

Alumno/a:	Laura Álvarez Flórez	NIA:	100363965
Alumno/a:	Laura Yunta García	NIA:	100363785

1 Introducción

En este proyecto se ha llevado a cabo la mejora del diseño físico de una base de datos que almacena información sobre la base de datos tratada ya en prácticas anteriores.
 Este documento recoge detalladamente todo el proceso, desde el análisis hasta la conclusión obtenida.

Concretamente, esto es lo que será tratado en los siguientes puntos:

2. Análisis

Se ha analizado el diseño inicial globalmente y específicamente para hacer una primera determinación de posibles aspectos a mejorar.

3. Diseño físico

En este punto basándonos en el conocimiento obtenido en el apartado 1 detallamos el diseño propuesto y especificamos con detalle las fortalezas de este.

4. Evaluación

Esta sección documenta la comparación directa entre el diseño inicial y el diseño propuesto y analiza los resultados.

5. Conclusiones

Resumen global del desarrollo del contenido del documento y comentarios importantes al respecto de la asignatura.


2 Análisis

El diseño físico inicial produce numerosos accesos y es costoso cuando se realizan búsquedas, inserciones o actualización por una clave que no es primaria, el diseño presenta debilidades en ese aspecto que podrían ser mejoradas añadiendo organizaciones auxiliares.

A continuación, esta tabla presenta los datos obtenidos tras realizar los test sobre el diseño actual:

INSTRUCCIÓN	SALIDAS		
	TEST → SERVEROUTPUT	TEST → TIMING	TEST → AUTOTRACE
Proceso general	RESULTS AT 04-MAY-18 TIME CONSUMPTION: 2333858 milliseconds. CONSISTENT GETS: 13129465,4 blocks PL/SQL procedure successfully completed.	RESULTS AT 04-MAY-18 TIME CONSUMPTION: 2214529 milliseconds. CONSISTENT GETS: 17358020 blocks PL/SQL procedure successfully completed. Elapsed: 00:40:34.81	RESULTS AT 05-MAY-18 TIME CONSUMPTION: 2226200 milliseconds. CONSISTENT GETS: 26770644,4 blocks PL/SQL procedure successfully completed

Query 1	RESULT AT 05-MAY-18 TIME CONSUMPTION: 31 milliseconds. CONSISTENT GETS: 214,5 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:55.94	RESULT AT 05-MAY-18 TIME CONSUMPTION: 16 milliseconds. CONSISTENT GETS: 214,5blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:35.03	RESULT AT 05-MAY-18 TIME CONSUMPTION: 31 milliseconds. CONSISTENT GETS: 214,5 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:38.90
Query 2	RESULT AT 05-MAY-18 TIME CONSUMPTION: 47 milliseconds. CONSISTENT GETS: 528,8 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:41.19	RESULT AT 05-MAY-18 TIME CONSUMPTION: 46 milliseconds. CONSISTENT GETS: 528,8 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:37.45	RESULT AT 05-MAY-18 TIME CONSUMPTION: 47 milliseconds. CONSISTENT GETS: 528,8 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:39.81
Query 3	RESULT AT 05-MAY-18 TIME CONSUMPTION: 0 milliseconds. CONSISTENT GETS: 75,5 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:40.86	RESULT AT 05-MAY-18 TIME CONSUMPTION: 15 milliseconds. CONSISTENT GETS: 75,5 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:42.34	RESULT AT 05-MAY-18 TIME CONSUMPTION: 16 milliseconds. CONSISTENT 75,5 GETS: blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:38.22
Query 4	RESULT AT 05-MAY-18 TIME CONSUMPTION: 46 milliseconds. CONSISTENT GETS: 76,8 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:27.70	RESULT AT 05-MAY-18 TIME CONSUMPTION: 31 milliseconds. CONSISTENT GETS: 76,8 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:32.32	RESULT AT 05-MAY-18 TIME CONSUMPTION: 47 milliseconds. CONSISTENT GETS: 76,8 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:29.15
Paying tickets	RESULT AT 05-MAY-18 TIME CONSUMPTION: 281 milliseconds. CONSISTENT GETS: 264,4 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:22.75	RESULT AT 05-MAY-18 TIME CONSUMPTION: 281 milliseconds. CONSISTENT GETS: 264,4 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:23.76	RESULT AT 05-MAY-18 TIME CONSUMPTION: 281 milliseconds. CONSISTENT GETS: 264,9 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:27.45
Delete	RESULTS AT 05-MAY-18 TIME CONSUMPTION: 57047 milliseconds. CONSISTENT GETS: 220574,4 blocks. PL/SQL procedure successfully	RESULTS AT 05-MAY-18 TIME CONSUMPTION: 52391 milliseconds. CONSISTENT GETS: 1899526,3 blocks. PL/SQL procedure successfully completed. Elapsed: 00:02:02.1	RESULT AT 05-MAY-18 TIME CONSUMPTION: 37610 milliseconds. CONSISTENT GETS: 1686740,7 blocks PL/SQL procedure successfully completed. Elapsed: 00:01:30.58
Insert	RESULT AT 05-MAY-18 TIME CONSUMPTION: 9219 milliseconds.	RESULT AT 05-MAY-18 TIME CONSUMPTION: 6891 milliseconds.	RESULT AT 05-MAY-18 TIME CONSUMPTION: 6328 milliseconds.

Titulación: GRADO INGENIERIA INFORMATICA Año Académico: 2017/2018 Curso: 2º Asignatura: Ficheros y Bases de Datos Título: Memoria Práctica 3 – Diseño Físico en Oracle	 uc3m Universidad Carlos III de Madrid
--	--

	CONSISTENT GETS: 52636,8 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:07.44	CONSISTENT GETS: 49487,9 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:22.35	CONSISTENT GETS: 49478,2 blocks PL/SQL procedure succesfully completed. Elapsed: 00:01:45.19set
--	--	--	---

En la tabla anterior se puede ver el coste de cada requerimiento de la carga de trabajo individual y el de la carga en su totalidad, si analizamos independientemente cada uno vemos que el diseño podría mejorarse en los siguientes aspectos:

- *Query 4* es muy costosa debido al elevado número de joins que presenta
- El *Update* sobre tickets afecta a numerosas filas y podría crearse un índice para lograr una actualización menos costosa y mas eficiente.
- Existen numerosas operaciones *to_char*, *to_date*... lo que nos indica que quizás podría realizarse un cambio físico-lógico en el diseño pero nos ceñiremos al diseño físico por lo que podríamos solucionarlo creando un índice para esas funciones.
- Observamos numerosos accesos a el atributo *road* que creemos que podrían ser mejorados con un cluster en dicha columna.

3 Diseño Físico

En vista a mejorar el diseño de la base de datos ya existente, proponemos los siguientes cambios o modificaciones. Estos cambios han sido determinados tras realizar numerosas pruebas y comprobaciones donde quedaba demostrado que mejoraban y no empeoraban el diseño en algún proceso concreto. Finalmente hemos elegido la combinación de modificaciones que mejor resultado global creíamos que iba a tener.

1. Alteración del espacio libre distribuido.

Se ha realizado una modificación en el diseño del cubo de la tabla tickets. Hemos elegido estas tablas debido a que en un análisis global nos parecían candidatas adecuadas por no sufrir muchas inserciones, borrados o actualizaciones que provocasen una necesidad de más espacio para las diferentes filas.

```
ALTER TABLE tickets PCTFREE 5;
```

Hemos optado por mantener un 5%, disminuyendo el valor por defecto de ORACLE 20%, este ha sido determinado tras realizar varias pruebas con valores menores como 0% o 2% y comprobar que era la estimación que mejores resultados ofrecía.

2. Índices

Para continuar con la descripción del nuevo diseño implementado, nombraremos los índices creados y explicaremos su uso o importancia en el diseño.

Tick_ind

```
CREATE INDEX tick_ind ON tickets ( state, pay_type, pay_date) TABLESPACE users;
```

Este índice a la tabla tickets contiene las columnas utilizadas en el UPDATE “paying tickets” de esta manera se realiza un update mucho más eficiente y rápido.

Obs_ind

```
CREATE INDEX obs_ind ON observations ( nPlate) TABLESPACE users;
```

Con esta estructura conseguimos simplificar la Query 1, indexando el atributo n_plate se puede acceder a dicho atributo más fácilmente. Estos índices son apropiados ya que están realizados sobre atributos con un alto nivel de filas, es decir, existen numerosas filas con diferentes valores de matrícula (nPlate). Además estas columnas no se modifican frecuentemente.

Veh_ind

```
CREATE INDEX veh_ind ON vehicles ( nPlate, reg_driver) TABLESPACE users;
```

Este índice permite operaciones veloces si se desea acceder a dos atributos muy usados en la carga de trabajo como nPlate y regDriver, mejora los accesos para la query 3 la cual utiliza la vista qw_drivers.

ind_func_toChar1 y ind_func_toChar2

```
CREATE INDEX ind_func_toChar1 ON tickets (TO_CHAR(pay_date,'YYYYMM')) TABLESPACE users;  
CREATE INDEX ind_func_toChar2 ON observations (TO_CHAR(odatetime,'YYYYMMDD')) TABLESPACE users;
```

Con la creación de estos índices logramos incrementar la velocidad de la realización de la función del sistema To_Char, hemos optado por esta decisión de diseño ya que tanto en la Query 1 como en la Query 2 se realiza dicha operación. La definición de un índice basado en función en la columna o expresión transformada permite que los datos se devuelvan utilizando el índice.

3. Road_cluster

```
CREATE CLUSTER road_cluster (road VARCHAR2(5));  
CREATE INDEX road_cluster_ind ON CLUSTER road_cluster;  
CREATE TABLE RADARS(...)CLUSTER road_cluster (road);  
CREATE TABLE OBSERVATIONS(...)CLUSTER road_cluster (road);
```

Hemos decidido que la columna road común en diferentes tablas como radars o observations tendría sentido conseguir un almacenamiento de estas conjunto para mejorar el tiempo de acceso.

Por ello hemos decidido realizar un cluster a dicha variable en las tablas mencionadas, consiguiendo así mejoras en el borrado, inserción o actualización de operaciones que impliquen dicha columna y un decremento directo en el tiempo/espacio consumido por los joins de las Querys 2, 3 y 4.

4. Hints

Finalmente nuestro diseño físico introduce las siguientes instrucciones:

```

/*+ USE_NL(A B) */
/*+ USE_NL(E F) */

```

Las cuales animan a ORACLE a unir dichas tablas (procedentes de view new_ticket y view qw_drivers respectivamente) especificadas.

Este hint es de uso común en joins por lo que nos ha parecido una buena idea implementarlo.

4 Evaluación

Tras el trabajo realizado analizando las diferentes partes de la carga de trabajo, las posibles mejoras y como los afectan de distinta manera a unos procesos o a otros, hemos sido testigos de una evolución en el rendimiento de la base de datos muy considerable.

La propuesta del diseño físico mejora respecto al diseño inicial, pudiéndose observar los resultados concretos en la siguiente tabla:

Diseño		
Inicial	RESULTS AT 04-MAY-18 TIME CONSUMPTION: 2333858 milliseconds. CONSISTENT GETS: 13129465,4 blocks PL/SQL procedure successfully completed. SQL>	RESULTS AT 04-MAY-18 TIME CONSUMPTION: 2214529 milliseconds. CONSISTENT GETS: 17358020 blocks PL/SQL procedure successfully completed. Elapsed: 00:40:34.81 SQL> SQL>
Propuesto	RESULTS AT 05-MAY-18 TIME CONSUMPTION: 617711 milliseconds. CONSISTENT GETS: 7372675 blocks PL/SQL procedure successfully completed.	RESULTS AT 05-MAY-18 TIME CONSUMPTION: 1607708 milliseconds. CONSISTENT GETS: 16065199 blocks PL/SQL procedure successfully completed. Elapsed: 00:28:34.01 SQL>

Hemos rescatado la información presentada en el apartado 1 donde analizábamos el diseño inicial dado, y a su vez utilizado estos datos como referencia para medir el nivel de mejora que hemos alcanzado.

Como se puede observar en la primera columna de la entrada en el test 1 se ha disminuido el número de blocks aproximadamente al 40%, en el test numero 2 también se observa un decremento algo más leve del número de bloques.

El resultado es el esperado, haciendo un cómputo global, con los cambios de diseño físico se esperaba una mejora en los resultados de los test respecto a los resultados anteriores. Por otra parte no teníamos conocimiento suficiente para estimar cual sería el porcentaje de mejora que creíamos poder obtener por lo que no podemos afirmar nada sobre el número de bloques que ha disminuido o el tiempo.

5 Conclusiones Finales

En esta práctica hemos comprendido el proceso de gestionar una base de datos, comenzando con el análisis de un diseño que no ha sido creado de nuestra mano y documentándonos para proponer posibles mejoras de este. Una vez ya hubiesen sido determinados los fallos del diseño y cómo podríamos mejorarlo, hemos tenido que determinar cuál era el mejor resultado global e intentar obviar pequeñas mejoras de procesos individuales.

Esta práctica nos ha proporcionado conocimientos extra para la asignatura y soltura en cuanto al manejo de organizaciones auxiliares, habiendo asentado ya lo que es un diseño físico y como mejorarlo con índices y clústers principalmente.

Cierto es que no poseemos el conocimiento todavía para poder determinar con seguridad que los resultados obtenidos son buenos, pero tras la comparación realizada y observando que la diferencia de resultado en los test del diseño antiguo al propuesto casi alcanza el 50% nos indica que el resultado obtenido es bueno y hemos conseguido superar el diseño inicial.

En general creemos haber avanzado desde el inicio de la asignatura considerablemente, conseguido fluidez en el uso de ORACLE y conocido como es el trabajo de realizar una base de datos. Sobre todo hemos aprendido a ser autodidactas en cuanto a la búsqueda de información para resolver un error ocurrido o alguna duda concreta del programa o el lenguaje.

Nos parece que la distribución de una práctica grande, en prácticas más pequeñas es un punto positivo. Al igual que el hecho de que se proporcione al alumno el material para poder continuar con el siguiente paso aun habiendo realizado mal el anterior.

Una propuesta que nos gustaría comentar es el hecho de disminuir la dimensión de los problemas, para que así el alumno tenga tiempo para llegar más lejos o recapacitar más en cuanto a la propuesta del diseño físico o del esquema relacional. Además el tiempo a esperar cada vez que se lleva a cabo un test es demasiado elevado y extiende mucho la práctica en algo que es insustancial a la hora de aprender y formarse en el temario.

En cuanto a la asignatura creemos que se imparte de una manera correcta y la estructura es adecuada, aunque nos hubiera gustado que hubiese más clases de laboratorio y menos de ejercicios teóricos ya que es más útil a la hora de avanzar en la evaluación continua y superar las prácticas y los test teóricos.

Titulación: GRADO INGENIERIA INFORMATICA

Año Académico: 2017/2018

Curso: 2º

Asignatura: Ficheros y Bases de Datos

Título: Memoria Práctica 3 – Diseño Físico en Oracle

uc3m

uc3m | Universidad Carlos III de Madrid