

FICHEROS y BB.DD.

Práctica 3

Diseño Físico en Oracle®

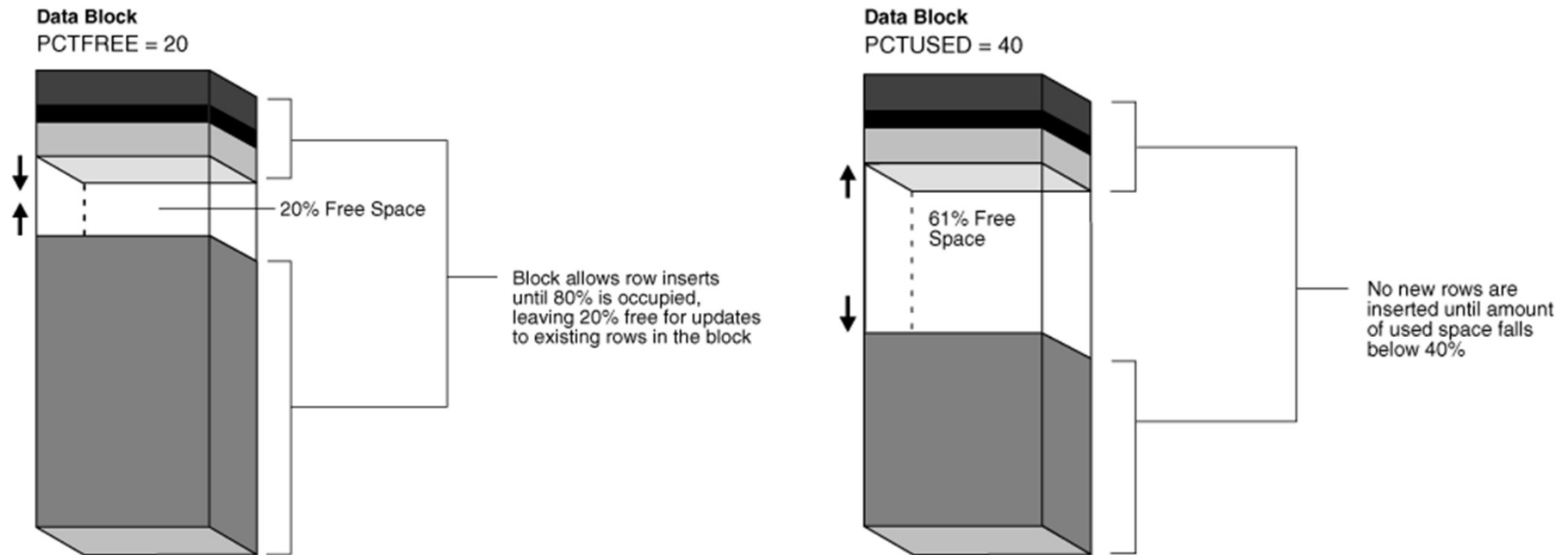
uc3m	Prácticas de la asignatura: hoja de ruta	
sesión 1	<ul style="list-style-type: none"> ➤ Modelado Relacional (<i>esquema relacional</i>) ➤ Implementación: entorno SQL+ (consola interacción) ➤ Estática Relacional: creación de tablas en SQL (LDD) 	práctica 1
sesión 2	<ul style="list-style-type: none"> ➤ Dinámica Relacional <ul style="list-style-type: none"> ➤ consultas básicas en SQL y gestión transaccional ➤ carga de datos (ejecución de scripts + volcado) ➤ del álgebra relacional al SQL 	
sesión 3	<ul style="list-style-type: none"> ➤ Mecanismos de SQL avanzados <ul style="list-style-type: none"> ➤ vistas y disparadores 	práctica 2
sesión 4	<ul style="list-style-type: none"> ➤ Diseño Físico <ul style="list-style-type: none"> ➤ Parametrización de la base ➤ Organizaciones base y auxiliares ➤ Hints 	práctica 3



- Observar **situación actual**: estructuras (volumen), organización, procesos (coste individual y frecuencias), claves de proyección, claves de selección (cardinalidad dominios), combinaciones,...
- Proponer **mejoras** para cada proceso
- Analizar el **impacto** de cada mejora sobre el resto de procesos
- Realizar una **propuesta** final → **diseño físico**
 - Organizaciones base (clusters)
 - Parametrización (parámetros físicos de cubo, archivo, sistema...)
 - Estructuras auxiliares
 - Otras mejoras (hints, reescritura consultas y procedimientos, ...)
- **Evaluar**

- Algunos parámetros ya viene fijados, por ejemplo
 1. Espacio de cubo (8KB, denominado bloque Oracle[®]), aunque a partir de la v. 11 puede cambiarse para cada *tablespace*, y por ende para los objetos creados en él.
 2. Los ficheros de datos son seriales no-consecutivos
 3. El espacio libre distribuido es PCTFREE=10 & PCTUSED=60, pero puede cambiarse para cada segmento (objeto).
- Pero se puede *clusterizar* para redefinirlos...
- Por otro lado, se debe realizar un *diseño físico*

Ejemplo de cubo en ORACLE® con PCFREE y PCUSED



- En nuestra instancia de BD hemos creado varios *tablespaces* para poder trabajar con diferentes tamaños de cubo

```
CREATE TABLESPACE TAB_2k DATAFILE 'TAB_2k.dbf' BLOCKSIZE 2048;  
CREATE TABLESPACE TAB_8k DATAFILE 'TAB_8k.dbf' BLOCKSIZE 8192; -- default  
CREATE TABLESPACE TAB_16k DATAFILE 'TAB_16k.dbf' BLOCKSIZE 16384;
```

- Se puede indicar el *tablespace* donde se quiere almacenar una tabla, índice, cluster..

```
CREATE ... TABLESPACE tu_tablespace;
```

- Algunos objetos Oracle se pueden mover a otro *tablespace*.

```
ALTER INDEX nombre_ind REBUILD TABLESPACE nuevo_tablespace;
```

- La selección de índices forma parte del diseño físico
- Consiste en decidir las estructuras auxiliares para optimizar el rendimiento de la BD de acuerdo a los procesos que la actualizan o consultan
- La sintaxis básica de creación de índices es:

```
CREATE [ind_type] INDEX ind_name ON  
      table_name(ind_key) TABLESPACE name;
```

donde

 **primario**

 **bitmap**

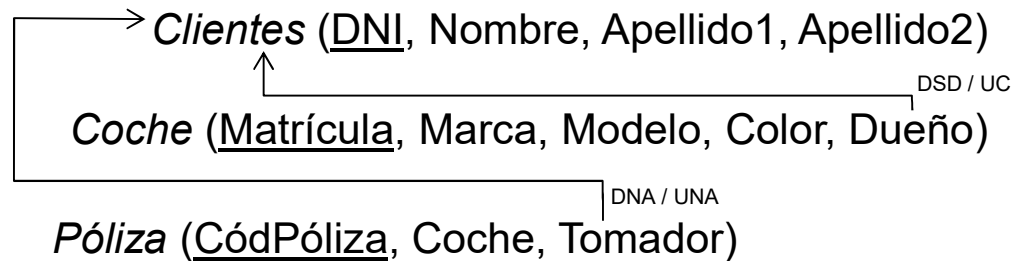
 **secundario**

ind_type := UNIQUE | BITMAP | (default)

ind_key := <columnas de tabla> (sep. por comas)

- En ORACLE®, un *cluster* es la definición de clave privilegiada.
- Las tablas de un *cluster* se almacenan conjuntamente (**toda la fila combinada se almacena físicamente en el mismo cubo**). Beneficia JOIN y acceso por la clave privilegiada, perjudica a todo lo demás.
- La elección de la **clave** es **crítica**: puede bajar la densidad.
- Se puede redefinir el tamaño de cubo (celdas).
- *Se puede hacer un cluster mono-tabla* (cambia organización base)
- El *cluster* puede ser **indizado** o **disperso** (con **ordenación** opcional) para mejorar la **densidad**, los **procesos selectivos**, y los **procesos ordenados**, respectivamente).
- El cluster debe **crearse antes de crear la tabla**
http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_5001.htm#i2105031

Ejemplo:



```
CREATE CLUSTER identidad (DNI VARCHAR2(9)) TABLESPACE users;
```

Por defecto, el cluster es indizado

Añadimos las tablas la cluster

```
CREATE TABLE cliente(...) CLUSTER identidad (DNI);
```

```
CREATE TABLE coche(...) CLUSTER identidad (dueño);
```

```
CREATE TABLE poliza(...) CLUSTER identidad (tomador);
```

```
CREATE INDEX ind_dni ON CLUSTER identidad TABLESPACE users;
```



```
identidad
```

```
( DNI C(9),
```

```
  cliente (nombre C(25), apellido1 C(15), apellido2 C(15)),
```

```
  coche (matrícula C(7), marca C(20), modelo C(20), color C(10) )*,
```

```
  poliza (cod C(30), coche C(7) )*
```

```
);
```

Sintaxis de creación de cluster:

```
CREATE CLUSTER clustername (atributo tipo(tam) [, atributo ...])  
    [PCTFREE XX]  
    [PCTUSED YY]  
    [SIZE]  
    [TABLESPACE name]  
    [STORAGE (...)]  
    [INDEX  
    | [SINGLE TABLE] HASHKEYS N [HASH IS ...] ] ;
```

Ejemplos de cluster:

```
CREATE CLUSTER dni_clust(dni NUMBER(8)) INDEX;  
  
CREATE CLUSTER catalog_clust(num_ref NUMBER(10))  
    STORAGE(BUFFER_POOL KEEP);  
  
CREATE CLUSTER student_clust(nia NUMBER(9))  
    SINGLE TABLE HASHKEYS 797 HASH IS MOD(nia,10000);  
  
CREATE CLUSTER ident(DNI NUMBER(8), tlf NUMBER(9))  
    HASHKEYS 503 HASH IS DNI+tlf;  
  
CREATE CLUSTER ident(DNI NUMBER(8), apellido VARCHAR2(15) SORT)  
    HASHKEYS 100 HASH IS DNI;
```

Ejemplo2:

```
CREATE CLUSTER student_clust(nia NUMBER(9))  
      SINGLE TABLE HASHKEYS 797 HASH IS MOD(nia,10000);  
CREATE CLUSTER estudiante (NIA          VARCHAR2(9));  
  
DROP CLUSTER student_clust;
```

→ **ANTES HAY QUE DESTRUIR LAS TABLAS (o como mínimo truncarlas)**

```
EXPLAIN PLAN SET statement_id = 'plan_name' FOR  
<dml_sentence>;
```

```
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

```
SELECT PLAN_TABLE_OUTPUT
```

```
FROM TABLE(DBMS_XPLAN.DISPLAY(NULL, 'plan_name', 'BASIC'));
```

```
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY(FORMAT=>' +ALLSTATS' )) ;
```

- con privilegio select en ...

```
V_$SESSION , V_$SQL , V_$SQL_PLAN , V_$SQL_PLAN_STATISTICS_ALL
```

- Es necesario para medir el rendimiento:
 1. Accediendo a Ora Stats (a través del paquete dado)

```
set serveroutput on
begin
PKG_COSTES.RUN_TEST;
end;
```

2. Activando autotrace on (execution plan & stats)

```
set serveroutput on
set timing on
set autotrace on
http://docs.oracle.com/cd/B28359\_01/server.111/b28274/ex\_plan.htm#i18300
```

(tabla 12 todas las operaciones)

NOTA: las estadísticas de Oracle® pueden dar distintos valores para la misma carga. Por lo que se recomienda ejecutar varias veces la carga (escoger la media)

- Oracle® puede que no elija un camino óptimo (porque no utilice los índices creados, o sí los utiliza pero no de manera eficiente) .
- Los HINTS fuerzan el camino físico para resolver sentencias (select, insert, delete, update), especificándolos como comentarios:

```
SELECT /*+ HINT */ attribute FROM tablename;
```

- Se pueden especificar varios HINTS para la misma sentencia (separados por comas).
- *Repercute en las opciones que puede barajar el optimizador de ORACLE*
- **Ejemplos:** si se especifica **index(clients)**, tomará el mejor índice o combinación de índices sobre la tabla clientes. Si ponemos **index(clients ind1 ind2)** restringe la elección a estos dos (uno de ellos o ambos). Por último, **and_equal(clients ind1 ind2)** fuerza a usar todos los índices especificados (como máx. 5).

Sintaxis del HINT	Descripción
<code>/*+ FULL(tablename) */</code>	full scan of table <i>tablename</i>
<code>/*+ ROWID(tablename) */</code>	rowid scan of table <i>tablename</i>
<code>/*+ INDEX(tablename [indexname [...]]) */</code>	use an index (or several of them)
<code>/*+ NO_INDEX(tablename) */</code>	forbids any index (on <i>tablename</i>)
<code>/*+ NO_INDEX(tablename [indexname [...]]) */</code>	forbids specific index/es
<code>/*+ INDEX_FFS(tablename [indexname [...]]) */</code>	full scan of the index
<code>/*+ AND_EQUAL(tablename [index1 index2 [...]]) */</code>	use more than one index (up to 5)
<code>/*+ INDEX_JOIN(tablename [indexname [...]]) */</code>	join indexes (sort of inverted access)
<code>/*+ CLUSTER(tablename) */</code>	use cluster for <i>tablename</i>
<code>/*+ HASH(tablename) */</code>	use a hash for clustered <i>tablename</i>
<code>insert /*+ append */ ... select...</code>	inserción directa (no buffer, no fl, no RI,...)

- Pasos para realizar la práctica (entrega semana 14):
 1. Preparar la BD
 - a. Ejecuta el script de creación
 - b. Ejecuta el script de carga
 2. Medir la eficiencia (ejecuta `PKG_COSTES.RUN_TEST`)
 3. Plantear e implementar un diseño físico
 4. Medir su eficiencia
 5. Escribir la memoria
 - a. Introducción y análisis del problema
 - b. Diseño físico
 - c. Resultados de la evaluación (comparativa)
 - d. Conclusiones