SISTEMAS MULTIDIMENSIONALES

Tema 1

1. **Roll-Up:** Ocultamos detalle (se agrupan los datos). Operación de “muchos a uno” .(Ejemplo: pasamos de 365 días a 1 año)
2. **Drill-Down:** Mostramos más detalle (se expanden los datos). Operación de “uno a muchos”. (Ejemplo: pasamos de 1 año a 365 días)
3. **Slice&Dice:** Tenemos el mismo nivel de detalle, con la diferencia de que sólamente se seleccionan unos datos específicos. (Ejemplo: Los días de verano de todo el año)

Tema 2

ETL: Extraer, transformar y cargar
Llevar los datos de una fuente a un destino (modelo lógico del cubo)

Carga inicial: obtener todos los datos iniciales y meterlos en el cubo.

¿De dónde cogemos los datos?
Del sitio más cercano a la fuente que los produce”

Diferido: perdemos la información intermedia
Inmediato: la información intermedia sí se registra, es una ventaja.
Se genera cualquier mov. Que haya, reflejamos cualquier modificación
Lo ideal es que cada aplicación que genera la base de datos guarde los movimientos.

Cuando construímos la aplicación y queremos generar un cubo, si modificamos la base de datos , meter los cambios que hagamos aparte.

Diapositiva 18/36
¿Son métodos inmediatos o diferidos?

-Comparación de imágenes: Diferido porque lo que pasa entre una foto y otra se pierde (no tenemos los mov. Entre medio)
-Generados por las aplicaciones:
-Mediante disparadores:
-Huella de tiempo: Necesitamos que esté grabado la fecha de la última modificación
Es diferido ya que sólo se guarda la última modificación

¿Cuál de los datos es mejor? DEPENDE
Generados por las aplicaciones es el mejor si nosotros hemos desarrollado las aplicaciones

Resumir en un folio como máximo los métodos de extracción de las modificaciones de los datos. (de los 5 enfoques)
Ventajas, inconvenientes, diferido, inmediato de la diapos 17,18

Transformación de los datos:

Tema 3

Normalizar = paso a tablas
Modelo de datos multidimensional (hechos + dimensiones)
Los hechos tienen mediciones (valor) y en las mediciones hay jerarquías con niveles
MOLAP = Multidimensional OLAP

Modelo de datos conceptual (es el de entidad relación, pero no lo usamos ya que tenemos otro objetivo)
Granularidad = nivel de detalle

Cuando preguntan por el nivel de detalle significa a cuál es el nivel de detalle más bajo en el conjunto de las dimensiones.

Granularidad mes es menor que día, ya que tenemos menor nivel de detalle.
Bases: Niveles que **identifican** a los hechos.
Identificar = que lo representa

Aditividad = tiene la suma sentido sumar al hacer Roll UP?
Sumar el dinero de toda la gente de la clase = chicos + chicas
Dinero que tienes ahora mismo != sumar todo el dinero que has tenido esta semana

Medición: (tienen que ser datos numéricos)

Aditiva: suma toda dimensión
Semi aditiva: suma alguna dimensión
No Aditiva: no suma ninguna dimensión

Transacciones: Cada vez que vendo un producto
Instantanea: La situación de venta
Instantánea acumulada: venta de artículos ..?

DISEÑO LÓGICO:

ROLAP = Relational OLAP
HOLAP = Hibrido MOLAP/ROLAP
Una parte a medida(MOLAP) y la otra en tablas (ROLAP)

NOSOTROS: ROLAP en estrella (llaves generadas, surrogate keys)
En copo de nieve
En estrella va mejor que copo de nieve

Fases
Bases:
Niveles de las dimensiones que identifican los hechos (unívocamente)

Las jerarquías tenían que ser equilibradas, balanceada, completa

Pregunta examen teoría: Diferencias entre sistemas OLTP y sistemas OLAP
Hacer un diseño conceptual de un tema concreto.
(al nivel de detalle más bajo = granularidad más fina)
Pide que no haya disparates… pag 15/53
Conformar dimensiones y mediciones =

Significado de los hechos =(significa) resumen de producto de tienda en un día (pag18,53)
Diseño conceptual 19/53, tener clara las FASES.

¿Cuál es la regla general?
Hechos, tabla de hechos, dimensiones y por cada dimensión una tabla.
Esa es la definición de ROLAP en estrella 24/67
En diseño lógico nosotros usamos ROLAP en estrella.
Copo de nieve: hechos, tabla, nivel , tabla (tema anterior)

Dimensiones degeneradas (distintas situaciones)

1. Retail Sales Facts (hechos)
   Tenemos las dimensionescajero, producto, promoción, método pago, fecha y almacenamiento.

Natural Key = NK (significa que tienen un identificador (ID))
Primary Key = PK
Foreign key = clave externa FK
DD = degenerated dimension (excepción a la regla general)
DD = Dimensión a nivel lógico que no tiene una tabla asociada sino que solamente tiene una llave externa pero sin tabla asociada.
(representadas por una llave externa que no tienen tabla asociada. Son frecuentes a la hora de estudiar los datos a muy bajo nivel)

El nº de tickets, elementos comprados a la vez -> son excepciones.
El ticket permite agrupar cosas que se han vendido juntas

Ejemplos: Expediente (nº expediente), factura, ticket…

Dimensión cajón cambiante:

En hechos tenemos una serie de códigos independientes (flags) que califican a los hechos y que son independientes entre ellos: tipo de pedido, forma de pago, código de envío, código de empaquetado (simple, de regalo..)
Tienen la característica de que todos califican a los hechos.

Problema: tenemos tablas muy sencillas. Aunque se refieran a cosas distintas podríamos pensar en agrupar todo esto en una dimensión (cajón de sastre) para reducir la complejidad.
Proceso: Se une todo en única fila, se comparan los valores con los existentes y si no existe (no se ha dado antes la combinación) lo añadimos con una llave generada

Esto ocurre en situaciones en las que hay exceso de calificativos que pueden responder a “como” y lo metemos todo en “como”. A lo mejor no todos responden a esta dimensión pero por eso la denominamos dimensión cajón de sastre.

Excepción: varias dimensiones a nivel conceptual y se traduce a 1 tabla a nivel lógico y no en varias, es decir, no tenemos muchas tablas sino únicamente 1.

Dimensiones lentamente cambiantes (SCD):

Tenemos una serie de dimensiones y hechos:
-En los hechos tenemos datos que se modifican cada vez que se registre un pedido.
-En las dimensiones tenemos datos de fechas, cliente, vendedor, características..
Hacemos referencia a las SCD a las dimensiones cuyos valores de algunos atributos cambian lentamente (no con mucha frecuencia).

Ejemplo1:
Tenemos una tabla Empleado:
Llega un empleado nuevo -> lo añadimos
Qué ocurre sin el estado civil de dicho empleado pasa de soltero a casado=
-->Hacemos un ALTER y modificamos/actualizamos.

Ahora, la empresa quiere realizar un estudio de ventas solteros vs casados:
Los datos que teníamos se contabilizaban desde hacía años, entonces todas las ventas de dicho vendedor se habían hecho como soltero y ahora, al sobreescribir su estado civil, pasan a ser ventas de casado. Esto no estaría bien ya que “se ha reescrito la historia, es decir, todos los hechos de la historia se han modificado”

Lo ideal sería que las ventas se quedasen asociadas a una persona soltera y asociar las nuevas ventas a una persona casada.

Ejemplo2:
Nos ha pasado en la práctica 2, con el tamaño de los municipios.
Si los municipios cambian “lentamente” de habitantes, su nivel pasa a otro, modificándose el código de municipio. La solución sería duplicar dicho municipio (aplicaríamos solución de tipo2 (nuevo registro), que explicaremos a continuación)

Diferentes métodos para solucionar el **SCD**:

1. **Reescribir los registros (sobreescribir**): Se realiza una modificación o, mejor dicho, un update de la tabla (ALTER). Por ejemplo si tenemos el num tlfn y le cambiamos el número de teléfono, lo sobreescribimos.
   Otras modificaciones válidas pueden ser el cambio de nombre o la fecha de nacimiento.
   Efecto: se reescribe la historia.
   Modificaciones no válidas pueden ser el **Ejemplo1** que hemos puesto anteriormene porque reescribimos la historia.

2. **Crear nuevos registros**: Se realiza este paso2 cuando cambia un dato que a nosotros nos interesa analizar en un futuro, es decir, nos interesa tomar decisiones en función de ese dato y no queremos que se pierda. Luego tenemos los hechos asociados a la versión vieja y a la nueva situación.
   Este paso sí sería una solución al Ejemplo1 y 2 propuestos anteriormente.

   Ejemplo3: Si el granada ficha a messi, queremos 'recordar' todos los goles que había marcado en su anterior club, luego los que meta en el Granada los almacenamos aparte, es decir, preservamos su historia (NO la olvidamos).

   Desventaja: Aumenta el nº de registros de las dimensiones.
   Más problema aún cuando NO es lentamente cambiante (hay muchos cambios de población) --> problema de desdoblamiento de dimensiones

   -Tener versiones de registros: Añadimos un registro pero con versiones (paso 2 con versiones).

   -Tener registros enlazados, nuevo registro vinculado: Si tengo 124, 124 PK es problema porque se repite la llave primaria, luego cojo la llave primaria, añado otro registro con la llave primaria y me olvido de versiones. Lo único que requiere esto es que haya algo que relacione a los dos registros (por ejemplo un identificador DNI). La clave es el uso de llaves generadas ya que nos simplifican todo, ocupan menos, nos permiten la historia si hay cambios en la historia! Por qué? El dni nos sirve para enlazar las instancias.

3. **Tener campos viejos y los actuales**: Suponiendo que nos interesan el resto de campos, añadimos un nuevo campo. La diferencia con el anterior es que los hechos se asocian a unas 'relaciones' distintos de la dimensión.
   2 campos, actual y anterior, no se tienen más de 2 campos

Cerveza alhambra: pasa a bebidas alcohólicas y pasa a dieta mediterranea.
Este tipo se usa en situaciones muy especiales, su uso no es muy frecuente.

Cómo documentarlo?
En el tipo 1 y tipo 2 se pone un 1, 2 a la derecha
En el tipo 3
TIPO 3: current 1, historic 2

Ejemplo: 45/67 tenemos un cliente con current y previous region
Qué pasa si se cambia de nuevo? Nos permite hacer análisis con valores viejos y los nuevos (un análisis dual). Ver las ventas en las dos regiones

¿Qué pasa si hay más cambios?

Tipo 6= 3+2+1
Combinación de las anteriores

Partimos de un tipo 3.
El primer campo (current) lo sobreescribimos siempre, para reinterpretar todo según lo actual)
Tipo2 : se mantiene como se ha generado cada dato.
Esto no es de uso tan frecuente. El uso más frecuente es tipo 1 y 2

Desdoblamiento de Dimensiones: cosos que cambian mucho:
Ejemplo: nº hijos, nivel de estudios, edad
Pasa a ser cambiante y cambiamos la regla.
Según su estabilidad 1 dimensión no se corresponde con 1 tabla

Dimensión de mediciones:
Pedido que se pide en un día y se envía en otro día, tenemos 2 fechas distintas y queremos almacenar ambas. La mitad de los hechos están vacíos.

Solución: 2 focos de atención

Solución 2: considerar el tipo de hecho, es decir, una dimensión de mediciones (pedido, enviado)

Diseño físico:
La partición de la tabla de hechos es una función del SGBD
El SGBD nos permite definir particiones de las tablas.

Esto nos permite adaptar el diseño lógico en función del volumen de datos
Una partición consiste en tratar una tabla lógica como varias tablas físicas y tratarlas de forma independiente.

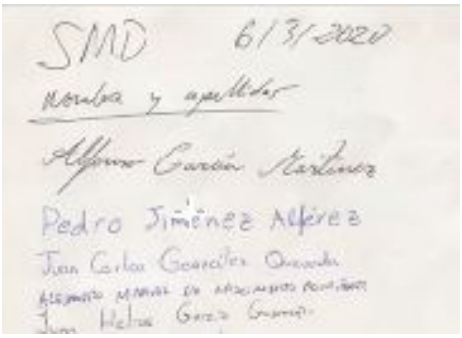
-Esto es muy bueno de cara a las copias de seguridad, podemos realizar esto junto a raid y así mejorar la distribución en disco.

La otra posibilidad es indexar las tablas:
Índice de mapa de bits: asociamos una columna de la tabla en un campo
Se suele hacer la operación AND/OR entre tiras de bits (Los que se llaman maría y tienen 20 años)

Los mapas de bits se pueden usar para materializar un join

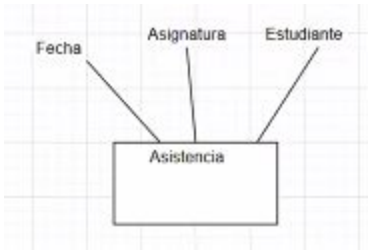
Examen: Construir índice mapa de bits (de nombres y edades se construye la gris)

Diseño conceptual. Tenemos datos de entrada

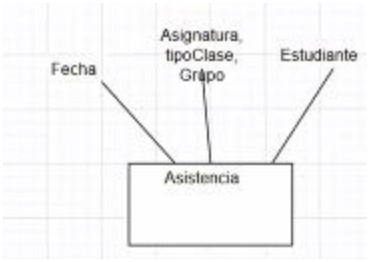


Seleccionamos el proceso de negocio a modelar

Determinamos el significado de los hechos

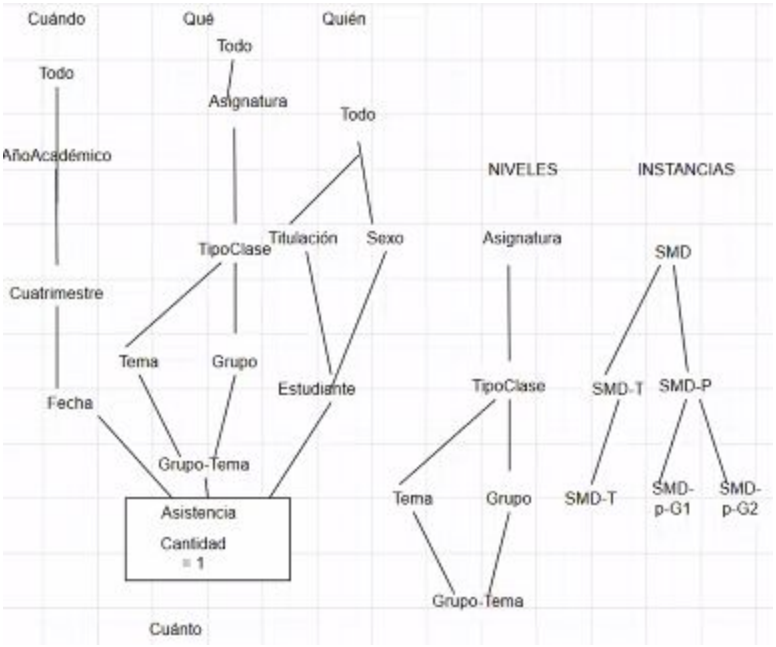


Queremos más granularidad



Significado de los hechos: asistencia a clase de un estudiante en una fecha a una asignatura, un tipo de clase de esa asignatura y un grupo.

Lo siguiente sería **diseñar las dimensiones** (niveles y jerarquías)



Bases: Conj de dim que ident univocamente a los hechos

Errores frecuentes: pensar en teoría/prácticas ó grupo1/grupo2. , es decir, es un error poner instancias