SISTEMAS DE BASES DE DATOS – INGENIERÍA INFORMÁTICA

INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS

En este módulo analizaremos los sistemas que manejan datos masivos.
Este tipo de sistema tiene dos características que lo diferencian del resto. Una es que la mayoría de los recursos utilizados están relacionados con la lectura y grabación de los datos con un procesamiento relativamente simple de los mismos, mientras que en los otros su actividad principal está centrada en el procesamiento más complejo de unos pocos datos. La segunda diferencia está asociada al hecho de que la mayoría de estos sistemas está relacionado con la gestión de las organizaciones, y éstas se modifican permanentemente adecuándose a los cambios legales, comerciales, etc. o para hacerlas más eficientes, lo que implica que los sistemas tengan un alto costo de mantenimiento.
Por lo tanto se debe seleccionar un sistema de gestión de datos que facilite la programación, disminuya los costos de mantenimiento y haga más eficiente las operaciones de entrada-salida de datos mejorando así el rendimiento, la performance y el tiempo de respuesta de los sistemas.
En ese sentido, se ha ido evolucionando desde los sistemas de archivos tradicionales hacia los sistemas de gestión de bases de datos.
Veremos también que existen diferentes modelos conceptuales de bases de datos y que cada uno de ellos apunta a resolver cierto tipo de problemas.
El objetivo de este módulo será entonces, estudiar los sistemas de gestión de datos, desde los sistemas de archivos hasta los sistemas de bases de datos, su evolución, características, ventajas y desventajas, de tal manera que le permitan a Ud. adquirir el conocimiento necesario para determinar cual es el tipo de aplicación donde su utilización es la más adecuada.

Queremos aquí desarrollar una introducción a esta problemática, muy poco mencionada en la bibliografía actual, con el fin de que Ud. pueda acceder al material de estudio de la asignatura, sabiendo por qué estudia los sistemas de bases de datos.

Sistemas de archivos vs. Sistemas de Bases de Datos

La pregunta es ¿por dónde comenzar? ¿cómo comenzar el análisis de un problema con el fin de elegir la mejor tecnología para obtener la solución esperada?

Las decisiones se deben tomar con la mayor y mejor información posible. ¿Cuáles son entonces, las preguntas que debiéramos hacernos ante este tipo de situaciones?

Veamos algunas de ellas.
¿De qué forma y con que tecnología puedo satisfacer un requerimiento o dar una solución?

¿Cuál es la razón para utilizar un paradigma, metodología o tecnología? En este caso, ¿Por qué utilizar bases de datos?
¿Cuáles son las limitaciones de esta tecnología? ¿Para qué casos se aplica?
¿Existen soluciones en la actualidad para estos problemas o limitaciones? ¿Cuáles son?
¿Cuáles son las soluciones futuras a estos problemas?¿Hacia dónde apuntan las nuevas investigaciones?

No se pueden responder estas preguntas solo conociendo el presente y adivinando el futuro.
Debemos acostumbrarnos a aprender del pasado, a entender la evolución, realizando un análisis desde los primeros problemas y soluciones. De otra manera vamos a tener que creer en lo que nos cuentan.

Conocer la historia, nos permite ubicarnos en el contexto actual de la evolución, saber por qué se ha llegado a esto, que cosas no debemos repetir para no volver a equivocarnos, y cuales soluciones pasadas son aún vigentes aunque más no sean en solo algunos casos.

Saber que los requerimientos evolucionan a medida que evoluciona la tecnología que le da soluciones a requerimientos previos nos permite pensar más allá de la solución puntual al problema de hoy.
La evolución es un ciclo donde la tecnología permite satisfacer requerimientos de hoy, esto permite a la gente pensar en nuevas aplicaciones que, a su vez, se transformarán en los requerimientos del futuro.

En resumen, para poder aprovechar al máximo los nuevos modelos conceptuales, metodologías y tecnología, es necesario conocer la historia, es necesario evaluar los problemas iniciales planteados, su resolución en cada caso, sus falencias, la evolución para su corrección y, fundamentalmente no perder de vista el horizonte hacia el cual debemos apuntar, es decir el objetivo buscado.
Además, para aplicar los nuevos modelos, se debe tener el dominio de los conceptos en los cuales se basan, lo cual permite decidir donde son aplicables con ventajas y donde se debe elegir otro.

Para introducirnos a la temática específica de las bases de datos, primero tenemos que analizar de donde surge la necesidad de las mismas, y en este análisis, nos debemos remontar a los problemas donde éstas son aplicables.

Historia de la computación (realmente era computación)

REQUERIMIENTO	TECNOLOGÍA
- Cálculo	- Calculadora

- Iteración	- Memoria - Calculadora programable (*)
- Persistencia - Respaldo	- Papel - Batería - Medios magnéticos (discos, cintas)

(*) Primera vez que se plantea el problema de la persistencia -> ¿dónde almacenar el programa para no tener que volver a cargarlo?
(*) Primera vez que se plantea el problema del respaldo o backup → ¿si se pierde el programa como lo recupero?
Hoy las supercomputadoras, mainframes, mini-computadoras, PCs, laptops, palmtops, etc. han reemplazado a las calculadoras programables. Las aplicaciones en las cuales se utilizan las computadoras son variadas:

- Robótica
- Científicas
- Técnicas
- Diseño
- Búsqueda y manejo de información (multimedia)
- Oficina
- Gestión
- Control
- Decisiones
- Juegos
- Etc.

Desde el punto de vista de los datos las podemos dividir en dos grupos:

Grupo 1:

- Robótica
- Científicas
- Técnicas
- Diseño
- Juegos

Grupo 2:

- Búsqueda y manejo de información (multimedia)
- Oficina
- Gestión
- Control
- Decisiones

¿Qué características o diferencias encontramos en estos grupos?

- La característica del grupo 1 es en general la complejidad del procesamiento de la información (tanto para procesos de cálculo como para la presentación gráfica)
- La característica del grupo 2 es en general la complejidad y cantidad de datos que se requieren para las soluciones, lo que implica requerimientos importantes de manejo de:

    - persistencia
    - respaldo

    Ambos requerimientos implican la necesidad de archivos.

Dejemos de lado las aplicaciones del grupo 1, donde la necesidad de archivos o de bases de datos que proporcionan la solución tecnológica a los requerimientos de persistencia y respaldo no es relevante, y concentrémonos en las aplicaciones del grupo 2.

Decimos que la solución a los requerimientos de las aplicaciones del grupo 2 son los archivos, pero, ¿qué tipo de archivos?

Archivos:

Podemos agrupar los archivos en dos tipos:

- No estructurados: no tiene una estructura interna determinada, sino libre. Por ejemplo: un archivo Word.

- Estructurados: se componen de registros y cada uno de ellos se compone de campos, donde se almacenan datos individuales. Por ejemplo: un archivo de empleados, contendrá un registro por cada empleado y cada registro un campo para almacenar el valor de cada atributo del empleado.

Fijemos la atención en los archivos estructurados, ya que éstos son los que nos interesan. ¿Qué otras características podemos encontrar en estos archivos?

Registros de:

- Longitud fija: Todos los registros del archivo tienen una longitud determinada

- Longitud variable: Los registros de un archivo pueden tener longitudes diferentes.

Esto puede responder a una o varias de las siguientes razones:

- Contenido: El contenido de los campos no es el mismo en cada registro. Por ejemplo, un apellido puede ser más corto que otro. En un archivo de longitud fija el campo tiene una longitud fija y si el valor almacenado tiene una longitud menor, se rellena el campo con espacios en blanco. En un archivo de longitud variable solo se almacena el dato relevante.
- Compresión: Los valores almacenados se pueden comprimir con diversas técnicas y la longitud final del registro va a depender de su contenido.
- Repetición: Parte de la información almacenada puede contener varias ocurrencias. Por ejemplo: una factura puede tener uno o más artículos y cada una podrá tener una cantidad diferente de los mismos.

Los registros de longitud variable en general hacen un mejor uso del almacenamiento (utilizan menos almacenamiento), pero requieren trabajo extra para su reacomodamiento en ciertas ocasiones. Por ejemplo, en el caso de la factura, si se agrega un artículo, el registro se agranda y los que lo siguen deberán desplazarse.

PROBLEMÁTICA DEL TRATAMIENTO DE DATOS MASIVOS – SISTEMAS DE ARCHIVOS:

Ahora bien, hemos solucionado el problema de la persistencia y del respaldo de la información, a través de los sistemas de archivos, pero esta solución no es completa. Podemos almacenar los datos y recuperarlos para su procesamiento o visualización, pero el tiempo requerido para ello (tiempo de acceso) es mayor a lo esperado, sobre todo en el caso de datos masivos. Por lo tanto, habrá que hacer esto de la manera más eficiente posible, mientras la evolución tecnológica nos da medios más rápidos.

Medios de almacenamiento

- Los medios de almacenamiento habituales son magnéticos, y en particular, discos magnéticos.
- Los datos están almacenados físicamente en páginas de un tamaño fijo determinado.
- Cada lectura física “levanta” una página.
- Cada escritura física “vuelca” una página.

¿Cómo se pueden bajar los tiempos de acceso?

- Cuanto menos páginas necesitemos leer o grabar mejores tiempos de respuesta tendremos.
- Si las páginas accedidas en un orden temporal lógico están almacenadas físicamente en el mismo orden, de tal manera que se requiera menos movimientos mecánicos de la unidad de almacenamiento, también tendremos mejores tiempos de acceso.

De estas aseveraciones se deduce que los archivos físicos deben ser del menor tamaño posible y que los datos debieran estar físicamente organizados tal como lógicamente se acceden. Pero esto puede ser verdad para operaciones de lectura realizadas siempre de la misma manera (en el mismo orden lógico), ya que un archivo puede estar ordenado físicamente de una única manera.

Veamos algunos ejemplos de consulta de datos:

- Archivos comprimidos vs. archivos no comprimidos. Un archivo comprimido ocupará menor almacenamiento (menos páginas), pero requerirá mayor tiempo de procesamiento ya que sus registros se deberán comprimir antes de grabarlos y descomprimir antes de leerlos (aunque el tiempo de descompresión seguramente será mucho menor que el ganado por el menor tamaño del archivo).
  También requerirá reacomodamientos de sus registros, en el caso de crecimiento en el tamaño de sus datos. Esto requerirá un mayor tiempo para las operaciones de actualización y sí puede cambiar la decisión, en el caso que las actualizaciones sean más frecuentes que las consultas.

- Páginas llenas vs. páginas con un porcentaje libre. Los archivos que se graban completando las páginas con registros ocupan menos, pero necesita un reacomodamiento durante la grabación, si alguno de los registros crece. Incluso algún registro puede tener que "caerse" de la página y moverse a una nueva página (página de overflow). Las páginas con porcentaje libre, tienen espacio disponible para el crecimiento de los registros o para inserciones de nuevos registros y en general, no requieren o requieren menos reacomodamientos, pero ocupan mayor espacio de almacenamiento. El porcentaje de llenado de página (fill factor) es un criterio de diseño.

- Archivos con redundancia vs. archivos que no contienen redundancia. Los archivos con redundancia ocupan más espacio pero los registros leídos traen toda o gran parte de la información necesaria en una única lectura, mientras los no redundantes, donde la información está distribuida en más de un archivo, requieren más lecturas (desde varios archivos), aunque éstos son más pequeños.

En el caso de requerirse solo ciertos datos, que están almacenados en solo uno de los archivos, al ser más pequeño que el único archivo redundante, su uso sería más eficiente.
El archivo no normalizado, al tener redundancias, tiene un peligro adicional. Su información puede ser inconsistente. Para asegurar la consistencia se requiere un proceso extra para evaluarla o compensarla.

- Archivos con índices vs. archivos sin índices. Los índices se utilizan para mejorar los tiempos de respuesta de las consultas, pero se debe tener en cuenta que aumentan los costos de las operaciones de actualización. La decisión pasa por varios factores, entre ellos: tamaño del archivo, tamaño del índice, tipo de índice, frecuencia de actualizaciones vs. frecuencia de consultas, etc.

Ejemplos de tratamiento de archivos:

Se tiene información acerca de ventas. La información a almacenar es la siguiente:

- Facturas (nro_factura, fecha_factura, cliente, {artículo, cantidad, precio_unitario})

Los datos cerrados entre {}, se repiten una o más veces en cada factura.

Las alternativas de almacenamiento podrían ser:

a. Un único archivo de longitud variable:

   Facturas (nro_factura, fecha_factura, cliente, cant_articulos, {artículo, cantidad, precio_unitario})

   cant_articulos indica la cantidad de veces que se repite {artículo, cantidad, precio_unitario} en el registro.

b. Dos archivos de longitud fija o variable:

   Facturas (nro_factura, fecha_factura, cliente)
   Artículos (nro_factura, artículo, cantidad, precio_unitario)

   El nro. de factura, se registra en ambos archivos para relacionar los artículos con su correspondiente factura.

Veamos diferentes requerimientos y cual alternativa es la más conveniente en cada caso.

i. Consulta: Nro. de factura y fecha de factura de todas las facturas del cliente 'JUAN CÁCERES'.

   Para esta consulta se debe recorrer el archivo completo leyendo todas sus páginas y seleccionando de ellas los registros que cumplen con la condición. En la alternativa b, solo se debe acceder al archivo facturas, que ocupa una menor cantidad de páginas que el archivo de la alternativa a, y por lo tanto, el tiempo de acceso será menor.

   Una mejora adicional se podría lograr con el agregado de un índice aplicado al archivo facturas de la segunda alternativa (o de la primera). Este índice se construirá sobre el campo cliente y permitirá que el recorrido del archivo no sea completo, sino solo de aquella parte en la cual están almacenadas las facturas del cliente buscado.

ii. Consulta: Nro. de factura, cliente e importe de la factura, de todas las facturas. Donde el importe de la factura se obtiene de la sumatoria del producto (cantidad * precio_unitario) de los artículos de dicha factura.

   Mientras que en la alternativa a, cada registro trae toda la información necesaria para el cálculo del importe, en la alternativa b, se deben acceder a los registros de artículos de la factura almacenados en otro archivo.
   En general este proceso será más lento, ya que además del mayor espacio ocupado por los dos archivos debido a la inclusión del nro. de factura en cada registro de artículo, se suma que los registros de artículos están seguramente separados físicamente de los registros de las facturas y esto requerirá un mayor movimiento mecánico del dispositivo para acceder a ellos aumentando así el tiempo de acceso.

   Otra alternativa para mejorar los tiempos de acceso para este requerimiento sería agregar redundancia, incluyendo en el archivo facturas de la alternativa b un campo con el importe de la factura.
   Esto permitiría acceder solo a este archivo para resolver la consulta y este archivo sería de un tamaño bastante menor al de la alternativa a, con lo cual el tiempo de acceso bajaría de manera importante.
   Como contrapartida, esta alternativa trae tres problemas:

   - Aumento en el espacio de almacenamiento ocupado (aunque no es importante)

- Aumento en los tiempos de actualización de los artículos de la factura, ya que cambios en estos datos implicarán un recálculo del importe de la factura y una regrabación del registro correspondiente a la misma.

- La redundancia agregada posibilita la aparición de inconsistencia. La única forma de asegurar la consistencia, sería que el mismo sistema de archivos ejecutara el proceso de compensación (actualización del importe de la factura) en forma automática.

Podríamos continuar dando ejemplos y alternativas al diseño, pero con éstos ya tenemos una aproximación bastante completa de la problemática a la cual nos enfrentamos. Las preguntas que surgen son las siguientes:

1. ¿Cuál es la estructura de archivos más adecuada?
2. ¿Cuál es el método de acceso más adecuado?
3. ¿La elección permanecerá en el tiempo siendo la más adecuada?

Es probable que podamos encontrar la respuesta a las dos primeras preguntas, aunque esto dependerá de nuestra capacidad en el diseño y la programación.

La tercer pregunta plantea un problema que va más allá de nuestra capacidad.
Es evidente que la solución para los requerimientos y la información almacenada hoy, podrá no ser la misma si mañana algún requerimiento o la información cambia.
No es lo mismo acceder a un archivo con 100 registros que acceder al mismo archivo si éste contiene 1.000.000 de registros.
Tampoco es lo mismo, si se agrega el requerimiento: mostrar las facturas del año 2005.

Pero, ¿cuál es el problema?.

La respuesta es que "los programas son dependientes de la estructura física de los datos y de los métodos de acceso a esas estructuras".
Es decir, si un archivo tiene 100 registros puedo accederlo secuencialmente recorriendo todos sus registros y extrayendo aquellos que necesito procesar. Si ahora, el archivo tiene 1.000.000 de registros, es probable que la creación de un índice mejore el tiempo de acceso, pero tendremos que cambiar el método de acceso.
Esto quiere decir, que permanentemente, se debe estar monitoreando la performance y modificando los programas, con el aumento de costos de mantenimiento y probabilidades de cometer errores que se traducen a su vez en mayor costo de mantenimiento.

En resumen:

Los sistemas de archivos masivos nos permiten dar soluciones a los requerimientos de persistencia y respaldo, pero tienen inconvenientes relacionados con:

- Tiempos de acceso: es difícil lograr un diseño con la performance requerida, balanceada en cuanto a los requerimientos y estable en el tiempo
- Los programas son dependientes de las estructuras de los archivos: cambios en las estructuras físicas (para mejorar la performance) implican cambios en los programas que acceden a esas estructuras
- Es más difícil el control de acceso a los datos: El sistema de archivos solo controla la grabación y lectura de los registros en los archivos y no tiene ninguna inteligencia adicional.
- No se puede asegurar la integridad: No tienen inteligencia adicional.
- Es más difícil asegurar la consistencia: No tienen inteligencia adicional.

SISTEMAS DE BASES DE DATOS:

Los sistemas de bases de datos (DBMSs) nacen para dar solución a estos problemas. Su característica principal es que al software que compone el sistema de archivos se agrega una capa de software que aumenta la inteligencia del sistema.

Esta inteligencia otorga al sistema las siguientes características:

- Servidor de datos: Transforma al servidor de archivos en un servidor de requerimientos. El servidor ya no devuelve simplemente un archivo para que las aplicaciones lo procesen, sino que directamente recibe el requerimiento sobre los datos, lo procesa y devuelve la información solicitada.
- Arquitectura en capas: Los programas ya no acceden a los datos almacenados sino a vistas externas creadas para los diferentes usuarios de los datos. La estructura de los datos puede cambiar internamente (capa interna) pero permanecer estable para los usuarios (capa externa).
- Independencia de datos: Como los programas ya no acceden a los datos almacenados ya no debe programarse la lógica de acceso. El DBMS es responsable de la optimización del requerimiento para lograr la mejor performance. Los programas son inmunes a los cambios en las estructuras de almacenamiento y métodos de acceso.
- Lenguaje de alto nivel: Un lenguaje no procedural está disponible para los usuarios y programas con el fin de permitir expresar los requerimientos de datos sin especificar como obtenerlos. El DBMS, analiza el requerimiento, lo transforma en un lenguaje interno, lo optimiza de acuerdo a las condiciones actuales, lo transforma en lenguaje de bajo nivel, lo procesa y devuelve el resultado.

Con este lenguaje y la arquitectura en capas se logra la independencia física de datos que permite la inmunidad de los programas ante cambios de estructuras de almacenamiento.

- Administrador de la base de datos (DBA): El DBA es responsable del monitoreo y cambio en las estructuras de almacenamiento para el logro de la mejor performance del sistema. El DBMS es el encargado de "aprovechar" estos cambios en forma automática.
- Integridad: La capa de software inteligente, permite agregar reglas de integridad que son evaluadas automáticamente por el DBMS, no permitiendo el registro de un dato si no cumple con las reglas de integridad definidas.
- Consistencia: EL DBMS también dispara los procesos de "propagación automática de actualizaciones" para asegurar la consistencia cuando existe redundancia.

MODELOS DE BASES DE DATOS:

Existen muchos tipos de aplicaciones donde los sistemas de bases de datos son la tecnología más adecuada para dar soluciones a los requerimientos.
Pero estas aplicaciones son muy variadas y las soluciones no serían óptimas si se utilizara un único modelo de bases de datos. Por ello, hoy existen diferentes modelos teóricos que permiten diseñar diferentes soluciones, cada uno de ellos con ventajas y desventajas con respecto a los otros.
De todas maneras, todos estos modelos fueron definidos con mayor o menor profundidad, siempre dentro de un marco teórico, sin tener en cuenta las posibilidades que brinda la tecnología para su implementación real.
Por lo tanto, antes de elegir un modelo, Ud. deberá preguntarse si ¿la tecnología actual permite implementarlo completamente?; y si no es así, ¿cómo se verá limitado dicho modelo y como lo afectarán estas limitaciones?

Los modelos teóricos siempre están adelante de la tecnología, aunque muchas veces, los requerimientos del momento hacen que la tecnología les de soluciones al margen de los modelos. Algunas de esas soluciones o una combinación de ellas, finalmente ampliarán las definiciones del modelo o lo convertirán en un nuevo modelo.
Ahora bien, ¿como afectan las restricciones que impone la tecnología?. Esto depende del modelo; hay modelos más antiguos, cuyas restricciones son menores, y modelos más modernos, para los cuales la tecnología aún no está tan avanzada para garantizar una implementación amplia y confiable.
Por eso, aunque en muchos casos las ventajas de un modelo son obvias con respecto a otro, a veces es probable que se seleccione este último, por su confiabilidad.

Tal es el caso del modelo relacional, que en la mayoría de las situaciones de tratamiento de datos masivos es el más adecuado; y en otros casos –en los que no lo

sería– también es utilizado (ya sea en su versión pura como en su versión extendida) debido a su mayor confiabilidad.

Hoy, este modelo es el más ampliamente utilizado en la actividad profesional y es el que estudiaremos.

Como mencionábamos al comienzo, queríamos brindar esta introducción con el objetivo de que Ud. pueda ubicarse en la problemática sobre la cual trabajaremos en esta materia, y darle una primer base de conocimientos a partir de la cual Ud. pueda acceder de forma más fluida al material de estudio recomendado.