* tener cuidado con la cantidad de bloques y de accesos a memoria que consumo según las querys que me dan
* necesito crear los índices para una columna de una tabla
  + ver los tipos de índices que hay y cual es el que más nos compensa
  + ver que criterio es el que nos piden para saber cual hago
* area de memoria fija, - cluster, join de cosas que tienen que ver x
* acelerar los procesos de los join
  + buscar las claves que sirven de union y una vez que lo tengo creo el cluster
  + union de todas las tablas en el disco para acelerar las prácticas de búsqueda

flags → is serveroutput on bla bla bla

1. solución 1
   1. eliminar (vistas, índices, *triggers*, …) + datos de las pruebas
2. solución p2
   1. poner esto (“createBD\_2023.sql” y “insert2023 - errors\_fixed.sql”)
   2. modificar TRACKS añadiendo dos nuevas columnas (*searchk* y *lyrics*)
      1. crea estas columnas y que proporciona los valores a una de ellas debe ejecutarse una vez (tras ejecutar los dos scripts mencionados). Este código se ofrece en el primer anexo de este documento, y está también incorporado al script de la P3
3. analizar las transacciones que se producen en la base de datos (operaciones o procesos más costosos) - no todos con la misma frecuencia

|  | frecuencia relativa |  |  |
| --- | --- | --- | --- |
| Proceso P1: modificar una fila (selección por el atributo *searchk*) | 0.98 |  |  |
| Proceso P2: consulta 1 de la práctica 2. | 0.01 |  |  |
| Proceso P3: consulta 2 de la práctica 2. | 0.01 |  |  |
|  | por cada vez que se ejecuta cada consulta, se ejecutan 1000 operaciones de actualización |  |  |
|  | código en el anexo |  |  |

1. A partir de ese código, se ha creado una carga de trabajo (*workload*) que incluye estas operaciones para ser ejecutadas en la proporción y orden adecuados. Esto es especialmente relevante, ya que los procesos de actualización afectan al rendimiento del resto de procesos. Por ello, en este escenario de la práctica, no se permite alterar el orden de ejecución de las operaciones. Esta metodología de análisis y evaluación de diseños físicos (evaluación de un workload sobre un entorno controlado) no es la única forma de trabajar para este fin, pero es adecuada en esta práctica.
2. (script\_statistics\_2023.sql) —- paquete (PKG\_COSTES)
   1. procedimiento PR\_PREPARE, inicializa la variable que decide qué operaciones de actualización se van a realiza
   2. procedimiento PR\_WORKLOAD, carga de trabajo propiamente dicha
      1. OPCIONAL → añadir directrices de ejecución (*hints*)
   3. procedimiento PR\_RESET, devolver la base de datos a la situación inicial (ya que la ejecución de la carga de trabajo altera el estado de la base de datos)
      1. OPCIONAL → inicializar el estado de alguna estructura de su diseño físico que se pudiera ver afectada por las operaciones de actualización del WL (es decir, que afecte a la tabla *tracks*)
   4. procedimiento auxiliar control del tiempo
   5. procedimiento (RUN\_TEST) , repetir la carga de trabajo una o más veces (indicándolo en el parámetro *ite* del procedimiento).
      1. ejecutar para realizar las mediciones (calcula y muestra el tiempo medio y el número de accesos de las n repeticiones de la carga de trabajo) → SOLO EJECUTAMOS ESTE
   6. OPCIONAL → Modificar el experimento, eliminando la inicialización de las estructuras para estudiar otro escenario.
3. medir el rendimiento de la carga de trabajo estándar para ver la situación inicial de partida
   1. ejecutar el paquete 5.
   2. Se debe tener presente que el número de iteraciones a repetir el workload no debería ser menor a 5 si se quiere tener una estimación mínimamente fiable (aunque se recomiendan entre 10 y 20 iteraciones, no son necesarias más de 10 iter.)
4. analizarla (LO MÁS IMPORTANTE )
   1. el diseño físico de partida
   2. la naturaleza de las operaciones que se van a ejecutar
   3. el plan de ejecución de cada operación (algoritmos utilizados en su resolución), proporcionando detalles de su ejecución
   4. los costes medios calculados sobre varias ejecuciones
5. propuesta de diseño físico
   1. reducir el número de accesos a memoria secundaria, y el tiempo de ejecución global
   2. cambiar parámetros físicos de los objetos (pctfree, tablespace, etc), crear estructuras (clusters monotabla o multitabla, índices, etc), añadir directrices de ejecución, ... No se permite utilizar vistas materializadas. De modo obligatorio, se deberá considerar (al menos) la inclusión de un índice y de un clúster (se incluirán su diseño, implementación y evaluación en la memoria; si bien al analizar su rendimiento se puede concluir que no es adecuado y eliminarlo del diseño final).
6. implementado sobre createBD\_2023.sql
7. ejecutarlo
8. probar insert2023 - errors\_fixed.sql a ver si va bien
9. crear tablas de tracks – ese código puede añadirse al final del script createBD para mayor comodidad
10. En caso de haber alterado el WL, será también necesario volver a crear PKG\_COSTES ejecutando el script.
11. ver y evaluar los efectos de los cambios
    1. comparar datos iniciales con los obtenidos con los cambios

Opcionalmente, se puede ampliar el análisis realizando un nuevo diseño físico y evaluándolo (habitualmente, el proceso de monitorizar y afinar el sistema es un proceso de refinamiento continuo, si bien en esta práctica esto no es preciso, y sólo se propone mejorar la situación de partida). También se podría (opcionalmente) modificar el experimento para evaluar el rendimiento de sucesivas ejecuciones sin que se realice la operación de reinicio (del estado de la BD).

- **Situación de partida**: restablece el estado original de la BD.

- **Mide** el rendimiento de la carga de trabajo estándar (tanto en tiempo como en número de accesos) a través de varias repeticiones.

- **Analiza** el plan de ejecución para cada operación, extrayendo los procesos (la tipología) de los que se compone. Clasifica y ordena todos los procesos en la estructura física, su frecuencia y una aproximación del coste.

- A partir de ese estudio, desarrolla una **propuesta** para reducir el número de accesos a soporte secundario.

- Describe el **diseño físico completo**, comenzando por las organizaciones básicas involucradas en el mismo, así como las estructuras auxiliares utilizadas para mejorar el rendimiento. - **Implementa** el diseño físico completo propuesto, y mide su rendimiento sobre la carga de trabajo estándar propuesta.

- **Compara** el rendimiento del diseño físico original frente al propuesto e interpreta los resultados. Opcionalmente, puedes refinar la propuesta, e implementar mejoras al diseño físico (cuyo rendimiento después analizarás y compararás con los anteriores).

- **Documenta** todo el trabajo realizado en esta práctica, poniendo especial atención al diseño físico, a las medidas de rendimiento, a los planes de ejecución, a las propuestas de mejora, y a la comparación de resultados.