MODELOS DE UNIFICACIÓN DE DATOS PARA LA TOMA DE DECISIONES.

Una vez que conocemos las diferencias entre los ambientes transaccionales, y aquellos usados para la toma de decisiones, es importante mencionar varias consideraciones para el diseño y la implementación de los repositorios de dicha información.

Modelos multidimensionales.

A diferencia de los ambientes transaccionales, donde por la naturaleza puntual de las consultas, la base de datos puede, y debe, estar normalizada, al menos hasta la 3FN, no es así en este tipo de repositorios, donde las operaciones de agregación (suma, promedio, cuenta, máximos, mínimos, agrupación, etc..) y de join están penalizadas.

El éxito de la implementación de estos repositorios depende de varios factores, entre los cuales se incluyen:

- Tener una versión única y oficial de la verdad.
- Contener la información mínima necesaria para la explotación de la misma, el exceso de datos puede ser tan perjudicial como la carencia de los mismos.*
- Evitar, en la medida de lo posible, los joins innecesarios entre objetos.
- Evitar hacer agregaciones innecesarias.

* Existen consideraciones adicionales para repositorios que involucran datos no tradicionales como BigData.

Para esto es necesario **entender la naturaleza de las búsquedas** que se harán en el repositorio, y que dependerán directamente de las necesidades del negocio.

Para ello normalmente se presentará la data en una sola tabla, con aquellos datos a explotar y contando sólo con una llave primaria, de manera tal que no se hagan joins entre otras tablas.

Así mismo, dado que en este tipo de procesos lo que importa es la tendencia general, y no el comportamiento individual de cada uno de los movimientos del ambiente transaccional, esta tabla única deberá contará con aquellos agregados que la toma de decisiones requiera, agrupada por las combinaciones de los criterios usados por los responsables de la toma de decisiones.

A esta tabla central consolidad le llamaremos **tabla de hechos** y a todos los criterios por los cuales podremos hacer las consultas le llamaremos **dimensiones.**

Es posible que una sola tabla de hechos sea demasiado grande para todos los requerimientos de la compañía, cuándo éste sea el caso, es posible que se requiera la creación de más de una tabla de hechos, con diferentes dimensiones, pero es preferible mantener el número de tablas

de hechos al mínimo, a menos que haya una razón de peso para no hacerlo.

La representación de esta tabla de hechos y sus dimensiones, en un diagrama entidad-relación, es parecida a una estrella, aunque las relación no es necesariamente la de una llave foránea, sino una tabla que podrá ser accedida por la herramienta de análisis o reporteo para simplificar su acceso, así mismo no es raro que dentro de una dimensión pueda estar compuesta por otras subdimensiones (ej.: Departamentos, subdepartamentos), por lo que estos diagramas llegan a semejar, burdamente, un copo de nieve, por ello a estos diagramas o estructuras se les conoce también como **"Diagrama de estrella"** o **"Diagrama de copo de nieve"**.

La implementación de estas tablas de hechos es particularmente interesante y algo truculenta. Ya que, como mencionamos en capítulos anteriores, las recomendaciones y consideraciones de diseño para un ambiente OLTP, pueden ser fatales para un ambiente OLAP y viceversa (ej.: Un índice que puede ser la solución óptima para una búsqueda OLTP, puede ser fatal para una búsqueda OLAP al cargar demasiados datos en los buffers, además de cargarlos dos veces: La primera en las páginas de datos y la segunda en la página de índices).

Cada manejador de base de datos tiene diferentes herramientas y funcionalidades para implementar estas tablas de hechos: Por ejemplo: Fragmentación, tablas clusterizadas, tablas columnares, etc.. Pero todas tienen en común que los datos son agrupados o clusterizados en función de las dimensiones de la tabla de hechos.

A la implementación de esta clusterización le llamamos "cubos" y, al igual que la tabla de hechos, deben mantenerse al mínimo, aunque es posible que una sola tabla de hechos tenga presente varios cubos, siendo la razón principal para ello:

- Que la estrategia de clusterización de un cubo sea benéfico para un tipo de búsqueda, pero no para otra.
- Que el tamaño de la tabla de hechos haga que el poblado y actualización de un cubo sea extremadamente tardado (no es raro que el tiempo de creación inicial y poblado de los cubos se incremente de manera geométrica con respecto al número de dimensiones).

En datamarts, así como en datawarehouses pequeños, no es raro que exista una sola tabla de hechos, y un solo cubo para dicha tabla de hechos, pero hay que recordar que el número de tabla de hechos, así como el número de cubos implementados, dependerá de la naturaleza y necesidades del negocio.

Los cubos pueden implementarse de dos formas:

- Inicial
- Incremental

En el primer escenario, el cubo es creado a partir de la tabla de hechos desde cero, y forma parte de los procesos por lotes agendados para la implementación de estos repositorios, si bien el tiempo empleado en la creación de los cubos en cuestión, pudiera considerarse como tiempo "adicional", en la gran mayoría de los casos es tiempo "invertido", ya que el tiempo ahorrado acumulado en la consulta a dichos cubos, es mucho mayor que el tiempo invertido en la población del cubo.

En el segundo escenario, si bien es necesaria una población inicial del cubo, no es factible o conveniente crear éste desde cero cada vez, sino que, periódicamente, se realizan actualizaciones masivas a dicho cubo, a través de dos operaciones:

- UP-SERTs (mezcla de Updates e Inserts), en la que un dato que ya existe es actualizado, y uno que no existe es insertado.
- Depuración. Dado que hacer DELETE's masivos sobre un cubo es normalmente poco aconsejable, y dado que la dimensión "tiempo" es la más común, y prácticamente omnipresente, la práctica de depuración más común es el "detach" de la parte del cluster cuyo periodo en el tiempo ya ha dejado de ser vigente, y borrar el objeto detachado como un todo. (DROP en lugar de DELETE).

Si bien la creación de cubos y validación de cubos se puede hacer manualmente, existen diferentes herramientas para la implementación y administración de cubos (ej. IBM Infosphere Warehouse). Estas herramientas pueden explotar las funcionalidades básicas de ciertos motores de base de datos para el poblado y validación de cubos.

Modelo UDM (*Unified Dimensional Model*)

Además del uso de cubos, hay otras aproximaciones a la implementación y explotación de tablas de hechos, una de ellas es el modelo UDM (Unified Dimensional Model), que permite acceder datos directamente de las fuentes, que busca explotar las ventajas de un ambiente OLTP y del ambiente OLTP.

Si bien este modelo permite el poblando un ambiente OLAP multidimensional directamente desde los ambientes OLTP, este sólo puede ser explotado desde un conjunto de soluciones como Microsoft SQL Server Analysis Services (MS SSAS), usando un lenguaje conocido como MDX (Multidimensional Expressions), parecido a SQL, pero orientado a ambientes OLAP.

Conceptos vistos:
Modelo multidimensional.
Cubos OLAP
Dimensiones
Tabla de hechos.
UPSERT
Diagrama de estrella, o de copo de nieve.
Modelo UDM (Unified Dimensional Model)