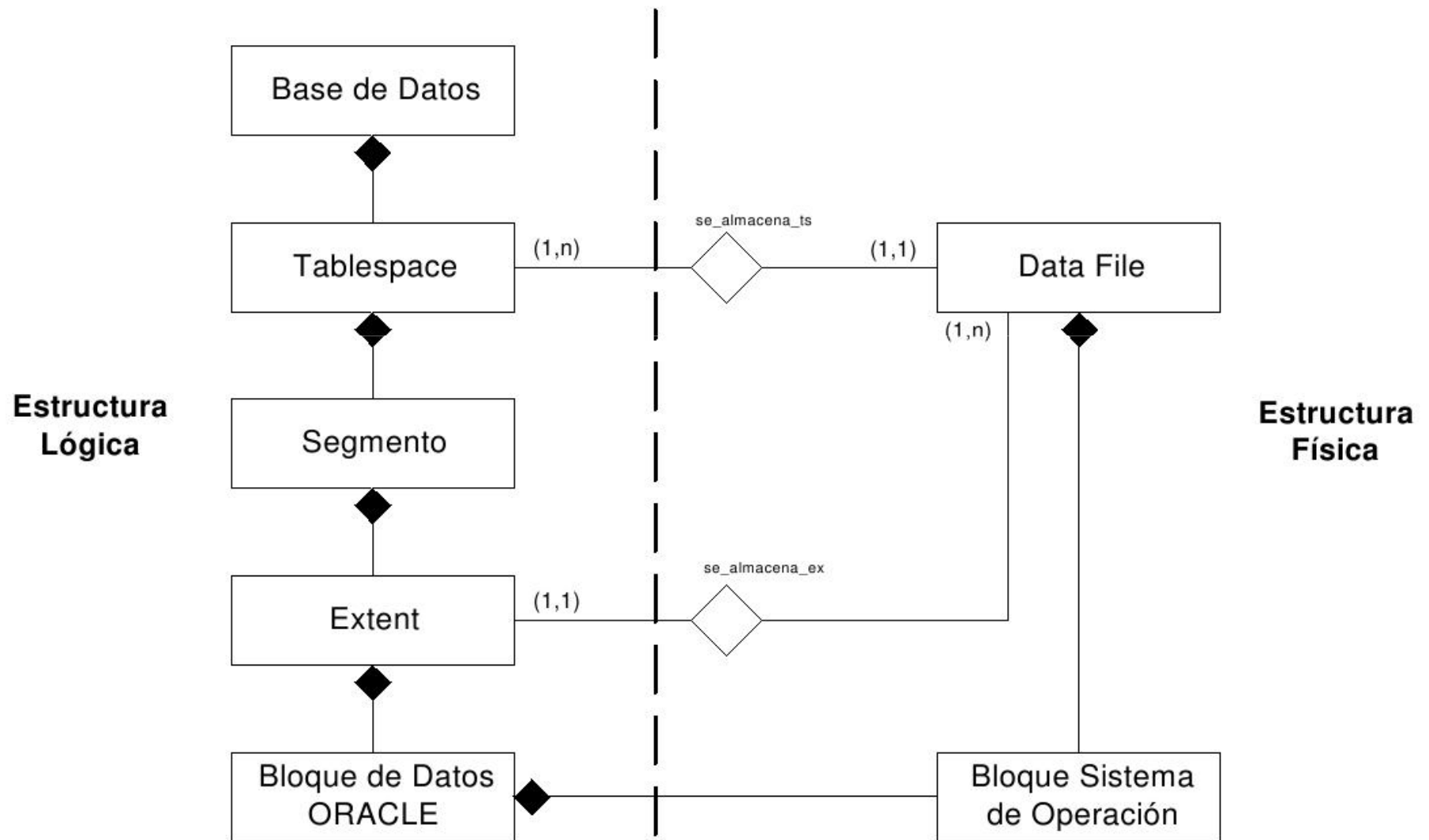

Estructuras de Almacenamiento y sus Relaciones

Estructuras Lógicas y Físicas



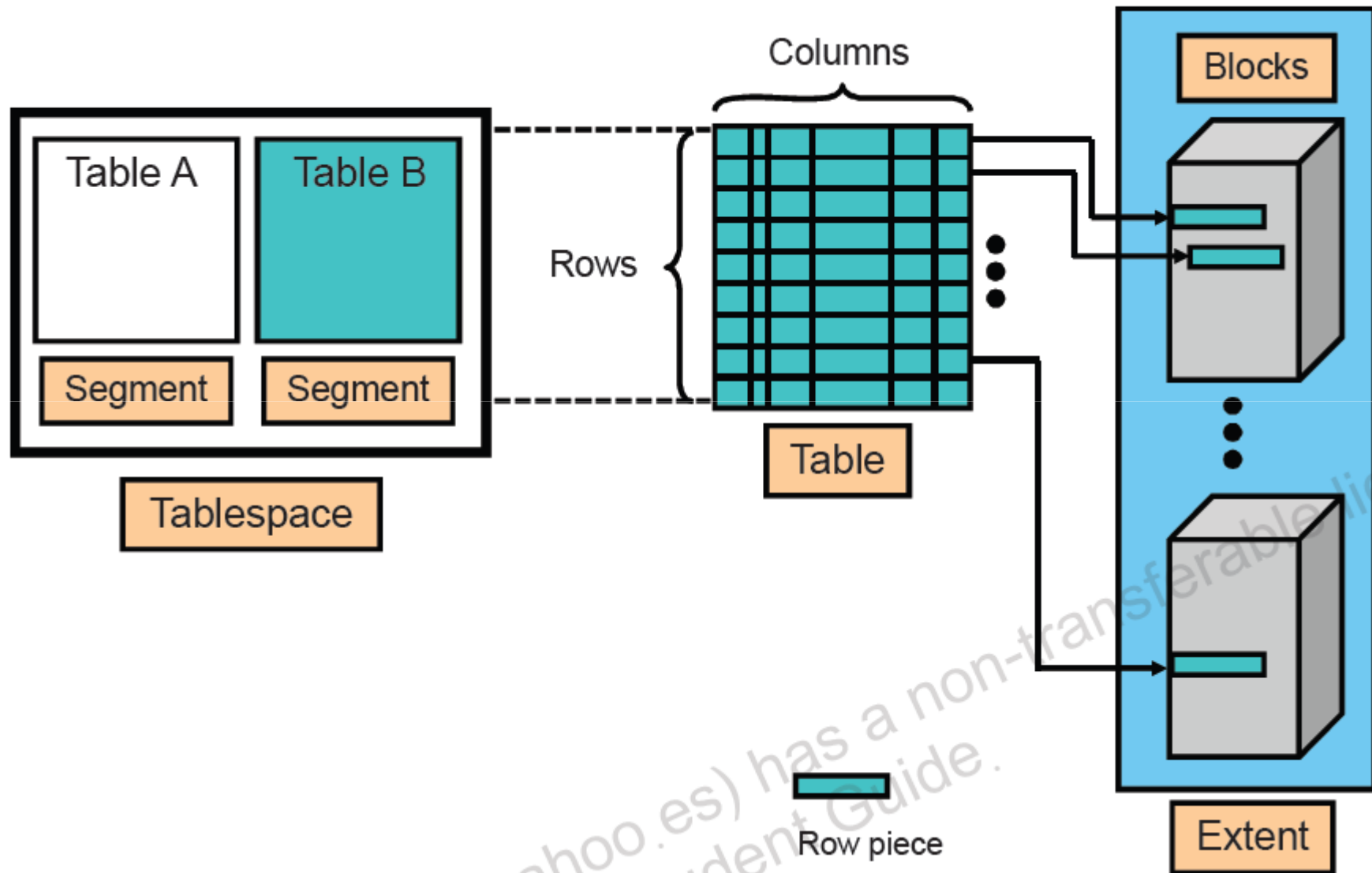
Estructuras Lógicas y Físicas

1. Las unidades de asignación de espacio en la base de datos son los bloques de datos, extents y segmentos.
 2. El nivel más fino de granularidad en que Oracle almacena los datos es el bloque de datos (también llamados bloques lógicos, bloques de Oracle o páginas).
 3. Un bloque de datos corresponde a un número de bytes físicos de espacio de la base de datos en disco
 4. Un segmento es un conjunto de extents, cada uno de los cuales ha sido asignado a una estructura de datos específicas, pertenecientes a un *tablespace*.
-

Estructuras Lógicas y Físicas

- Un Tablespace puede agrupar lógicamente los datos; ejemplo Tablespace Contabilidad, Tablespace RRHH.
 - Los segmentos de un tablespace pueden ser:
 - Datos (tablas)
 - Índice
 - Rollback
 - Temporales (para procesar instrucciones SQL)
 - A medida que se actualicen las tablas, los extents pueden aumentar o disminuir de número. Se establece un número mínimo de extents, y número de extents a ser agregados.
 - Existe siempre el tablespace SYSTEM, el cual almacena datos del diccionario.
-

Almacenamiento de Tablas

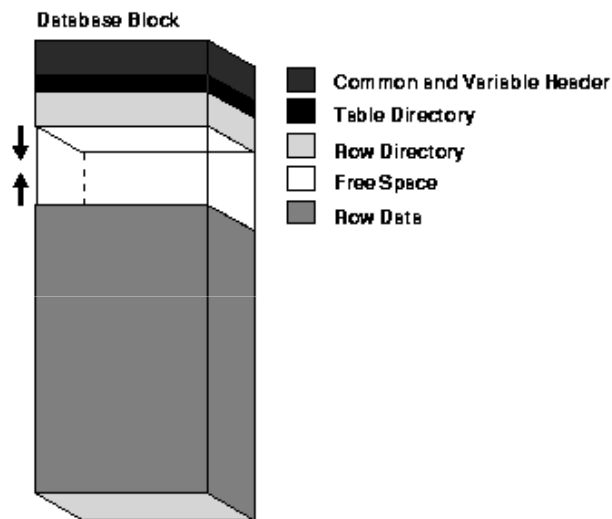


Bloque de Datos

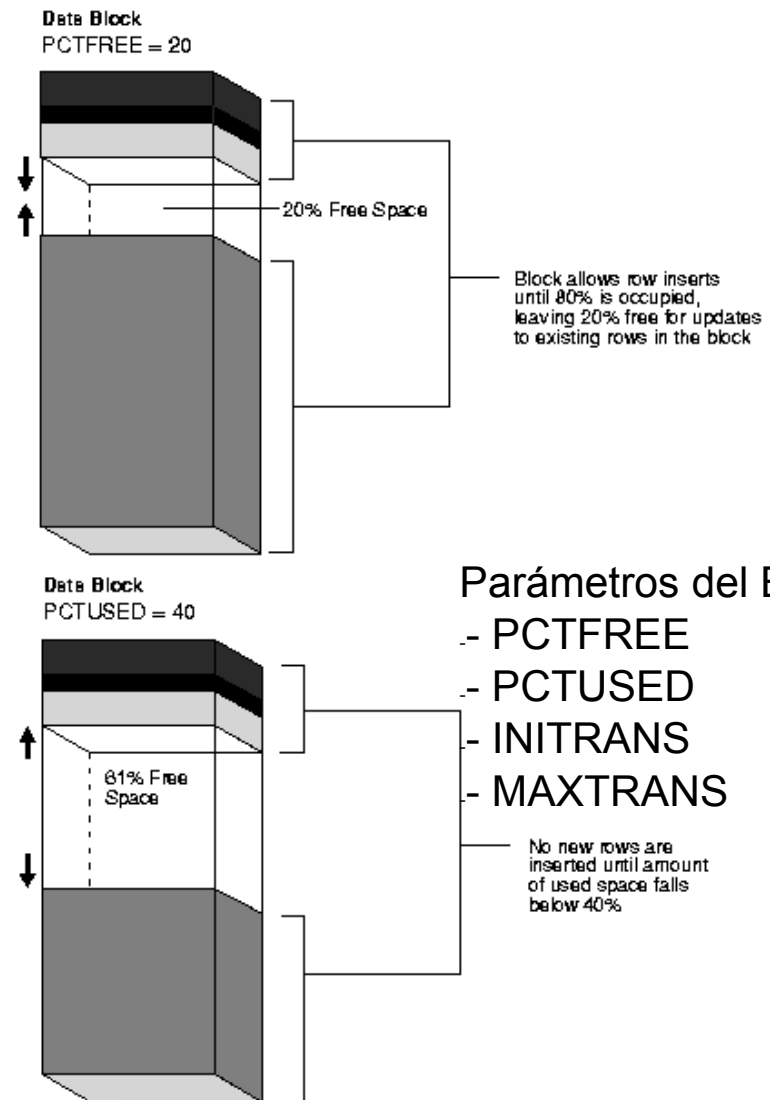
- Oracle maneja el espacio de almacenamiento en *datafiles de una base de datos en unidades* llamadas bloques de datos.
 - Un bloque de datos es la unidad más pequeña de datos usada por una base de datos.
 - Oracle solicita los datos en múltiplos de los bloques de datos de Oracle y no del sistema operativo.
 - El tamaño estándar del bloque se puede conocer consultando el parámetro DB_BLOCK_SIZE.
 - Los tamaños de los bloques de datos deben ser un múltiplo de los tamaños de bloques del sistema operativo, entonado para evitar I/Os innecesarios.
-

Bloque de Datos

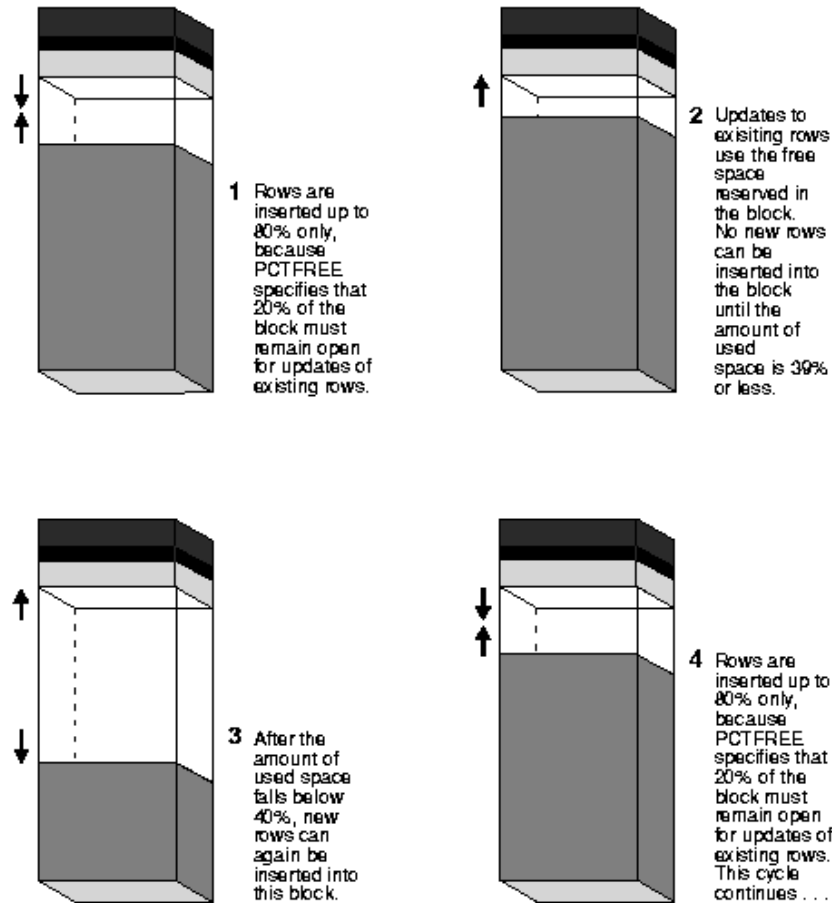
Estructura del bloque



Manejo del Espacio libre



Interacción PCTFREE y PCTUSED



PCTUSED y PCTFREE

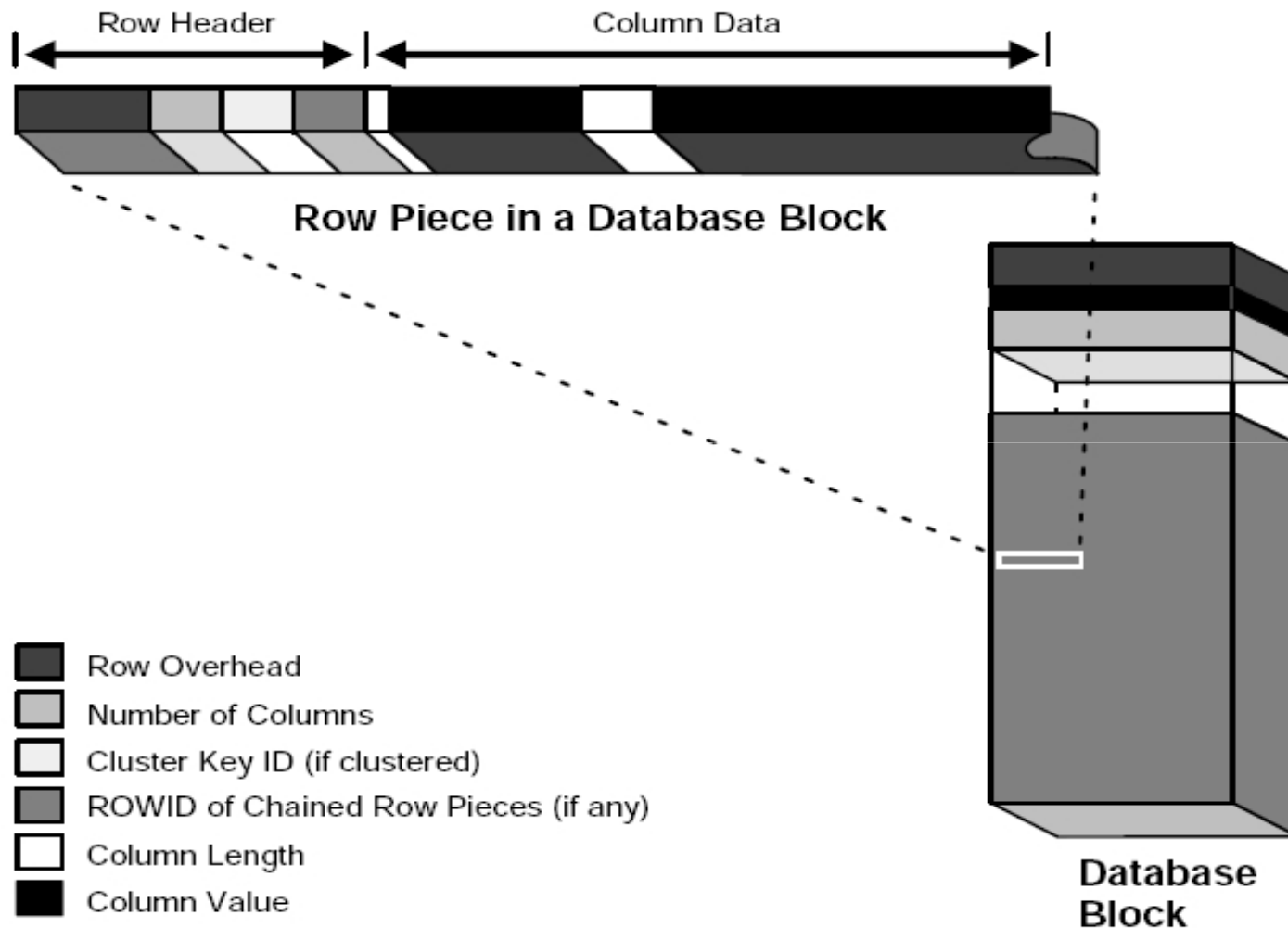
Un PCTFREE alto permite más actualizaciones en un bloque y el bloque puede acomodar menos filas.

Coloque un valor alto si la tabla contiene:

- Columnas que son inicialmente NULL y luego actualizadas.
- Columnas que pueden aumentar de tamaño cuando se actualizan.

El PCTUSED se configura para que el bloque retorne a la lista de 'libres' cuando hay suficiente espacio para acomodar una tupla promedio.

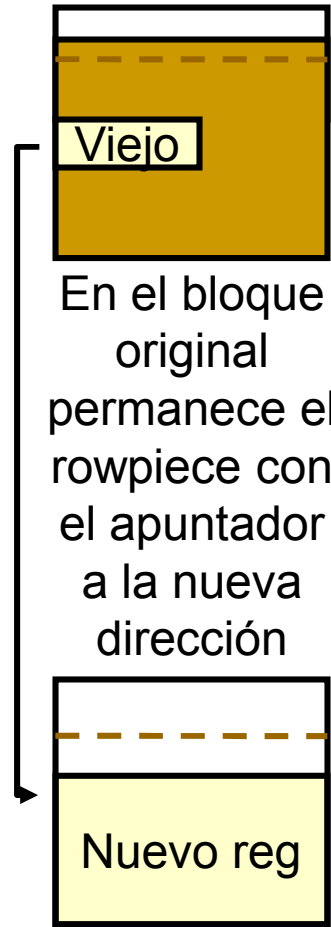
Formato de un Row Piece



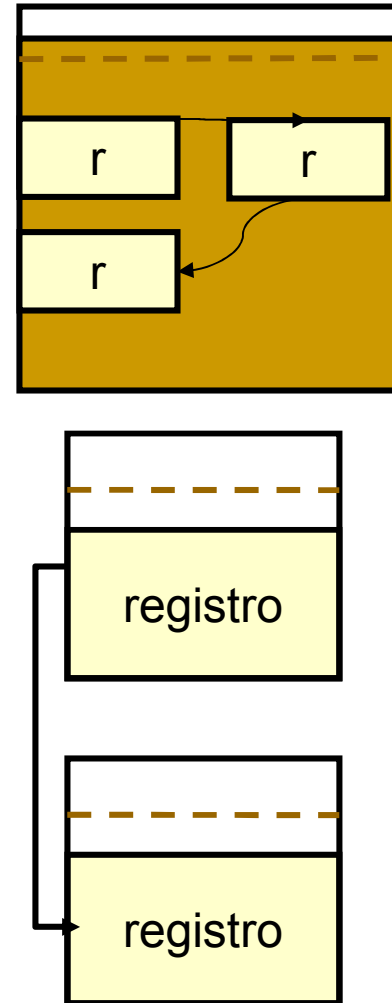
Encadenamiento y Migración

- ***Encadenamiento (chaining)***: la fila es demasiado larga para ser almacenada en un bloque. Oracle almacena la fila en una cadena de uno o más bloques.
- ***Migración***: si un UPDATE aumenta la cantidad de espacio ocupado por la fila, de forma que la misma no pueda ser almacenada en un bloque de datos.

Migración



Encadenamiento



Cuando un registro es encadenado o migrado, el desempeño de I/O asociado decrece porque se debe buscar más de un bloque de datos para devolver la información del registro

Encadenamiento y Migración

- Oracle trata de encontrar otro bloque con suficiente espacio para almacenar la fila completa. Si el bloque existe, se mueve la fila completa. Si no, Oracle separa la fila en varios “row pieces”, se mueve el (los) pedazo(s) que pueda(n) ser almacenado(s) y se realiza **encadenamiento**.
- Oracle mantiene en el bloque original de una fila migrada, el apuntador a un nuevo bloque que contiene la fila actual; el ROWID de una fila migrada no cambia. Los índices no se actualizan, ellos apuntan a la localización original del registro.

Revisando Encadenamiento y Migración

!more \$ORACLE_HOME/rdbms/admin/utlchn1.sql

```
CREATE TABLE CHAINED_ROWS(  
  OWNER_NAME          VARCHAR2(30),  
  TABLE_NAME         VARCHAR(30),  
  CLUSTER_NAME        VARCHAR(30),  
  PARTITION_NAME      VARCHAR(30),  
  HEAD_ROWID          ROWID,  
  ANALYZE_TIMESTAMP   DATE  
);
```

```
ANALYZE TABLE TABLE_X LIST CHAINED_ROWS;
```

```
SELECT COUNT(*)  
FROM CHAINED_ROWS  
WHERE OWNER_NAME='OWNER' AND  
      TABLE_NAME='TABLE_X';
```

Eliminando Encadenamiento y Migración

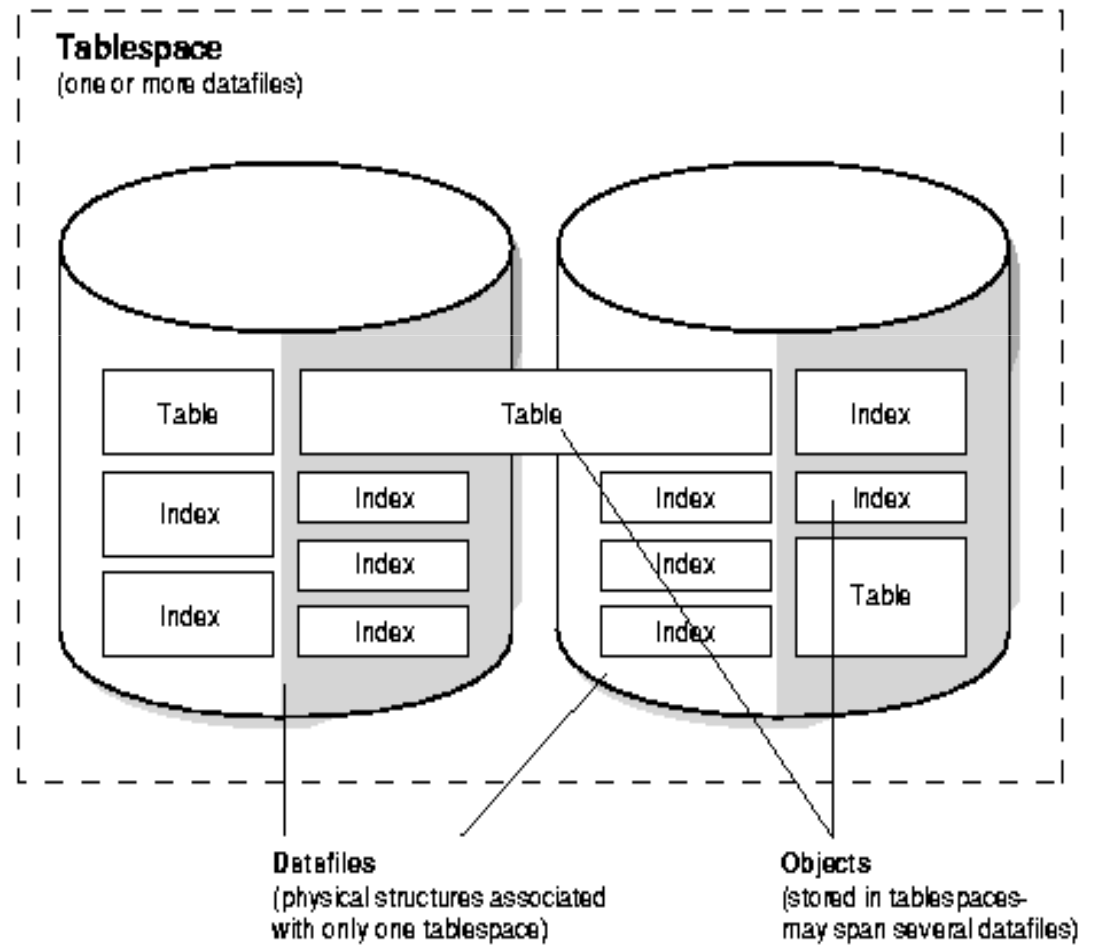
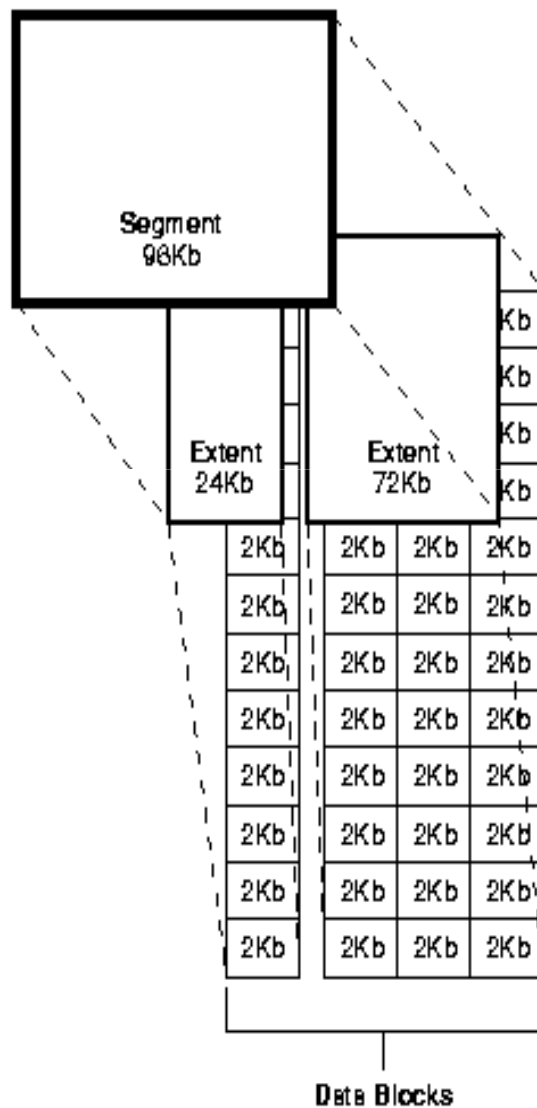
```
CREATE TABLE TABLE_NAME_TEMP AS  
SELECT *  
FROM OWNER.TABLE_X  
WHERE ROWID IN (SELECT HEAD_ROWID  
                 FROM CHAINED_ROWS  
                 WHERE OWNER_NAME='OWNER' AND  
                       TABLE_NAME='TABLE_X');
```

```
DELETE FROM OWNER.TABLE_X  
WHERE ROWID IN (SELECT HEAD_ROWID  
                 FROM CHAINED_ROWS  
                 WHERE OWNER_NAME='OWNER' AND  
                       TABLE_NAME='TABLE_X');
```

```
INSERT INTO OWNER.TABLE_X  
SELECT * FROM TABLE_NAME_TEMP;
```

```
DROP TABLE TABLE_NAME_TEMP;  
DELETE FROM CHAINED_ROWS;  
ANALYZE TABLE OWNER.TABLE_X LIST CHAINED_ROWS;
```

Estructuras Lógicas

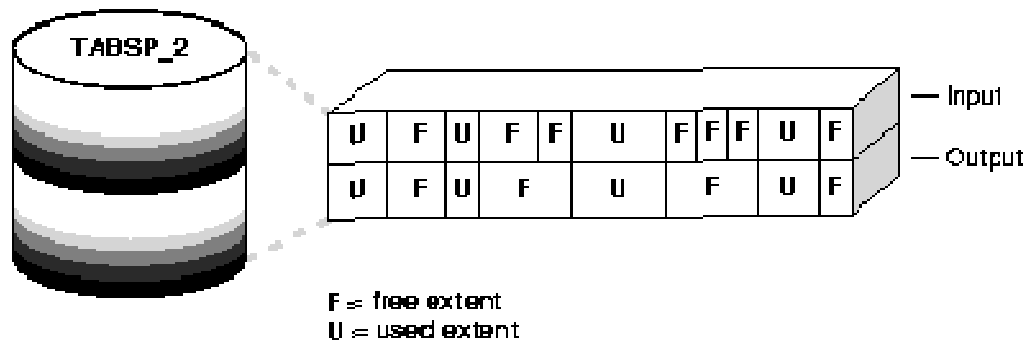


Gestión del Espacio

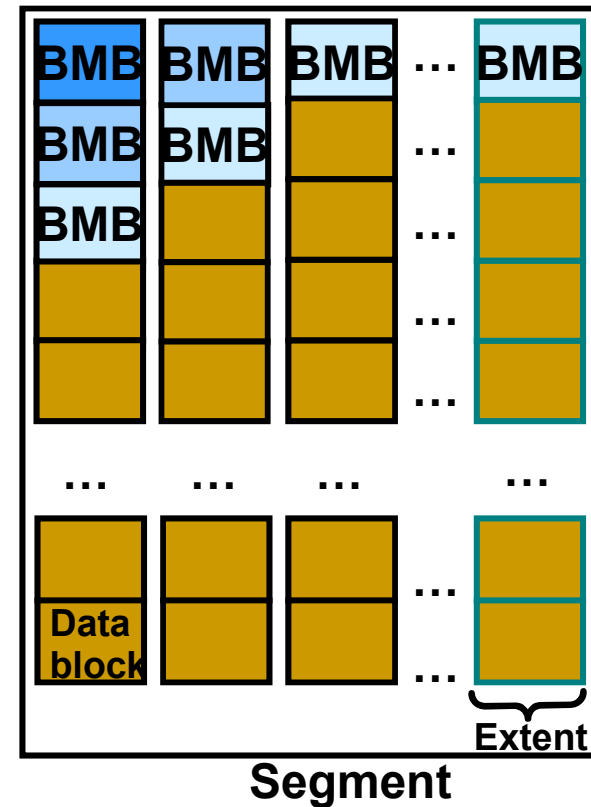
- ¿Cómo se gestionan los extents en el tablespace?
 - Diccionario de datos
 - Localmente
 - ¿Cómo se gestionan los segmentos en tablespaces manejados localmente?
 - Manual
 - Automática
 - ¿Cómo se asignan los extents?
 - Manual
 - Uniforme
 - Autoallocate
 - ¿Cómo se administran los datafiles?
 - Fijo
 - Extensible
-

¿Cómo se manejan los extents en el tablespace?

Diccionario de Datos

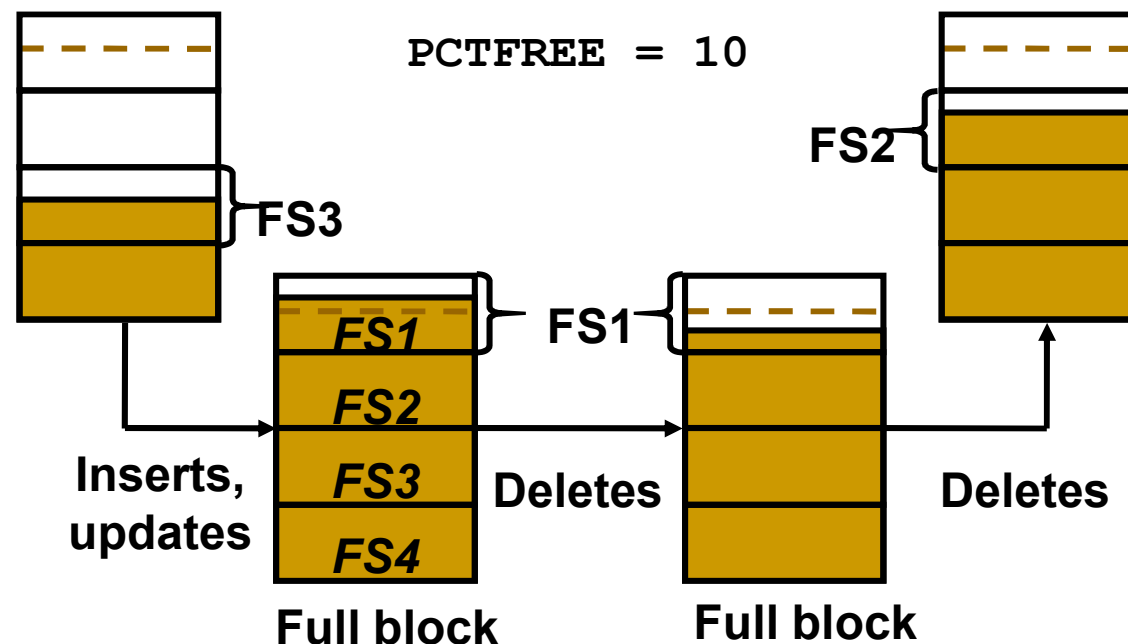


Localmente (freelist o bitmap local)



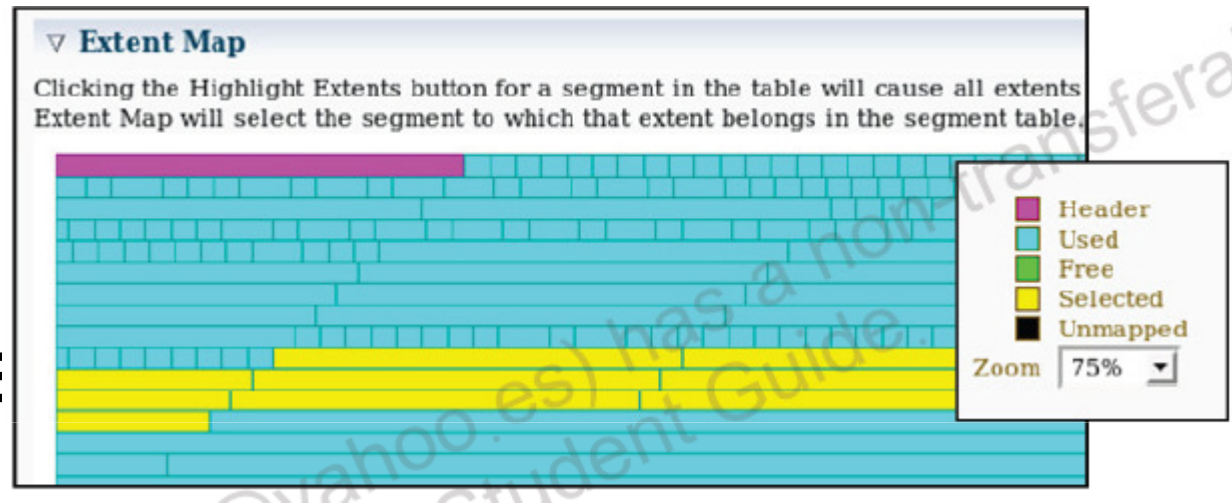
¿Cómo se manejan los segmentos en tablespaces manejados localmente?

- Manual (PCTFREE, PCTUSE, FREELIST, FREELIST GROUPS).
- Automática



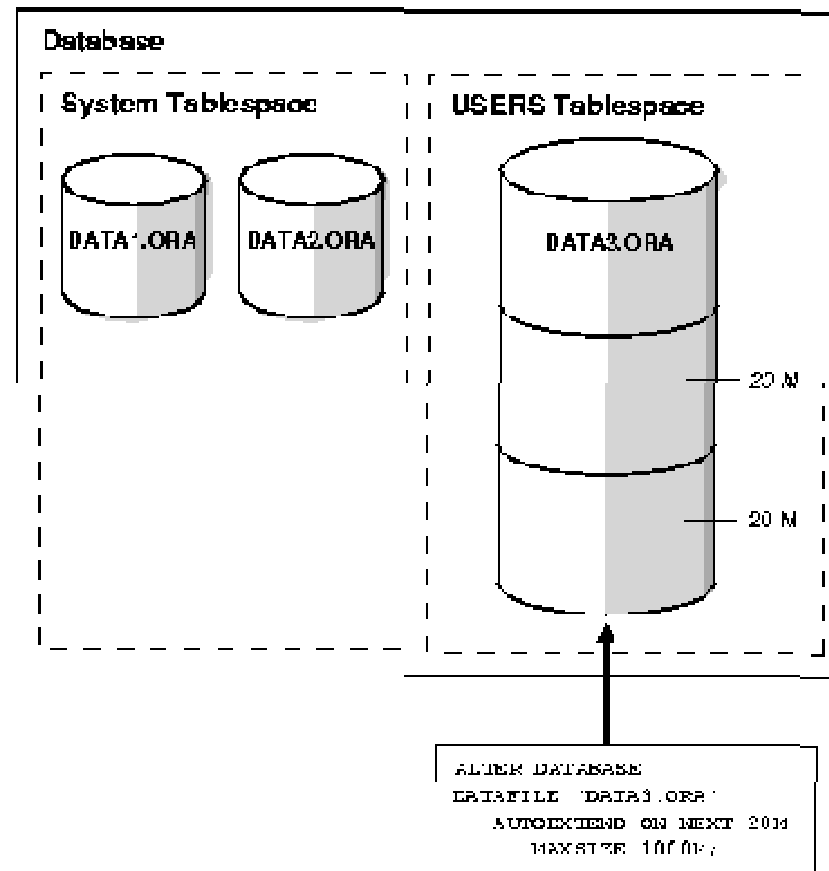
¿Cómo se asignan los extents?

- **Manual**
 - ❑ **INITIAL**
 - ❑ **NEXT**
 - ❑ **PCTINCREASE**
 - ❑ **MINEXTENTS**
- **Uniform**
- **Autoallocate**



¿Cómo se manejan los datafiles?

- Tamaño Fijo
- Extensibles



```
ALTER DATABASE DATAFILE  
'/u02/oracle/rbdb1/users03.dbf' AUTOEXTEND  
OFF;
```

```
ALTER DATABASE DATAFILE  
'/u02/oracle/rbdb1/users03.dbf' RESIZE  
100M;
```

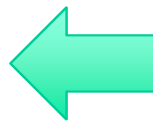
Laminas de explicación

Estructura de un Bloque Oracle

- **Encabezado:** información general del bloque, como la dirección del bloque y el tipo de segmento (datos, índices).
 - **Directorio de tablas:** información acerca de las tablas que contienen tuplas en el bloque
 - **Directorio de Registros:** información de las tuplas actuales en el bloque (incluyendo direcciones para cada row piece “trozo de registro” en el área de datos). Después que el espacio ha sido asignado en el directorio de registros de un bloque de datos, éste no es reclamado cuando el registro es borrado. Oracle reutiliza el espacio sólo cuando se insertan nuevos registros en el bloque.
 - **Overhead:** El encabezado del bloque de datos, directorio de tablas y directorio de registros
 - **Datos de los registros:** Contiene la tabla o datos de índices. Los registros puede atravesar bloques.
-

Estructura de un Bloque Oracle

- **Espacio libre:** es asignado para la inserción de nuevos registros y para la actualización de registros que requieran espacio adicional
- En los bloques de datos asignados para segmentos de datos de una tabla o clúster o para los segmentos de un índice, el espacio libre puede almacenar registros de transacciones.
- Un registro de una transacción es requerido en un bloque para cada instrucción de INSERT, UPDATE, DELETE y SELECT FOR UPDATE que accede uno o más registros en el bloque.
- En la mayoría de los SO un registro de una transacción requiere 23 bytes.



Manejo del espacio libre

- El espacio libre en un segmento se maneja mediante el uso de bitmaps en lugar de listas.
- El espacio liberado por una transacción de DELETE y UPDATE puede utilizarse para nuevas instrucciones de INSERT si:
 - La instrucción de INSERT es de la misma transacción y aparece luego de la instrucción que libera espacio, esta instrucción puede usar el espacio libre inmediatamente.
 - La instrucción de INSERT esta en otra transacción, no pertenece a aquella que liberó el espacio, por lo que debe esperar por una instrucción de commit de aquella que libera el espacio para poder utilizarlo.
- El espacio liberado puede o no ser contiguo con el área de espacio libre en el bloque. Oracle fusiona los espacios libres sólo cuando: una instrucción de INSERT o UPDATE intenta usar un bloque que contiene suficiente espacio libre para contener un nuevo trozo de un registro y el espacio libre esta fragmentado



Parámetros del bloque

- PCTFREE: especifica el mínimo porcentaje de un bloque de datos para ser reservado como espacio libre para posibles actualizaciones de registros que han sido insertados en el bloque
- PCTUSED: especifica el porcentaje mínimo de un bloque que puede ser usado para un registro más el overhead antes de que nuevos registros sean añadidos al bloque
- INITTRANS y MAXTRANS permiten especificar el número inicial y máximo de “slots” para transacciones concurrentes que se crearán en el bloque que almacene índices o datos. Los “slots” de transacciones se utilizan para almacenar la información de las transacciones que están haciendo cambios sobre el bloque de datos en un instante dado. Una transacción puede utilizar un “slot” si está cambiando mas de una fila o entrada de un índice
- INITTRANS, cuyo valor por defecto es 1 para los segmentos de datos y 2 para los segmentos de índices, garantiza un mínimo nivel de concurrencia
- MAXTRANS, cuyo valor por defecto es de 255, delimita el número de transacciones que pueden hacer cambios a los datos en un bloque de datos o índices



Formato de un Row Piece

Row Header:

1. Overhead: Indica que comienza un *row piece*.
2. Número de columnas en el *row piece*
3. Columna clave: Cuales columnas son las claves del cluster.
4. ROWID: ROWID del siguiente row piece (para encadenamiento).

Una columna NULL sólo almacena el tamaño 0.

Si las últimas K columnas de una fila poseen el valor NULL, no almacena el tamaño de la columna

Formato de un Row Piece

Un bloque de datos usualmente contiene sólo un *row piece* por cada registro.

Cuando Oracle debe almacenar un registro en más de un *row piece*, éste es encadenado a través de múltiples bloques.

Cuando una tabla tiene más de 255 columnas, los registros que tienen datos después de la columna 255 se encadenan pero dentro del mismo bloque (*intrablock chaining*).

Las piezas de un registro encadenado son encadenadas usando los *rowids* de las piezas.

ROWIDs

El ROWID identifica cada *row piece* por su localización o dirección. Toda fila de una tabla posee un único ROWID que permite ubicar el primer *row piece* de la fila.

Características:

- No se almacenan físicamente como una columna en la tabla.
- Puede ser utilizada para localizar una fila ya que indica su ubicación dentro de la abstracción física que maneja Oracle.
- Proveen un mecanismo de acceso más rápido para las filas de una tabla.
- Se utilizan en las estructuras de índices.



PCTUSED y PCTFREE

PCTFREE: valor entre 0 y 99. El valor de 0 permite que el bloque entero se llene con inserciones de tuplas nuevas.

- Valor por defecto 10.
- Si no existen actualizaciones, asignar 0.
- En cualquier otro caso:
$$\text{PCTFREE} = 100 * \text{tam_prom_tupla} / (\text{tam_prom_tupla} + \text{tam_inic_tupla})$$

- PCTUSED: valor entre 1 y 99. No es un parámetro configurable para tablas organizadas por índice.

- Valor por defecto 40.
- Se especifica si se eliminan filas.
- $$\text{PCTUSED} = 100 - \text{PCTFREE} - 100 * \text{tam_prom_tupla} / \text{blocksize}$$

Donde:

tam_prom_tupla: tamaño promedio de una fila

tam_inic_tupla: tamaño promedio de una fila al insertarse.

La suma del PCTFREE y PCTUSED debe ser menor a 100.

