Tema 6. Organización de Ficheros: Organización Base (física)

- **Introducción**:
  - **Que buscamos optimizar:**
    - **Tiempo de respuesta:** Disminuyendo el numero de accesos y evitando reorganizaciones en tiempo de proceso, etc.
    - **Espacio de Almacenamiento**: Incrementar la densidad, minimizar almacenamiento de estructuras auxiliares, etc. Este también afecta al tiempo de respuesta.
    - **Coste de desarrollo y mantenimiento**.
  - **Partimos de ciertos requisitos:**
    - **Características del dispositivo**: Bloque(tamaño), tiempo acceso(secuencialidad), etc. Son las características físicas.
    - **Características de los archivos**: Tamaño registro, cardinalidad, volatilidad, etc.
    - **Características de los procesos:** Tipología, frecuencia, criticidad (aquellos que tengo en cuenta para optimizar, los que marcan la diferencia)
- **Organizaciones consecutivas:**
  - **Organización básica:** Organización serial.
    - Registro físicos en serie, están uno detrás del otro y se acceden en ese orden. La inserción se hace siempre al final, sin ningún criterio de colocación.
      Consecutivos: n bloques. No consecutivos: N cubos.
    - El ultimo cubo es privilegiados al ser el mas frecuente, ya que es el que se accede mas, y lo mantengo en memoria intermedia, de esta manera es mas rápido el acceso y nos ahorra tener que buscar el ultimo a la hora de insertar.

$d_{ocup} = 100\%$

    - **Características**:
      - Aprovechamiento de espacio y Coste de accesos a la totalidad, ÓPTIMO.
        Ya que están amontonados, por lo que están cercanos.
      - No existen claves privilegiadas, no se puede filtrar.
      - Tamaño area de busqueda: n bloques (consecutivos) y N cubos (no-consecutivo)
    - **Procesos de organización serial:**
      - **Actualización**: Implica escritura.
        - **Inserción**: Se añaden los registros al final del fichero.
          - Coste 1 acceso normalmente, si es un registro reciclado implica 2 accesos.
        - **Borrado**: Eliminar un registro. Coste para todo tipo: selección + k accesos
          - **Borrado físico:** Se da en la organización no consecutiva, se borra el registro y se recolocan el resto de registros del cubo.
          - **Borrado lógico:** Se da en la organización consecutiva, se hace una marca en el hueco del registro que se quiere eliminar indicando que tamaño tiene para que en la lectura se pueda saltar. En realidad no se eliminar el contenido, solo se hace la marca. Este borrado no desplaza los registros tras el borrado, seria muy costoso.
        - **Modificación**: Casi siempre serán para ampliar el tamaño.
          - **Registros fijos:** Es la que se produce en registros que tienen siempre el mismo tamaño, por lo tanto se hace en ese hueco la modificación.
          - **Serial no consecutiva:** Dependerá de si cabe o no, si hay ELD
            - Si hay hueco, se modifica en el mismo cubo, el ELD lo permite.
            - Si no hay hueco, se busca otro cubo en el que quepa, muy probablemente sea el ultimo.
          - **Coste en registros fijo o en no consecutivos:** selección + k accesos.
          - **Otros casos:** Se borra el antiguo y se reinserta modificado.
            - Coste: Selección + k + k

- **Recuperación**: Es la lectura de registros.
  - **Consulta selectiva identificativa:** Leemos hasta encontrarlo, solo hay 1, por lo que una vez encontrado termina.
    - Coste: (N+1)/2 en no consecutiva y (n+1)/2 en consecutiva.
    - N numero de cubos y n numero de bloques.
  - **Consulta selectiva no identificativa:** Leemos todos los registros, ya que no sabemos cuantos habrá que cumplan esa clave.
  - **Consultar selectiva multiclave:** Leemos todos los registros, ya que no sabemos cuantos habrá que cumplan esa clave.
  - **Consulta a la totalidad:** Leemos todos los registros, ya que no sabemos cuantos habrá que cumplan esa clave.
  - **Coste en el resto:** N en no consecutiva y n en consecutiva.
- **Mantenimiento Serial:** Gestión de huecos.
  - **Espacio Libre Distribuido:** Se mantiene un espacio libre, para que si hay que realizar una modificación esta quepa en el cubo, y no halla que buscar otro. Ya que es muy costosa la modificación si cambia de ubicación, hay que actualizar todo.
    - Mover registros es engorroso y genera costes adicionales.
    - En organizaciones consecutivas, se puede evitar dejando espacio libre distribuido, que es un porcentaje de espacio del cubo reservado para modificaciones.
    - En Oracle se reserva un 10 %.
  - **Gestión de Huecos o Pila de inserción:** Lista de cubos candidatos para insertar, los mantiene localizados. Son aquellos que tienen la ocupación por debajo del umbral de ocupación, que nos indica que el registro puede ser reciclado. Se reduce el tamaño del fichero, aunque pasara la inserción a costar dos accesos. Esta estructura se realiza en el primer full scan.
  - **Compactación:** Proceso que desplaza registros para eliminar grandes huecos producidos por el borrado o modificación. Refresca los cubos: Si falta hueco a ELD, se añade espacios y si falta se le quita. Se hace forma periódica en las organizaciones consecutivas y en las no consecutivas se hacen en el momento.

- **Organización ordenada:** Organización secuencial.
  - Surge de la organización serial introduciendo un orden.
  - El acceso aleatorio a bloques obliga a contar con un mecanismo para interpretar su contenido, localizar el comienzo del primer registro.
    - A nivel físico, cubos, comienza con un registro completo.
    - A nivel físico-lógico, registros consecutivos, comienza en una marca de inicio/fin.
  - Almacena registros con un criterio de orden.
  - **Características:**
    - El aprovechamiento de espacio y el coste de accesos a la totalidad son ÓPTIMOS.
    - En esta organización si hay una clave privilegiada de orden, que nos permita hacer búsqueda dicotomía.
    - Tamaño área de búsqueda: n bloques (consecutivos) y N cubos (no-consecutivo).
  - **Procesos de organización secuencial:**
    - **Actualización:**
      - **Inserción no ordenada:** Se añaden registros al final del fichero, lo que provoca que la organización degenere y se reduzca la eficiencia.
        - Coste: 1 acceso, se introduce al final.
      - **Inserción ordenada:** Se localiza la ubicación del registro, para introducirlo, si no cabe provoca desbordamiento que requiere gestión de desbordamientos. Este tipo de inserción es mas costoso, pero los mantiene ordenados.
        - Coste: log2(x+1) +1

$d_{ocup} = 100\%$

Ojo inserción consecutiva, puede ir directa al A. desborde.

Ojo, aprovechar la secuencialidad con el $\frac{k}{Tc}$

- **Borrado:** Se mantienen las dos posibilidades, hacer una marca en los consecutivos, y en los no consecutivos eliminar y colocar el resto de registros.
  - Coste: selección + k/Tc accesos. Tc tamaño del cubo. Es encontrarlo y después todos los accesos a cubos para los k registros.
- **Modificación:** Se mantienen los de serial, los no consecutivos lo modifican en el mismo cubo y los consecutivos se borra e introduce ya modificado. La diferencia es que modificar la clave de ordenación hace que tengamos que eliminar el registro y reinsertarlo, para que ocupe la posición correcta.
  - Coste en no consecutivas y no modifica Co: selección + k/Tc accesos.
  - Coste otros casos: coste borrado + coste inserción

Lo inserta al final desordenado, lógue degenera la org. (Area desborda.)

- **Recuperación:** El coste de la sección (selección accesos)
- **Selección:**
  - **Consulta por clave no privilegiada:** Recorre todos.
  - **Consulta por clave privilegiada**
    - **Clave identificativa:** Mediante búsqueda dicotomía.
      - Coste: log2(x+1)
    - **Clave no identificativa:** Mediante búsqueda dicotomía extendida.
      - Coste: coste de la búsqueda dicotomía extendida + coste desbordamiento.
  - **Consulta selectiva multiclave:** Primero filtra, luego hace búsqueda dicotómica.
    - Coste: Primer filtra y luego búsqueda dicotomía.
  - **Consulta a la totalidad ordenada(por clave privilegiada):** Optima.
    - Coste: Coste serial de un full scan de N.

- **Búsqueda dicotómica**: Mira el elemento central del espacio de búsqueda, si coincide termina, pero sino restringe el espacio de búsqueda a la mitad. Y se repite el proceso.
- **Búsqueda dicotómica extendida:** Primero hace búsqueda dicotómica hasta encontrar uno de los elementos, después busca serialmente hacia delante y hacia atrás, hasta que encuentre un fallo por ambos lados.

- **Mantenimiento Secuencial:**
  - **Gestión de desbordamiento:**
    - **En organización consecutiva:** Hay un área de desbordamiento, donde están desordenadas.
    - **En organización no consecutiva**:
      - **Rotación:** Traspasa elementos de un cubo lleno a su vecino, si tiene espacio libre.
      - **Intercalar cubos completamente vacíos**, son virtuales, realmente no están en medio físicamente, pero actúan como tal.
      - **Partición celular:** Al desbordarse, se añade en medio un cubo vacío (también virtual) para que entre el desbordamiento y se mantenga en orden. Y se reparten los registros, entre el nuevo cubo y el desbordado.

- **Espacio Libre Distribuido para inserción:** Se usa en organizaciones secuenciales, para reducir la tasa de desbordamiento. Reserva una parte del cubo para posibles futuras inserciones o modificaciones.
- **Lista de huecos:** Se usa poco esta estructura que localiza los huecos, ya que estos tienen un orden, tamaño...
- **Reorganización o Reordenación:** Compacta y reescribe todos los registros ordenados, mezcla las áreas. Es una operación costosa, pero reduce el tamaño y mejora la eficiencia.

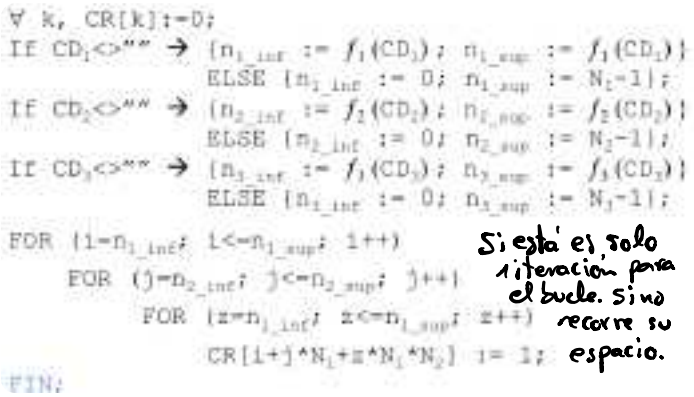
- **Organizaciones Direccionadas (Siempre es no-consecutivo)**
  - **El acceso aleatorio**: proporciona el registro fisico indicado. Conociendo la clave de busqueda puede conocer directamente su posición.
  - Clave privilegiada, la CD (Clave de Direccionamiento), tiene gran capacidad de filtrado maxima.
  - **Aprovechamiento de espacio:** reducido.
  - **Coste de accesos a la totalidad:** elevado.
  - **Espacio de direccionamineto:** N cubos.
  - **Tipos de Direccionamiento:**
    - **Organizacion Direccionada Directa:**
      - Cada registro tiene su dirección reservada y cada cubo un solo registro.
        - Si el tamaño maximo del registro es mucho menor que el del bloque, se pueden considerar **celdas**.
      - Los valores de CD que no ocurren implican cubos vacíos. (baja densidad).
      - Se puede transformar la dirección, si solo ocurre en un rango de valores:
        - **Org. Direccionada Directa Absoluta**: la CD es la dirección del cubo.
        - **Org. Direccionada Directa Relativa:** existe una biyeccion entre CD y la dirección del cubo. Ejem: Truncar la clave.
    - **Organizacion Direccionada Dispersa (HASH):**
      - La CD se transforma en dirección del cubo, pero la funcion no es un biyeccion. Si la distribución es buena, aumenta la densidad.
      - **Algoritmo de Transformación:**
        - **Conversión numérica:** Se pasa la clave a un valor numérico. Ejem: ASCII
        - **Función de dispersion:** Se encarga de repartir los distintos cubos en las posibles direcciones, de 0 a N-1. Busca que estén distribuidos uniformemente, sin que haya cubos vacíos o muchos registros en el mismo cubo.
      - Si la densidad es baja, se puede reorganizar cambiando:
        - La dispersion, el diseño del cubo o el espacio de Direccionamiento.

Al calcular la densidad ocup. no se tienen en cuenta los r' registros desb.

    - **Conceptos:**
      - **Claves sinónimas:** Producen la misma dirección.
      - **Claves homónimas:** Tienen el mismo valor.
      - **Potencia de Direccionamiento:** #valores(CD) >= N
      - **Colisión:** Inserción en un cubo no vacío.
      - **Desbordamiento**: Colisión en un cubo sin espacio suficiente.
  - **Procs. Direccionamiento:**
    - **Actualización:**
      - **Inserción:** Calcula la dirección, y se añade el registro allí. Si no cabe, se desborda. Coste: ② accesos. Siempre
      - **Borrado:** Localiza el registro y se elimina. Coste: selección + k accesos
      - **Modificación**: Localiza el registro y se modifica, si no cabe borra e inserta.
        - No actualiza la CD: selección + k accesos.
        - Actualiza CD: Borrado y reinserción.
    - **Recuperacion:** El coste de la selección.
      - **Localizacion:** Filtra por CD
        - **Consulta selectiva identificativa:** leer cubos no filtrados, hasta encaje.

        - 1 + Prob.desb.*(Ndesb.+1)/2 accesos.
    - **Consulta selectiva no identificativa:** leer todos los cubos no filtrados.
        - 1 + Ndesb. Accesos.
    - **Consulta selectiva multiclave:** Filtrado multiclave.
        - 2^b+Ndesb. Accesos. b, numero de bits de la dir. que desconocemos.
    - **Otras consultas**: full scan= N + Ndesb.
- **Gestion Desbordamiento:**
    - **Dos criterios**
        - **Donde ser ubique el registro desbordado:**
            - **Saturación**: Dentro del espacio de Direccionamiento.
            - **Area de Desbordamiento:** Fuera del area de datos.
        - **Mecanimso de ubicación:**
            - **Direccionamiento abierto:** la dirección nueva es la anterior mas k.
            - **Encadenamiento**: Se marca donde esta el registro desbordado.
            - **Otros (organizacion independiente)**
    - **Saturacion con Direccionamiento Abierto:** La nueva dirección se averigua a partir de la direccion desbordada.
        - **Sondeo lineal:** dir'= dir+1
        - **Rehashing**: dir'= dir+k
        - **Doble hash**: dir'=f(CD)
        - Si esta estuviera ocupada se produce un **choque**. Si ademas no cabe, sera un **rebote**.
        - Si se produce un rebote ser buscara otra dirección nueva hasta encontrar una posición libre o hasta haber recorrido todo el espacio.
        - Se marca con un **Byte de desbordamiento** en el cubo que indica si ha desbordado, que nos permite saber cuando dejar de buscar.
        - En media ser recorren: N/(#cubosConByteDesbordamiento0 +1)
    - **Saturación Progresiva Encadenada:** La nueva dirección se deja apuntada en el cubo desbordado.
        - **Puntero**: Indica la ubicación de otra informacion. Puede ser logico(clave), relativo(dir.) o fisico.
            - **Puntero relativo de precision simple**: Indica en que cubo.
            - **Puntero relativo de precision doble:** En que cubo y en que posición. Es el que usa para este desbordamiento, al ser un registro especifico.
        - Los registros que desbordan en un mismo cubo se van encadenando, el cubo apunta al segundo y el segundo al primero.
    - **Area de Desbordamiento Independiente:** Los registros desbordados son almacenados fuera del area de datos, en un archivo aparte.
        - **Ventajas**: Se eliminan los choques y rebotes.
        - **Desventajas**: Necesita mas espacio, un area auxiliar. La organizacion degenera y baja la eficiencia. Este area no se filtra en busqueda por clave privilegiada. Y si es por clave alternativa, se tienen mas cubos.
        - En el area de desbordamiento normalmente la organizacion es serial, y si desborda el area de desbordamiento se hace otra gestión.
    - **Encadenamiento en Área de Desbordamiento**: Los registros que desbordan se almacenan en un area aparte, y su direccion se deja apuntada en el cubo desbordado.
        - **Encadenamiento a registro**: Los registros en area de desbordamiento se almacenan serialmente pero incorporan un puntero de encadenamiento.
        - **Encadenamiento a cubo (Extensión del cubo de datos)**: Cuando un cubo desborda, se le asigna a esa direccion un cubo completo dentro del area de desbordamiento:
            - El puntero de encadenamiento es de precision simple.
            - El cubo solo contiene registros de la direccion que lo apuntan.

- **Dir. Disperso Multi-Clave**: Consiste en ampliar el algoritmo de transformación, para operar varias CD. Combina las dispersiones de varias CD, cada una con su espacio de direccionamineto y funcion de dispersion.
  - Aunque no conozcamos todas las CD, podemos filtrar solo con conocer uno de esos atributos. El algoritmo de filtrado utiliza bucles anidados.
    - N son la cantidad de direcciones que representa cada CD.
    - Cuantos mas bits se le asigne mas filtra la CD.



- **Mantenimiento Hash**:
  - El encadenamiento a cubo proporciona extensiones automáticas de espacio para cada direccion con poco coste.
  - Por tanto, definir ELD para inserción no es (en general) una ventaja, pero si un N mas grande.
  - Si el area de datos esta demasiado saturada o demasiado vacío, es necesario reorganizar. Aunque la reorganización automática no suele ofrecer buen rendimiento.
- **Cluster de datos:** Agrupación física de registros que tengan el mismo valor para una clave privilegiada (Clave de agrupación).
  - Puede haber registros de distintos tipos, pero con el atributo común.
  - Son una **organizacion no consecutiva** y puede ser:
    - Simple: Serial.
    - Indizado: Serial con indice sobre la clave de agrupación.
    - Disperso: Mejora procesos selectivos, por CD.
    - Ordenado: Mejora procesos selectivos y ordenados.
  - CREATE CLUSTER nombreCluster (nombre TIPO(TaMAÑO));
  - CRÉATE TABLE nombreTabla... CLUSTER nombreCluster(AtribTabla);
  - Mirar diapositivas para las características en Oracle.