



**Módulo 2: Gestión y procesamiento de datos**

**IFCD66 – Data Science**

**Barcelona, a 14 de noviembre 2022**





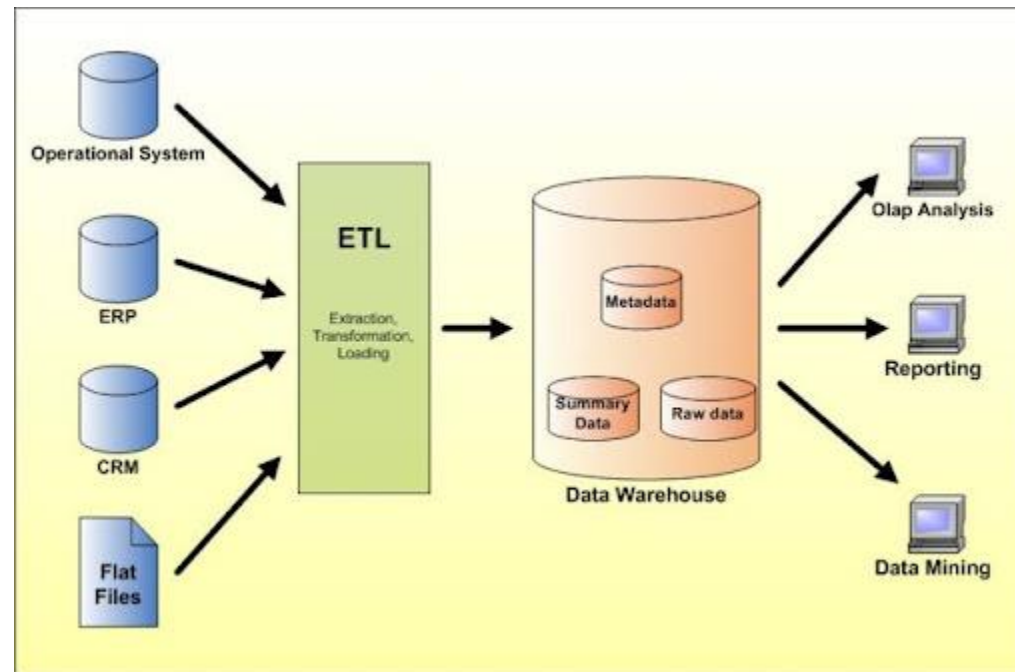
# **IFCD66 – Data Science ETL**

## ETL: Principios básicos

### Que es y para que sirve

Como ya comentamos cuando hablábamos de DataFrames y sus funciones para leer datos externos: "Si tenemos que leer datos de varias fuentes heterogéneas... **no lo hagamos con pandas...** para eso están las ETL, de las que hablaremos más adelante.... "

Pues ese más adelante ya está aquí. Veamos una visión general de donde se ubica un proceso de ETL



## ETL: Principios básicos

### Que es y para que sirve

Abreviatura del término **Extract-Transform-Load**, ETL es un sistema que tiene la capacidad de leer diferentes formatos de archivo y tipos de datos, y transportarlos de un entorno a otro.

El ETL sirve para consolidar datos de diferentes fuentes y entregarlos a un nuevo entorno. Además, se puede utilizar cuando una empresa cambia sistemas antiguos por nuevos. Un sistema ETL también ayuda a recopilar todos los datos de una empresa, contribuyendo a la toma de decisiones.

Básicamente, consiste en tres procesos:

- **Extracción:** El primer proceso ETL es la extracción de datos de sus sistemas originales (fuentes de datos o sistemas operativos). En esta etapa, estos datos se extraen y se transfieren de entornos, ya sea a otras BB.DD. o a un entorno DataWarehouse, donde podemos trabajar estos datos desde una perspectiva más analítica. Más adelante usaremos varios "Jobs" de Pentaho-PDI para generar nuestro DataWarehouse puede ser trabajado por el sistema ETL de forma independiente.
- **Transformación:** Este segundo proceso, los datos se transforman, se consolidan, se enriquecen o bien se "limpian" (**data cleaning**) y ajustan según nuestras necesidades. Sobre estos datos pueden ser atribuidas diferentes reglas que ayudarán a mejorar su calidad (**data quality**). Uno de los propósitos de este paso es la combinación de datos de diferentes fuentes, de modo que puedan ser enviados a un mismo sistema, incluso si provienen de sistemas diferentes.

## ETL: Principios básicos

### Que es y para que sirve

- **Transformación:** por lo que hace a la transformación, las acciones más comunes suelen ser:
  - Eliminar las entradas duplicadas, incompletas, incorrectas o expiradas (deduplicación, normalización)
  - Eliminar los campos innecesarios de los registros
  - Realizar un filtrado y una validación de los datos
  - Reordenar los datos no estructurados en datos estructurados
  - Unir los datos de diferentes fuentes con operaciones JOIN (muy útil para crear nuestras tablas de hechos y dimensiones en un DataWarehouse...)
  - Etc...
- **Carga:** En este último paso, los datos se cargan en uno o varios destinos, una vez pasado por nuestros procesos de transformación. Depende de lo que queramos hacer las cargas pueden ser:
  - Carga completa: Todos los datos se mueven al almacén al mismo tiempo.
  - Incremental: El movimiento de datos se produce en lotes, con una velocidad de refresco.

## ETL: Principios básicos

### Herramientas disponibles

Como en todo, tenemos multitud de herramientas disponibles para construir nuestros procesos de ETL. Un artículo interesante es el del siguiente link (<https://aprenderbigdata.com/herramientas-etl/>), donde se hace un análisis de varias herramientas, como son:

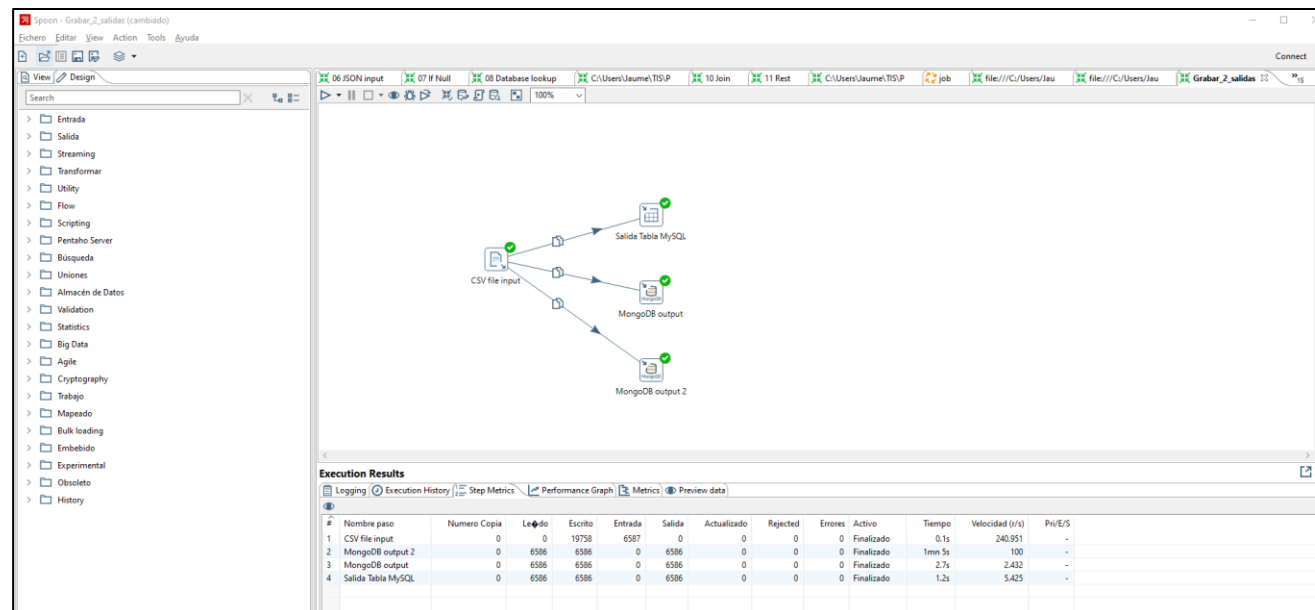
- Apache NiFi
- Streamsets
- Apache Airflow
- AWS Data Pipeline
- AWS Glue
- Talend
- Informatica PowerCenter
- Ab Initio
- Pentaho
- Azure Data Factory
- ....

## ETL: Pentaho

De todas las herramientas antes comentadas, para la realización del curso vamos a usar Pentaho Data Integration (PDI) de Hitachi.

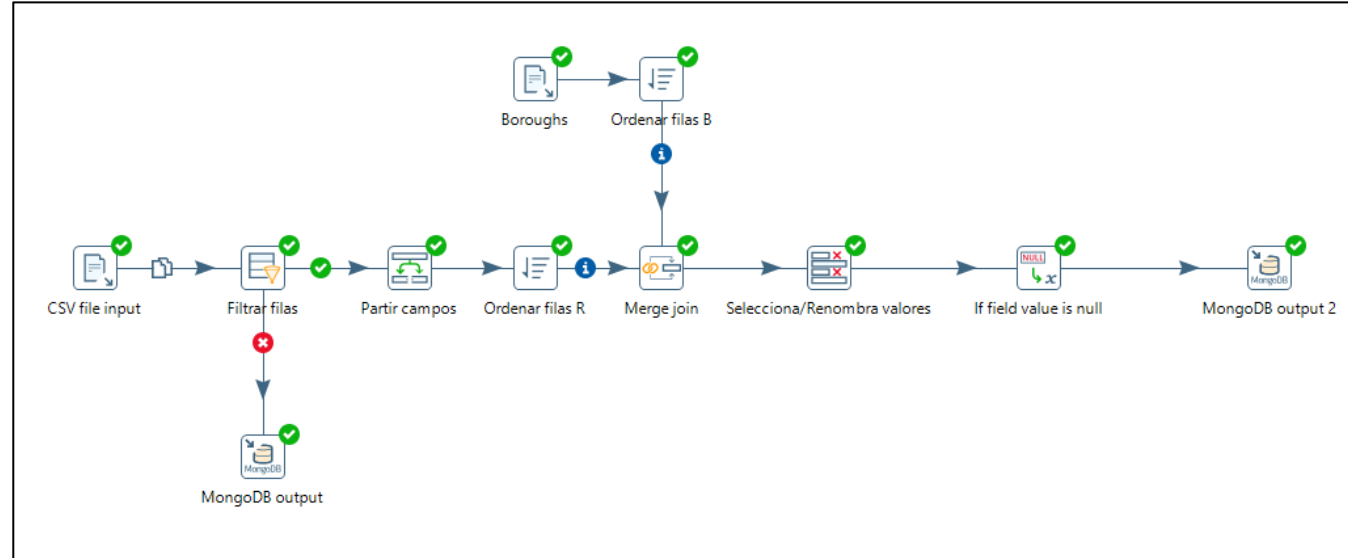
Pentaho se define a sí mismo como una plataforma de BI “orientada a la solución” y “centrada en procesos” que incluye todos los principales componentes requeridos para implementar soluciones basadas en procesos tal como ha sido concebido desde el principio.

Las soluciones que Pentaho pretende ofrecer se componen fundamentalmente de una infraestructura de herramientas de análisis e informes integrado con un motor de workflow de procesos de negocio. La plataforma será capaz de ejecutar las reglas de negocio necesarias, expresadas en forma de procesos, actividades, además es capaz de presentar y entregar la información adecuada en el momento adecuado.

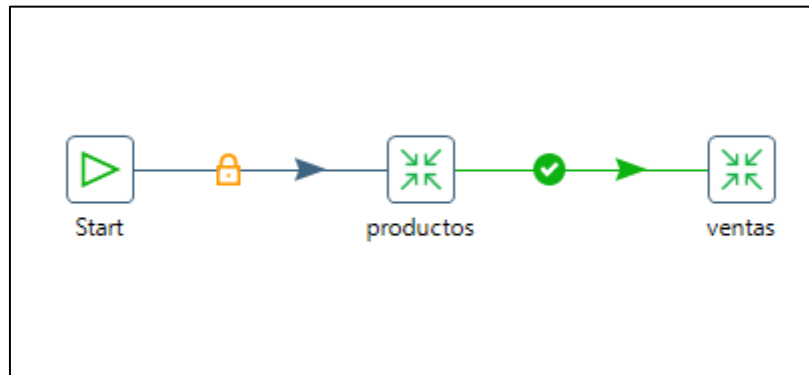


## ETL: Pentaho –principales componentes

Transformaciones: contienen las instrucciones de transformación requeridas.



Trabajos: contienen transformaciones que podemos ejecutar en el orden que queramos.





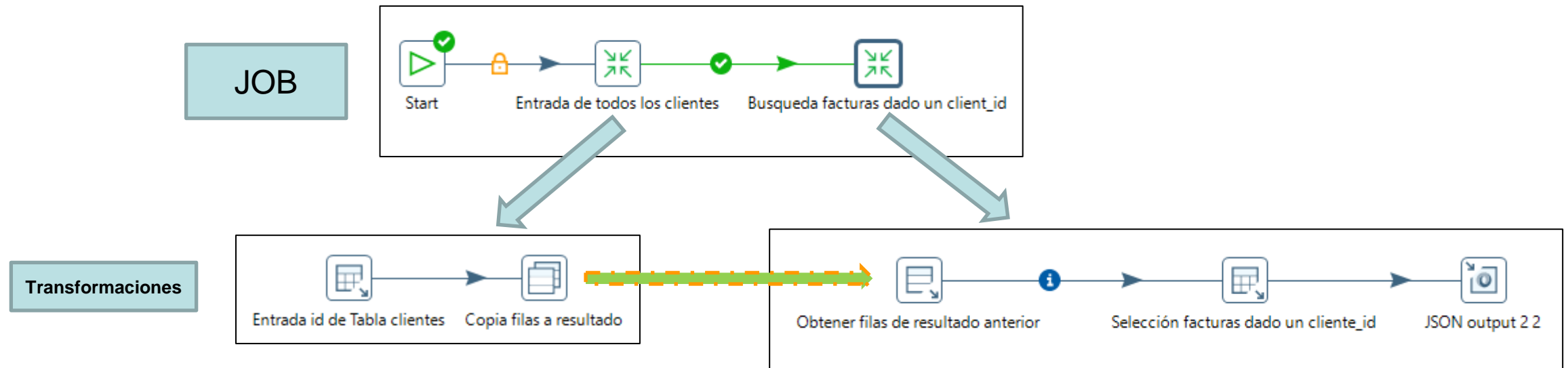
## Trabajos (JOB): Pasos más utilizados

Un trabajo siempre tiene que empezar por un STAR, seguido de las transformaciones que queramos ejecutar secuencialmente.

Una funcionalidad interesante al hacer un JOB, y a tener en cuenta en la transformación, son los pasos siguientes:

- Copia filas a resultado
- Obtener filas de resultado anterior

Veamos un ejemplo de uso:



## ETL: Ejercicio

ETL-00: Teniendo en cuenta los datos necesarios para los ejercicios de MONGO-0X:

- Define las necesidades de datos para poder realizar los ejercicios desde el MONGO-02 al MONGO-08
- Define que data-sets necesitarás y su origen
- Define que BB.DD. necesitarás
- Describe qué flujo de información sería necesario
- Define las reglas de calidad (data quality) que implementarás (si son necesarias)
- Define las reglas de limpieza (data cleaning) que implementarás (si son necesarias)
- Impleméntalo con una ETL en PDI

## ETL: Ejercicios

### ETL-01: Table input

Definición BBDD, Table input, ejecutar, step metrics, preview data.



Table input

### ETL-02:Exportación

Exportación de datos a un fichero.



Table input



Text file output

### ETL-03: Calculadora

Añadir constantes, uso de la calculadora y salida a fichero.



Table input



Add constants



Calculator



Text file output

### ETL-04: Orden

Ordenar resultados



Table input



Add constants



Calculator



Sort rows



Text file output

## ETL: Ejercicios

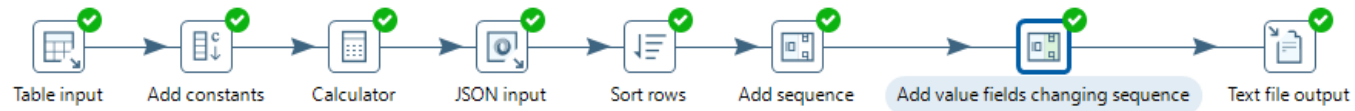
### ETL-05: Secuencia

Añadir secuencia y usar una subsecuencia al "stream" de datos



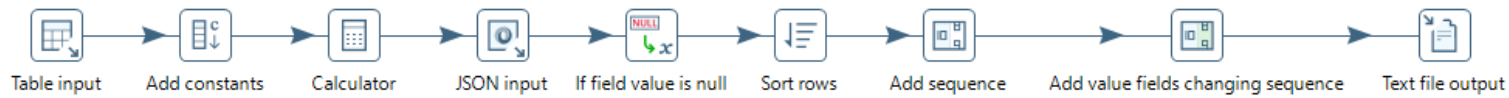
### ETL-06: JSON input

Sacar datos de un campo en JSON y añadirlos al "stream" de datos



### ETL-07: If Null

Rellenar con valores si algunos campo son NULL



### ETL-08: Database lookup

Usar el paso "Database lookup" para buscar literales de claves.



## ETL: Ejercicios

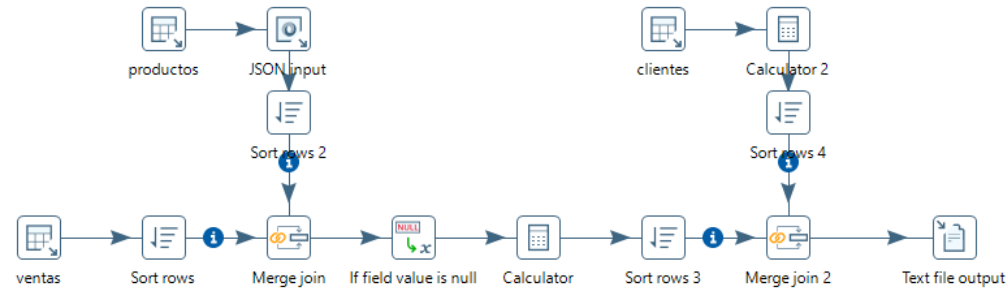
### ETL-09: Select

Seleccionar valores del "stream" de datos



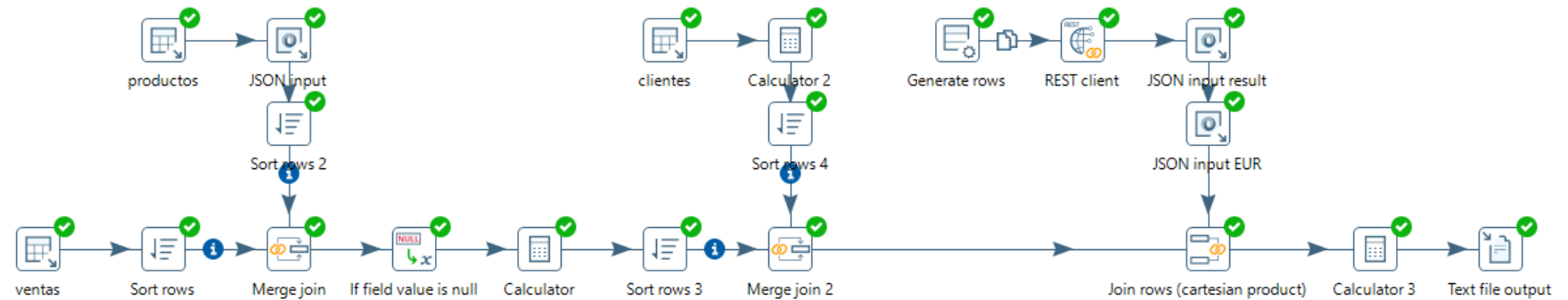
### ETL-10: Join

Ejemplo de Left Join



### ETL-11: Rest

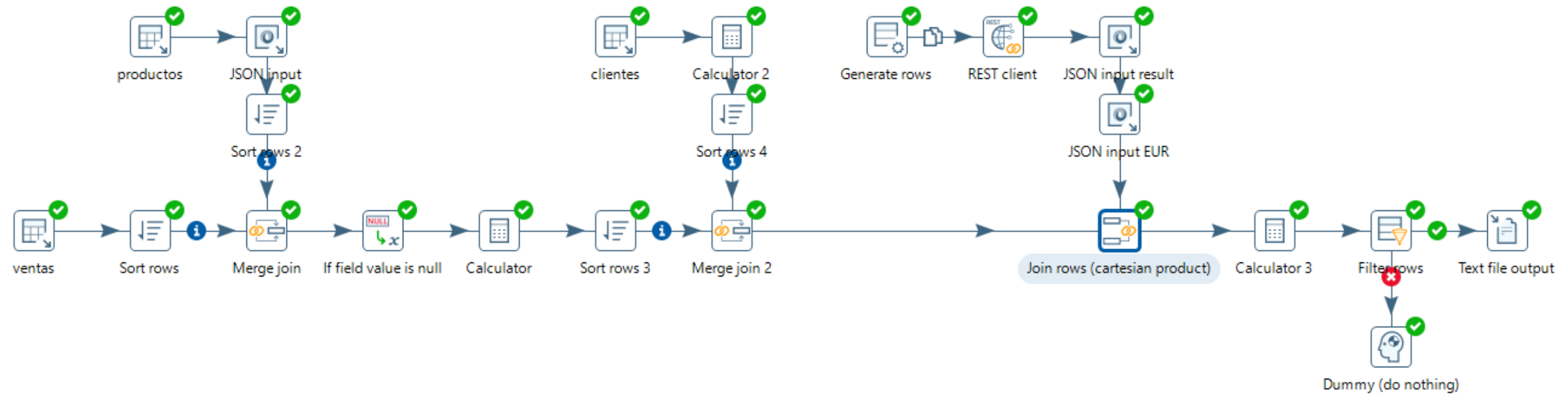
Ejemplo de llamada a una API más un producto cartesiano



## ETL: Ejercicios

ETL-12: Filter

Filtro del "stream" de datos





# **IFCD66 – Data Science DWH**

## DWH

Bill Inmon, considerado el padre del DWH, proporcionó la siguiente definición del concepto: "Un DWH es una colección de datos orientada al tema, integrada, no volátil y variante en el tiempo para dar soporte a decisiones de dirección."

Tiene 4 características básicas:

- **Orientado al tema:** Un DW guarda los datos en un modelo que sigue conceptos de negocio a diferencia de los procesos de negocio.
  - Por ejemplo, para una compañía de seguros, la parte transaccional contendrá información sobre automóviles, vida, salud y accidentes y en cambio, las principales áreas temáticas del DWH pueden ser los clientes, las pólizas, las primas y las reclamaciones.
- **Integrado:** Los datos provienen de una cantidad de fuentes de datos diversa y el modelo lógico del DW tiene que ser integrado y consistente.
- **No volátil:** Un DWH no cambia con las actualizaciones transaccionales. Esto es especialmente importante en el DWH casi en tiempo real. Una vez que un registro se escribe en un DWH, está completo. Es decir, no se elimina a menos que sea un error de carga real. Es responsabilidad del arquitecto de la ETL asegurarse de que las reglas de escritura de datos en el DWH siguen la definición de negocio correcta.
- **Variable en el tiempo:** Significa que el proceso de integración de los datos reconoce cuando dado un registro en un sistema transaccional exhibe cambios y crea una segunda versión del registro (o tercera, y así sucesivamente).



## DWH

Hay dos métodos de modelado multidimensional: Inmon y Kimball. Es este curso usaremos el método Kimball, ya que es más sencillo y rápido de implementar que el método Inmon.

La Metodología Kimball: Iniciado por Ralph Kimball, el modelo de datos de Kimball sigue un enfoque de abajo hacia arriba para el almacenamiento de datos y diseño de arquitectura, formándose primero en función de los requisitos comerciales.

Sigue, generalmente, los siguientes pasos:

- Se evalúan las fuentes de datos primarias y se usa una herramienta de extracción, transformación y carga (ETL) para obtener datos de varias fuentes y cargarlos en un área de preparación del servidor de base de datos relacional.
- Una vez que los datos se cargan en el área de ensayo del almacén de datos, la siguiente fase incluye la carga de datos en un modelo de almacén de datos dimensional que no está normalizado por naturaleza.

Este modelo divide los datos en:

- la **tabla de hechos**, que son datos transaccionales numéricos y
- la **tabla de dimensiones**, que es la información de referencia que respalda los hechos, las dimensiones sobre las cuales queremos analizar.

También hay que tener en cuenta conceptos como la **cardinalidad**, la **granularidad** y la **agregación** de los cuales hablaremos más adelante.

# Modelización Multidimensional

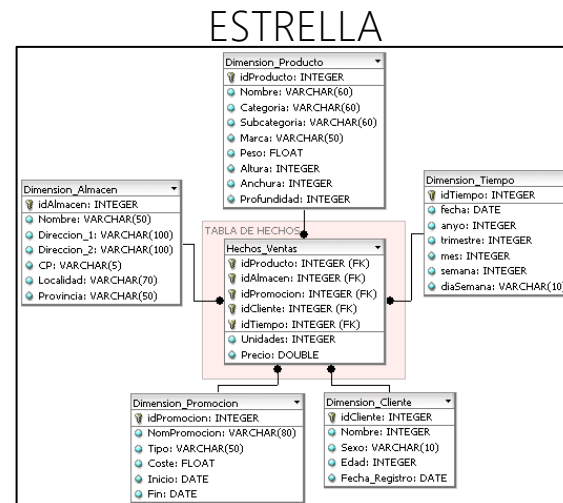
## DWH

Existen dos esquemas para montar un DWH:

- El esquema en **estrella**: es un modelo de datos que tiene una tabla de hechos que contiene los datos para el análisis, rodeada de las tablas de dimensiones. Este aspecto, de tabla de hechos (o central) más grande rodeada de radios o tablas más pequeñas es lo que asemeja a una estrella, dándole nombre a este tipo de construcciones.

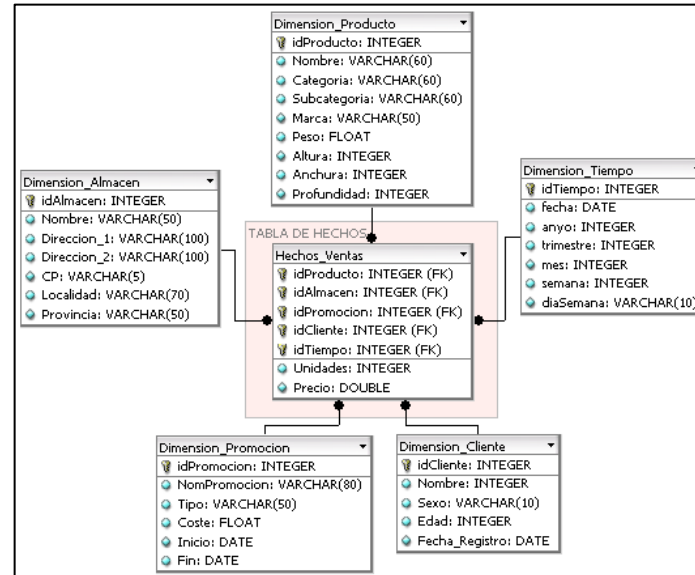
Las tablas de **dimensiones** tendrán siempre una **clave primaria simple**, mientras que en la tabla de hechos, la **clave principal** estará compuesta por las claves principales de las tablas dimensionales.

- El esquema en **copo de nieve**: como ya hemos dicho, es una estructura algo más compleja que el esquema en estrella. Se da cuando **alguna de las dimensiones se implementa con más de una tabla de datos**. La finalidad es normalizar las tablas y así reducir el espacio de almacenamiento al eliminar la redundancia de datos; pero tiene la contrapartida de generar peores rendimientos al tener que crear más tablas de dimensiones y más relaciones entre las tablas (JOINS) lo que tiene un impacto directo sobre el rendimiento.



## DWH

**Tabla de hechos:** como ya hemos dicho, una tabla de hechos es la tabla central de un esquema dimensional (en estrella o en copo de nieve) y **contiene los valores de las medidas de negocio** o dicho de otra forma **los indicadores de negocio**. Cada medida se toma mediante la intersección de las dimensiones que la definen, dichas dimensiones estarán reflejadas en sus correspondientes tablas de dimensiones que rodearán la tabla de hechos y estarán relacionadas con ella.



En el ejemplo, la tabla central (Ventas) es la tabla de hechos de un diseño de modelo de datos en estrella, las cinco tablas que la rodean (Producto, Tiempo, Almacén, Promoción y Cliente) son las cinco dimensiones de que consta esta tabla de hechos, en dicha tabla se almacenan, en este caso, las unidades vendidas y el precio obtenido por dichas ventas, estos son los hechos o medidas de negocio almacenados y que, gracias al diseño multidimensional en estrella, podrán ser analizados de forma exhaustiva, típicamente mediante técnicas OLAP (procesamiento analítico on-line).

## DWH

Las medidas más útiles para incluir en una tabla de hechos son aquellas **medidas que pueden ser sumadas** como por ejemplo la cantidad de producto vendido, los costes de producción o el dinero obtenido por las ventas; son medidas numéricas que pueden calcularse con la suma de varias cantidades de la tabla. En consecuencia, **por lo general** los hechos a almacenar en una tabla de hechos van a ser **casi siempre valores numéricos, enteros o reales**.

**Cardinalidad** : Las tablas de hechos pueden contener un gran número de filas, a veces cientos de millones de registros cuando contienen uno o más años de la historia de una gran organización, esta cardinalidad estará acotada superiormente por la cardinalidad de las tablas dimensionales.

Por ejemplo, si se tiene una tabla de hechos "TH" de tres dimensiones D1, D2 y D3, el número máximo de elementos que tendrá la tabla de hechos TH será:

$\text{Card(TH)} = \text{Card(D1)} \times \text{Card(D2)} \times \text{Card(D3)}$  , donde 'Card(x)' es la cardinalidad de la tabla 'x'

Naturalmente, estas cardinalidades no son fijas, ya que, por ejemplo, si una de las dimensiones se refiere a los clientes de la empresa, cada vez que se dé de alta un nuevo cliente se estará aumentando la cardinalidad de la tabla de hechos. Una de las dimensiones suele ser el tiempo, éste puede medirse de muy distintas formas (por horas, días, semanas, ...), pero lo cierto es que transcurre continuamente, y para que el sistema funcione se deben añadir registros periódicamente a la tabla de esta dimensión (tabla de tiempos) y esto también produce un aumento de la cardinalidad de la tabla de hechos, esta es la principal causa de que las tablas de hechos lleguen a tener una cantidad de registros del orden de millones de elementos.

## DWH

**Granularidad** : Una característica importante que define a una tabla de hechos es el nivel de granularidad de los datos que en ella se almacenan, entendiéndose por 'granularidad' **el nivel de detalle de dichos datos**, es decir, la granularidad de la tabla de hechos representa el nivel más atómico por el cual se definen los datos.

Por ejemplo, no es lo mismo contar el tiempo por horas (grano fino) que por semanas (grano grueso); o en el caso de los productos, se puede considerar cada variante de un mismo artículo como un producto (por ejemplo, en una empresa textil, cada talla y color de pantalón podría ser un producto) o agrupar todos los artículos de una misma familia considerándolos como un único producto (por ejemplo, el producto pantalón genérico).

Como se puede observar, la granularidad afecta a la cardinalidad, tanto de las dimensiones como de la tabla de hechos, a mayor granularidad (grano más fino) mayor será el número de registros final de la tabla de hechos.

Cuando la granularidad es mayor, es frecuente que se desee disponer de subtotales parciales, es decir, si tenemos una tabla de hechos con las ventas por días, podría interesar disponer de los totales semanales o mensuales, estos datos se pueden calcular haciendo sumas parciales, pero es frecuente añadir a la tabla de hechos registros donde se almacenan dichos cálculos para no tener que repetirlos cada vez que se requieran y mejorar así el rendimiento de la aplicación. En este caso se dispondrá en la misma tabla de hechos de datos de grano fino y de grano más grueso aumentando aún más la cardinalidad de la tabla.

## DWH

Agregación : es un proceso de cálculo por el cual se resumen los datos de los registros de detalle. Esta operación consiste normalmente en el cálculo de totales dando lugar a medidas de grano grueso. Cuando se resumen los datos, el detalle ya no está directamente disponible para el analista, ya que este se elimina de la tabla de hechos.

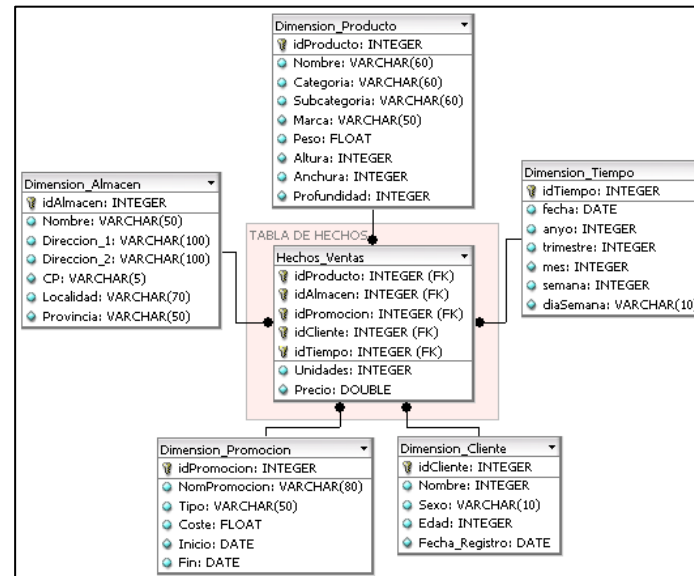
Esta operación se realiza típicamente con los datos más antiguos de la empresa con la finalidad de seguir disponiendo de dicha información (aunque sea resumida) para poder eliminar registros obsoletos de la tabla de hechos para liberar espacio.

Vigilemos con los tipos de los datos: Como ya se ha comentado, es normal que las tablas de hechos almacenen muchos millones de registros, por esta razón es muy importante que no se despilfarre memoria, hay que procurar utilizar los tipos de datos adecuados, si una medida a almacenar puede guardarse en un campo de tipo entero, no debemos definir ese campo como de tipo entero largo o como tipo real.

Del mismo modo, si una magnitud necesita decimales, si las características de ésta lo permiten, será mejor utilizar un tipo real simple que un tipo real de doble precisión. Nótese que elegir uno u otro de estos campos, en principio sólo supondría una diferencia de unos pocos bytes en un registro, pero dado que en una tabla de hechos estamos hablando de cientos de millones de registros, en realidad, esa diferencia no es despreciable (**5 bytes x 200 millones de registros = 1GB de memoria**).

## DWH

Tablas de dimensiones: acompañan a la tabla de hechos y determinan los parámetros (**dimensiones**) de los que dependen los hechos registrados en la tabla de hechos. Contienen atributos (o campos) que se utilizan para restringir y agrupar los datos almacenados en una tabla de hechos cuando se realizan consultas sobre dicho datos. Estos datos sobre dimensiones son parámetros de los que dependen otros datos que **serán objeto de estudio y análisis** y que están contenidos en la tabla de hechos. Las tablas de dimensiones ayudan a realizar ese estudio/análisis **aportando información** sobre los datos de la tabla de hechos



Tablas de dimensiones típicas que tendremos en cualquier DWH pueden ser la de tiempo y geografía.

## DWH

**Granularidad** : cada dimensión puede referirse a conceptos como 'tiempo', 'productos', 'clientes', 'zona geográfica', etc. Ahora bien, cada dimensión puede estar medida de diferentes maneras según la granularidad deseada, por ejemplo, para la dimensión "zona geográfica" podríamos considerar 'localidades', 'provincias', 'regiones', 'países' o 'continentes'.



La unidad de medida (por localidades, provincias, etc.) determinará esa granularidad, cuanto más pequeña sea esta unidad de medida más fina será esta granularidad (grano fino); si las unidades de medida son mayores, entonces hablaremos de granularidad gruesa (grano grueso).

En muchas ocasiones interesa disponer de los datos a varios niveles de granularidad, es decir, es importante para el negocio poder consultar los datos (siguiendo el ejemplo de las zonas) por localidades, provincias, etc., en estos casos se crea una jerarquía con la dimensión, ya que tenemos varios niveles de asociación de los datos (con otras dimensiones como el tiempo, se podrían crear niveles jerárquicos del tipo 'días', 'semanas', 'meses'...).

En todo caso, siempre es una buena práctica atomizar en el DWH la granularidad, que luego podemos ajustar con herramientas tipo Power BI. Un link interesante de leer : <https://learn.microsoft.com/es-es/power-bi/guidance/star-schema>



## DWH

EJERCICIO: partiremos de nuestra BB.DD. "Awlite" para crear un DWH estrella de la misma:

Tablas de dimensiones:

- dim\_clientes
- dim\_productos
- dim\_tiendas
- dim\_geografía (\*)

Tabla de hechos:

- fact\_ventas

## DWH

EJERCICIO: partiremos de nuestra BB.DD. "Awlite" para crear un DWH estrella de la misma:

- dim\_clientes
  - c.cliente\_id,
  - c.cuenta\_num
  - c.nombre
  - c.apellidos
  - c.email
  - c.telefono
  - c.direccion1, c.direccion2, c.poblacion, c.cp
  - c.fecha\_nacimiento
  - c.fecha\_alta
  - c.sexo, c.num\_hijos, c.casa\_propiedad, c.ingresos\_anuales, c.coches\_propiedad, c.ocupacion, c.educación
  - c.zona\_id, c.provincia\_id

## DWH

EJERCICIO: partiremos de nuestra BB.DD. "Awlite" para crear un DWH estrella de la misma:

- dim\_productos
  - p.producto\_id, p.producto, p.SKU
  - p.elaborado, p.acabado, p.stock\_seguridad, p.nivel\_reorden
  - p.coste, p.precio
  - um.unidad\_medida as peso\_um, p.peso
  - p.dias\_produccion
  - p.linea\_producto
  - pc.categoria
  - ps.subcategoria
  - m.modelo
  - p.fecha\_alta, p.fecha\_baja
  - p.caracteristicas
- left join unidades\_medida um on p.unidadpeso\_id = um.unidad\_medida\_cod
- left join productos\_subcategoria ps on p.subcategoria\_id = ps.subcategoria\_id
- left join productos\_categoria pc on ps.categoria\_id = pc.categoria\_id
- left join productos\_modelos m on p.modelo\_id = m.modelo\_id

## DWH

EJERCICIO: partiremos de nuestra BB.DD. "Awlite" para crear un DWH estrella de la misma:

- dim\_tiendas
  - t.tienda\_id, t.tienda
  - , t.anyo\_apertura
  - , t.especialidad, t.empleados
  - , t.metros2
  - , t.direccion
  - , t.poblacion
  - , t.cp
  - , t.zona\_id
  - , t.provincia\_id
  - , e.nombre , e.apellidos
  - , e.superior\_id
- left join empleados e on t.director\_id = e.empleado\_id

## DWH

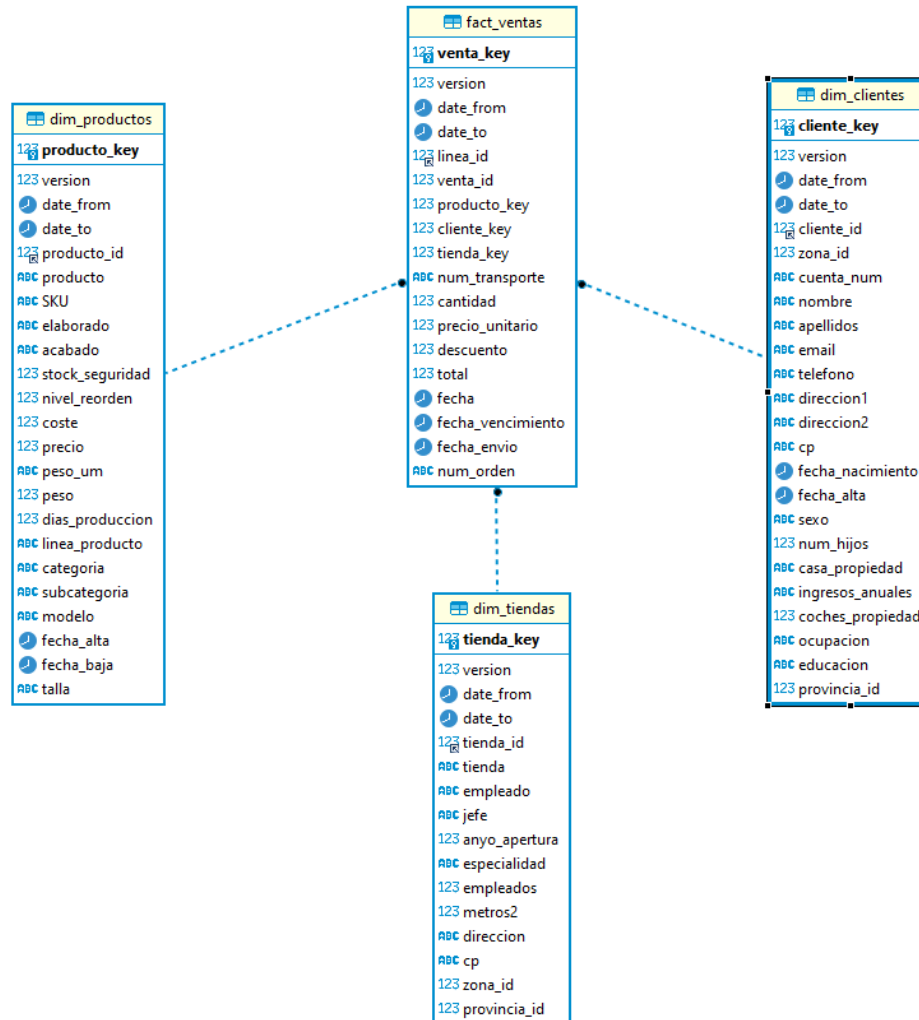
EJERCICIO: partiremos de nuestra BB.DD. "Awlite" para crear un DWH estrella de la misma:

- fact\_ventas
  - linea\_id
  - venta\_id
  - producto\_key
  - cliente\_key
  - tienda\_key
  - num\_transporte
  - cantidad
  - precio\_unitario
  - descuento
  - total
  - fecha
  - fecha\_vencimiento
  - fecha\_envio
  - num\_orden

# Modelización Multidimensional

## DWH

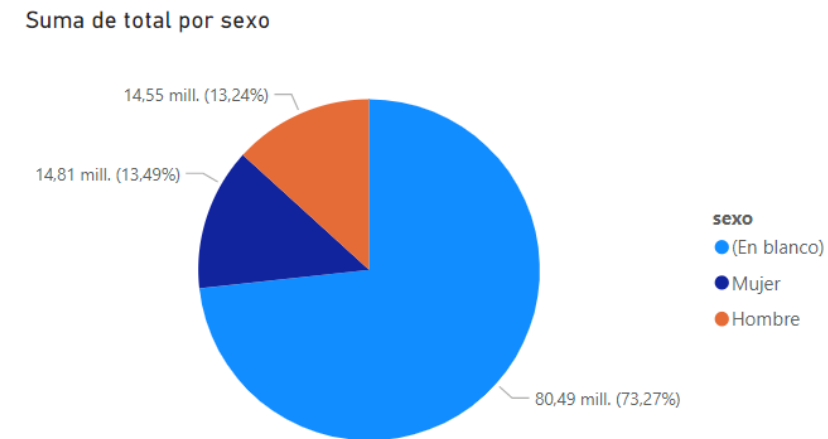
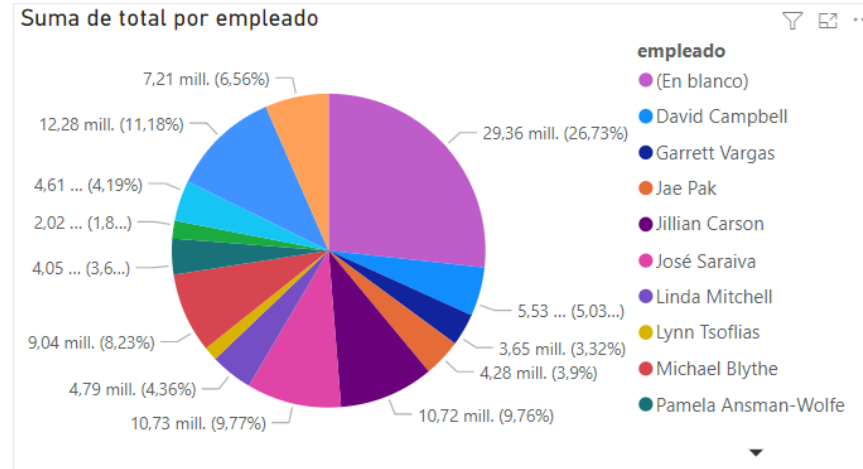
EJERCICIO: partiremos de nuestra BB.DD. "Awlite" para crear un DWH estrella de la misma:



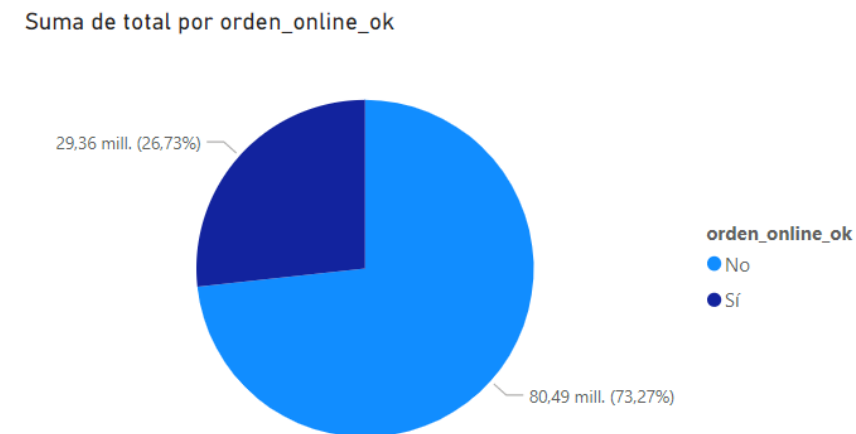
# Modelización Multidimensional

## DWH

EJERCICIO: partiremos de nuestro DWH estrella y empezaremos a explotar los datos con PowerBI. Primero generaremos una página con datos genéricos.



Suma de total	Año	orden_online_ok
8.065.435,94	2001	No
3.266.373,86	2001	Sí
24.144.433,08	2002	No
6.530.343,49	2002	Sí
32.219.974,87	2003	No
9.791.060,80	2003	Sí
16.057.859,12	2004	No
9.770.899,74	2004	Sí
109.846.380,90		

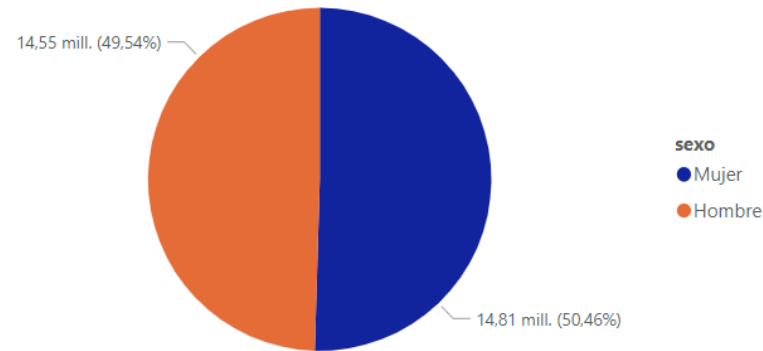


# Modelización Multidimensional

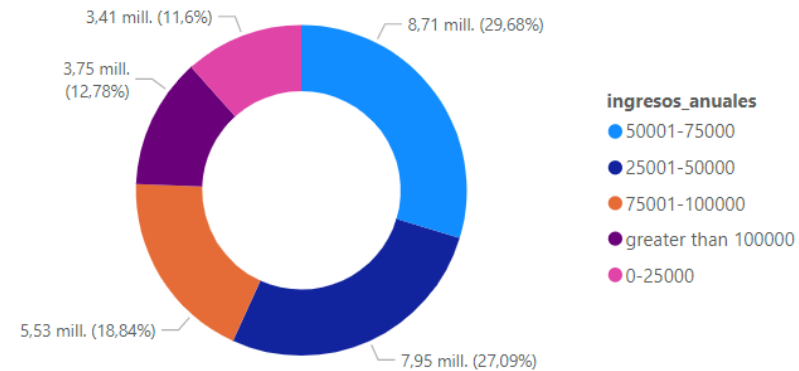
## DWH

EJERCICIO: Luego generaremos una página sólo con los datos de ventas on-line, poniendo un filtro de página y haciendo una jerarquía de categoría y subcategoría... y modelo.

Suma de total por sexo



Suma de total por ingresos\_anuales

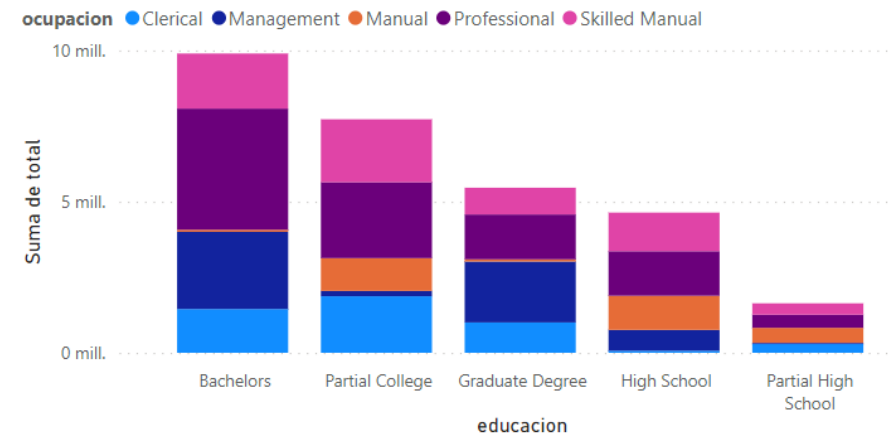


Suma de total por categoría y subcategoría



Suma de total	Año	orden_online_ok
3.266.373,86	2001	Sí
6.530.343,49	2002	Sí
9.791.060,80	2003	Sí
9.770.899,74	2004	Sí
29.358.677,89		

Suma de total por educación y ocupación



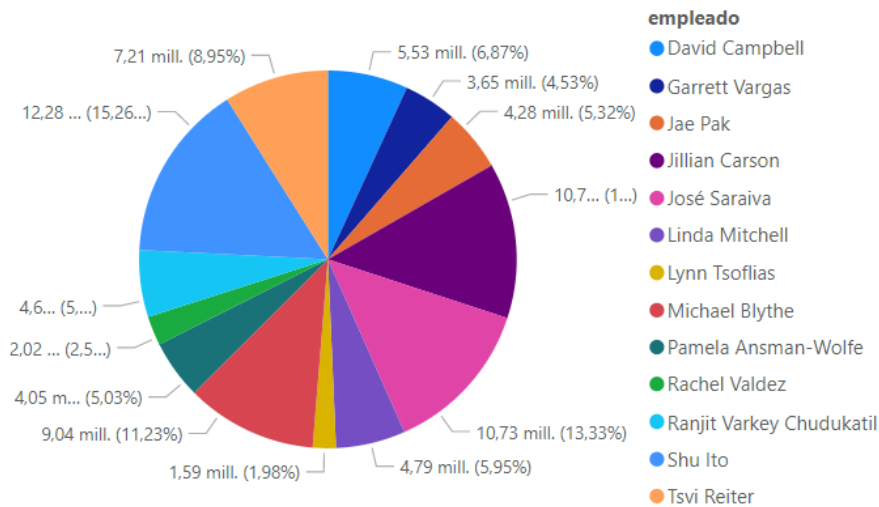


# Modelización Multidimensional

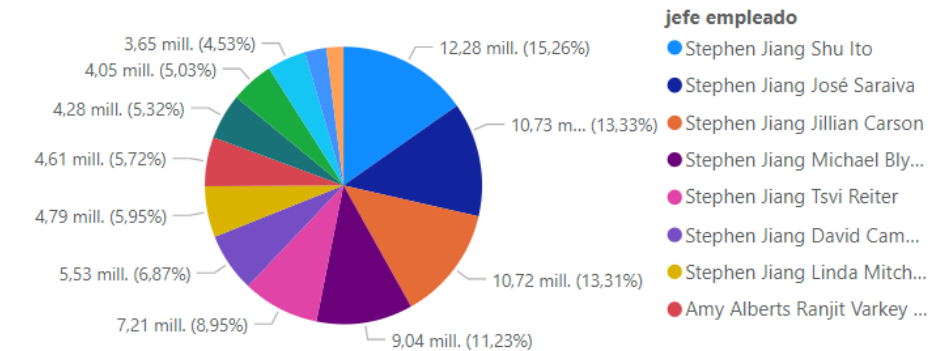
## DWH

EJERCICIO: Luego generaremos otra página sólo con los datos de ventas en tiendas...

Suma de total por empleado



Suma de total por jefe y empleado



Suma de empleados y Suma de total por empleado y jefe



Suma de total	Año	orden_online_ok
8.065.435,94	2001	No
24.144.433,08	2002	No
32.219.974,87	2003	No
16.057.859,12	2004	No
80.487.703,01		

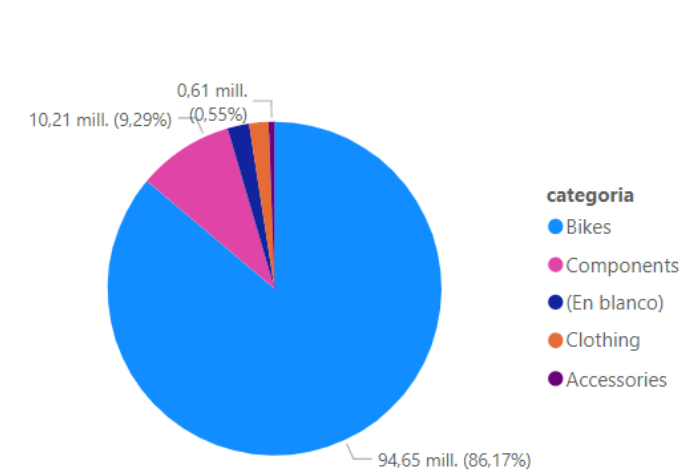
jefe	empleado
Amy Alberts	Jae Pak
Amy Alberts	Rachel Valdez
Amy Alberts	Ranjit Varkey Chudukatil
Stephen Jiang	David Campbell
Stephen Jiang	Garrett Vargas
Stephen Jiang	Jillian Carson
Stephen Jiang	José Saraiva
Stephen Jiang	Linda Mitchell
Stephen Jiang	Michael Blythe
Stephen Jiang	Pamela Ansman-Wolfe
Stephen Jiang	Shu Ito
Stephen Jiang	Tsvi Reiter
Syed Abbas	Lynn Tsoflias

# Modelización Multidimensional

## DWH

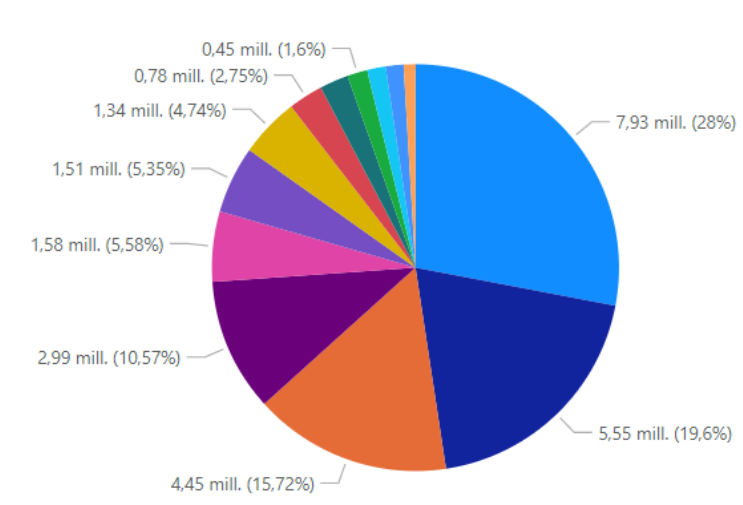
EJERCICIO: Y por último, generaremos otra página sólo con datos de productos, haciendo jerarquías y explorando las posibilidades de "zoom"...

Suma de total por categoría



Suma de total	Año	orden_online_ok
8.065.435,94	2001	No
3.266.373,86	2001	Sí
24.144.433,08	2002	No
6.530.343,49	2002	Sí
32.219.974,87	2003	No
9.791.060,80	2003	Sí
16.057.859,12	2004	No
9.770.899,74	2004	Sí
109.846.380,90		

Suma de total por categoría, subcategoría y modelo



- categoria subcategoria modelo
- Bikes Mountain Bikes Mountain-200
- Bikes Road Bikes Road-150
- Bikes Road Bikes Road-250
- Bikes Touring Bikes Touring-1000
- Bikes Road Bikes Road-350-W
- Bikes Road Bikes Road-550-W
- Bikes Mountain Bikes Mountain-100
- Bikes Road Bikes Road-750
- Bikes Road Bikes Road-650
- Bikes Touring Bikes Touring-2000
- Bikes Mountain Bikes Mountain-4...
- Bikes Touring Bikes Touring-3000
- Bikes Mountain Bikes Mountain-500

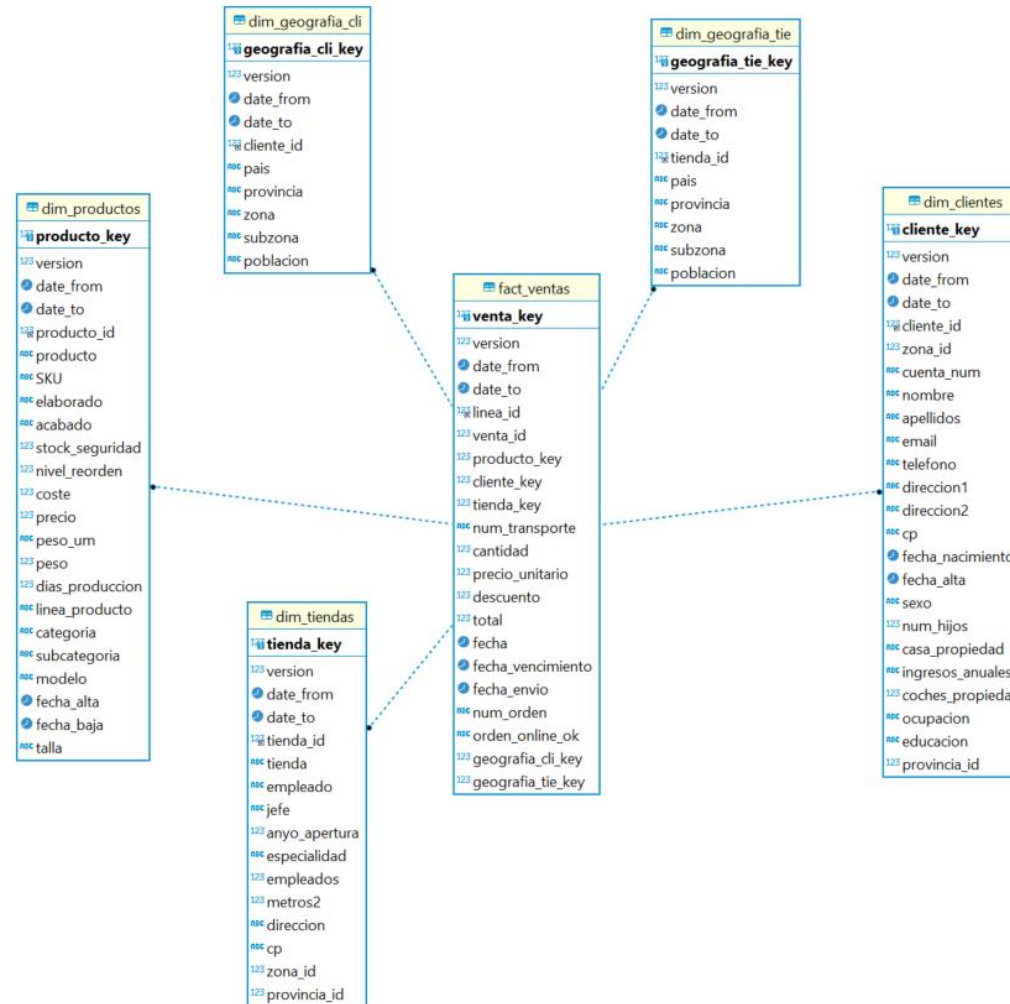
categoria	subcategoria	modelo	Suma de total
			2.309.029,63
Accessories	Bike Racks	Hitch Rack - 4-Bike	237.096,21
Accessories	Bike Stands	All-Purpose Bike Stand	39.591,00
Accessories	Bottles and Cages	Mountain Bottle Cage	20.229,75
Accessories	Bottles and Cages	Road Bottle Cage	15.390,88
Accessories	Bottles and Cages	Water Bottle	28.654,32
Accessories	Cleaners	Bike Wash	18.407,03
Accessories	Fenders	Fender Set - Mountain	46.619,58
Accessories	Hydration Packs	Hydration Pack	105.826,62
Accessories	Locks	Cable Lock	16.240,22
Accessories	Pumps	Minipump	13.514,71
Accessories	Tires and Tubes	Mountain Tire Tube	15.444,05
Accessories	Tires and Tubes	Patch kit	8.232,49
Total			109.846.380,90



# Modelización Multidimensional

## DWH

EJERCICIO: agregaremos nuestras dimensiones de geografía:



# Modelización Multidimensional

## DWH

EJERCICIO: Y generamos más páginas de nuestro informe de PowerBi: visión por cliente

Suma de total por cliente\_pais



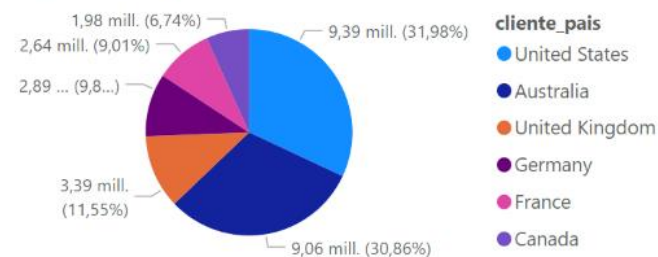
Suma de total por cliente\_pais



Suma de total	Año	orden_online_ok
3.266.373,86	2001	Sí
6.530.343,49	2002	Sí
9.791.060,80	2003	Sí
9.770.899,74	2004	Sí

**29.358.677,89**

Suma de total por cliente\_pais





# Modelización Multidimensional

## DWH

EJERCICIO: Y generamos más páginas de nuestro informe de PowerBi: visión por tienda

Suma de total por tienda\_pais

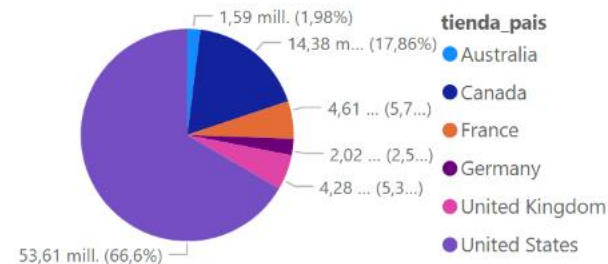


Suma de total por tienda\_pais



Suma de total por tienda\_pais

Suma de total	Año	orden_online_ok
8.065.435,94	2001	No
24.144.433,08	2002	No
32.219.974,87	2003	No
16.057.859,12	2004	No
<b>80.487.703,01</b>		

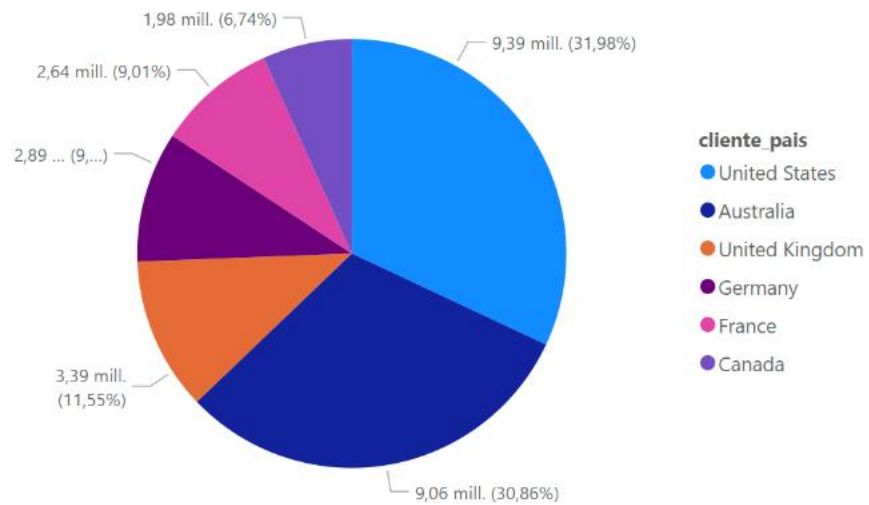


# Modelización Multidimensional

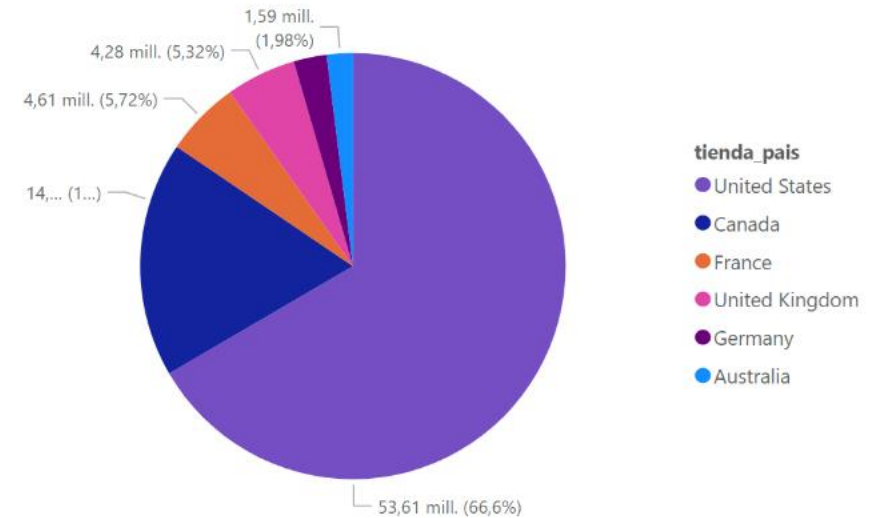
## DWH

EJERCICIO: Y una general por geografía:

Suma de total por cliente\_pais



Suma de total por tienda\_pais



Suma de total	Año	orden_online_ok
8.065.435,94	2001	No
3.266.373,86	2001	Sí
24.144.433,08	2002	No
6.530.343,49	2002	Sí
32.219.974,87	2003	No
9.791.060,80	2003	Sí
16.057.859,12	2004	No
9.770.899,74	2004	Sí
109.846.380,90		



**Carrer Vicenç Llivina, 08940 Cornellà de Llobregat**

**[www.thecorner.cat](http://www.thecorner.cat)**

**[info@thecorner.cat](mailto:info@thecorner.cat)**