

Fundamentos Tecnológicos para el Tratamiento y  
Análisis de Datos

---

# Tema 4. Inteligencia de negocios

# Índice

## Esquema

### Ideas clave

- 4.1. Introducción y objetivos
- 4.2. Inteligencia de negocios
- 4.3. Almacén de datos
- 4.4. Modelado del almacén de datos y cubos de datos
- 4.5. Arquitecturas OLAP
- 4.6. Consultas en almacén de datos
- 4.7. Conexión con herramientas de inteligencia de negocio y lenguajes de programación
- 4.8. Referencias bibliográficas

### A fondo

Data warehouse 4u

### Test

Inteligencia de negocios	
Inteligencia de negocios vs. analítica de negocios	Almacén de datos y OLAP
<ul style="list-style-type: none"><li>- Ambas metodologías se utilizan para mejorar la <b>toma de decisiones</b> en la organización.</li></ul> <p>Inteligencia de negocios</p> <ul style="list-style-type: none"><li>- Técnicas de recoger y entender datos del pasado y usar ese conocimiento en la toma de decisiones. ¿Qué sucedió?</li></ul> <p>Analítica de negocios</p> <ul style="list-style-type: none"><li>- Conjunto de técnicas (algoritmos predictivos y modelos estadísticos) que le permiten a la organización predecir posibles eventos o resultados. ¿Qué sucederá?</li></ul>	<p>Almacén de datos</p> <ul style="list-style-type: none"><li>- Arquitecturas y herramientas para sistemáticamente organizar, entender y analizar los datos.</li><li>- Se caracteriza por estar orientado a <b>tema, integrado</b>, considerar la <b>variación con el tiempo</b> de los datos y ser <b>no volátil</b>.</li></ul> <p>Cubos OLAP</p> <ul style="list-style-type: none"><li>- Modelos de datos multidimensionales.</li><li>- Un cubo OLAP está definido por sus dimensiones y los datos numéricos que contiene.</li><li>- Operaciones: <i>Slice, dice, Rotar, Roll-up, Drill-down</i>, etc.</li></ul> <p>Modelos de datos OLAP</p> <ul style="list-style-type: none"><li>- <b>Estrella</b>. Gran tabla central y un conjunto de tablas auxiliares menores.</li><li>- <b>Copo de nieve</b>. Derivación del modelo estrella eliminando la redundancia presente.</li></ul> <p>Arquitecturas OLAP</p> <ul style="list-style-type: none"><li>- ROLAP. Basada en BBDD relacionales. Escalables.</li><li>- MOLAP. Basada en <i>arrays</i> multidimensionales. Eficientes.</li><li>- Híbridas. Combinación de las anteriores.</li></ul>
Arquitectura para la Inteligencia de negocios	
<p>Capa primera: Almacén de dato</p> <ul style="list-style-type: none"><li>- Repositorio de los datos <b>separado físicamente</b> de las bases de datos operacionales.</li><li>- Conjunto de <b>datos heterogéneos</b> que proceden fuentes externas de datos. A estos datos hay que aplicarles un proceso de <b>extracción, transformación y carga (ETL)</b> en el almacén de datos.</li></ul> <p>Capa segunda: Servidores OLAP (<i>On-Line Analytical Processing</i>)</p> <ul style="list-style-type: none"><li>- Capa que implementa los denominados <b>cubos de datos</b> de información.</li></ul> <p>Capa segunda: Interfase con el usuario</p> <ul style="list-style-type: none"><li>- Herramientas para realizar consultas, generar informes o detectar tendencia.</li></ul>	

## 4.1. Introducción y objetivos

En los temas anteriores se han presentado las bases de datos relacionales, estas son la tecnología básica con la que se implementan las bases de datos operacionales. Entendemos por **bases de datos operacionales** aquellas que permiten el trabajo diario de las organizaciones, son las encargadas de hacer posible las transacciones en línea y las consultas sencillas.

Las transacciones en línea se refieren a las operaciones de compras, ventas, pagos, altas, etc., que se realizan diariamente en la organización y que se deben procesar y almacenar. Por otro lado, las consultas sencillas se refieren a consultas sobre datos actuales. Por ejemplo, un cliente puede utilizar este tipo de bases de datos para consultar el *stock* de cierto producto en venta.

Las bases de datos operacionales están diseñadas para asegurar las transacciones en línea y proporcionar consultas de datos actuales.

Sin embargo, en este tema se presenta un segundo nivel en el sistema de información de la organización. La **inteligencia de negocios** es una infraestructura pensada para dar un servicio de un nivel superior, se intenta ayudar a las personas de la organización que tienen que tomar decisiones.

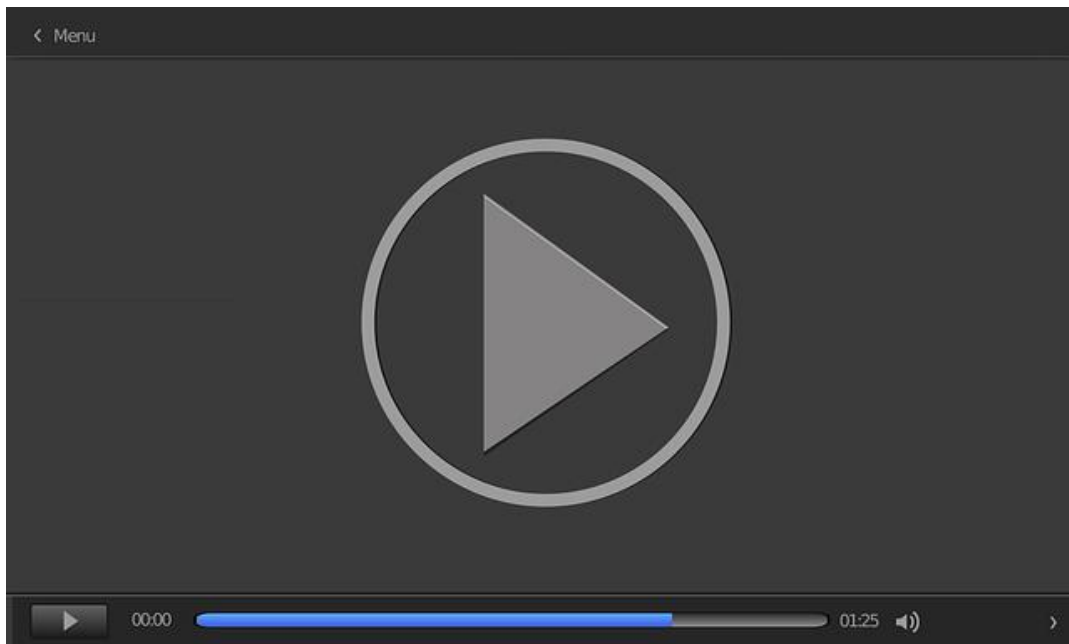
Para ello, se genera una infraestructura diferenciada de la operacional que permita realizar consultas complejas en datos históricos. Por ejemplo, ver el patrón de compras en los últimos cinco años de cierto grupo de clientes o calcular las ventas agregadas de los últimos diez años de las distintas regiones donde opera la organización son consultas que pueden ayudar en la toma de decisiones.

La infraestructura de la inteligencia de negocios está diseñada para almacenar y organizar grandes cantidades de datos históricos y poder realizar consultas complejas en tiempos de respuesta bajos.

En este momento, el alumno podría tener la duda siguiente: ¿por qué no implementar las consultas mediante, por ejemplo, SQL en las bases de datos operacionales? Efectivamente, a nivel técnico es posible. Sin embargo, hay que tener en cuenta que cualquier consulta en una base de datos consume recursos (tiempo, computación, ancho de banda, etc.). Por lo que, si el número de consultas crece en volumen, se podría llegar a entorpecer la operativa diaria de los sistemas, lo que conllevaría consecuencias negativas como retardos en responder a clientes, pérdida de clientes por espera, etc.

Objetivos que se pretenden conseguir:

- ▶ Entender qué es la inteligencia de negocios.
- ▶ Conocer la infraestructura tecnológica que hace posible la inteligencia de negocios.
- ▶ Entender el modelo y la estructura de un almacén de datos.
- ▶ Diferenciar entre tecnologías OLTP y OLAP.



Inteligencia de Negocios

---

Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=d9ddcbd7-3264-4477-bea0-b16c0095a284>

---

## 4.2. Inteligencia de negocios

El *business intelligence* (BI), que se traduce como «inteligencia de negocios», es un proceso de análisis y exploración de la información estructurada de la empresa (con frecuencia almacenada en un *data warehouse* o almacén de datos). El objetivo es **detectar tendencias o patrones del pasado**, a partir de los cuales derivar ideas y extraer conocimiento para mejorarla empresa.

El proceso de la inteligencia de negocios incluye la comunicación de los resultados y la ejecución de los cambios. Las áreas que abarcan por lo general son clientes, proveedores, productos, servicios y competidores.

Las **soluciones** de la inteligencia de negocios son una herramienta que ayuda en la toma de decisiones de la empresa. Contribuyen, a nivel interno, para apoyar la gestión del personal y, a nivel externo, para producir ventajas sobre los competidores.

Los resultados de la inteligencia de negocios son útiles para aquellas personas de la empresa que tienen que tomar decisiones. Dependiendo del tipo de negocio, se deben hacer las preguntas necesarias para responder y establecer el modelo de inteligencia de negocios que mejor se adapte.

### Inteligencia de negocios vs. analítica de negocios

La gestión de la empresa está fundamentada en la toma de decisiones más apropiada para cumplir con los objetivos del negocio, satisfacer las necesidades de los clientes y empleados y mantener o mejorar la calidad de los productos. Con el avance en las tecnologías de la información y las comunicaciones, el aumento en la capacidad de almacenamiento de datos ha dado paso a **nuevas metodologías**, tales como la inteligencia de negocios y la analítica de negocios, que ayudan y facilitan el proceso de toma de decisiones.

Antes se ha establecido que la **inteligencia de negocios** (BI) es un instrumento de apoyo a la toma de decisiones, basada en información precisa y oportuna, almacenada previamente en la organización para garantizar la generación del conocimiento necesario que permita seleccionar la alternativa que sea más conveniente para el éxito de la empresa.

Por otro lado, la **analítica de negocios** es un conjunto de técnicas (algoritmos predictivos y modelos estadísticos) que permiten a la organización predecir posibles eventos o resultados. Esto es, se enfoca en el análisis futuro, en función de la información de la empresa y en modelos predictivos para apoyar la toma de decisiones y mejorar los procesos y, por ende, la competitividad del negocio.

En resumen, se puede entender la inteligencia de negocios como las técnicas para recoger y entender los datos del pasado, mientras que la analítica de negocios permite alcanzar una visión más clara del futuro. Ambas metodologías se pueden **complementar** para construir un análisis minucioso de la actividad y futuro de la empresa, con el propósito de mejorar la toma de decisiones.

Mientras que la inteligencia de negocios responde a la pregunta ¿Qué sucedió?, la analítica de negocios responde a ¿Por qué sucedió, volverá a pasar? La inteligencia de negocios incluye informes, monitorización automatizada, alertas, tableros y cuadros de mando integral; la analítica de negocios, alternativamente, incluye análisis estadísticos cualitativos y cuantitativos, minería de datos, modelado predictivo y pruebas multivariantes. Cuando escuchas el término **inteligencia empresarial**, normalmente engloba toda la inteligencia y la analítica de negocios.

Este tema está dedicado a estudiar la arquitectura que hace posible la inteligencia de negocios. Las técnicas y herramientas usadas en la analítica de negocios se estudiarán en profundidad en otra asignatura. Es interesante puntualizar que una infraestructura de la primera puede servir como infraestructura base para la implementación de la segunda.

## Arquitectura en tres capas de un sistema de información para la inteligencia de negocios

La figura 1 muestra un esquema estándar de un sistema de información diseñado para la inteligencia de negocios. A continuación, se comenta desde un punto de vista general:

### ETL

El sistema tiene como entrada un conjunto de datos heterogéneos que proceden de las bases de datos operacionales de la organización, de ficheros planos, de registros de transacciones o de otras fuentes externas de datos. A estos datos hay que aplicarles un proceso de **extracción, transformación y carga (ETL)**:

- ▶ La extracción se aplica a un conjunto heterogéneo de fuentes de datos. A la extracción le sigue una fase de limpiado de datos que detecta errores en ellos y los subsana cuando es posible.
- ▶ La transformación homogeneiza los datos con el fin de guardarlos en el almacén de datos.
- ▶ La fase de carga incluye todas las operaciones necesarias para incorporar los datos al almacén de datos. Entre esas operaciones se tienen ordenaciones, agregaciones, consolidaciones, creación de vistas, chequeos de integridad o creación de índices y particiones.
- ▶ Finalmente, existe una fase de actualización que permite incorporar nuevos datos desde las fuentes de datos.

### Primera capa: el almacén de datos

La primera capa del sistema es propiamente el almacén de datos. Se suele implementar separado físicamente de las bases de datos operacionales. Contiene:

- ▶ El repositorio de **metadatos**, información sobre los datos incluidos en el almacén.

- ▶ Los denominados **data marts**, que se pueden considerar almacenes de datos especializados en ciertos temas concretos (cliente, producto, etc.) o diseñados para dar servicio a cierta área o departamento de la organización.

### **Capa intermedia: OLAP**

La capa intermedia es el denominado servidor OLAP (*online analytical processing*), que implementa los denominados cubos de datos de información.

### **Tercera capa: interfase**

Finalmente, la última capa es la que se comunica o hace de interfase con el usuario de la infraestructura de inteligencia de negocios. Se incluyen herramientas para realizar consultas, generar informes o detectar tendencias. También es posible incluir técnicas de analítica de negocios como predictores temporales, segmentadores o agrupadores de entidades.

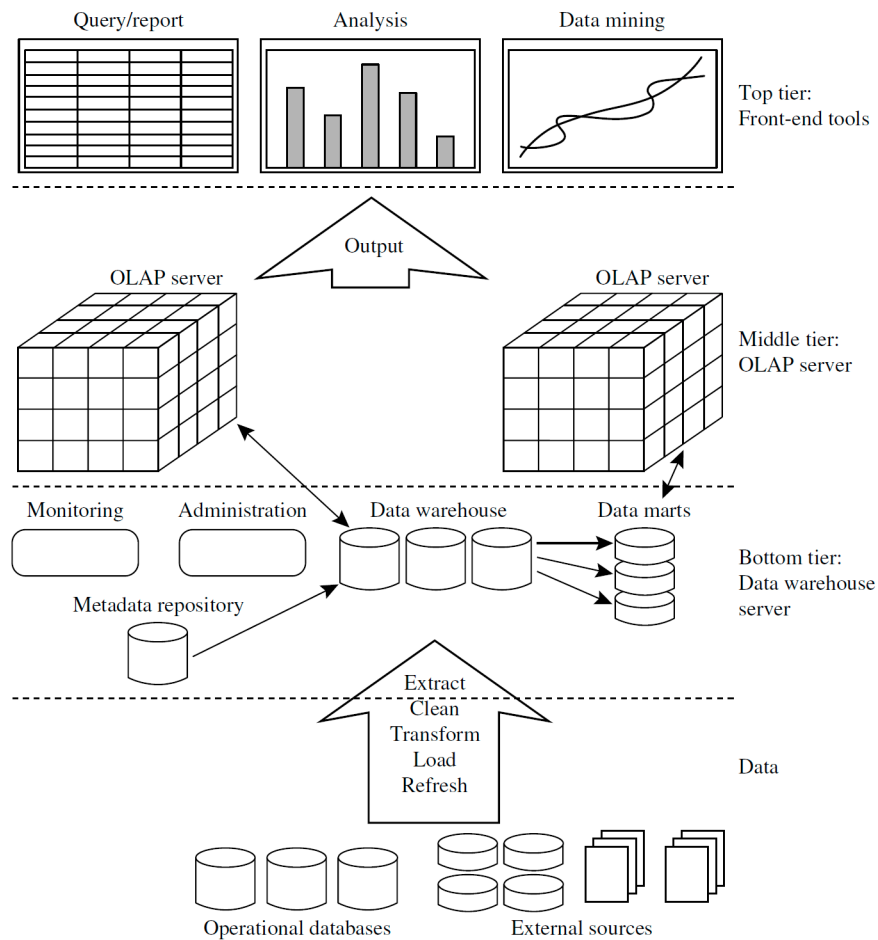


Figura 1. Esquema general de un sistema de información. Fuente: Han y Kamber (2012).

Desde el punto de vista de las aplicaciones empresariales, el almacén de datos tiene como entradas los datos utilizados por los sistemas de gestión empresarial (CRM, ERP, PLM, SCM y SRM) y genera salidas para las aplicaciones de ayuda en la toma de decisiones (por ejemplo, EIS y DSS).

## 4.3. Almacén de datos

El denominado almacén de datos (*data warehouse*) proporciona arquitecturas y herramientas a la inteligencia de negocios para organizar sistemáticamente, entender y analizar los datos con el fin de tomar decisiones estratégicas. Los sistemas de almacén de datos son herramientas muy importantes en el mundo competitivo y de rápida evolución en el que vivimos.

Como su nombre indica, es un repositorio de datos distinto a las bases de datos relacionales, los sistemas transaccionales y los sistemas de ficheros. Se caracteriza por estar orientado a un tema, estar integrado, considerar la variación con el tiempo de los datos y no ser volátil. A continuación, se explica cada una de estas características:

### Orientado a un tema

Se organiza en torno a temas como: cliente, suministrador, producto o ventas. No se organiza pensando en las operaciones que se realizan diariamente (transacciones). Está orientado a facilitar las tareas de análisis y modelado de los datos necesarios en los sistemas de toma de decisión.

### Integrado

Un almacén de datos se construye integrando diversas y heterogéneas fuentes de datos, tales como las bases de datos relacionales, ficheros planos o registros de transacciones *online*. Por tanto, el limpiado e integración de datos son necesarios para asegurar la consistencia en nombres, estructuras, atributos, etc.

### Variante con el tiempo

Los datos se almacenan para proporcionar una perspectiva histórica de estos. Cada estructura del almacén de datos contiene de forma explícita o implícita el elemento tiempo.

## No volátil

El almacén de datos está siempre físicamente separado de los datos usados para la operativa diaria de la empresa. Por tanto, no requiere procesamiento de transacciones, sistemas de recuperación ni mecanismos de control de la concurrencia (necesarios en los datos operativos). Normalmente solo requiere dos operaciones: la carga inicial de datos y el acceso a estos.

En resumen, se puede decir que el almacén de datos es la implementación física del modelo de datos usado por el sistema de ayuda a la toma de decisiones. Permite la integración de datos de múltiples fuentes para proporcionar consultas, informes y análisis.

Las organizaciones utilizan la información proporcionada por el almacén de datos para:

- ▶ Incrementar el conocimiento sobre los clientes y realizar un análisis de patrones de compra (preferencias en las compras, tiempo de compra, ciclos presupuestarios o de gastos).
- ▶ Reposicionamientos de productos y gestión de la cartera de productos, comparando rendimientos de ventas por trimestre, por año o por región geográfica con el objeto de afinar las estrategias de producción.
- ▶ Análisis de operaciones y localización de fuentes de ingresos.
- ▶ Gestión de las relaciones con los clientes.

## Diferencias entre almacén de datos y base de datos operacional

Es importante diferenciar entre las ampliamente conocidas bases de datos relacionales utilizadas para la operativa diaria de la empresa y el almacén de datos.

La principal tarea de un sistema de bases de datos operacional es realizar transacciones en línea (compras, ventas, altas, bajas, etc.) y procesar consultas (seleccionar un conjunto de datos almacenados en función de ciertos criterios). A este tipo de sistemas se les denomina **sistemas de procesamiento de transacciones en línea (OLTP)**. Este tipo de sistemas son los utilizados en la mayoría de las operaciones diarias realizadas por la organización como compras, inventario, fabricación, operaciones con banca, nóminas, registro de operaciones y contabilidad.

Por otro lado, el almacén de datos proporciona servicio a los usuarios encargados de realizar análisis de datos y toma de decisiones. Los datos se organizan y presentan en formatos acomodados a las necesidades específicas de los analistas. A este tipo de sistemas se les conoce como **sistemas de procesamiento analítico en línea (OLAP)**.

A continuación, se resume las principales diferencias entre los sistemas.

### Orientación al usuario o al mercado

- ▶ Los sistemas OLTP están orientados al **cliente** y están diseñados para procesar transacciones y consultas realizadas por empleados, clientes o informáticos.
- ▶ El sistema OLAP está orientado al **mercado** y se usa por los gestores, ejecutivos o analistas para analizar datos.

### Datos almacenados

- ▶ Los sistemas OLTP gestionan datos **actuales** y muy detallados.
- ▶ Los sistemas OLAP gestionan grandes cantidades de datos **históricos**, facilitando operaciones de agregación o suma. Se gestiona y almacena información con diferentes niveles de detalle. El objetivo es facilitar el manejo de datos para la toma de decisiones.

## Diseño de la base de datos

- ▶ Un sistema OLTP adopta normalmente el modelo entidad-relación (**E-R**) y un diseño orientado a aplicación.
- ▶ Un sistema OLAP adopta normalmente un **modelo tipo estrella** o similar y un diseño de base de datos orientado a tema.

## Patrones de acceso

- ▶ En los sistemas OLTP los accesos son principalmente **transacciones atómicas** (no divisibles) y pequeñas para actualizar información. Se requiere control de la concurrencia y mecanismos de recuperación ante un fallo en la transacción.
- ▶ En los sistemas OLAP la mayoría de las operaciones son de **lectura de datos**, aunque pueden ser consultas complejas.

Aunque se ha intentado aclarar las diferencias entre sistemas OLTP y OLAP, el alumno todavía podría tener cierta duda. Por ejemplo, podría pensar que las funcionalidades proporcionadas por un sistema OLAP se podrían implementar en un sistema OLTP ahorrando muchos recursos (no se utiliza un nuevo almacenamiento físico) y tiempo. Sin embargo, la principal razón por la que crear un sistema OLAP con **almacenamiento físico independiente** del OLTP es ayudar a tener un alto rendimiento en ambos sistemas.

Los sistemas OLTP están diseñados y optimizados para realizar transacciones o consultas sencillas, usando datos no procesados. Las consultas que necesitan los analistas, que pueden requerir datos agregados, cargarían con excesivo trabajo al sistema OLTP y bajar su rendimiento; limitando incluso su capacidad para realizar su principal trabajo, las transacciones.

Por otro lado, el diseño de un sistema OLAP facilita la realización de consultas complejas orientadas a un tema, por lo que la productividad de los analistas o

gestores se verá incrementada al disponer de un sistema ágil ante sus necesidades.

En resumen, como los dos sistemas proporcionan funcionalidades diferentes y requieren diferentes clases de datos, es razonable utilizar almacenamientos físicos diferentes.

Sin embargo, hay que indicar que algunos proveedores de tecnología OLTP están optimizando sus sistemas para aceptar consultas OLAP. En caso de equilibrar el rendimiento y proporcionar una solución a la necesidad de fuentes adicionales de datos por parte de OLAP, la separación del almacenamiento físico entre OLTP y OLAP puede decrecer.

La figura 2 ilustra la diferencia entre los sistemas OLTP y OLAP. Se puede observar también la diferencia entre el tipo de consulta de cada sistema. En los primeros, las consultas se refieren a datos concretos. Sin embargo, en los sistemas OLAP, las consultas requieren de cierto procesado, como sumas o agregados de datos. Realizar ese tipo de consultas en el sistema OLTP puede generar pérdidas de rendimiento importantes.

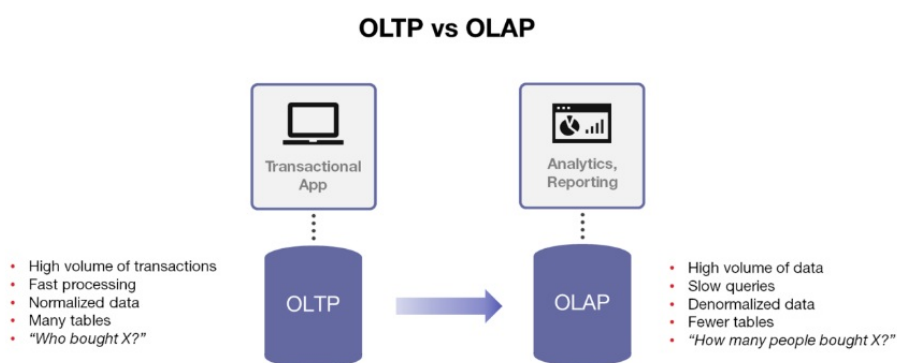


Figura 2. OLTP vs. OLAP. Fuente: <https://www.marklogic.com/blog/relational-databases-are-not-designed-for-mixed-workloads/>

## 4.4. Modelado del almacén de datos y cubos de datos

La inteligencia de negocios se basa en modelos de **datos multidimensionales**, los cuales se pueden visualizar mediante los llamados cubos de datos o cubos OLAP, donde cada dimensión representa una cara del cubo. En esta sección se presentarán los modelos de datos basados en los modelos estrella y copo de nieve.

### Cubos OLAP

Los cubos de datos son formas de ver o entender los datos incluidos en el almacén de datos. Un cubo está definido por sus dimensiones y los datos numéricos que contiene. Por ejemplo, en la figura 3 se puede observar un ejemplo de cubo OLAP. En este caso, el cubo está orientado al tema ventas y tiene tres dimensiones (localización, tiempo y producto).

Por cada localización, tiempo y producto considerados se tiene un valor numérico que representa la venta de ese producto en esa localización y en ese período temporal. Nótese que la dimensión tiempo está definida por cuatrimestres. Se podrían definir otros cubos OLAP donde la dimensión tiempo tuviese otro tipo de agregación (mensual, anual, etc.).

Por tanto, un cubo OLAP se crea en torno a un tema central y los valores numéricos correspondientes. Se pueden tener distintos cubos OLAP para distintos temas de interés. Las dimensiones de un cubo OLAP no están acotadas a tres, se puede disponer de **cubos n-dimensionales**.

Los cubos OLAP permiten realizar ciertas operaciones:

### Slice y dice

- ▶ La operación *slice* permite tomar uno de los valores numéricos de una dimensión construyendo un subcubo más sencillo, de una dimensión menor. Por ejemplo, a partir del cubo de la figura xx se puede construir un subcubo donde la dimensión tiempo se reduce, por ejemplo, a Q1.
- ▶ La operación *dice* también crea un subcubo, pero permitiendo seleccionar en varias dimensiones. Por ejemplo, se puede construir un subcubo con las localizaciones Toronto y Vancouver, los cuatrimestres Q1 y Q2 y los productos *computer* and *phone*. Sigue siendo un cubo de tres dimensiones, pero simplificado.

### Rotar o pivotar

Simplemente es cambiar la forma de visualización de los datos. Por ejemplo, se puede invertir el orden de los cuatrimestres.

### Roll-up

Operación de agregación. Por ejemplo, la dimensión localización se podría agregar a países, con lo que el nuevo cubo solo tendría dos valores en esa dimensión, USA y Canadá.

### Drill-down

Es una operación de desagregación. Por ejemplo, se podría crear un nuevo cubo donde la dimensión tiempo fuese mensual y no cuatrimestral.

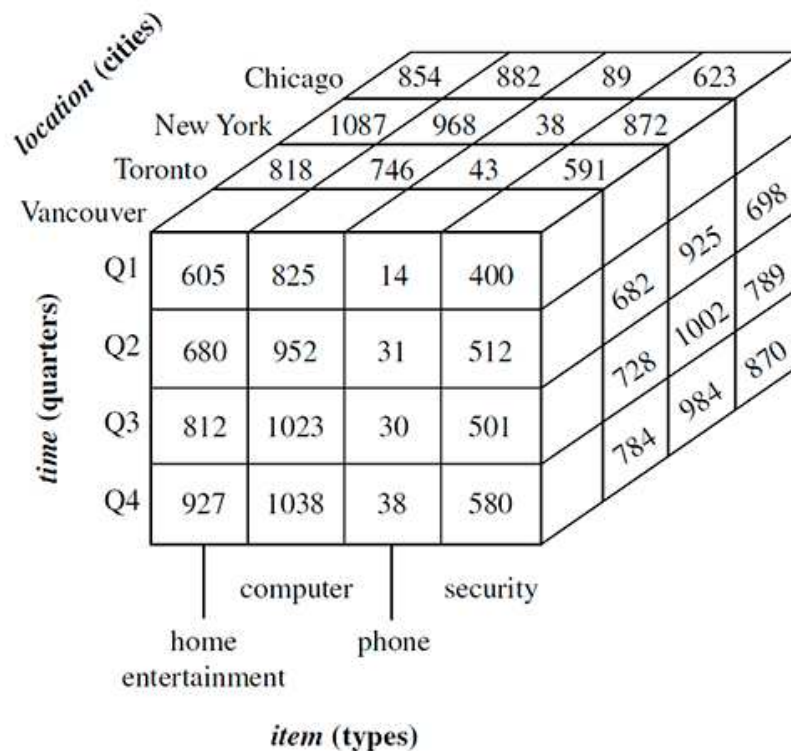


Figura 3. Cubo OLAP. Fuente: Han y Kamber (2012).

## Modelos de datos OLAP

Al igual que en las bases de datos operacionales, se utiliza el modelo entidad-relación como herramienta descriptiva. En el almacén de datos se usan los modelos estrella y copo de nieve.

### Modelo estrella

El modelo estrella se basa en tener una gran tabla central que contiene la mayor parte de los datos de forma no redundante y un conjunto de tablas auxiliares más pequeñas que corresponden a cada una de las dimensiones. En la figura 4 se puede observar un modelo estrella donde la tabla de las ventas es el tema central, además del tiempo, producto y localización.

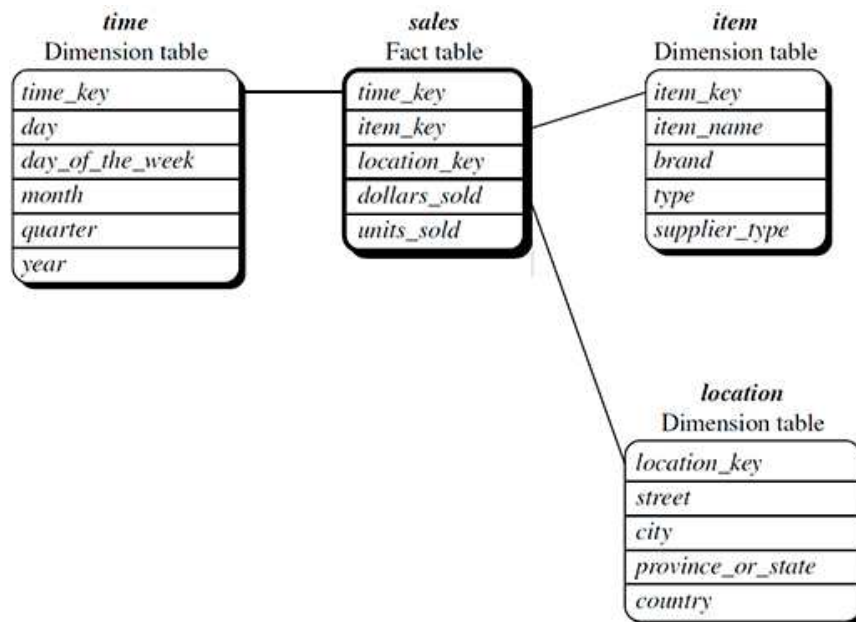


Figura 4. Modelo estrella. Fuente: Han y Kamber (2012).

## Modelo copo de nieve

El modelo copo de nieve es una derivación del modelo estrella. El objetivo es eliminar la redundancia presente en las tablas del modelo anterior. Por ejemplo, en las tablas de producto y localización había cierta redundancia que se elimina añadiendo nuevas tablas, tal y como se ve en la figura 5, que representa el modelo en copo de nieve equivalente.

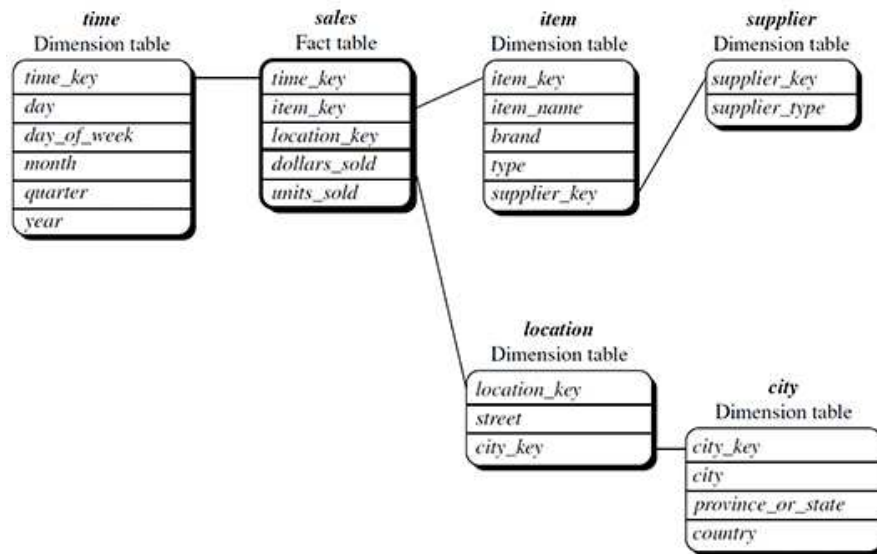


Figura 5. Modelo de copo de nieve. Fuente: Han y Kamber (2012).

## 4.5. Arquitecturas OLAP

La tecnología OLAP presenta los datos a sus usuarios como cubos multidimensionales. A los usuarios no les interesa conocer cómo están almacenados, sino realizar consultas en ellos. Por otro lado, esos datos efectivamente deben de almacenarse con cierta estructura. Las opciones disponibles son varias:

- ▶ **OLAP relacional (ROLAP).** En este caso se utilizan bases de datos relacionales como almacén de datos. Se dispone de un *software* OLAP intermedio que realiza la conversión a cubos de información. Este tipo de tecnología facilita la escalabilidad (tamaño) del sistema OLAP, pero presenta limitaciones en el rendimiento de las consultas.
- ▶ **OLAP multidimensional (MOLAP).** En este caso se utilizan almacenamientos multidimensionales basados en *arrays*. Se mapean directamente los datos a estructuras de cubos. Esto acelera ciertas operaciones realizadas en los sistemas OLAP, como las agregaciones.
- ▶ **Sistemas híbridos.** Los OLAP híbridos combinan tecnologías ROLAP y MOLAP. Intenta beneficiarse de la escalabilidad de ROLAP y de la rapidez de computación de MOLAP.

## 4.6. Consultas en almacén de datos

En el tema anterior analizamos el lenguaje de consultas SQL y lo aplicamos en una base de datos simulada. Existe una versión de la misma en su variante *datawarehouse* o ‘almacén de datos’, compuesta por diferentes tablas de hechos y tablas de dimensiones. En este apartado nos centraremos en utilizar un almacén existente, pero debe hacerse mención a la importancia de la planificación y el contexto empresarial para desarrollar un almacén de datos óptimo. Una vez se instala la base de datos AdventureWorksDW2017 podemos observar que existen diversas tablas cuya información contenida es la misma que en la versión OLTP, aunque su estructura es distinta. Analicemos tres escenarios presentes: Inventario, Ventas y Finanzas.

### Escenario de inventario

En relación al inventario existente en AdventureWorks, tenemos disponible la tabla de hechos `dbo.FactProductInventory`, en la que la clave primaria son el `ProductKey` y el `DateKey`, lo que permite conectar con las dimensiones de fechas (`DimDate`) y producto (`DimProduct` y `DimProductSubcategory`). En la tabla de hechos tenemos la información disponible del balance, movimientos, coste por unidad y día. Por otro lado, las dimensiones nos ofrecen un filtro robusto en términos de fechas (día, semana, mes, año, etc...) y categorías de producto (nombre de artículo y grupo).

Mediante la conexión vía dimensiones podemos realizar **diversos análisis**, como, por ejemplo:

- ▶ Variación año a año de los niveles de *stock* por tipo de producto en distintos niveles temporales (días, semanas, meses, etc.).
- ▶ Analizar los movimientos de *stock* de las diferentes unidades en función del parámetro tiempo.

- Analizar la tendencia de variaciones a lo largo de los años por grupo de producto.

Podemos realizar las consultas a las tablas a través de SQL:

```
-- Tabla de Hechos
SELECT * FROM dbo.FactProductInventory
-- Tablas de Dimensiones
SELECT * FROM dbo.DimDate
SELECT * FROM dbo.DimProduct
SELECT * FROM dbo.DimProductSubcategory
-- Combinación
SELECT * FROM dbo.FactProductInventory AS ft
    INNER JOIN dbo.DimDate AS dd ON dd.DateKey = ft.DateKey
    LEFT JOIN dbo.DimProduct AS dp ON dp.ProductKey = ft.ProductKey
    LEFT JOIN dbo.DimProductSubcategory AS dsp ON dsp.ProductSubcategoryKey=
        dp.ProductSubcategoryKey;
```

A continuación, podemos observar el diagrama del presente almacén de datos:

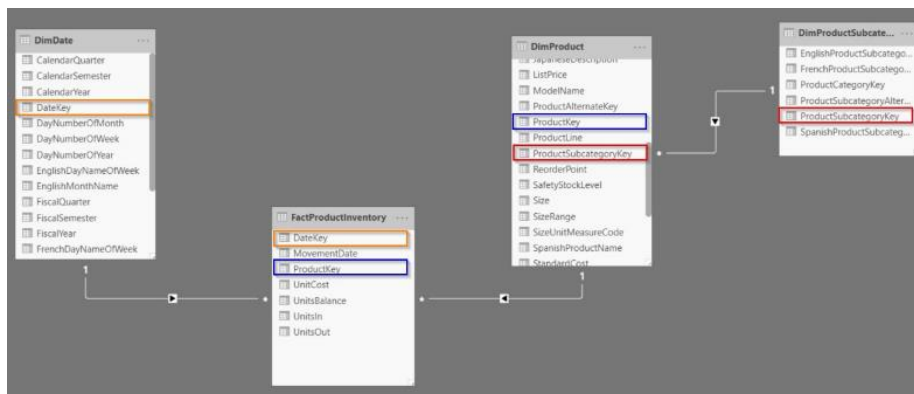


Figura 6. Modelo Inventario.

Es importante considerar las **cardinalidades** entre dichas relaciones. Las dimensiones fecha y producto están relacionadas uno a muchos con la tabla de hechos. Asimismo, la dimensión de subcategoría de producto está conectada con la dimensión producto de la misma forma, uno a muchos.

## Escenario financiero

Este almacén de datos se centra en dar respuesta al **apartado financiero y**

**contable** de la empresa, con sus correspondientes montantes por año. Volvemos a reutilizar la dimensión de fechas para construir nuestras operaciones en inteligencia temporal. Respecto a la tabla de hechos `dbo.FactFinance`, esta contiene información respecto a los asientos contables, escenario, departamento y división. Para poder explotar la información necesitamos utilizar distintas dimensiones, de forma que la conectemos a las claves ajenas de la tabla de hechos. Tenemos a nuestra disposición las dimensiones de escenario (`dbo.DimScenario`), referentes a la situación actual, presupuestos y predicción; la dimensión de organización (`dbo.DimOrganization`), compuesta por las divisiones de la empresa; la dimensión de departamento (`dbo.DimDepartmentGroup`), y finalmente la dimensión contable (`dbo.DimAccount`), formada por la información referente a los asientos contable y a la moneda.

El presente almacén de datos nos permitirá, entre otras operaciones:

- ▶ Analizar y automatizar la contabilidad general y su segmentación por divisiones.
- ▶ Comparar entre años los diferentes indicadores de rendimiento (KPIs) en función de la división y del tipo de departamento.
- ▶ Planificar los presupuestos para años venideros en función del análisis cuantitativo realizado con datos históricos.

```
-- Tabla de Hechos
SELECT * FROM dbo.FactFinance
-- Tablas de Dimensiones
SELECT * FROM dbo.DimDate
SELECT * FROM dbo.DimScenario
SELECT * FROM dbo.DimOrganization
SELECT * FROM DimDepartmentGroup
SELECT * FROM dbo.DimAccount
--Combinación
SELECT *
FROM dbo.FactFinance AS ff
    INNER JOIN dbo.DimDate AS dd ON dd.DateKey = ff.DateKey
    LEFT JOIN dbo.DimScenario AS ds ON ds.ScenarioKey = ff.ScenarioKey
    LEFT JOIN dbo.DimOrganization AS do ON do.OrganizationKey = ff.OrganizationKey
    LEFT JOIN dbo.DimDepartmentGroup AS ddg ON ddg.DepartmentGroupKey =
ff.DepartmentGroupKey
```

LEFT JOIN dbo.DimAccount AS da ON da.AccountKey = ff.AccountKey

A continuación, podemos observar el **diagrama** del presente almacén de datos:

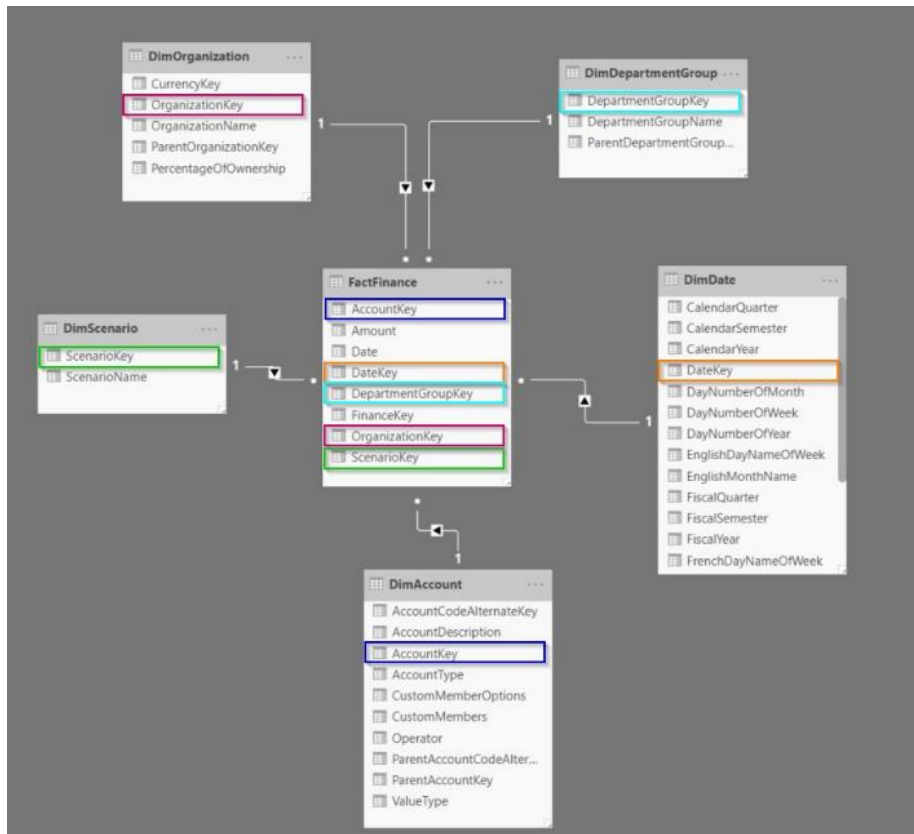


Figura 7. Modelo Financiero.

Al igual que en el caso anterior, es importante comprender las **cardinalidades** asociadas. Para que el funcionamiento sea óptimo, cada dimensión debe estar relacionada con la tabla de hechos con cardinalidad uno a muchos.

## Escenario de ventas

La principal fuente de ingresos de una empresa es siempre la venta de productos o la prestación de servicios. En este caso simulado vamos a utilizar el almacén de datos referente a ventas. La tabla de hechos que contiene toda la información imprescindible referente a ventas es `dbo.FactInternetSales`. Tenemos a nuestra

disposición información relevante como factura, producto, cantidad, promoción, precio final y la fecha de los pedidos. Debemos considerar la estructura de la presente tabla, puesto que la clave primaria está compuesta por dos atributos: SalesOrderNumber («factura») y SalesOrderNumberLine («línea de Factura»). Por su parte, las claves ajenas disponibles nos proporcionan enlaces a las dimensiones de fecha ( dbo.DimDate ), cliente ( dbo.DimCustomer ), territorio de venta ( dbo.DimSalesTerritory ), promoción ( dbo.DimPromotion ), producto ( dbo.DimProduct ) y moneda ( dbo.DimCurrency ).

El caso de la dimensión fecha es especial, puesto que podemos conectarla a través de tres claves: fecha de pedido, de vencimiento y de envío. Podemos establecer y cambiar las conexiones en función de la herramienta (algunas permiten comandos de selección) o de nuestras necesidades de información. La diversa combinación de dimensiones posibles permite realizar un análisis cuantitativo completo del proceso de venta, lo que posibilitará conseguir *insights* relevantes para la toma de decisiones de cara a distintos departamentos, así como a la empresa en general, entre los cuales podemos destacar:

- ▶ Análisis de venta de artículos por región y tipología de producto más vendida.
- ▶ Comparación anual, mensual y semanal entre las ventas de artículos, lo que posibilita descubrir patrones estacionales de venta.
- ▶ Segmentación por moneda y homogeneización de las cifras de venta.
- ▶ Análisis de las promociones realizadas por tipo de producto y territorio.
- ▶ Segmentación y analítica de cliente, tanto de tiendas como de individuos.

```
-- Tabla de Hechos
SELECT * FROM dbo.FactFinance
-- Tablas de Dimensiones
```

```

SELECT * FROM dbo.DimDate
SELECT * FROM dbo.DimScenario
SELECT * FROM dbo.DimOrganization
-- Tabla de Hechos
SELECT * FROM dbo.FactInternetSales
-- Tabla de Dimensiones
SELECT * FROM dbo.DimDate
SELECT * FROM dbo.DimCustomer
SELECT * FROM dbo.DimPromotion
SELECT * FROM dbo.DimProduct
SELECT * FROM dbo.DimCurrency
SELECT * FROM dbo.DimSalesTerritory
-- Combinación
SELECT * FROM dbo.FactInternetSales AS fca
-- 3 Posibilidades respecto a la dimensión Fecha
--INNER JOIN dbo.DimDate as dd ON dd.DateKey = fca.DueDateKey
INNER JOIN dbo.DimDate AS dd ON dd.DateKey = fca.OrderDateKey
--INNER JOIN dbo.DimDate as dd ON dd.DateKey = fca.ShipDateKey
LEFT JOIN dbo.DimCustomer AS dc ON dc.CustomerKey = fca.CustomerKey
LEFT JOIN dbo.DimPromotion AS dp ON dp.PromotionKey = fca.PromotionKey
LEFT JOIN dbo.DimProduct AS dpr ON dpr.ProductKey = fca.ProductKey
LEFT JOIN dbo.DimCurrency AS dcu ON dcu.CurrencyKey = fca.CurrencyKey
LEFT JOIN dbo.DimSalesTerritory AS dst ON dst.SalesTerritoryKey = fca.SalesTerritoryKey

```

Para visualizar mejor la relación existente entre las diferentes tablas podemos conceptualizar el siguiente diagrama:

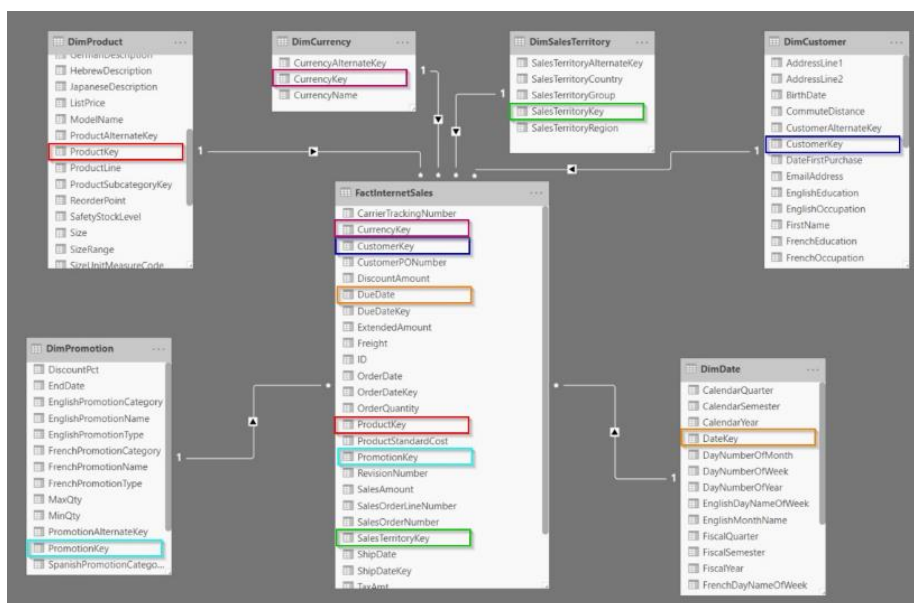


Figura 8. Modelo Ventas.

A lo largo del presente epígrafe hemos visto diversos almacenes de datos, todos ellos centrados en un tema. La base de datos actual dispone de algunos más referentes a los *call centers*, las encuestas, los tipos de cambio, las reventas y la descripción de productos.

### 4.7. Conexión con herramientas de inteligencia de negocio y lenguajes de programación

Como analistas de *business intelligence* nuestro trabajo consiste tanto en la obtención, limpieza y carga (ETL) de datos como en el análisis cuantitativo y cualitativo de los mismos. En temas anteriores hemos desarrollado las nociones y habilidades necesarias para extraer y transformar los datos utilizando bases de datos relacionales, presentes en la gran mayoría de empresas. La exportación de dichos datos a diferentes aplicaciones BI o lenguajes de programación, tanto transformados como en bruto, es el siguiente paso en cualquier proyecto de inteligencia de negocio. Si nos referimos estrictamente a BI, tenemos diferentes herramientas de análisis muy completos, como pueden ser PowerBI, Tableau o QlikView. Por otro lado, para llevar a cabo análisis estadísticos mediante técnicas complejas y algoritmos es necesario recrear una conexión con lenguajes de programación, como R o Python, o incluso con *software* de análisis avanzado, como Matlab.

#### PowerBI

Esta herramienta de Microsoft es muy versátil y completa. Permite configurar nuestro ETL de diversas fuentes de datos y aplicarles potentes fórmulas de cálculo (DAX) para crear informes y cuadros de mandos significativos.

Para conectarnos a una base de datos podemos elegir entre SQL Server, MySQL y Oracle entre otros. A modo de ejemplo, podemos plantear nuestra conexión desde el punto de vista de SQL Server. Para ello accedemos a «Obtener datos», seleccionamos «SQL Server» e indicamos el servidor de conexión y la base de datos a la que deseamos acceder.

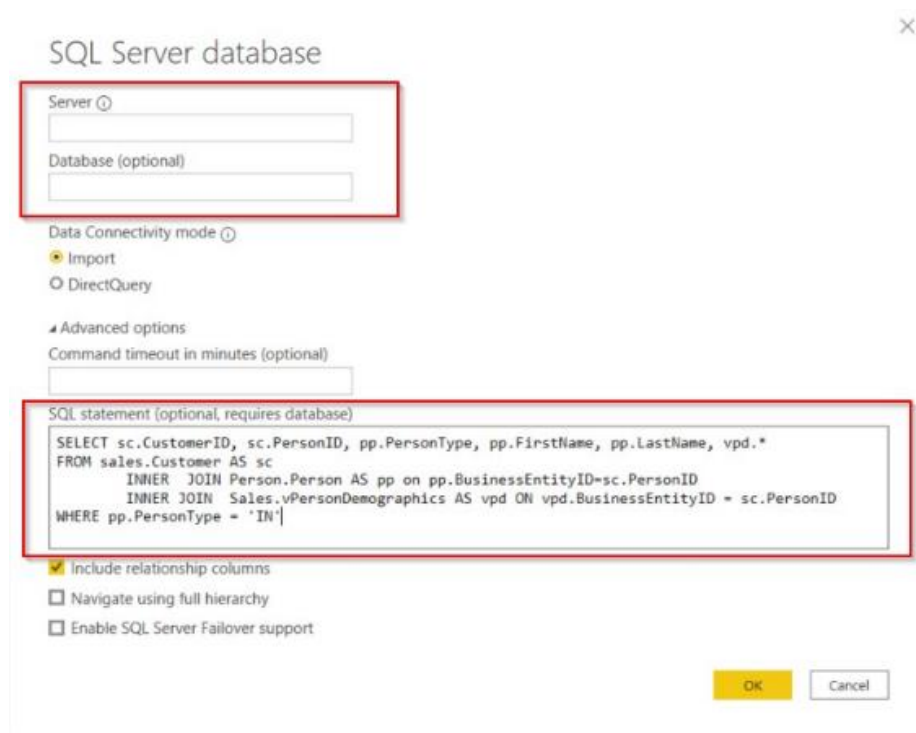


Figura 9. Conexión SQL Server con PowerBI.

Tenemos dos posibilidades de carga de datos:

- ▶ «**Import**»: importamos los datos directamente en PowerBI almacenándolos en la memoria.
- ▶ «**Direct Query**»: se establece una conexión directa de refresco continuo. En estos casos las transformaciones son más limitadas en comparación con la versión «Import».

Por otra parte, tenemos la opción de especificar una consulta determinada (figura 9) o de presionar «OK» y elegir las tablas acordes a nuestras necesidades de información (figura 10). En ambos casos tendríamos a nuestra disposición la información relevante en PowerQuery, donde realizaríamos las transformaciones y modificaciones que consideremos oportunas.

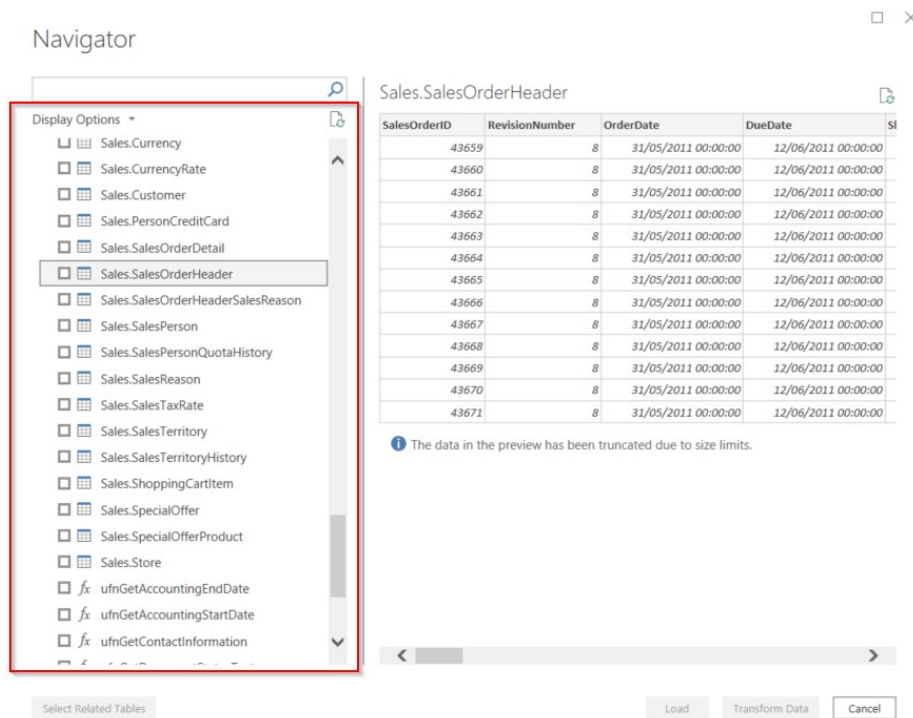


Figura 10. Conexión genérica SQL Server con PowerBI.

## Lenguajes de programación

Tanto R como Python son lenguajes orientados a los objetos más utilizados entre los analistas de datos e investigadores. Al ser de código abierto poseen diversas librerías enfocadas al análisis cuantitativo y cualitativo, con una amplia variedad de técnicas y algoritmos, además de grandes paquetes de visualización de la información. En estos casos establecer una conexión con la base de datos e importar nuestra información en un *dataframe* es de vital importancia de cara a continuar con nuestro proyecto.

Para el caso de **R** tenemos diversas librerías disponibles, tales como ODBC y RODBC. Una vez abierto nuestro *script* de R, importamos e instalamos la librería. A continuación, procedemos a crear una conexión mediante `odbcDriverConnect` especificando el tipo de conexión, el servidor y la base de datos. Una vez establecida la conexión, creamos un *dataframe* y ejecutamos el comando `sqlQuery`

especificando la conexión y la consulta SQL. Para cerrar la conexión con la base de datos ejecutamos el comando `odbcCloseAll()`.

```
## Librería RODBC
library(RODBC)

## Establecemos la conexión:
## 1. El driver (en caso de SQL Server es: SQL Server Native Client 11.0)
## 2. El servidor es: (localdb)\ + nombre de la instancia que le dimos en la línea de comandos
## 3. Base de datos: AdventureWorks2017
## 4. Incluir trusted_connection

cn <- odbcDriverConnect(connection="Driver={SQL Server Native Client 11.0};
server=(localdb)\miinstancia;
database=AdventureWorks2017;
trusted_connection=yes;")

dataframesql <- sqlQuery(cn, "SELECT * FROM Sales.SalesOrderHeader AS SOH")

print(dataframesql)
## Cerrar la conexión
odbcCloseAll()
```

Con **Python**, la operativa es similar. En este caso debemos importar las librerías Pandas y Pyodbc y, al igual que con R, necesitamos especificar los parámetros de conexión mediante `pyodbc.connect`. Debemos indicar el controlador, el servidor y la base de datos. Una vez creada la misma, procedemos a utilizar la función `read_sql` de Pandas especificando la consulta y la conexión.

```
#Importamos las librerías Pandas y Pyodbc
import pandas as pd
import pyodbc

#Creamos la conexión
cnxn = pyodbc.connect("Driver={SQL Server Native Client 11.0};"
"Server=localhost\SQLSINSTANCE;"
"Database=AdventureWorks2017;"
"Trusted_Connection=yes;")

#Almacenamos los datos en un dataframe

dataframe = pd.read_sql('SELECT * FROM Sales.SalesOrderHeader AS SOH', cnxn)
```

```
print(dataframe)
```

### 4.8. Referencias bibliográficas

Han, J. y Kamber, M. (2012). *Data mining: Concepts and techniques* (3rd ed). San Francisco: Morgan Kaufmann Publishers.

### Data warehouse 4u

Data Warehouse 4u. Página web oficial. <https://www.datawarehouse4u.info/>

Página web dedicada a los almacenes de datos. Incluye algunos artículos aclaratorios.

1. La inteligencia de negocios y la analítica de negocios:
  - A. Son incompatibles pues tienen objetivos estratégicos diferentes.
  - B. Necesitan almacenamiento físicamente separado.
  - C. Se complementan al tener objetivos estratégicos compatibles.
  - D. Son exactamente lo mismo.
  
2. Una arquitectura en tres capas de la inteligencia de negocios normalmente incluye:
  - A. El conjunto de bases de datos operacionales, el servidor OLAP y un conjunto de interfases con los usuarios para realizar consultas e informes.
  - B. El conjunto de bases de datos operacionales, el almacén de datos y un conjunto de interfases con los usuarios para realizar consultas e informes.
  - C. El conjunto de bases de datos operacionales, el almacén de datos y el servidor OLAP.
  - D. El almacén de datos, el servidor OLAP y un conjunto de interfases con los usuarios para realizar consultas e informes.
  
3. El repositorio de datos de las bases de datos operacionales y del almacén de datos:
  - A. Deben ser físicamente el mismo para aumentar el rendimiento de las consultas.
  - B. Se suelen separar físicamente para aumentar el rendimiento de las consultas.
  - C. Tienen el mismo contenido, aunque puede variar su estructura.
  - D. Ninguna de las respuestas anteriores es correcta.

4. El diseño del almacén de datos está orientado a:
- A. Permitir la recuperación de transacciones en línea fallidas.
  - B. Gestionar la concurrencia en las transacciones en línea.
  - C. Temas como el cliente, suministrador, producto o venta.
  - D. Facilitar las consultas sencillas de datos actuales.
5. El proceso de extracción, transformación y carga (ETL) es necesario pues:
- A. El almacén de datos puede utilizar distintas fuentes heterogéneas de datos.
  - B. Los datos almacenados en las bases de datos operacionales son no estructurados.
  - C. Permiten la visualización de los resultados al usuario del almacén de datos.
  - D. Ninguna de las respuestas anteriores es correcta.
6. El almacén de datos contiene normalmente datos:
- A. Que proporcionan una amplia visión histórica para permitir consultas complejas.
  - B. Actuales con los que consultar el estado inmediato del sistema.
  - C. De las próximas transacciones a realizar.
  - D. Del futuro comportamiento de las entidades de interés.
7. Las operaciones típicas que realizar en el almacén de datos son:
- A. Procesado de transacciones en línea.
  - B. Tratamiento de transacciones en línea concurrentes.
  - C. Carga inicial de datos y lectura de datos.
  - D. Recuperación de transacciones en línea fallidas.

8. El modelo cubo de datos es ampliamente usado en:
- A. Sistemas OLTP.
  - B. Sistemas OLAP.
  - C. Sistemas OLTP y OALP.
  - D. Ninguna de las respuestas anteriores es correcta.
9. Una de las diferencias entre el modelo estrella y el modelo copo de nieve es:
- A. Una menor repetición de datos en las tablas del modelo estrella.
  - B. Una menor repetición de datos en las tablas del modelo copo de nieve.
  - C. El modelo entidad-relación correspondiente.
  - D. Ninguna de las respuestas anteriores es correcta.
10. Si se quiere agregar los elementos de una dimensión de un cubo OLAP hay que realizar una operación:
- A. *Slice*.
  - B. *Roll-up*.
  - C. *Drill-down*.
  - D. Pivotación.