

INFS602 Physical Database Design

Storage Management in Oracle



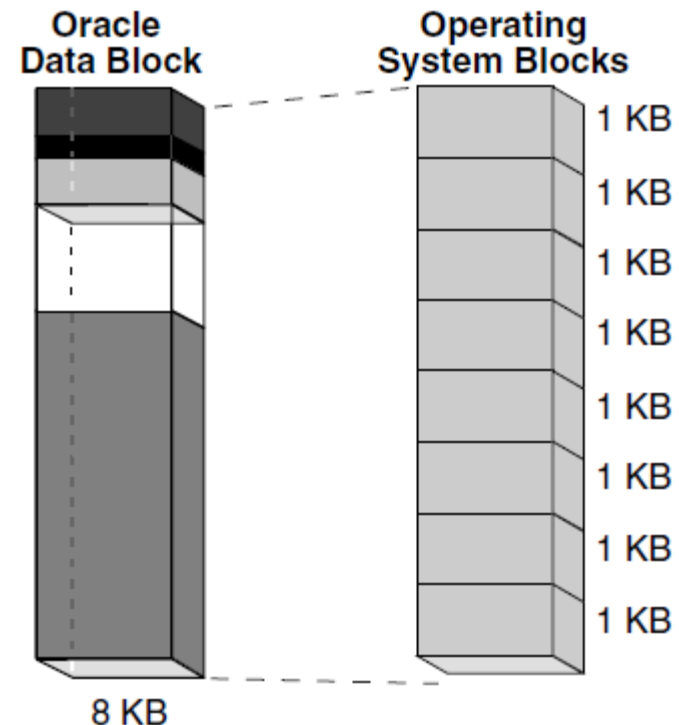
Learning Outcomes

- Understand how the table parameters PCTFREE and PCTUSED work together
- Differentiate between block level and table level parameters
- Estimate the space required for storing tables in Oracle

Data Blocks VS OS Blocks

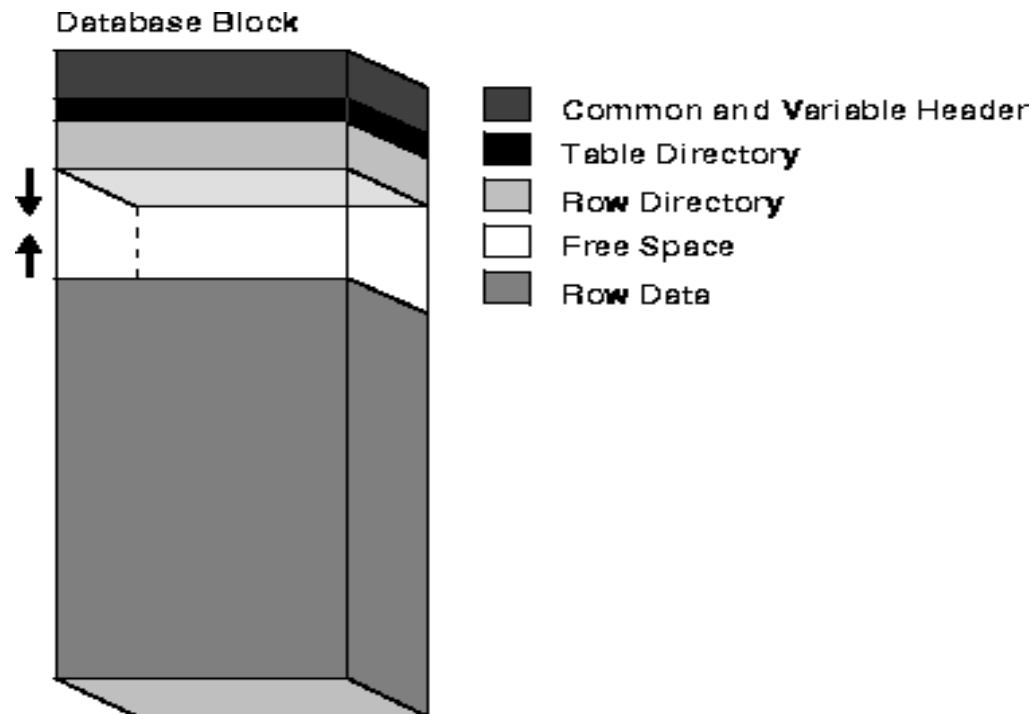
At the physical level, database data is stored in disk files made up of operating system blocks.

The database requests data in multiples of data blocks, not operating system blocks.



Data Block Organization

- Data blocks are the smallest unit of storage that Oracle allocates



Data Block Components

- Oracle Database uses the **block overhead** to manage the block itself
- Some parts of the block overhead are fixed in size, but the total size is variable. On average, the block overhead totals 84 to 107 bytes.
 - **Header** contains general information about the block, such as the block address and the type of segment
 - **Table Directory** identifies the tables having rows in this block.
 - **Row Directory** contains the *rowids* – i.e. the addresses of individual rows

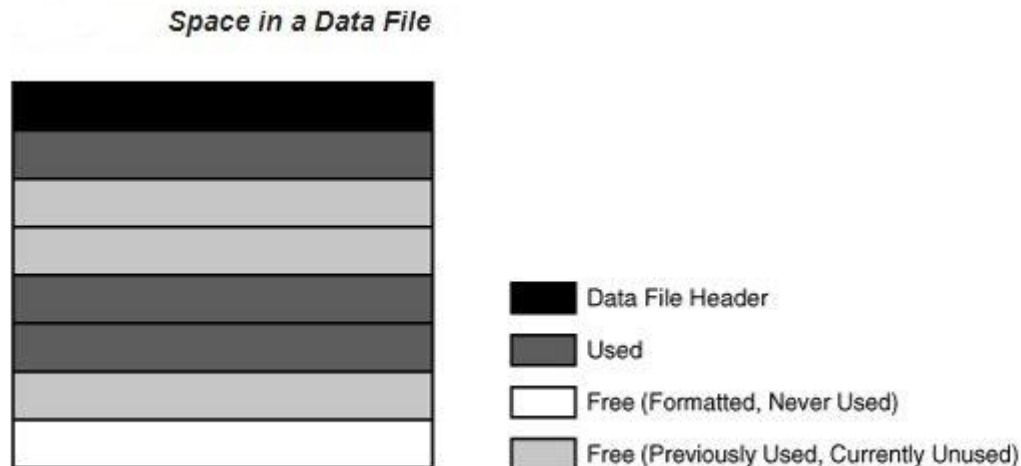
Data Block Structure

Free space in a block is used for:

1. Updates to existing rows – text (varchar2) fields expanding in size
2. Insertion of new rows
3. Transaction entries – these are typically used to store information on updates for recovery purposes

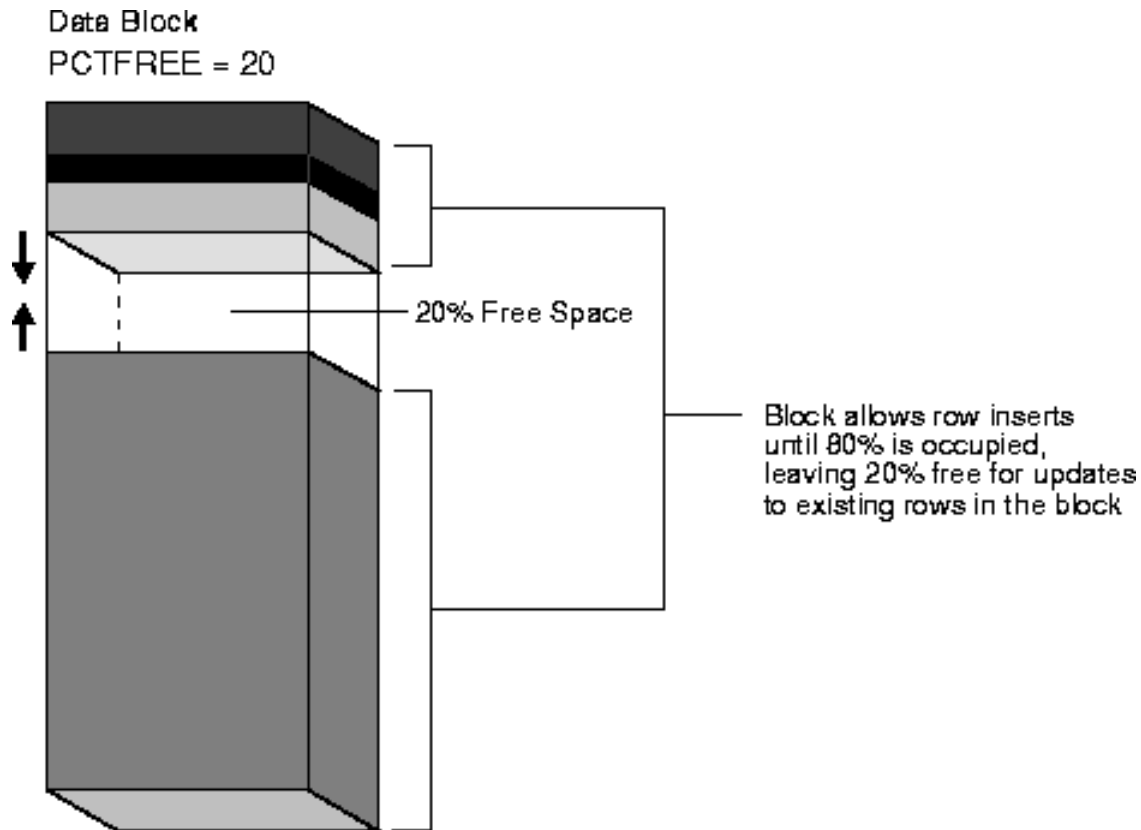
Space Management in Data Blocks

- As the database fills a data block from the bottom up, the amount of **free space** between the row data and the block header decreases.
- The database manages free space in the data block to **optimize performance** and **avoid wasted space**.



Data Block Parameters: PCTFREE

- `CREATE TABLE test_table (n NUMBER) PCTFREE 20;`



- Q: How this settings affects space management?

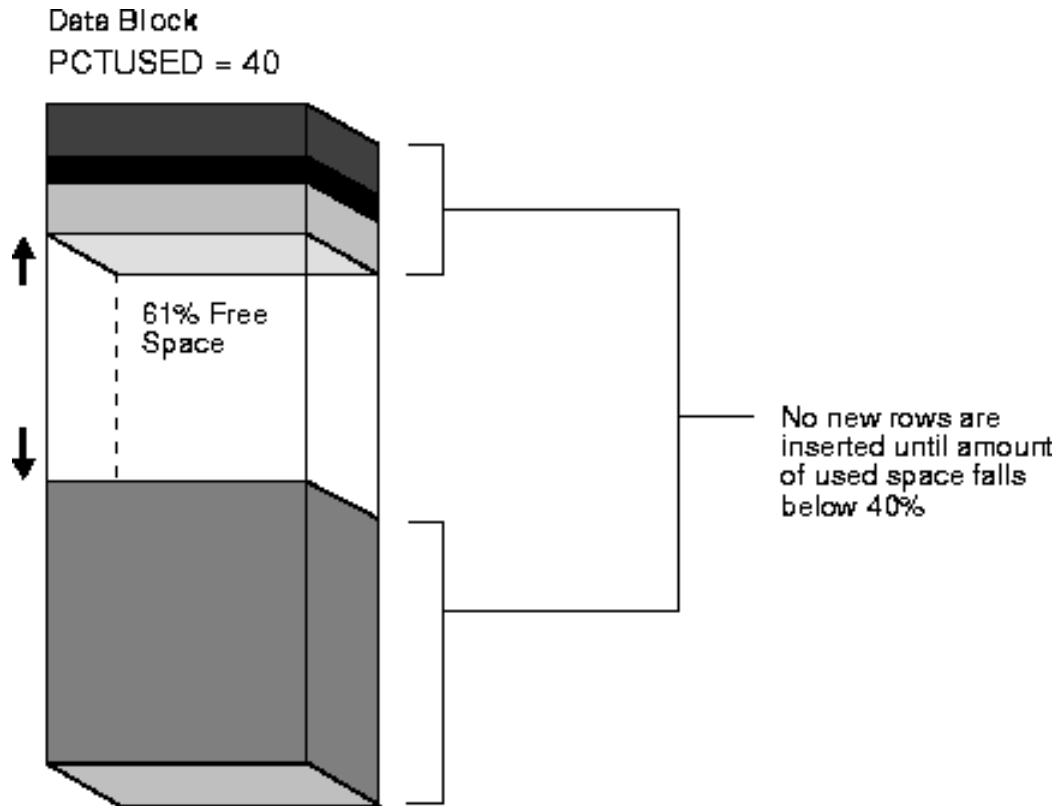
PCTFREE

- The PCTFREE parameter sets the minimum percentage of a data block to be reserved as free space for possible updates to rows that already exist in that block.

For example: PCTFREE 20

- 20% of each data block in this table's data segment is to be kept free and available for possible updates to the existing rows already within each block.

PCTUSED



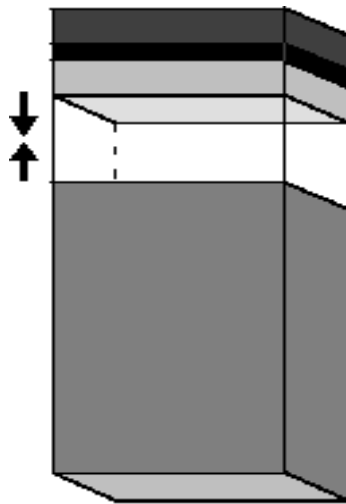
- sets the percentage of free space that must exist in a currently used block for the database to put it on the free list



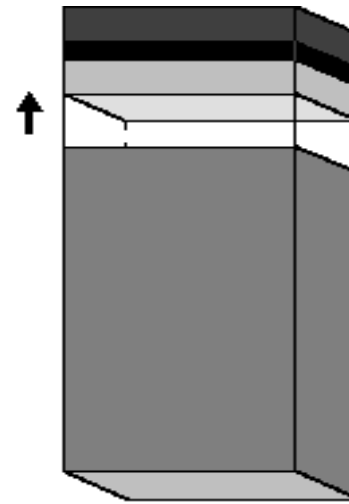
PCTUSED

- This controls when new rows will be added to a data block
- After a data block is filled no new rows can be added until free space falls below PCTUSED
- Thus PCTFREE and PCTUSED specify high and low water marks respectively

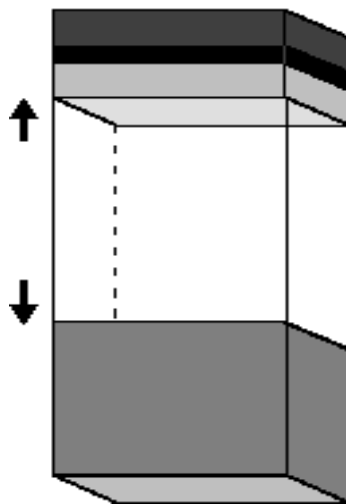
PCTFREE and PCTUSED



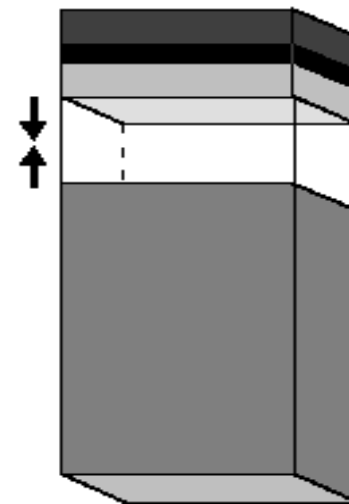
1 Rows are inserted up to 80% only, because PCTFREE specifies that 20% of the block must remain open for updates of existing rows.




2 Updates to existing rows use the free space reserved in the block. No new rows can be inserted into the block until the amount of used space is 39% or less.



3 After the amount of used space falls below 40%, new rows can again be inserted into this block.



4 Rows are inserted up to 80% only, because PCTFREE specifies that 20% of the block must remain open for updates of existing rows. This cycle continues...

- 
- For rows that are too large to fit into a single block the following situations may happen:
 - row is too large to fit into one data block when it is first inserted.
 - A row that originally fit into one data block is updated so that the overall row length increases, but insufficient free space exists to hold the updated row.
 - row has more than 255 columns.

Row Chaining and Migrating

- **Row chaining** occurs when the row size exceeds the block size ie. the row is too large to fit into one data block when it is first inserted.
 - In this case Oracle stores the data in a chain of data blocks
- **Row migration** occurs when an expansion to an existing row causes the block to run out of free space
 - In this case Oracle migrates the data for the entire row to a new data block, assuming the entire row can fit in a new block
 - uses a pointer to point to the new block containing the migrated row
 - Oracle preserves the rowid of the row in its original data block and



Row Migration

- Thus row migration will reduce performance as more than one data block will need to be scanned
- This problem can be controlled by setting PCTFREE appropriately
- It can be solved by deleting all rows and then re-inserting them into the table

Setting PCTFREE

- PCTFREE can be set to any value between 0 and 99 (default is 10)
- A small PCTFREE value allows:
 1. Less room for updates
 2. More room for inserts
 3. Better space utilization, because the table data is stored in fewer blocks



Block Level Parameters

Setting PCTFREE

- On the other hand, a larger value favours updates over inserts, thus reducing the chances of row migration
- A larger value is suitable for tables with a high rate of update activity

Block Level Parameters - Setting PCTUSED

- Can specify any value (default 40) between 0 and 99, providing that $PCTFREE + PCTUSED \leq 100$
- A smaller value of PCTUSED results in:
 1. Reducing processing costs during UPDATE and DELETE statements
 2. Increasing the unused space in the database
- A larger value has the opposite effects

General Guidelines

- Scenario: Common activity includes UPDATE statements that increase the size of rows
- Settings:
 1. PCTFREE
 2. PCTUSED

Guidelines

- Scenario: Most activity includes INSERT and DELETE statements, and UPDATE statements do not increase the size of rows
- Settings:
 1. PCTFREE
 2. PCTUSED



Guidelines

- Scenario: The table is very large; most activity includes read-only transactions
- Settings:
 1. PCTFREE
 2. PCTUSED

Table Level Parameters

Four main parameters need to be specified:

1. INITIAL – the size in bytes (K or M) of the first extent of a segment (default 5 blocks, minimum 2 blocks)
2. NEXT – the size of each incremental extent (default 5 blocks, minimum 1 block)
3. PCTINCREASE – the percentage by which each incremental extent, after the second (NEXT) extent, grows over the last incremental extent (default 50%, minimum 0%)
4. INITTRANS – specifies the number of transaction entries for which space should be initially reserved in each data block

Precedence in Storage Parameters

- Parameters can be specified at both Table and Tablespace level
- When specified at both levels the table level specification overrides the tablespace specs.
- The order of precedence is:
 1. Alter Table
 2. Create Table
 3. Alter Tablespace
 4. Create Tablespace
 5. Oracle default values

Space Estimation

Calculation of space required for a table requires calculation of:

1. Block header size
2. Available data space per block
3. Space used per row
4. The total number of rows that fit into a data block

Total Space After Headers Size (HSIZE)

Given by the formula:

$$\text{HSIZE} = \text{DB_BLOCK_SIZE} - \text{KCBH} - \text{UB4} - \text{KTBBH} - (\text{INITTRANS} - 1) * \text{KTBIT} - \text{KDBH}$$

where DB_BLOCK_SIZE is the database block size and can be obtained by querying the V\$PARAMETER view

- KCBH, UB4, KTBBH, KTBIT, KDBH are Oracle system constants and can be obtained by querying the V\$TYPE_SIZE view
- INITTRANS is the initial number of transaction entries allocated to the table

Available Data Space (availspace)

$$\text{availspace} = \text{CEIL}(\text{hsize} * (1 - \text{PCTFREE}/100)) - \text{KDBT}$$

- CEIL rounds a fractional result to the next highest integer
- PCTFREE is the space reserved for updates in a data block
- KDBT is a constant whose size you can obtain by selecting the entry from the V\$TYPE_SIZE view

Space used per row (rowspace)

1- calculate the column size, including byte lengths:

$\text{total column size} = \text{column size} + \text{byte length}$

$\text{byte length} = 1, \text{ if column size} < 250; \text{ Else } 3$

2- calculate the row size:

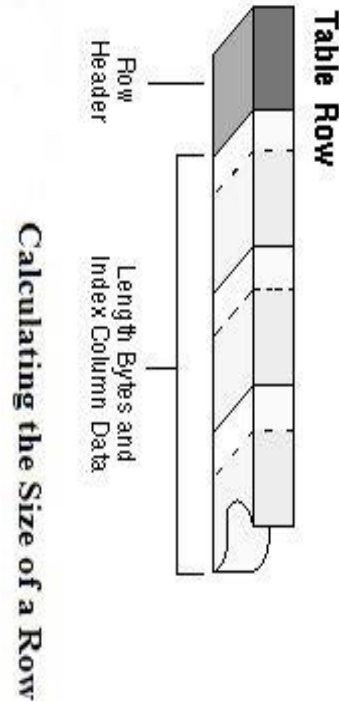
$\text{row size} = \text{row header} + \text{sum of all column sizes}$

where $\text{row header} = 3 * \text{UB1}$

3- calculate the space used per row:

$\text{row space} = \max(\text{row header} + \text{UB4} + \text{SB2}, \text{row size}) + \text{SB2}$

- Values of **UB1**, **UB4** and **SB2** can be obtained from the V\$TYPE_SIZE view



Total Rows per block (RB)

You can calculate the total number of rows that will fit into a data block:

Number of rows in block (RB) = $\text{FLOOR}(\text{availspace} / \text{row space})$

- FLOOR rounds a fractional result to the next lowest integer
- The total table size can now be obtained by working out the total number of blocks required



References

- Oracle 11g Concepts Chapters 2 & 3
- Oracle 10g Administrator's Guide Chapter 8



Review Questions

1. Describe the role of the PCTFREE parameter
2. Describe the role of the PCTUSED parameter
3. Write down the steps involved in estimating the storage space required to store a table in Oracle