**ISIS2304 – Sistemas Transaccionales**

**Iteración 4**

Daniel Junco – 201213839

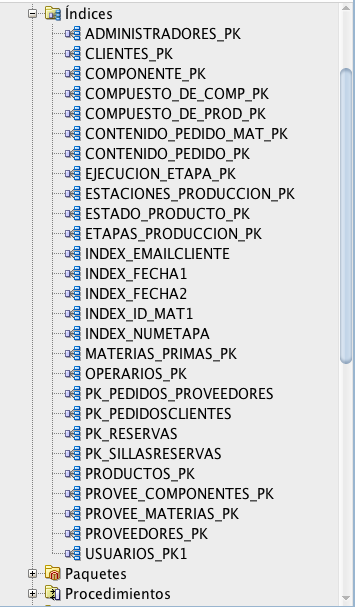
**Análisis. Diseño de la aplicación**

* **Justificación selección índices.**

El índice de una base de datos es una estructura que optimiza la velocidad de las operaciones, a través del uso de identificadores en una tabla, permitiendo así un rápido acceso a los registros de una tabla. El espacio en memoria requerido para almacenar un índice es comúnmente menor que el almacenamiento de la tabla. Se usan índices para aquellos valores sobre los cuales se hacen búsquedas frecuentes.

Así mismo existen condiciones que nos guían para saber cuándo se deben usar índices. Cuando la selectividad está alrededor o menor de un 25% según Oracle, o cuando el WHERE se hace sobre una columna que tenga un rango, también sobre funciones agregadas (AVG, SUM, MAX) sin WHERE o GROUP BY, también cuando se usa ORDER BY sobre columnas que tengan un índice, y un CONSTRAINT de nulidad.

* **Índices creados de forma automática por Oracle**



*Ilustración 1. Índices*

*Creados por Oracle*

Como podemos ver en la ilustración 1, los índices creados automáticamente por Oracle son en su gran mayoría atributos primary Key, fueron creados debido a que ayudan al rendimiento de las consultas pues, cuando se generan las consultas con distintos *join* estos índices son los que hacen referencia a las distintas columnas por donde se combinan distintas tablas a la hora de cumplir una condición dada. Por otra parte la mayoría de estos índices hacen referencia a columnas donde la selectividad es menor a 20% por lo cual ni incumple ninguna regla, ninguno de estos índices tiene una selectividad mayor a 20%. Así mismo es el principal factor para reducir I/O en el disco cuando son usados correctamente. Al ser atributos con valores distintos y no nulos, Oracle crea un índice automático.

**RFC11. Consultar Material 2**

|  |  |
| --- | --- |
| Sin índices | **RFC11** |
| Enunciado Requerimiento | Dado el identificador de un material, mostrar la información de los pedidos en los que ha estado involucrado, incluyendo los que están en etapa de producción. Para los pedidos no activos actualmente debe especificarse el rango de fechas en los que estuvo activo. |
| SQL1 | *select distinct id\_mat, ped, email\_cliente, fecha\_pedido, fecha\_entrega, estado, cant from (select id\_mat, ped, email\_cliente, fecha\_pedido, fecha\_entrega, estado, cant, id\_comp as comp  from (select pedidos\_clientes.id\_pedido as ped, email\_cliente, fecha\_pedido, fecha\_entrega, estado, id\_comp as idcomp, cantidad from pedidos\_clientes join contenido\_pedido\_mat on pedidos\_clientes.id\_pedido=contenido\_pedido\_mat.id\_pedido) join (select id\_comp, id\_mat, cantidad as cant from compuesto\_de\_comp) on idcomp=id\_comp order by ped) join (select codigo, id\_producto, id\_comp from etapas\_produccion where num\_etapa between 1 and 9) on comp = id\_comp where id\_mat='&material';* |
| SQL2 | *select codigo, id\_producto, num\_etapa, codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant, idMat from etapas\_produccion inner join (select codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant, id\_mat as idMat from compuesto\_de\_comp inner join (select codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant from (select id\_pedido as codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado from pedidos\_clientes) left join (select id\_pedido as idPedido, id\_comp as idComp, cantidad as cant from contenido\_pedido\_mat) on codPedido=idPedido* ***intersect*** *select codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant from (select id\_pedido as codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado from pedidos\_clientes) right join (select id\_pedido as idPedido, id\_comp as idComp, cantidad as cant from contenido\_pedido\_mat) on codPedido=idPedido) on idComp=compuesto\_de\_comp.ID\_COMP) on idComp=etapas\_produccion.ID\_COMP;* |
| Plan 1 |  |
| Plan 2 |  |
| Análisis Plan1 | En este plan, Oracle recorre la tabla verificando que tuplas cumplen con las dos condiciones. En este existen dos índices automáticos generados por Oracle que son las llaves primarias de pedidos y y contenido pedido materiales. |
| Análisis Plan 2 | En este plan primero se realizan los join y después se intersecan pedidos\_clientes con contenido\_peido\_mat para después hacer un join entre compuesto\_de\_comp y etapas de producción |

|  |  |
| --- | --- |
| Con índices | **RFC11** |
| Enunciado Requerimiento | Dado el identificador de un material, mostrar la información de los pedidos en los que ha estado involucrado, incluyendo los que están en etapa de producción. Para los pedidos no activos actualmente debe especificarse el rango de fechas en los que estuvo activo. |
| Primer índice | *CREATE INDEX INDEX\_ID\_MAT1 ON COMPUESTO\_DE\_COMP(ID\_MAT);* |
| Segundo índice | *CREATE INDEX INDEX\_NUMETAPA ON ETAPAS\_PRODUCCION(NUM\_ETAPA);* |
| SQL1 | *select distinct id\_mat, ped, email\_cliente, fecha\_pedido, fecha\_entrega, estado, cant from (select id\_mat, ped, email\_cliente, fecha\_pedido, fecha\_entrega, estado, cant, id\_comp as comp  from (select pedidos\_clientes.id\_pedido as ped, email\_cliente, fecha\_pedido, fecha\_entrega, estado, id\_comp as idcomp, cantidad from pedidos\_clientes join contenido\_pedido\_mat on pedidos\_clientes.id\_pedido=contenido\_pedido\_mat.id\_pedido) join (select id\_comp, id\_mat, cantidad as cant from compuesto\_de\_comp) on idcomp=id\_comp order by ped) join (select codigo, id\_producto, id\_comp from etapas\_produccion where num\_etapa between 1 and 9) on comp = id\_comp where id\_mat='&material';* |
| SQL2 | *select codigo, id\_producto, num\_etapa, codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant, idMat from etapas\_produccion inner join (select codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant, id\_mat as idMat from compuesto\_de\_comp inner join (select codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant from (select id\_pedido as codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado from pedidos\_clientes) left join (select id\_pedido as idPedido, id\_comp as idComp, cantidad as cant from contenido\_pedido\_mat) on codPedido=idPedido* ***intersect*** *select codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado, idComp, cant from (select id\_pedido as codPedido, email\_cliente, fecha\_pedido, fecha\_entrega, estado from pedidos\_clientes) right join (select id\_pedido as idPedido, id\_comp as idComp, cantidad as cant from contenido\_pedido\_mat) on codPedido=idPedido) on idComp=compuesto\_de\_comp.ID\_COMP) on idComp=etapas\_produccion.ID\_COMP;* |
| Plan 1 – Primer índice |  |
| Plan 1 – Segundo Índice |  |
| Plan 2 – Primer Índice |  |
| Plan 2 – Segundo Índice |  |
| Análisis Índice 1 | Al comparar los resultados con los del plan sin índices, se puede notar que con índices, el costo es optimizado.  |Sin indices |Con indices  Plan 1:| 8 |6  Plan 2:| 68 |18 |
| Análisis Índice 2 | Comparando los resultados con los planes sin índices, podemos concluir que hubo mejoría solo en el caso del plan 1, donde se redujo nuevamente a 5, y el plan 2 quedo igual.  |Sin indices |Con indices  Plan 1:| 8 |5  Plan 2:| 68 |68 |

**RFC10. Consultar Pedidos 2**

|  |  |
| --- | --- |
| Sin índices | **RFC10** |
| Enunciado Requerimiento | Mostrar todos los pedidos que usan materiales de un tipo X con costo valor mayor a Y. Los valores de X y Y son dados por el usuario. |
| SQL1 | *select distinct id\_pedido, id\_producto, id\_mat, nombre, cant, costo\_unitario from materias\_primas join (select distinct id\_pedido, cant, id\_producto, id\_mat, costo\_unitario  from productos join (select distinct  id\_pedido, idComponente, cant, id\_producto, id\_mat from contenido\_pedido\_mat join (select id\_producto, compuesto\_de\_prod.id\_componente as idComponente, compuesto\_de\_comp.cantidad as cant, id\_mat from compuesto\_de\_prod join compuesto\_de\_comp on compuesto\_de\_comp.id\_comp=compuesto\_de\_prod.id\_componente) on contenido\_pedido\_mat.id\_comp=idComponente order by id\_pedido) on codigo=id\_producto) on codigo=id\_mat;* |
| SQL2 | *select id\_pedido, codMaterial, codProducto, cantFinal, costo\_unitario from productos join (select codMaterial, id\_producto as codProducto, id\_pedido, cantFinal from compuesto\_de\_prod inner join (select id\_mat as codMaterial, id\_pedido, codComponente, cantFinal from (select id\_pedido, id\_comp as codComponente, cantidad as cantFinal from contenido\_pedido\_mat) left join (select id\_comp as comp, id\_mat from compuesto\_de\_comp) on codComponente = comp* ***intersect*** *select id\_mat as codMaterial, id\_pedido, codComponente, cantFinal from (select id\_pedido, id\_comp as codComponente, cantidad as cantFinal from contenido\_pedido\_mat) right join (select id\_comp as comp, id\_mat from compuesto\_de\_comp) on codComponente = comp) on codComponente = compuesto\_de\_prod.ID\_COMPONENTE) on codProducto=productos.codigo order by id\_pedido;* |
| Plan 1 |  |
| Plan 2 |  |
| Análisis Plan1 | En este plan, Oracle recorre la tabla verificando que tuplas cumplen con las dos condiciones. En este existen tres índices automáticos generados por Oracle que son las llaves primarias de compuesto de componentes, compuesto de productos y contenido pedido material. |
| Análisis Plan 2 | En este plan primero se realizan los join y después se intersecan pedidos\_clientes con contenido\_pedido\_mat para después hacer un join entre compuesto\_de\_comp y etapas de producción |

|  |  |
| --- | --- |
| Con índices | **RFC10** |
| Enunciado Requerimiento | Mostrar todos los pedidos que usan materiales de un tipo X con costo valor mayor a Y. Los valores de X y Y son dados por el usuario |
| SQL1 | *select distinct id\_pedido, id\_producto, id\_mat, nombre, cant, costo\_unitario from materias\_primas join (select distinct id\_pedido, cant, id\_producto, id\_mat, costo\_unitario  from productos join (select distinct  id\_pedido, idComponente, cant, id\_producto, id\_mat from contenido\_pedido\_mat join (select id\_producto, compuesto\_de\_prod.id\_componente as idComponente, compuesto\_de\_comp.cantidad as cant, id\_mat from compuesto\_de\_prod join compuesto\_de\_comp on compuesto\_de\_comp.id\_comp=compuesto\_de\_prod.id\_componente) on contenido\_pedido\_mat.id\_comp=idComponente order by id\_pedido) on codigo=id\_producto) on codigo=id\_mat;* |
| SQL2 | *select id\_pedido, codMaterial, codProducto, cantFinal, costo\_unitario from productos join (select codMaterial, id\_producto as codProducto, id\_pedido, cantFinal from compuesto\_de\_prod inner join (select id\_mat as codMaterial, id\_pedido, codComponente, cantFinal from (select id\_pedido, id\_comp as codComponente, cantidad as cantFinal from contenido\_pedido\_mat) left join (select id\_comp as comp, id\_mat from compuesto\_de\_comp) on codComponente = comp* ***intersect*** *select id\_mat as codMaterial, id\_pedido, codComponente, cantFinal from (select id\_pedido, id\_comp as codComponente, cantidad as cantFinal from contenido\_pedido\_mat) right join (select id\_comp as comp, id\_mat from compuesto\_de\_comp) on codComponente = comp) on codComponente = compuesto\_de\_prod.ID\_COMPONENTE) on codProducto=productos.codigo order by id\_pedido;* |
| Primer índice | *CREATE INDEX INDEX\_CANTIDAD ON CONTENIDO\_PEDIDO\_MAT(CANTIDAD);* |
| Plan 1 – Primer Índice |  |
| Plan 2 – Primer Índice |  |
| Análisis Plan1 | Al comparar los resultados con los del plan sin índices, se puede notar que con índices, el costo ni aumento ni disminuyo.  |Sin indices |Con indices  Plan 1:| 6 |6 |
| Análisis Plan 2 | Comparando los resultados con los planes sin índices, podemos concluir que hubo mejoría solo en el caso del plan 2 con índice debido a que se redujo su costo  |Sin indices |Con indices  Plan 2:| 12 |9  Plan 2:| 12 |68 |

**RFC9. Consultar Ejecución de etapas 2**

|  |  |
| --- | --- |
| Sin Índices | **RFC9** |
| Enunciado Requerimiento | Mostrar las etapas de producción ejecutadas en un rango de tiempo (dado por el usuario), que NO correspondan a un criterio de búsqueda asociado con la ejecución de dicha etapa (material, tipo de material, pedido, cantidades, ...), también dado por el usuario. |
| SQL 1 | *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion right join ejecucion\_etapa on ejecucion\_etapa.id\_etapa=etapas\_produccion.codigo* ***intersect*** *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion left join ejecución\_etapa on ejecución\_etapa.id\_etapa=etapas\_produccion.codigo where fecha\_inicio >= '26/01/14' and fecha\_fin <= '17/05/16';* |
| SQL 2 | *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion* ***inner join*** *ejecucion\_etapa on etapas\_produccion.codigo=ejecucion\_etapa.id\_etapa where fecha\_inicio >= '26/01/14' and fecha\_fin <= '17/05/16';* |
| Plan 1 |  |
| Plan 2 |  |
| Análisis Plan 1 | En este plan se genera el join izquierdo y derecho entre etapas de producción y ejecución de dichas etapas y después se procede hacer el intersecto de dichas sentencias. |
| Análisis Plan 2 | En este plan, Oracle recorre la tabla verificando que tuplas cumplen con las dos condiciones. En este existen tres índices automáticos generados por Oracle que son las llaves primarias de compuesto de componentes, compuesto de productos y contenido pedido material. |

|  |  |
| --- | --- |
| Con Índices | **RFC9** |
| Enunciado Requerimiento | Mostrar las etapas de producción ejecutadas en un rango de tiempo (dado por el usuario), que **NO** correspondan a un criterio de búsqueda asociado con la ejecución de dicha etapa (material, tipo de material, pedido, cantidades, ...), también dado por el usuario. |
| SQL 1 | *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion right join ejecucion\_etapa on ejecucion\_etapa.id\_etapa=etapas\_produccion.codigo* ***intersect*** *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion left join ejecución\_etapa on ejecución\_etapa.id\_etapa=etapas\_produccion.codigo where fecha\_inicio >= '26/01/14' and fecha\_fin <= '17/05/16';* |
| SQL 2 | *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion* ***inner join*** *ejecucion\_etapa on etapas\_produccion.codigo=ejecucion\_etapa.id\_etapa where fecha\_inicio >= '26/01/14' and fecha\_fin <= '17/05/16';* |
| Índice 1 | *CREATE INDEX INDEX\_FECHA\_INICIO\_ETAPA ON EJECUCION\_ETAPA(FECHA\_INICIO,FECHA\_FIN);* |
| Índice 2 | *CREATE INDEX INDEX\_FECHA\_INICIO\_ETAPA ON EJECUCION\_ETAPA(FECHA\_INICIO);* |
| Plan 1 – Índice 1 |  |
| Plan 2 – Índice 1 |  |
| Plan 1 – Índice 2 |  |
| Plan 2 – Índice 1 |  |
| Análisis Plan 1 | Al comparar los resultados con los del plan sin índices, se puede notar que con índices, el costo es optimizado.  |Sin indices |Con indices  Plan 1:| 15 |14  Plan 2:| 53 |52 |
| Análisis Plan 2 | Comparando los resultados con los planes sin índices, y a su vez con el índice combinado podemos ver que la consulta se optimiza pero no es diferente con índices combinados índices simples.  |Sin indices |Con indices  Plan 1:| 15 |14  Plan 2:| 53 |52 |

**RFC8. Consultar Ejecución de etapas 2**

|  |  |
| --- | --- |
| Sin Índices | **RFC9** |
| Enunciado Requerimiento | Mostrar las etapas de producción ejecutadas en un rango de tiempo (dado por el usuario), que NO correspondan a un criterio de búsqueda asociado con la ejecución de dicha etapa (material, tipo de material, pedido, cantidades, ...), también dado por el usuario. |
| SQL 1 | *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion right join ejecucion\_etapa on ejecucion\_etapa.id\_etapa=etapas\_produccion.codigo* ***intersect*** *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion left join ejecución\_etapa on ejecución\_etapa.id\_etapa=etapas\_produccion.codigo where fecha\_inicio >= '26/01/14' and fecha\_fin <= '17/05/16';* |
| SQL 2 | *select codigo as codEtapa, descripcion, fecha\_inicio, fecha\_fin from etapas\_produccion* ***inner join*** *ejecucion\_etapa on etapas\_produccion.codigo=ejecucion\_etapa.id\_etapa where fecha\_inicio >= '26/01/14' and fecha\_fin <= '17/05/16';* |
| Plan 1 |  |
| Plan 2 |  |
| Análisis Plan 1 | En este plan se genera el join izquierdo y derecho entre etapas de producción y ejecución de dichas etapas y después se procede hacer el intersecto de dichas sentencias. |
| Análisis Plan 2 | En este plan, Oracle recorre la tabla verificando que tuplas cumplen con las dos condiciones. En este existen tres índices automáticos generados por Oracle que son las llaves primarias de compuesto de componentes, compuesto de productos y contenido pedido material. |