3 Diseño de cubos OLAP

3.1 Cubos de Datos

- 3.1.1 OLAP (On-Line Analytical Processing, Procesamiento Analítico en Línea).
- 3.1.2 Comparativa entre OLAP y OLTP (OnLine Transaction Processing, Procesamiento de Transacciones En Línea)

3.2 Modelado de Datos para almacenes de datos

- 3.2.1 Modelo Relacional
- 3.2.2 Modelo dimensional OLAP
- 3.2.3 Tipos de Esquemas (Estrellas, copo de nieve y otros)

3.3 Operadores de cubos de datos

3.4 Cubos multidimensionales

3.1 Cubos de Datos

3.1.1 OLAP (On-Line Analytical Processing, Procesamiento Analítico en Línea).

- Es una solución utilizada en la Inteligencia de Negocios cuyo objetivo es agilizar la consulta de grandes cantidades de datos.
  - Para ello utiliza estructuras de datos diversas, normalmente multidimensionales (o Cubos OLAP), que contienen datos resumidos de Sistemas Transaccionales (OLTP).
  - La principal característica que potencia OLAP, es que es lo más rápido al ejecutar sentencias SQL de tipo SELECT, en contraposición con OLTP que es la mejor opción para operaciones de tipo INSERT, UPDATE Y DELETE

3.1.2 Comparativa entre OLAP y OLTP (OnLine Transaction Processing, Procesamiento de Transacciones En Línea)

Se hace referencia a las aplicaciones de software que están en el corazón de los sistemas operativos como procesamiento de transacciones en línea (on line transactional processing, OLTP).

Por otro lado, todo el conjunto de herramientas destinadas a realizar análisis de inteligencia de negocios y soporte los procesos de toma de decisiones se den por el nombre de procesamiento analítico en línea (on line analitycal processing, OLAP).

Por lo tanto, podemos suponer que la función de un almacén de datos es proporcionar datos de entrada a las aplicaciones OLAP.

Comparativa OLTP vs. OLAP

Característica	OLTP	OLAP
Volatilidad	datos dinámicos	datos estáticos
Puntualidad	sólo datos actuales	datos actuales e históricos
Dimensión de tiempo	implícito y actual	explícito y variante
Granularidad	datos detallados	datos agregados y consolidados
Actualización	continuo e irregular	periódico y regular
Actividades	Repetitivo	imprevisible
Flexibilidad	Bajo	Alto
Rendimiento	alto, pocos segundos por consulta	puede ser bajo para consultas complejas
Usuarios	Empleados	trabajadores del conocimiento
Funciones	Operacional	analítico
Finalidad del uso	Transacciones	consultas complejas y apoyo a la toma de decisiones
Prioridad	alto rendimiento	alta flexibilidad
Métricas	tasa de transacción	respuesta eficaz
Tamaño	megabytes a gigabytes	gigabytes a terabytes

3.2 Modelado de Datos para almacenes de datos

3.2.1 Modelo Relacional

Definición de tablas

Definición_relación_básica ::= **CREATE TABLE** *nom_relación* (elemento_relación1, elemento_relación2, …)

elemento_relación ::= definición_atributo | restricción_relación

definición_atributo::= *nom_atributo* tipo_datos [**DEFAULT** (*expresión*)]

[restricción_atributo1 restricción_atributo2…]

tipo_datos::= **CHAR** (*longitud*) | **VARCHAR2** (*longitud*) | **NUMBER** [(*precisión*[, *escala*])] | **DATE**

restricción_atributo ::= [**CONSTRAINT** nom_restricción] {[**NOT**] **NULL** | **UNIQUE** | **PRIMARY KEY** | **REFERENCES** *nom_relación** [(*nom_atributo**)] [**ON DELETE** {**CASCADE** | **SET NULL**}] | **CHECK** (condición)} [cuándo_comprobar]

restricción_relación::= [**CONSTRAINT** *nom_restricción*] { **UNIQUE** (*nom_atributo1*, *nom_atributo2*, …) | **PRIMARY KEY** (*nom_atributo1*, *nom_atributo2*, …) | **FOREIGN KEY** (*nom_atributo1*, *nom_atributo2*, …) **REFERENCES** *nom_relación* [(*nom_atributo1*, *nom_atributo2*, …)] [**ON DELETE** {**CASCADE** | **SET NULL**} | **CHECK** (*condición*)} [cuándo_comprobar]

cuándo_comprobar::= [**NOT**] **DEFERRABLE** [**INITIALLY** {**IMMEDIATE** | **DEFERRED**}]

La integridad de los datos es la propiedad que asegura que información dada es correcta, al cumplir ciertas aserciones.

Las restricciones de integridad son propiedades de la base de datos que se deben satisfacer en cualquier momento.

Entre los diferentes tipos de restricciones dentro de un SABD tenemos:

- Tratamiento de valores nulos.
- Valores por defecto.
- Integridad de clave primaria.
- Claves alternativas.
- Integridad referencial.
- Restricciones de integridad estáticas (reglas del negocio).

Todas las definiciones de restricciones (tratamiento de nulos, claves primarias, ajenas etc.) ya sean a nivel de columna o a nivel de tabla, tienen un nombre.

Como esta asignación de nombre por parte del diseñador es opcional, si no se le asigna nombre en el momento de la definición, el SABD le asigna uno interno. En cualquier caso, es conveniente declarar una restricción con su nombre porque así es posible referenciarla posteriormente en sentencias ALTER TABLE, para activarla o desactivarla.

Restricciones de Integridad

Restricciones en un SABD

Restricción de Dominio: sólo los objetos especificados pueden servir como dominio de un atributo en el rango de valores definidos por el tipo de dato.

Restricción de Unicidad: un objeto podrá ser unívocamente identificable usando un determinado atributo y no permitirá valores duplicados.

Restricción de Integridad de Entidad: ningún componente de clave primaria de una relación puede aceptar nulos. Nulo se referirá a información faltante. El concepto de unicidad es parte de la definición de clave primaria en sí.

Restricción de Integridad Referencial: la base de datos no puede contener valores de clave ajena sin concordancia. La posibilidad de aceptación de valores nulos deberá ser evaluada por el analista, al igual de las opciones a seguir (restricción, propagación o anulación)

CHAR(n)	Cadena de caracteres de longitud fija, tiene un tamaño n bytes. Si no se especifica n la ORACLE le da un tamaño de 255 bytes. El tamaño máximo es 2000 bytes y el mínimo 1 byte. El tamaño máximo en PL/SQL es 32767 bytes y el mínimo 1 byte. CHARACTER es sinónimo de CHAR.
VARCHAR2(n)	Cadena de caracteres de longitud variable, tiene un tamaño máximo de n bytes. Es obligatorio especificar el tamaño. El tamaño máximo es 4000 bytes y el mínimo 1 byte. El tamaño máximo en PL/SQL es 32767 bytes y el mínimo 1 byte. STRING y VARCHAR son sinónimos de VARCHAR2.
NUMBER(p,s)	Número de *p* dígitos de los cuales *s* son decimales. No es obligatorio especificar el tamaño. El tamaño de p va de 1 a 38 y el s desde -84 a 127. El tamaño en PL/SQL 1E-130 .. 10E125. Sinónimos: números de coma fija: DEC,DECIMAL,NUMERIC enteros: INTEGER (sinónimo de NUMBER(38)), INT, SMALLINT coma flotante: DOUBLE PRECISION FLOAT REAL.
DATE	Fecha válida. Desde el 1 de enero del 4712 AC hasta el 31 de diciembre del 9999 DC. (en Oracle7 = 4712 DC)
LONG	Cadena de caracteres de longitud variable. Es una versión más grande de VARCHAR2. El tamaño máximo en BD es 2 Gigabytes.
CLOB	Cadena de caracteres de longitud variable. Es una versión más grande de VARCHAR2. El tamaño máximo en BD es 4 Gigabytes. Es recomendable usar CLOB o BLOB en lugar de LONG.
BLOB	Objeto binario de longitud variable. Es una versión más grande de RAW. El tamaño máximo en BD es 4 Gigabytes.
BFILE	Apuntador a un archivo en disco. El tamaño máximo en BD es 4 Gigabytes
TIMESTAMP (f)	Es un fecha que contiene que contiene fracciones de segundo. Con *f* se define el número de dígitos en la fracción de segundo. Así, f puede valer desde 0 hasta 9, el valor por defecto es 6.
INTERVAL YEAR (y) TO MONTH	Periodo de tiempo definido en años y meses donde *y* es el número de dígitos del año. Puede valer de 0 a 9. (por defecto es 2)
INTERVAL DAY (d) TO SECOND (f)	Periodo de tiempo definido en días, horas, minutos y segundos. *d* es el máximo numero de dígitos en el día, *f* es el máximo numero de dígitos en el campo de segundos. *d* va de 0 a 9. (por defecto es 2), *f* va de 0 a 9. (por defecto es 6)
ROWID	Cadena hexadecimal que representa de forma única una fila en una tabla (pero no única en cualquier tabla).
UROWID	Cadena hexadecimal que representa de forma única una fila ORDENADA en una tabla (pero no única en cualquier tabla).
RAW(n)	Objeto binario de longitud variable. Es obligatorio especificar el tamaño. El tamaño máximo es 2000 bytes y el mínimo 1 byte. El tamaño máximo en PL/SQL es 32767 bytes y el mínimo 1 byte.
LONG RAW	Objeto binario de longitud variable. El tamaño máximo es 2 Gigabytes. El tamaño máximo en PL/SQL es 32767 bytes y el mínimo 1 byte.

Tratamiento de clave primaria

Se designa una columna o combinación de columnas como clave primaria de la tabla.

Se puede definir la restricción a nivel de columna o a nivel de tabla.

- A nivel de columna: Se define dentro de una restricción (*constraint)* asociada a la misma. La definición de una clave primaria lleva implícita la restricción de valores no nulos. La misma columna puede ser definida como primaria y foránea.
- A nivel de tabla: Se utiliza para definir claves primarias compuestas. Se declara antes o después de haber introducido las columnas de la tabla en cuestión.

Formato:

A nivel columna:

```
atributo tipo [CONSTRAINT nombre] PRIMARY KEY
```

A nivel tabla:

```
[CONSTRAINT nombre] PRIMARY KEY(colum1,.. colum2 ..)
```

Ejemplo:

A nivel columna:

```
Dni integer [CONSTRAINT pk_dni] PRIMARY KEY
```

A nivel tabla:

```
CONSTRAINT pk_nomapelli PRIMARY KEY(nom, apelli)
```

Tratamiento de valores nulos

Esta restricción especifica si una columna puede contener o no valores nulos. Por defecto se asume que la columna admite valores nulos.

```
atributo tipo [CONSTRAINT nombre]
NOT NULL | NULL
```

Ejemplo:

```
Dni integer CONSTRAINT nn_dni NOT
NULL
```

Tratamiento de claves alternativas

Para las claves alternativas, se designa una columna o combinación de columnas como clave única alternativa a la clave seleccionada como primaria (sin duplicados).

Existen dos posibilidades para definir una clave alternativa:

- asociada a la columna:

  ```
  atributo tipo [CONSTRAINT nombre] UNIQUE
  ```

  Ejemplo:

  ```
  nombre char CONSTRAINT unq_nombre UNIQUE
  ```

- asociada a la tabla

  ```
  [CONSTRAINT nombre] UNIQUE (columna1, columna2 ...)
  ```

  Ejemplo:

  ```
  CONSTRAINT unq_nomapell UNIQUE(nombre, apellidos)
  ```

Tratamiento de claves foráneas

La integridad referencial de claves foráneas a nivel columna se declara mediante la palabra REFERENCES. Si no se especifica la columna o columnas, se asume por defecto que estamos referenciando a su clave primaria.

Acciones a realizar:

- Restringir (*restrict, no action),*
- Propagar (*cascade*),
- Poner a nulos (*set null*),
- Poner valor por omisión (*set default*) (no implementado en Oracle)

Se especifica mediante la cláusula **on [delete | update] [restrict | no action | cascade | set null | set default]**. Por omisión la operación se asume restringida.

```
atributo tipo [CONSTRAINT nombre] REFERENCES tabla[(columna)] [ON DELETE CASCADE]
```

Ejemplo:

```
Dep char(10) CONSTRAINT fk_dep_nom REFERENCES departamento(nom) ON DELETE CASCADE
```

Si las restricciones se definen a nivel de tabla se especifica la columna o composición de columnas que forman parte de la clave ajena después de la palabra clave FOREIGN KEY y la columna o columnas a la que se hace referencia después de la palabra clave REFERENCES.

```
[CONSTRAINT nombre]  FOREIGN KEY (columna1, columna2, ...)
REFERENCES n_tabla[(columna1, columna2,...)] [ON DELETE CASCADE]
```

Ejemplo:

```
CONSTRAINT fk_dep_nom FOREIGN KEY(dep) REFERENCES
departamento(nom) ON UPDATE SET NULL
```

Tratamiento de las restricciones de integridad estáticas

Las restricciones de integridad estáticas serán aquellas fórmulas bien formadas de primer orden construidas con atributos de la tabla como términos básicos, que debe satisfacerse en todos los estados válidos para las tuplas de una relación.

Estas restricciones de integridad están especificadas mediante una restricción de tipo **CHECK** asociada a una columna o una tabla.

La implementación de esta restricción en una columna sería:

```
atributo tipo [CONSTRAINT nombre] CHECK (condición)
```

Ejemplo:

```
Saldo integer CONSTRAINT chk_saldo CHECK (saldo != 2000.00)
```

La condición puede hacer referencia a cualquier columna de la tabla.

No se pueden introducir en estas condiciones atributos de otras tablas. En ese caso, sólo queda el recurso de programar explícitamente la restricción de integridad, lo que habitualmente se hace generando disparadores ("triggers") de la base de datos.

La implementación a nivel de tabla sería equivalente.

La sentencia *ALTER TABLE* sirve para añadir o redefinir una columna, añadir o borrar una restricción de integridad y para activar o desactivar cualquier restricción de integridad o disparo.

```
ALTER TABLE [esquema.]nom_tabla
[ADD nom_col + def_de_restriccion_col]
[MODIFY nom_col tipo + def_de_restriccion_col]
[ADD + def_restriccion_tabla]
[DROP PRIMARY KEY | UNIQUE | CONSTRAINT nombre [CASCADE]]
[ENABLE PRIMARY KEY | UNIQUE | CONSTRAINT nombre | ALL TRIGGERS]
[DISABLE PRIMARY KEY | UNIQUE | CONSTRAINT nombre [CASCADE]]
```

en donde:

- *def_de_restriccion_col* y *def_restriccion_tabla* se refieren a las declaraciones de restricciones que se pueden realizar en el ámbito de una columna o de la tabla en su conjunto respectivamente.
- Si deshabilitamos una restricción tenemos la posibilidad de deshabilitar en cascada todas las restricciones que dependan de ella a través de la cláusula *cascade*.
- Si habilitamos una restricción de integridad referencial, la restricción de clave primaria o única asociada debe ser habilitada con anterioridad (si no lo estaba ya).
- Las opciones asociadas a UNIQUE identifican la clave única implicada en la modificación indicándola entre paréntesis.
- La cláusula *ENABLE ALL TRIGGERS* permite activar disparos que puedan haber sido desactivados con la orden *ALTER TRIGGER* opción *DISABLE* proporcionada por ORACLE.

Modificación de restricciones

Diccionario de Datos de ORACLE

Es un conjunto de tablas de sólo lectura con los metadatos (descripción del esquema).

Información: Nombres de usuarios, seguridad (accesos permitidos, privilegios, roles...), información sobre los objetos, restricciones de integridad, diversas estadísticas. En general, mantienen información sobre las llamadas ESTRUCTURAS INTERNAS de la BD.

Las tablas del diccionario son propiedad del usuario SYS. El usuario SYSTEM es propietario de diversas vistas sobre esas tablas, las cuales pueden ser utilizadas por el resto de los usuarios de la BD.

Sólo ORACLE debe escribir y leer en las tablas del diccionario:

Oracle accede cada vez que se ejecuta una sentencia DDL .

Los usuarios podrán acceder a algunas vistas : En general, no se deben modificar estas vistas.

Prefijos en las Vistas del Diccionario de Datos:

USER_ Objetos que pertenecen al propio usuario.

ALL_ Todos los objetos accesibles por el usuario.

DBA_ Todos los objetos existentes (sólo para los DBA).

Tipos de vistas del catálogo del sistema

Algunas vistas con el prefijo **USER_** (pueden verse en ALL_VIEWS) :

USER_OBJECTS: Lista de todos los objetos pertenecientes al usuario (tablas, vistas, paquetes, índices, triggers, sinónimos...).

USER_TABLES: Lista de todas las tablas del usuario.

USER_VIEWS : Vistas del usuario.

USER_USERS : Diversos datos sobre el usuario.

USER_UPDATABLE_COLUMNS : Columnas que pueden ser modificadas.

USER_JOBS : Tareas pertenecientes al usuario.

USER_TRIGGERS: Disparadores (triggers) del usuario.

USER_SYNONYMS: Sinónimos pertenecientes al usuario.

USER_INDEXES: Índices pertenecientes al usuario.

USER_CONSTRAINTS :Restricciones pertenecientes al usuario.

USER_TAB_PRIVS: Permisos sobre objetos con el usuario involucrado. Si pone _COL_en vez de _TAB_se refiere a las columnas. Se puede distinguir entre:

- **USER_TAB_PRIVS_MADE**: Permisos sobre los objetos del usuario.
- **USER_TAB_PRIVS_RECD**: Permisos recibidos por el usuario.

USER_TAB_COLUMNS :Descripciones de las columnas del usuario.

USER_TAB_COMMENTS y **USER_COL_COMMENTS**: Comentarios sobre las tablas y columnas del usuario, si se han insertado con el comando COMMENT:

```
COMMENT ON [TABLE|COLUMN] <Tabla>[.<Columna> ] IS '<Texto>';
```

Restricciones en Oracle

Las almacena todas en la vista del Diccionario de Datos:

- **USER_CONSTRAINTS**
  - ***CONSTRAINT_NAME**: Nombre de la restricción. Si no se le ha dado uno, Oracle le asigna uno con un código.*
  - ***TABLE_NAME**: Nombre de la tabla con dicha restricción.*
  - ***CONSTRAINT_TYPE**: Es un carácter*
    - **P**: PRIMARY KEY
    - **U**: UNIQUE
    - **R:restricción** de INTEGRIDAD REFERENCIAL
    - **C**: restricción de tipo CHECK (o NOT NULL) con la condición almacenada en el atributo SEARCH_CONDITION
    - **STATUS**: Estado de la restricción (ENABLE o DISABLE).
- **USER_CONS_COLUMNS**

Algunas Sentencias SQL del DBA

El comando **CREATE** para crear objetos puede sustituirse por DROP y ALTER para borrar y modificar el objeto en cuestión:

- **CREATE USER**: Crea un usuario, una cuenta para acceder a la BD.
- **CREATE ROLE**: Crea un conjunto de privilegios con un nombre.
- **CREATE SYNONYM**: Crea un sinónimo. Puede establecerse como sinónimo público (sinónimo accesible para todos los usuarios).
- **CREATE TABLESPACE**: Crea un tablespace , espacio en la BD que puede contener objetos.
- **CREATE ROLLBACK SEGMENT**: Crea un segmento de anulación (rollback), un objeto donde Oracle almacena los datos para deshacer modificaciones.

GRANT: Otorga roles y permisos (o privilegios) del sistema o de objetos a usuarios. Los privilegios se retiran con el comando REVOKE.

ANALYZE: Almacena o borra en el Diccionario de Datos estadísticas sobre el objeto que se especifique. Por ejemplo, para una tabla el resultado se almacenará en USER_TABLES.

AUDIT: Realiza un seguimiento sobre las operaciones ejecutadas o sobre objetos accedidos (usuario, tipo de operación, objeto implicado, fecha y hora). Para detener la auditoria usar NOAUDIT. Los datos se guardan en tablas del diccionario con el texto AUDIT_en su nombre, como DBA_AUDIT_OBJECT, DBA_AUDIT_TRAIL…

Conceder privilegios

- Un propietario de una tabla puede ceder los privilegios a otro usuario mediante la clausula GRANT
- Forma general:

```
GRANT {lista_privilegios | ALL PRIVILEGES}
ON    nombre_objeto
TO{lista_id_aut | PUBLIC}
[WITH GRANT OPTION]
```

- *lista_id_aut* consiste de uno o más de los privilegios vistos, separados por comas
- *nombre_objeto* puede ser una tabla base, una vista, un dominio, un conjunto de caracteres, una comparación o una traducción
- ALL PRIVILEGES cede todos los privilegios a un usuario
- PUBLIC habilita a todos los usuarios (presentes y futuros) los privilegios mencionados
- WITH GRANT OPTION permite que un usuario ceda los privilegios obtenidos a otro usuario

Revocación de privilegios

- Para revocar los privilegios cedidos a un usuario mediante GRANT, se usa la cláusula REVOKE
- Forma general:

```
REVOKE [GRANT OPTION FOR]
   {lista_privilegios | ALL PRIVILEGES}
ON     nombre_objeto
FROM {lista_id_aut | PUBLIC}
[RESTRICT | CASCADE]
```

- ALL PRIVILEGES se refiere a todos los privilegios cedidos a un usuario por el usuario que cedió los privilegios
- GRANT OPTION FOR habilita el paso de los privilegios mediante WITH GRANT OPTION del GRANT para ser revocados separadamente de los privilegios por sí mismos
- REVOKE falla si resulta en un objeto abandonado, como una vista, a menos que sea especificado CASCADE
- Los privilegios otorgados hacia un usuario por otro usuario no son afectados

Roles

- Un rol permite agrupar privilegios comunes para una clase de usuarios.
- Los privilegios pueden ser cedidos o revocados desde los roles, así como a los usuarios.
- Los roles pueden se asignados a los usuarios, e inclusive a otros roles.
- El estándar SQL:99 soporta roles

```
create role manager
grant select on branch to manager
grant update(balance) on account to
  manager
grant all privileges on account to
  manager
grant manager to user1, user2
```

3.2.2 Modelo dimensional OLAP

El diseño de DW y DM se basa en un paradigma multidimensional para la representación de datos que proporciona al menos dos ventajas principales:

- en el lado funcional, puede garantizar tiempos de respuesta rápidos incluso a consultas complejas,
- mientras que en el lado lógico las dimensiones coinciden naturalmente con los criterios seguidos por los trabajadores del conocimiento para realizar sus análisis.

La representación multidimensional se basa en un esquema en estrella que contiene dos tipos de tablas de datos: tablas de dimensiones y tablas de hechos.

Tablas de hechos

Las tablas de hechos suelen hacer referencia a transacciones, y contienen dos tipos de datos:

- Los vínculos -establecidos como pares de llaves primarias (PK) / llaves foráneas (FK) del modelo relacional- a cada tabla de dimensiones que hacen referencia a los datos de la tabla de hechos;
- Los valores numéricos de los atributos que caracterizan a las transacciones y son la medidas que serán empledas para las operaciones OLAP.

Por ejemplo, una tabla de hechos puede contener transacciones de ventas y hacer referencia a varias tablas de dimensiones, como clientes, puntos de venta, productos, proveedores, tiempo.

- Las medidas de interés correspondientes son atributos como la cantidad de artículos vendidos, el precio unitario y el descuento.
- La tabla de hechos permite a los analistas evaluar las tendencias de las ventas a lo largo del tiempo, ya sea totales, o referidas a un grupo de clientes.

Tablas de dimensiones

Las tablas de dimensiones están asociadas a las entidades en torno a las cuales giran los procesos de una organización y corresponden a las entidades principales contenidas en el DW.

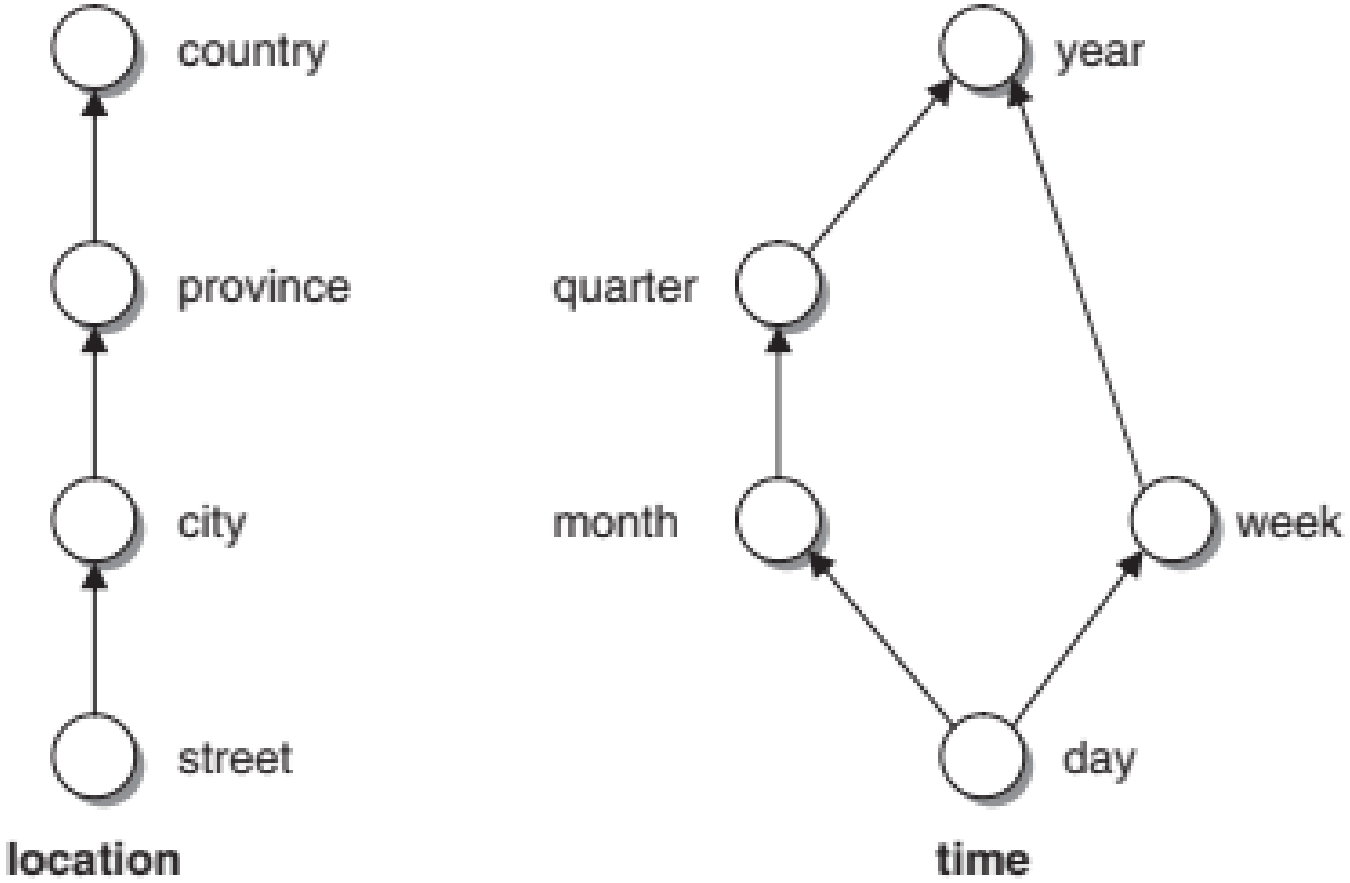
- En la mayoría de los casos, derivan directamente de las tablas maestras almacenadas en sistemas OLTP, como clientes, productos, ventas, ubicaciones y tiempo.

Cada tabla de dimensiones se estructura interna según las relaciones jerárquicas (si están presentes).

- Por ejemplo, la dimensión temporal se basa normalmente en la jerarquías {día, semana,año}. Del mismo modo, la dimensión de ubicación puede organizarse jerárquicamente como {calle, código postal, ciudad, provincia, región, país, área}.

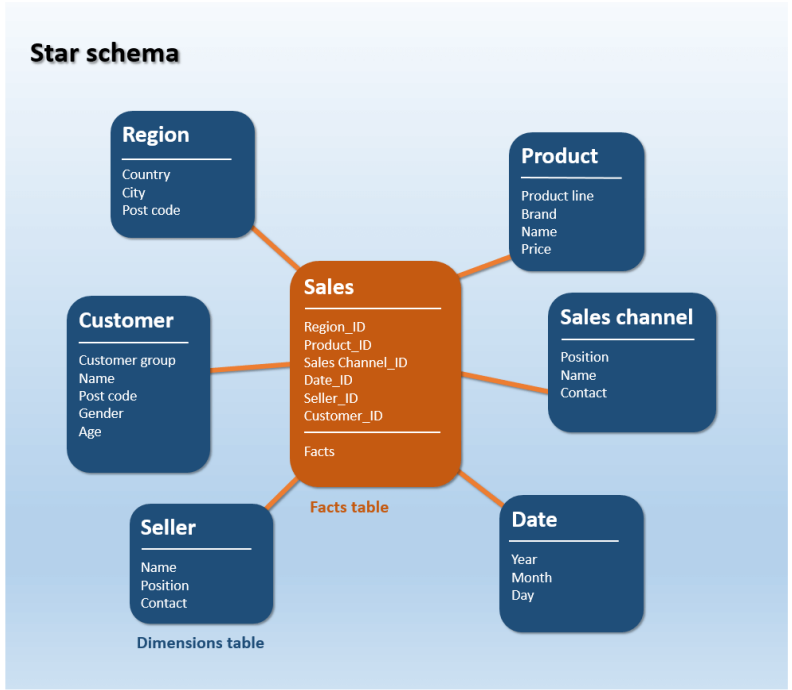
Las dimensiones predeterminan las principales trayectorias a lo largo de las cuales se desarrollará el análisis OLAP.

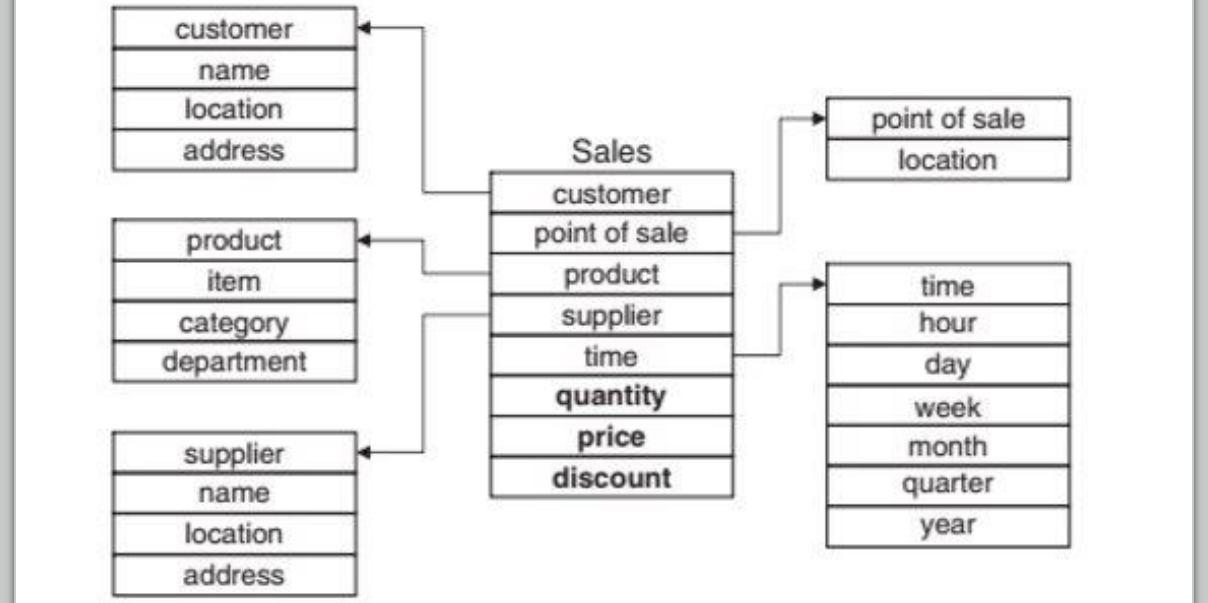
Jerarquías en un modelo multidimensional



3.2.3 Tipos de Esquemas (Estrellas, copo de nieve y otros)

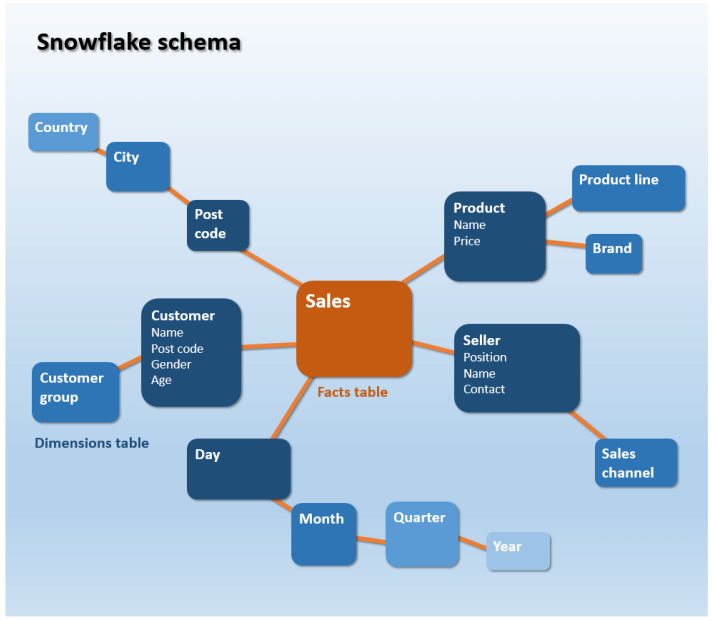
- En el esquema estrella, la tabla de hechos se coloca en el centro del esquema y se enlaza a las tablas de dimensiones a través de las referencias PK/FK.

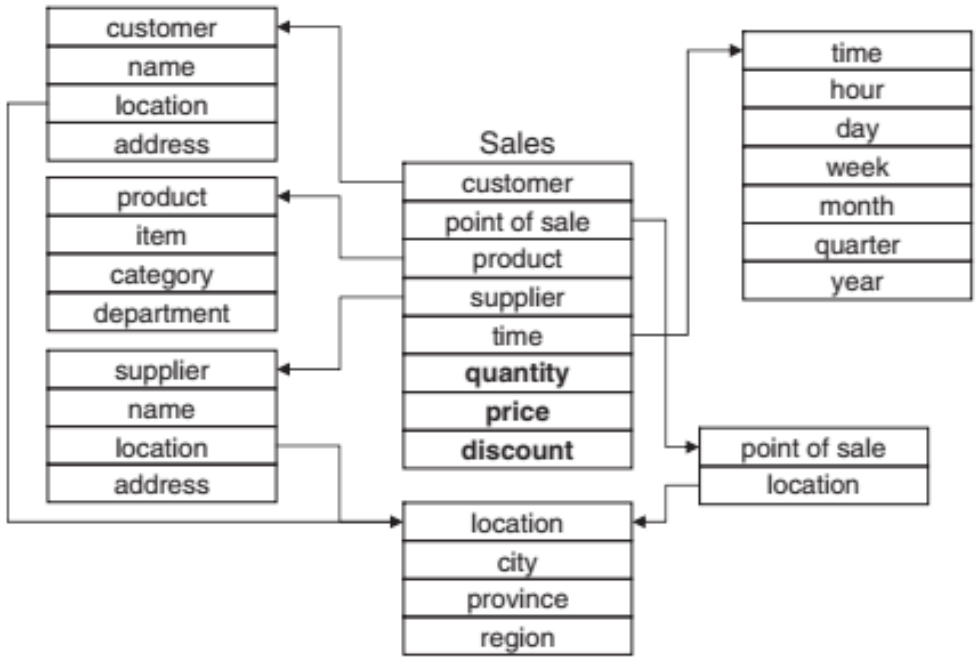




Esquema copo de nieve

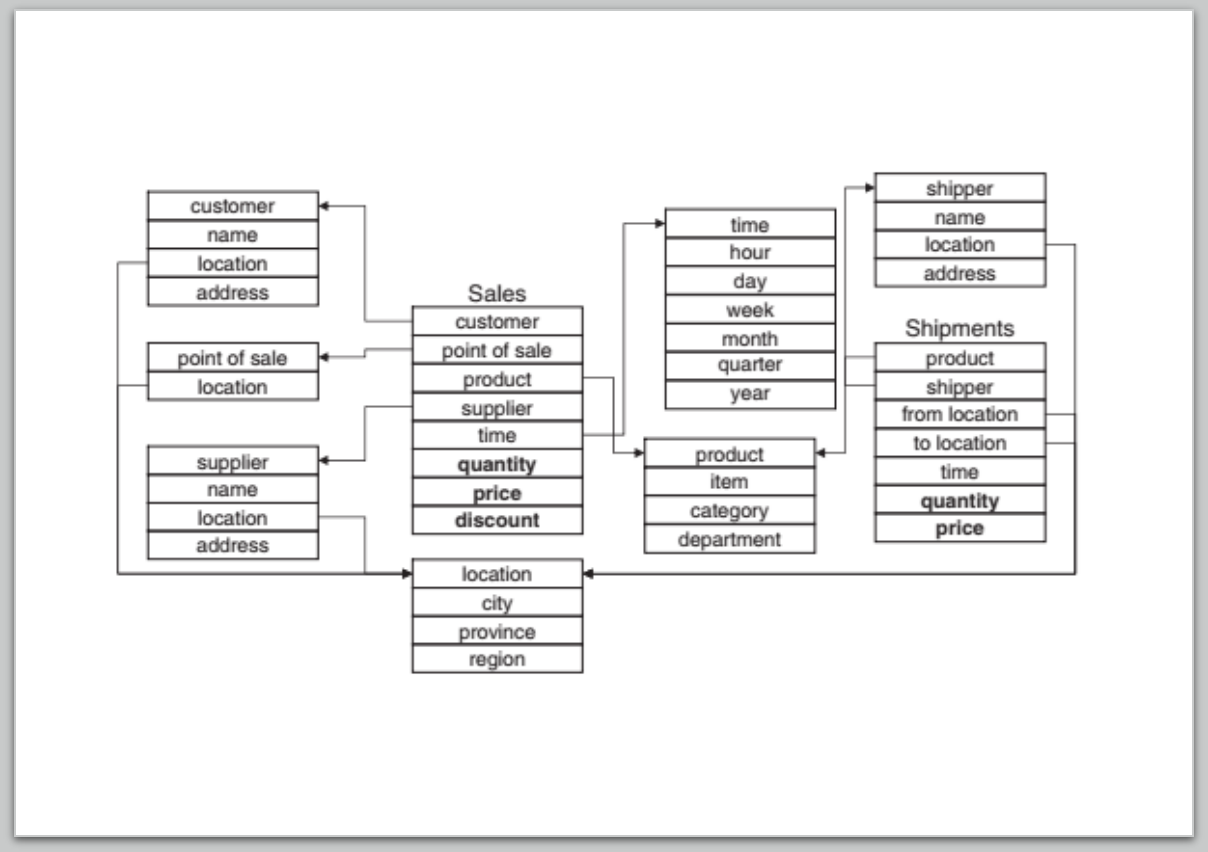
- En el esquema copo de nieve, las tablas de dimensiones se conectan a su vez a otras tablas de dimensiones, con el fin de reducir el uso de memoria

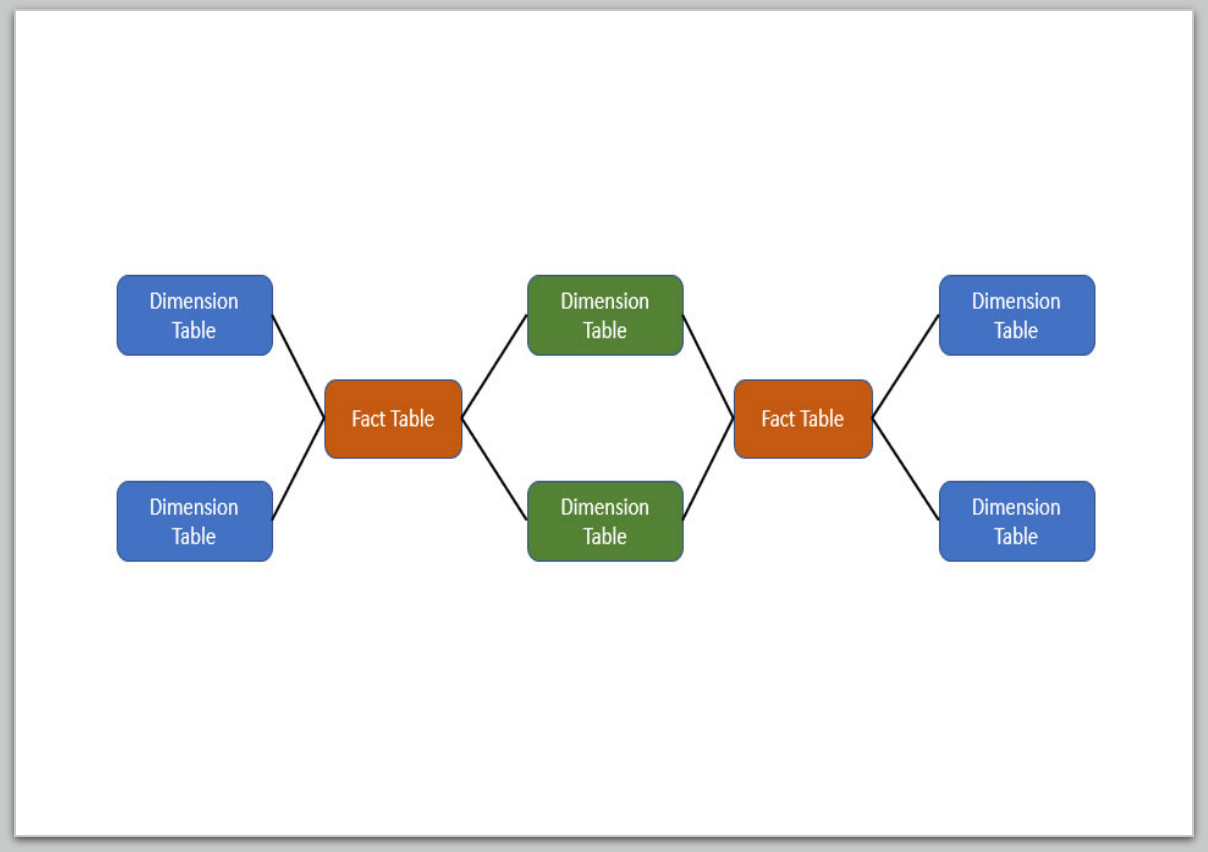




Esquema galaxia

- El esquema galaxia incluye varias tablas de hechos, interconectadas con tablas de dimensiones, vinculadas a su
- vez con otras tablas de dimensiones





3.3 Operadores de cubos de datos

Jerarquías de conceptos y operaciones OLAP

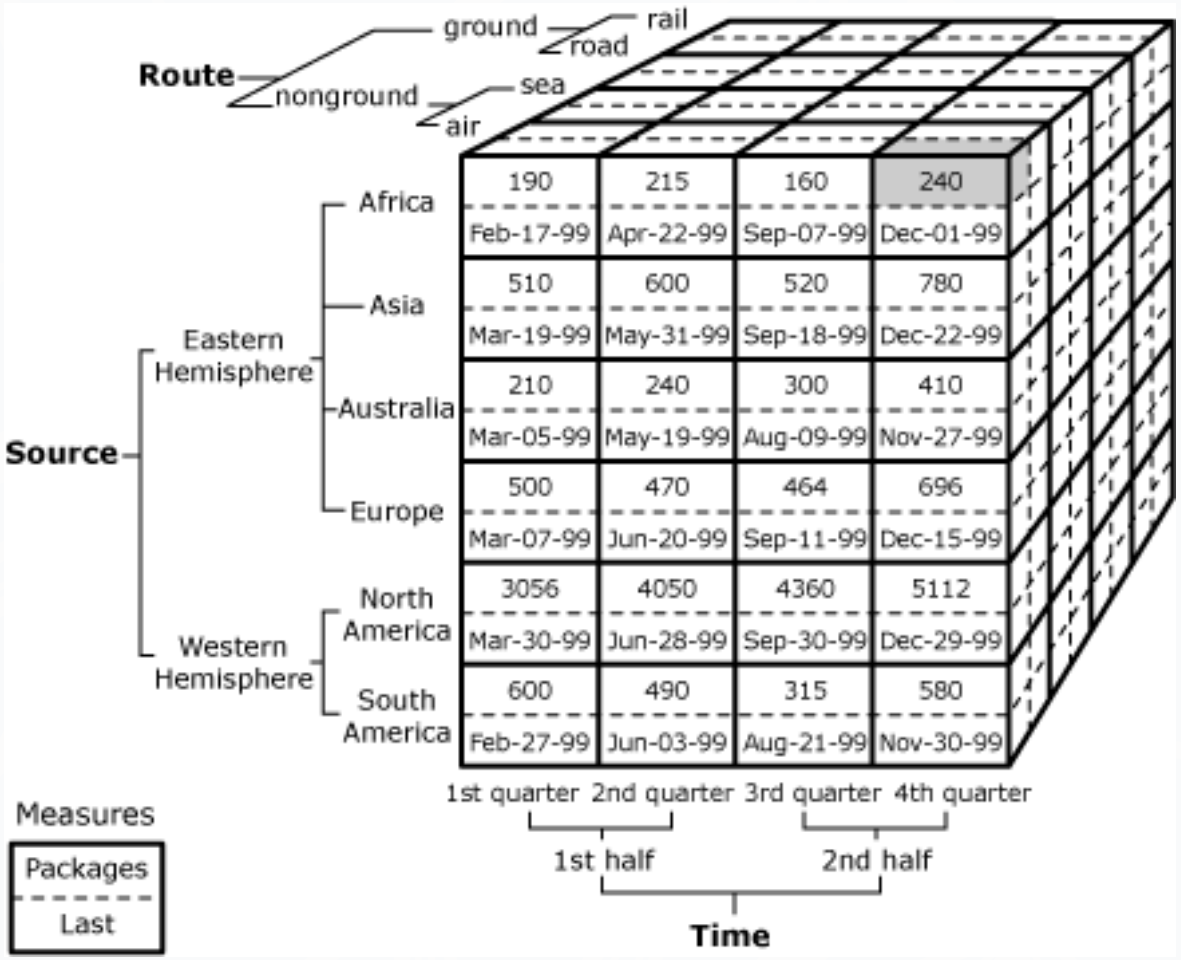
Una jerarquía de conceptos define un conjunto de mapeos de un nivel inferior de conceptos a un nivel superior.

- Por ejemplo, la dimensión {localización} puede originar una jerarquía totalmente ordenada, que se desarrolla a lo largo de la relación {dirección, municipio, provincia, país}

Los análisis OLAP se pueden basar en jerarquías de conceptos para consolidar los datos y crear vistas lógicas a lo largo de las dimensiones de un DW.

- Los tipos de jerarquía específicos pueden estar predefinidos en la plataforma de software utilizada para la creación y administración de un almacén de datos. Para otras jerarquías es necesario que los analistas definan explicativamente las relaciones entre los conceptos.

Las jerarquías de conceptos también se usan para realizar varias operaciones de visualización que tratan con cubos de datos en un DW.



Cubo de datos

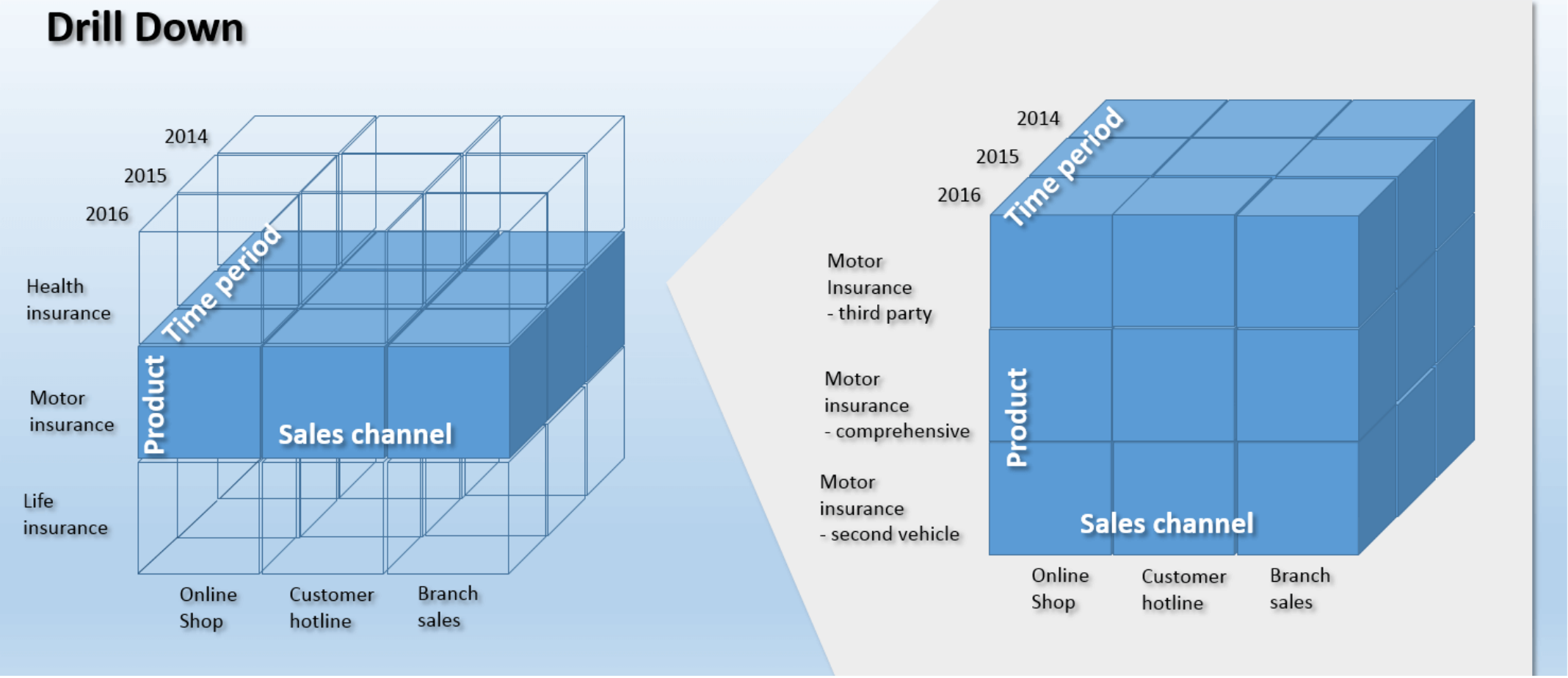
Drill-down

Permite la navegación a través de un cubo de datos desde información agregada y consolidada hasta información más detallada.

Una operación de desglose puede llevarse a cabo de dos maneras:

- Bajar a un nivel inferior a lo largo de una jerarquía de una sola dimensión.
  - Por ejemplo, en el caso de la dimensión {ubicación}, es posible pasar del nivel {provincia} al nivel {ciudad} y desagregar las medidas temáticas de interés sobre todos los registros en los que la ciudad pertenece a la misma provincia.
- Agregar una dimensión.
  - Por ejemplo, la introducción de la dimensión {time} lleva a desagregar las medidas de interés en todos los períodos de tiempo existentes en un cubo de datos.

Drill-down


Drill Down
2014
2015
2016
Time period
Health insurance
Motor insurance
Life insurance
Product
Sales channel
Online Shop
Customer hotline
Branch sales
2014
2015
2016
Time period
Motor Insurance - third party
Motor insurance - comprehensive
Motor insurance - second vehicle
Product
Sales channel
Online Shop
Customer hotline
Branch sales

Roll-up

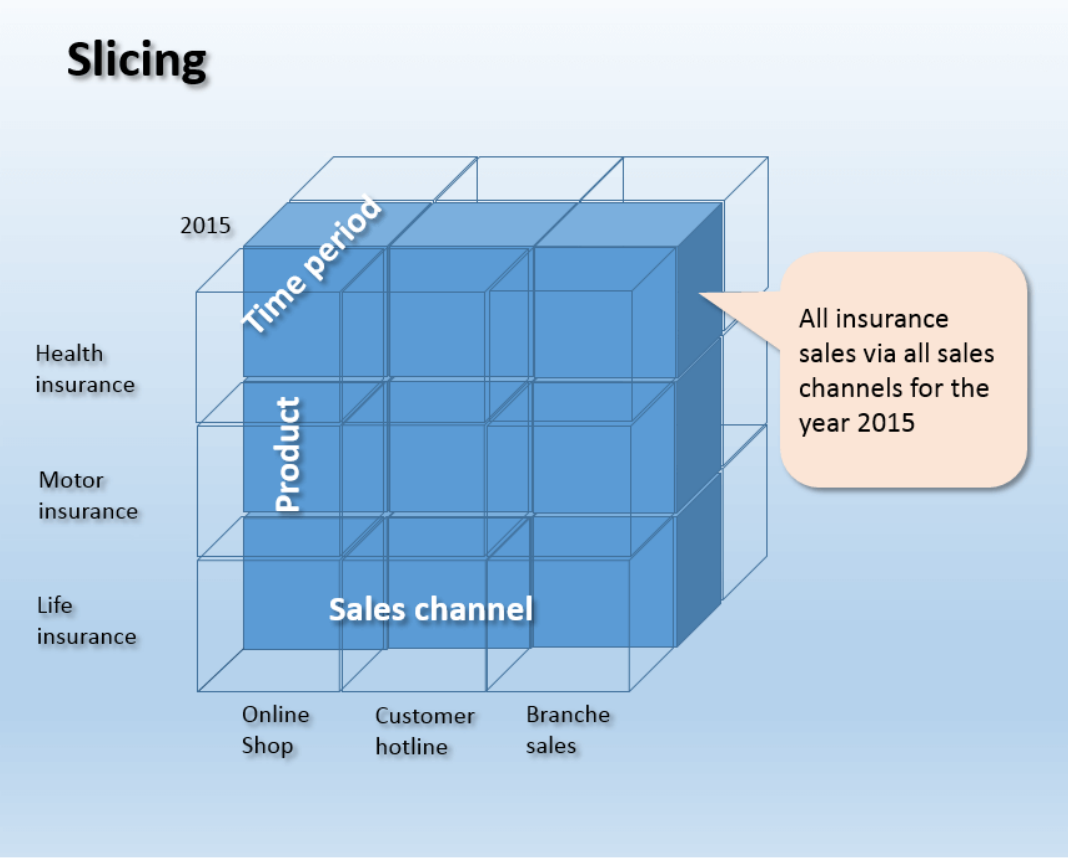
Consiste en una agregación de datos en el cubo, que se puede obtener alternativamente en las dos vías siguientes.

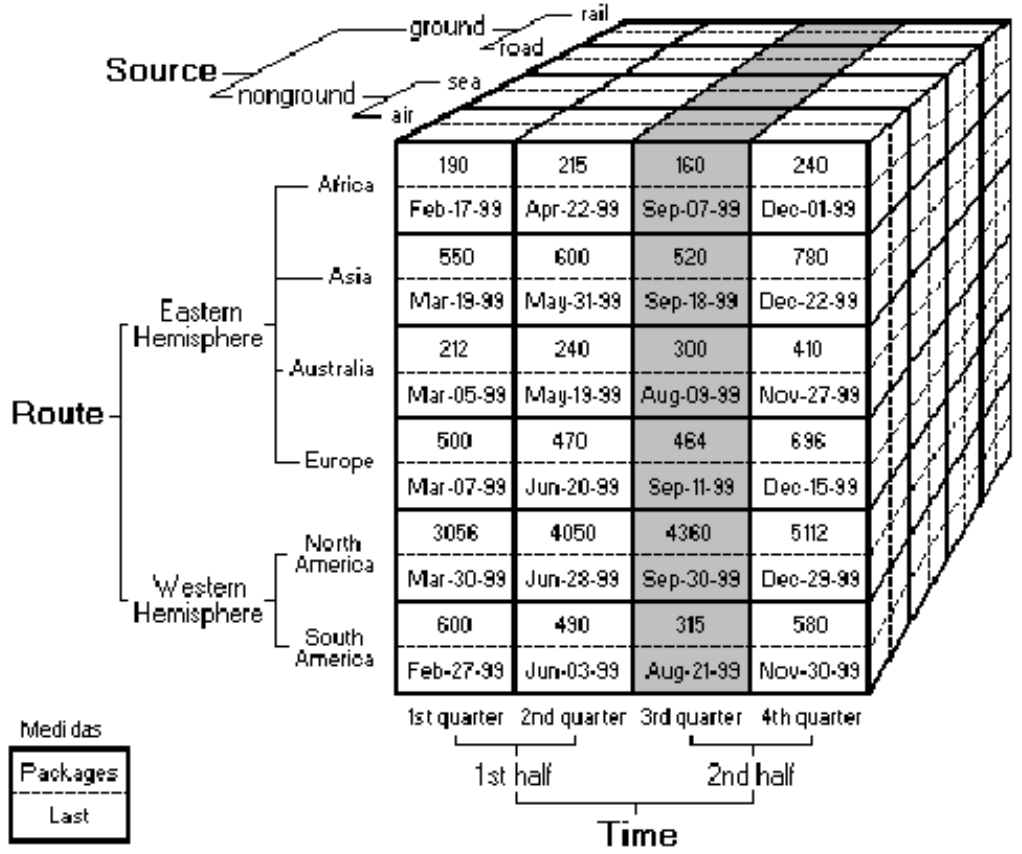
- Avanzar hacia arriba a un nivel superior a lo largo de una única dimensión definida en una jerarquía de conceptos.
  - Por ejemplo, para la dimensión {ubicación} es posible pasar del nivel {ciudad} al nivel {provincia} y consolidar las medidas de interés a través de una suma condicionada grupo por todos los registros en los que la ciudad pertenece a la misma provincia.
- Reducción en una dimensión.
  - Por ejemplo, la eliminación de la dimensión {time} conduce a medidas consolidadas a través de la suma durante todos los períodos de tiempo existentes en el cubo de datos.

Es la operación opuesta a drill-down.

Slice

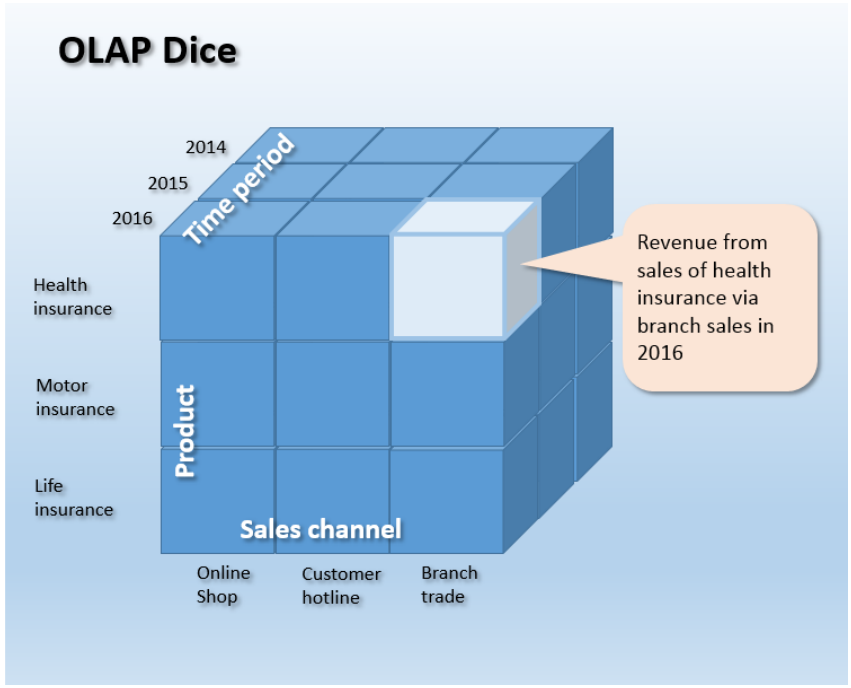
El valor de un atributo se selecciona y se fija a lo largo de una dimensión.

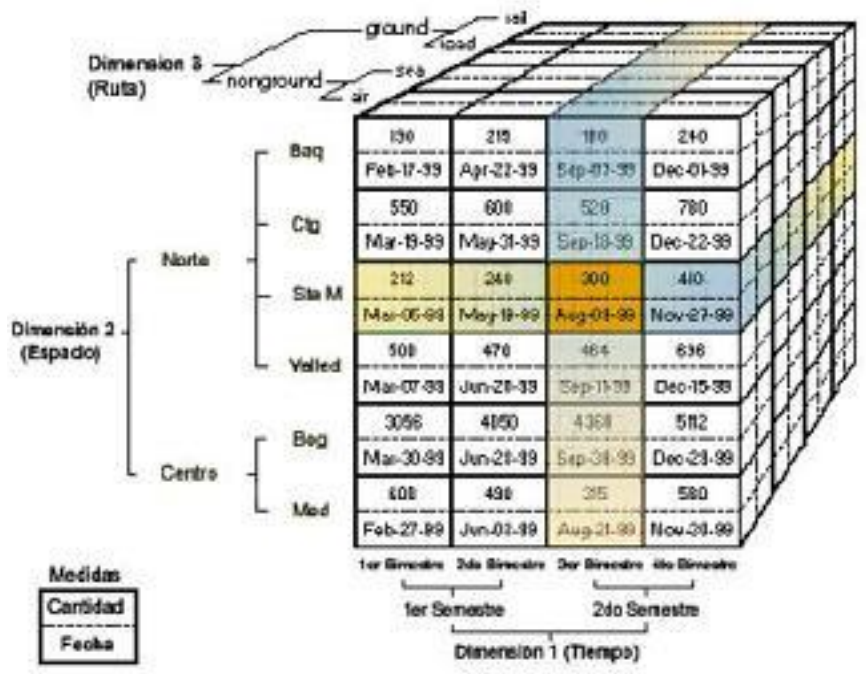


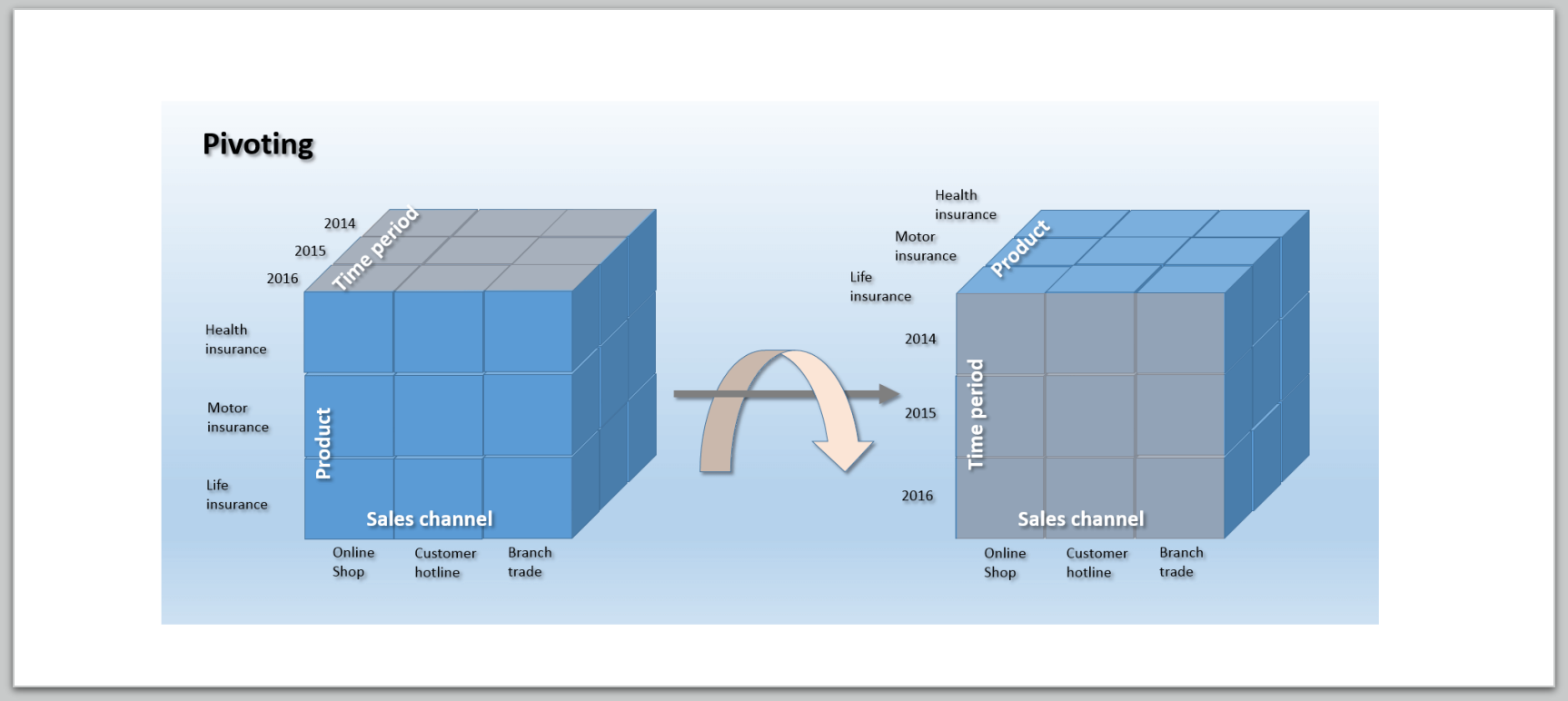


Dice

- Obtiene un cubo en un subespacio seleccionando varias dimensiones simultáneamente.







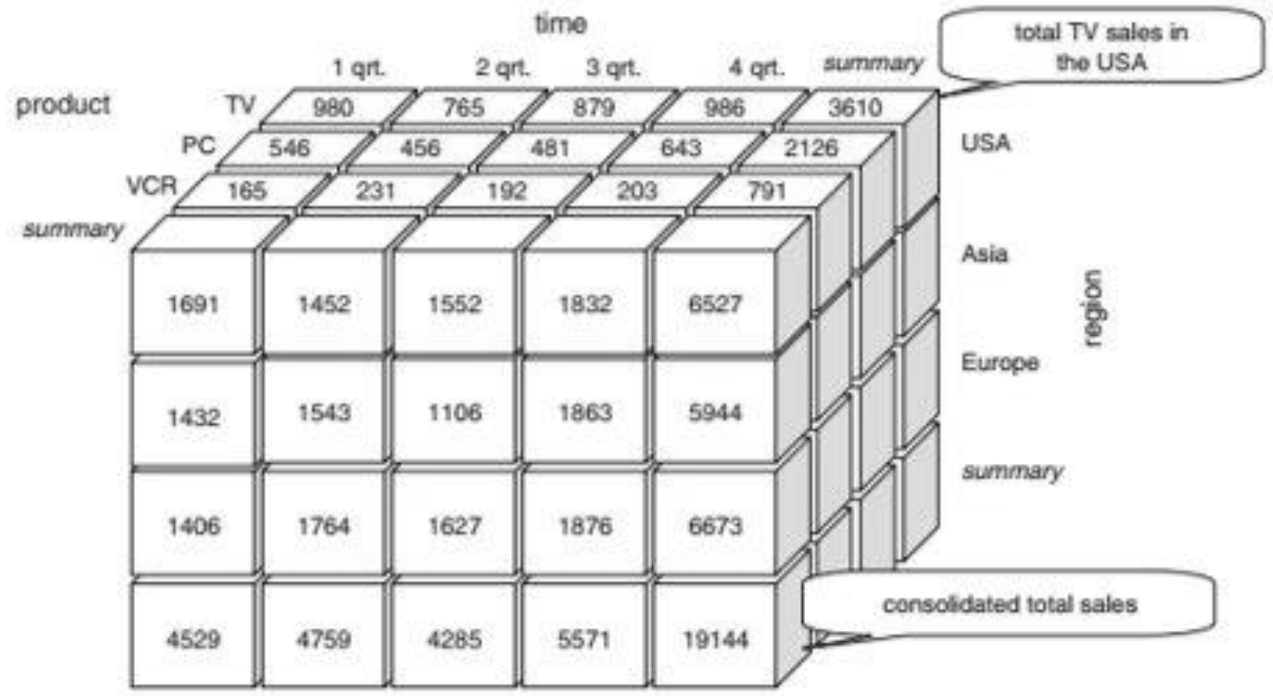
Pivot

- Produce una rotación de los ejes, intercambiando algunas dimensiones para obtener una vista diferente de un cubo de datos.

3.4 Cubos multidimensionales

Una tabla de hechos conectada con n tablas de dimensiones puede representarse mediante un cubo de datos n-dimensional, donde cada eje corresponde a una dimensión.

Los cubos multidimensionales son una extensión natural de las hojas de cálculo populares de dos dimensiones, que se pueden interpretar como cubos bidimensionales.



Multidimensionalidad

Un cubo de datos de cuatro dimensiones no se puede representar gráficamente.

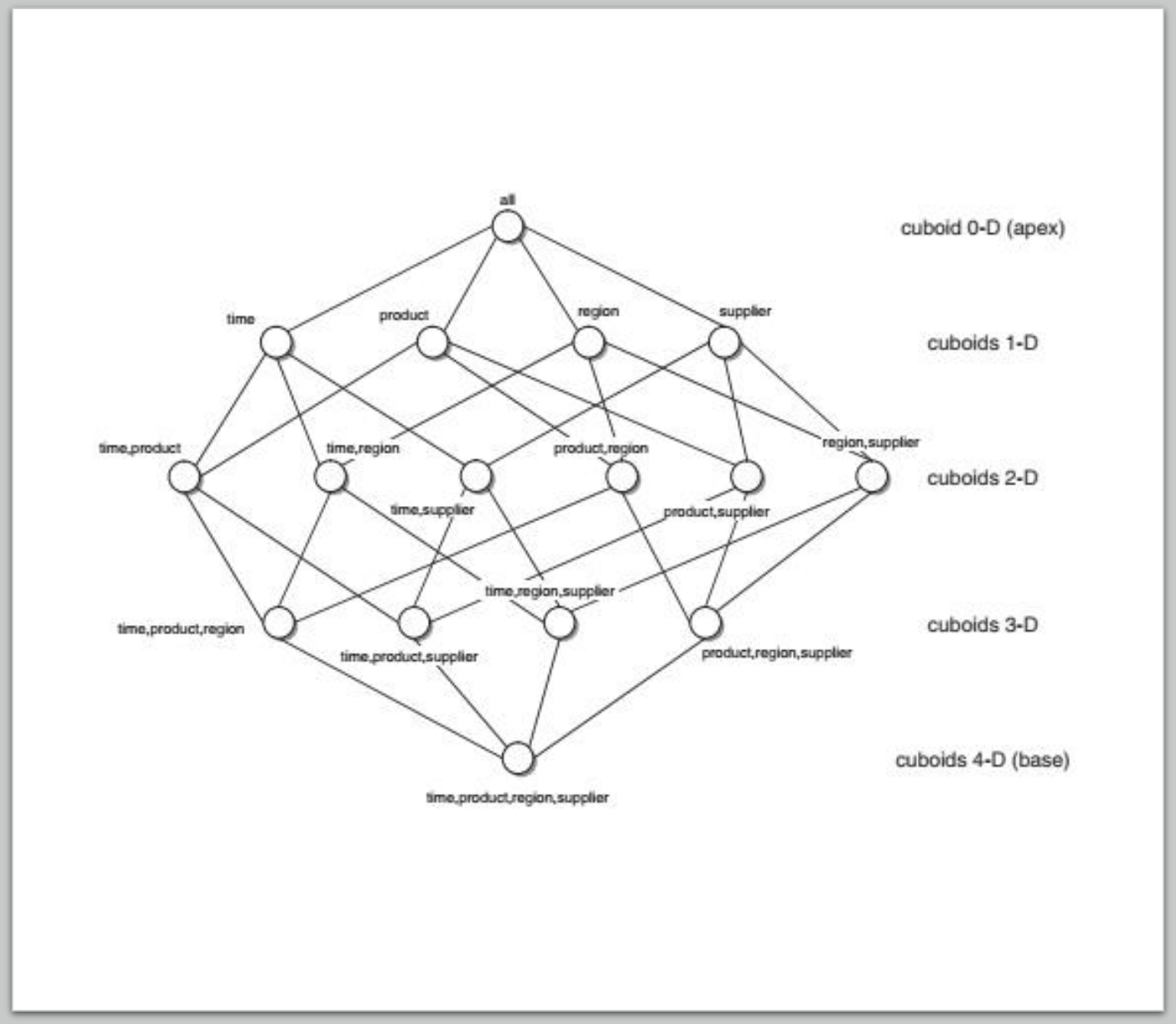
Sin embargo, se pueden obtener cuatro vistas lógicas compuestas por cubos tridimensionales, llamados cuboides, dentro del cubo de cuatro dimensiones, fijando los valores de unidimensionalidad.

Más generalmente, a partir de una tabla de hechos vinculada a n tablas de dimensiones, es posible obtener una red de cuboides, cada uno de ellos correspondiente a un nivel diferente de consolidación a lo largo de una o más dimensiones.

Este tipo de agregación es equivalente en SQL a una suma de consulta derivada de una condición de agrupamiento.

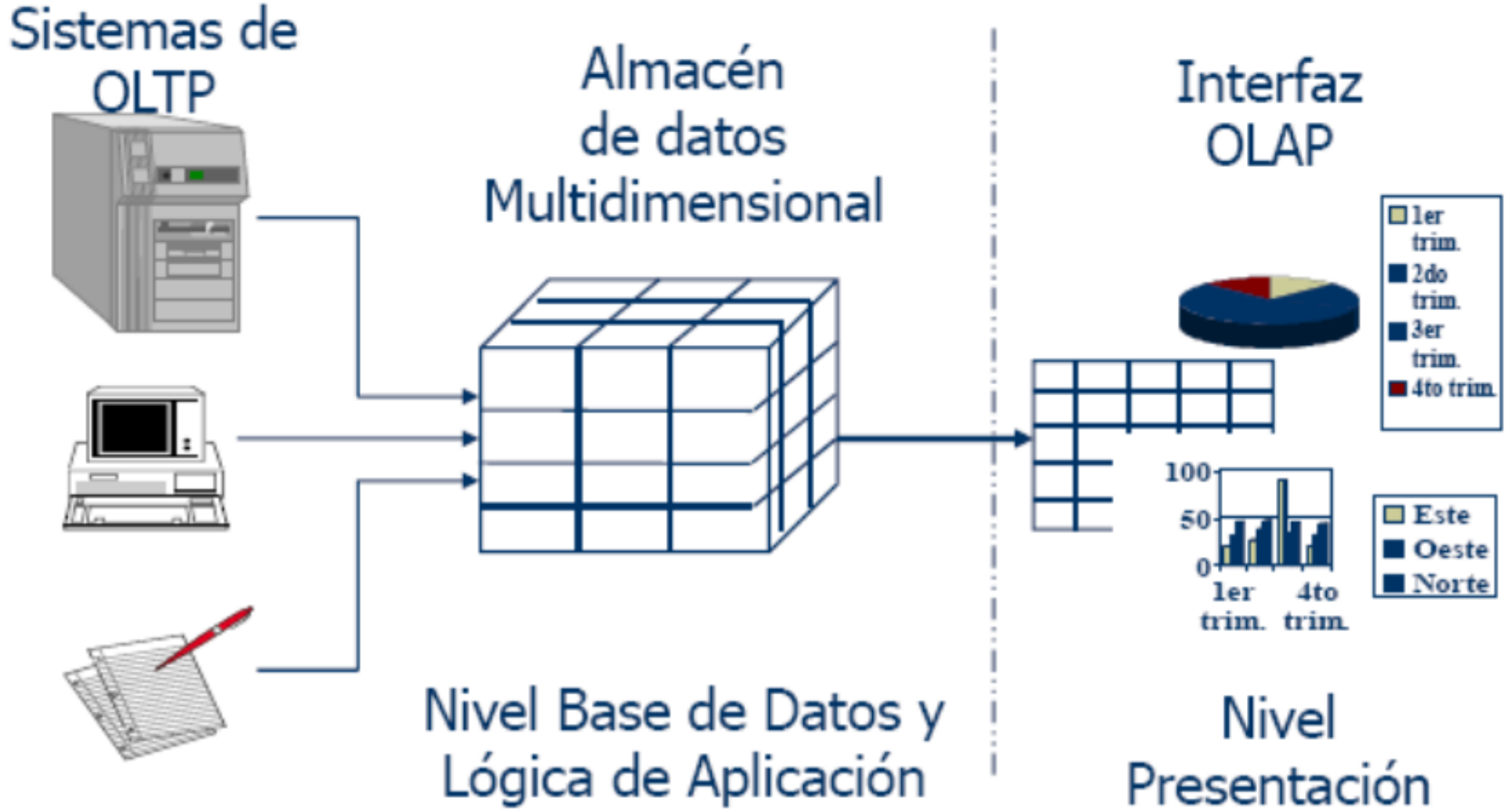
Latice multidimensional

- El cuboide asociado a los datos atómicos, que por lo tanto no implica ningún tipo de consolidación, se denomina cuboide base.
- En el otro extremo, el cuboide ápice se define como el cuboide correspondiente a la consolidación a lo largo de todas las dimensiones, por lo tanto asociado con el total general de la medida de interés.



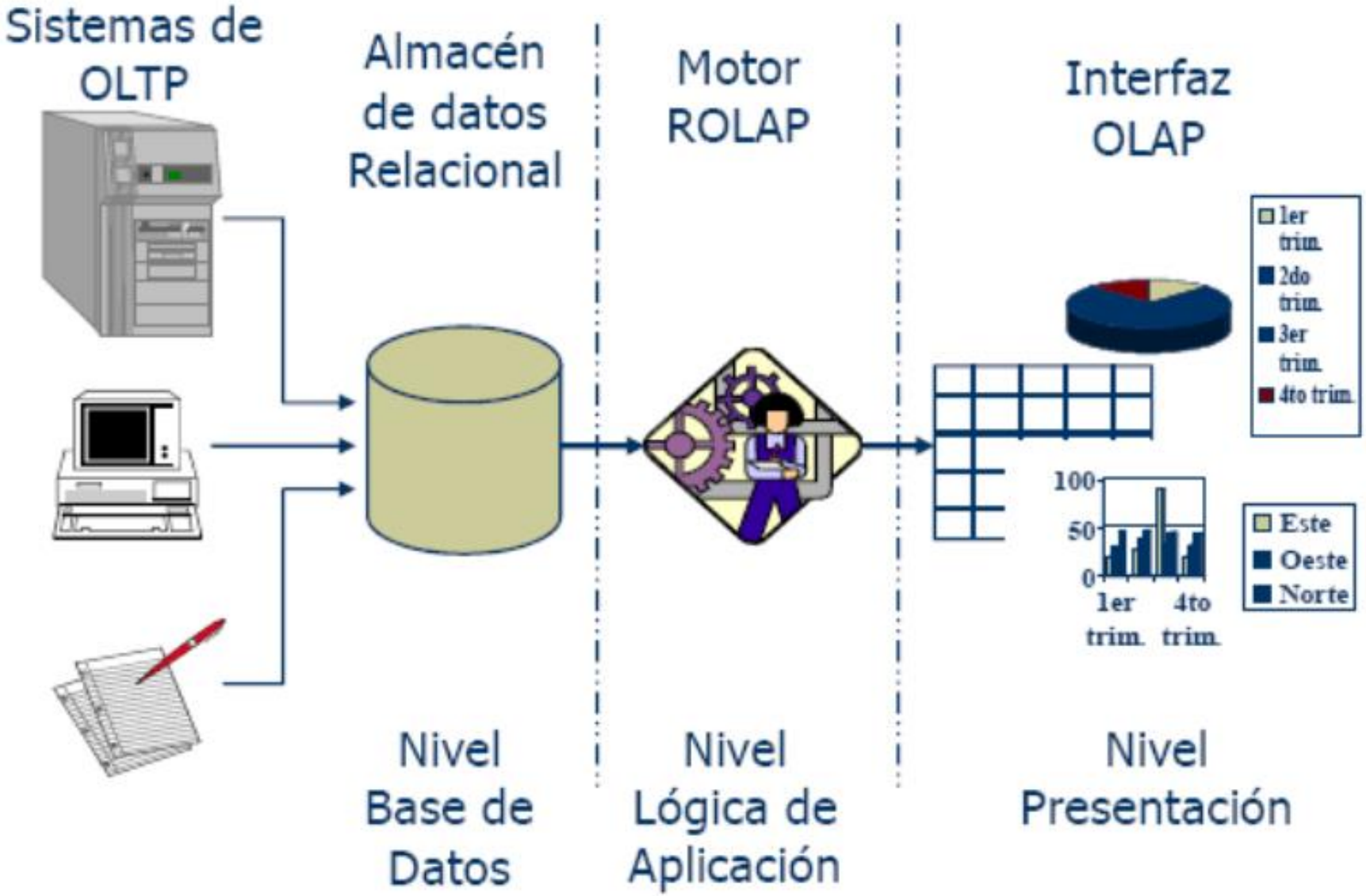
Arquitectura MOLAP

Almacenan y proporcionan acceso a datos y metadatos en formato multidimensional
Especialmente preparadas para entregar datos en estructuras adecuadas al proceso analítico
No llegan a compararse con ROLAP en potencia y capacidad de almacenamiento cuando hay muchas dimensiones
Carga más difícil

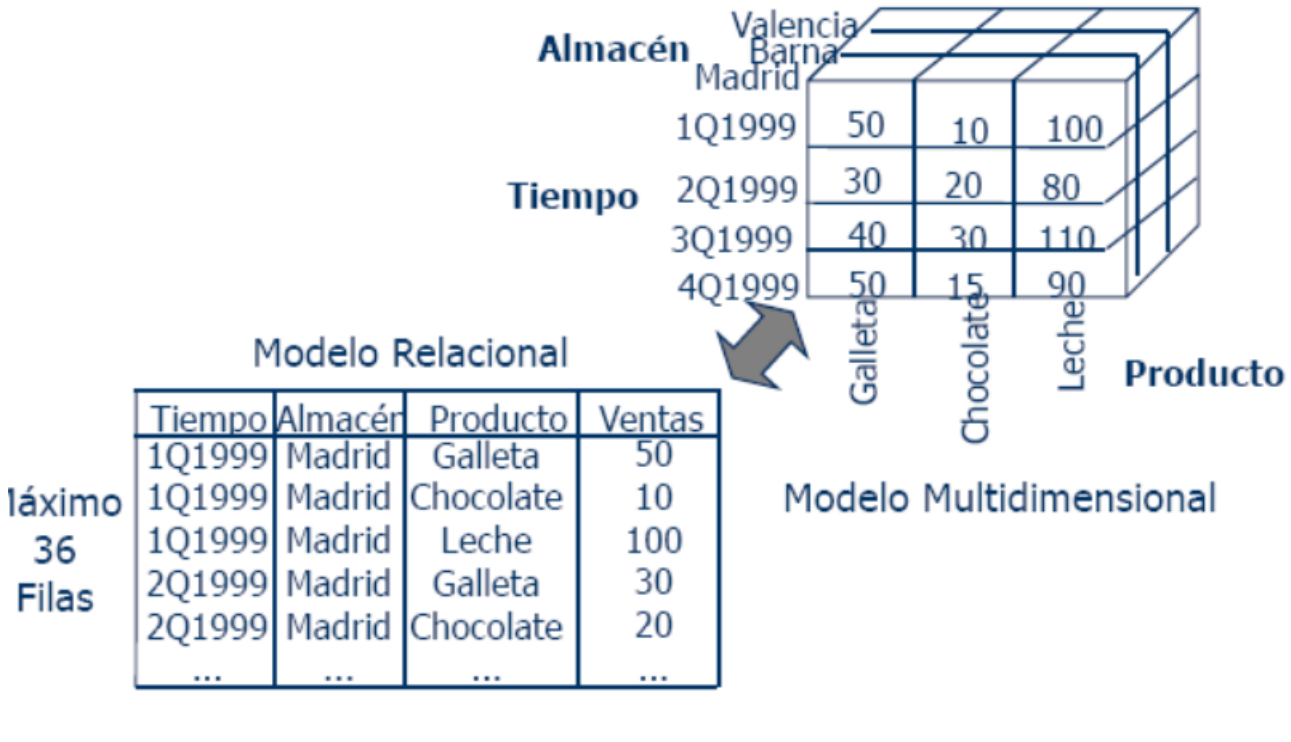

Sistemas de OLTP
Almacén de datos Multidimensional
Interfaz OLAP
1er trim.
2do trim.
3er trim.
4to trim.
100
50
0
1er trim.
4to trim.
Este
Oeste
Norte
Nivel Base de Datos y Lógica de Aplicación
Nivel Presentación

Arquitectura ROLAP

Se integra naturalmente con la tecnología y estándares existentes.
Soporta consultas ad-hoc con eficiencia
Actualización más difícil (carga en tablas relacionales)
La eficiencia puede ser mejorada empleando técnicas como la codificación y la compresión
Superior en paralelismo, concurrencia y distribución
Puede tomar ventaja de la tecnología relacional paralela


Sistemas de OLTP
Almacén de datos Relacional
Motor ROLAP
Interfaz OLAP
1er trim.
2do trim.
3er trim.
4to trim.
100
50
0
1er trim.
4to trim.
Este
Oeste
Norte
Nivel Base de Datos
Nivel Lógica de Aplicación
Nivel Presentación

MOLAP vs ROLAP



MOLAP	ROLAP
2 Niveles	3 Niveles
10 Dimensiones	Más Dimensiones
Volúmenes Medios (10 GB)	Grandes Volúmenes de datos
Mayor necesidad de procesos por lotes (batch)	Mayores costes en cada consulta, se requiere recalcular
Rendimiento crítico	Alta volatilidad de datos
Baja escalabilidad	Alta Escalabilidad
Implantación directa y clara del modelo multidimensional	Implantación simulada, se requiere capa intermedia

MultiDimensional eXpressions

- MDX es un lenguaje de consultas OLAP creado en 1997 por Microsoft. No es un estándar pero diversos fabricantes lo han adoptado como el estándar de facto.
- La principal diferencia del mundo OLAP respecto al mundo relacional radica en que las estructuras dimensionales están jerarquizadas y se representan en forma de árbol y por lo tanto existen relaciones entre los diferentes miembros de las dimensiones.
- Tiene similitudes con el lenguaje SQL, incluyendo funciones y fórmulas especiales orientadas al análisis de estructuras jerarquizadas que presentan relaciones entre los diferentes miembros de las dimensiones.

Sintaxis de MDX

- La sintaxis de MDX es compleja; la mejor manera de ejemplificarla es a través de un caso determinado. Imaginemos un cubo de ventas con las siguientes dimensiones:
  - Temporal de las ventas con niveles de año y mes.
  - Productos vendidos con niveles de familia de productos y productos en sí.
  - Medidas: importe de las ventas y unidades vendidas.
- Para obtener, por ejemplo, el importe de las ventas para el año 2018 para la familia de productos lácteos, la consulta seria:

```
SELECT {[medidas].[importe ventas]} on columns,
{[tiempo].[2018]} on rows
FROM [cubo_ventas]
WHERE ([Familia].[lácteos])
```

Consultas MDX

- Es posible observar que la estructura general de una consulta es de la forma

`SELECT... FROM... WHERE...:`

  - En el *select* se especifica el conjunto de elementos que se van a visualizar y debe especificarse si se devuelve en columnas o filas.
  - En el *from*, el cubo de donde se extrae la información.
  - En el *where*, las condiciones de filtrado.
  - { } permite crear listas de elementos en las selecciones.
  - [ ] encapsulan elementos de las dimensiones y niveles.

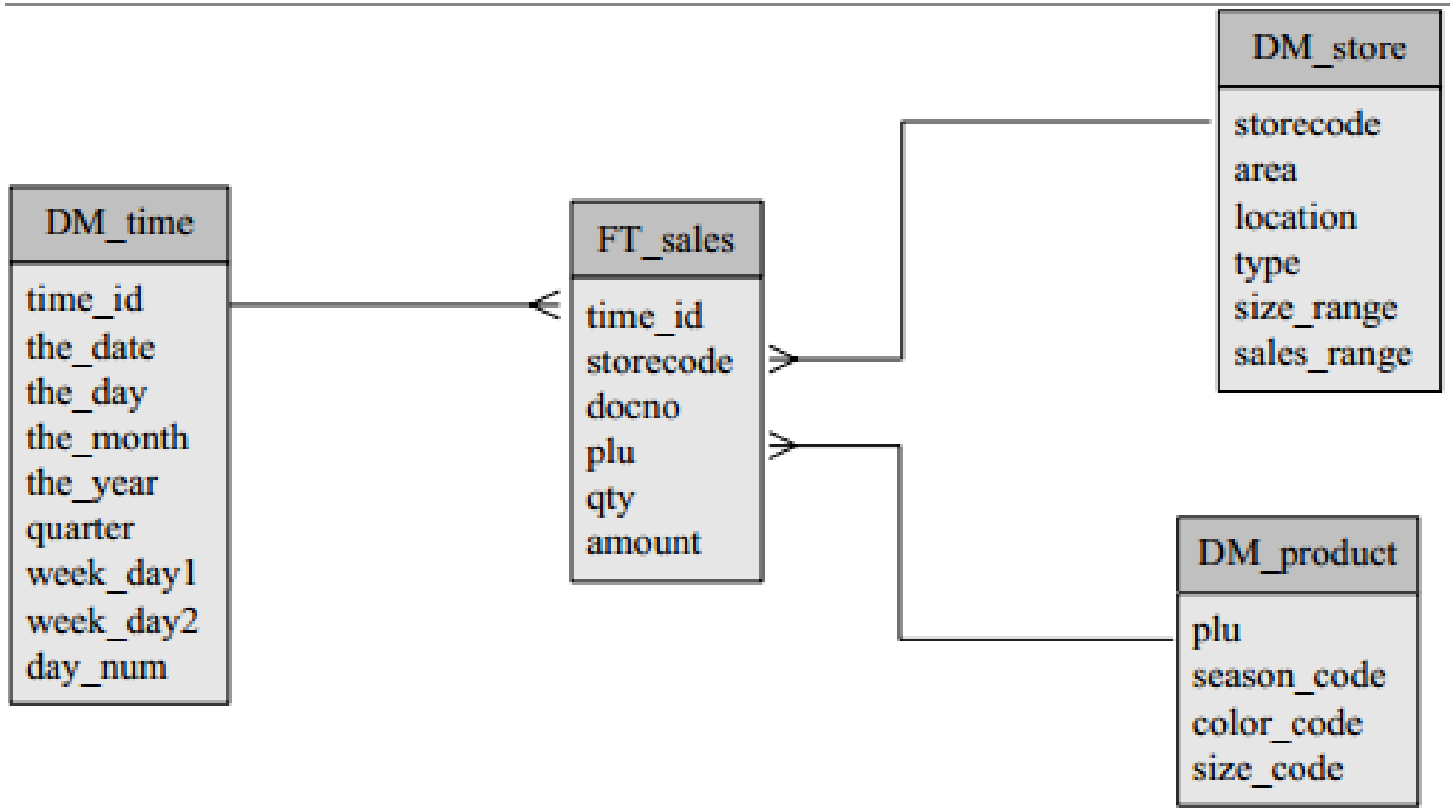
Ejemplo

- Obtener las ventas totales en los paises de Francia, Alemania y Reino Unido:

```
SELECT Measures.[Internet Sales Amount]
on COLUMNS,
{[Customer].[Country].[France],
[Customer].[Country].[Germany],
[Customer].[Country].[United Kingdom]}
on ROWS
FROM [Adventure Works]
```

Country	Internet Sales Amount
France	120,000
Germany	999,999
United Kingdom	55,000

Esquema multidimensional



Ejemplo roll-up

- MDX

```
SELECT [SALES].[AMOUNT] ON
COLUMNS,
[store].[Houston] ON ROWS
FROM SALES
```

- SQL

```
select sum(amount), area
from SALES
where (area='Houston') group by area
```

Ejemplo drill-down

- MDX

```
SELECT [SALES].[AMOUNT] ON COLUMNS,
[time].[2003].[Q4].[Dec].[31],
[time].[2003].[Q4].[Dec].[30],… …,
[time].[2003].[Q4].[Dec].[2],
[time].[2003].[Q4].[Dec].[1] ON ROWS FROM
SALES
```

- SQL

```
select sum(amount), the_date
from SALES
where (the_date='2003-Dec-31')
or (the_date='2003-Dec-30')
or… …or (the_date='2003-Dec-2')
or (the_date='2003-Dec-1') group by
the_date
```

Ejemplo slice

- MDX

```
SELECT [SALES].[AMOUNT] ON COLUMNS,
[store].[Houston].[292] ON ROWS FROM
SALES
```

- SQL

```
select sum(amount), storecode
from SALES
where (storecode='292') group by
storecode
```

Ejemplo dice

- MDX

```
SELECT [SALES].[AMOUNT] ON COLUMNS,
[store].[HK],[store].[NT],
[store].[Houston] ON ROWS
FROM SALES WHERE
[time].[2003].[Q4].[Dec].[24]
```

- SQL

```
select sum(amount), area
from SALES
where ( (area='HK') or (area='NT') or
(area='Houston'))
and (the_date='2003-Dec-24')
group by area
```

Principales funciones y elementos MDX

- El objetivo de MDX está orientado a temas de comparaciones y relaciones jerárquicas entre elementos
- Una de las funcionalidades que distinguen al MDX es el acceder a los elementos utilizando estructura de árbol. Así para un determinado nivel de una dimensión tenemos comandos como:
  - **[Familia].[lácteos].CurrentMember** : Permite acceder al miembro actual
  - **[Familia].[lácteos].Children** : Permite acceder a los hijos de la familia de los lácteos
  - **[Familia].[lácteos].prevMember**: Permite acceder al miembro anterior de la dimensión

Funciones de MDX

- MDX incluye múltiples funciones para realizar consultas a través de la jerarquía existente en el esquema OLAP. Podemos destacar entre ellas:
  - *CurrentMember*: permite acceder al miembro actual.
  - *Children*: permite acceder a todos los hijos de una jerarquía.
  - *prevMember*: permite acceder al miembro anterior de la dimensión.
  - *CrossJoin(conjunto_a,conjunto_b)*: obtiene el producto cartesiano de dos conjuntos de datos.
  - *BottomCount(conjunto_datos,N)*: obtiene un número determinado de elementos de un conjunto, empezando por abajo, opcionalmente ordenado.
  - *BottomSum(conjunto_datos,N,S)*: obtiene de un conjunto ordenado los N elementos cuyo total es como mínimo el especificado (S).

Miembros calculados en MDX

- Una de las funcionalidades más potentes que ofrece el lenguaje MDX es la posibilidad de realizar cálculos complejos tanto dinámicos (en función de los datos que se están analizando en ese momento) como estáticos.
- Los cubos multidimensionales trabajan con medidas (del inglés measures) y con miembros calculados (calculated members).
  - Las medidas son las métricas de la tabla de hechos a las que se aplica una función de agregación (count, distinct count, sum, max, avg, etc.).
- Un miembro calculado es una métrica que tiene como valor el resultado de la aplicación de una fórmula que puede utilizar todos los elementos disponibles en un cubo, así como otras funciones de MDX disponibles.

Funciones de MDX

- *Except(conjunto_a,conjunto_b)*: obtiene la diferencia entre dos conjuntos.
- AVG COUNT VARIANCE VARIANCE y todas las funciones trigonométricas (seno, coseno, tangente, etc.).
- *PeriodsToDate*: devuelve un conjunto de miembros del mismo nivel que un miembro determinado, empezando por el primer miembro del mismo nivel y acabando con el miembro en cuestión, de acuerdo con la restricción del nivel especificado en la dimensión de tiempo.
- *WTD*(<Miembro>): devuelve los miembros de la misma semana del miembro especificado.
- *MTD*(<Miembro>): devuelve los miembros del mismo mes del miembro especificado.
- *QTY*(<Miembro>): devuelve los miembros del mismo trimestre del miembro especificado.
- *YTD*(<Miembro>): devuelve los miembros del mismo año del miembro especificado.
- *ParallelPeriod*: devuelve un miembro de un periodo anterior en la misma posición relativa que el miembro especificado.

Consideraciones del Modelo MD

Ventajas

- Modelo simple, expresado en terminología de negocio y usuario
- Permite realizar consultas muy complejas que van a combinar varias dimensiones de información

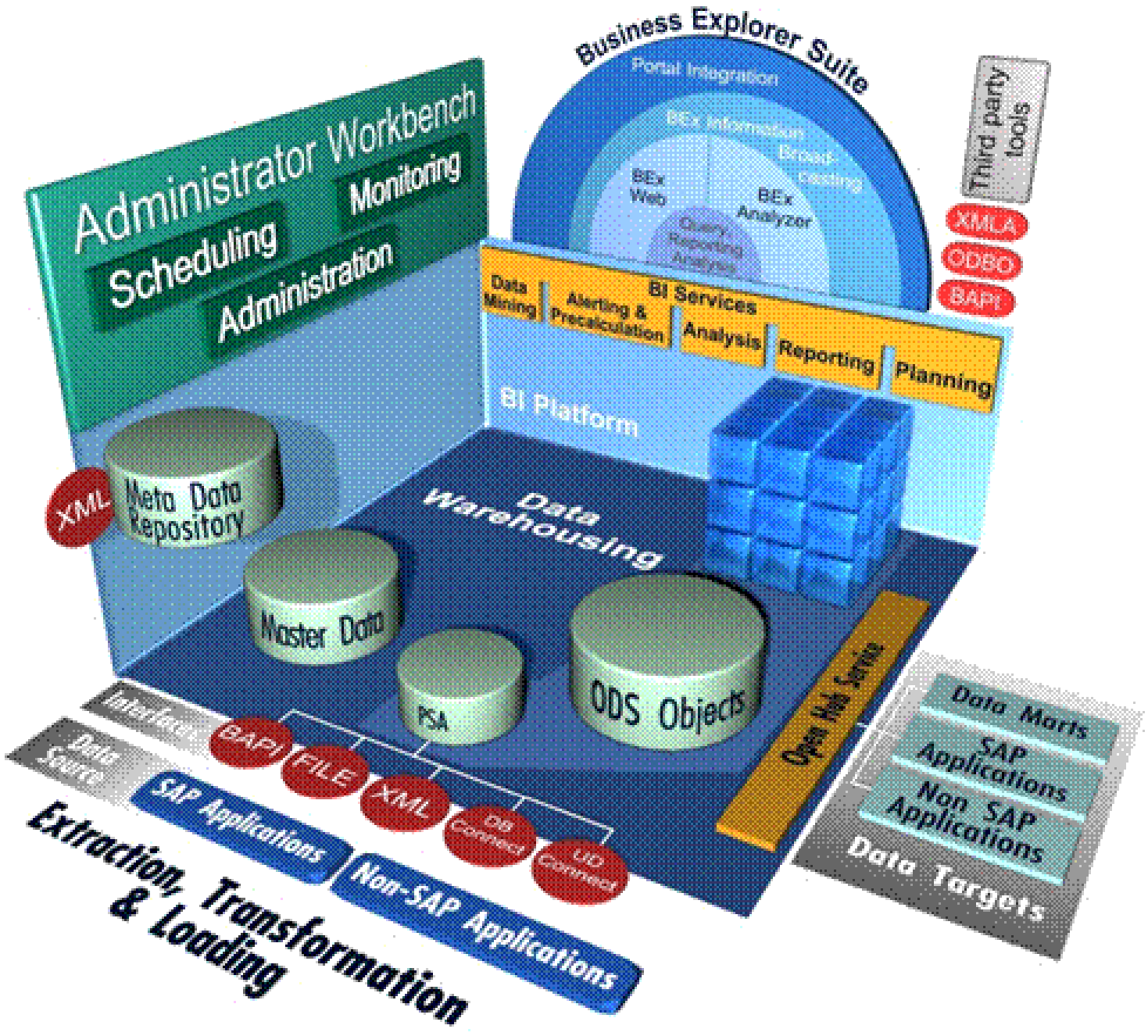
Inconvenientes

- El espacio de almacenamiento crece (densidad)
- Deben relacionarse las dimensiones adecuadas
- El nivel de detalle de los análisis debe ser el adecuado
- El modelo podrá evolucionar con el tiempo, pero los datos deben mantenerse
- El diseño no es tan fácil como puede parecer, y, aunque puede modificarse, muchas decisiones tomadas son críticas, ya que no podremos prescindir de los datos cargados

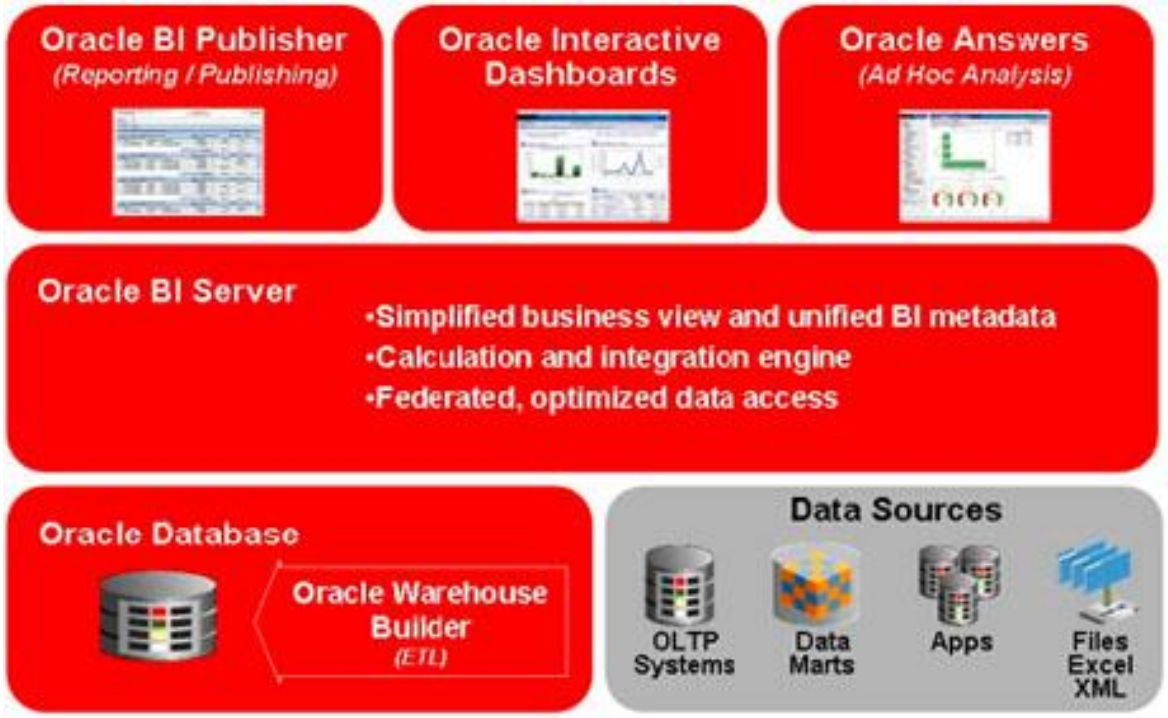
Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms

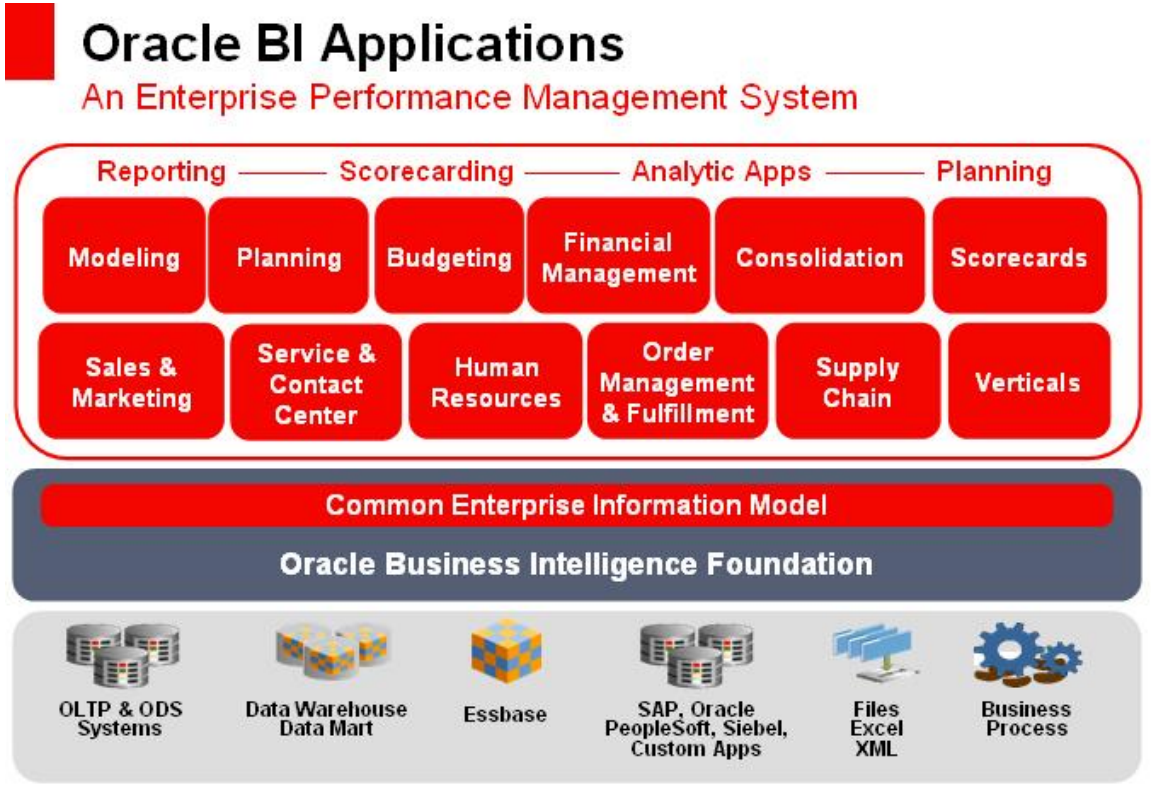
Source: Gartner (February 2020)

SAP


Business Explorer Suite
Portal Integration
BEx Information Broadcasting
BEx Web
BEx Analyzer
Query, Reporting, Analysis
Third party tools
XMLA
ODBO
BAPI
Administrator Workbench
Scheduling
Monitoring
Administration
Data Mining
Alerting & Precalculation
BI Services
Analysis
Reporting
Planning
BI Platform
XML
Meta Data Repository
Data Warehousing
Master Data
PSA
ODS Objects
Open Hub Service
Data Marts
SAP Applications
Non SAP Applications
Data Targets
Interface
Data Source
BAPI
FILE
XML
DB Connect
UD Connect
SAP Applications
Non-SAP Applications
Extraction, Transformation & Loading

Oracle


Oracle BI Publisher
(Reporting / Publishing)
Oracle Interactive Dashboards
Oracle Answers
(Ad Hoc Analysis)
Oracle BI Server
•Simplified business view and unified BI metadata
•Calculation and integration engine
•Federated, optimized data access
Oracle Database
Oracle Warehouse Builder
(ETL)
Data Sources
OLTP Systems
Data Marts
Apps
Files Excel XML


Oracle BI Applications
An Enterprise Performance Management System
Reporting
Scorecarding
Analytic Apps
Planning
Modeling
Planning
Budgeting
Financial Management
Consolidation
Scorecards
Sales & Marketing
Service & Contact Center
Human Resources
Order Management & Fulfillment
Supply Chain
Verticals
Common Enterprise Information Model
Oracle Business Intelligence Foundation
OLTP & ODS Systems
Data Warehouse Data Mart
Essbase
SAP, Oracle PeopleSoft, Siebel, Custom Apps
Files Excel XML
Business Process

Cuadro de Mando

Se entiende por cuadro de mando (o dashboard) al sistema que informa de la evolución de los parámetros fundamentales de negocio de una organización o de un área del mismo.

Un cuadro de mando está formado principalmente por diversos elementos combinados:

- Tabla: tiene forma de matriz y permite presentar una gran cantidad de información. La tabla puede ser estática, dinámica, o incluso un análisis
- OLAP. Se persigue con este elemento presentar información de forma estructurada al usuario final.
- Métricas: valores que recogen el proceso de una actividad o los resultados de la misma. Estas medidas proceden del resultado de la actividad de negocio. Como ya sabemos, existen diferentes tipos de métricas. En un cuadro de mando, se suelen usar KPI.
- Listas: comúnmente formadas por KPI. En caso de que el cuadro de mando sólo esté formado por este tipo de elemento, se denomina scorecard.

Diseño de cuadros de mando

Tanto los informes como los resultados OLAP son herramientas que proporcionan información a los usuarios finales. La gran cantidad de información que normalmente incluyen estas herramientas puede hacerlas inadecuadas para usuarios que necesiten tomar decisiones de forma rápida a partir de ellas.

El cuadro de mando proviene del concepto francés tableau du bord, y permite mostrar información consolidada a alto nivel. Se enfoca en:

- Presentar una cantidad reducida de aspectos de negocio.
- Uso mayoritario de elementos gráficos.
- Inclusión de elementos interactivos para potenciar el análisis en profundidad y la comprensión de la información consultada.

Los cuadros de mando son una herramienta muy popular dado que permiten entender muy rápidamente la situación de negocio y son muy atractivos visualmente.

Elementos de un cuadro de mando

- **Gráficos**: persigue el objetivo de mostrar información con un alto impacto visual que sirva para obtener información agregada o sumarizada con mucha más rapidez que a través de tablas. El gráfico puede estar formado por la superposición de diferentes tipos de visualización
- **Mapas**: este elemento permite mostrar información geolocalizada. No toda la información es susceptible de estar en este tipo de formato. Se combina con otros elementos para presentar el detalle de la información.
- **Alertas visuales y automáticas**: consisten en alertas que i nforman del cambio de estado de información. Pueden estar formadas por elementos gráficos como fechas o colores resultados, y deben estar automatizadas en función de reglas de negocio encapsuladas en el cuadro de mando.
- **Menús de navegación**: facilitan al usuario final realizar operaciones con los elementos del cuadro de mando.

Ventajas

- La implantación de un cuadro de mando integral proporciona los siguientes beneficios:
  - Define y clarifica la estrategia.
  - Suministra una imagen del futuro mostrando el camino que conduce a él.
  - Comunica la estrategia a toda la organización.
  - Permite alinear los objetivos personales con los departamentales.
  - Facilita la vinculación entre el corto y el largo plazo.
  - Permite formular con claridad y sencillez las variables más imp ortantes objeto de control.
  - Constituye un instrumento de gestión.
  - Facilita el consenso en toda la empresa gracias a su capacidad de explicitar un modelo de negocio y traducirlo en indicadores.
  - Se puede utilizar para comunicar los planes de la empresa, aunar los esfuerzos en una sola dirección y evitar la dispersión.
  - Permite detectar de forma automática desviaciones en el plan estratégico u operativo, e incluso indagar en los datos operativos de la compañía hasta descubrir la causa original que dio lugar a esas desviaciones.