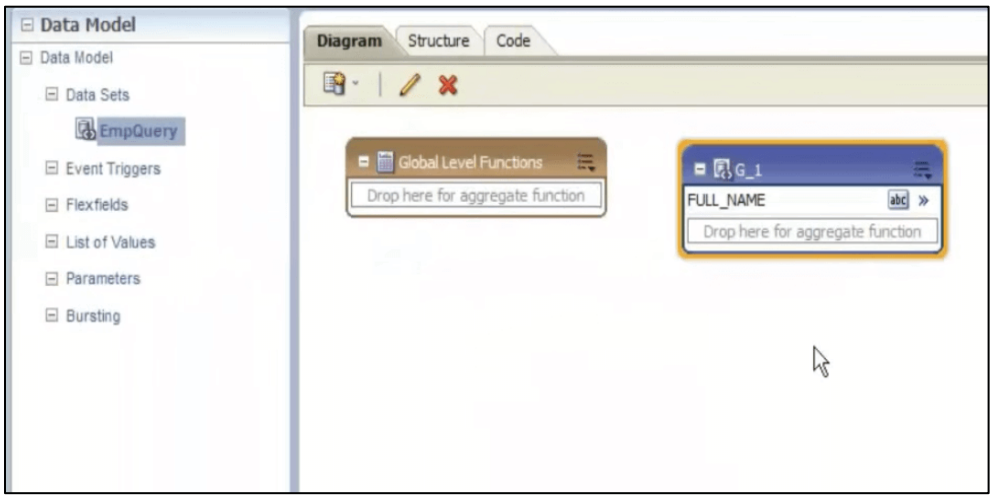


Figura 55. Consulta añadida al reporte

Cuando una etiqueta es creada, la estructura permite modificar los datos que se asignan. En la siguiente figura se muestra un ejemplo de la configuración de la consulta que se guarda en una etiqueta.

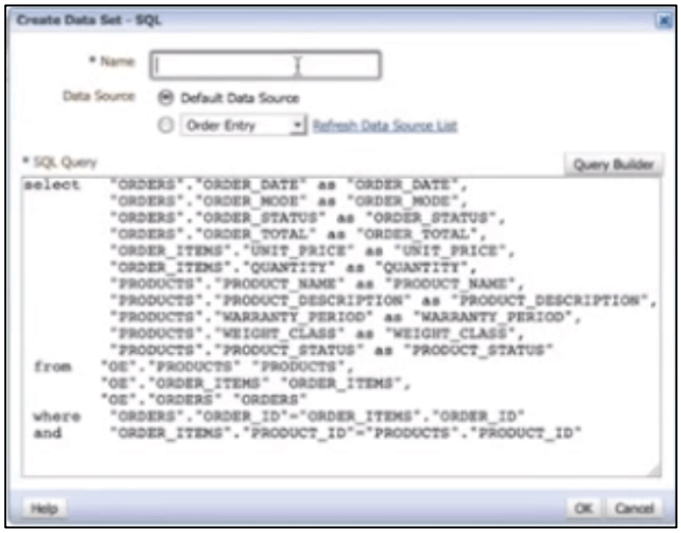


Figura 56. Configuración de consulta

Al igual que Pentaho, cuenta con un panel de configuración para los elementos que se van a desplegar. En la figura 57 se muestra el panel de propiedades de los elementos. Al igual que Pentaho, también incluye propiedades parecidas a las utilizadas en elementos de diseño gráfico, como los tipos de fuente, colores, margenes, bordes, etc.

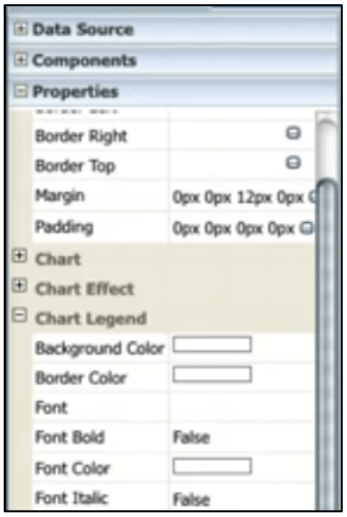


Figura 57. Configuración de elementos

En la figura 58 se muestran los elementos disponibles para la visualización. A diferencia de Pentaho, Oracle posee un gran repertorio de presentaciones visuales. Las gráficas no se configuran tanto como en Pentaho, pero en Oracle hay otras opciones, como las gráficas 3d. Para cada tipo de gráfica, Oracle presenta algunas variantes predeterminadas de cada una. También tiene una vista previa que permite mostrar cómo se verían los datos en el servidor antes de publicarlos. (Más detalles en el Anexo C, página 170)

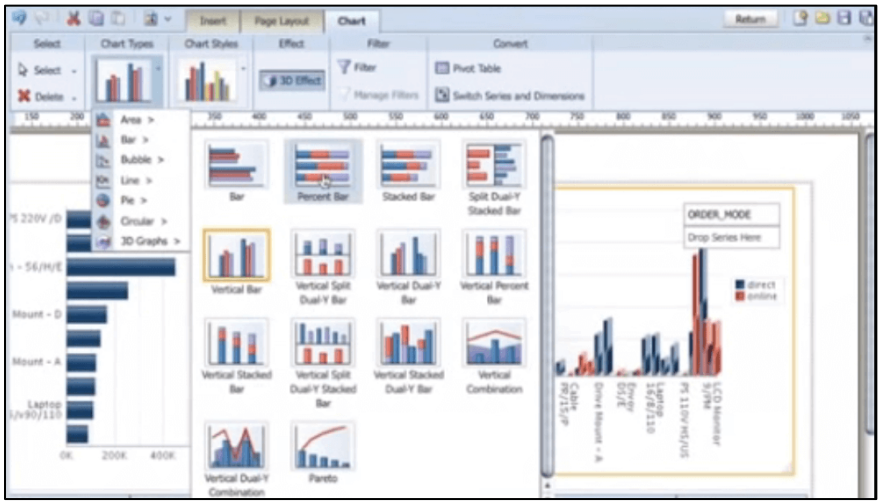


Figura 58. Panel de gráficas

Con respecto a la parte de textos, la mejora se ve en la posibilidad de editar opciones. También es posible crear reportes dinámicos creando variables que calculan información desde el servidor. Una vez que se publica un reporte dentro del servidor, es posible exportarlo a los siguientes formatos:

- Interactive
- HTML
- PDF
- RTF
- Excel
- Powerpoint
- Flash
- CSV
- XML
- MHTML
- PDFZ
- FO

Figura 59. Vista previa del reporte

3.7.3 Reportes en SQL Server

Reporting Services es un servicio que también es utilizado desde Visual Studio. Al igual que Analysis Services, el usuario debe asegurarse de que la herramienta está instalada y activar en los servicios de Windows. Al momento de desplegar Reporting Services, el usuario puede elegir, al igual que Pentaho, si desea ayuda del asistente en la creación del reporte (figura 60). Luego de seleccionar una opción, el usuario ingresa los datos de su conexión (figura 61).

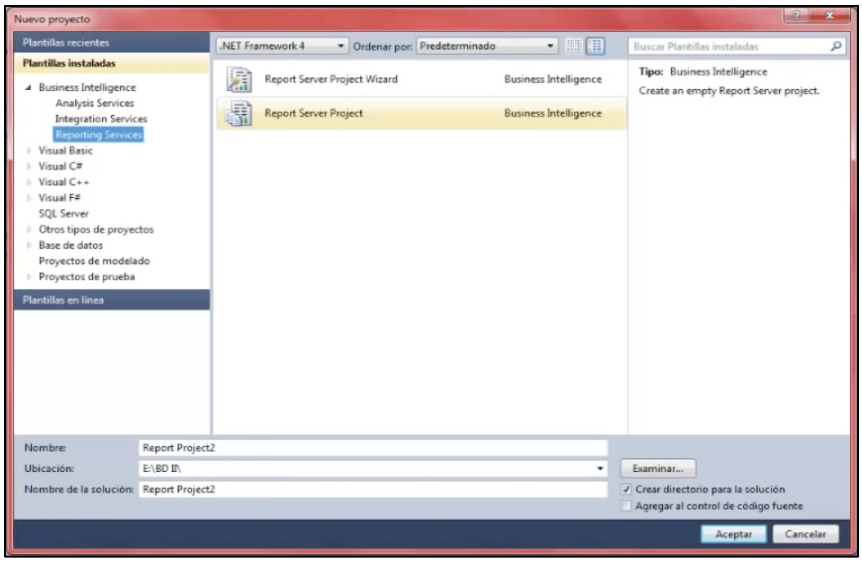


Figura 60. Creación de nuevo reporte

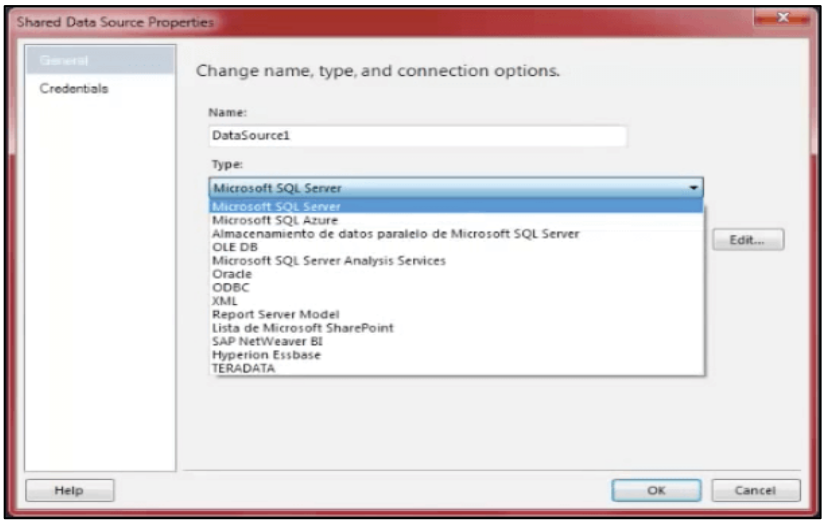


Figura 61. Conexión de base de datos

La creación del reporte se facilita gracias a que el asistente identifica las tablas y el usuario puede elegir sobre que campos se pedirá información. El asistente permite realizar consultas de manera automatizada para facilitar los reportes a los usuarios. La principal ventaja se encuentra en el tiempo de creación, gracias al asistente es posible diseñar un reporte simple pero eficiente. La desventaja es que al utilizar el asistente, ofrece pocas opciones a los usuarios. En este sentido, Oracle destaca por el hecho de ofrecer mayor libertad al momento de darle un formato a un reporte.

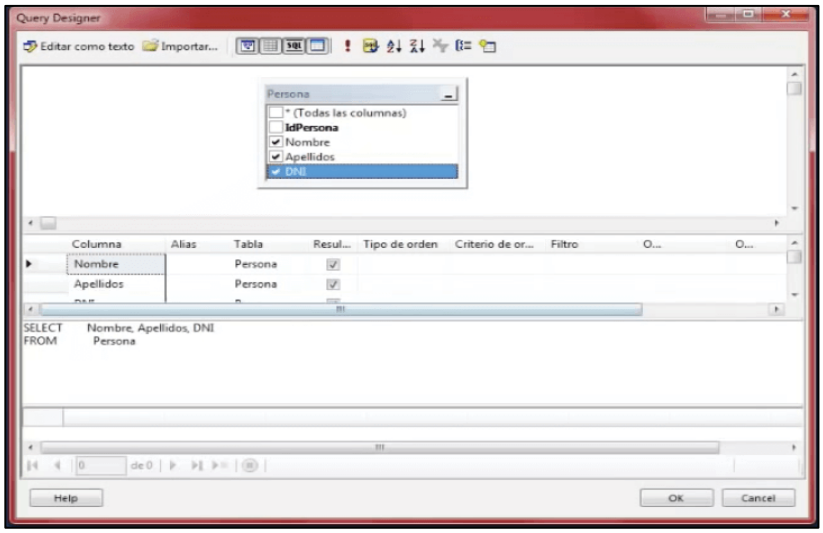


Figura 62. Creación de consulta

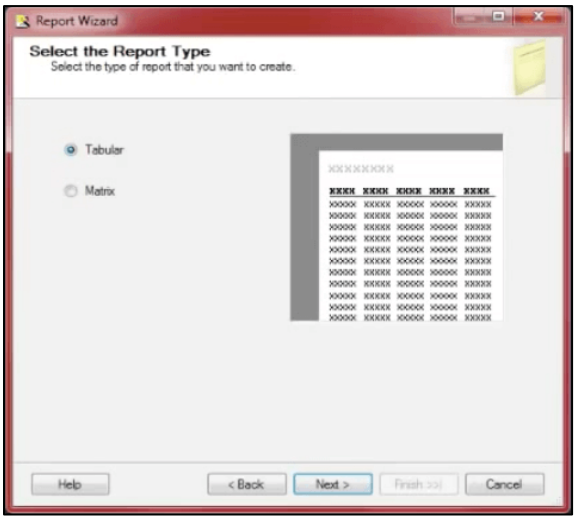


Figura 63. Tipo de reporte

A diferencia de Pentaho y Oracle, en Reporting Services el usuario puede seleccionar que formato quiere para el despliegue de su información. Puede elegir entre diseño tabular o diseño de matriz (figura 63). Luego el usuario puede acomodar en qué orden sus datos serán agrupados en el tipo de reporte que haya seleccionado.

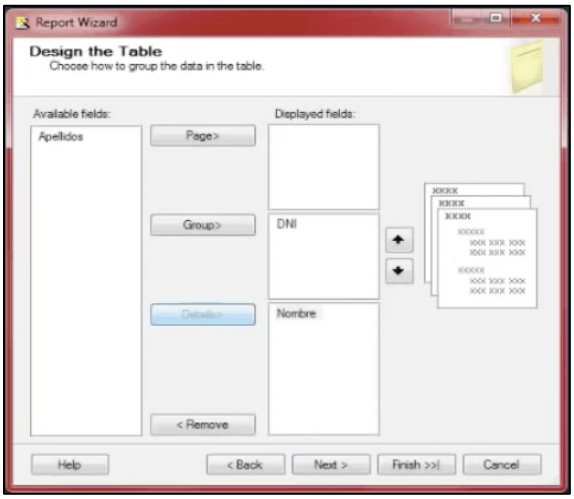


Figura 64. Diseño de tabla

Al observar la vista previa del reporte, se puede mostrar que la tabla se ha construido con los datos del usuario. La creación es más eficiente, ya que el usuario tiene la facilidad de crear un reporte rápido y no requiere tanto conocimiento de la herramienta.

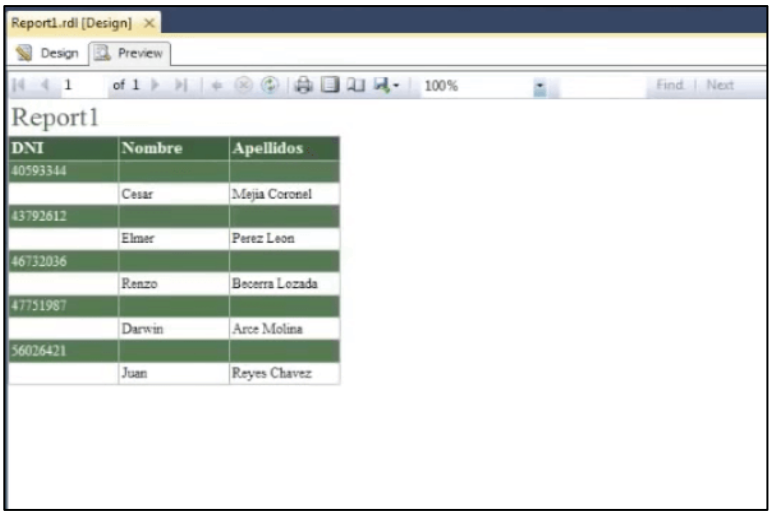


Figura 65. Reporte con texto

En cuanto a la presentación de los datos, se ofrece un buen contenido gráfico para el usuario. Además de una gran variedad de elementos, también es posible modificar las etiquetas de las gráficas. (Más detalles en el Anexo C, página 175)

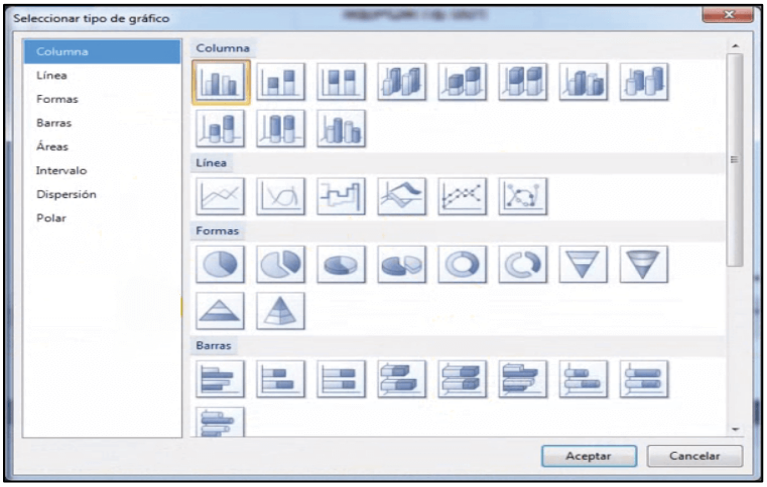


Figura 66. Opciones de gráfico

Una vez concluido el reporte, es posible publicarlo mediante el administrador de publicación. Aquí es importante mencionar que solo SQL Server realiza esta función apoyado por un asistente. Oracle y Pentaho lo incluyen como una característica más dentro de sus plataformas mientras que aquí hay una ventana exclusiva para la configuración del servidor donde se desplegará el reporte.

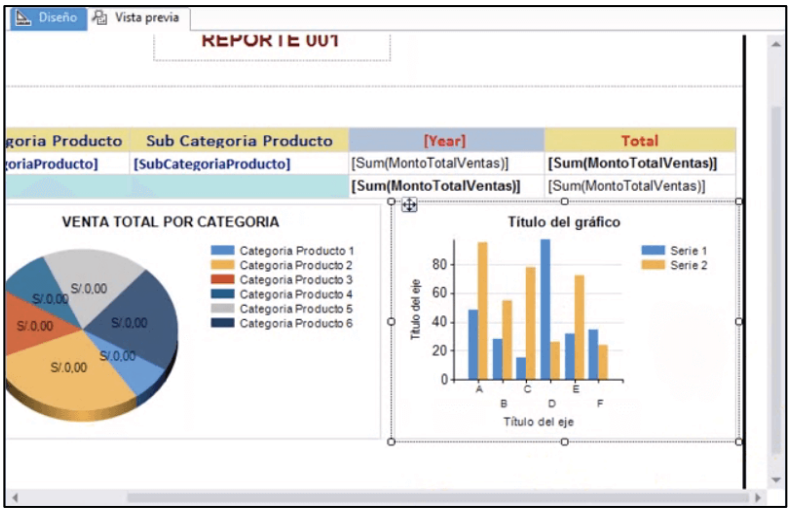


Figura 67. Reporte con gráficos

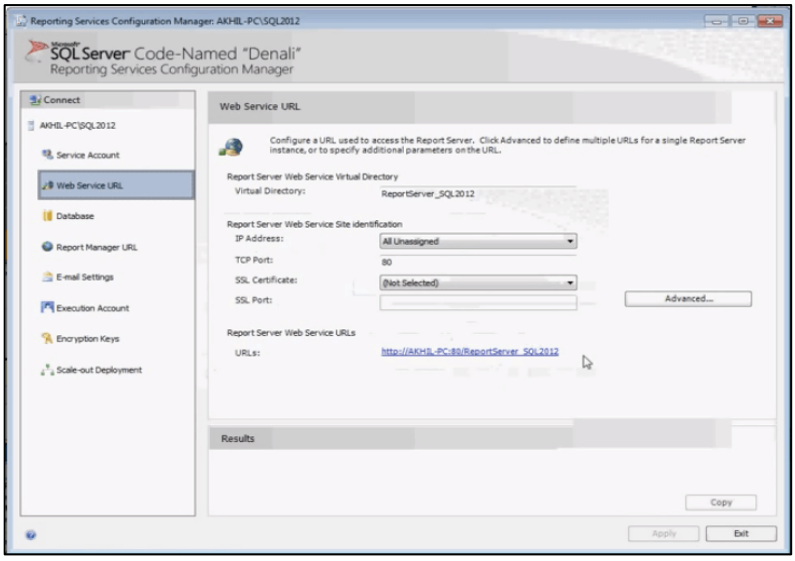


Figura 68. Publicación desde el administrador

Finalmente, los formatos de exportacion que se ofrecen por parte de SQL Server son los siguientes:

- XML
- CSV
- MHTML
- PDF
- TIFF
- Excel
- Powerpoint
- Word

3.8 Tableros de mando

La parte más importante de realizar un proceso de inteligencia de negocios, es poder convertir los datos en información de valor para una empresa. Mostrar indicadores clave a usuarios que toman decisiones apoya el conocimiento sobre el comportamiento de un área o una empresa. Aun cuando los reportes son efectivos para detallar algún indicador, lo ideal es un tablero de mando, para observar rápidamente los aspectos importantes sobre los que se debe aplicar alguna medida.

3.8.1 Tableros en Pentaho

Los tableros de mando en Pentaho pueden realizarse en diferentes formas según la versión. En la versión de comunidad se utiliza Community Dashboard Editor. Para desplegarlo, el usuario debe seleccionar la opción desde el servidor de Pentaho (figura 69). Posteriormente, se debe seleccionar la fuente de datos. En este caso, el menú de opciones incluye la selección de lenguajes para agilizar la información cuando ya se añadió una fuente de datos previamente (figura 70).

Figura 69. Selección de CDE desde servidor de pentaho

Una vez que se ha seleccionado una fuente, se debe de proporcionar una consulta por cada elemento gráfico a mostrar. Aquí es similar a los reportes, donde cada

vez que se crea un despliegue de datos, se debe crear una consulta. En los tableros de mando lo que predominan son los elementos gráficos. El objetivo de un tablero es mostrar relevante que pueda ser interpretada en un tiempo breve, a diferencia de los reportes, que dan información para ser analizada con detalle.

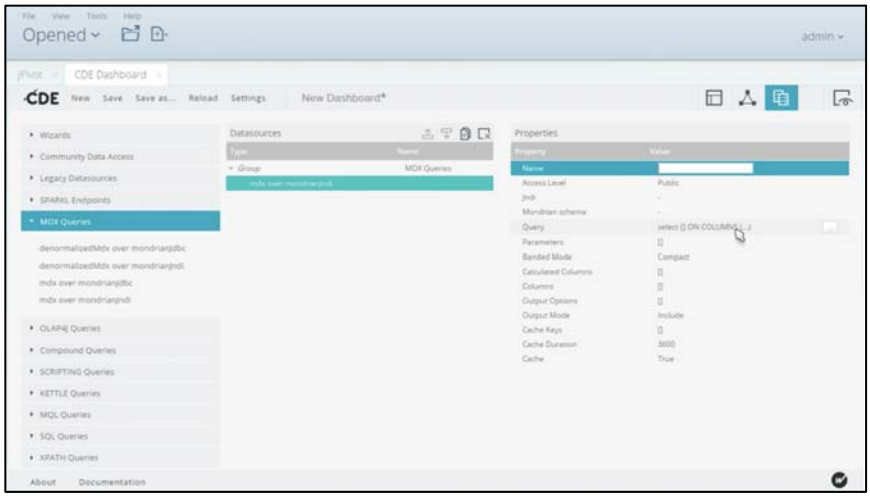


Figura 70. Selección de fuente de datos

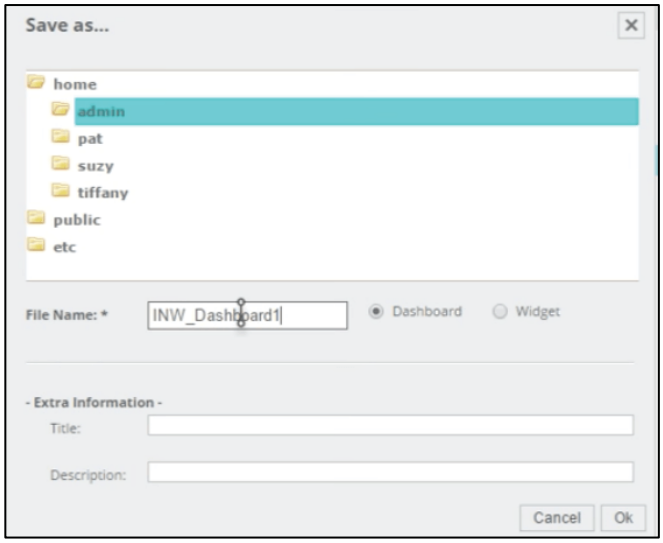


Figura 71. Almacenamiento de consulta

Los tableros de mando en Pentaho requieren algunas habilidades de diseño para proporcionar estilos en los gráficos. Como se muestra en la figura 72, el panel izquierdo permite ajustar el *layout* donde estará el contenido de la consulta, y en el

lado derecho, opciones para darle diseño a la salida de datos. En cuanto a la presentación de los datos, esta herramienta no es tan intuitiva como las anteriores herramientas de Pentaho.

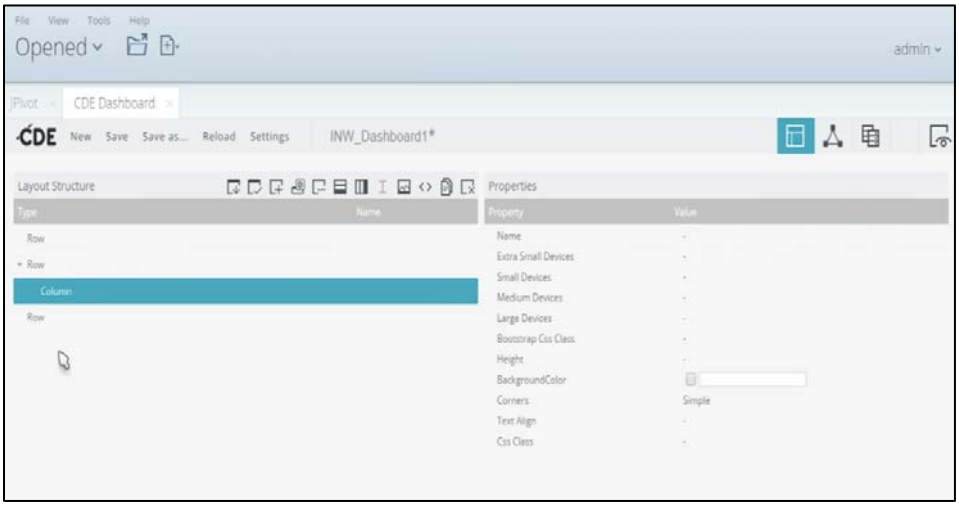


Figura 72. Panel de diseño del tablero

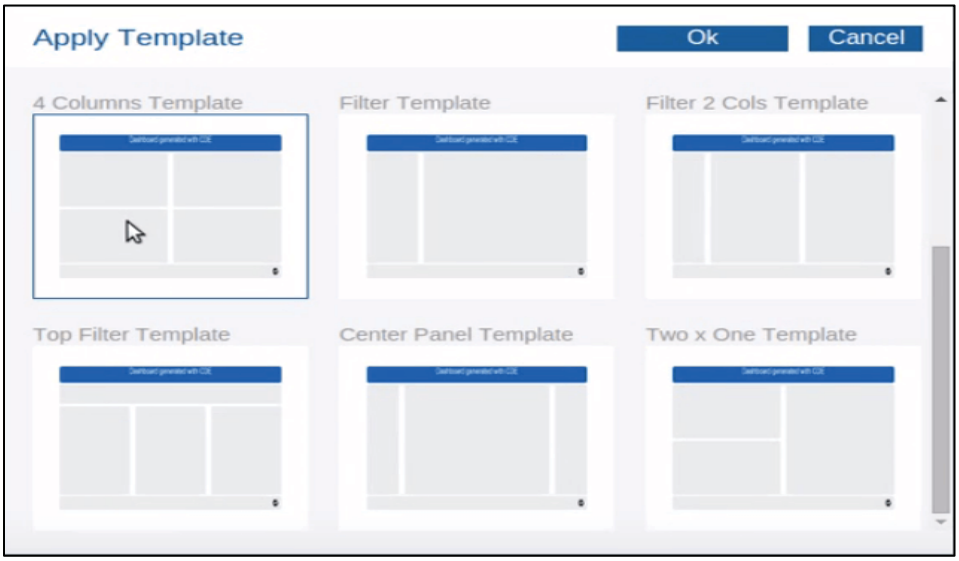


Figura 73. Formatos de tablero

Como se muestra en la figura anterior, hay diferentes formas de construir un tablero en CDE. Aunque la mecánica de construcción de consultas es la misma que en los reportes, aquí el aspecto visual es lo que juega el papel primordial. Para la construcción de gráficos se cuenta con un conjunto más grande de

componentes, pero debido a la falta de iconos, no es tan simple conocer el efecto de cada una de las gráficas disponibles (figura 74).

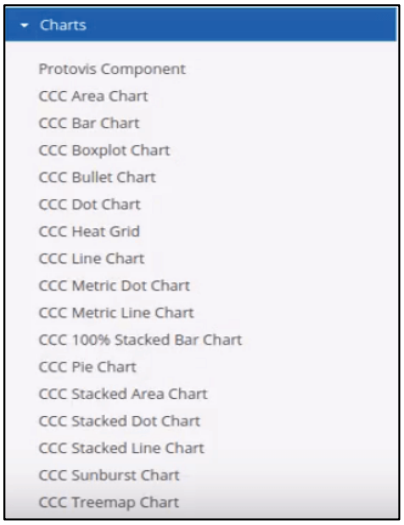


Figura 74. Presentaciones gráficas disponibles

Como una característica de CDE se puede mencionar que es posible añadir diferentes tipos de consulta, como MDX o SQL, para desplegarse sobre los tableros. Sin embargo algunas de sus funciones se encuentran muy escondidas, lo cual provoca muchas desventajas. Para poder usar mejor esta herramienta es necesario conocer a detalle donde se encuentran los componentes dentro de su interfaz. (Más detalles en el Anexo D, página 185)

Con respecto a desplegar la información, es posible ingresar dimensiones, medidas y filtros, justo como ya se han utilizado en otras herramientas. La diferencia es que en los tableros se pueden adaptar elementos de programación web (html, css, js, etc) para poder obtener datos de manera responsiva.

Estos elementos se guardan como tipo “Resource”, en los que se añaden los códigos correspondientes, como si fueran contenedores, en los que se aplican diseños o funciones específicas a las consultas. De este modo se pueden realizar consultas con parámetros que pueden hacer cálculos desde el servidor. Los datos también pueden actualizarse imitando el enfoque de “tiempo real”.

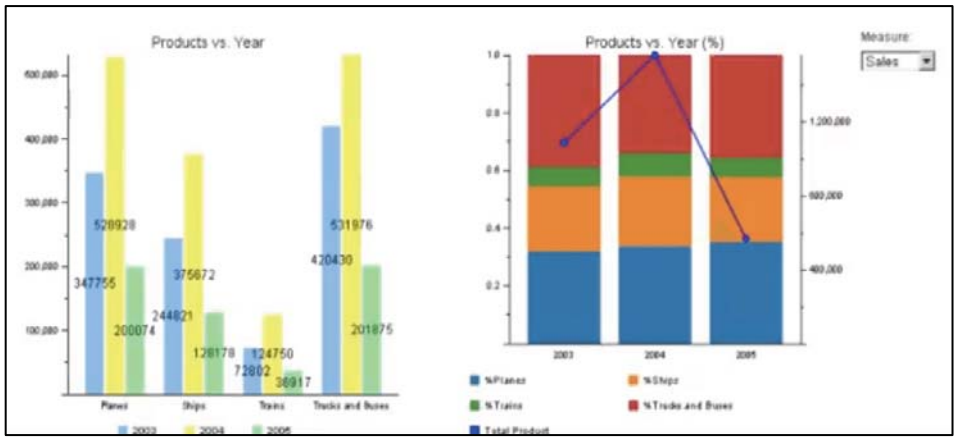


Figura 75. Vista previa de un tablero

3.8.2 Tableros en Oracle

Los tableros en Oracle son más sencillos y atractivos que Pentaho. Además de que no requieren conocimientos de diseño, hay muchos bocetos de gráficos ya diseñados que pueden usarse en Oracle. Para crear un dashboard, se debe seleccionar un tipo de *layout* (figura 76). Una vez que los datos hayan sido cargados, se dispondrá de un panel donde se arrastrarán los elementos gráficos a desplegar.

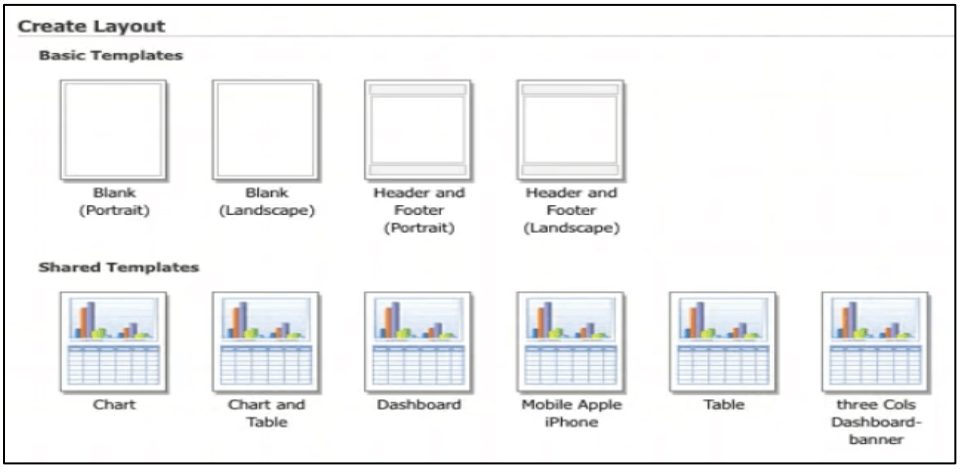


Figura 76. Selección de plantilla para el tablero

Muchas de las características de los reportes en Oracle se mantienen en la parte de creación de tableros. Al igual que para crear un reporte, se debe iniciar sesión

en la plataforma, y cargar los datos que se van a utilizar (el proceso es exactamente igual que los anteriores). Considerando que los datos ya están presentes, se dispone de un lienzo en el que los usuarios pueden libremente armar su estructura de columnas y secciones, dando formato personalizado. En otro caso se puede seleccionar uno predefinido para ahorrar tiempo de elaboración.

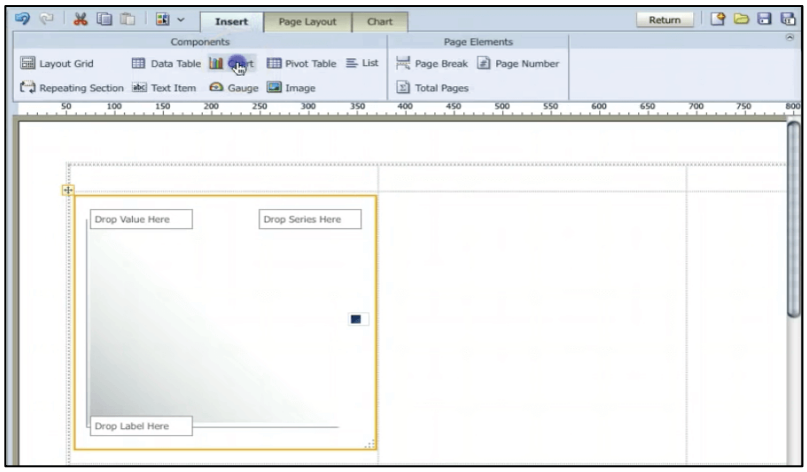


Figura 77. Lienzo en blanco

Los bocetos que se encuentran en la parte superior se arrastran hacia la parte de los recuadros, y posteriormente se llenan los parámetros con los datos de la izquierda (figura 78). Además de ser muy automatizado, los gráficos despliegan la información rápidamente.

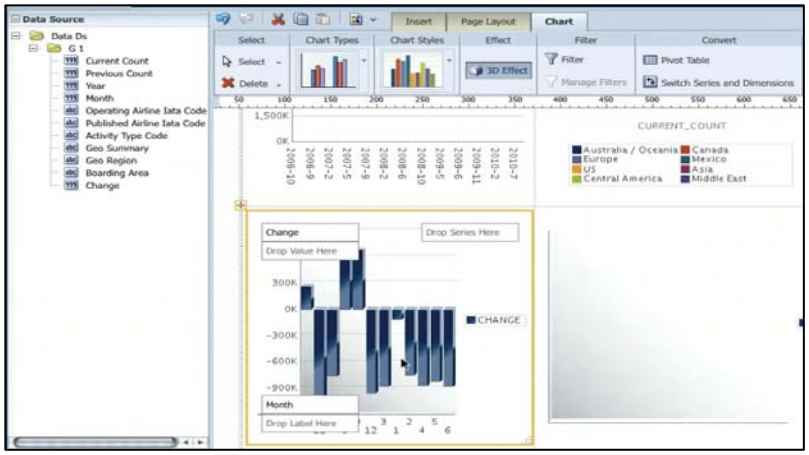


Figura 78. Consultas y elementos gráficos

A excepción de la carga de los datos, crear un tablero en Oracle consiste en solo arrastrar y soltar. Los datos interactúan con los elementos, y posteriormente los usuarios pueden editar las etiquetas para añadir las descripciones correspondientes. Los elementos disponibles para la construcción del tablero incluyen más variedad para los usuarios. (Más detalles en el Anexo D, página 189)

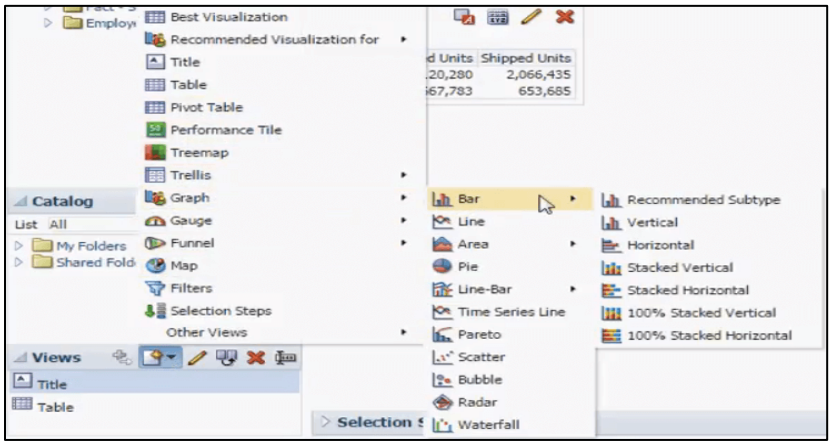


Figura 79. Presentaciones disponibles

Además de las gráficas de barras, se cuenta con diferentes modalidades de tablas, mapas, filtros, gráficas de calibración, series de tiempo, entre otras. Provee un buen contenido para solventar los requerimientos de los usuarios. Esta parte de creación de tableros destaca por su simplicidad y por una interfaz que facilita bastante a los usuarios, a comparación de otras herramientas de Oracle.

Figura 80. Vista previa del tablero

3.8.3 Tableros en SQL Server

Crear tableros en Oracle es sencillo, pero en Power BI la experiencia es aún más satisfactoria. Power BI es una herramienta de reciente creación, diseñada para complementar el flujo de almacenamiento de datos en SQL Server. Debe ser descargada independientemente y añade una interfaz parecida a los productos de Microsoft Office.

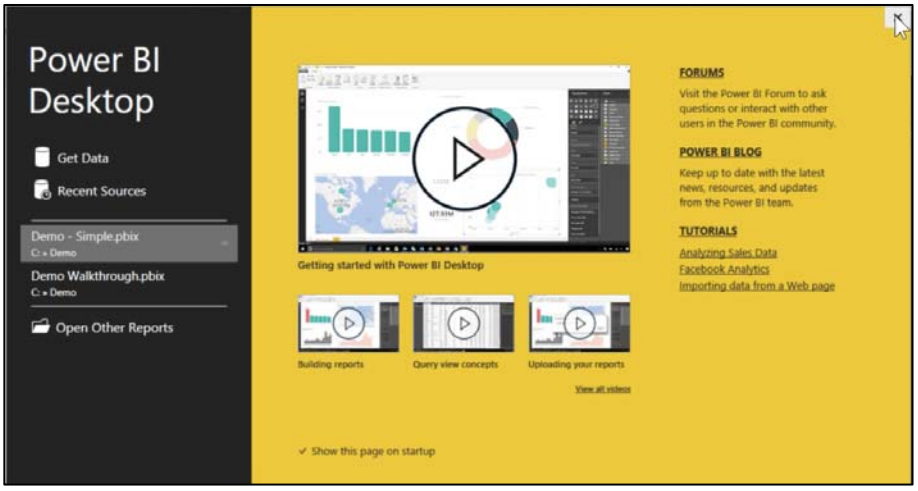


Figura 81. Creación de tablero en Power BI

Power BI ofrece a los usuarios la posibilidad de crear tableros, debido a que años atrás, SQL Server no incluía este servicio. Como es natural, al igual que en Reporting Services, esta herramienta favorece fuertemente la integración con otras herramientas de Microsoft. Como se muestra en la siguiente figura, al ingresar una fuente de datos, se muestran en primer instancia tres herramientas propias de Microsoft.

Esta no es exactamente una ventaja, puesto que otras herramientas también tienen la posibilidad de ingresar estas fuentes. Algo destacable en esta herramienta, es el gran parecido que tiene con herramientas como Word, lo que ayuda a los usuarios a familiarizarse con su interfaz.

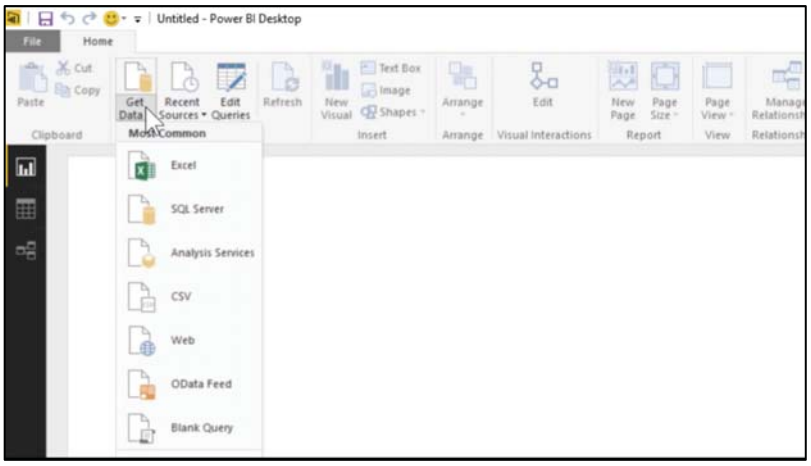


Figura 82. Selección de fuente de datos

Como se muestra en la figura 82, se debe seleccionar una fuente de datos. Posteriormente, se muestra un panel con elementos que pueden ser arrastrados hacia el tablero, y llenados con los datos de la fuente cargada (figura 83). Los elementos del panel son bastante intuitivos debido a sus pictogramas. Permiten al usuario detectar fácilmente que tipo de contenido requiere para su tablero.

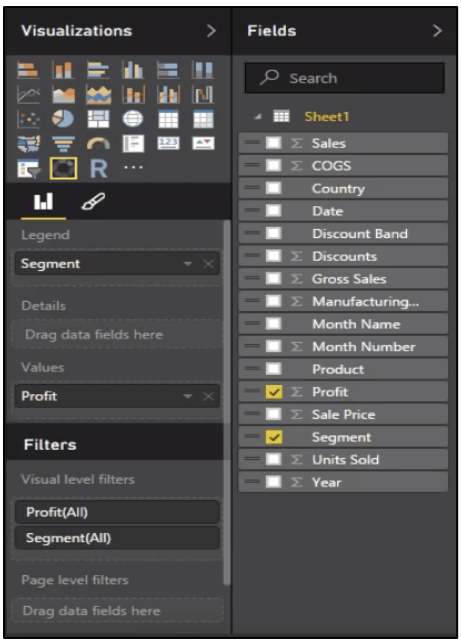


Figura 83. Elementos del tablero de mando

Hay una ventaja muy significativa en Power BI. En algunos casos podría considerarse una desventaja. Microsoft es una empresa que destaca por innovar y mejorar constantemente sus productos. Power BI tiene un buen catálogo de presentaciones gráficas, pero también incluye un amplio catálogo de fórmulas de cálculo, como las que se incluyen en Excel.

Aquí entra en juego una herramienta llamada Power Query, la cual puede usarse en Excel, o en Power BI. Es una herramienta que permite crear transformaciones o consultas para columnas personalizadas. En este caso, orientado principalmente hacia despliegue de datos de texto, justo como los archivos de Excel.

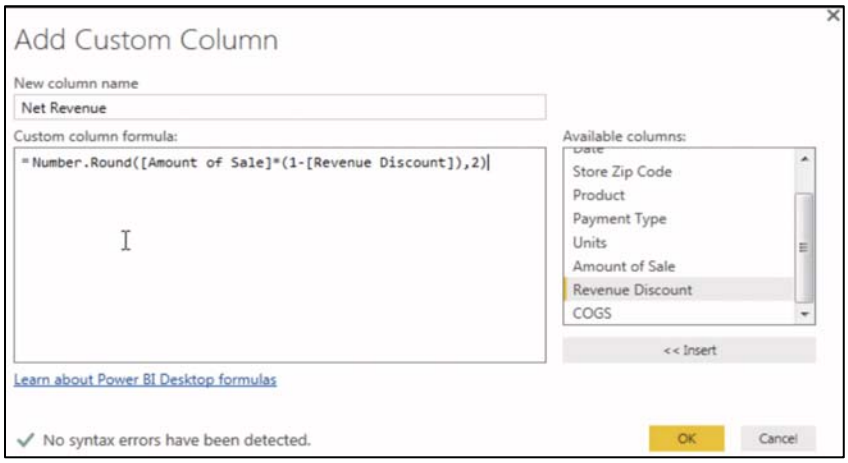


Figura 84. Fórmula de Power Query

La razón por la que podría ser una desventaja, es que es un lenguaje diferente a los lenguajes SQL y MDX, por lo que un usuario tendría que aprender un nuevo lenguaje, propio de esta herramienta. Este lenguaje puede ser más familiar para aquellos usuarios de Excel, que requieren de fórmulas para realizar reportes.

Aunque no es muy difícil de comprender, Microsoft tiene un buen soporte para el apoyo en cada una de sus herramientas, por lo que ofrece diversos tutoriales en línea para aplicar sus herramientas. El proceso para realizar un tablero termina siendo el mismo que en las anteriores herramientas.

Usando los datos cargados, se arrastran hacia el tablero y se aplican presentaciones correspondientes. En el caso de Power BI, el usuario puede acomodar sus elementos libremente, sin un formato formal (sin secciones ni columnas). Finalmente, se muestra un ejemplo de la visualización previa de un tablero de mando en Power BI. Microsoft complementó la falta de tableros en SQL Server con una herramienta muy simple de usar y con un diseño bastante atractivo. (Más detalles en el Anexo D, página 196)

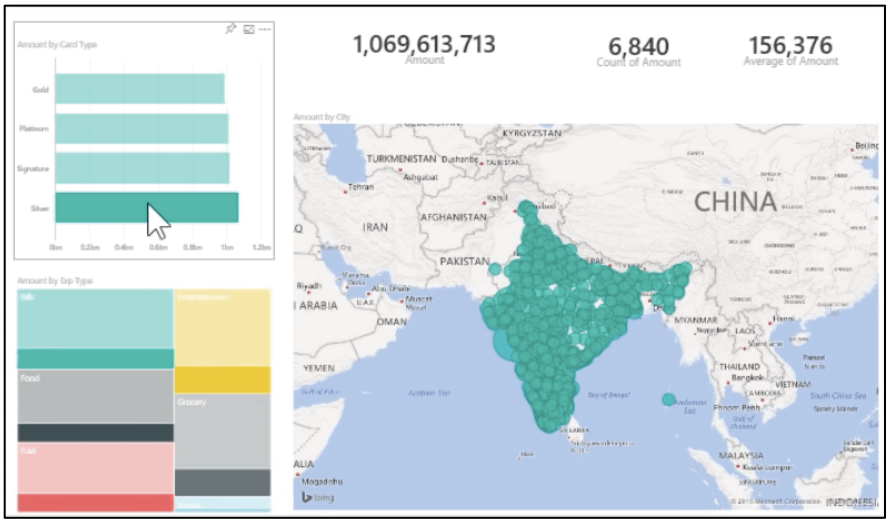
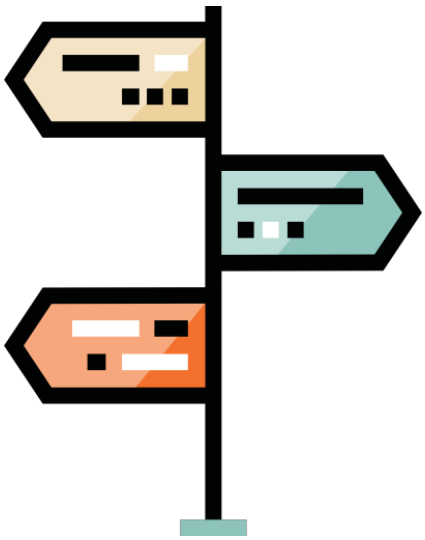


Figura 85. Tablero de mando en Power BI

A continuación se presentan, tanto la evaluación, como el resultado de la experiencia de usuario sobre el uso de estas herramientas, así como la conclusión personal aprendida sobre el desarrollo del presente trabajo.



CAPÍTULO 4. EVALUACIÓN, RESULTADOS Y CONCLUSIONES

4. CRITERIOS DE EVALUACIÓN

Para crear una evaluación personal, es necesario comentar que los criterios seleccionados han sido encontrados de forma empírica, es decir, basados en la experimentación y observando los hechos en cada herramienta. Tomando como base la propuesta de evaluación de Gartner, decidí considerar importantes todos aquellos aspectos que puedan ser comparados por cualquier usuario. Encontré factores diferentes para cada herramienta dependiendo de cual es su función. A continuación describo en forma general, cuales fueron los criterios mas destacados sobre el uso de estas herramientas:

• **Facilidad de instalación (ETL, OLAP, reportes y tableros de mando):** Engloba el acceso a la herramienta y el modo en que fue instalada. Algunas herramientas requieren de más tiempo de instalación, añadido al hecho de que puedan necesitar configuraciones especiales (por ejemplo, utilizar comandos de la terminal o configurar la dirección de los puertos).

• **Facilidad de uso (ETL, OLAP, reportes y tableros de mando):** Una vez que una herramienta esta instalada, la primera impresión de la herramienta permite saber que tan fácil será utilizarla. La experiencia visual importa mucho en este factor, ya que el modo en el que se ajustan las funciones (posición de los elementos) o la complejidad de estas, determinarán que un usuario nuevo pueda sentirse cómodo utilizando la herramienta.

• **Fuentes de datos (ETL, OLAP, reportes y tableros de mando):** Cada una de las herramientas tiene diferentes tipos de fuentes de datos. Este factor no le dará importancia a la cantidad, sino a que tan conocidas son estas fuentes. Un usuario nuevo puede sentirse más cómodo utilizando una herramienta que se adapta a formatos que son más común de usar (como los PDF, CSV, Excel, etc).

• **Módulos disponibles (por los proveedores, Oracle, Pentaho y Microsoft):** En general, mostrar si el proveedor apoya a los usuarios por medio de *plugins* o módulos extras que se puedan instalar o comprar.

• **Formatos de salida (ETL y reportes):** Al igual que en las fuentes de datos, este factor le dará importancia a los formatos conocidos.

• **Transformaciones (ETL):** Aquí se considera la cantidad y la calidad de las funciones. Importa mucho que un usuario cuente con variedad para poder ejecutar un proceso ETL complejo.

• **Implementación del cubo (OLAP):** Probablemente la parte más compleja en todo el proceso. Este factor se parece mucho a la definición de propiedades del cubo. Sin embargo, en el caso de la implementación, se evaluará principalmente la parte física de la herramienta. Considerando el desarrollo de un cubo, se tomará en cuenta el apoyo de la herramienta hacia el usuario (que tan simple es gráficamente y que tan largo es el proceso).

• **Definición de propiedades del cubo (OLAP):** Aquí se tomará en cuenta la parte lógica de la creación del cubo, desde el uso de la herramienta. Por ejemplo, que tan sencillo es crear una dimensión o añadir una jerarquía. También se considera importante notar que tan modulares se encuentran los componentes para poder definir las propiedades.

• **Exploración del cubo (OLAP):** Enfocado en que tanto apoya le herramienta visualmente para que un usuario pueda realizar una consulta con facilidad.

• **Tipos de reportes (Reportes):** Basado en los formatos y en que tanta personalización se puede dar a un reporte (visto desde los *layouts*).

• **Propiedades del reporte (Reportes):** Cantidad de elementos para añadir a un reporte, considerando también, que sean intuitivos para comprenderlos sin tener mucho conocimiento de su funcionalidad.

• **Visualización del reporte (Reportes):** Efectividad de la vista previa de un reporte (visualmente hablando, que tan bien se muestra).

• **Tipos de tableros (Tableros de mando):** Basado en los formatos y en que tanta personalización se puede dar a un tablero (visto desde los *layouts*).

• **Publicación del reporte (Reportes):** Simpleza para desplegar un reporte.

• **Propiedades de tableros (Tableros de mando):** Cantidad de elementos para añadir a un tablero, considerando también, que sean intuitivos para comprenderlos sin tener mucho conocimiento de su funcionalidad.

• **Visualización de tableros (Tableros de mando):** Efectividad de la vista previa de un tablero (visualmente hablando, que tan bien se muestra).

Consideré estos factores debido a mi experiencia utilizando estas herramientas, cada uno de los componentes se evaluó por separado tomando el siguiente rango de evaluación según sus funcionalidades:

• **No disponible**: Únicamente cuando una herramienta no contiene una determinada función o característica.

• **Regular:** La herramienta dispone de una función o característica muy deficiente. Su puntuación es de 1.

• **Bueno:** La herramienta dispone de una función o característica lo suficientemente aceptable. Su puntuación es de 2.

• **Muy bueno:** La herramienta dispone de una función o característica que aporta más de lo necesario, cumple muy bien su funcionalidad o es demasiado simple para los usuarios. Su puntuación es de 3.

4.1 Resultados

Considerando los criterios anteriores y las pruebas realizadas en el capítulo 3, llegué a las siguientes conclusiones, separadas por cada proceso:

• **Procesos ETL:** Las tres herramientas cumplieron bien su funcionalidad. Pentaho destaca por la simpleza de sus interfaces, mientras que Oracle y SQL Server rinden más en potencial funcional.

a) Pentaho tiene una configuración inicial tardada y poco intuitiva. Su facilidad de uso contrarresta su dificultad de instalación. Cuenta con fuentes de datos suficientes para orígenes de datos. Pentaho provee a sus usuarios de módulos disponibles (externos) que pueden adaptarse a nuevos requerimientos. Sus transformaciones y formatos de salida también son aceptables.

b) Oracle también tiene una configuración complicada, principalmente por la configuración de su middleware. El modo en el que aparecen los elementos crea dificultad para usar la herramienta. La página oficial de Oracle proporciona módulos propios, pero no es posible obtener módulos externos. Sin embargo, cuenta con buena cantidad de fuentes de datos, transformaciones y formatos de salida.

c) Microsoft es más simple de instalar. Al igual que la mayoría de sus herramientas, es muy intuitiva de usar. Tiene una cantidad muy buena de fuentes de datos y transformaciones. La cantidad de formatos de salida es limitada pero aceptable. A diferencia de Pentaho, Microsoft no provee directamente de módulos externos. Estos módulos deben ser obtenidos mediante otras páginas ajenas a Microsoft.

Consideración	**Pentaho**	**Oracle**	**SQL Server**
Facilidad de instalación	1	1	2
Facilidad de uso	3	1	2
Fuentes de datos	2	3	3
Módulos disponibles (externos)	2	N/D	1
Transformaciones	2	3	3
Formatos de salida	2	2	2
Resultado	**12**	**10**	**13**

Tabla 1. Comparativa de procesos ETL

• **Procesos OLAP:** Pentaho tiene un alcance limitado en cuanto a funcionalidad, pero al mismo tiempo le permite caracterizarse por la brevedad en la que un usuario puede entender y manejar sus capacidades. SQL Server y Oracle ofrecen un soporte OLAP poderoso en cuanto a funcionalidad, pero aquí la experiencia de usuario es más incomoda.

a) Pentaho no fue simple de configurar (principalmente por Mondrian). Esta primera impresión no debe de asustar a los usuarios, ya que una vez que la herramienta esta instalada cuenta con mucha facilidad de uso, una cantidad suficiente de fuentes de datos, facilidad para implementar y definir las propiedades del cubo. La facilidad de exploración del cubo es intermedia.

b) Oracle no es fácil de instalar. Las interfaces apoyan al usuario, de modo que su uso se vuelve muy simple. Las fuentes de datos son suficientes. La implementación, definición y exploración del cubo resulta algo limitada, pero suficiente un proceso OLAP.

c) Microsoft mantiene la simpleza para instalar. Su facilidad de uso es intermedia. Cuenta con suficientes fuentes de datos. Gracias a su asistente, el proceso de implementar y definir un cubo es muy fácil. La exploración del cubo resulta suficientemente aceptable.

Consideración	Pentaho	Oracle	SQL Server
Facilidad de instalación	1	1	2
Facilidad de uso	3	3	2
Fuentes de datos	2	2	2
Implementación del cubo	3	2	3
Definición de propiedades del cubo	3	2	3
Exploración del cubo	2	2	2
Resultado	**14**	**12**	**14**

Tabla 2. Comparativa de procesos OLAP

• **Creación de reportes:** Pentaho ofrece simpleza y libertad para la creación de reportes. La interfaz de Oracle para el diseño de reportes es poco atractiva, sin embargo ofrece buenas capacidades. SQL Server por su parte, tiene un diseño de reportes un poco más metódico, ya que apoya en el proceso de creación de un reporte sin ofrecerle al usuario muchas opciones para desplegar sus datos.

a) La instalación de Pentaho es muy simple, al igual que su uso. Cuenta con una cantidad muy baja de fuentes de datos y formatos de salida. En contraposición, ofrece una cantidad suficiente de tipos de reportes. Las propiedades del reporte son muy aceptables. La publicación y la visualización del reporte se encuentra en una rango aceptable.

b) Oracle repite la misma situación de dificultad para instalarse (no como herramienta individual, sino como conjunto por sus requerimientos). Sin embargo, esta herramienta destacó por su facilidad de uso, las propiedades y la publicación del reporte. También cuenta con fuentes de datos, formatos de salida, tipos de reportes y una visualización de reportes dentro de un rango aceptable.

Consideración	**Pentaho**	**Oracle**	**SQL Server**
Facilidad de instalación	3	1	3
Facilidad de uso	3	3	3
Fuentes de datos	1	2	1
Tipos de reportes	2	2	1
Propiedades del reporte	3	3	3
Publicación del reporte	2	3	3
Visualización del reporte	2	2	2
Formatos de salida	1	2	1
Resultado	**17**	**18**	**17**

Tabla 3. Comparativa de creación de reportes

c) Microsoft por su parte, cuenta con una instalación sencilla (también es fácil como un conjunto de herramientas, no por separado). Es muy fácil de usar, añadido a un buen conjunto de propiedades del reporte. La publicación también es simple. Esta herramienta destaca por su habilidad para hacer reportes rápidos, sin embargo cuenta con una cantidad limitada de tipos de reportes, fuentes de datos y formatos de salida.

• **Creación de tableros de mando:** Pentaho tiene muchas debilidades en este proceso, tiene un alcance limitado de opciones. En lo que corresponde a SQL Server y Oracle, ofrecen una calidad excelente para la creación de tableros. Sus bocetos prediseñados y su facilidad de uso permiten al usuario generar un tablero sin requerir mucho esfuerzo.

a) Pentaho cuenta con una instalación compleja (por uso de terminal y configuración). Esta herramienta no es intuitiva y no es fácil de usar. Las propiedades de los tableros son limitadas. En cuanto a sus fuentes de datos, sus tipos de tableros y la visualización, se encuentran en un rango aceptable. Cuenta con una puntuación baja debido a la necesidad de conocimiento de diseño para utilizar esta herramienta. Esto no implica que su funcionalidad disminuya, simplemente complica más el uso a los usuarios.

Consideración	**Pentaho**	**Oracle**	**SQL Server**
Facilidad de instalación	1	1	3
Facilidad de uso	1	2	3
Fuentes de datos	2	2	2
Tipos de tableros	2	2	2
Propiedades de tableros	1	3	3
Visualización de tableros	2	3	3
Resultado	**9**	**13**	**16**

Tabla 4. Comparativa de creación de tableros

b) Oracle requiere de configuración para instalar (igual que sus otros componentes). Tiene una facilidad de uso aceptable, al igual que sus fuentes de datos. Sus tipos de tableros por omisión también son aceptables. En cuanto a las propiedades y la visualización de los tableros, cumplen muy bien su función.

c) Microsoft destacó mucho en esta herramienta. Muy simple de instalar y usar. Sus fuentes de datos y tipos de tableros son aceptables. Las propiedades y la visualización de los tableros resultaron de gran calidad.

• **Complejidad de la infraestructura:** Las herramientas de Pentaho fueron experimentadas desde un equipo con el sistema operativo Linux (Ubuntu). El usuario requiere ciertos conocimientos para utilizar la terminal como medio de instalación y ejecución de algunos componentes. Sin embargo, la descarga y la instalación no requieren de mucho tiempo. Las herramientas de Oracle fueron probadas en Windows 7. Aunque no son tan complejas de instalar, si requieren tiempo de instalación, ya que algunos componentes como el gestor de base de datos requieren de una serie de pasos para su configuración. Por otra parte, SQL Server también fue instalado en Windows 7. Su instalación es más breve y simple de configurar. Principlamente porque a excepción de Power BI, las otras tres herramientas (ETL, OLAP y reportes) pueden instalarse con un solo proceso.

• **Rango de costos:** Pentaho no tiene ningún costo por sus herramientas. En el caso de que se requieran características adicionales o un plan de soporte, puede ser adquirido por un costo de 11,000 dolares anuales. Oracle tiene un costo de 3850 dólares para su edición estándar, y tiene como restricción un mínimo de compra de 5 unidades. SQL Server tiene un costo por licencia de 3,717 dólares para su versión estándar y 14,256 dólares para su versión de empresa. Power BI se maneja como servicio con un costo de 10 dólares mensuales por usuario. (Para mayor información revise el portal de cada proveedor)

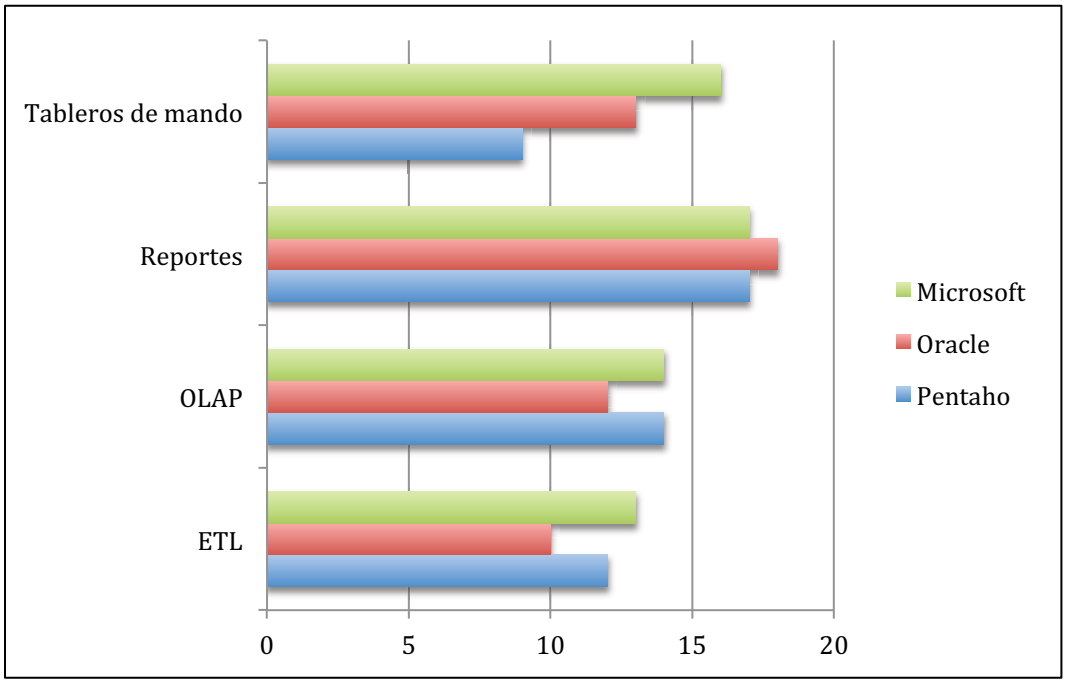


Figura 86. Comparativa final por cada área

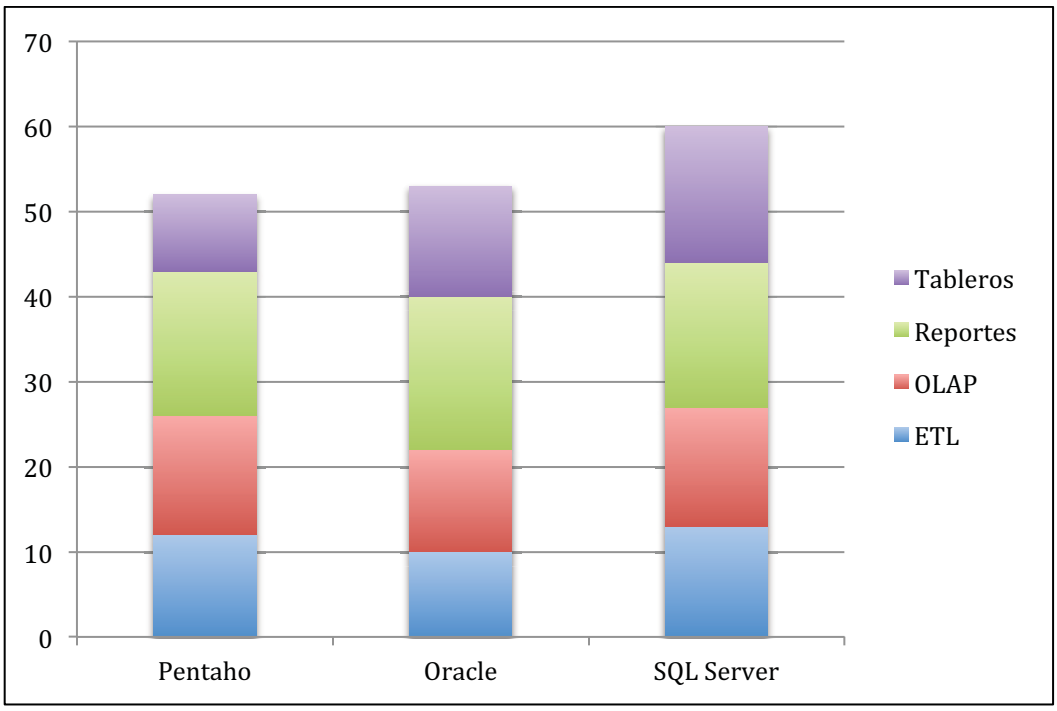


Figura 87. Comparativa final de proceso

Por medio de las tablas comparativas se obtiene un resultado final de 52 puntos para Pentaho, 53 para Oracle y 60 para SQL Server. Considerando el costo de Pentaho, es notable considerar que tiene el potencial para competir con una fuerte herramienta como Oracle. Sin embargo SQL Server destaca bien en cada uno de sus procesos, logrando igualar o superar a sus competencias. Considerando mi experiencia personal, puedo garantizar que SQL Server merece el puesto que tiene en el cuadro mágico de Gartner.

4.2 Conclusiones

Con la experiencia obtenida por medio de esta comparativa, se recomienda ampliamente considerar los factores costo / beneficio. Estas herramientas fueron seleccionadas debido a que son líderes en el campo de la inteligencia de negocios. Sin embargo, se debe considerar que los beneficios obtenidos entre una herramienta y otra no son muy diferentes, aunque los precios sí impactan en la elección del usuario.

Un usuario poco exigente, que requiera un almacén de datos para un negocio personal puede trabajar de forma excelente con Pentaho. Debido a que Pentaho no difiere mucho entre sus ediciones de empresa y comunidad, se puede hacer uso de su versión gratuita y explotar cada una de sus capacidades. Si un usuario de Pentaho no estuviera conforme con la construcción de sus tableros, puede comprar una suscripción a Power BI y así complementar su solución.

En el caso de una empresa consolidada, los servicios de Pentaho pueden ser efectivos pero no suficientes. Aquí SQL Server es completamente recomendado como solución a un sistema de almacenamiento de datos. Las capacidades de SQL Server pueden potenciar fuertemente el mando gerencial debido a su excelente paquete de herramientas. El costo puede ser una gran inversión si la empresa logra explotar las capacidades de esta herramienta.

4.3 Trabajo futuro

La presente tesis expuso la teoría básica de los almacenes de datos así como una comparativa entre herramientas líderes en el mercado. La comparativa demostró la funcionalidad principal en cada una de las herramientas, con la finalidad de comprender el resultado obtenido contra el costo del producto. Se logró llevar a cabo una experiencia completa en el proceso de creación de un almacén de datos, para luego realizar un análisis entre las herramientas que apoyan la toma de decisiones basadas en los datos.

El conocimiento adquirido consta de ver un enfoque general del almacenamiento de datos, así como de formar un criterio propio en cuanto a las herramientas que proponen soluciones a las necesidades de negocio. Recomiendo este trabajo a todos los usuarios que no tienen conocimiento en el área de inteligencia de negocios. Aunque la compañía Gartner publica anualmente los estudios sobre estas herramientas, algunas de sus estadísticas tienen un costo para acceder al reporte completo.

El futuro de la inteligencia de negocios es muy favorable, ya que los usuarios generan una mayor cantidad de datos que son usados por las empresas para generar conocimiento. La tecnología cada vez está más presente en el comercio electrónico por lo que los usuarios cada vez utilizan con mayor frecuencia dispositivos portátiles para realizar compras. Poco a poco, la tendencia del estudio de la inteligencia de negocios se aproxima al almacenamiento en la nube, mayor capacidad de datos sin necesidad de utilizar una base de datos. Las consultas empiezan a ser enfocadas al tiempo real, tratando de ser lo más rápidas posibles. Actualmente ya se cuentan con aplicaciones móviles que funcionan como visores de los tableros de mando. En el futuro se espera poder realizar procesos de inteligencia de negocios desde un celular o una tableta digital. Con el crecimiento exponencial de datos, el número de científicos especializados en el análisis de estos se incrementarán en los próximos años.

4.4 Reflexión final

La motivación principal para realizar este trabajo, fue la curiosidad para aprender sobre el tema de los almacenes de datos. Antes de realizarlo no sabía nada. Una de las cosas que considero más importantes que aprendí a lo largo de mi trayectoria en la Facultad de Ciencias, fue la capacidad de aprender a aprender. Sin duda alguna, el conocimiento evoluciona y es necesario aprender a adaptarse a los cambios con facilidad. Cuando termine mis materias de la carrera supe que algo me había faltado. Quería conocer más sobre la inteligencia de negocios, y está era la mejor manera de poder hacerlo.

Al realizar este trabajo trate de exponer el resumen de lo más importante para que cualquier interesado en realizar un almacén de datos pueda llevarlo a cabo satisfactoriamente. Personalmente construí este trabajo abordando los temas que considere más relevantes, a modo de tratar de crear un tutorial, del modo como a mí me hubiera gustado aprender. Cuando yo busqué información sobre almacenes de datos, comprendí que cada autor tenía un enfoque o una manera de enseñar diferente, pero solo aquello en lo que coincidían podía servir como factor común para obtener la mejor base de este material. Esperando que exista un crecimiento potencial en el área de las bases de datos, también espero que la cantidad de personas que se interesen por esta hermosa área incrementen, al igual que este trabajo pueda servir a alguien más.

REFERENCIAS

5. REFERENCIAS

5.1 Bibliograficas

[1] Ballard, C., Herreman, D., Schau, D., Bell, R., Kim, E. y Valencic, A. 1998. Data Modeling Techniques for Data Warehousing. San Jose, California. Ibm. 214 p.

[2] Bulusu L. 2013. Open Source Data Warehousing and Business Intelligence. Boca Raton, Florida. CRC Press. 432 p.

[3] Conesa, J. y Curto, J. 2010. Introducción al Business Intelligence. Rambla del Poblenou, Barcelona. Editorial UOC. 240 p.

[4] Kimball, R., Ross, M., Becker, B., Mundy, J. y Thornthwaite, W. 2008. The Data Warehouse Lifecycle Toolkit. Practical Techniques for Building Data Warehouse and Business Intelligence Systems. 2ª edición. Indianapolis, Indiana. John W. & Sons Inc. 636 p.

[5] Kimball, R. y Ross, M. 2013. The Data Warehouse Toolkit. The Definitive Guide to Dimensional Modeling. 3ª edición. Indianapolis, Indiana. John Wiley & Sons, Inc. 565 p.

[6] Langit, L., Goff, K., Mauri, D., Malik, S. y Welch, J. 2009. Smart Business Intelligence Solutions with Microsoft SQL Server 2008. Redmond, Washington. Microsoft Press. 791 p.

[7] Tsai, J., Shekhar, K. y Vasquez, J. 2007. Oracle Business Intelligence Standard Edition One Tutorial. Redwood City, California. Oracle USA, Inc. 226 p.

5.2 De internet

[8] BNamericas®. Oracle Corp.
https://www.bnamericas.com/company-profile/es/oracle-corp-oracle (24/10/2017)

[9] Dataprix®. Datawarehouse manager
http://www.dataprix.com/datawarehouse-manager#x1-520003.4.5.3 (24/10/2017)

[10] De los Angeles, M. 2006. Procesamiento Analítico en Linea (OLAP). Corrientes, Argentina.
http://exa.unne.edu.ar/informatica/SO/OLAPMonog.pdf (17/10/2015).

[11] Jiawei Han, Micheline Kamber y Jian Pei, Data Mining Concepts and Techniques. United States of America.
http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

[12] Magic Quadrant for Business Intelligence and Analytics Platforms. 2016.
http://www.gartner.com/doc/reprints (01/01/2017)

[13] Mendez, A., Mártire, A., Britos, P. y Garcia-Martínez, R. Fundamentos de Data Warehouse. Buenos Aires, Argentina.
http://artemisa.unicauca.edu.co/~ecaldon/docs/bd/fundamentosdedatawarehouse.pdf (16/10/2016).

[14] Microsoft®. Analysis Services
https://msdn.microsoft.com/es-es/library/bb522607.aspx (15/05/2017)

[15] Microsoft®. What is Power Bi?
https://powerbi.microsoft.com/en-us/what-is-power-bi/ (04/05/2017)

[16] Oracle®. Oracle® Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide. 2006.
http://docs.oracle.com/cd/E10415_01/doc/bi.1013/b31767.pdf (02/05/2017)

[17] Pentaho® Marketplace
https://www.pentaho.com/marketplace/ (30/06/2018)

[18] Ponniah, P. 2001. Data warehousing fundamentals. A Comprehensive Guide for IT Professionals.
https://anuradhasrinivas.files.wordpress.com/2013/03/data-warehousing-fundamentals-by-paulraj-ponniah.pdf (14/11/2015)

[19] ¿Qué es el cuadrante mágico de Gartner? 2017.
http://www.solopiensoentic.com/cuadrante-magico-de-gartner (13/05/2017)

[20] 1er. Taller de Investigación Red TIC´s Base de Datos Multidimensionales. María Trinidad Serna Encinas
http://slideplayer.es/slide/9512770 (12/01/2018)

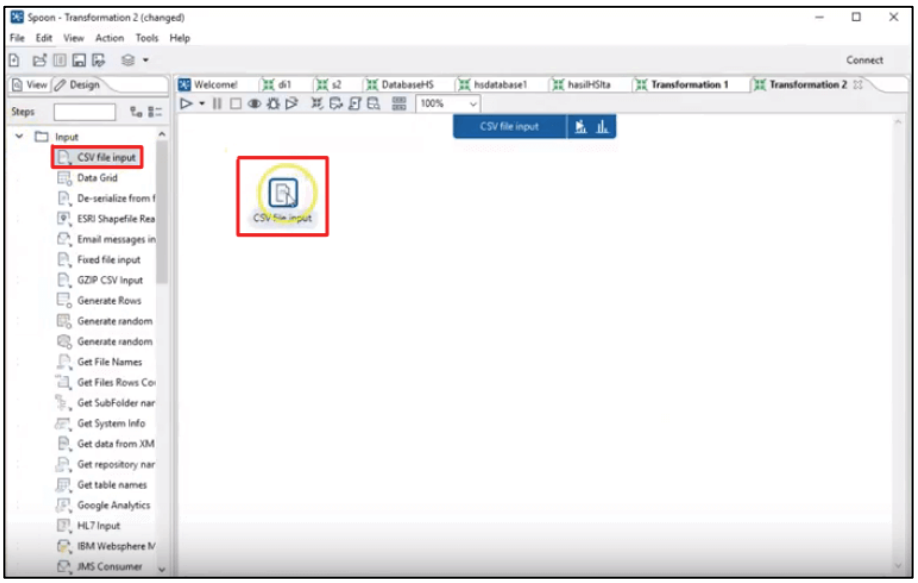
ANEXOS

Anexo A: Creación de un proceso ETL

a) Pentaho (Linux)

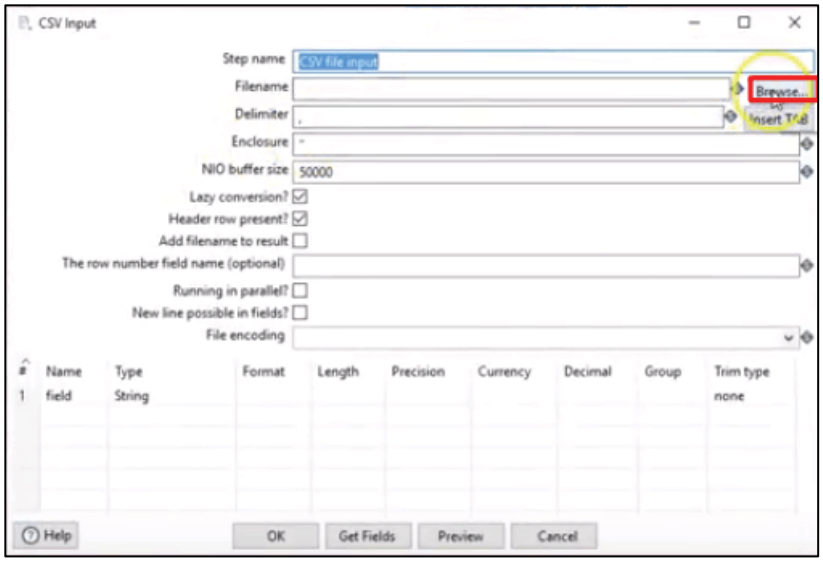
1.- Primero, ejecutar **Pentaho Data Integration** desde una terminal (cd /ruta donde se instalo PDI/data-integration → sh spoon.sh):

2.- Seleccionar una **fuente de datos de entrada** (panel izquierdo → input → CSV), arrastrar y pegar en la parte derecha:

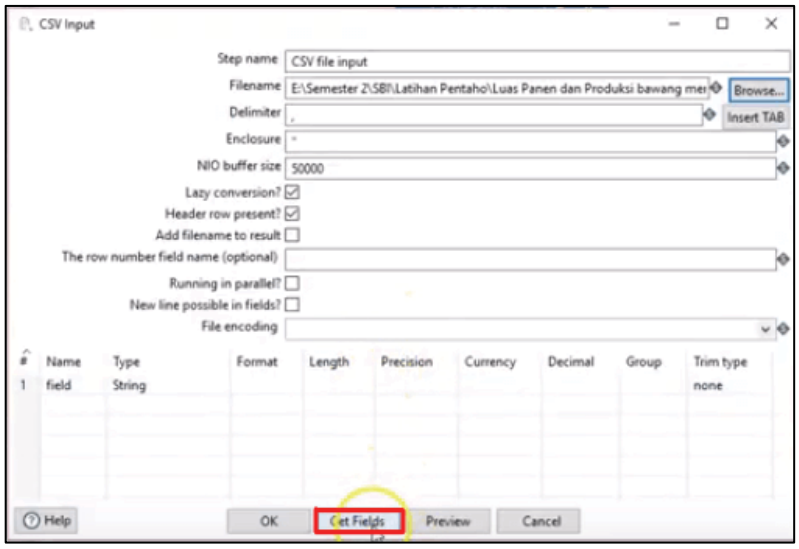


3.- Dar click en la **edición de configuración** de la fuente:

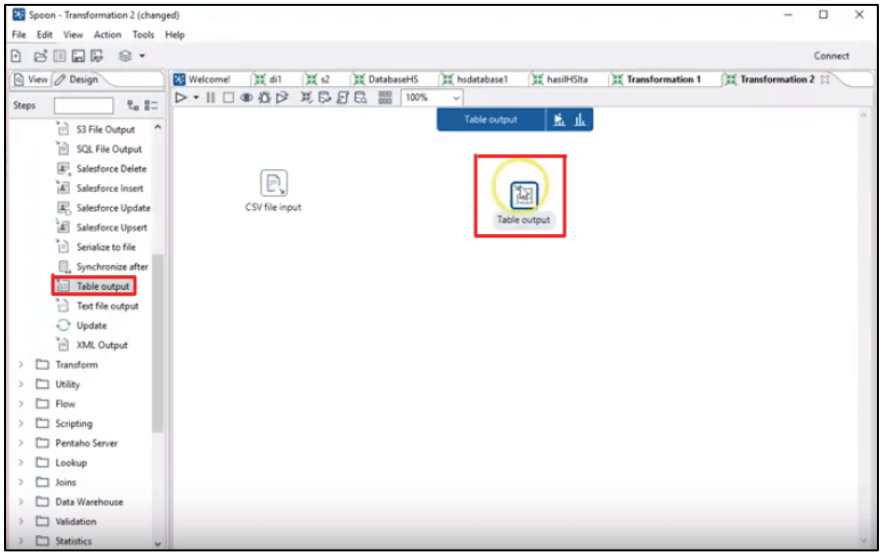
4.- Cargar el **archivo correspondiente** a la fuente de datos desde Browse:



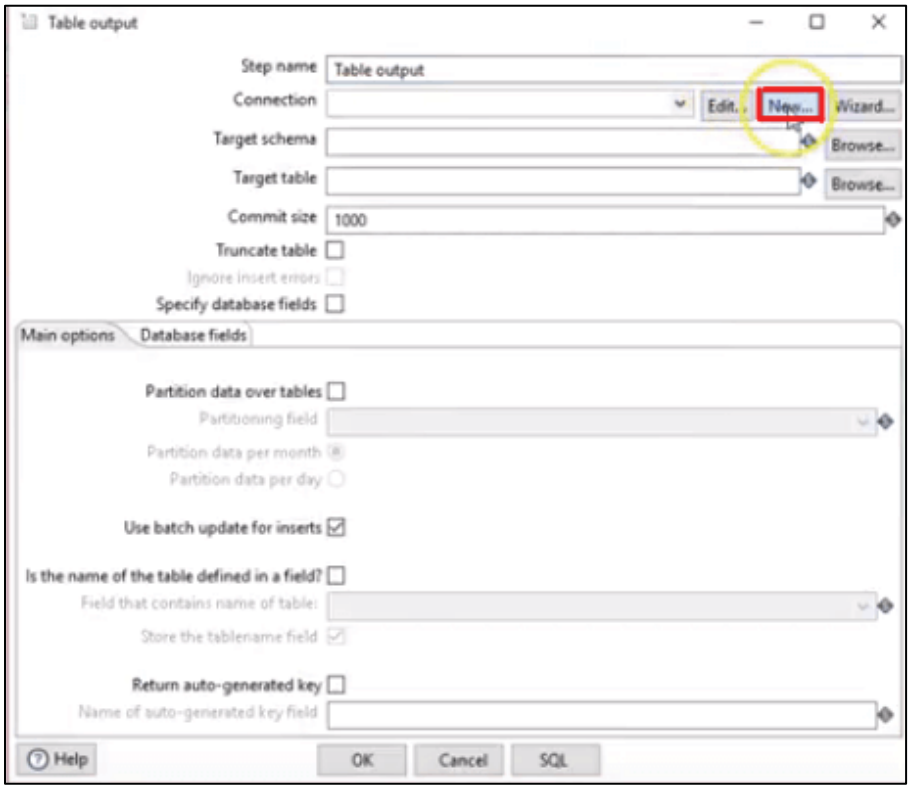
5.- Click en **get fields** para obtener los datos de la fuente:



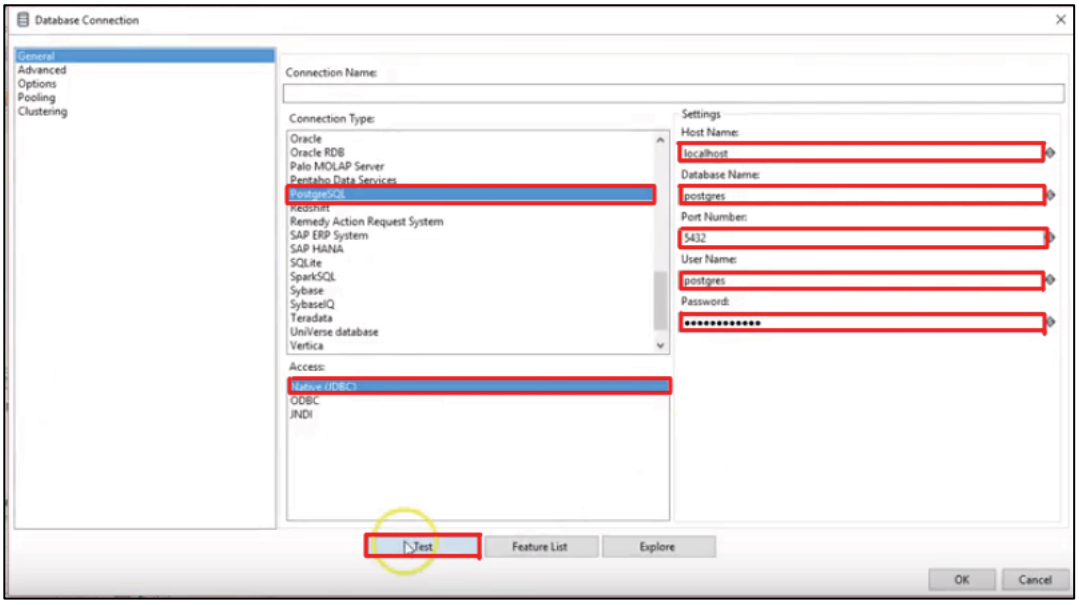
6.- Añadir una **fuente de datos de salida** (panel izquierdo → output → Table output), arrastrar y pegar en la parte derecha:



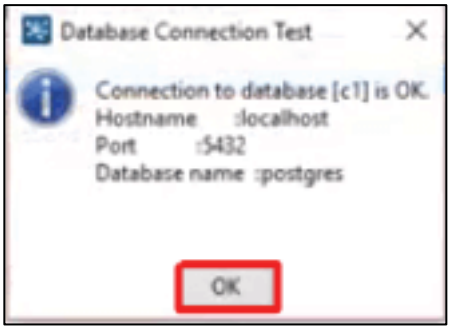
7.- Crear la configuración a una **nueva conexión** desde New:



8.- Crear la **conexión de base de datos.** Seleccionar una base, llenar los datos correspondientes a la base, y al finalizar dar click en Test:

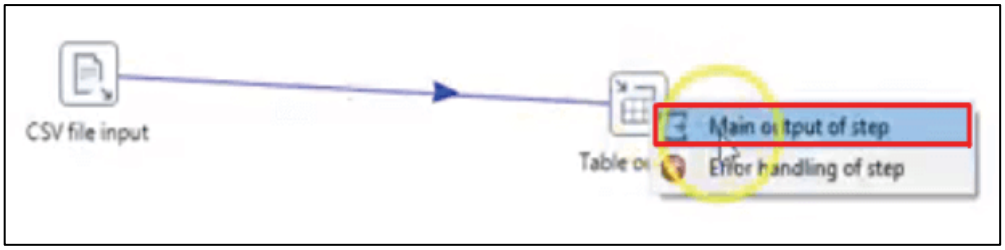


9.- Si la prueba es exitosa, se procede a dar click en OK**:**

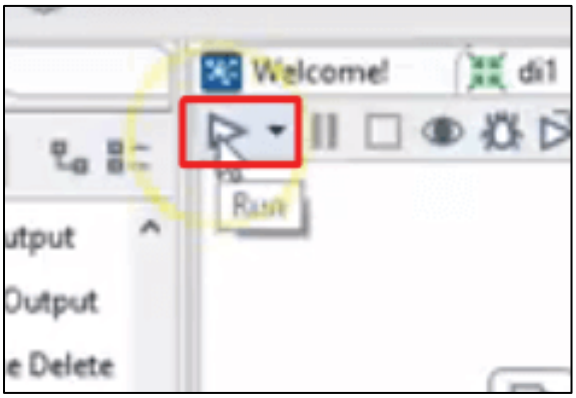


10.- Click en la **configuración de enlace** para conectar las fuentes:

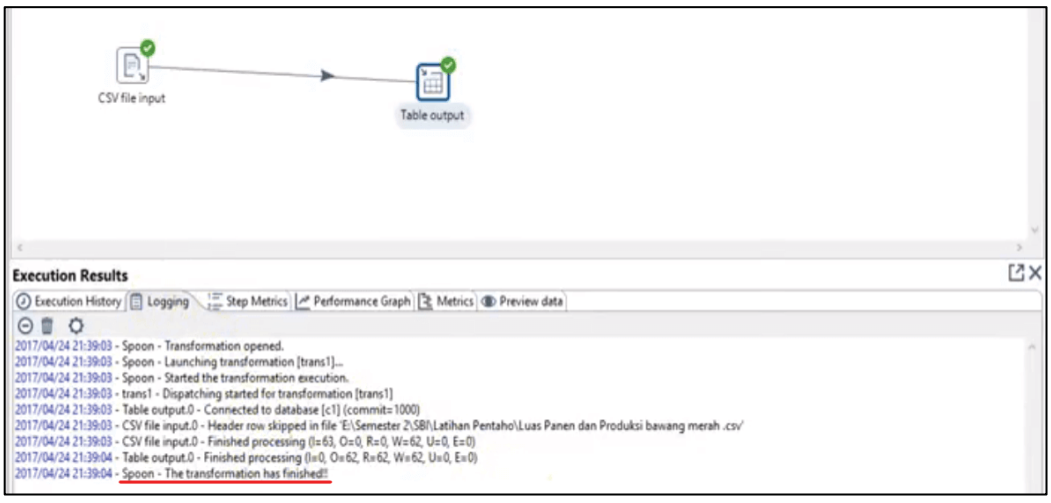
11.- Conectar las **fuentes** y luego dar click en Main output of step:



12.-Dar click en Run (Esquina izquierda superior del panel derecho):

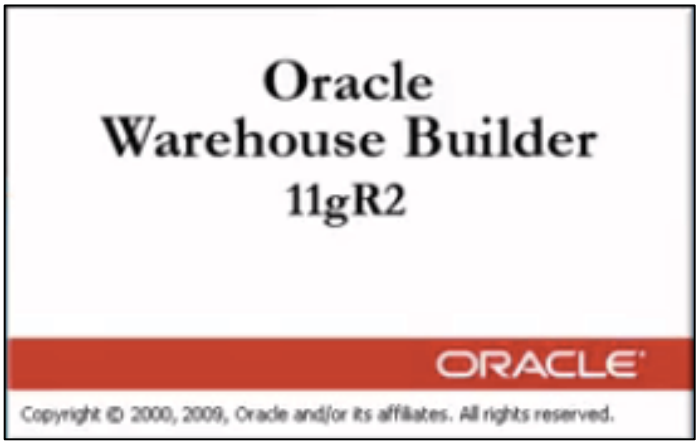


13.- Si la transformación es exitosa, aparece un mensaje (**The transformation has finished**) en la consola donde anuncia los resultados de la ejecución (Execution Results). También aparecen flechas verdes en las fuentes:



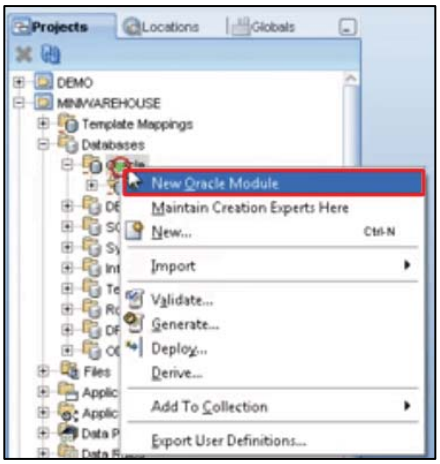
b) Oracle (Windows)

1.- Ejecutar **Oracle Warehouse Builder** (Inicio → Oracle-OraDbg_home1 → Warehouse Builder → Design Center):



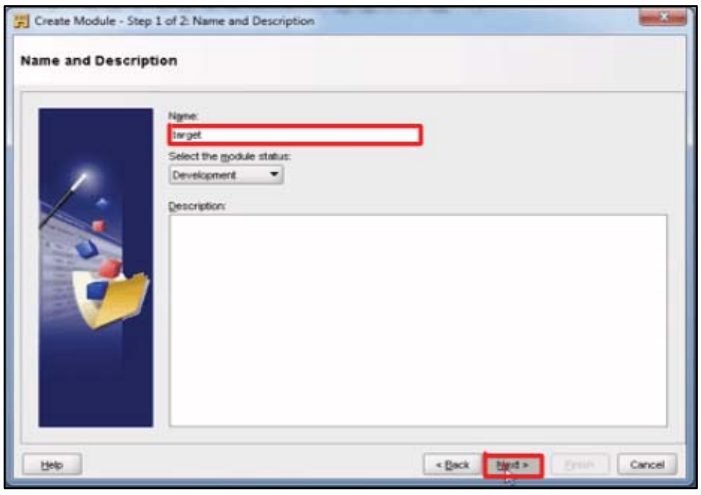
2.- Ingresar los datos correctos de la **base de datos.** Recordar que en el caso de Oracle, es necesario tener enlazado el motor de base de datos. Una vez que los datos ya están ingresados, dar click en OK:

3.- Una vez que se accede, en la parte izquierda se cargará como una fuente, los datos de la base en Oracle. Para añadir una nueva **fuente de datos** es necesario dar click en New Oracle Module (Projects → "Nombre del proyecto" → Databases → Oracle → click derecho):

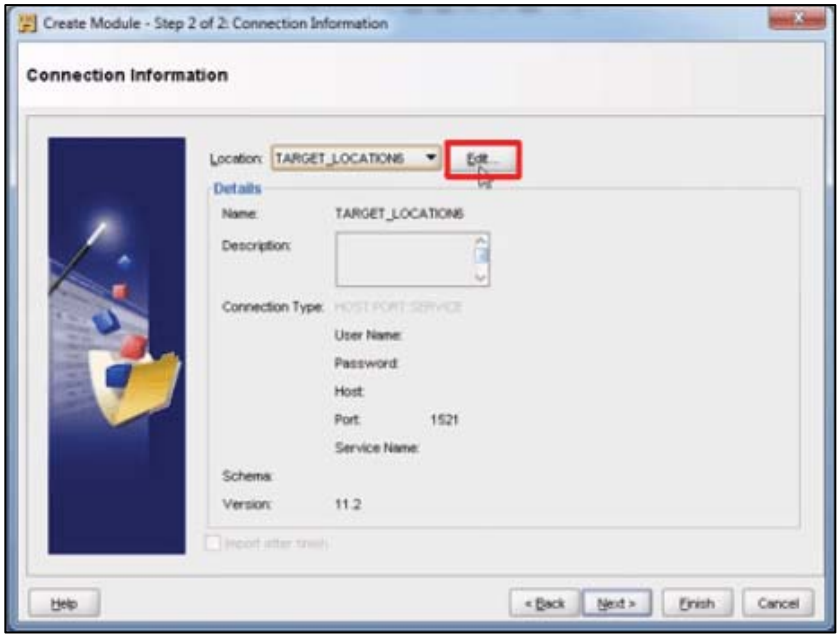


4.- Aparecerá un **asistente** para crear un modulo. Click en Next:

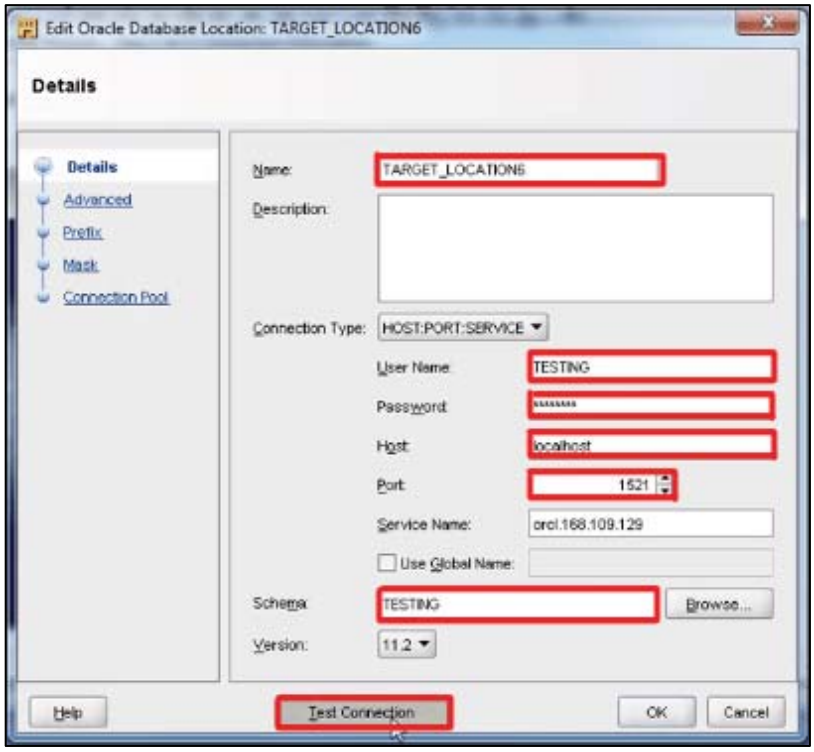
5.- Asignar un **nombre** (target) al modulo y luego click en Next:



6.- El siguiente paso consiste en **editar la conexión** del modulo. Click en Edit:

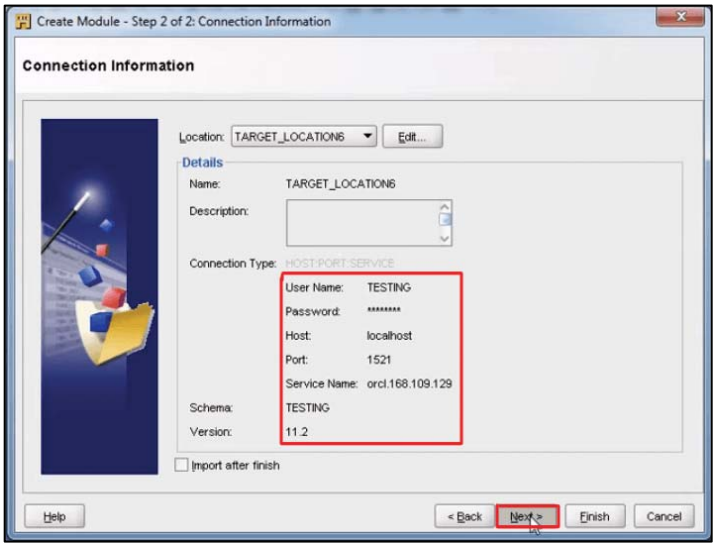


7.- Añadir los datos correspondientes a la **fuente de datos de salida.** Luego dar click en Test Connection:

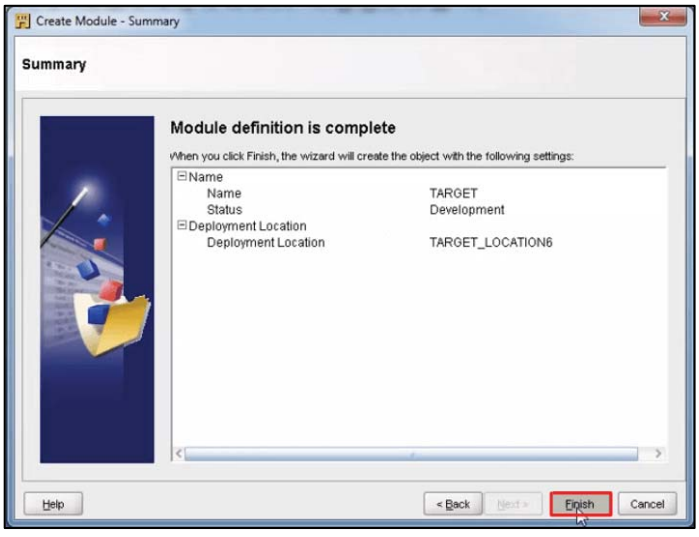


8.- Si la conexión es exitosa, dar click en OK**:**

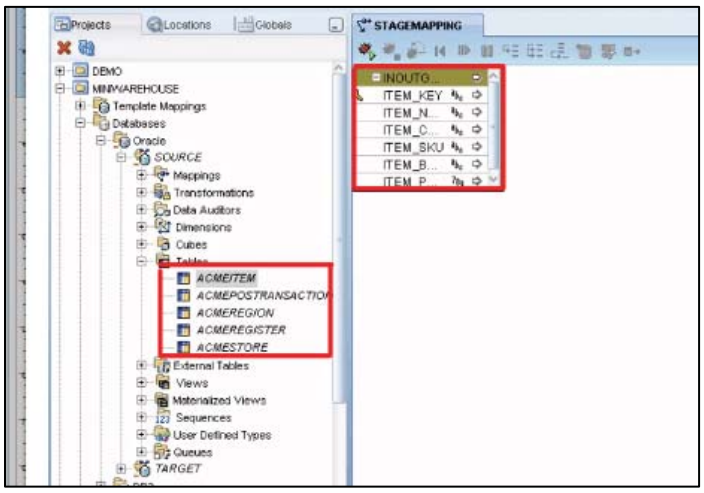
9.- Click en Next**:**



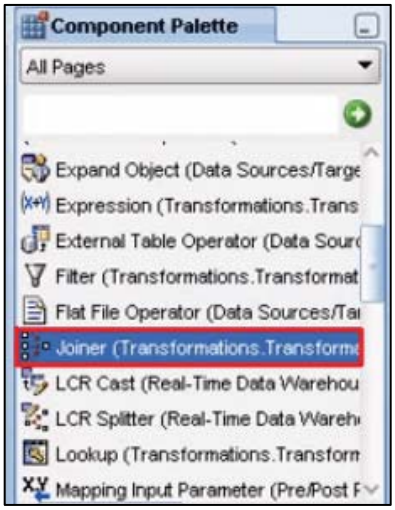
10.- Click en **finish.** La fuente de salida esta configurada:



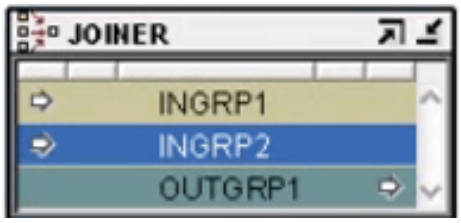
11.- Arrastrar las **tablas** que van a usarse del origen de datos ("Nombre del proyecto" → Databases → Oracle → Source → Tables) hacia la derecha:



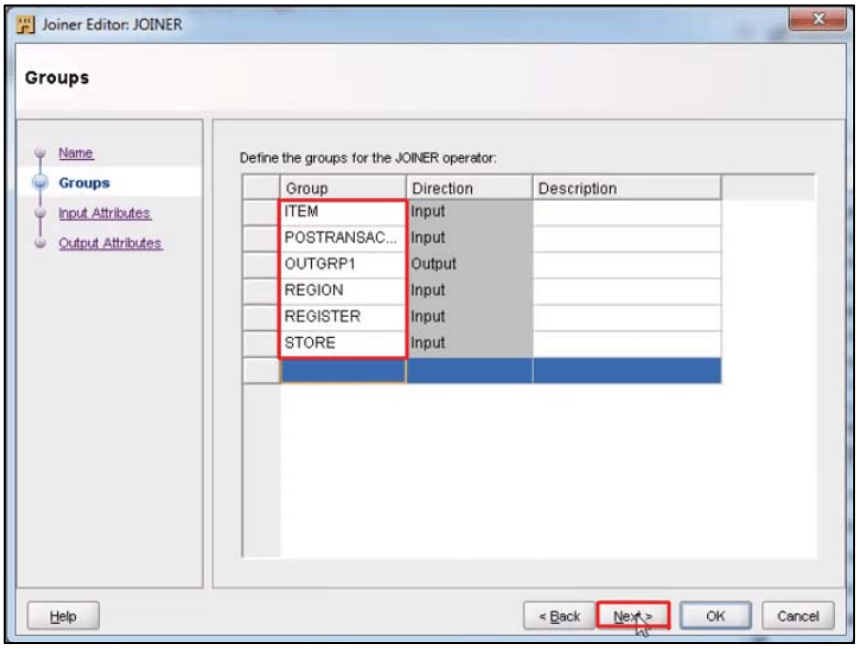
12.- Seleccionar la **transformación** a utilizar desde la paleta de componentes y arrastrarlo hacia la izquierda:



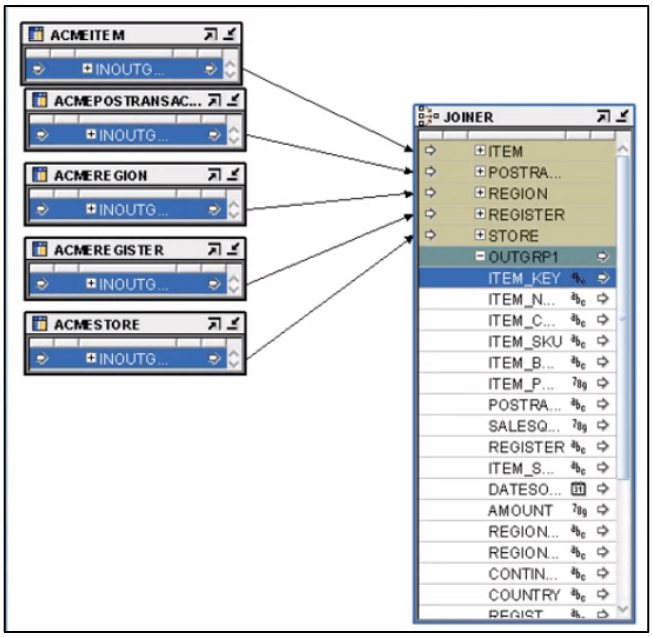
13.- Aparecerá la **transformación**. Configurar dando click derecho:



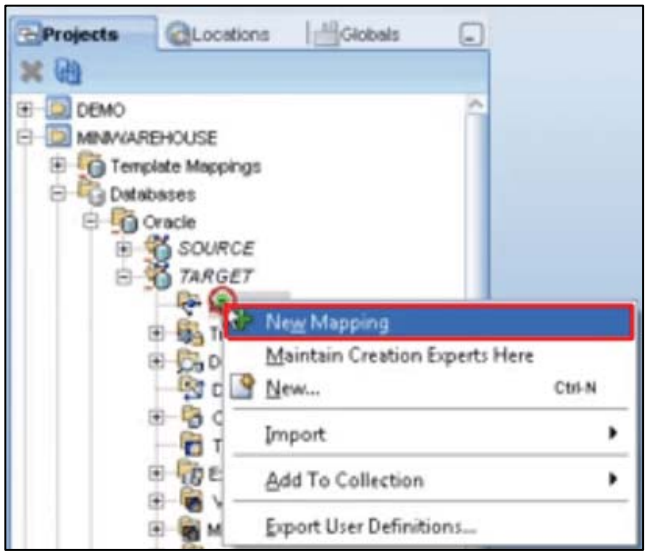
14.- Al desplegar la **configuración de la transformación** se muestra un panel, donde se deben ingresar los “grupos” que van a participar en la transformación. En este caso, los nombres de las tablas. Click en Next:



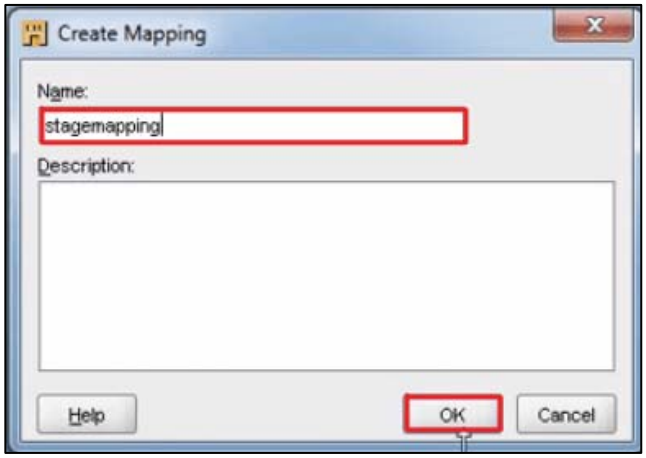
15.- Arrastrar cada tabla hacia su nombre en la transformación:



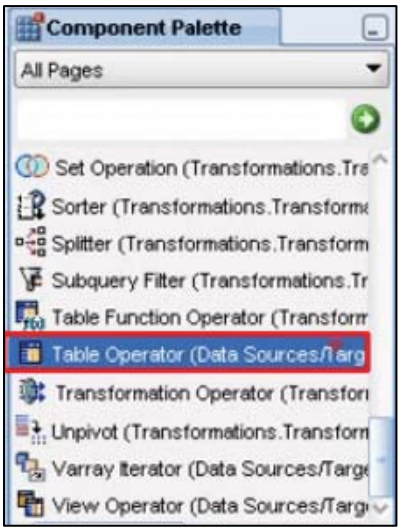
16.- En el panel izquierdo, es necesario generar un **mapping** para llevar a cabo la transformación ("Nombre del proyecto" → Databases → Oracle → TARGET → Mappings → click derecho → New Mapping):

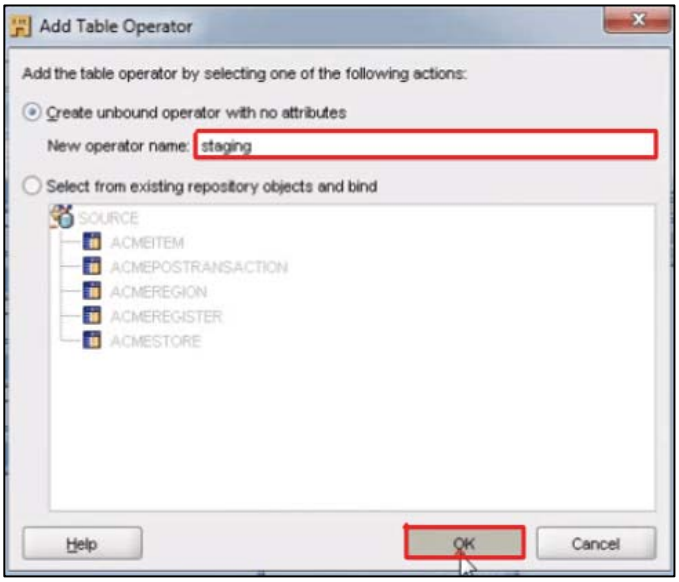


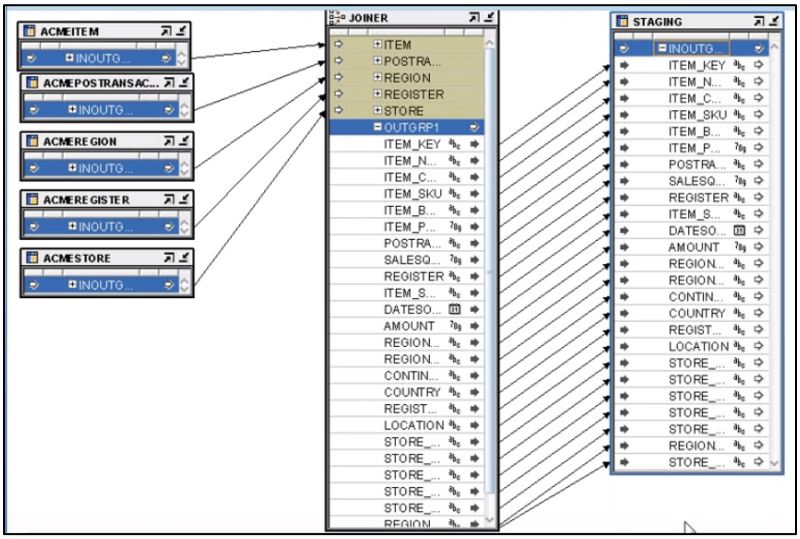
17.- Asignar un nombre al **mapping**. Un **mapping** es un objeto que se utiliza para realizar extracción, transformación y carga. Un mapping define los flujos para mover datos de fuentes al almacén. Click en OK:



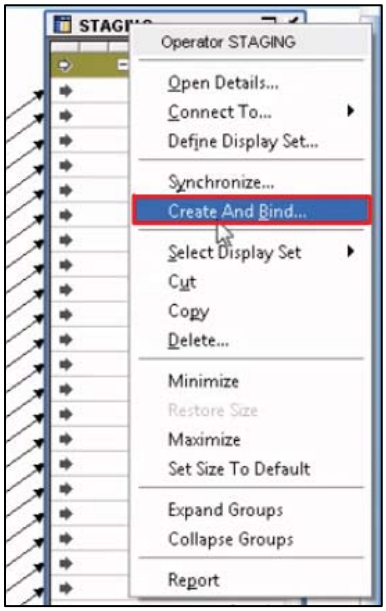
18.- En la paleta de componentes, buscar la opción de **Table Operator** y arrastrarlo hacia la izquierda. Luego de asignar un nombre (staging), dar click en OK y luego conectar con los datos de salida de la transformación:


Component Palette
All Pages
Set Operation (Transformations.Tra
Sorter (Transformations.Transforma
Splitter (Transformations.Transform
Subquery Filter (Transformations.Tr
Table Function Operator (Transform
Table Operator (Data Sources/Targ
Transformation Operator (Transfor
Unpivot (Transformations.Transform
Varray Iterator (Data Sources/Targe
View Operator (Data Sources/Targ

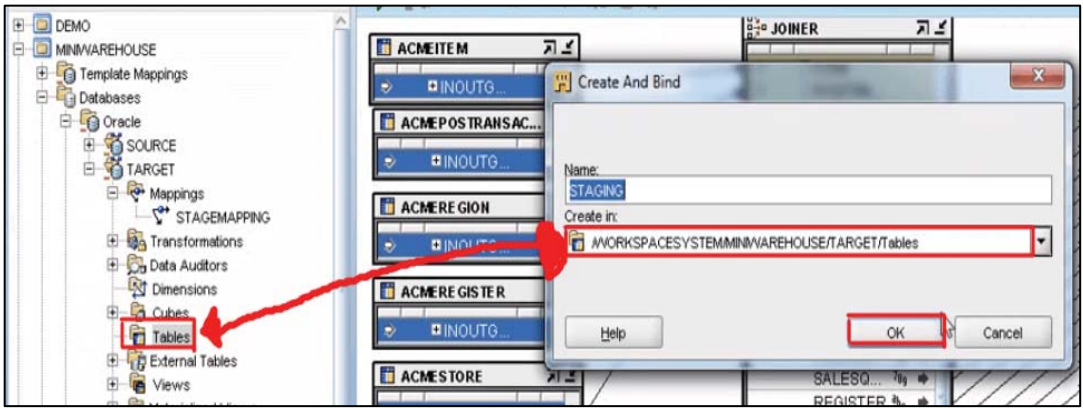

Add Table Operator
Add the table operator by selecting one of the following actions:
Create unbound operator with no attributes
New operator name: staging
Select from existing repository objects and bind
SOURCE
ACMEITEM
ACMEPOSTRANSACTION
ACMEREGION
ACMEREGISTER
ACMESTORE
Help
OK
Cancel


ACMEITEM
ACMEPOSTRANSAC...
ACMEREGION
ACMEREGISTER
ACMESTORE
INOUTG...
JOINER
ITEM
POSTRA...
REGION
REGISTER
STORE
OUTGRP1
ITEM_KEY
ITEM_N...
ITEM_C...
ITEM_SKU
ITEM_B...
ITEM_P...
SALESQ...
REGIST...
DATESO...
AMOUNT
CONTIN...
COUNTRY
LOCATION
STORE_...
STAGING

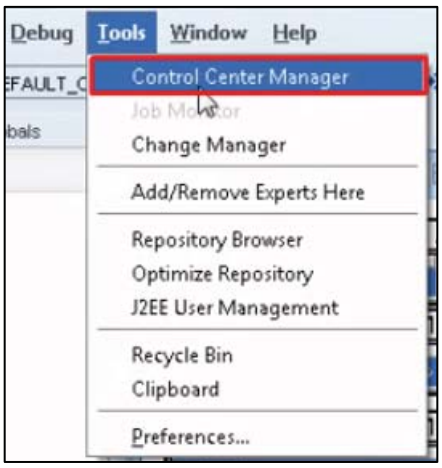
19.- Ahora es necesario enlazar la nueva tabla con la **fuente de datos de salida**. Click derecho sobre el componente STAGING y luego click en Create And Bind:



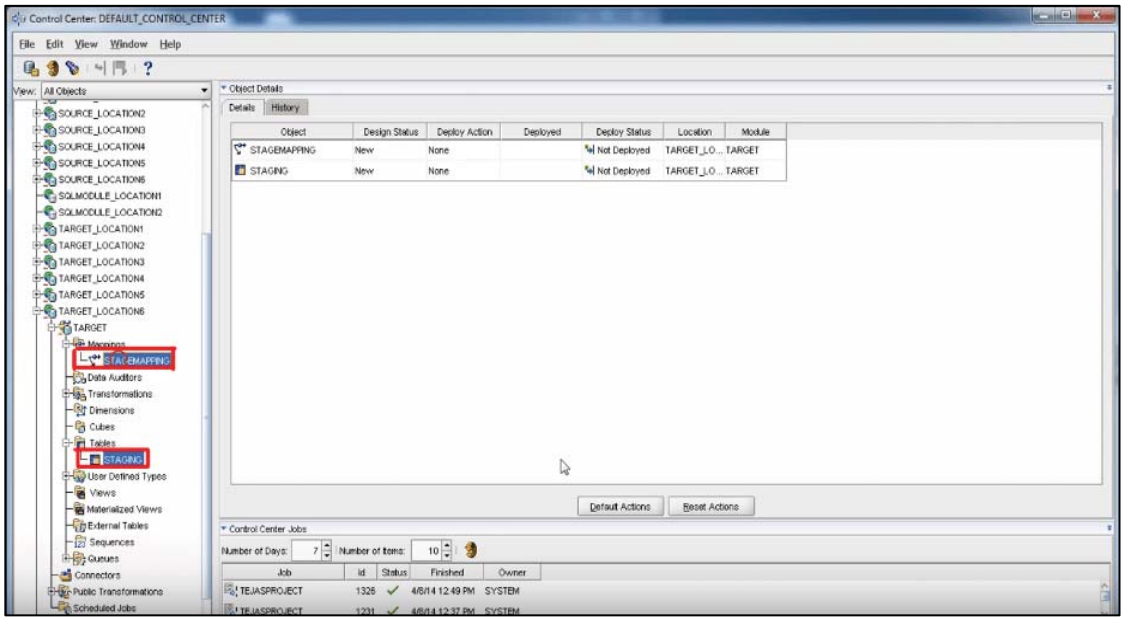
20.- Aparecerá una ventana donde será necesario ingresar un nombre (STAGING) y luego indicar la **ruta de la conexión** con las tablas ("Nombre del proyecto" → Databases → Oracle → TARGET → Tables). Esto permitirá que al llevar a cabo la transformación, los datos se moverán a la fuente de datos de salida. Primero los datos se van a "unir" por medio de la transformación JOINER y luego estos datos combinados se moverán en una sola tabla (STAGING):



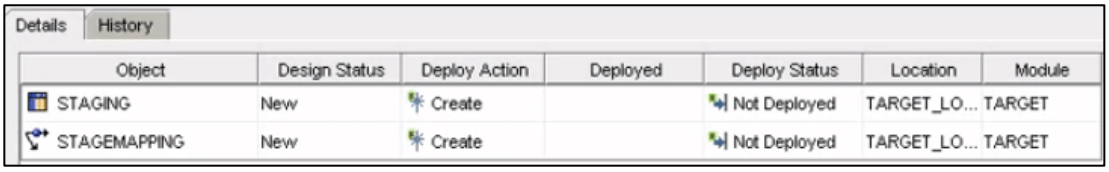
21.- Abrir **Control Center Manager** (Barra superior → Tools → Control Center Manager):



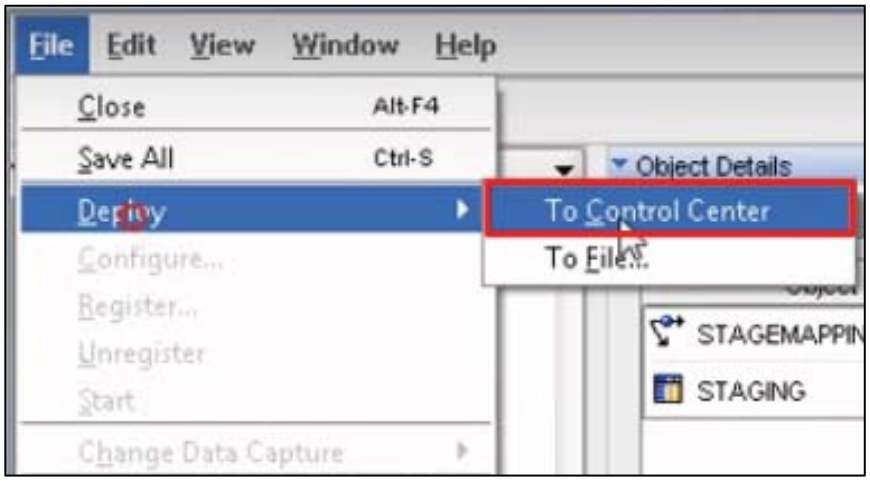
22.- En esta ventana aparecen los objetivos (TARGETS) que hay disponibles en el proyecto. Oracle los nombra por omisión “TARGET_LOCATION” y le asigna un numero por cada uno. En este caso, la fuente de salida que se ingreso tomo el nombre de “TARGET_LOCATION6”. En ese objetivo se encuentran los nombres del mapping y las tablas.



23.- En la parte de la derecha se muestra el **detalle de las operaciones** realizadas. En este punto el status muestra “Not Deployed”:

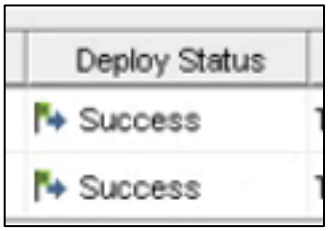


24.- Desplegar la **operación** (File → Deploy → To Control Center):



25.- En Warehouse Builder aparecerá el mensaje de “Compiling”:

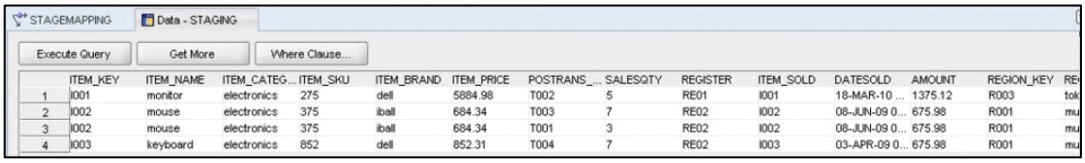
26.- Al finalizar la compilación, el status cambiará a “Success”:



27.- Para **comenzar el proceso** ETL, dar click en Start (File → Start):

28.- En la consola de Control Center se mostrará una flecha indicando que el proceso se esta llevando a cabo. Cuando finalice, verificar que los datos en la tabla sean correctos.

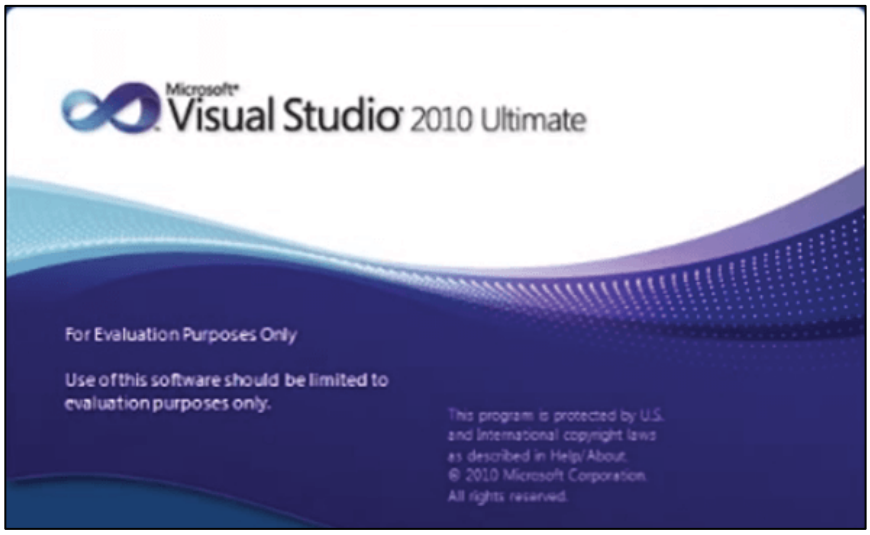
29.- Comprobar que los datos fueron transferidos observando la tabla objetivo ("Nombre del proyecto" → Databases → Oracle → TARGET → Tables → STAGING):



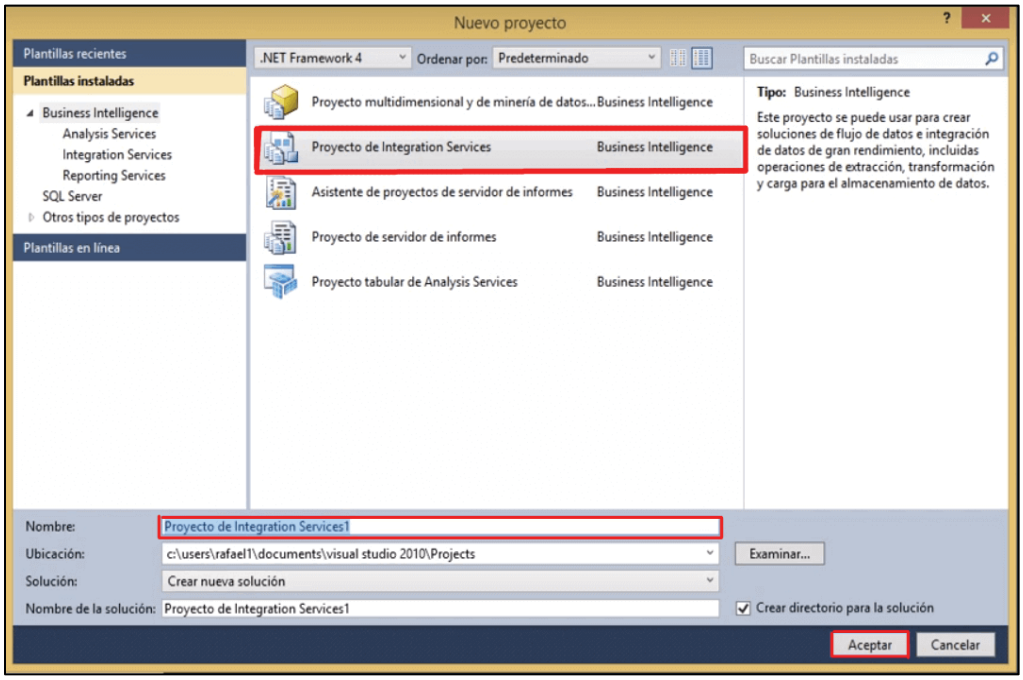
Al observar los datos, el proceso ETL fue llevado a cabo exitosamente. Los datos de las primeras tablas fueron combinados y transferidos a una tabla de STAGING.

c) SQL Server (Windows)

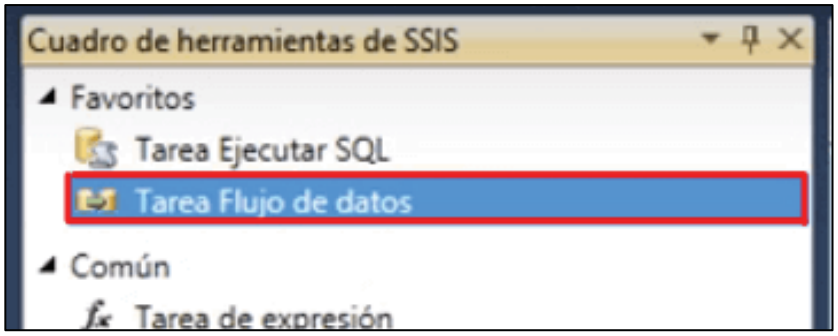
1.- Ejecutar Visual Studio 2010 (Inicio → Microsoft Visual Studio 2010 → Microsoft Visual Studio 2010):



2.- Crear un **proyecto nuevo** (Nuevo proyecto → Proyecto de Integration Services). Añadir un nombre al proyecto y especificar donde se guardará:

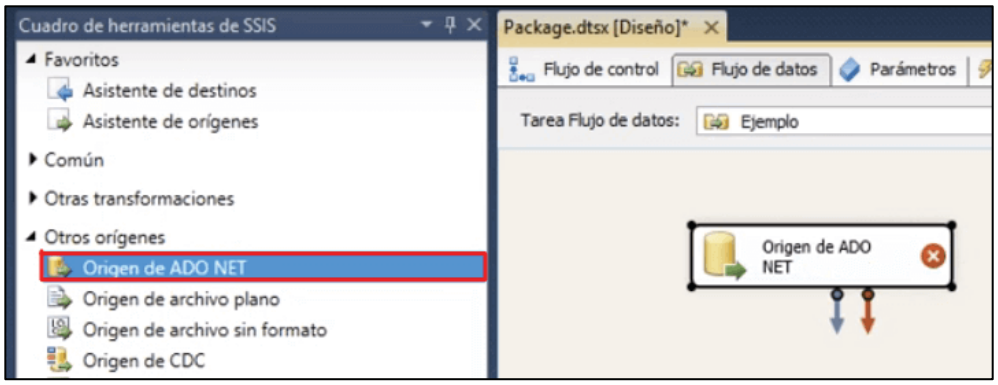


3.- En el cuadro de herramientas, seleccionar una **Tarea Flujo de datos**:

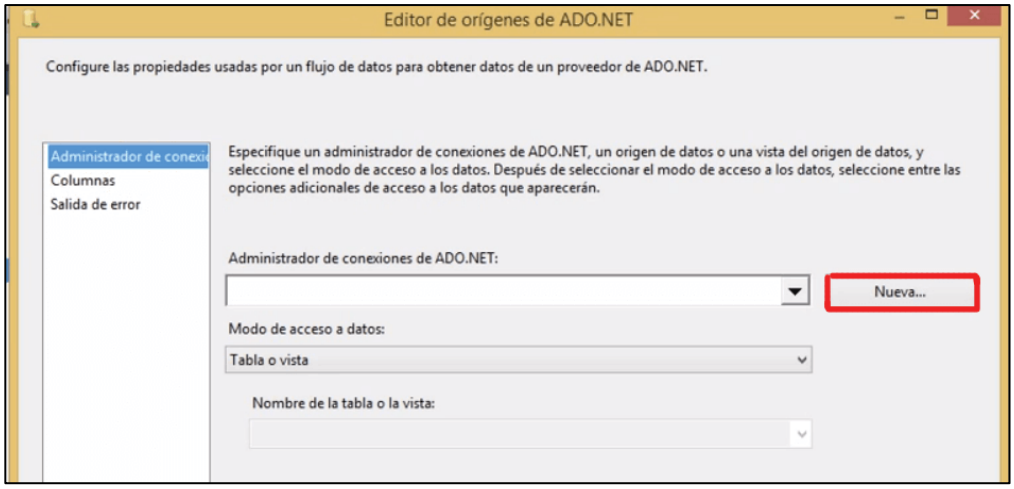


4.- Esto creará un elemento en el **Flujo de control.** Asignar un nombre y luego dar doble click:

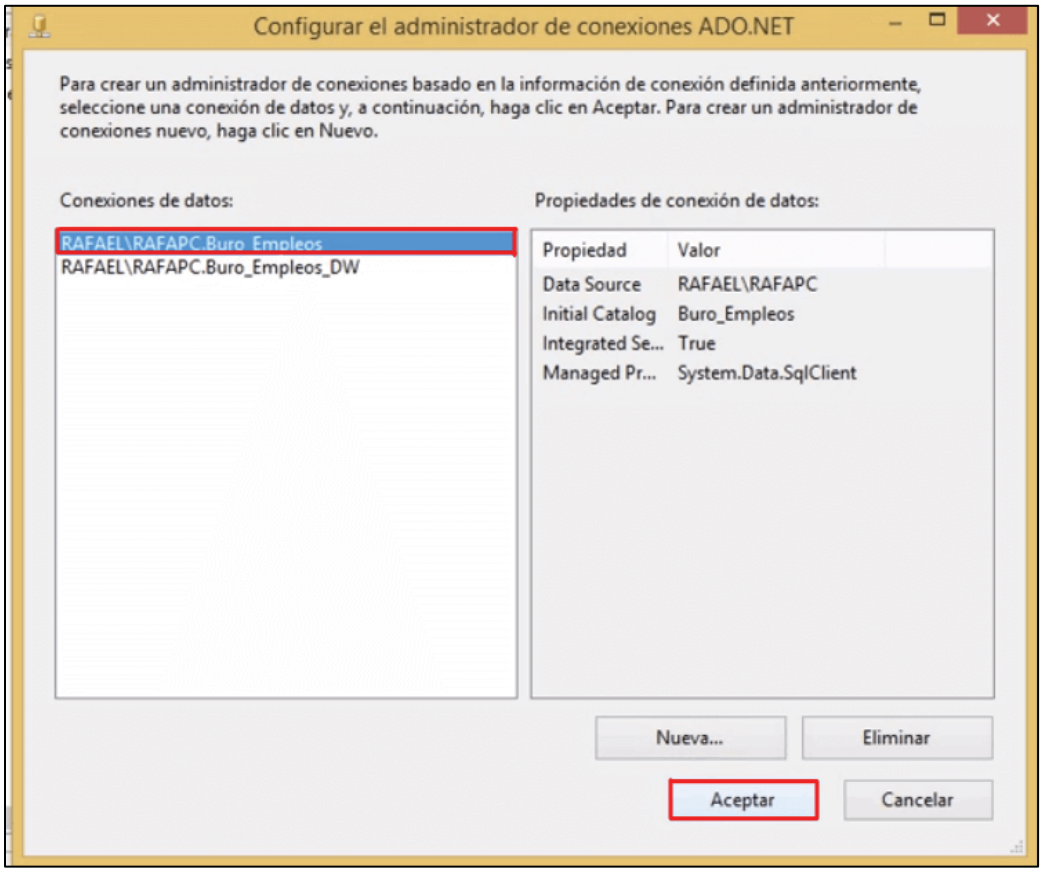
5.- En el cuadro de herramientas, seleccionar un **origen de datos** (Otros orígenes → Origen de ADO NET). En el panel de Flujo de datos, se creará un nuevo elemento, asignar un nombre al origen y luego dar doble click:



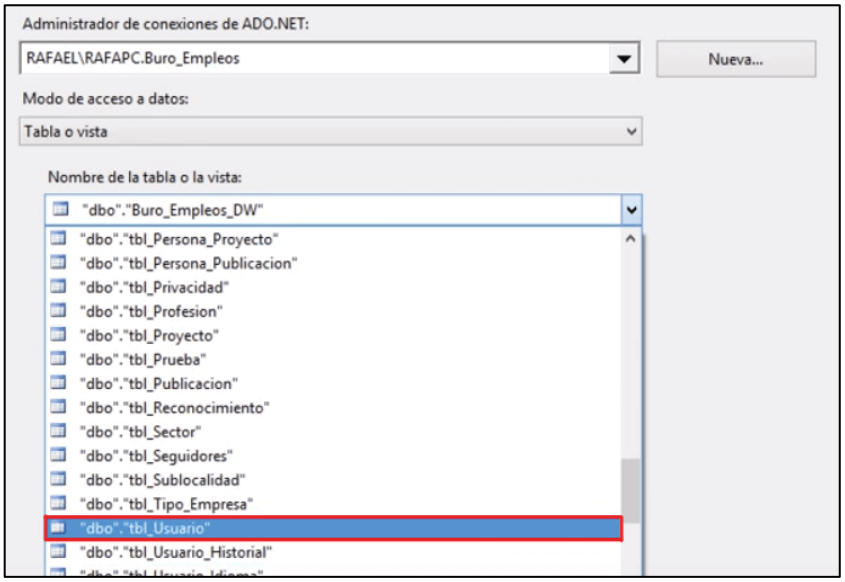
6.- Aparecerá el **editor de orígenes**. Click en **Nueva**:



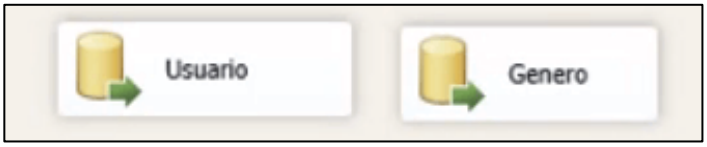
7.- En caso de no tener conexiones, se debe dar click en **Nueva**. En otro caso, seleccionar una conexión existente y luego click en Aceptar:



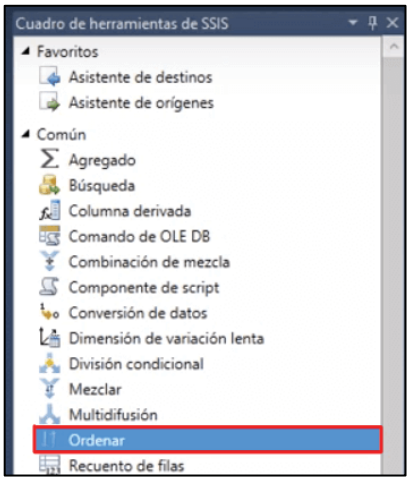
8.- En la sección de “Nombre de la tabla o vista” se debe elegir una tabla de la fuente seleccionada:



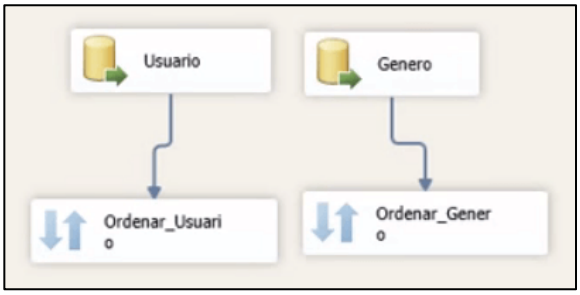
9.- Repetir los pasos 5, 6, 7 y 8 para generar una nueva tabla (Genero).



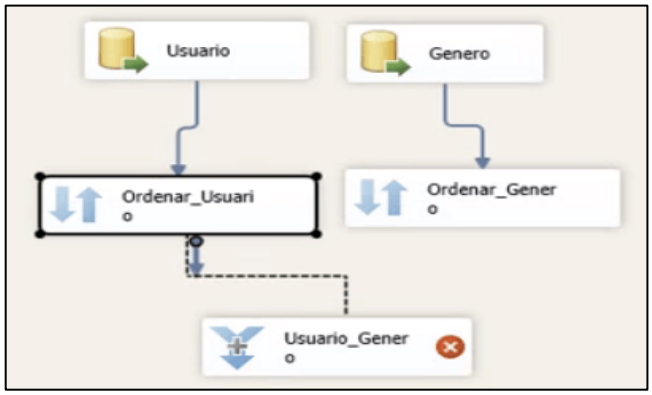
10.- Seleccionar la función de **Ordenar** (Cuadro de herramientas → Común → Ordenar) y conectar una para cada tabla:



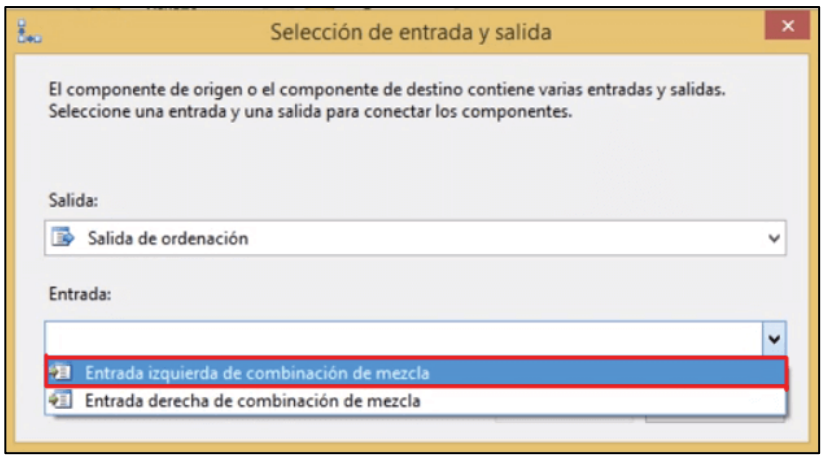
11.- Asignar un nombre para cada **función de ordenación**. Por ejemplo, Ordenar_Usuario para la tabla de Usuario:



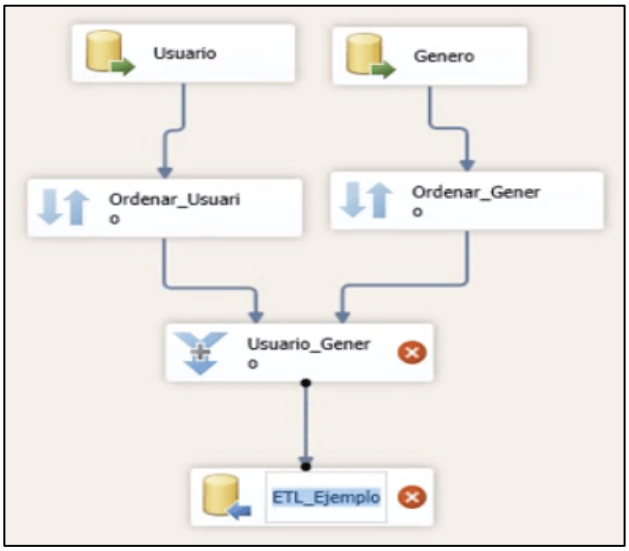
12.- Seleccionar la función de **Combinación de mezcla** (Cuadro de herramientas → Común → Combinación de mezcla) y conectar cada tabla:



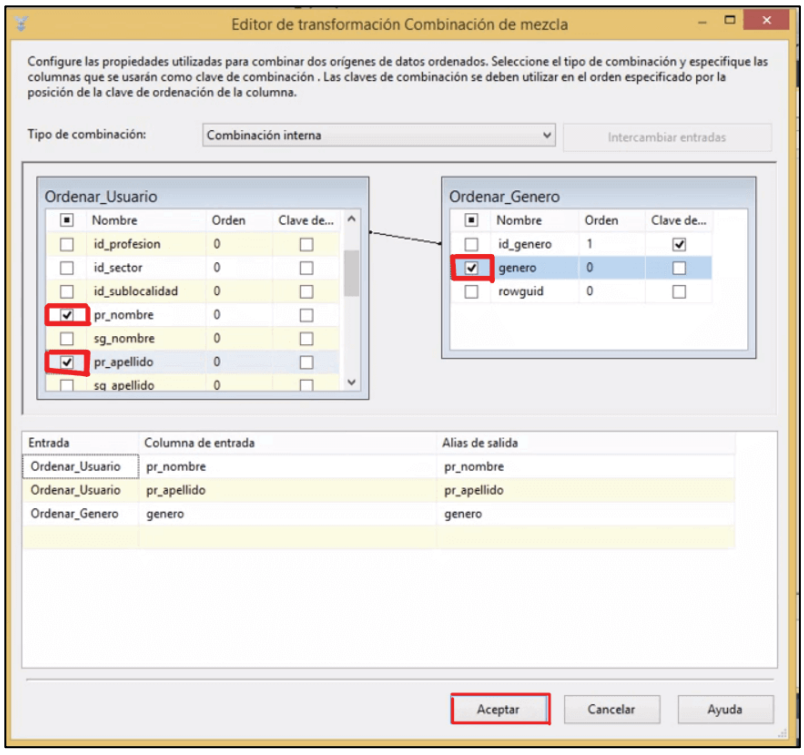
13.- Al conectar la primera tabla, aparece un panel para seleccionar la orientación de la entrada (izquierda o derecha):



14.- Añadir una fuente de salida para los datos (Cuadro de herramientas → Otros destinos → Destino de ADO NET), añadir un nombre y conectar con la función de combinación de mezcla:

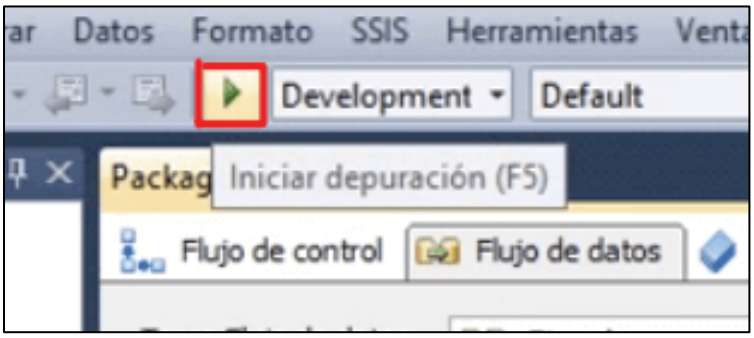


15.- Doble click sobre la combinación de mezcla y luego seleccionar los campos que van a utilizarse de cada tabla:



16.- Al igual que con las tablas “Usuario” y “Genero”, repetir los pasos 5, 6, 7 y 8 para conectar una tabla de salida con el destino (ETL_Ejemplo):

17.- Una vez que la tabla de salida fue asignada, se procede a ejecutar el proceso ETL. Para llevarlo a cabo, presionar F5 ó dar click al boton de “Iniciar depuración”:

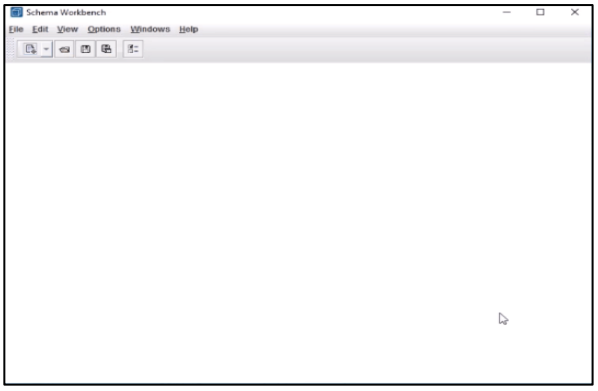


18.- Luego de dar click, se ejecuta el proceso, y aparecerá un mensaje anunciando que la ejecución del paquete se completó correctamente. Las flechas verdes en cada tabla indican que los flujos se llevaron a cabo.

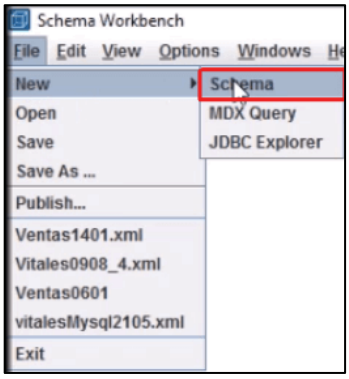
Anexo B: Creación de un cubo OLAP

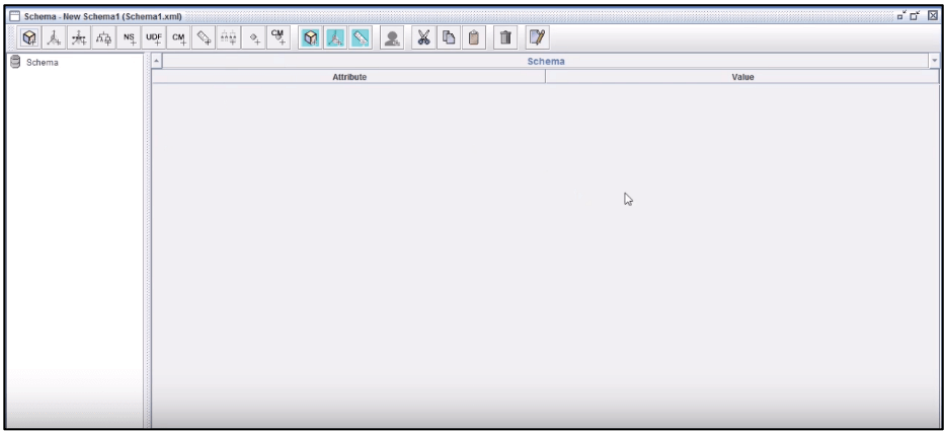
a) Pentaho (Linux)

1.- Primero, ejecutar **Schema Workbench** desde una terminal (cd /ruta donde se instalo Schema Workbench → sh workbench.sh):

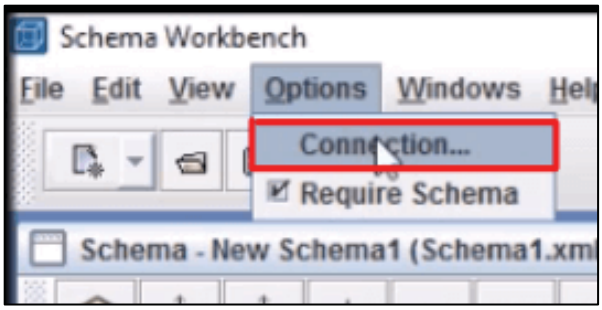


2.- Crear un nuevo **esquema** (File → New → Schema):

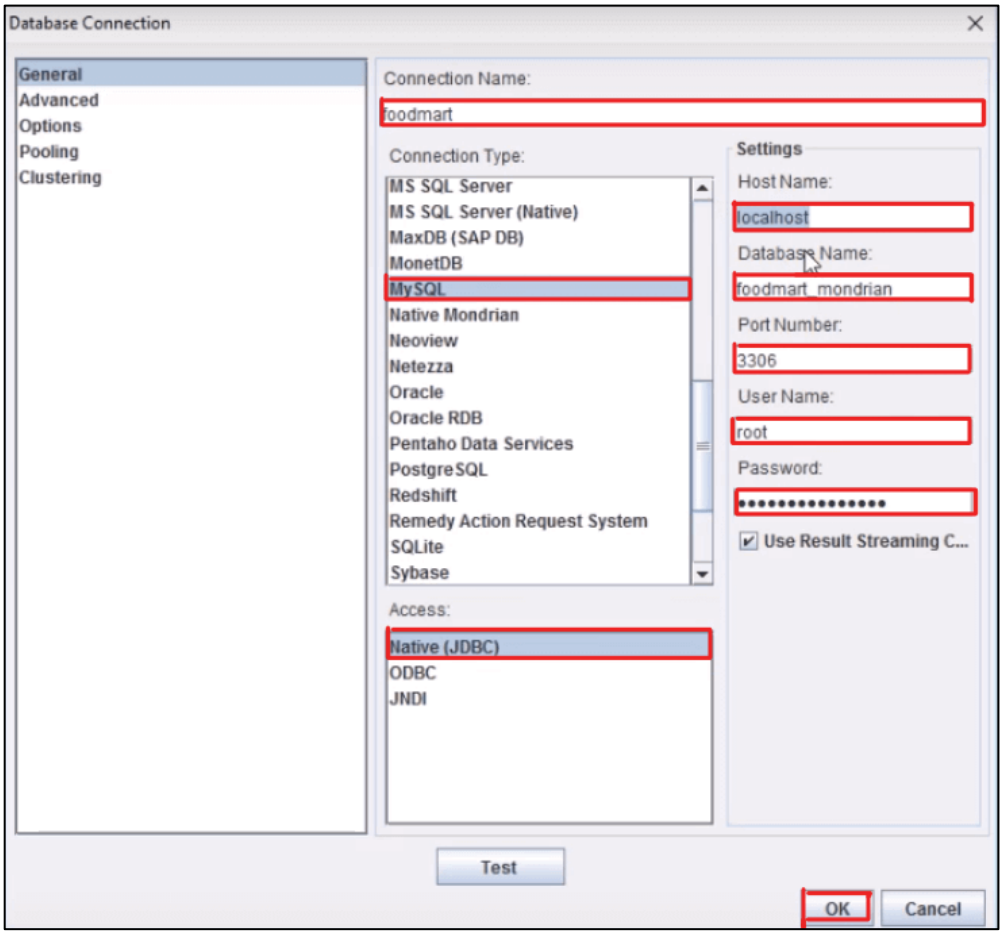




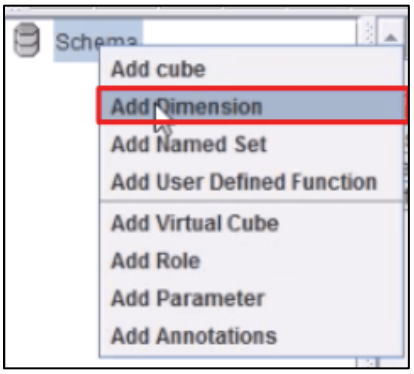
3.- Crear una **conexión** (Options → Connection…):



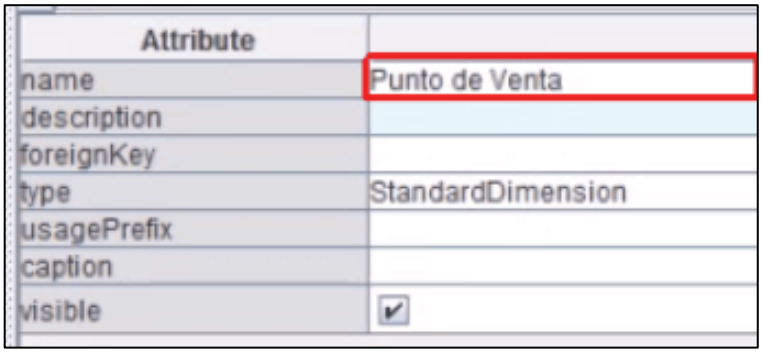
4.- Ingresar los datos correspondientes a la **fuente de datos de entrada.** Una vez que los datos están completos, se puede dar click a "Test" para verificar que la conexión se haya llevado a cabo. Luego dar click en OK.



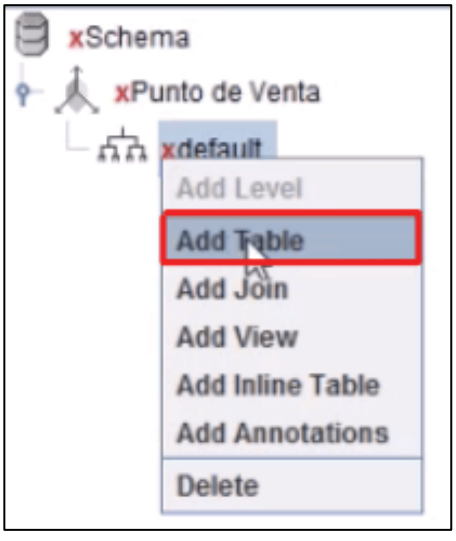
5.- Crear una **dimensión** (Schema → Add Dimensions):



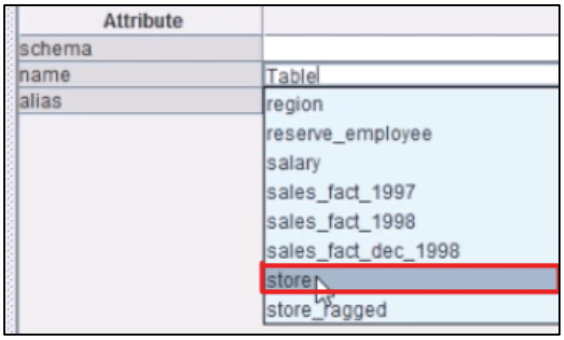
6.- Configurar un **nombre** para la nueva dimensión:



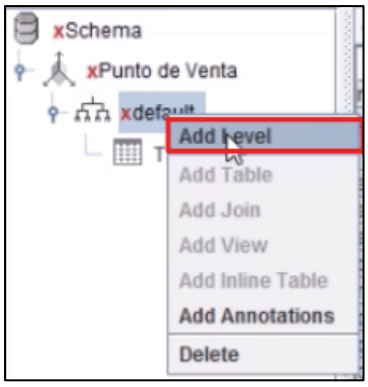
7.- Agregar una **tabla** (default → doble click → Add Table). Al dar click en la dimensión “Punto de venta” se genera automáticamente una jerarquía “default”.



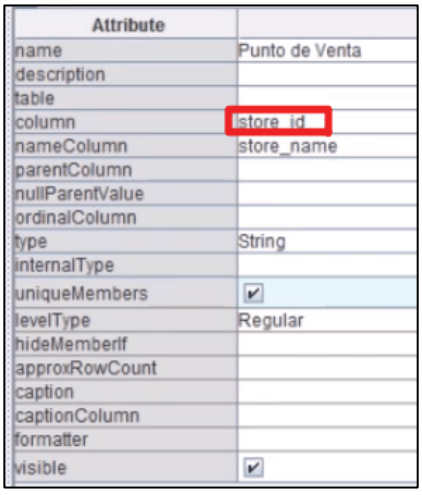
8.- En la sección “name”, al tratar de ingresar un nombre aparecen **todas las tablas** que están en la fuente de datos de entrada. Seleccionar una:



9.- Crear un **nivel** (Click derecho en la jerarquía → Add level):

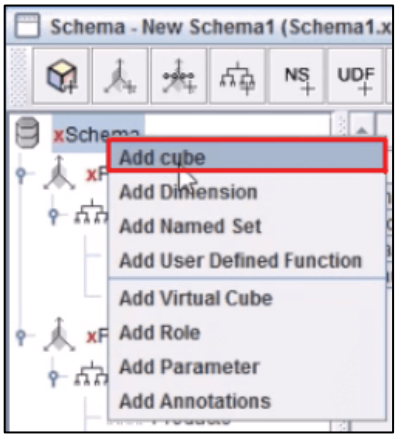


10.- Asignar los valores correspondientes al **nivel**. La sección “column” corresponde al campo de la dimensión que conecta con la tabla de hechos:

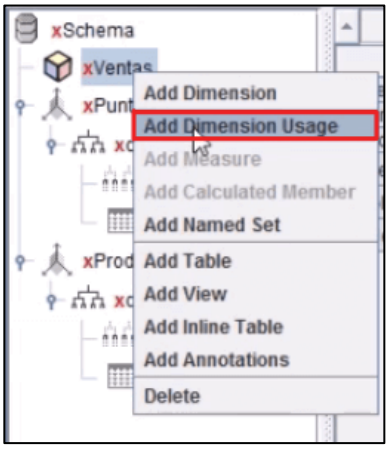


11.- Repetir desde el paso 5 al paso 10 para crear la **dimensión** “Producto”:

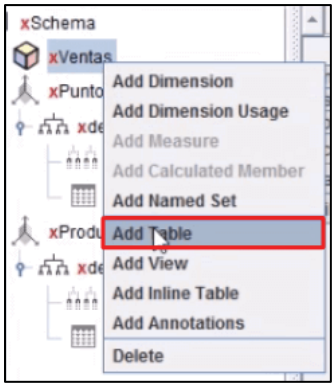
12.- Crear un **cubo** (Schema → Add cube). Asignarle un nombre (Ventas):



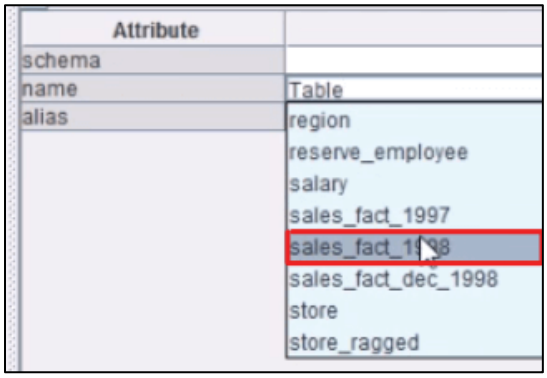
13.- Añadir la **tabla de hecho** (Click en el cubo → Add Table):



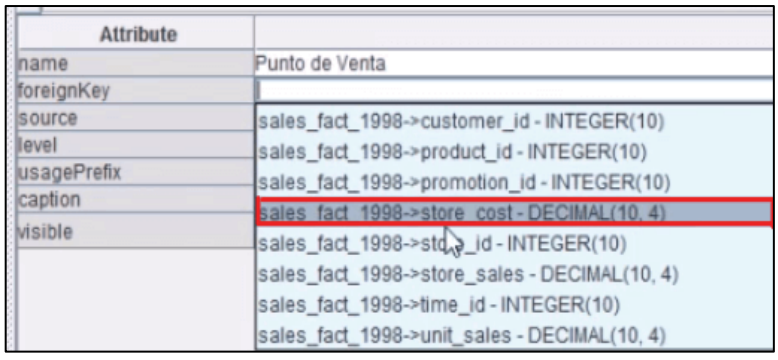
14.- En la sección “name” seleccionar la tabla que será la **tabla de hechos**:



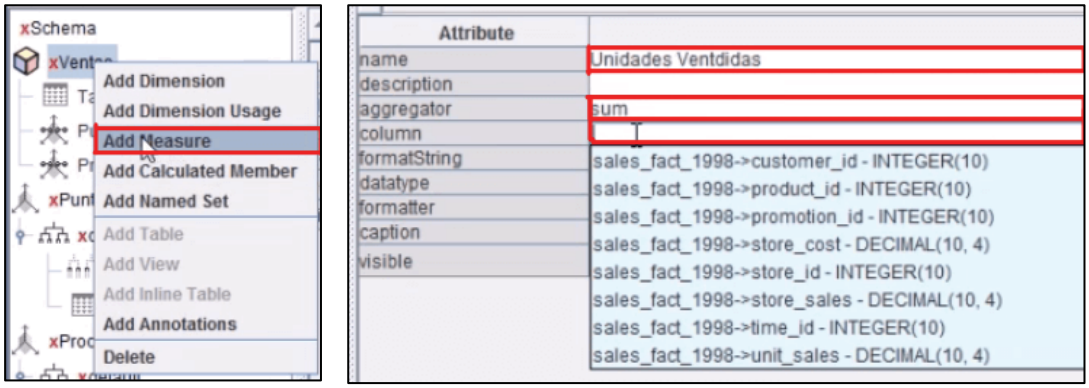
15.- Enlazar las **dimensiones** (Click en el cubo → Add Dimension Usage). La opción de Add Dimension Usage permite usar las dimensiones “locales”. Add Dimension permite crear dimensiones exclusivas para ese cubo.



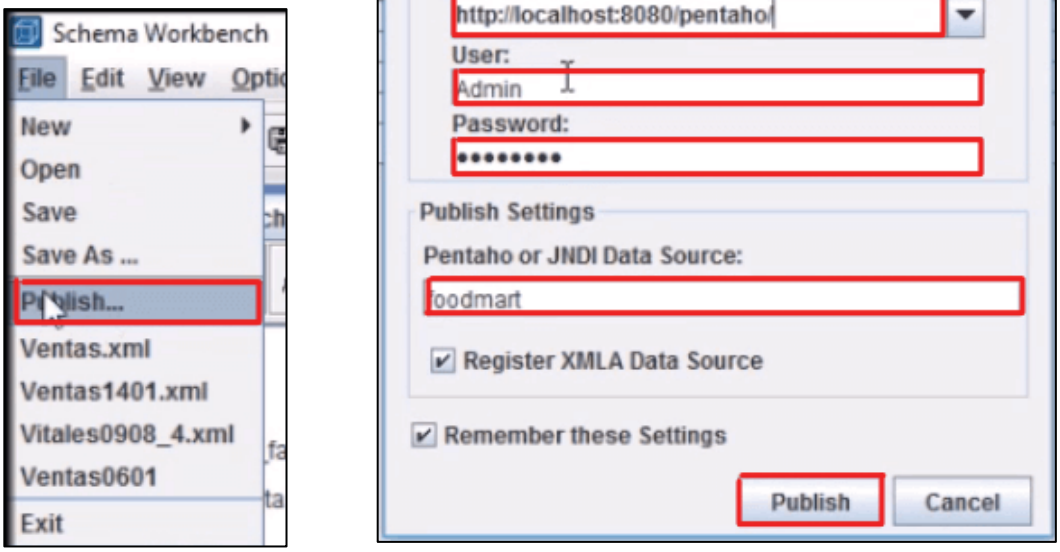
16.- Añadir un nombre (mismo que la dimensión que se va a enlazar) y luego en la sección “foreignKey” seleccionar el campo que corresponde a su **llave primaria**. Repetir el mismo proceso para la dimensión “Producto”:



17.- Hasta el punto anterior, el cubo ya había sido diseñado. Si se desea añadir una medida (Click en el cubo → Add Measure), ingresar el campo que se va a medir en la sección “column”, añadir un nombre (sección “name”) y luego elegir el tipo de medida en la sección “aggregator”:

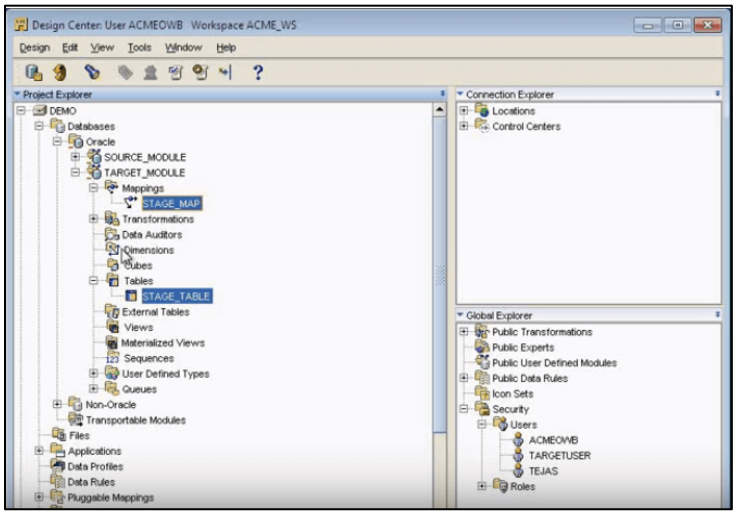


18.- Para finalizar, **publicar el cubo** para posteriormente analizar la información desde el servidor (File → Publish). Luego en la ventana de Publish Schema ingresar los datos de la conexión hacia el servido y luego click en Publish:

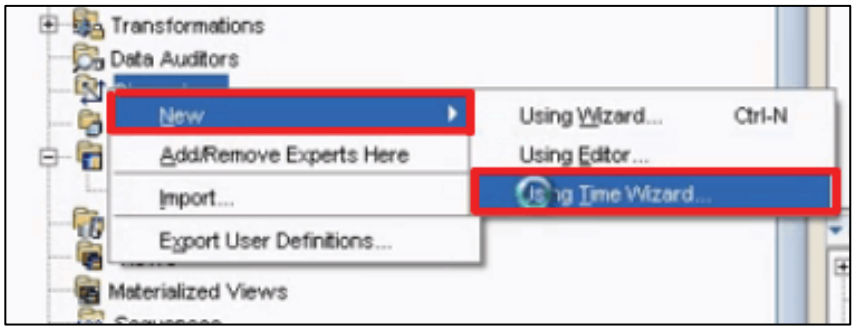


b) Oracle (Windows)

1.- Ejecutar **Oracle Warehouse Builder** (Inicio → Oracle-OraDbg_home1 → Warehouse Builder → Design Center). En caso de requerir crear una fuente de entrada, véase el Anexo A:

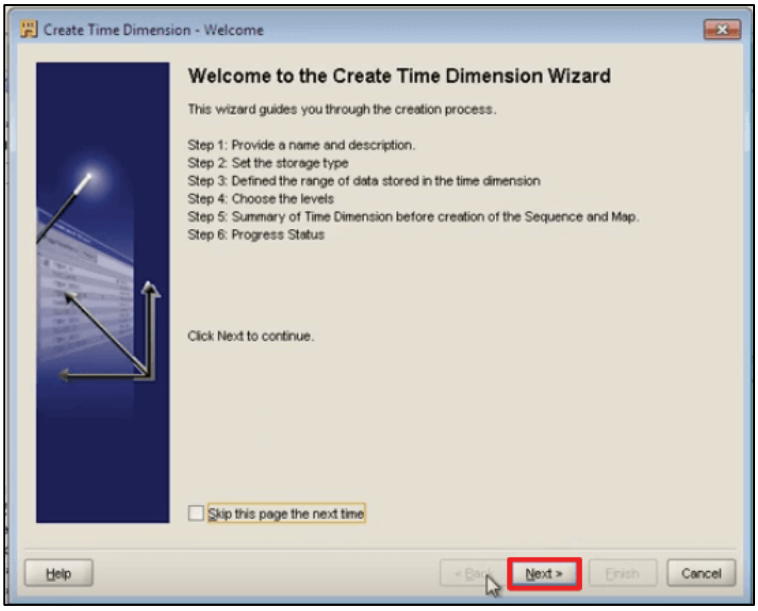


2.- Crear una **dimensión** ("Nombre del proyecto" → Databases → Oracle → TARGET_MODULE → Dimensions → Click derecho → New → Using Time Wizard):

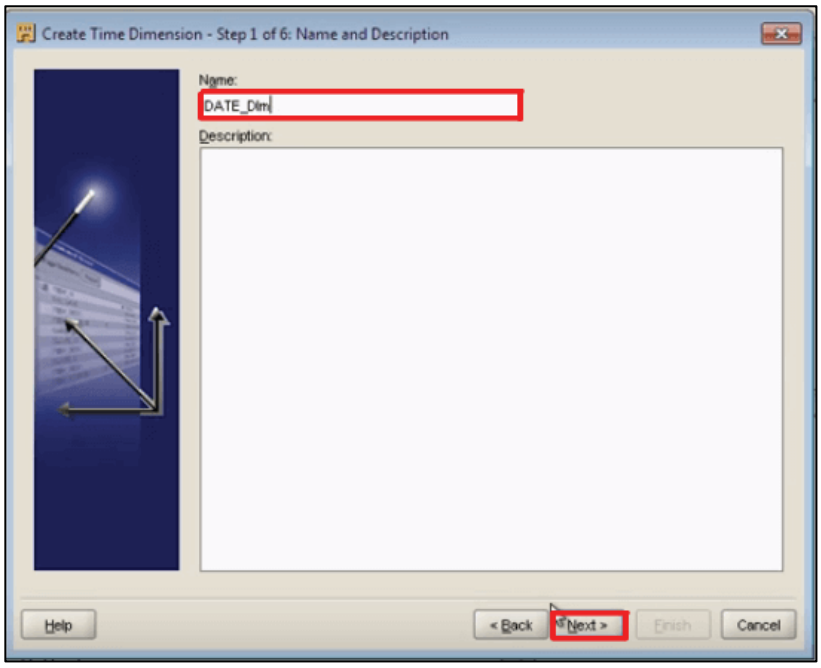


Oracle Utiliza un asistente personalizado para crear diferentes propiedades. En el caso de las dimensiones, **Using Time Wizard** permitirá construir una dimensión de tiempo con la ayuda de su asistente. La opción de **Using Wizard** permite crear otras dimensiones.

3.- Aparecerá la ventana del **asistente**. Click en Next:

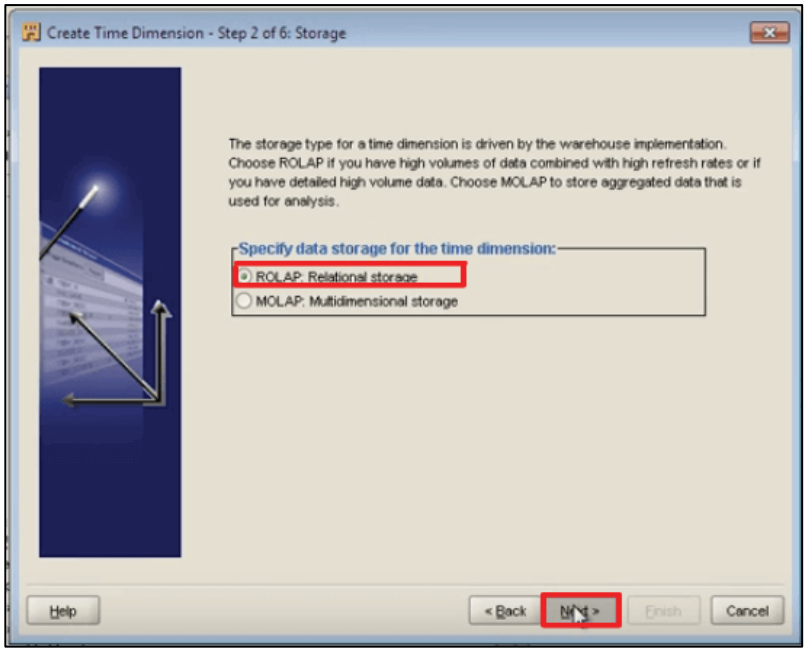


4.- Asignar un nombre a la dimensión. Click en Next:

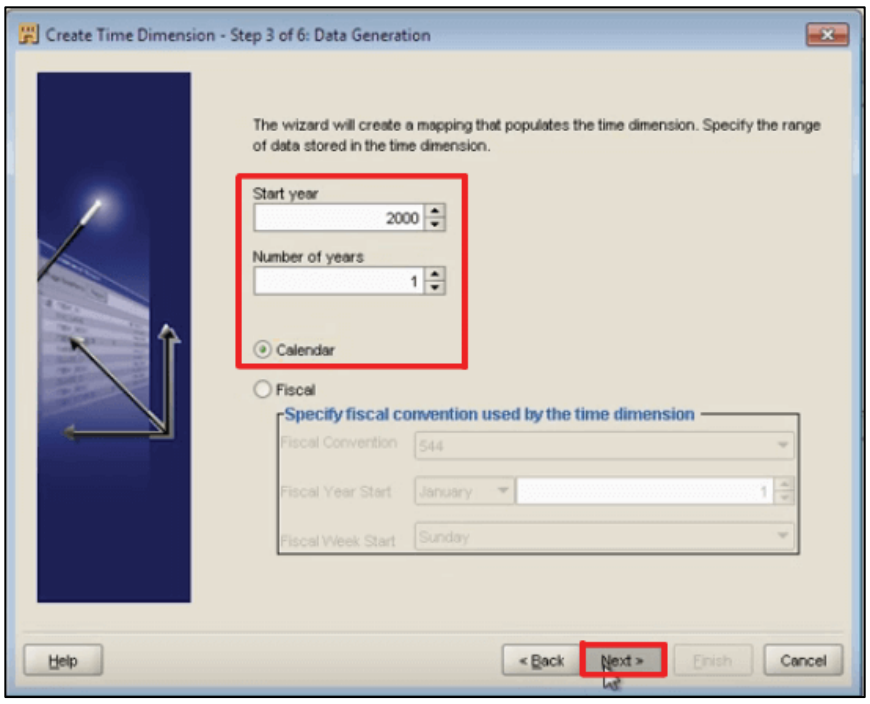


El siguiente paso se repite en la creación de varios elementos. Por lo tanto se omitirá en los casos posteriores. Aparece una ventana preguntando que tipo de almacenamiento requiere el usuario.

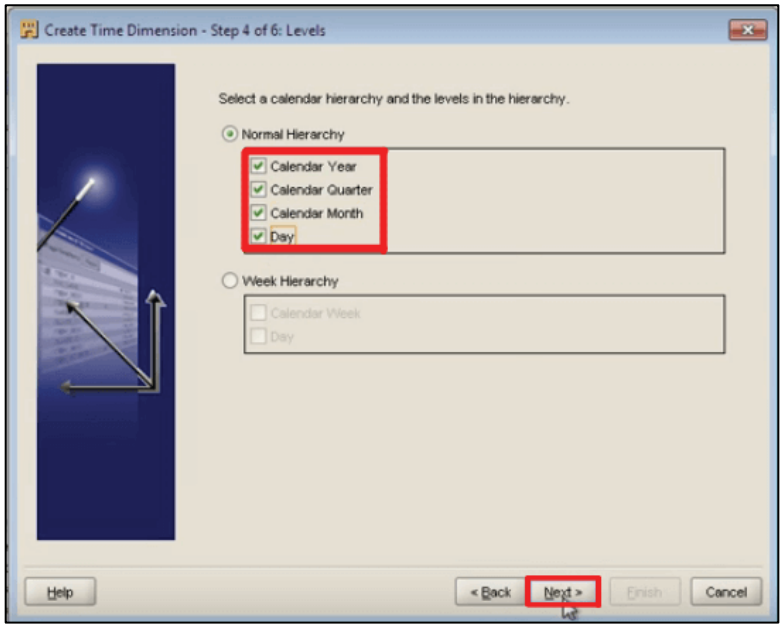
5.- Seleccionar el **tipo de almacenamiento** (Rolap):



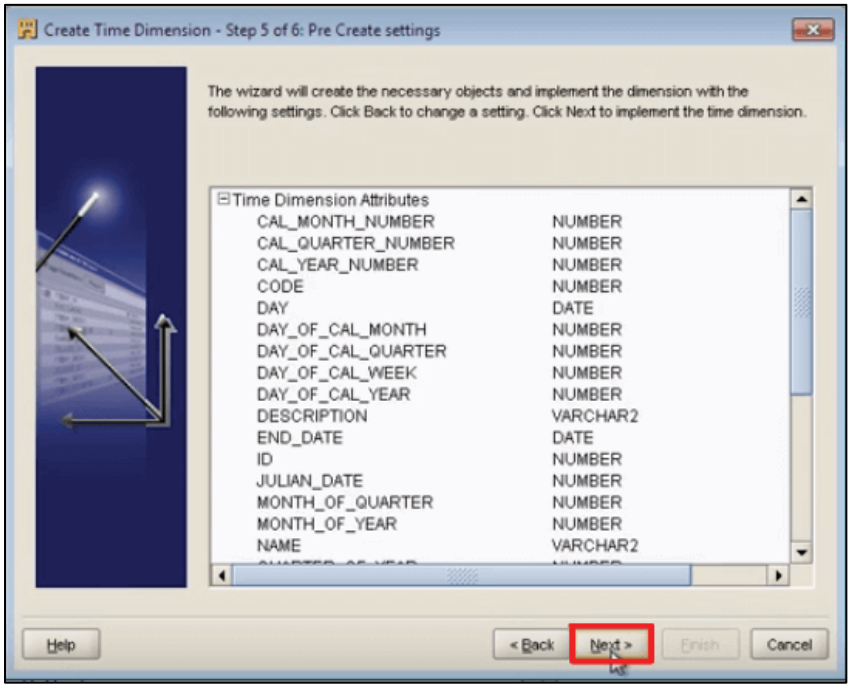
6.- Seleccionar las **propiedades** de la dimensión de tiempo. Por convención es mas usual el tipo de fecha “Calendar”. Por omisión el año inicial es 2000, pero el usuario puede cambiarlo según las necesidades de negocio. Click en Next:



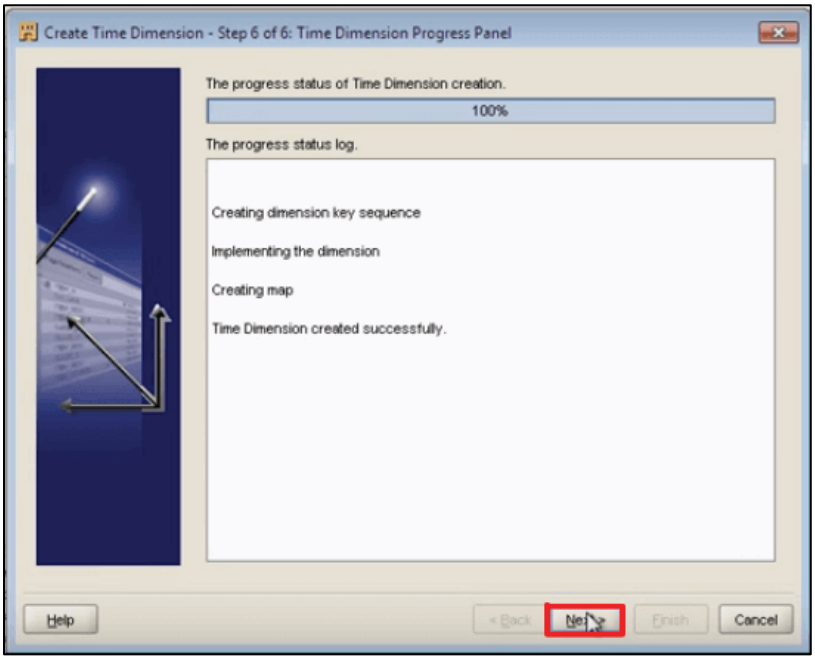
7.- Seleccionar la jerarquía y los niveles. Por omisión no están seleccionados, así que el usuario debe elegirlos. Luego dar click en Next:



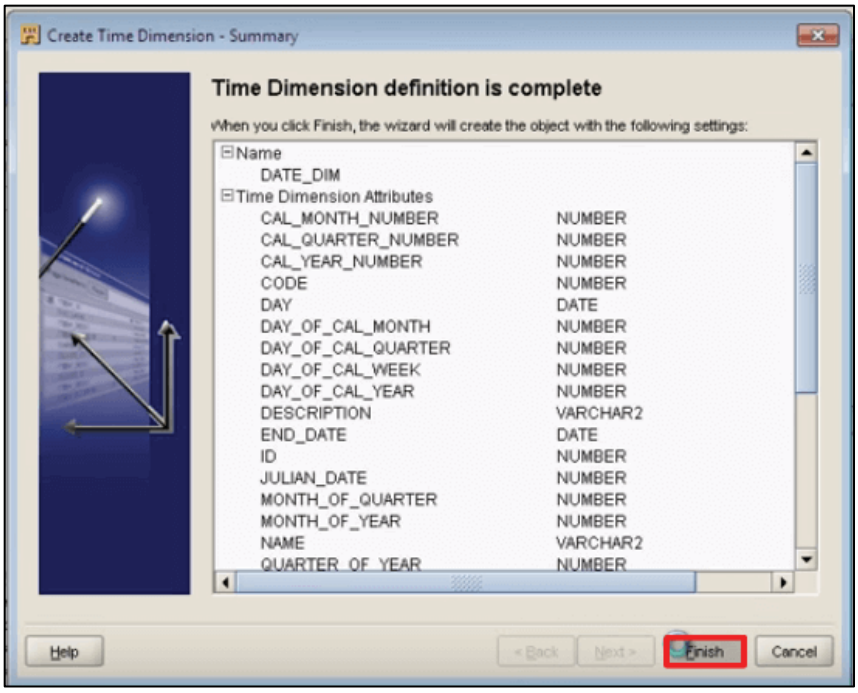
8.- Aparece una nueva venta con la **visualización previa** de la dimensión que el usuario esta creando. Este paso también se omitirá en las siguientes muestras del proceso. Click en Next:



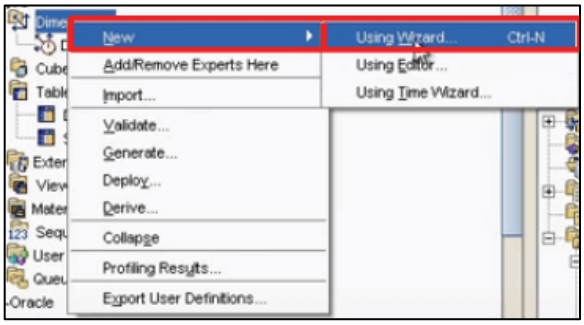
9.- Tardará un momento en generar la dimensión. Cuando la barra de progreso muestre el 100 por ciento, dar click en Next:



10.- Se muestra nuevamente una ventana con el resumen de la dimensión recién creada. Click en Finish:

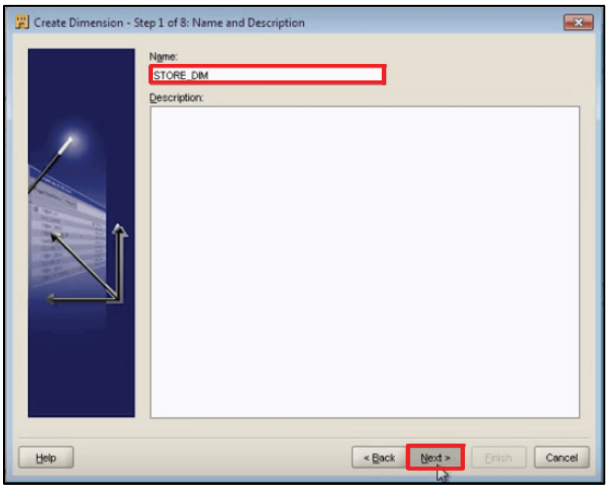


11.- Crear una **dimensión** (“Nombre del proyecto” → Databases → Oracle → TARGET_MODULE → Dimensions → Click derecho → New → Using Wizard):

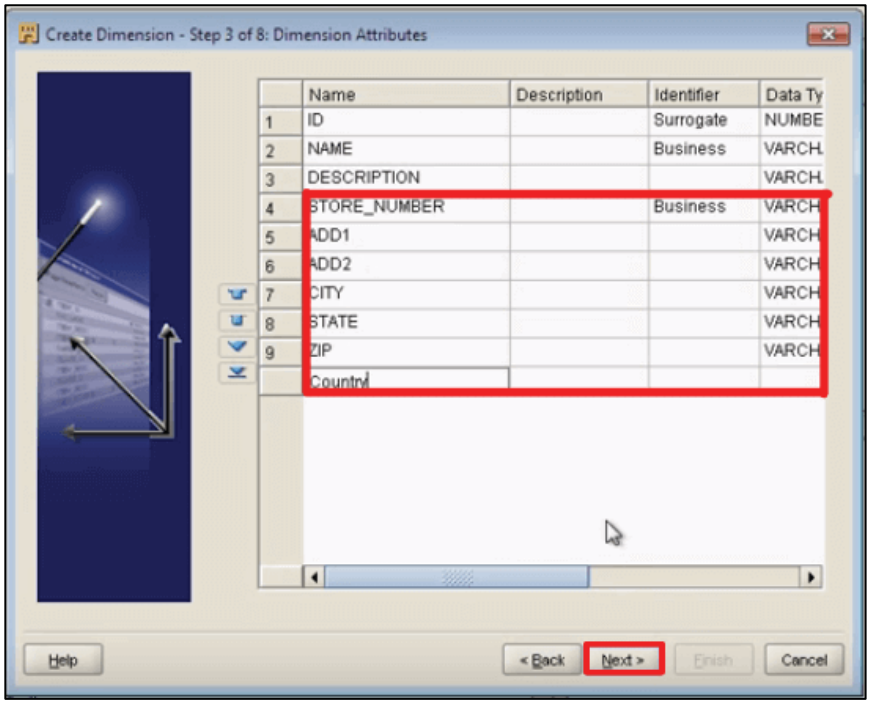


12.- Aparece la ventana del asistente. Click en Next:

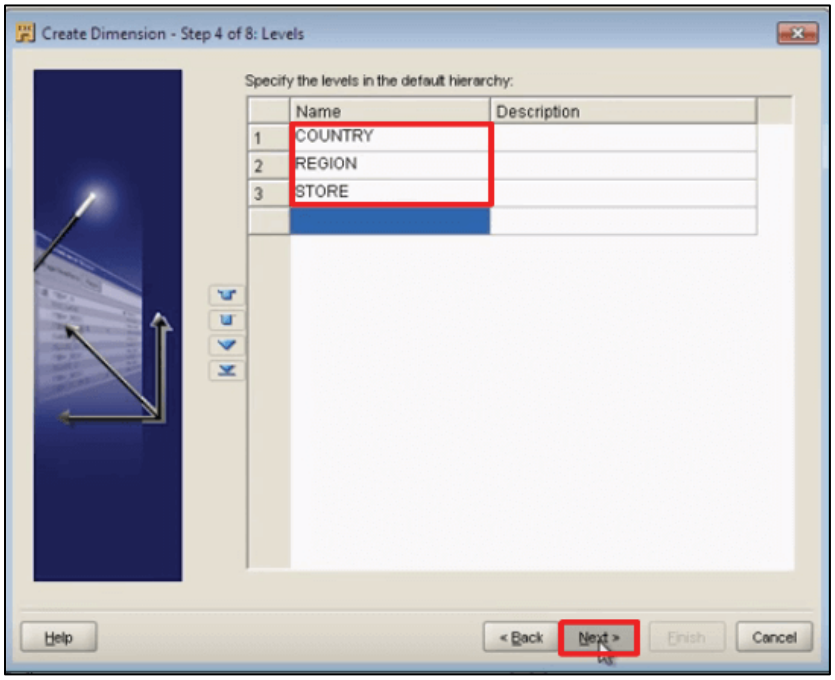
13.- Asignar **nombre** a la dimensión (STORE_DIM). Click en Next:



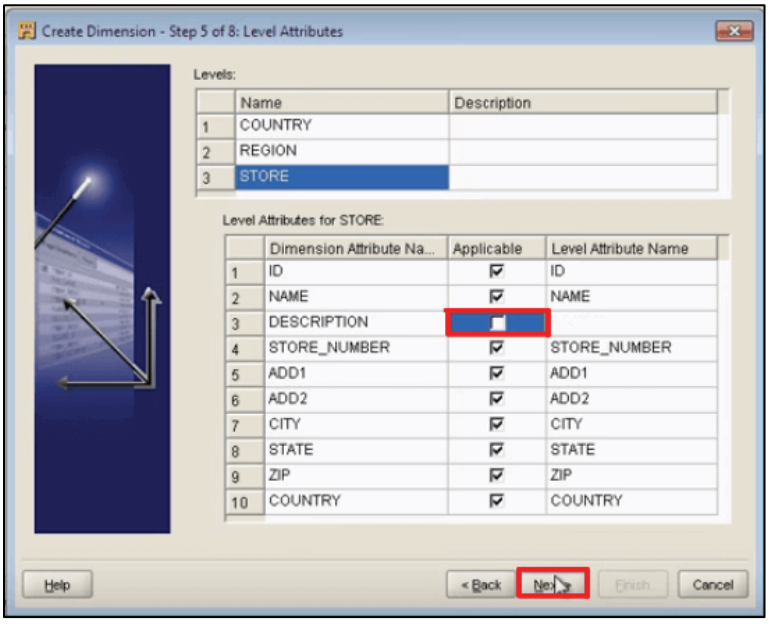
14.- Aparece la tabla de **atributos de la dimensión.** Por omisión tiene 3 atributos (ID, NAME, DESCRIPTION), añadir los correspondientes a la dimensión:



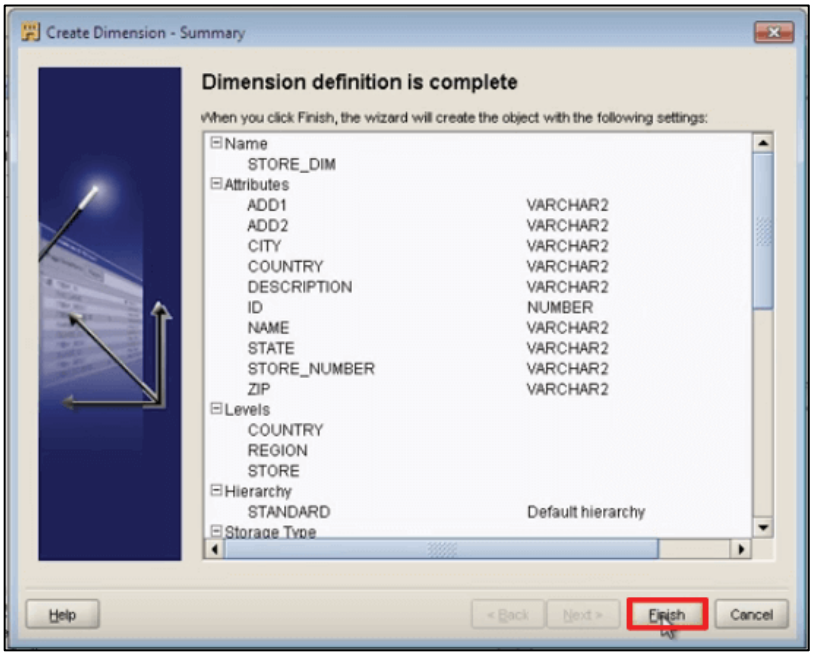
15.- Indicar los **niveles de la jerarquía**, luego click en Next:



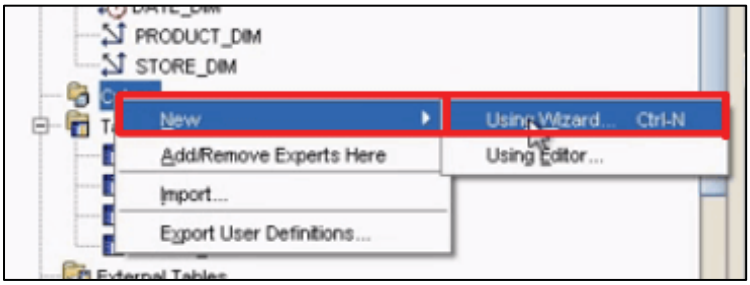
16.- Seleccionar los **attributos del nivel.** Todos están seleccionados por omisión, así que si no se requieren, se deben remover:



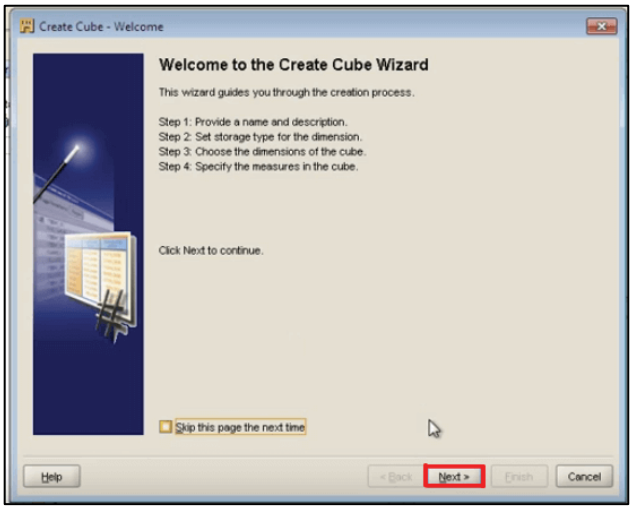
17.- Posteriormente, se mostrará la vista previa y la pantalla del progreso de carga de la dimensión. Luego aparece la ventana del **resumen de la definición** de la dimensión. Dar click en Finish:

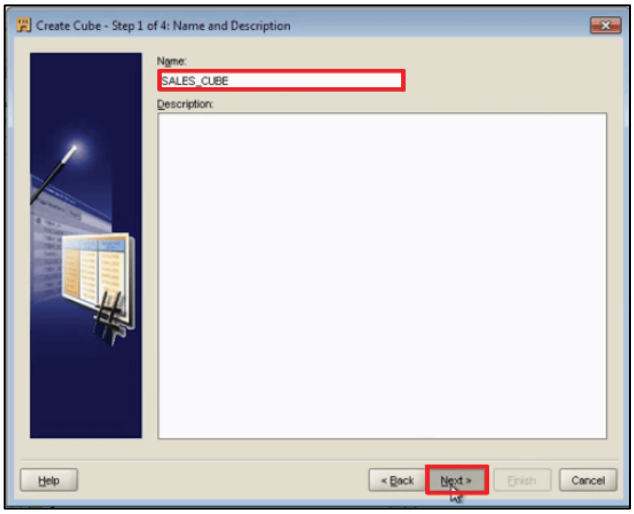


18.- Una vez que se han creado las dimensiones necesarias, se procede a crear el cubo (“Nombre del proyecto” → Databases → Oracle → TARGET_MODULE → Cubes → Click derecho → New → Using Wizard):

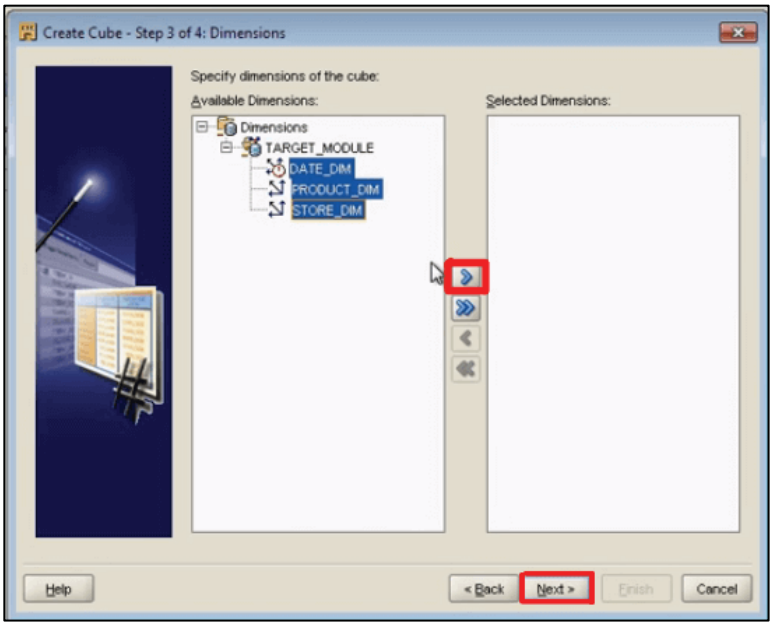


19.- Una nueva ventana aparece, dar click en Next y luego asignar un **nombre** al cubo en la siguiente ventana:

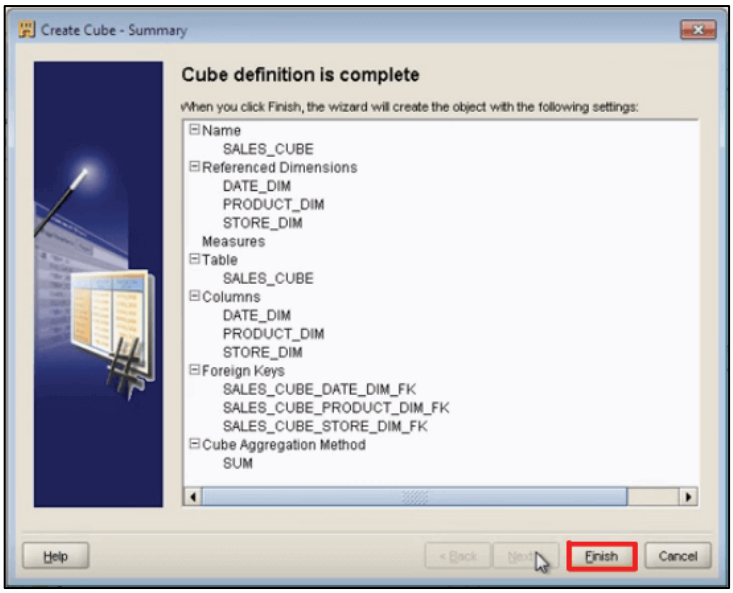




20.- Luego de seleccionar el tipo de almacenamiento, aparece esta nueva ventana, donde se muestran las **dimensiones creadas disponibles**. Al dar click en la flecha enmarcada con rojo, las dimensiones se preparan para ser cargadas al cubo. Click en Next:

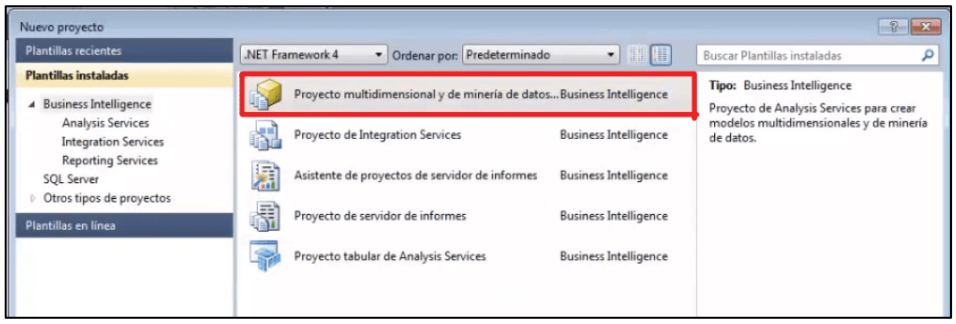


21.- Una vez cargadas las dimensiones, se muestra el resumen del cubo creado completamente. Click en Finish:

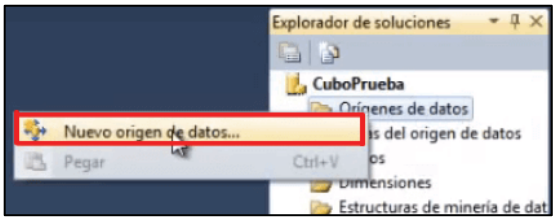


c) SQL Server (Windows)

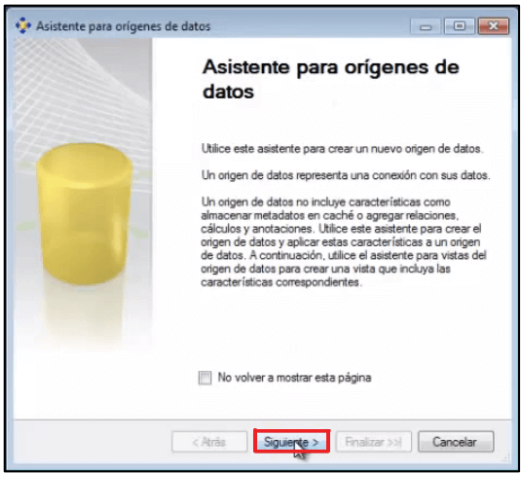
1.- Ejecutar Visual Studio 2010 (Inicio → Microsoft Visual Studio 2010 → Microsoft Visual Studio 2010). Al seleccionar nuevo proyecto, elegir la opción de "Proyecto multidimensional y de minería de datos". Asignar un nombre al proyecto y elegir donde se guardará:



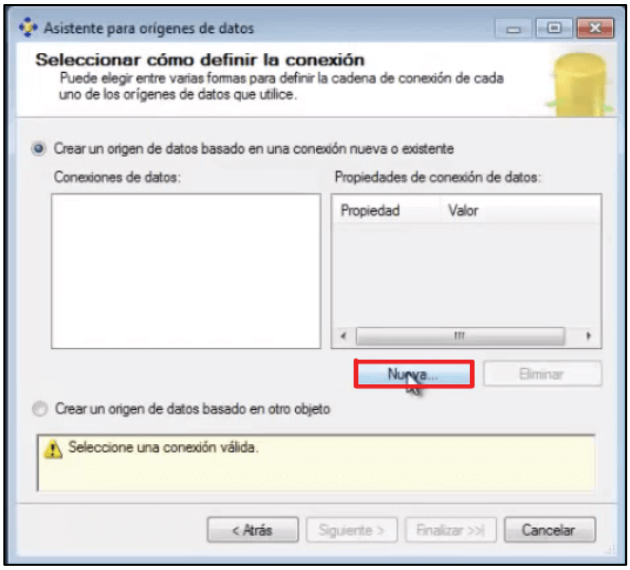
2.- Seleccionar un **origen de datos** ("Nombre del proyecto" → Origenes de datos → Nuevo origen de datos):



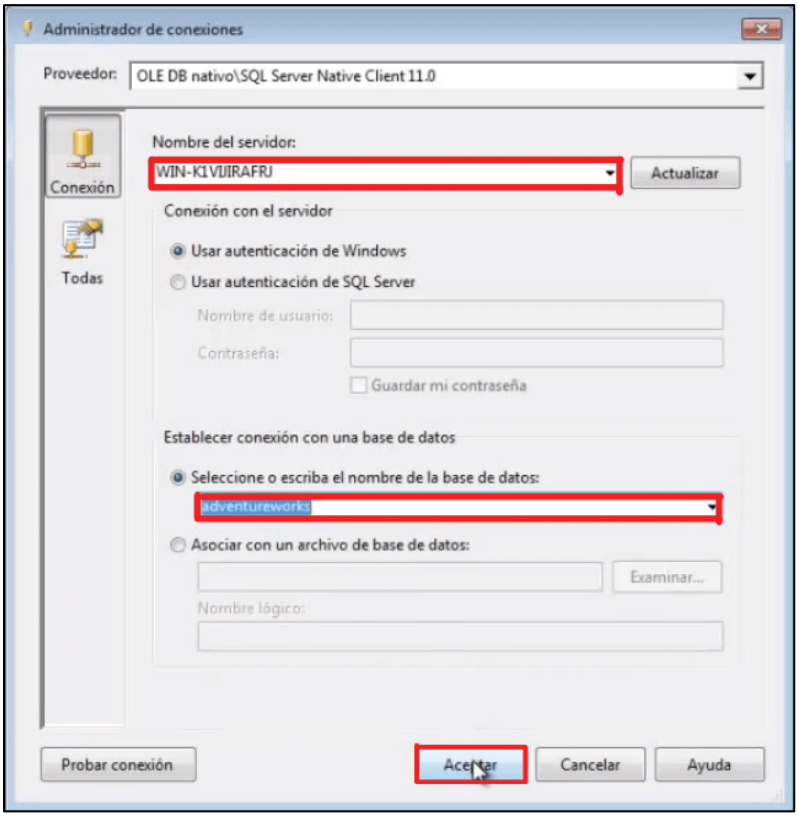
3.- Aparece el **asistente para orígenes** de datos. Click en siguiente:



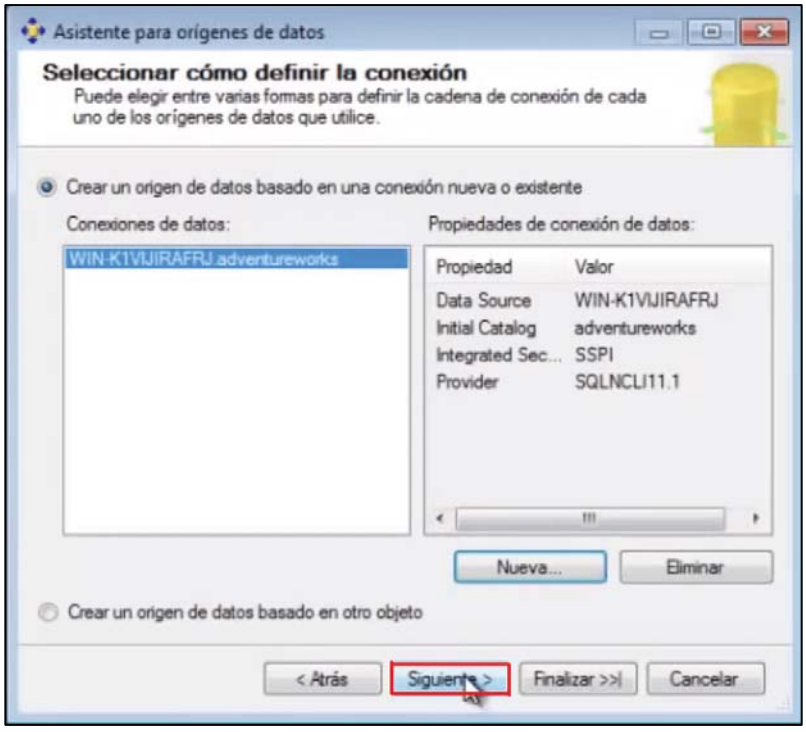
4.- Crear una nueva **conexión de datos.** Click en Nueva:



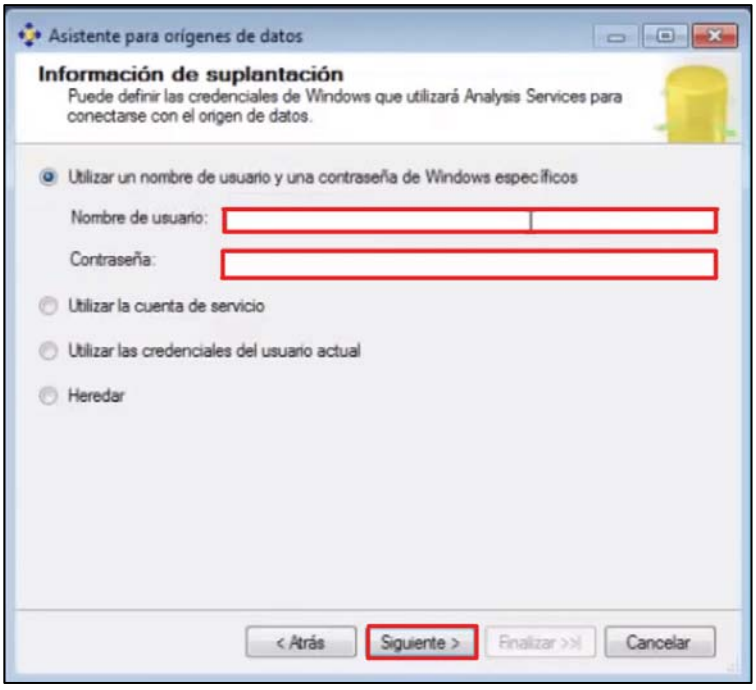
5.- Ingresar el nombre del servidor y el nombre de la base de datos. Luego dar click en Aceptar:



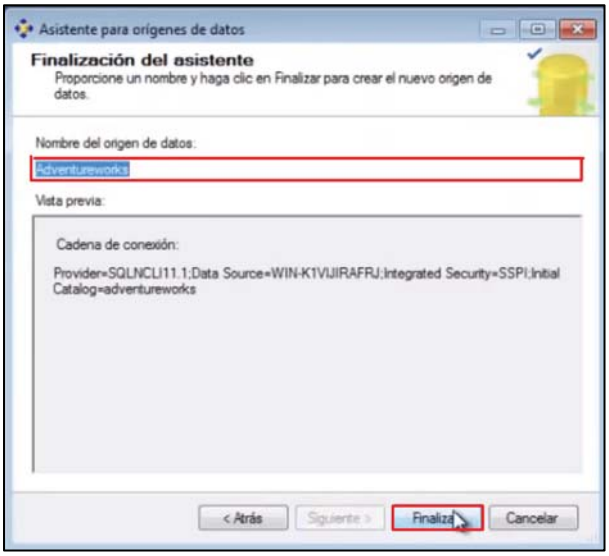
6.- Una vez configurada la conexión, dar click en Siguiente:



7.- El usuario debe ingresar sus datos personales (Nombre de usuario y contraseña) para continuar. Una vez ingresados, click en Siguiente:



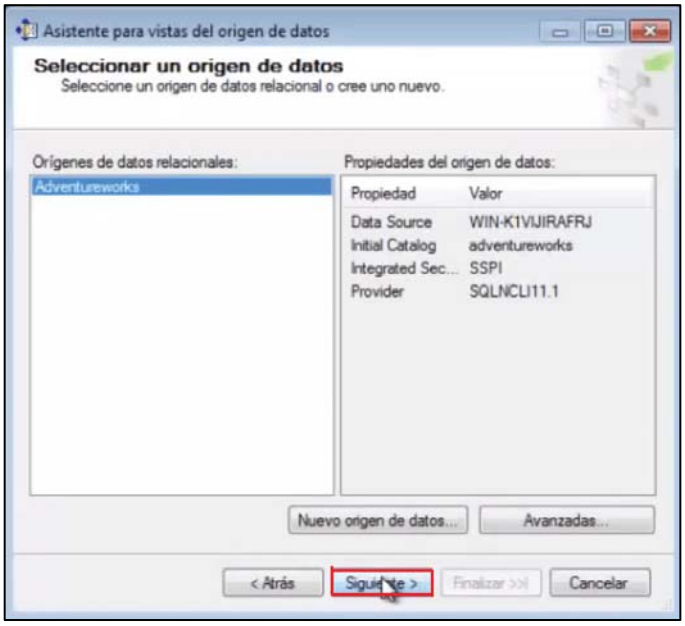
8.- Asignar un nombre al origen de datos (Adventureworks) y luego dar click en Finalizar:



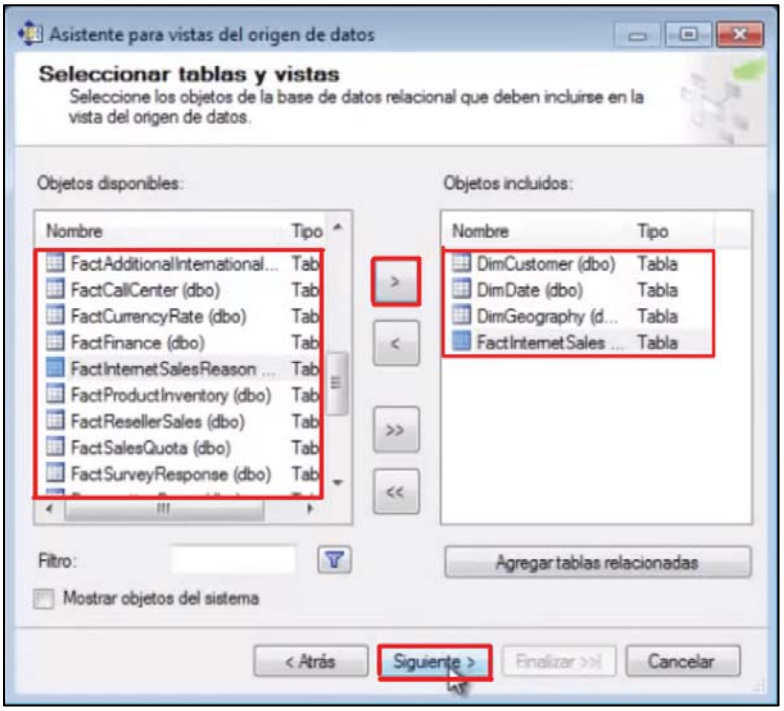
9.- Crear las **vistas de datos** (“Nombre del proyecto” → Vistas del origen de datos → Nueva vista del origen de datos). Luego dar click en Siguiente:

10.- Aparecerá el origen de datos recién creado. Click en Siguiente:

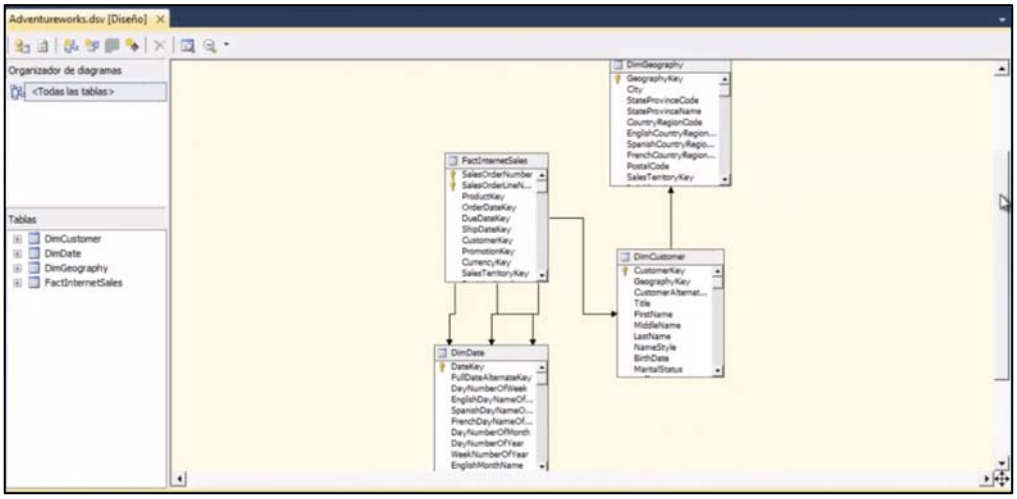
11.- En el lado izquierdo de la ventana aparecerán todas las **tablas del origen de datos**. Seleccionar cada tabla requerida y dar click en la flecha que apunta a la derecha (enmarcada con rojo) para incluir cada tabla en las vistas. Luego de seleccionar las necesarias, dar click en Siguiente:



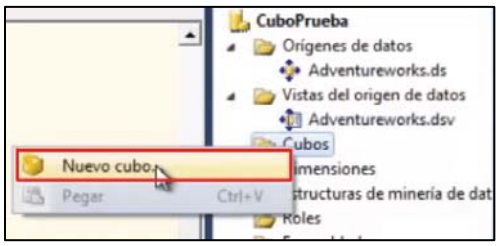
12.- Se muestra una ventana con las **vistas** previas a crearse. Dar click en Finalizar:

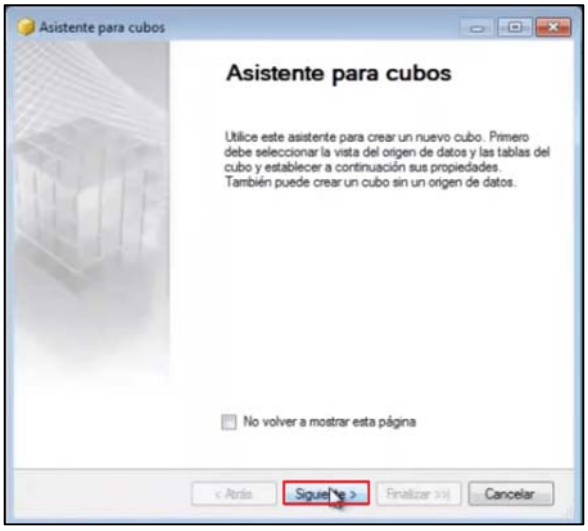


13.- Ahora se han generado las vistas de las tablas. Se muestra físicamente cómo está construido el modelo relacional con las tablas seleccionadas. Ahora que se cuenta con las vistas, es momento de proceder a crear el cubo:

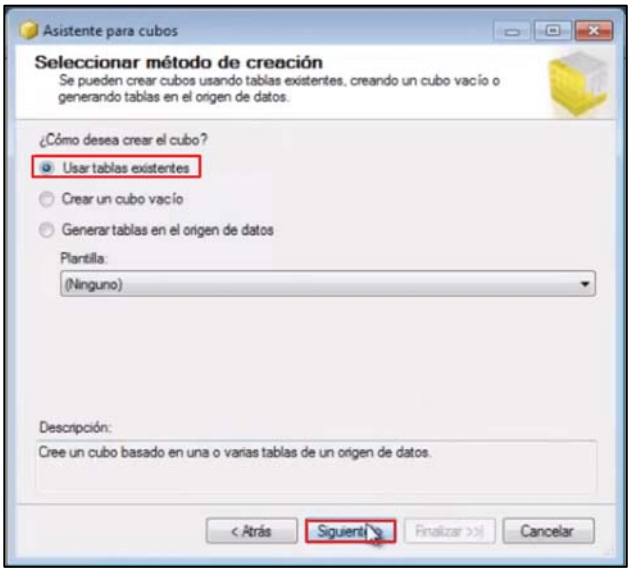


14.- Crear el **cubo** (“Nombre del proyecto” → Cubos → Nuevo cubo). Una vez que se muestre el asistente, dar click en Siguiente:

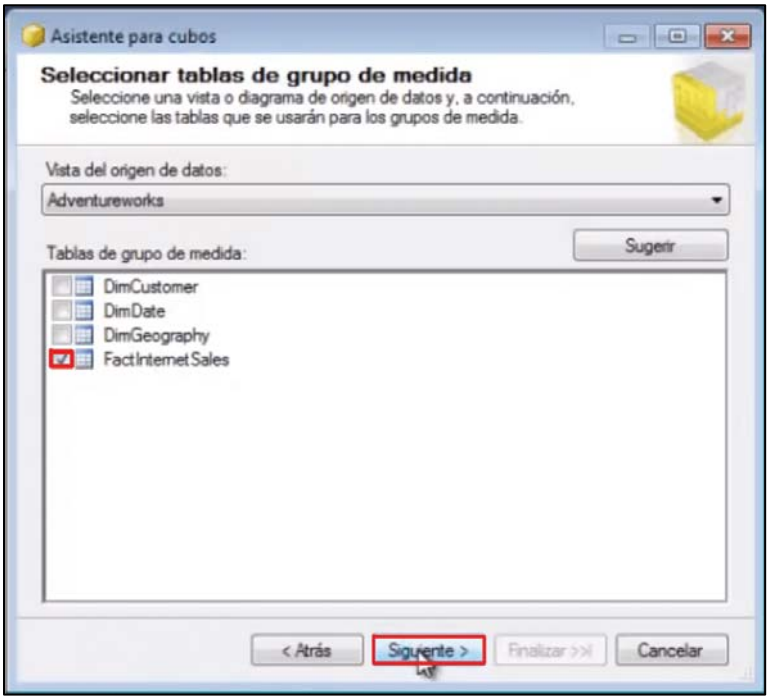




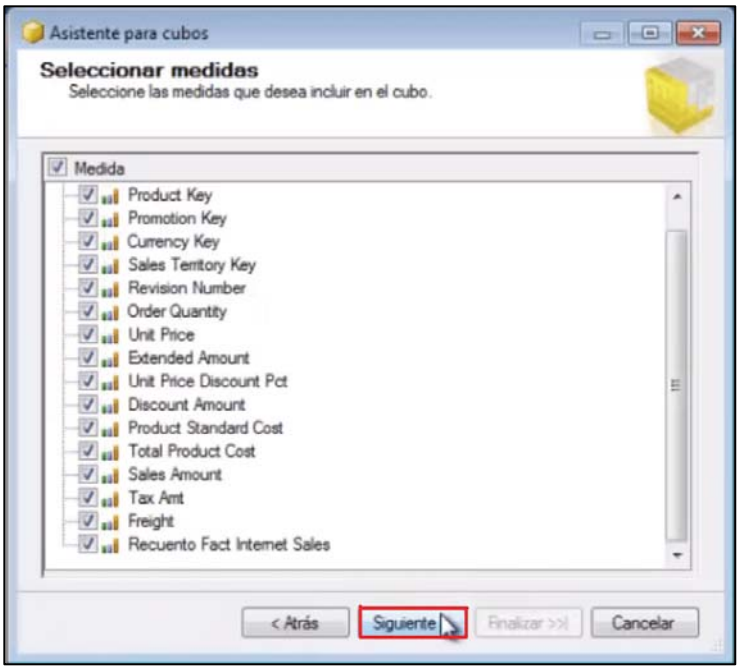
15.- Seleccionar un **modo de creación**. En este caso, seleccionar “Usar tablas existentes”. Luego dar click en Siguiente:



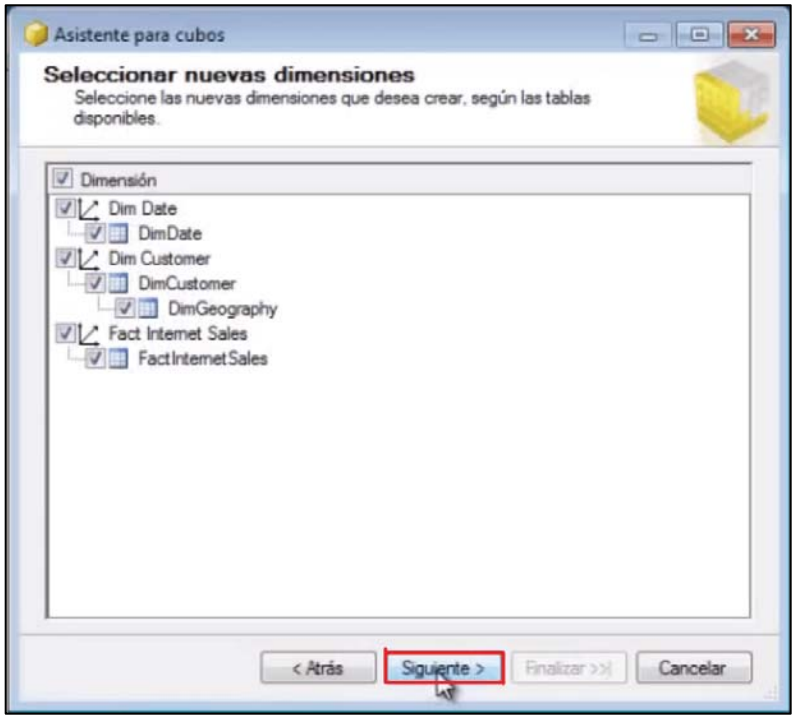
16.- Seleccionar la **tabla de hechos** (tabla de grupo de medida). Dar click en la tabla y luego dar click en Siguiente:



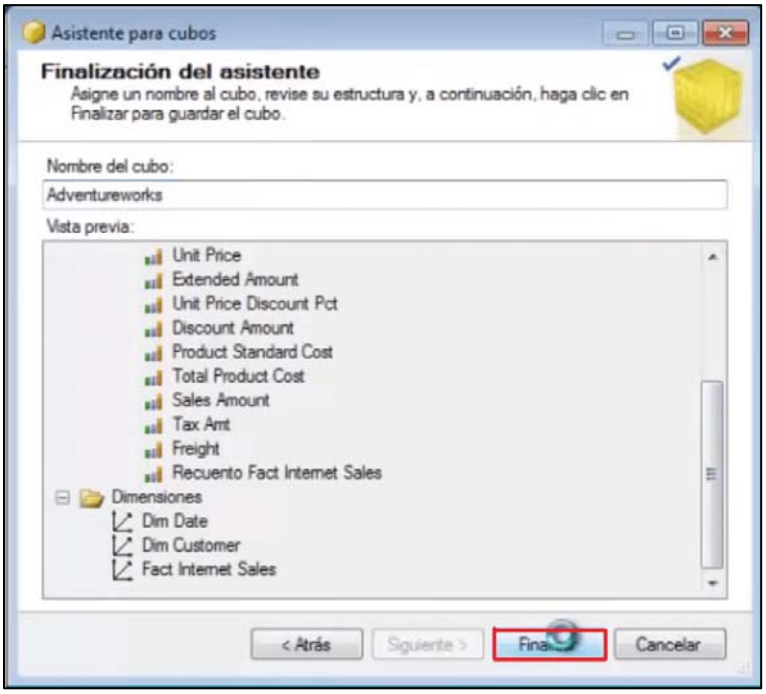
17.- Seleccionar las **medidas**. Por omisión, todas están seleccionadas. Luego de elegirlas, dar click en Siguiente:



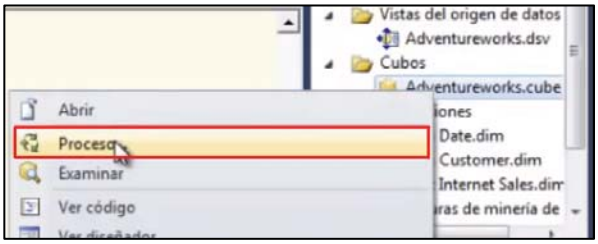
18.- Seleccionar las **dimensiones**. Por omisión, todas están seleccionadas. Luego de elegir las necesarias, dar click en Siguiente:



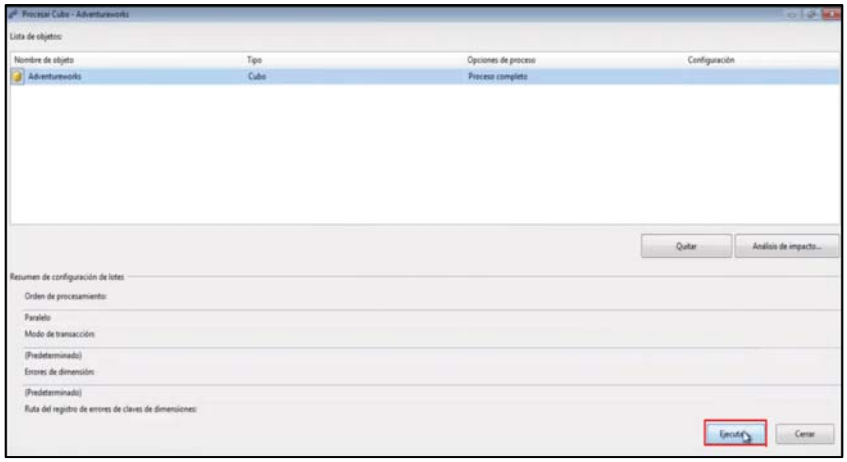
19.- Se mostrará un resumen del contenido del cubo. Dar click en Finalizar:

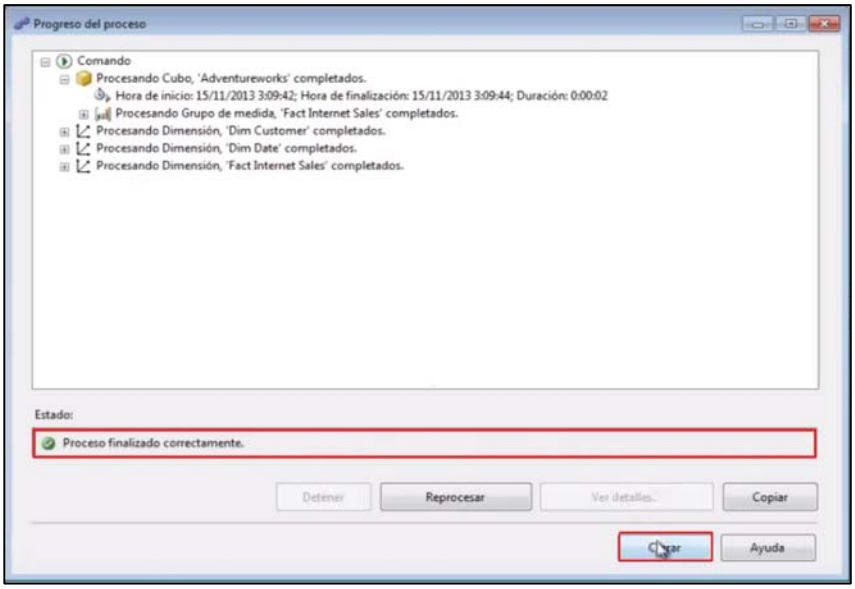


20.- Ahora solo es necesario **procesar el cubo** (“Nombre del proyecto” → Cubos → “Nombre del cubo” → Proceso).



21.- Aparecerá otra ventana, dar click en Ejecutar (enmarcado con rojo). Después se muestra una ultima ventana con el progreso del cubo. Al finalizar se muestra un mensaje “Proceso finalizado correctamente” (enmarcado con rojo):



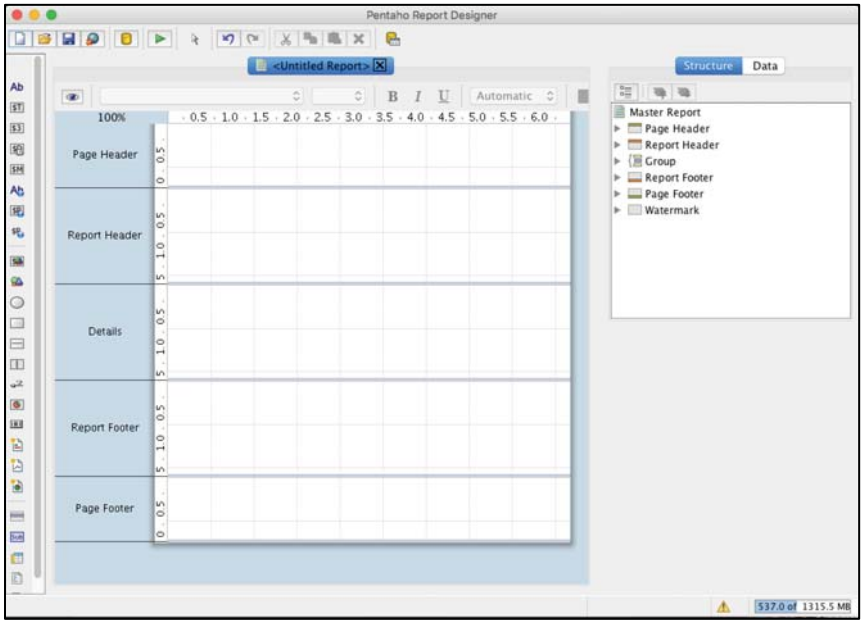


Anexo C: Creación de un reporte

a) Pentaho (Mac OS)

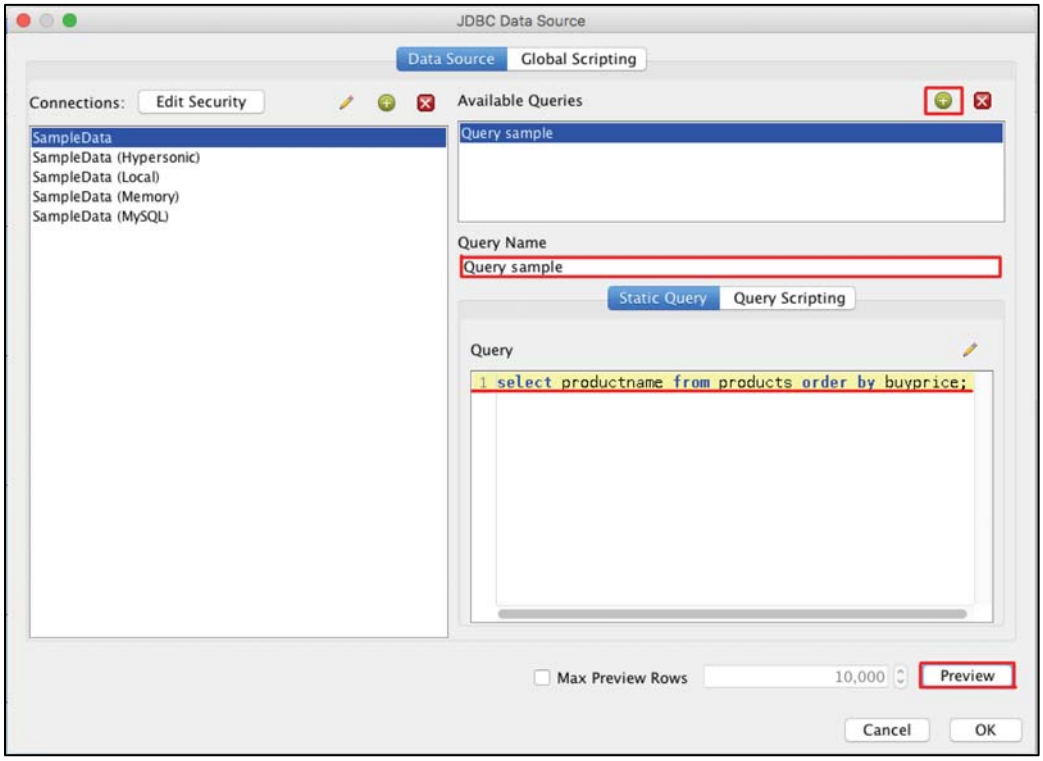
1.- Primero, ejecutar **Pentaho Report Designer** (doble click sobre la aplicación en Mac). Luego dar click en New Report:

2.- Se desplegará el **lienzo en blanco**. Cada sección del lienzo especifica como van a ordenarse los elementos:



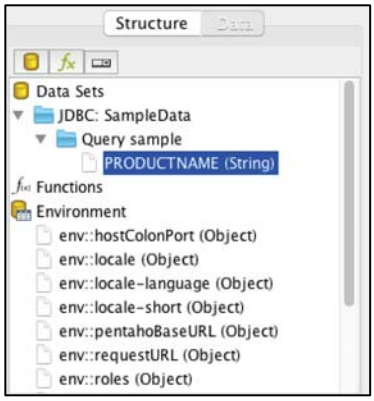
3.- Añadir un **origen de datos** (Data → JDBC):

4.- Se muestran las **conexiones disponibles** (si se requiere una nueva el proceso es similar al visto en el Anexo A). Pentaho incluye datos de prueba ya cargados para crear reportes. Dar click en el símbolo de agregar consulta (Available Queries). Proporcionar un nombre (Query Name) y luego, en la sección de Query, agregar la consulta que se requiere para extraer en el reporte. Para visualizar como se vera el reporte, dar click en Preview:

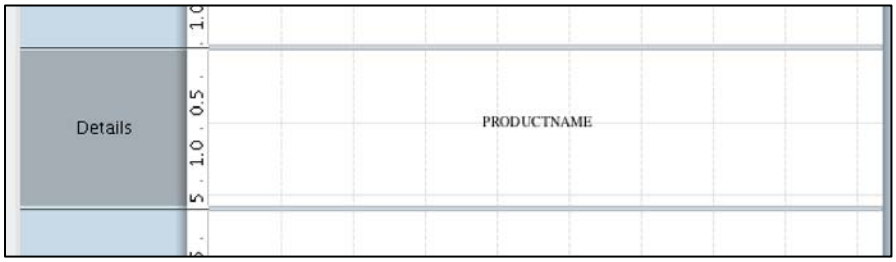


5.- Se presenta una **vista previa** de los datos que se obtendrán en la consulta. Estos datos se quedarán guardados en un conjunto de datos:

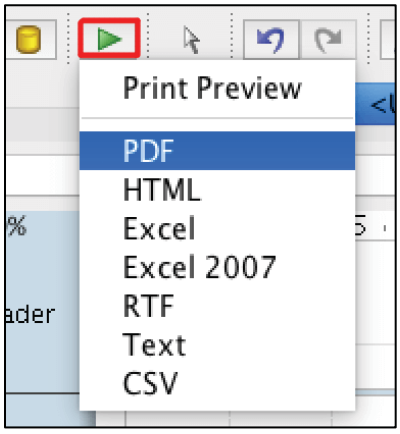
6.- En la sección **Data Sets**, se genera una nueva carpeta con la consulta recién creada. Este elemento (PRODUCTNAME) podrá ser arrastrado al reporte:



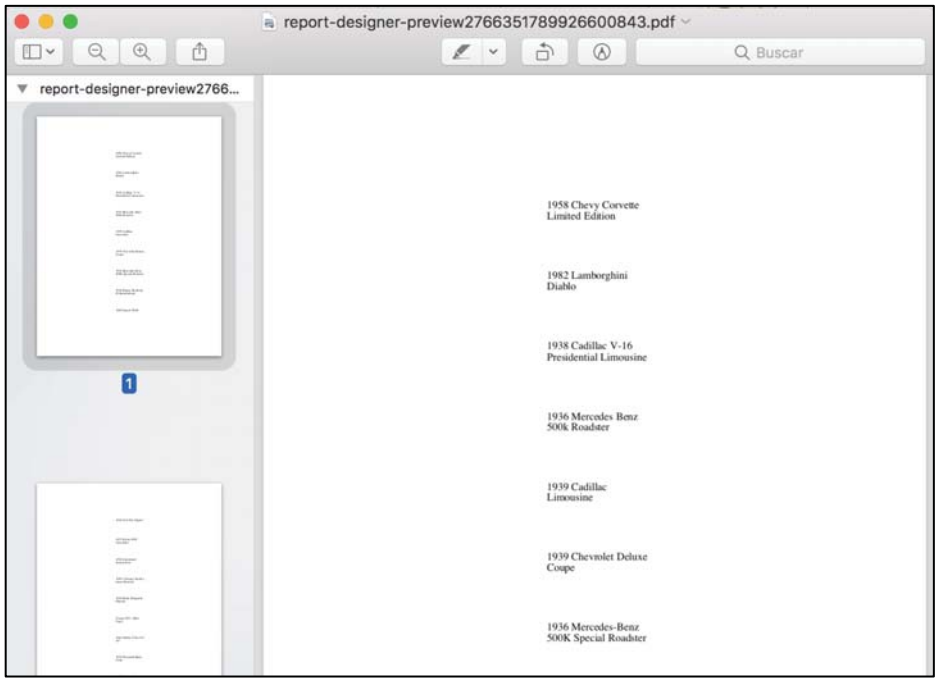
7.- Arrastrar la **consulta** en la sección Details:



8.- En la barra superior, ejecutar el reporte dando click en la flecha verde y luego seleccionando el tipo de archivo (por ejemplo PDF):

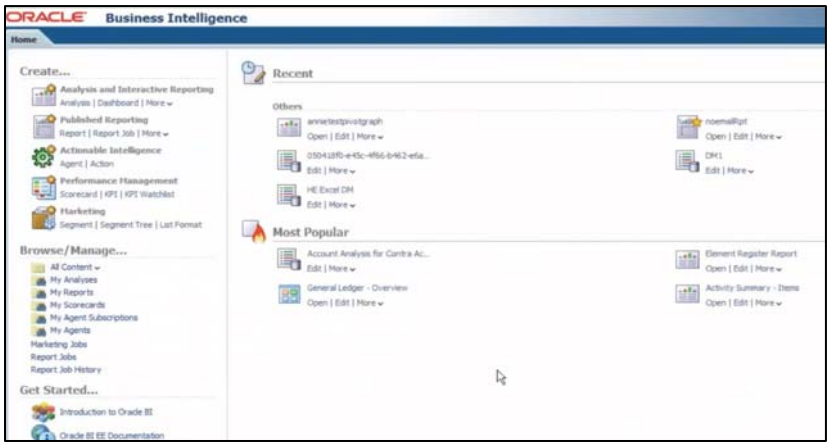


9.- Una vez que se muestra el archivo, se despliegan los datos de la consulta. Este archivo puede ser guardado y renombrado como un archivo PDF convencional:

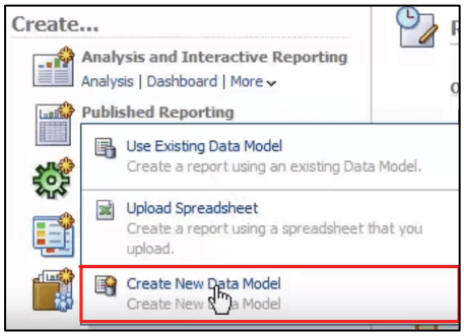


b) Oracle (Windows)

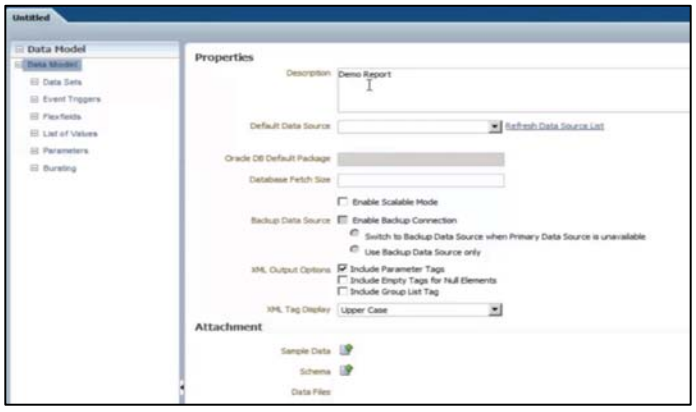
1.- Primero, ejecutar **Oracle Published Reporting** (desde servidor):



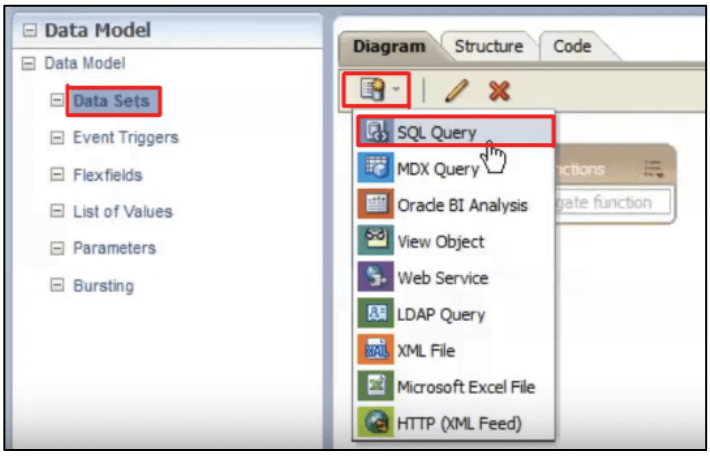
2.- Crear un **modelo de datos** (Create → Published Reporting → Report → Create New Data Model):



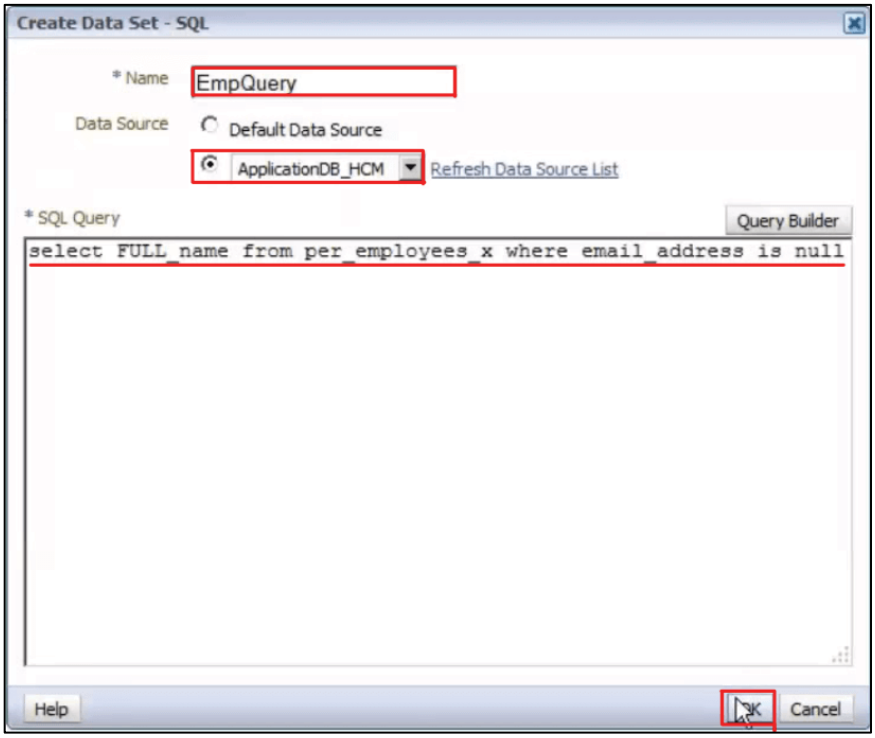
3.- Aparecerá la siguiente ventana:



4.- Añadir el **conjunto de datos** (Data Model → Data Sets → New Data Set → SQL Query):

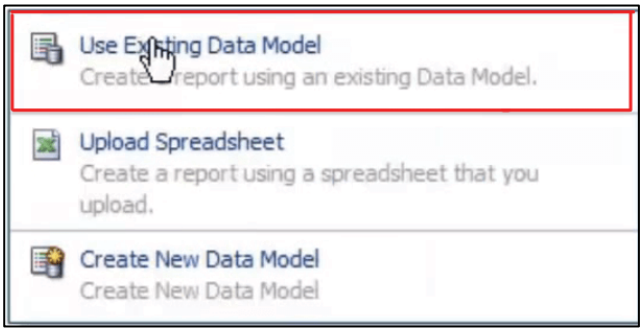


5.- Crear la **consulta** para extraer los datos. En la sección Name asignar un nombre (EmpQuery), en la sección Data Source elegir una fuente de datos y en la sección SQL Query escribir la consulta de los datos que se requieren extraer. Luego dar click en Ok:

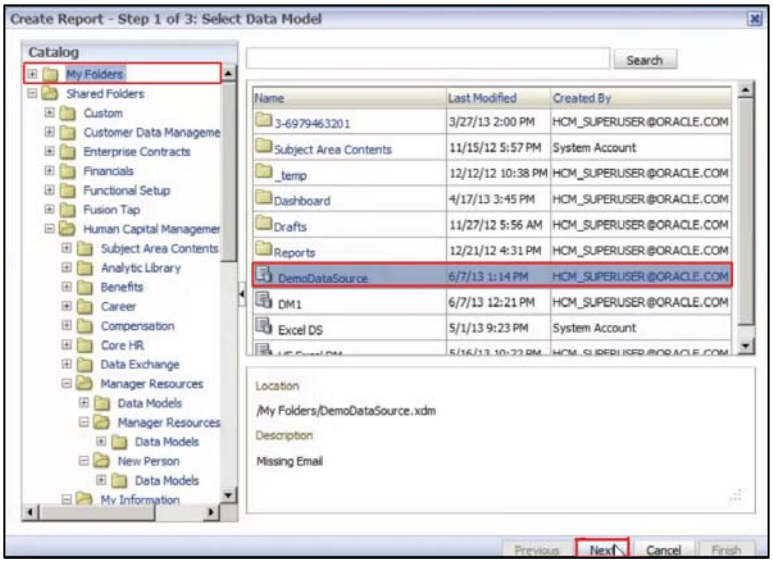


6.- Una vez almacenada la consulta, aparece el nuevo elemento (FULL_NAME):

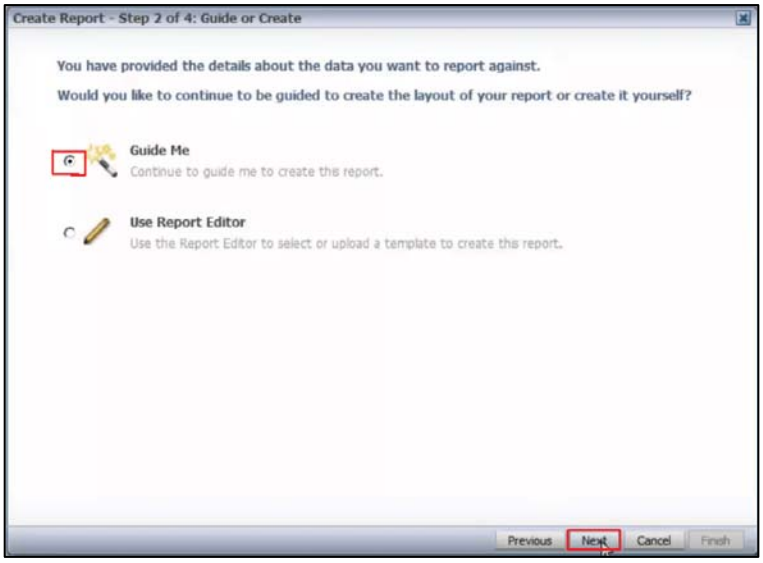
7.- Crear un **reporte** (New → Report → Use Existing Data Model):



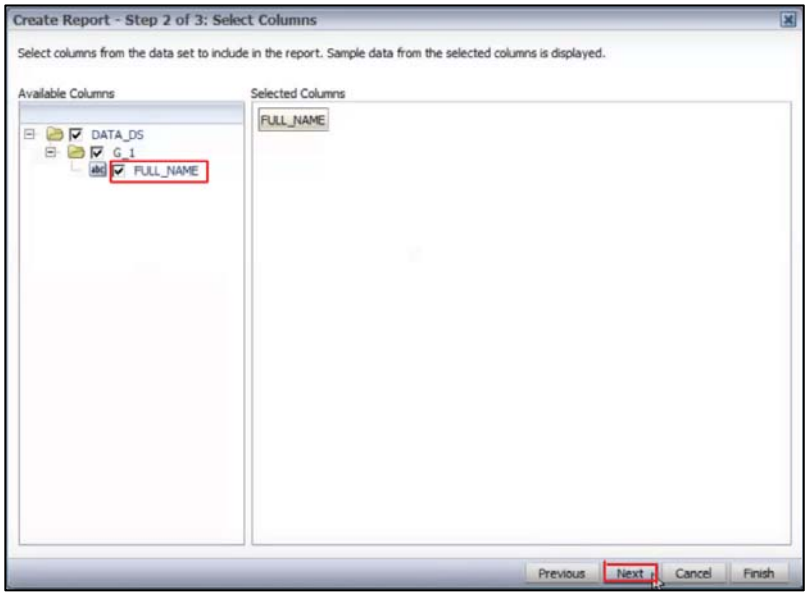
8.- Seleccionar un **modelo de datos** (ya configurado), luego dar click en Next:



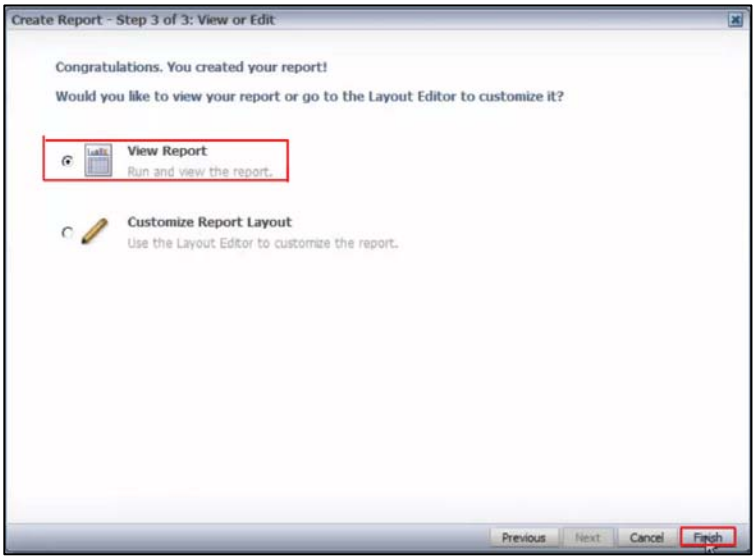
9.- Seleccionar el **tipo de creación** (guiado o manual). Luego dar click en Next:



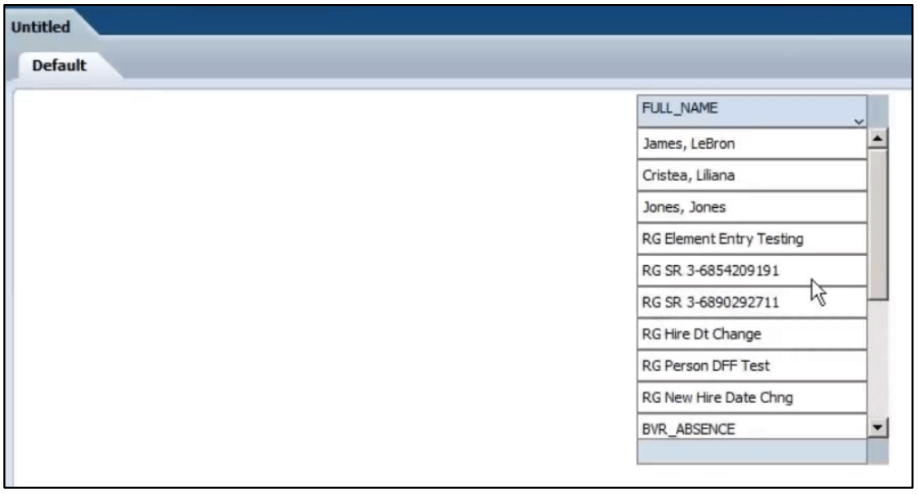
10.- Al seleccionar el modo Guide Me, se muestra una ventana que contiene los conjuntos de datos disponibles. Seleccionar los conjuntos que se requieren (en este caso solo existe FULL_NAME) y luego dar click en Next:



11.- En la siguiente ventana, se consulta al usuario si requiere ver el reporte (View Report) o si desea personalizarlo (Customize Report Layout). Seleccionar View Report y luego dar click en Finish:

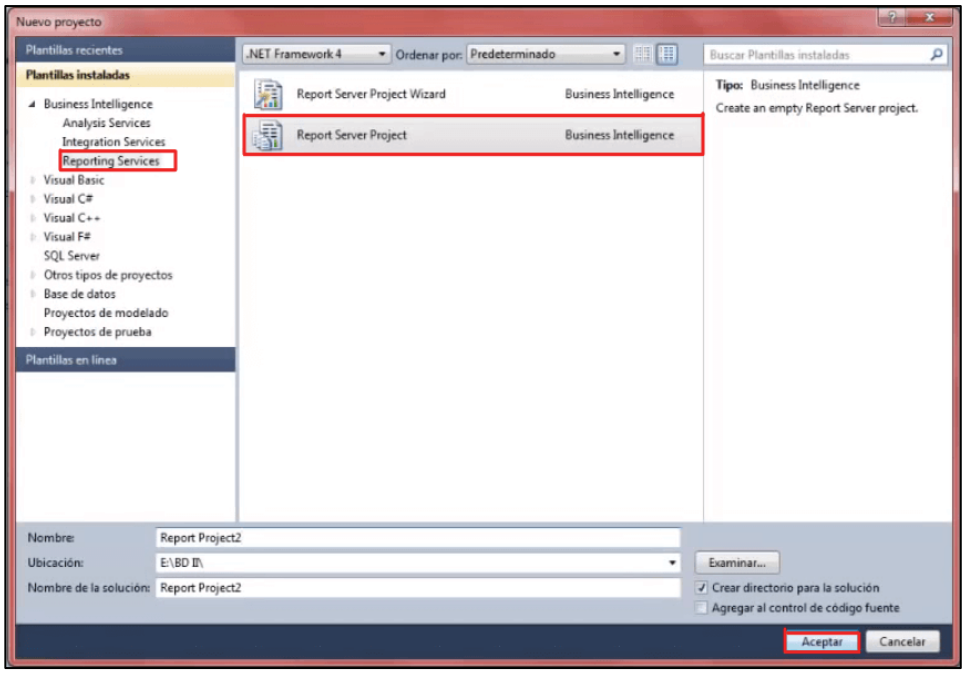


12.- Al finalizar, el usuario deberá guardar el reporte asignando un nombre. Se muestra un reporte como el siguiente. En este caso, el lienzo esta configurado para mostrar el tipo de reporte por omisión.

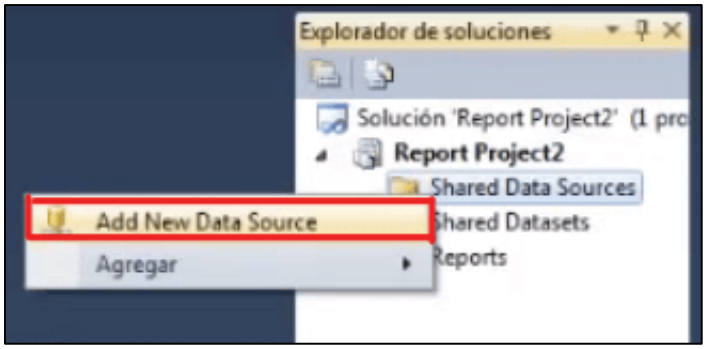


c) SQL Server (Windows)

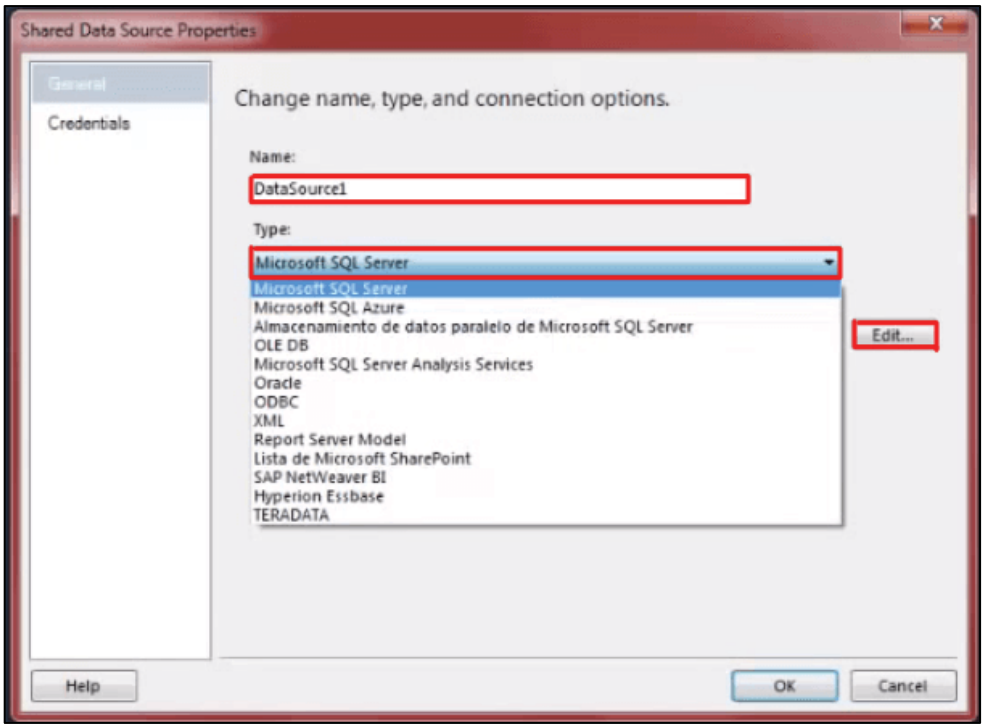
1.- Ejecutar Visual Studio 2010 (Inicio → Microsoft Visual Studio 2010 → Microsoft Visual Studio 2010). Crear un **nuevo proyecto** (Reporting Services -> Report Server Project) y luego dar click en Aceptar:



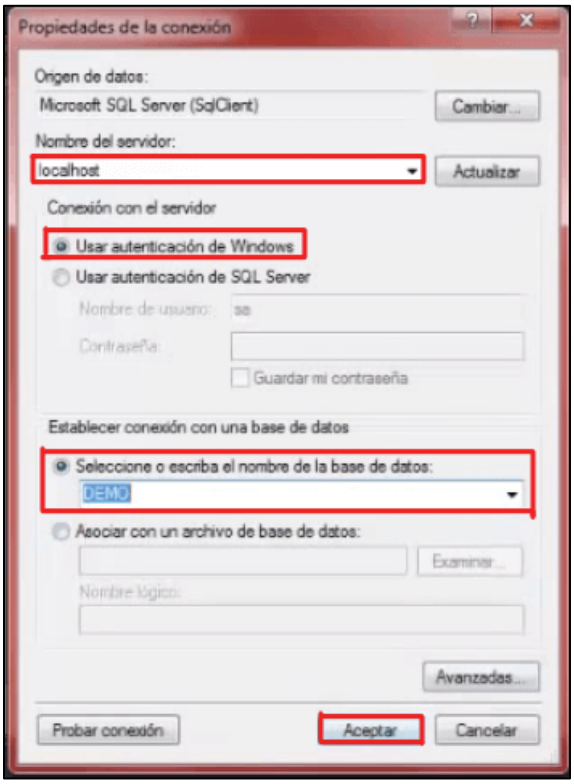
2.- En la sección del Explorador de soluciones, crear una nueva **fuente de datos** (“Nombre del proyecto” → Shared Data Sources → Add New Data Source):



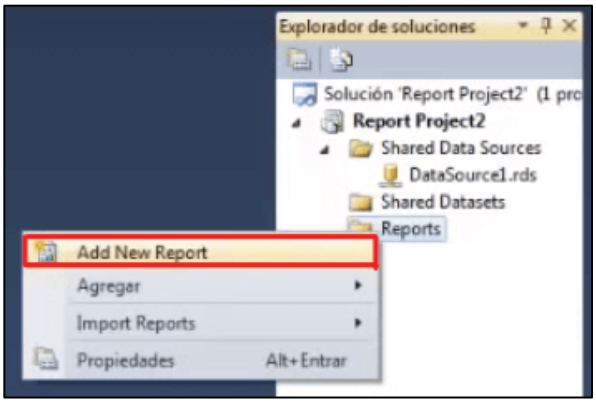
3.- En la siguiente ventana, seleccionar una tipo de origen de datos, asignarle un nombre (DataSource1) y luego dar click en Edit:



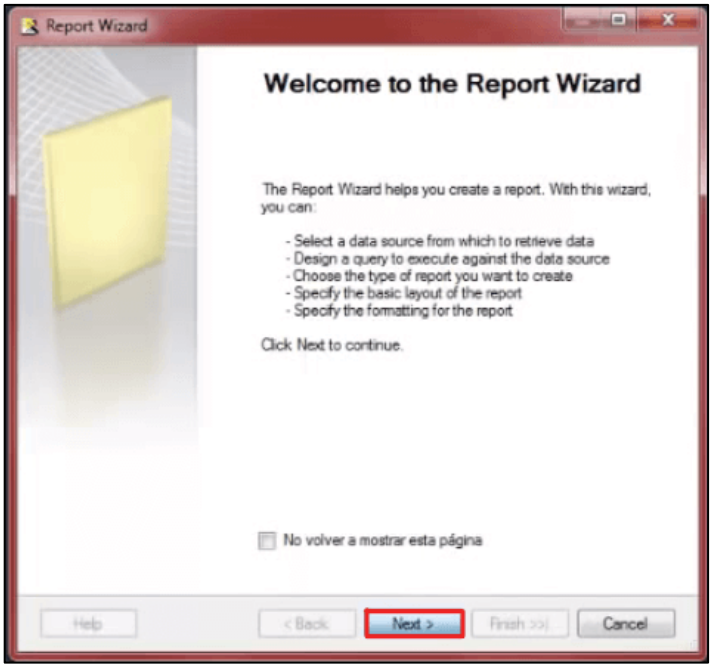
4.- Ingresar las propiedades de la conexión (Nombre del servidor, conexión con el servidor y nombre de la base de datos). Luego dar click en Aceptar:



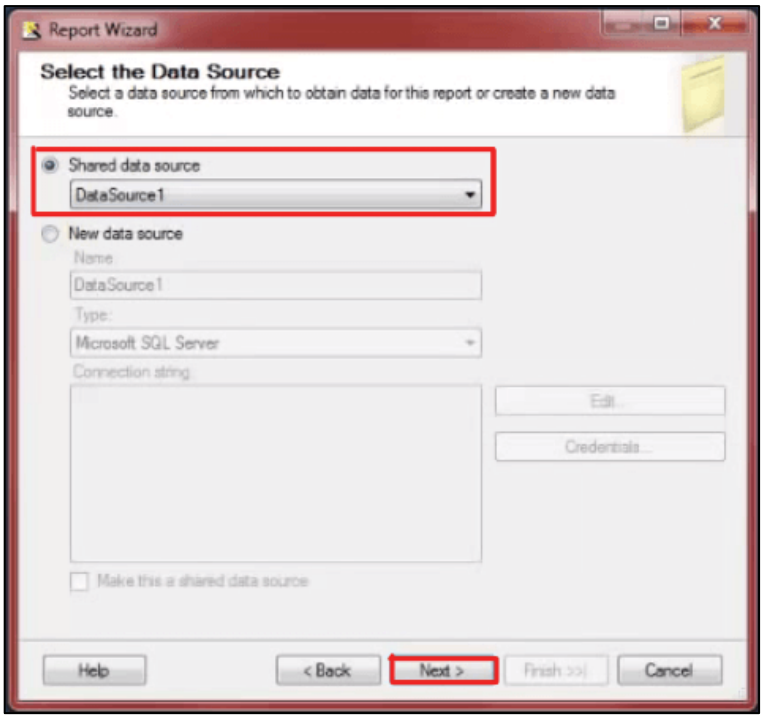
5.- Crear un **reporte** (“Nombre del proyecto” → Reports → Add New Report):



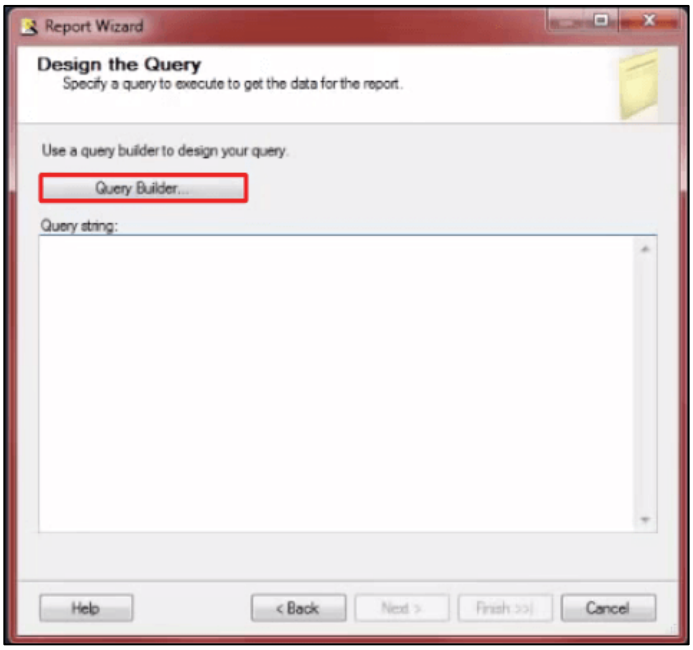
6.- Aparecerá el **asistente** de reportes:



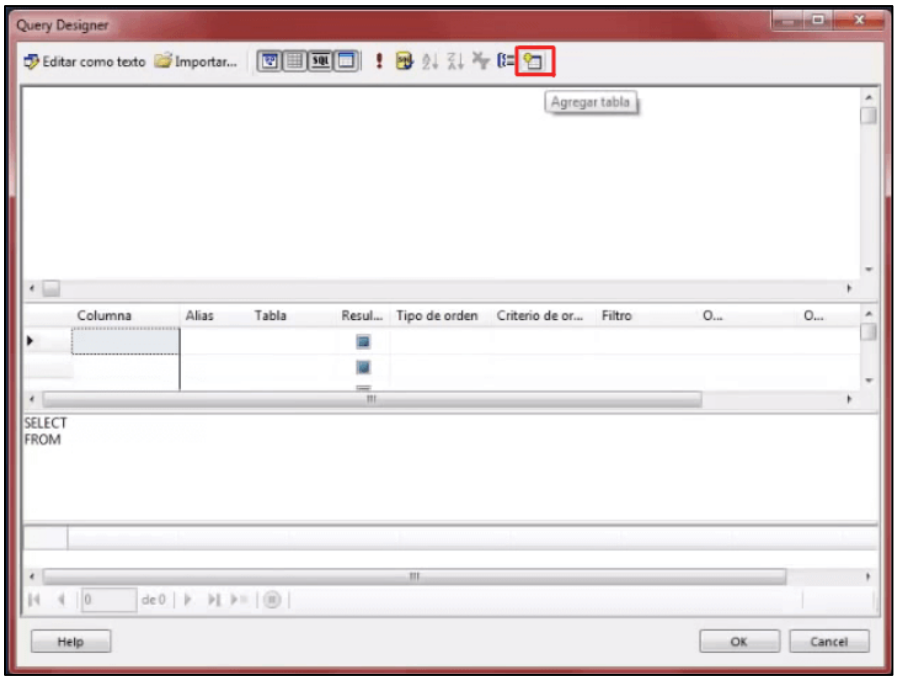
7.- Seleccionar la **fuente de datos**. Por omisión esta seleccionada la fuente recién creada. En caso de requerir otra fuente que ya este creada, dar click en el menú deslizable (DataSource1). En caso de requerir una nueva, dar click en New data source y luego configurar. En este caso, basta con dejar la fuente ya creada (DataSource1) y luego dar click en Next:



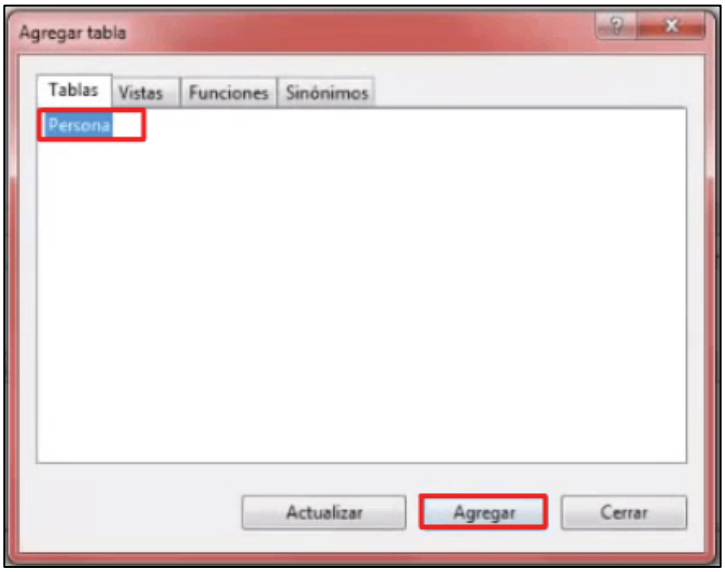
8.- Aparecerá la ventana para **diseñar consultas**. Click en Query Builder:



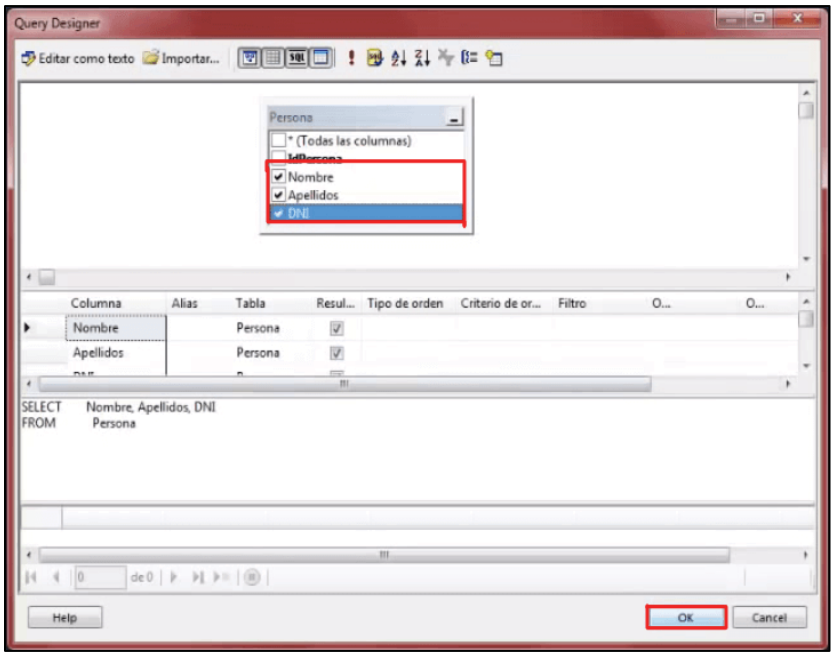
9.- En la siguiente ventana se muestra un **editor para consultas**. Es posible añadir la consulta mediante texto (sql) pero para facilitar el proceso es mejor agregar la tabla directamente dando click en el icono de Agregar tabla:



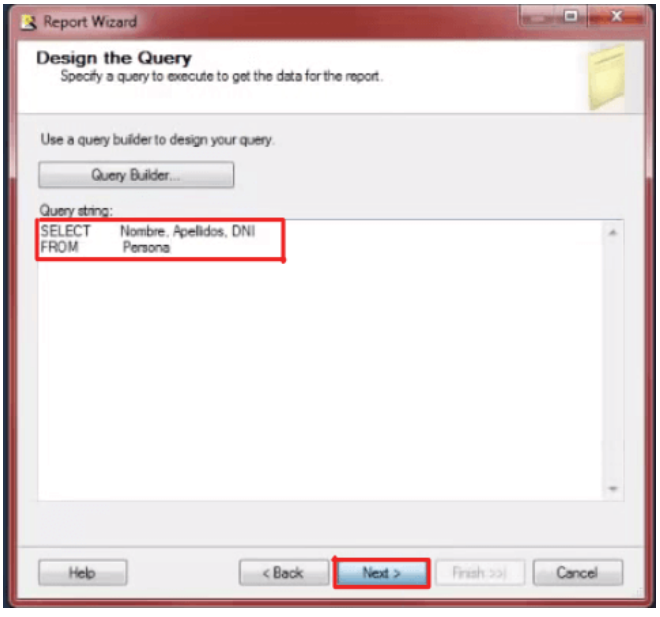
10.- Seleccionar la tabla de la que se van a extraer los datos:



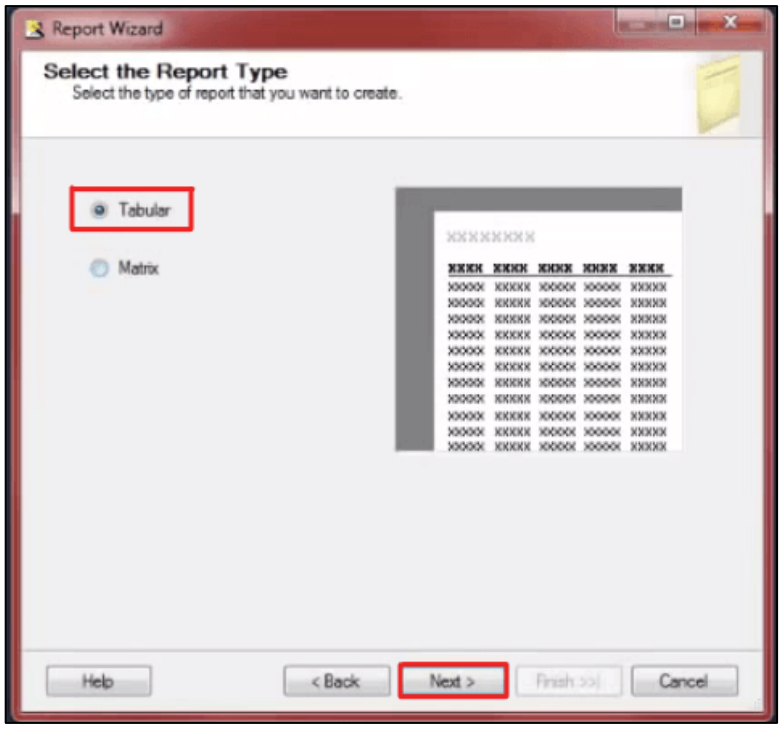
11.- En el editor de consultas se muestra la tabla que acaba de agregarse. Se observan todos los campos que incluye esa tabla. El usuario debe añadir manualmente, cuales de esos campos desea que se agreguen a su reporte. En este caso, de los 4 campos solo se han añadido 3. Luego dar click en Ok:



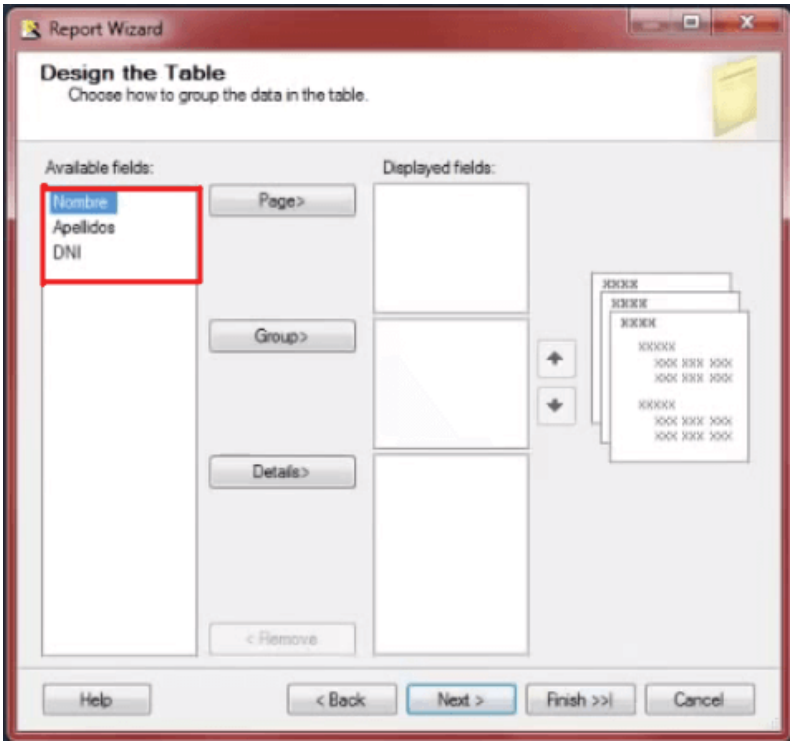
12.- En la ventana para diseñar consultas aparecerá el código que extraerá los datos. Los anteriores pasos pueden ser omitidos cuando el usuario ya conoce la consulta para extraer los datos de su reporte. Click en Next:



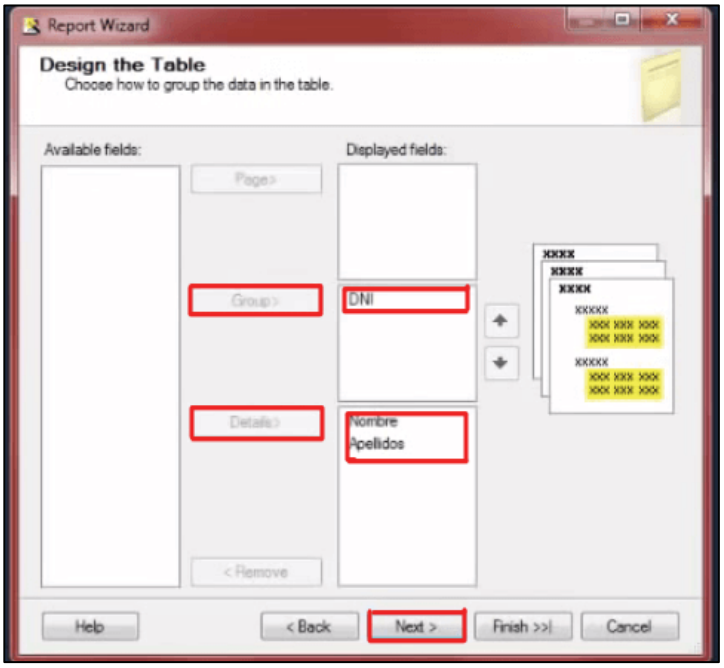
13.- En la siguiente ventana, elegir el **tipo de formato** del reporte. Click en Next:



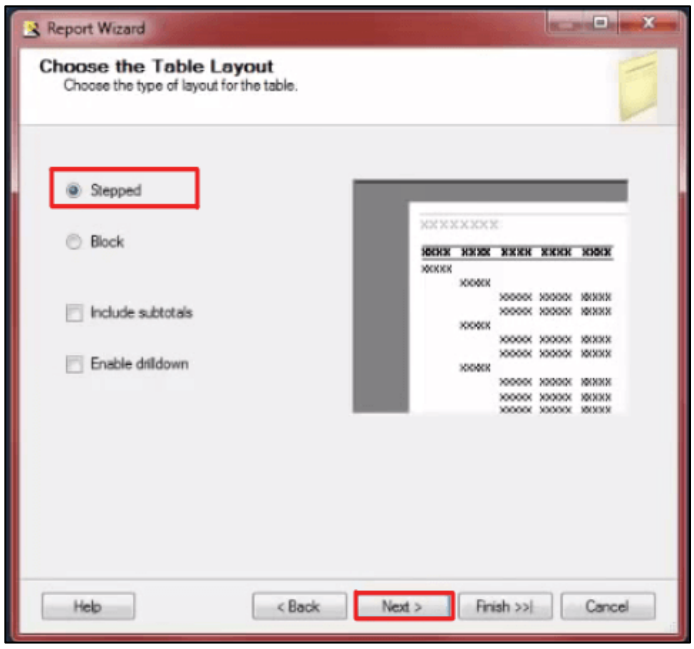
14.- Se muestran los campos disponibles. El usuario debe ordenar y agrupar como sus datos serán mostrados:



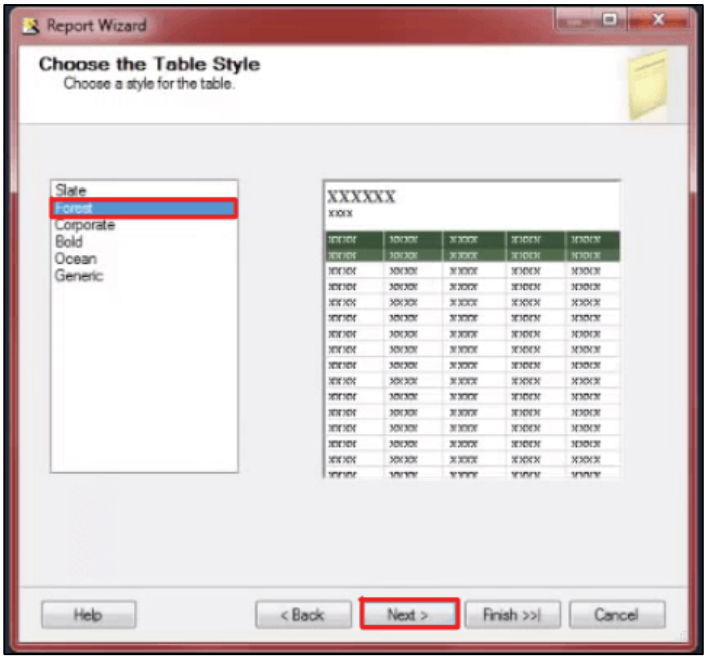
15.- Seleccionar el campo, y luego dar click en el orden. En este caso, los datos serán agrupados por el campo DNI y los detalles mostrarán el nombre y los apellidos. Click en Next:



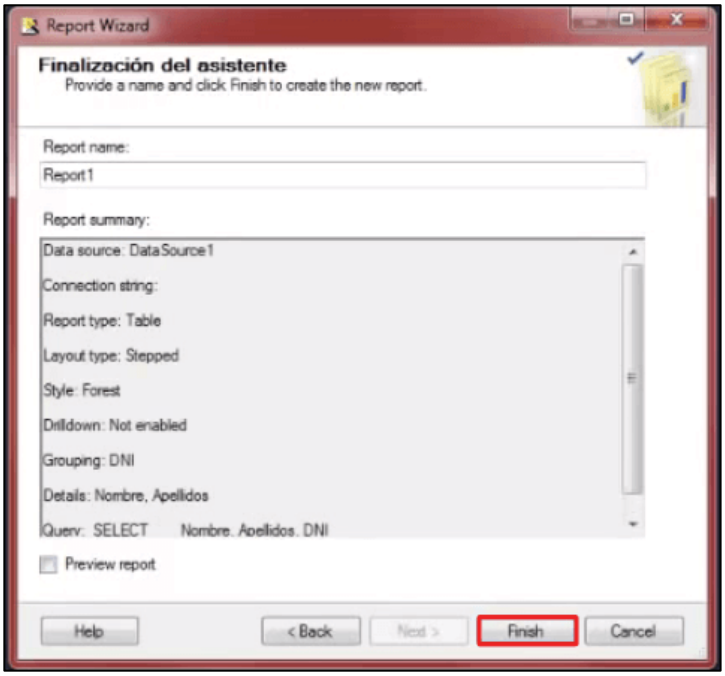
16.- Elegir el **diseño** de la tabla. Click en Next:



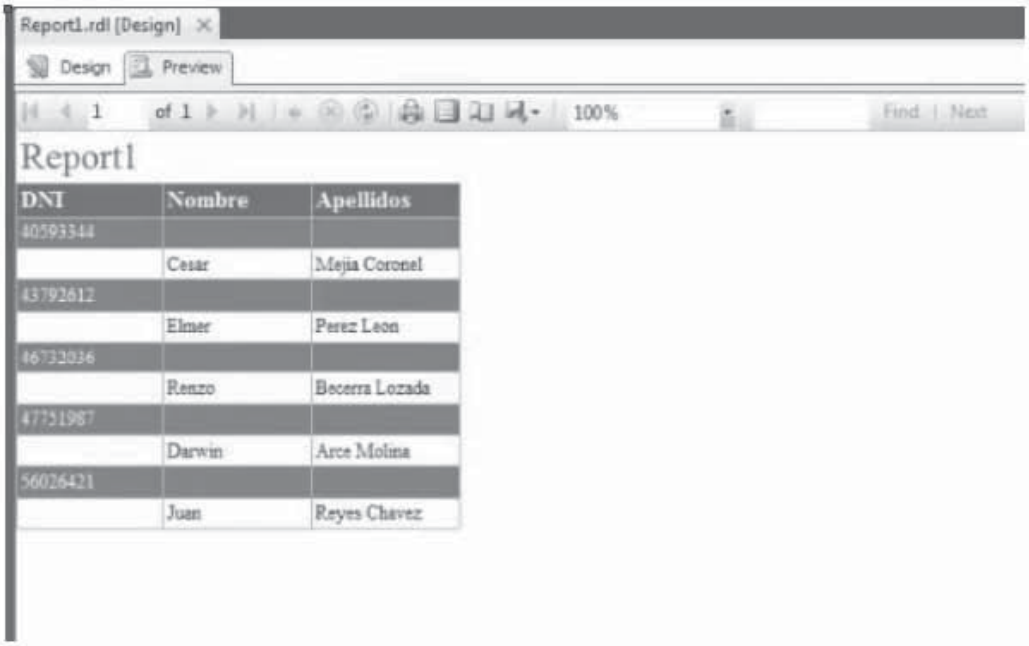
17.- Elegir el **estilo** de la tabla. Click en Next:



18.- Se muestra un resumen de la configuración del reporte. Dar click en Finish:



19.- El reporte se mostrará con las configuraciones recién mostradas. El reporte ya esta listo para ser exportado por el usuario:



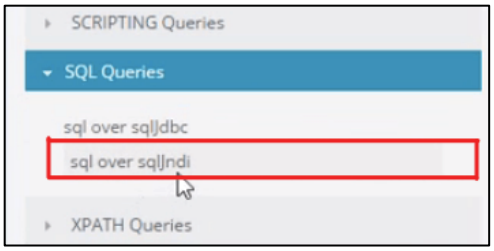
Anexo D: Creación de un tablero de mando

a) Pentaho (Linux)

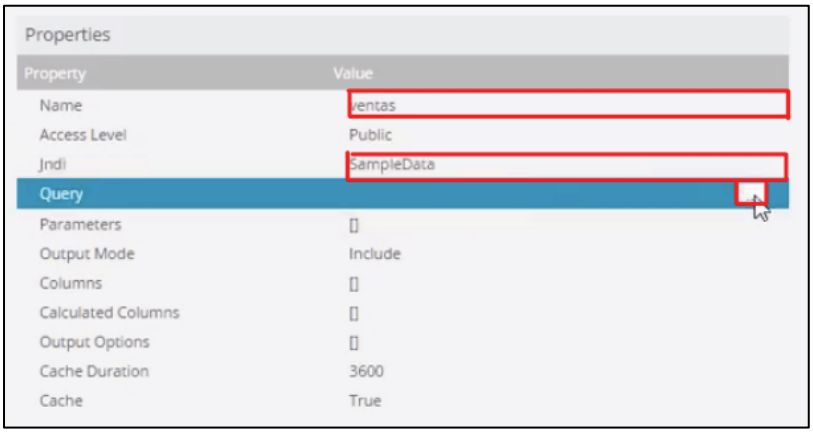
1.- Primero, ejecutar **Community Dashboard Editor** (desde servidor). Crear un nuevo tablero (Create New → CDE Dashboard):

2.- Se mostrará la siguiente ventana. Dar click en el icono azul (marcado con rojo) para desplegar el **panel de fuentes de datos** de la izquierda (marcado con rojo). Inicialmente, este panel estará oculto:

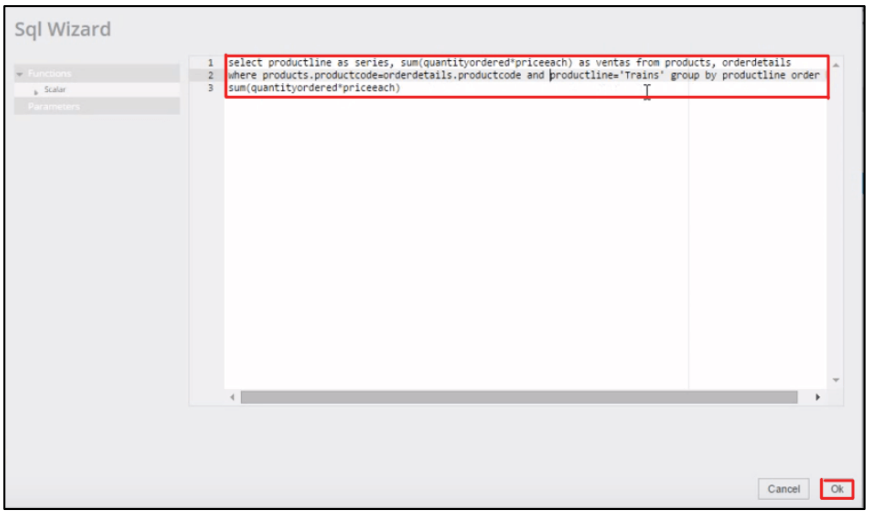
3.- Crear una **consulta** (SQL Queries → sql over sqljndi):



4.- En la sección Property añadir un **nombre** (ventas) y el **origen de datos** (SampleData). Luego añadir la consulta dando click en el campo Query:



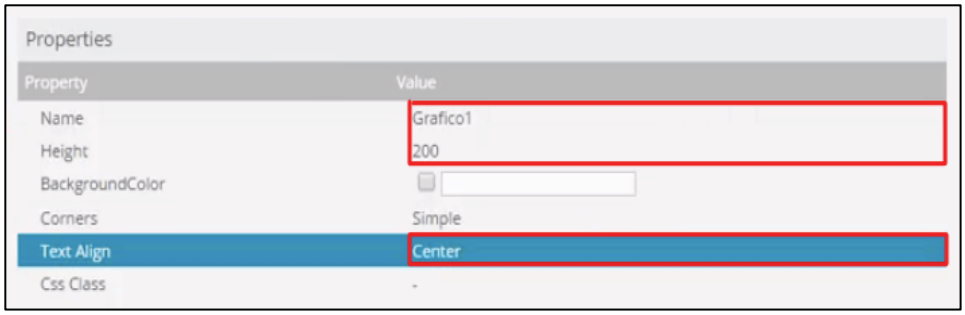
5.- Aparecerá una ventana nueva. Escribir la **consulta** de los datos que se requieren extraer, luego dar click en Ok:



6.- Desplegar el **panel de diseño** (Click en Layout Panel, marcado con rojo):

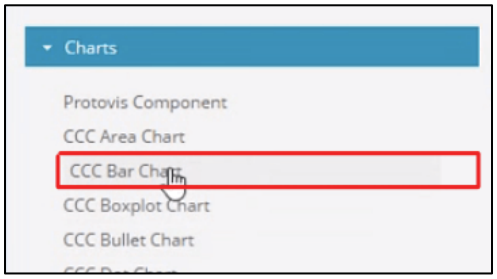
7.- En la sección Layout Structure añadir un **renglón**:

8.- En la sección Properties añadir un nombre (Grafico1), un tamaño (200) y la alineación de texto (Center) para el renglón recién creado:

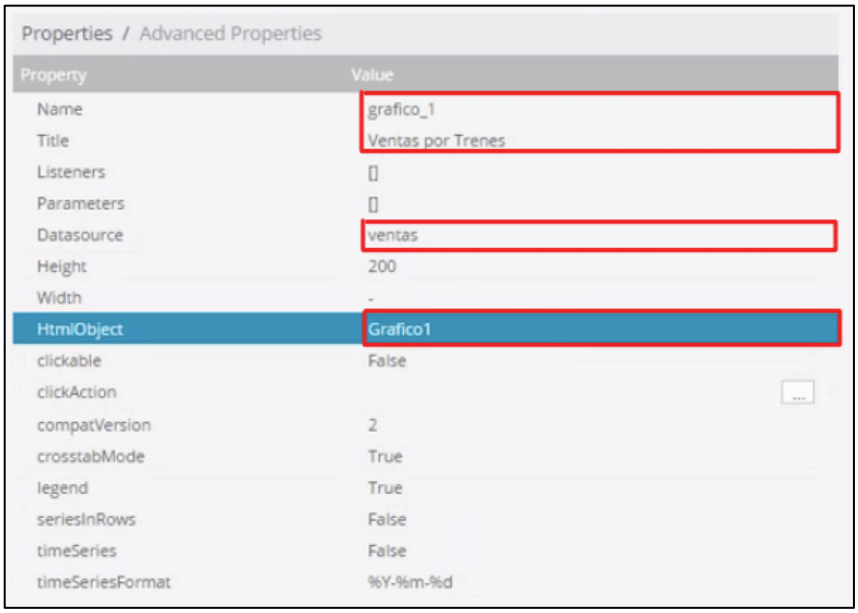


9.- Desplegar el **panel de componentes** (marcado con rojo):

10.- Añadir una **gráfica de barra** (Charts → CCC Bar Chart):



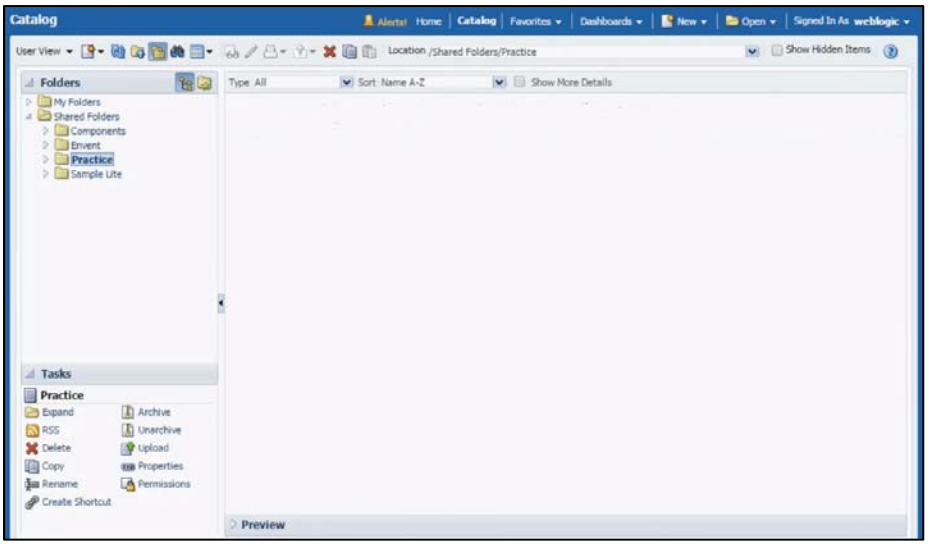
11.- En el panel de Properties, añadir un **nombre** (grafico_1) y un **título** (Ventas por Trenes). En la sección Datasource debe ingresarse el **nombre de la consulta** previamente creada (ventas) y en la sección HtmlObject el **nombre del diseño** creado anteriormente (Grafico1):



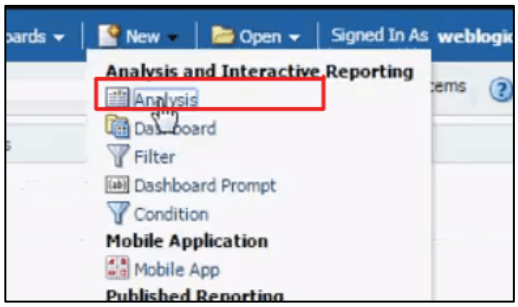
12.- Dar click en la visualización y se mostrara el resultado del tablero:

b) Oracle (Windows)

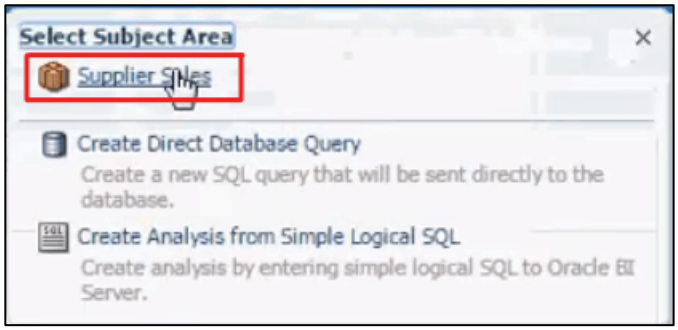
1.- Primero, ejecutar **Oracle** (desde servidor):



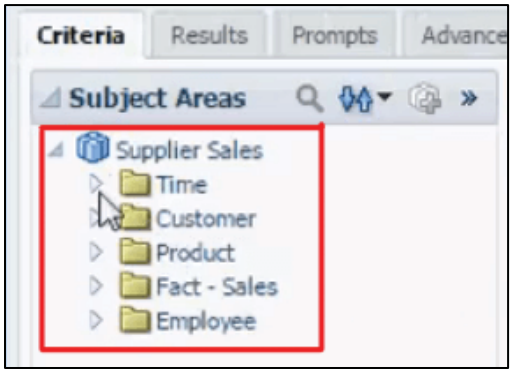
2.- Crear una **análisis** (New → Analysis and Interactive Reporting → Analysis):



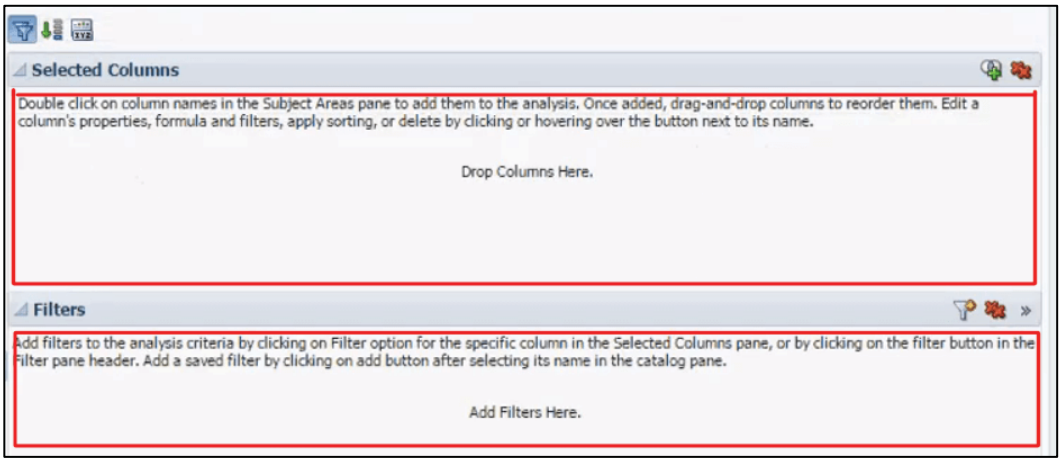
3.- Seleccionar un **conjunto de datos**. En este caso se selecciona uno ya creado (En caso de requerir uno nuevo, el proceso es parecido al visto en el Anexo A):



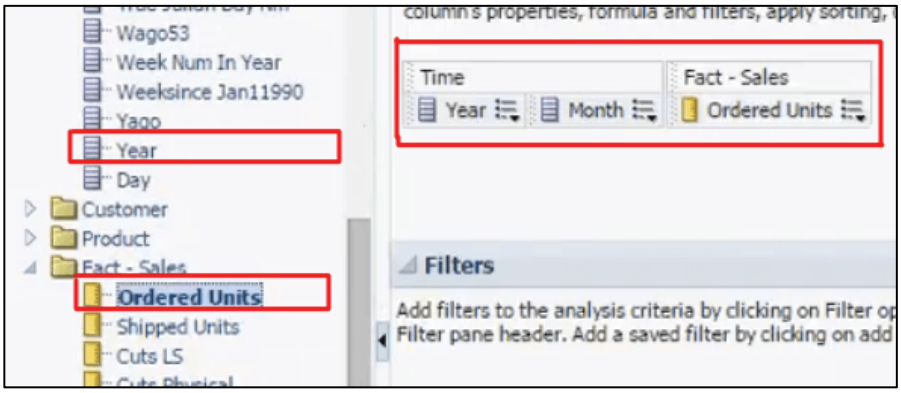
4.- En el lado izquierdo se mostrará el contenido del **conjunto de datos**:



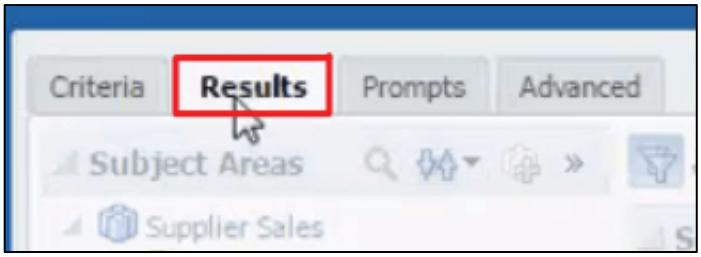
5.- En el lado derecho se mostrará el **editor** para generar el análisis. En la parte de arriba se agregan las columnas (campos) y abajo los filtros:



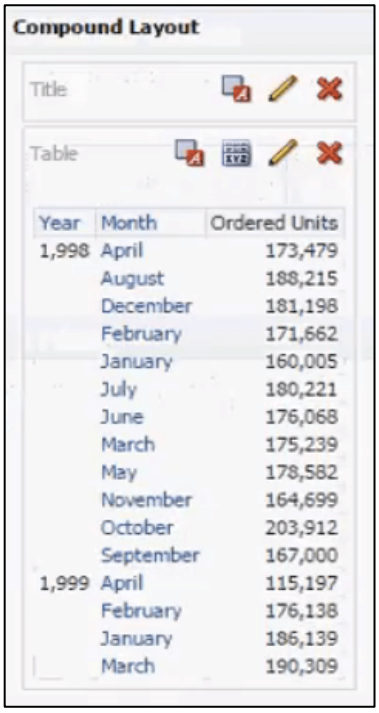
6.- Arrastrar los **campos** que se requieran para extraer sus datos hacia la parte derecha, se generarán objetos donde se muestra el titulo de los campos:



7.- Dar click en la sección Results para mostrar los datos:

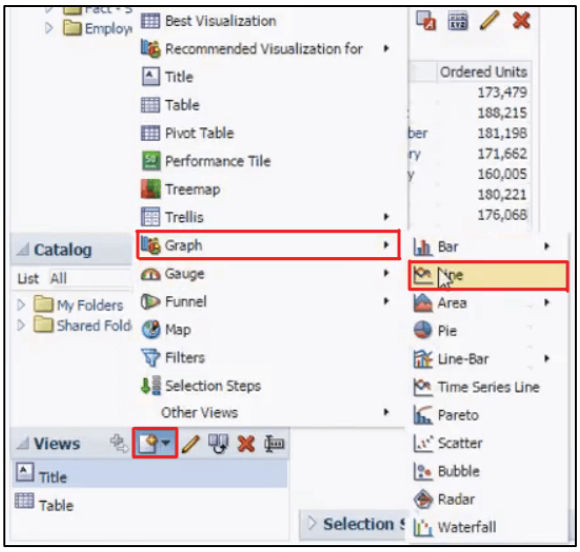


8.- Se mostrarán los **datos** que el usuario solicito:



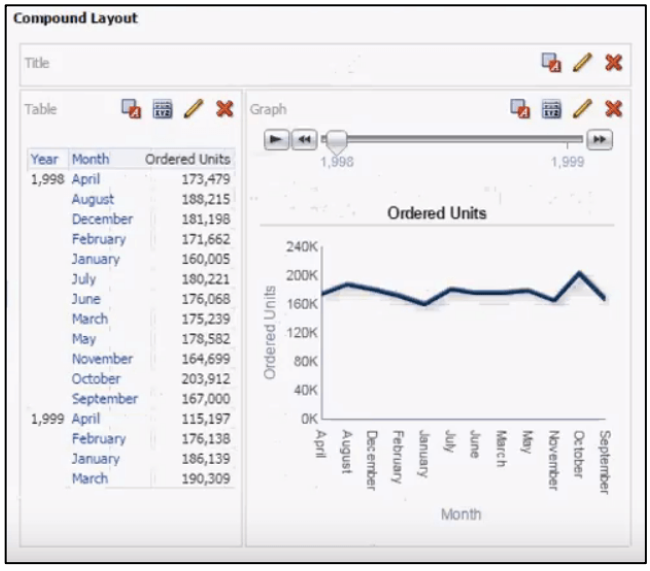
Year	Month	Ordered Units
1,998	April	173,479
	August	188,215
	December	181,198
	February	171,662
	January	160,005
	July	180,221
	June	176,068
	March	175,239
	May	178,582
	November	164,699
	October	203,912
	September	167,000
1,999	April	115,197
	February	176,138
	January	186,139
	March	190,309

9.- En la sección Views desplegar los **componentes** (icono marcado con rojo) para añadir una **gráfica**. Hasta este punto, los datos se muestran en texto plano, para interpretarlos mas fácilmente se desplegarán en una gráfica lineal. Primero será necesario seleccionar un componente para mostrar los datos, por ejemplo una gráfica lineal (Graph → Line):

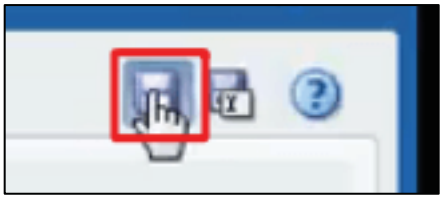


10.- Una vez que se ha seleccionado la **gráfica**, dar click en Done. Luego en la sección Views, dar click en el icono marcado para transferir los datos:

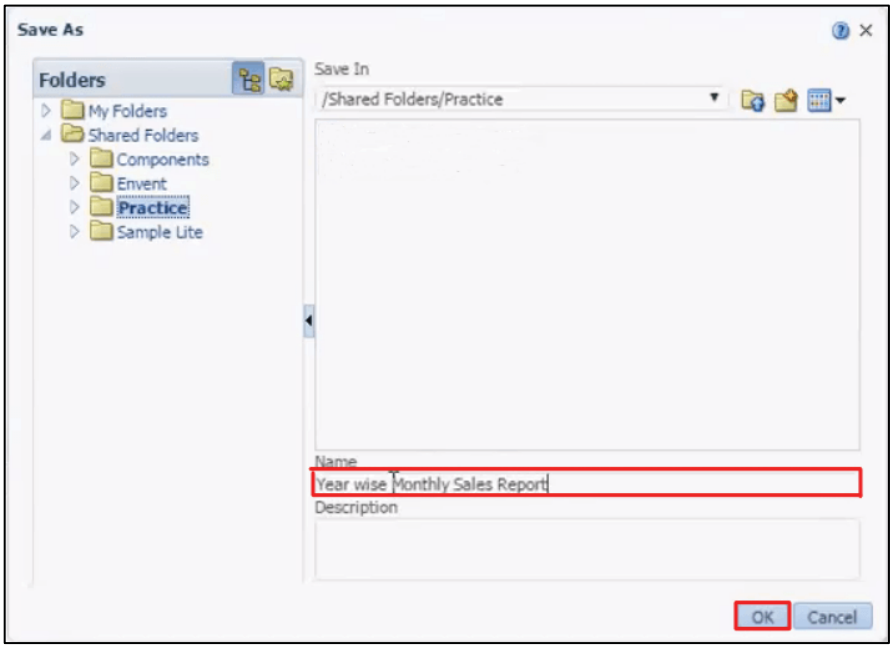
11.- Se mostrarán los **datos** y la **gráfica**:



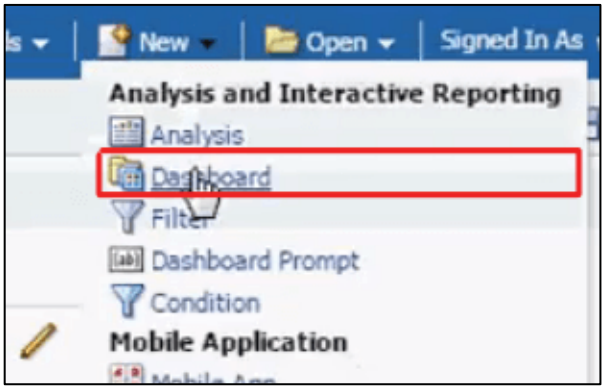
12.- Click en el icono de Guardar:



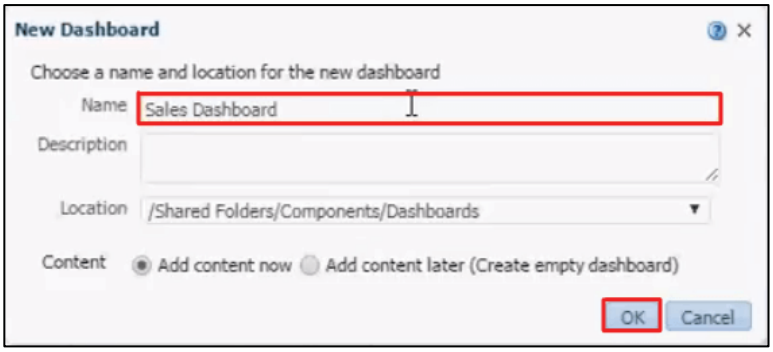
13.- Asignar un nombre al análisis (Year wise Monthly Sales Report):



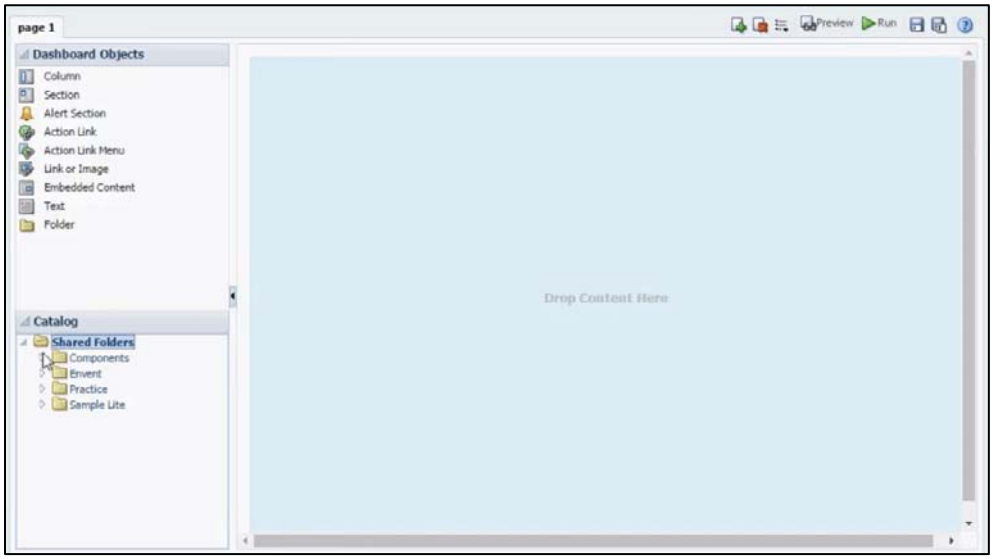
14.- Crear un **tablero de mando** (New → Analysis and Interactive Reporting → Dashboard):



15.- Seleccionar un **nombre** para el tablero, luego dar click en Ok:

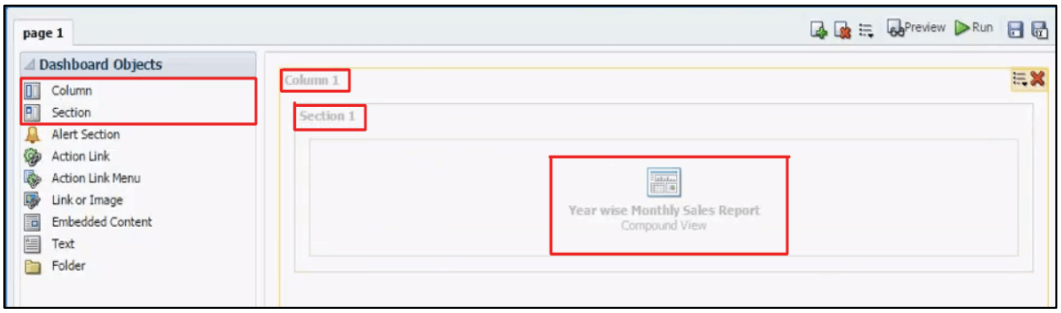


16.- Se mostrará una ventana diferente:

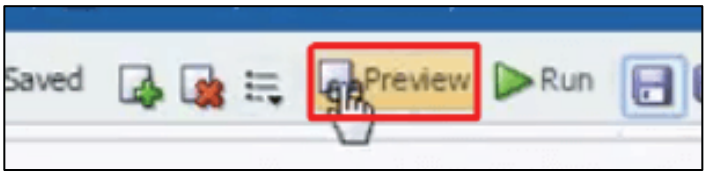


17.- El análisis (o reporte) recién creado aparecerá en las carpeta donde el usuario lo haya guardado:

18.- Para crear el tablero basta con darle un orden a los elementos. En la sección Dashboard Objects, tomar una **columna** (Column) y arrastrarla hacia el editor. Luego tomar una **sección** (Section) y arrastrarla dentro de la columna. Finalmente añadir el **análisis** dentro de la sección:



19.- Dar click en la **vista previa** (Preview):

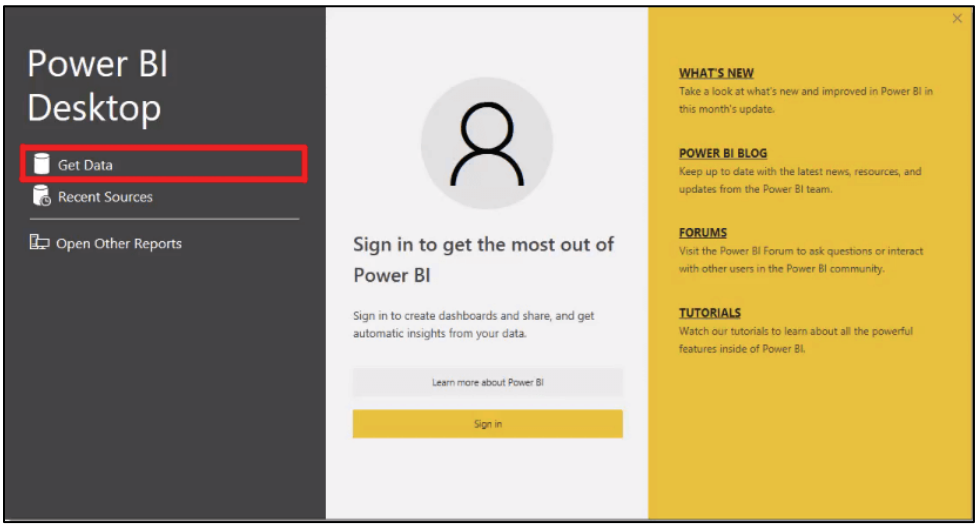


20.- Se mostrará el tablero con los elementos del análisis:

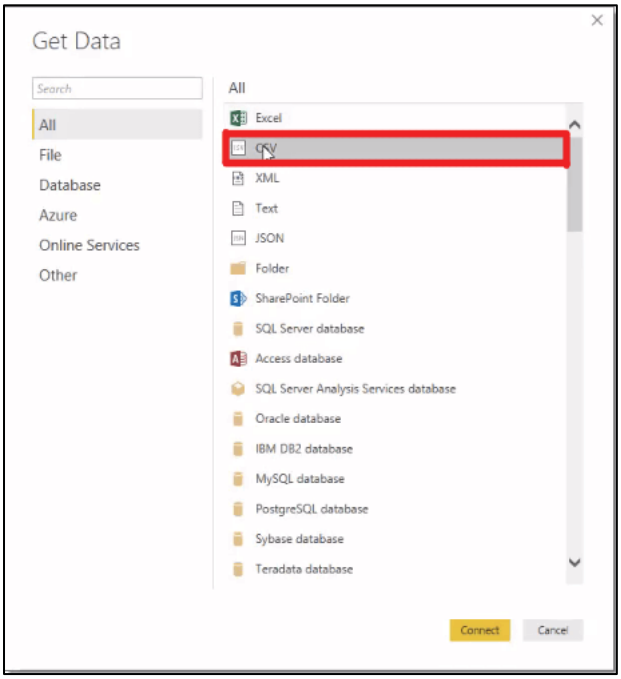
Year	Month	Ordered Units
1,998	April	173,479
	August	188,215
	December	181,198
	February	171,662
	January	160,005
	July	180,221
	June	176,068
	March	175,239
	May	178,582
	November	164,699
	October	203,912
	September	167,000
1,999	April	115,197
	February	176,138
	January	186,139
	March	190,309

c) SQL Server (Windows)

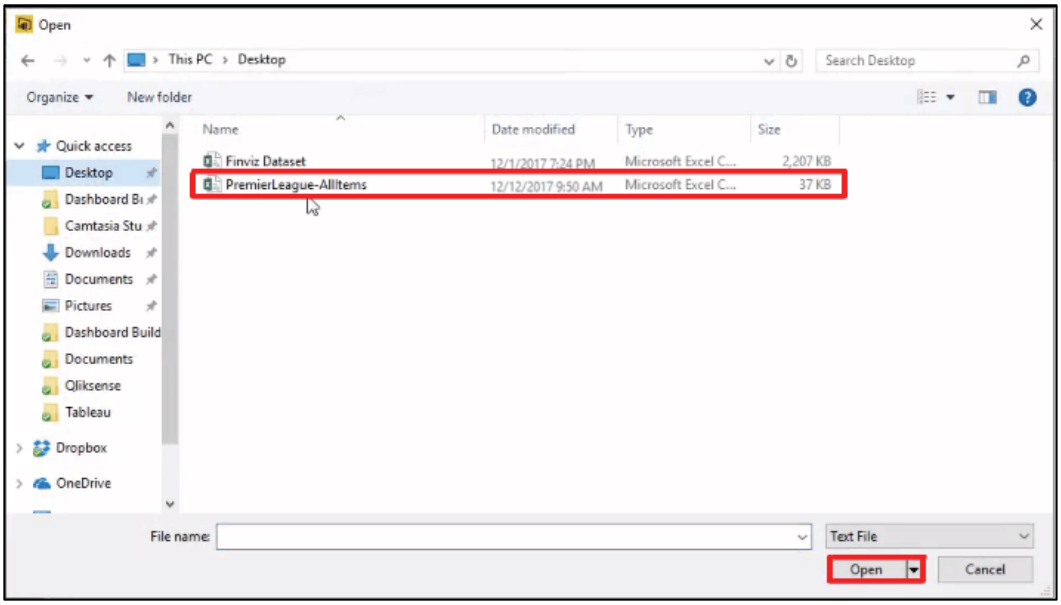
1.- Ejecutar **Power BI** (Click en la aplicación de escritorio). Cuando se muestre la bienvenida, dar click en Get Data:



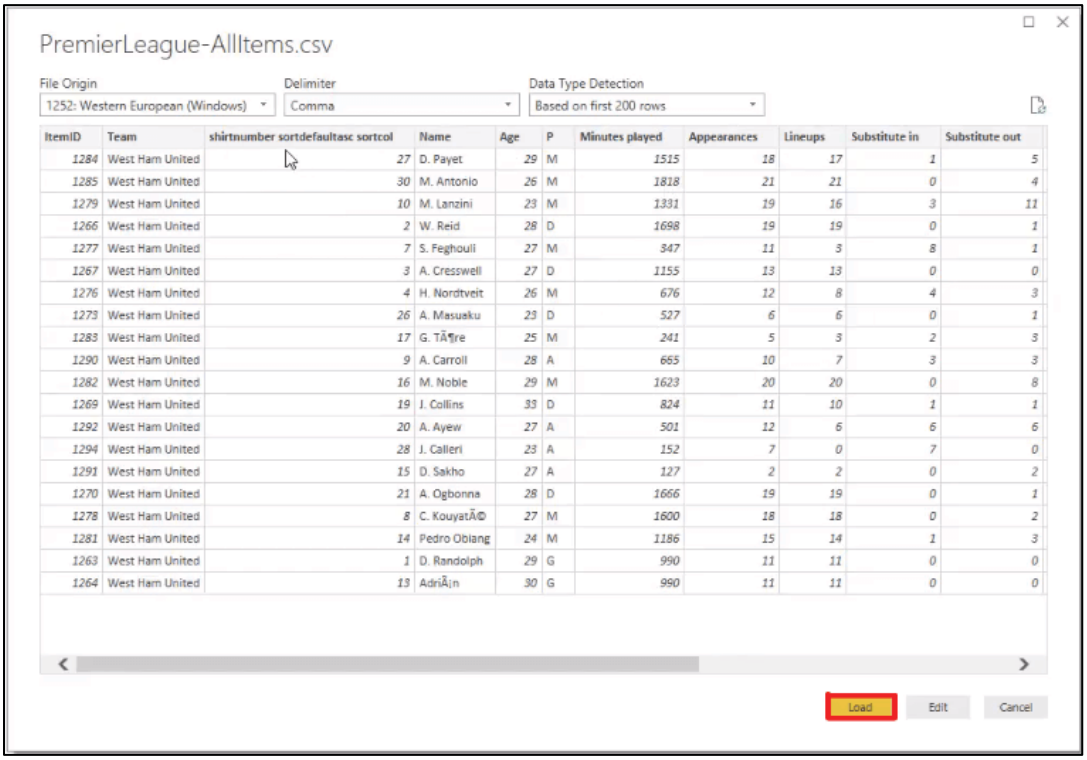
2.- Seleccionar la **fuente de datos** por ejemplo un CSV, luego dar click en Connect :



3.- Seleccionar el archivo que contiene los datos del tipo de fuente que se selecciono, luego dar click en Open:

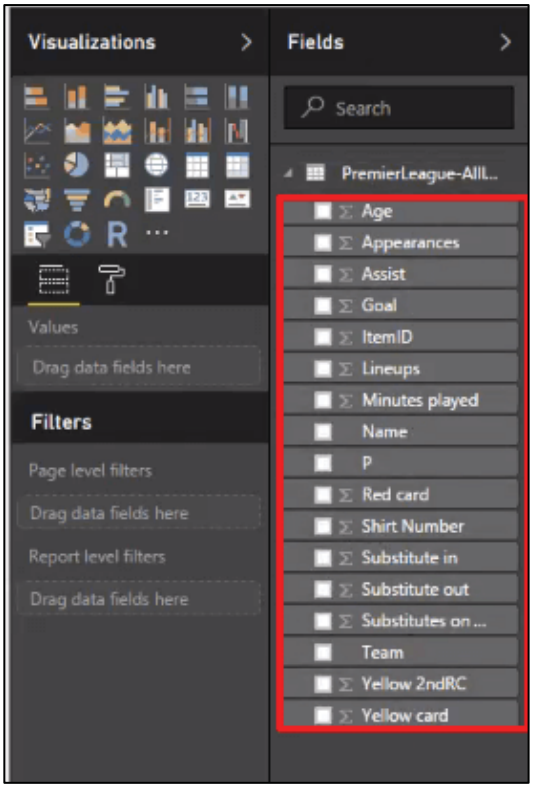


4.- Se mostrará una vista previa de los datos del archivo. Dar click en Load:

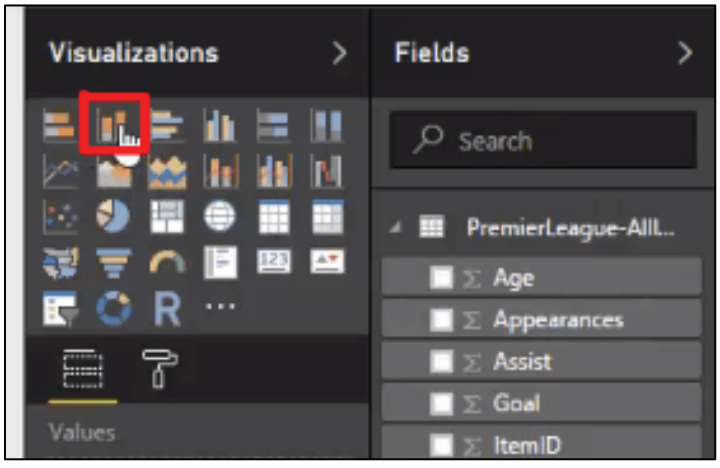


ItemID	Team	shirtnumber sortdefaultasc sortcol	Name	Age	P	Minutes played	Appearances	Lineups	Substitute in	Substitute out
1284	West Ham United	27	D. Payet	29	M	1515	18	17	1	5
1285	West Ham United	30	M. Antonio	26	M	1818	21	21	0	4
1279	West Ham United	10	M. Lanzini	23	M	1331	19	16	3	11
1266	West Ham United	2	W. Reid	28	D	1698	19	19	0	1
1277	West Ham United	7	S. Feghouli	27	M	347	11	3	8	1
1267	West Ham United	3	A. Cresswell	27	D	1155	13	13	0	0
1276	West Ham United	4	H. Nordtveit	26	M	676	12	8	4	3
1273	West Ham United	26	A. Masuaku	23	D	527	6	6	0	1
1283	West Ham United	17	G. TÃ¶re	25	M	241	5	3	2	3
1290	West Ham United	9	A. Carroll	28	A	665	10	7	3	3
1282	West Ham United	16	M. Noble	29	M	1623	20	20	0	8
1269	West Ham United	19	J. Collins	33	D	824	11	10	1	1
1292	West Ham United	20	A. Ayew	27	A	501	12	6	6	6
1294	West Ham United	28	J. Calleri	23	A	152	7	0	7	0
1291	West Ham United	15	D. Sakho	27	A	127	2	2	0	2
1270	West Ham United	21	A. Ogbonna	28	D	1666	19	19	0	1
1278	West Ham United	8	C. KouyatÃ©	27	M	1600	18	18	0	2
1281	West Ham United	14	Pedro Obiang	24	M	1186	15	14	1	3
1263	West Ham United	1	D. Randolph	29	G	990	11	11	0	0
1264	West Ham United	13	AdriÃ¡n	30	G	990	11	11	0	0

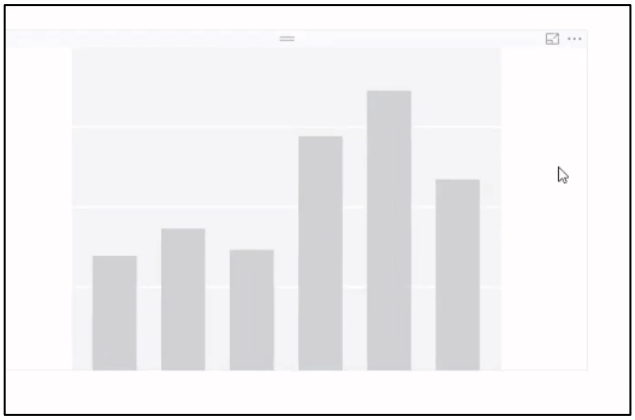
5.- En el panel de componentes se mostrarán los campos cargados del archivo:



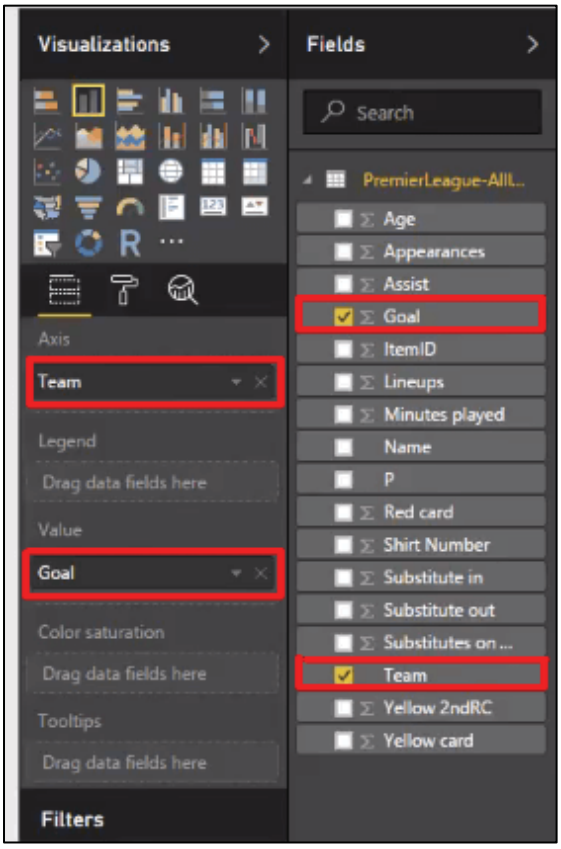
6.- En la sección Visualizations, seleccionar un componente para desplegar los datos, por ejemplo una gráfica de barras:



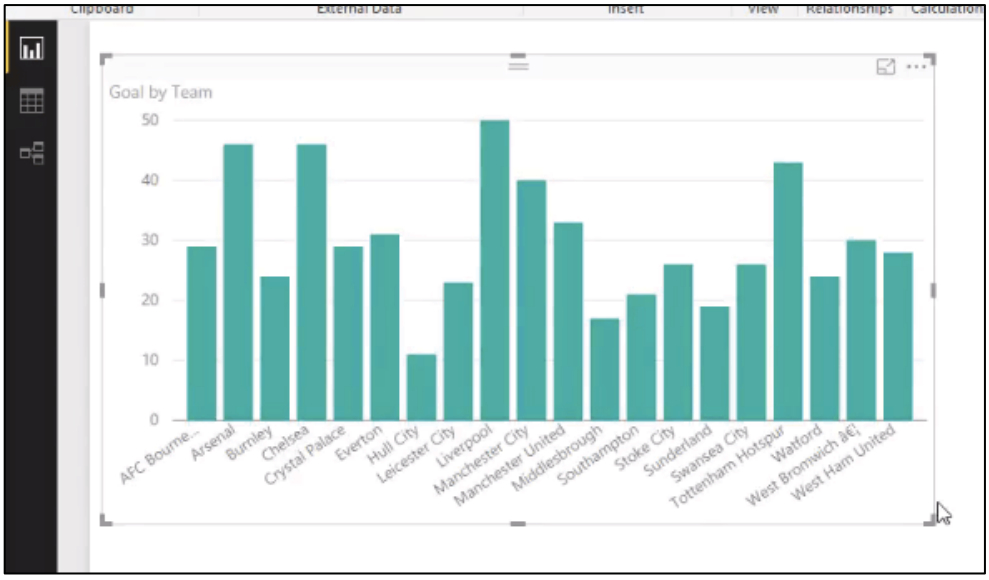
7.- En el editor se mostrará la gráfica vacia:



8.- Para llenar la gráfica es necesario arrastrar los campos a visualizar en los recuadros de Axis y Value. Los ejes (Axis) van a desplegar el valor de los equipos (Team) y los valores (Value) van a mostrar sus goles (Goal):



8.- En el editor se muestra el tablero con los datos del usuario:



9.- Como Power BI es una aplicación que se administra por medio de una aplicación (escritorio o movil), el tablero debe publicarse hacia su servidor. En este caso basta con dar click en Publish. (El uso de Power BI requiere previamente haber registrado una cuenta de Microsoft):