

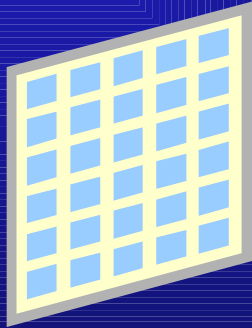
12

Managing Tables

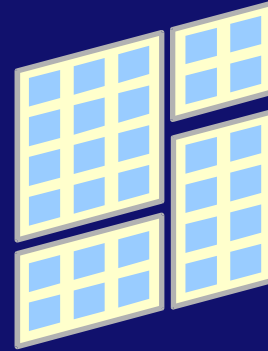
Objectives

- **Distinguishing between different Oracle data types**
- **Creating tables using appropriate storage settings**
- **Controlling the space used by tables**
- **Analyzing tables to check integrity and migration**
- **Retrieving information about tables from the data dictionary**
- **Converting between different formats of ROWID**

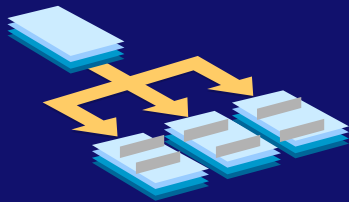
Storing User Data



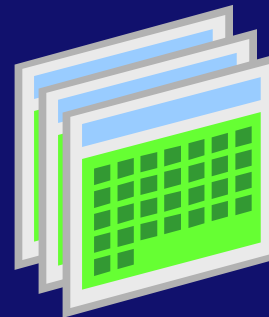
**Regular
table**



**Partitioned
table**

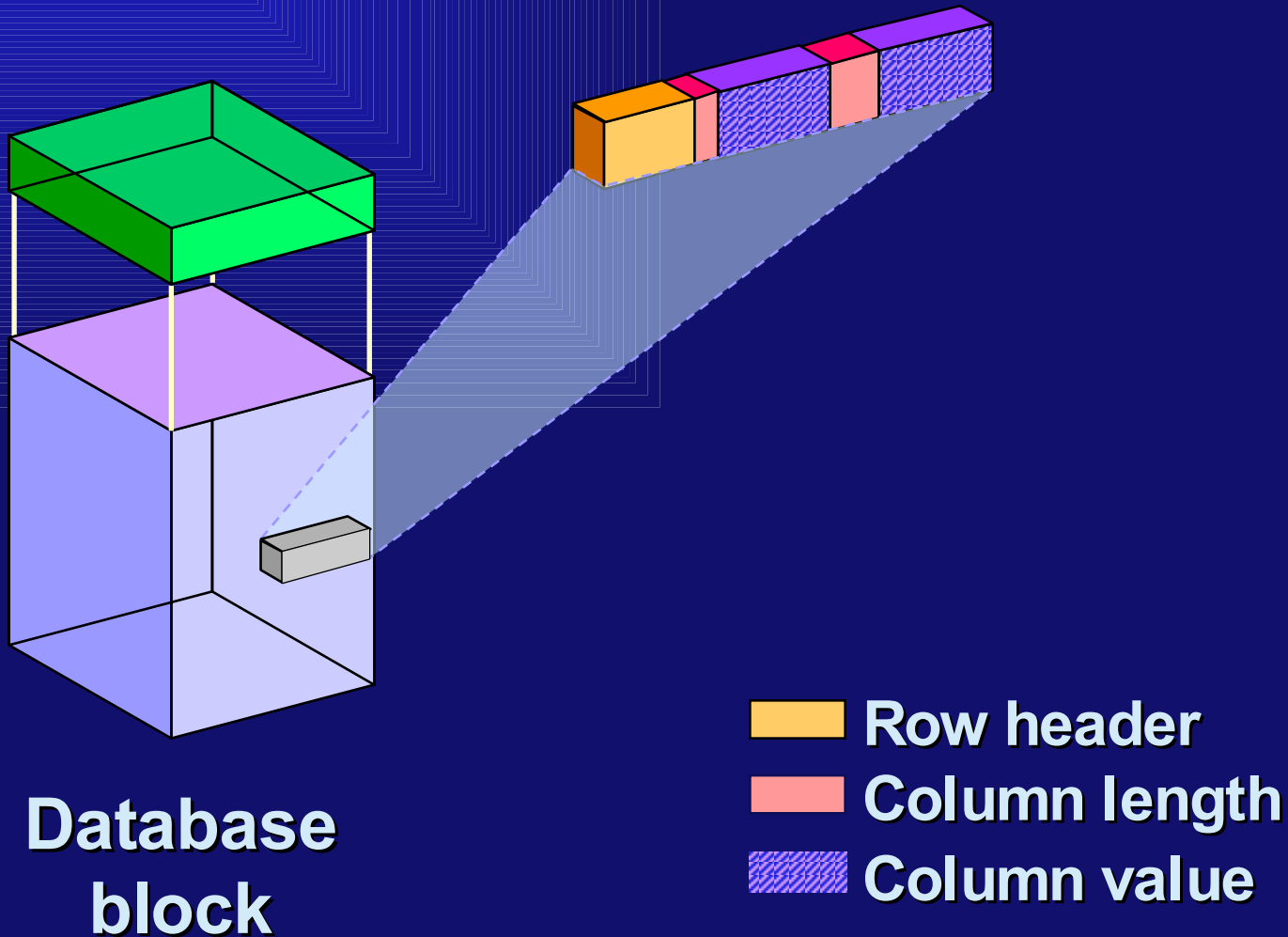


**Index-organized
table**

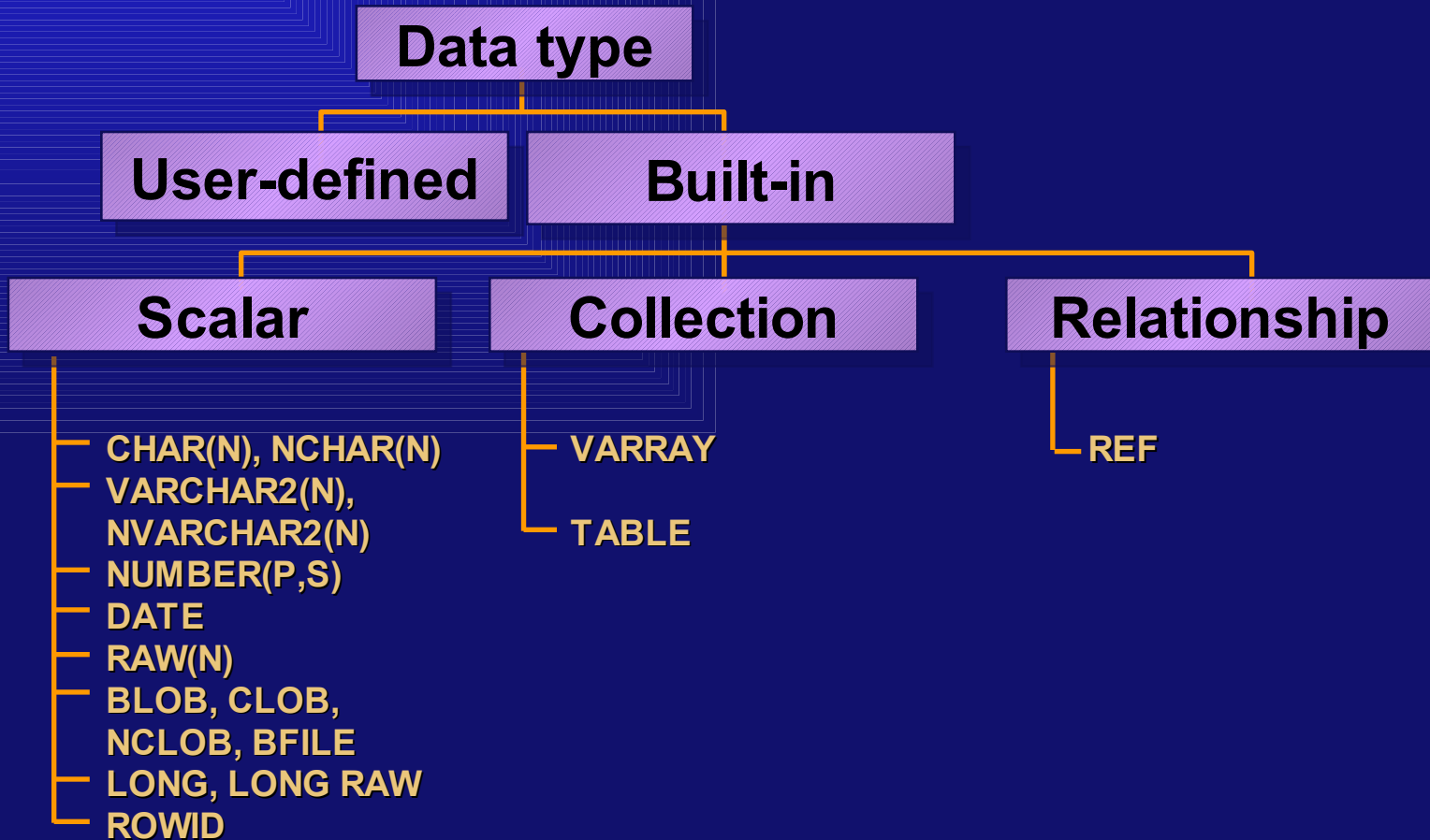


Cluster

Structure of a Row



Oracle Data Types



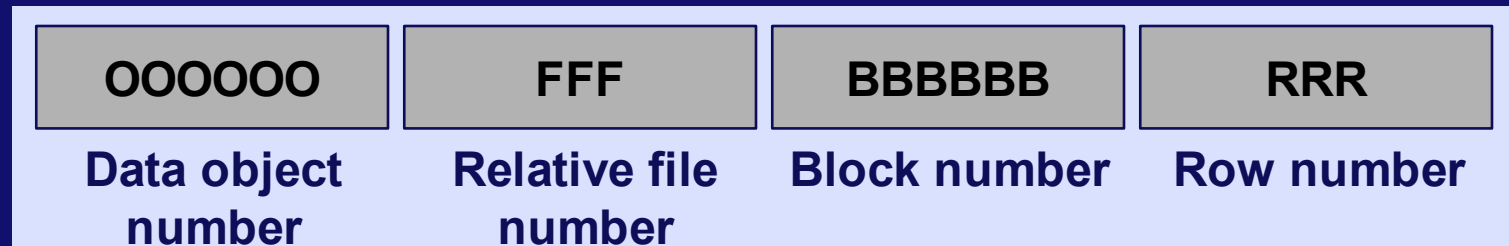
Data Types for Storing Large Objects

LONG, LONG RAW	LOB
Single column per table	Multiple columns per table
Up to 2 gigabytes	Up to 4 gigabytes
SELECT returns data	SELECT returns locator
Data stored in-line	Data stored in-line or out-of-line
No object type support	Supports object types
Sequential access to chunks	Random access to chunks

ROWID Data Type

- **Unique identifier for a row**
- **Used to locate a row**

ROWID Format



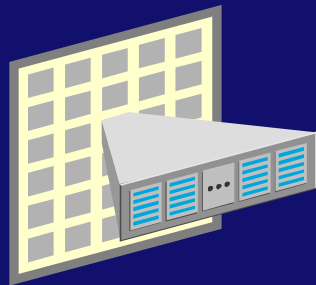
Restricted ROWID

- Can identify rows within a segment
- Needs less space

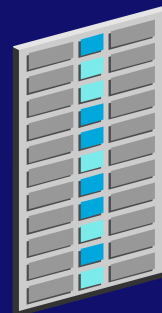


Collections

- **Collections are objects that contain objects.**
- **VARRAYs are ordered sets of elements containing a count and a limit.**
- **Nested tables are tables with a column or variable of the TABLE data type.**



VARRAY



**Nested
table**

Creating a Table

```
CREATE TABLE employees (  
  empno NUMBER(4) ,  
  last_name VARCHAR2(30)  
  deptno NUMBER(2) )  
  PCTFREE 20 PCTUSED 50  
  STORAGE (INITIAL 200K NEXT 200K  
  PCTINCREASE 0 MAXEXTENTS 50)  
  TABLESPACE data01;
```

Creating a Table: Guidelines

- **Use a few standard extent sizes for tables to reduce tablespace fragmentation.**
- **Use the `CACHE` clause for frequently used, small tables.**

Setting PCTFREE and PCTUSED

- **Compute PCTFREE**

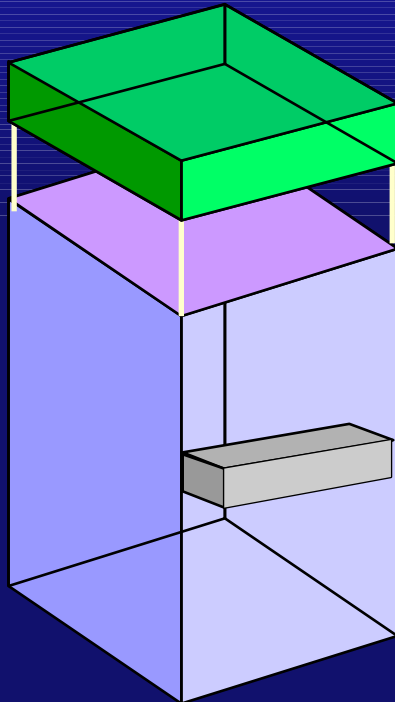
$$\frac{(\text{Average Row Size} - \text{Initial Row Size}) * 100}{\text{Average Row Size}}$$

- **Compute PCTUSED**

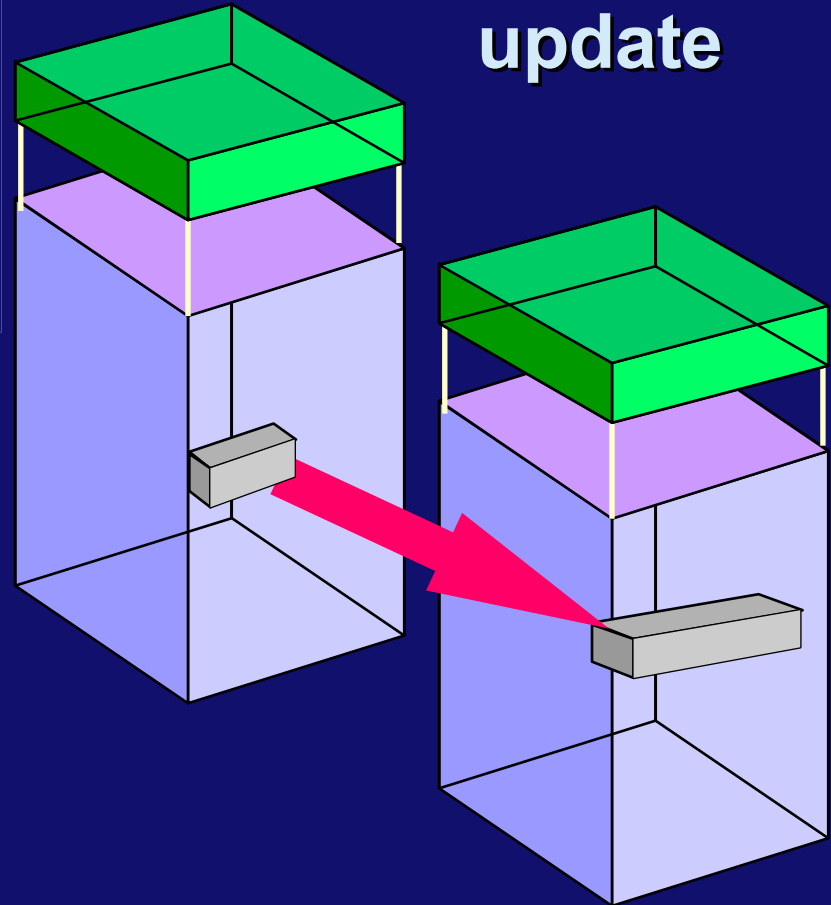
$$100 - \text{PCTFREE} - \frac{\text{Average Row Size} * 100}{\text{Available Data Space}}$$

Row Migration and Chaining

Before
update



After
update



Copying an Existing Table

```
CREATE TABLE new_emp
STORAGE (INITIAL 200K NEXT 200K
PCTINCREASE 0 MAXEXTENTS 50)
NOLOGGING
TABLESPACE data01
AS
SELECT * FROM scott.employees;
```

Changing Storage and Block Utilization Parameters

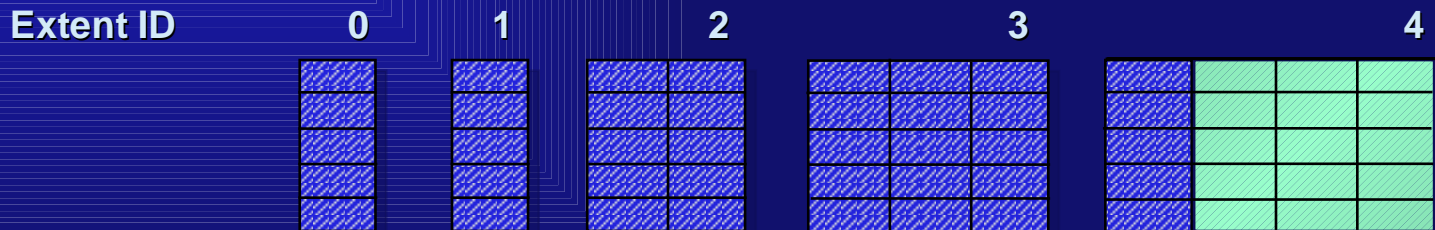
```
ALTER TABLE scott.employees  
PCTFREE 30  
PCTUSED 50  
STORAGE (NEXT 500K  
MINEXTENTS 2  
MAXEXTENTS 100) ;
```

Manually Allocating Extents

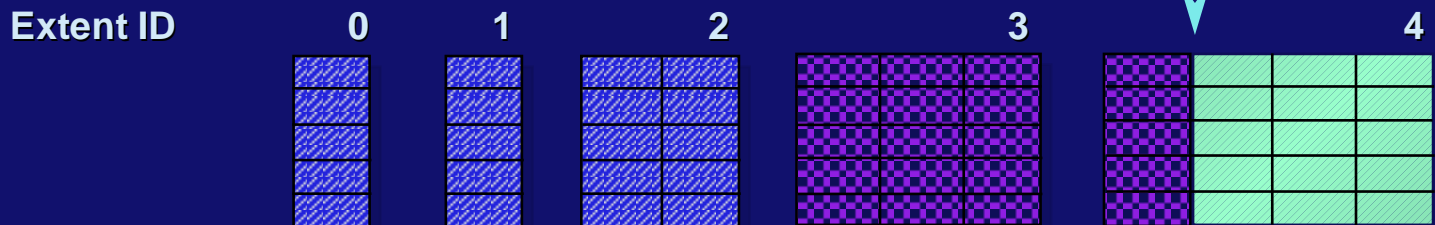
```
ALTER TABLE scott.employees  
ALLOCATE EXTENT (SIZE 500K  
DATAFILE '/DISK3/DATA01.DBF' );
```


High Water Mark

After inserts



After deletes



Used block

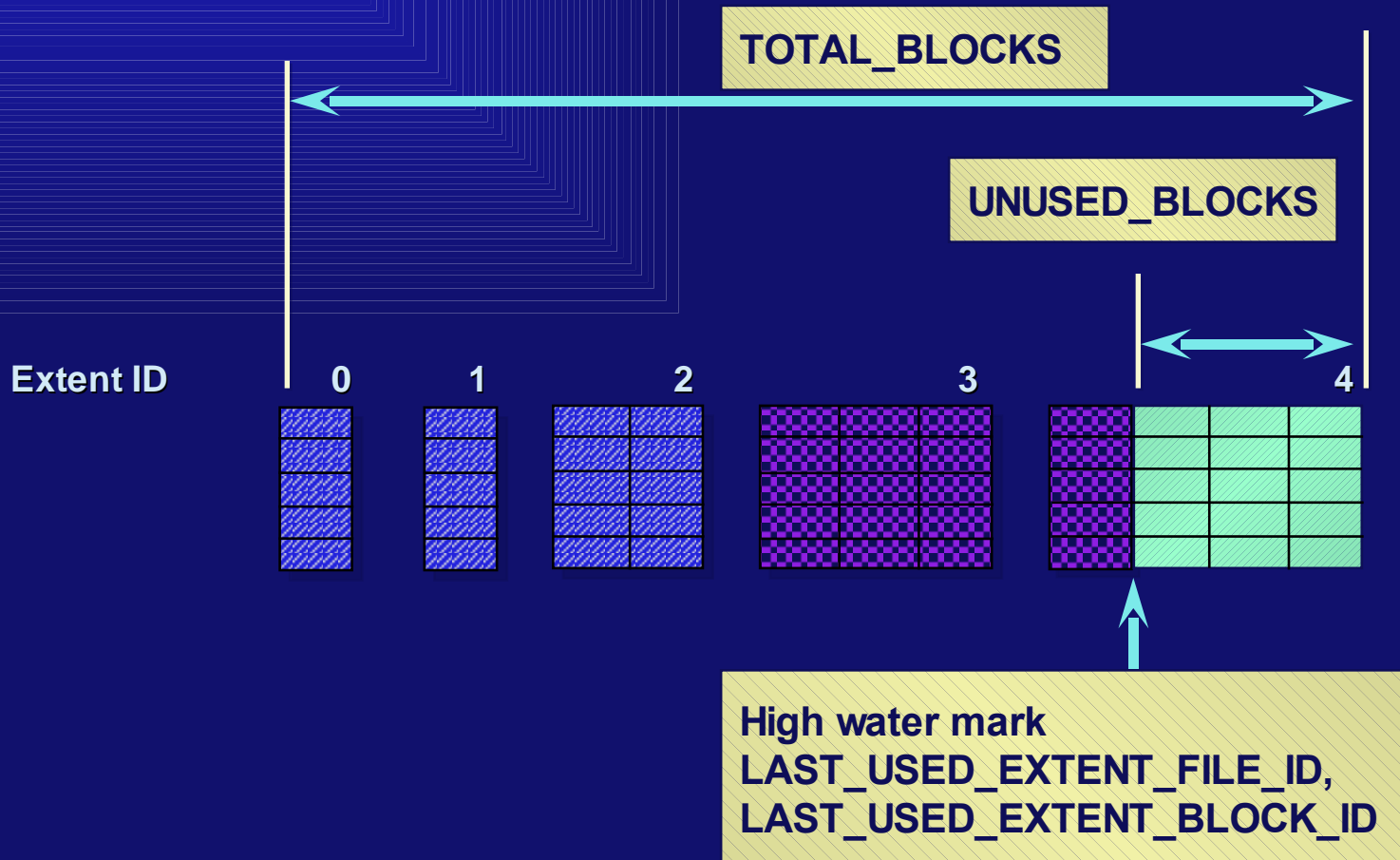


Unused block



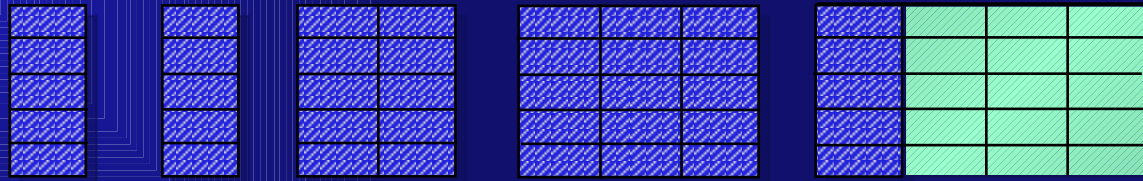
Free space after delete

Finding the High Water Mark: DBMS_SPACE.UNUSED_SPACE



Deallocation of Unused Space

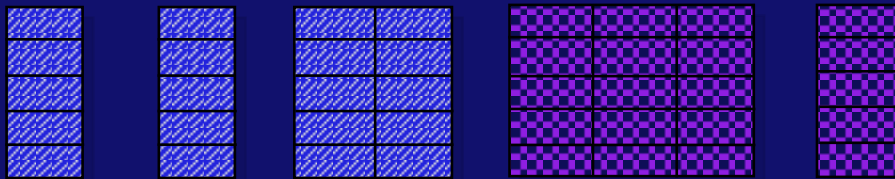
Before deallocation



```
ALTER TABLE scott.employees  
DEALLOCATE UNUSED;
```

High water mark

After deallocation



Used block



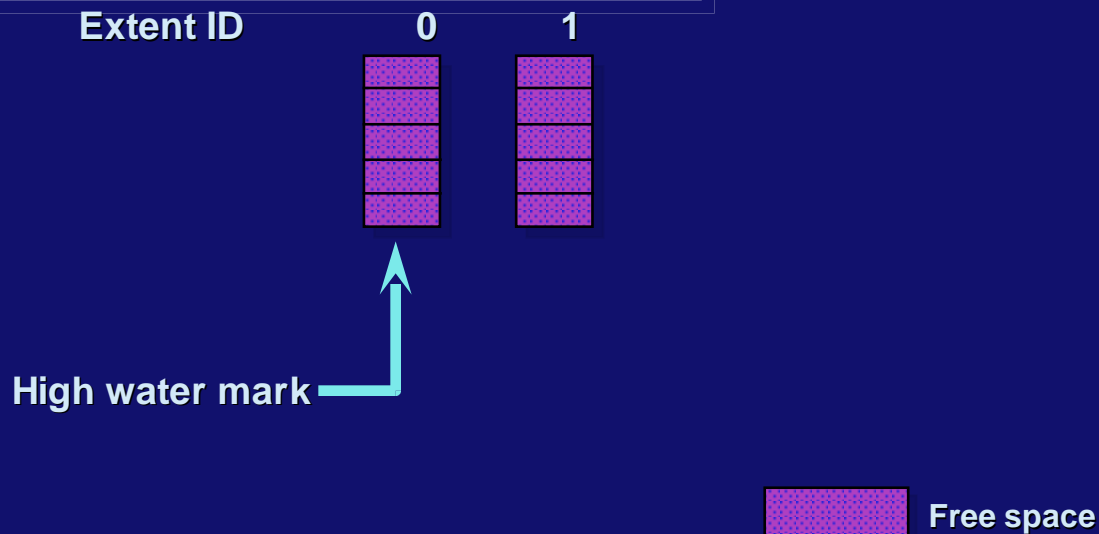
Unused block



Free space after delete

Truncating a Table

```
TRUNCATE TABLE scott.employees;
```



Dropping Tables

```
DROP TABLE scott.departments  
CASCADE CONSTRAINTS;
```

Validating Table Structure

- The Oracle server verifies the integrity of each data block.
- Use the **CASCADE** option to validate the structure of all indexes on the table, and perform cross-referencing between the table and its indexes.

```
ANALYZE TABLE scott.employees  
VALIDATE STRUCTURE;
```

Detecting Row Migration

- The Oracle server gathers statistics based on sample data and updates data dictionary.

```
ANALYZE TABLE scott.employees  
ESTIMATE STATISTICS;
```

- Check CHAIN_CNT

```
SELECT chain_cnt  
FROM DBA_TABLES  
WHERE table_name='EMPLOYEES'  
AND owner='SCOTT' ;
```

Retrieving Table Information

DBA_OBJECTS

OWNER
OBJECT_NAME
OBJECT_ID
DATA_OBJECT_ID
CREATED

DBA_SEGMENTS

OWNER
SEGMENT_NAME
TABLESPACE_NAME
HEADER_FILE
HEADER_BLOCK

DBA_TABLES

OWNER
TABLE_NAME
PCT_FREE
PCT_USED
INITIAL_EXTENT
NEXT_EXTENT
MIN_EXTENTS
MAX_EXTENTS
PCT_INCREASE
CACHE
BLOCKS
EMPTY_BLOCKS
CHAIN_CNT

Retrieving Extent Information

DBA_EXTENTS

- **OWNER**
- **SEGMENT_NAME**
- **EXTENT_ID**
- **FILE_ID**
- **BLOCK_ID**
- **BLOCKS**

DBMS_ROWID Package

Commonly used functions:

Function Name	Description
ROWID_CREATE	Creates a ROWID from individual components
ROWID_OBJECT	Returns the object identifier for a ROWID
ROWID_RELATIVE_FNO	Returns the relative file number for a ROWID
ROWID_BLOCK_NUMBER	Returns the block number for a ROWID
ROWID_ROW_NUMBER	Returns the row number for a ROWID
ROWID_TO_ABSOLUTE_FNO	Returns the absolute file number for a ROWID
ROWID_TO_EXTENDED	Converts a ROWID from restricted to extended
ROWID_TO_RESTRICTED	Converts a ROWID from extended to restricted

Summary

- **Creating a table with appropriate storage and block utilization parameters**
- **Controlling table storage**
- **Using of DBMS_ROWID package**