

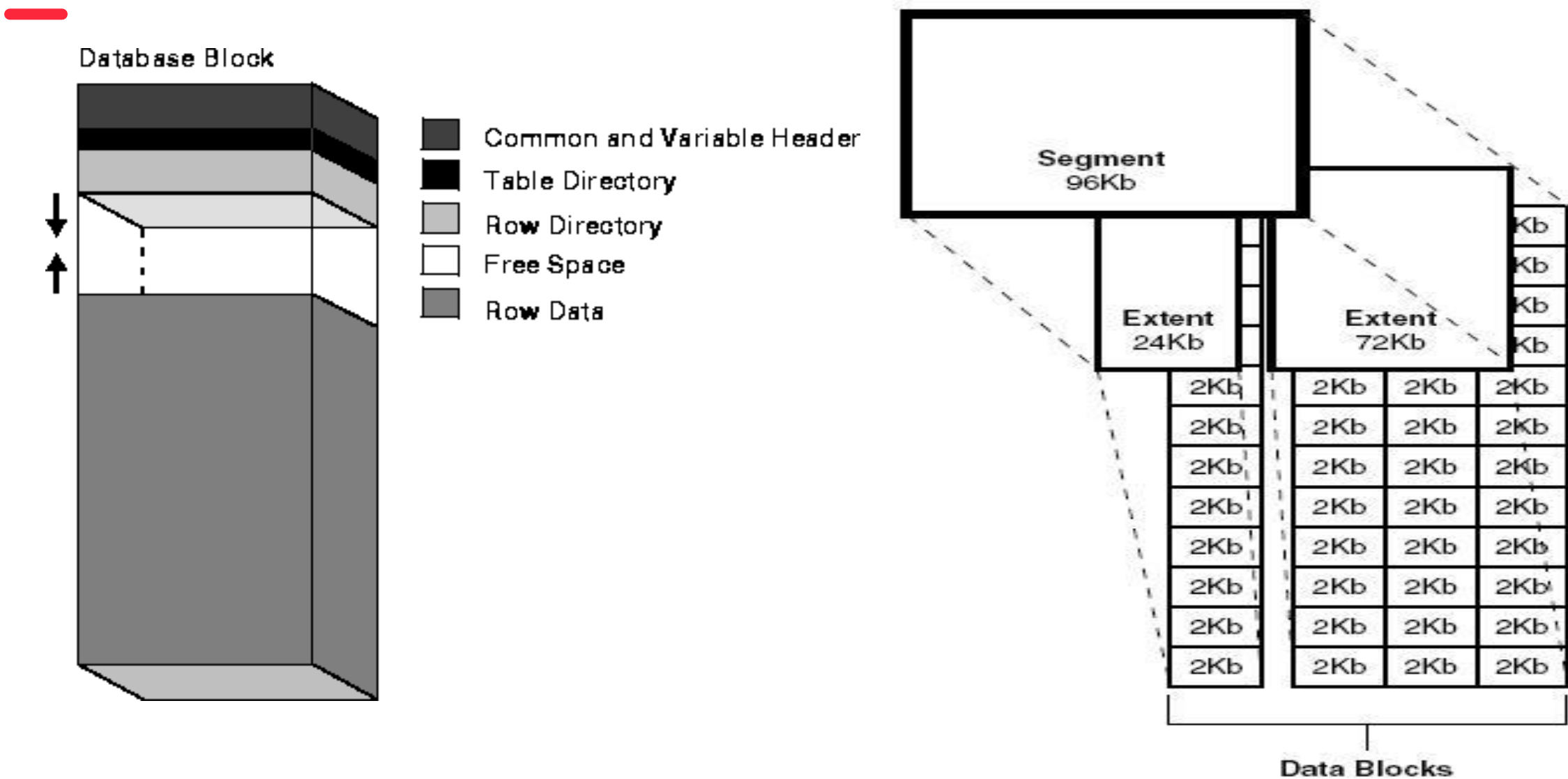
Storage, Structure & Relationship in Oracle



What are we going to see ?

- What are Segments, Extents ,Block?
- Dictionary Managed Tablespace & Locally Managed Tablespace
- Storage Parameters Concepts

Storage, Structure & Relationship in Oracle



Storage, Structure & Relationship in Oracle



Segments:

A segment is a set of extents that contains all the data for a specific logical storage structure within a tablespace. For example, for each table, Oracle allocates one or more extents to form that table's data segment; for each index, Oracle allocates one or more extents to form its index segment.

Oracle databases use four types of segments:

- **Table Segments:** Store table data.
- **Index Segments:** Store index data.
- **Temporary Segments:** Used for temporary storage during sorting and similar operations.
- **Undo Segments:** Used to store undo information for transactions.
- **Cluster Segments:** Store clustered table data.

Storage, Structure & Relationship in Oracle

B. Extent:

- i) Extent is collection of contiguous data blocks.
- ii) One or more extents make up a segment.

C. Segment:

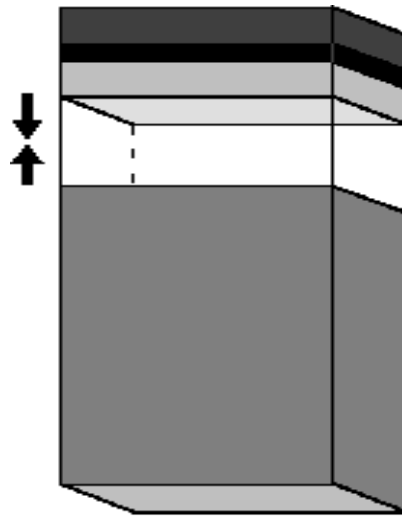
- i) Segment is set of extents allocated for specific data structure (like table or index).
- ii) Various kind of segments are table, index, cluster, rollback, temporary ...
- iii) Important views for segments are `dba_segments`, `user_segments`, `all_segments`
- iv) In a Segment, first block of first extent contains segment header information

Storage, Structure & Relationship in Oracle

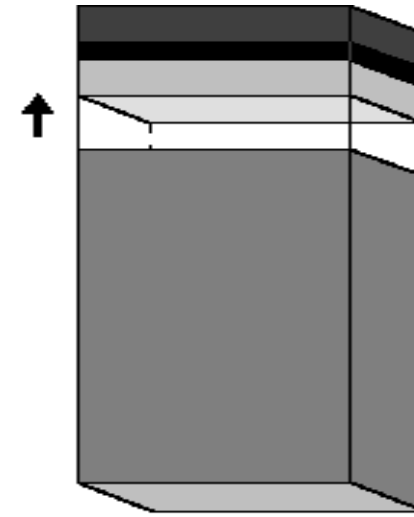
Data Blocks:

- i) Data blocks is smallest logical unit to store Oracle Data.
- ii) One data block represent specific number of bytes on physical hard disk.
- iii) Data Block in Oracle is also called as logical block
- iv) Data Block size is usually multiple of operating system block size
- v) You can have multiple block sizes within single database (max. five)
- vi) Block Size is specified by initialization parameter DB_BLOCK_SIZE
- vii) Format of Data Block is
 - a) Header : contains generic information like block address and type of segment (index, data..)
 - b) Table Directory : contains information about table having rows in that block
 - c) Row Directory : contains information about actual row contained in that block
 - d) Free Space : available space in data block for additional row or update of row which require more space.
 - e) Row Data : contains table or index data. First three component of data block (Header, Table & Row directory) collectively known as Overhead

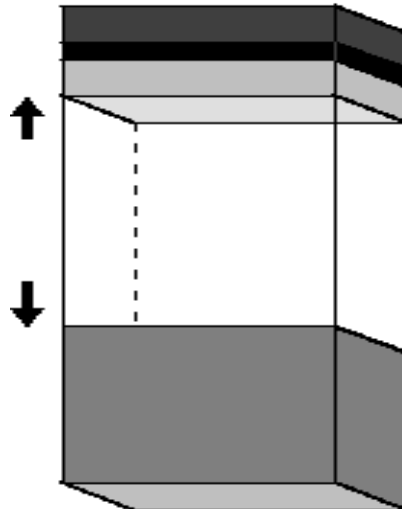
Storage, Structure & Relationship in Oracle



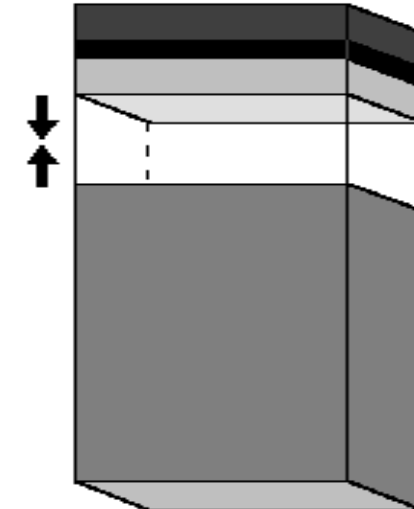
1 Rows are inserted up to 80% only, because PCTFREE specifies that 20% of the block must remain open for updates of existing rows.



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 . . .

Storage, Structure & Relationship in Oracle



Locally Managed Tablespaces:

- A tablespace manages its own extents maintains a bitmap in each datafile to keep track of the free or used status of blocks in that datafile.
- Each bit in the bitmap corresponds to a group of blocks.
- When an extent is allocated or freed for reuse, Oracle changes the bitmap values to show the new status of the blocks.
- These changes do not generate rollback information because they do not update tables (like sys.uet\$, sys.fet\$) in the data dictionary (except for special cases such as tablespace quota information).

Storage, Structure & Relationship in Oracle



Locally Managed Tablespaces:

- When you create a locally managed tablespace, header bitmaps are created for each datafile.
- If more datafiles are added, new header bitmaps are created for each added file.
- Local management of extents automatically tracks adjacent free space, eliminating the need to coalesce free extents.
- The sizes of extents that are managed locally can be determined automatically by the system.
- Alternatively, all extents can have the same size in a locally managed tablespace.

Storage, Structure & Relationship in Oracle



Dictionary Managed Tablespaces:

- In DMT, to keep track of the free or used status of blocks, oracle uses data dictionary tables.
- When an extent is allocated or freed for reuse, free space is recorded in the SYS.FET\$ table, and used space in the SYS.UET\$ table.
- Whenever space is required in one of these tablespaces, the ST (space transaction) enqueue latch must be obtained to do inserts and deletes against these tables.
- As only one process can acquire the ST enqueue at a given time, this often lead to contention.
- These changes generate rollback information because they update tables (like sys.uet\$, sys.fet\$) in the data dictionary.

Storage, Structure & Relationship in Oracle



Advantages in using Locally Managed Tablespaces Over Dictionary Managed Tablespaces:

- Because locally managed tablespaces do not record free space in data dictionary, it reduces contention on these tables.
- Local management of extents automatically tracks adjacent free space, eliminating the need to coalesce free extents.
- Sizes of extents that are managed locally can be determined automatically by the system. Alternatively, all extents can have the same size in a locally managed tablespace.
- Reduced fragmentation

Storage Parameters Concepts in Oracle

Oracle allocates and manages data storage in tablespaces. These parameters play a crucial role in optimizing storage usage, performance, and overall database health.

Key Storage Parameters:

INITIAL_EXTENT_SIZE:

- This parameter specifies the size of the first extent allocated to a segment. Setting it appropriately can minimize fragmentation and optimize performance for inserts and updates.

NEXT_EXTENT_SIZE:

- This parameter defines the size of subsequent extents allocated to a segment after the initial extent is filled. A larger size can reduce the number of extent allocation calls, improving performance, while a smaller size can prevent wasted space.

PCTFREE:

- This percentage specifies the amount of space Oracle reserves in an extent for future data growth. A higher percentage can prevent fragmentation but reduce available space for immediate use.

Storage Parameters Concepts in Oracle



Key Storage Parameters:

MIN_EXTENTS:

- This parameter guarantees a minimum number of extents for a segment, ensuring sufficient space for initial data and preventing frequent extent allocation calls.

MAXEXTENTS:

- This parameter defines the maximum number of extents allowed for a segment, limiting potential fragmentation and excessive disk usage.

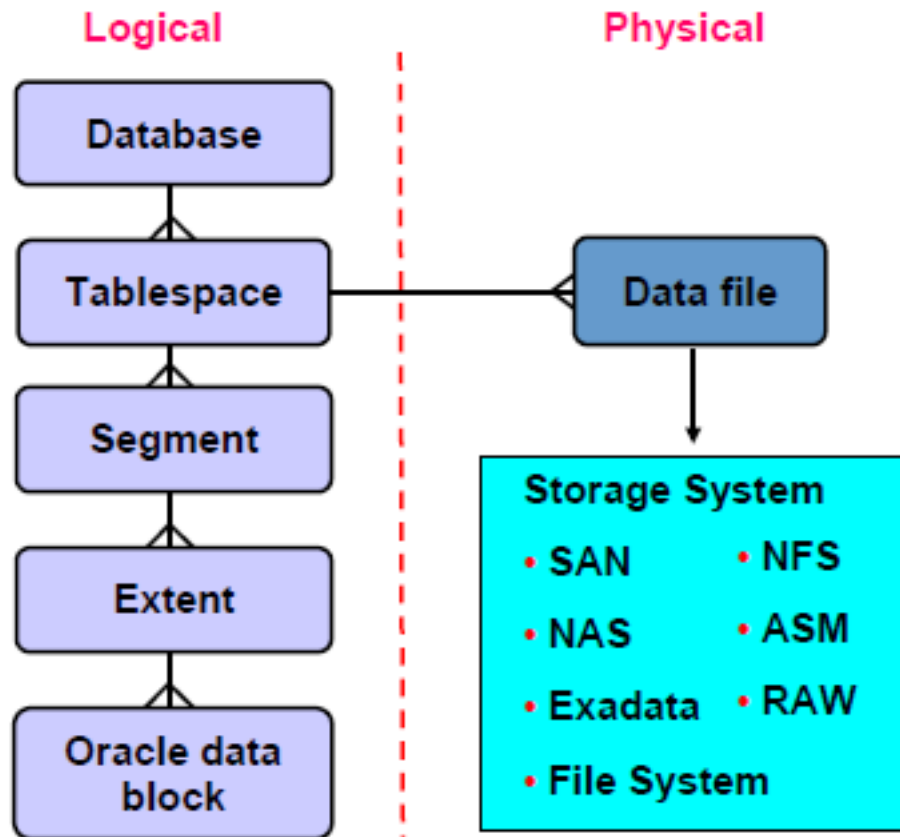
PCTUSED:

- This percentage specifies the threshold of used space in an extent before Oracle considers allocating a new one. Setting it appropriately can balance performance and space utilization.

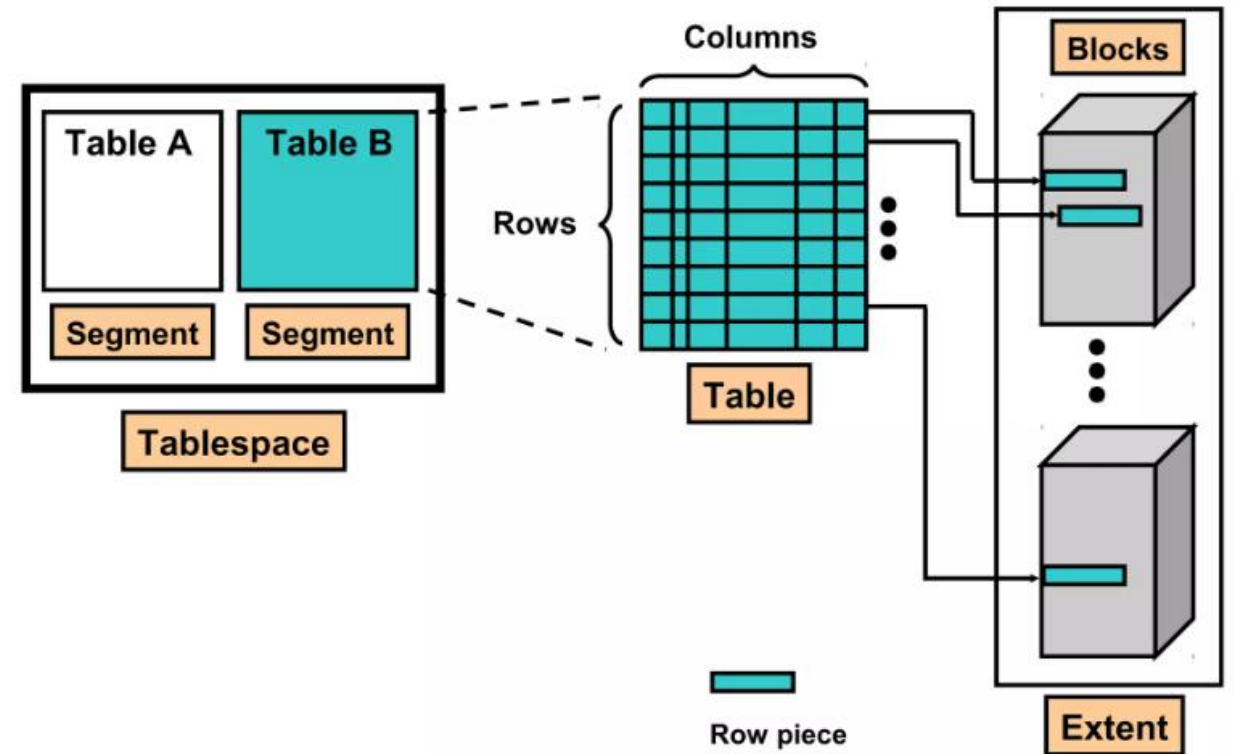
FREELISTS and FREELIST GROUPS:

- These parameters define how Oracle manages and distributes free space within an extent, impacting insert performance in OLTP workloads.

Storage Parameters Concepts in Oracle



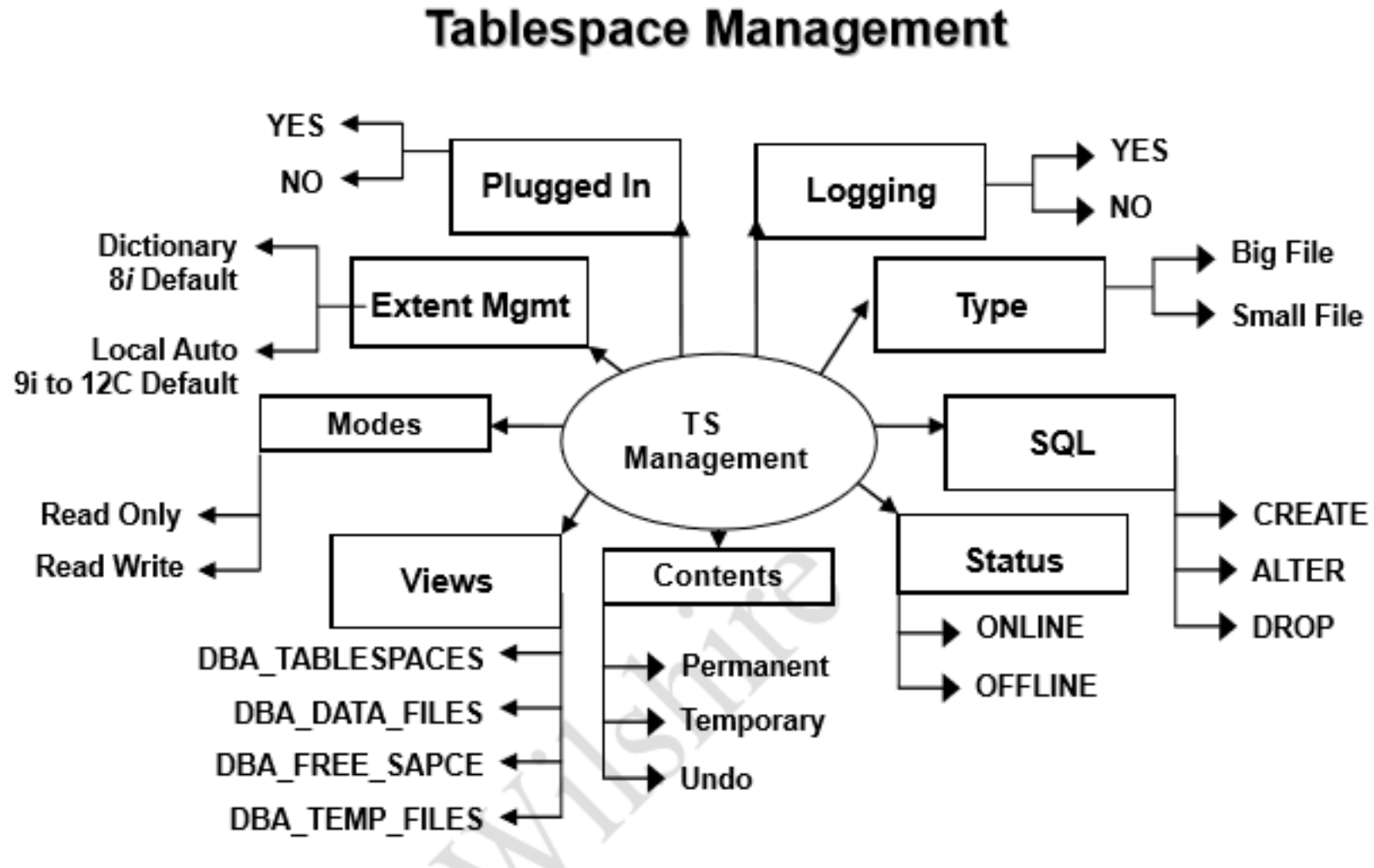
How Table Data Is Stored



Storage Parameters Concepts in Oracle

SMALL FILE Ts	BIG FILE Ts
Upto 9i we have Small file Ts	From 10g we have Big file Ts
Small file <u>ts</u> can have one or more datafiles.	Big file <u>ts</u> can have only one datafile.
A Single small file can support for 1022 files with 2k,4k,8k,16k,32k block sizes.	A single <u>Bigfile</u> tablespace file can be up to 128 terabytes for a 32K block tablespace and 32 terabytes for an 8K block tablespace

Storage Parameters Concepts in Oracle



Undo Management



What are we going to see ?

- Undo Management
- Creating UNDO tablespace , Monitor and Administer UNDO
- Configuring UNDO retention , Configuring UNDO retention with guarantee
- Dropping UNDO tablespace , How to change the default UNDO tablespace

Undo Management - What Is Undo and Why Undo in Oracle?



- Oracle Database has a method of maintaining information that is used to rollback or undo the changes to the database.
- Oracle Database keeps records of actions of transactions, before they are committed, and Oracle needs this information to rollback or undo the changes to the database.
- These records are called rollback or undo records.

Undo Management - What Is Undo and Why Undo in Oracle?



These records are used to:

- Rollback transactions - when a ROLLBACK statement is issued, undo records are used to undo changes that were made to the database by the uncommitted transaction.
- Recover the database - during database recovery, undo records are used to undo any uncommitted changes applied from the redo log to the datafiles.
- Provide read consistency - undo records provide read consistency by maintaining the before image of the data for users who are accessing the data while another user is changing it.
- Analyze data as of an earlier point in time by using Flashback Query.
- Recover from logical corruptions using Flashback features.

Undo Management - Automatic Undo Management

UNDO_MANAGEMENT

- The following initialization parameter setting causes the STARTUP command to start an instance in automatic undo management mode: UNDO_MANAGEMENT = AUTO
- The default value for this parameter is MANUAL i.e. manual undo management mode.

UNDO_TABLESPACE

- UNDO_TABLESPACE an optional dynamic parameter, can be changed online, specifying the name of an undo tablespace to use.
- An undo tablespace must be available, into which the database will store undo records.
- The default undo tablespace is created at database creation, or an undo tablespace can be created explicitly.

To find out the undo tablespaces in database:

SQL> *select tablespace_name, contents from dba_tablespaces where contents = 'UNDO';*

Undo Management - Automatic Undo Management

UNDO Data is :

- A copy of original , pre modified data
- Captured for every transaction that changes data
- Retained at least until the transaction is ended

Used to Support:

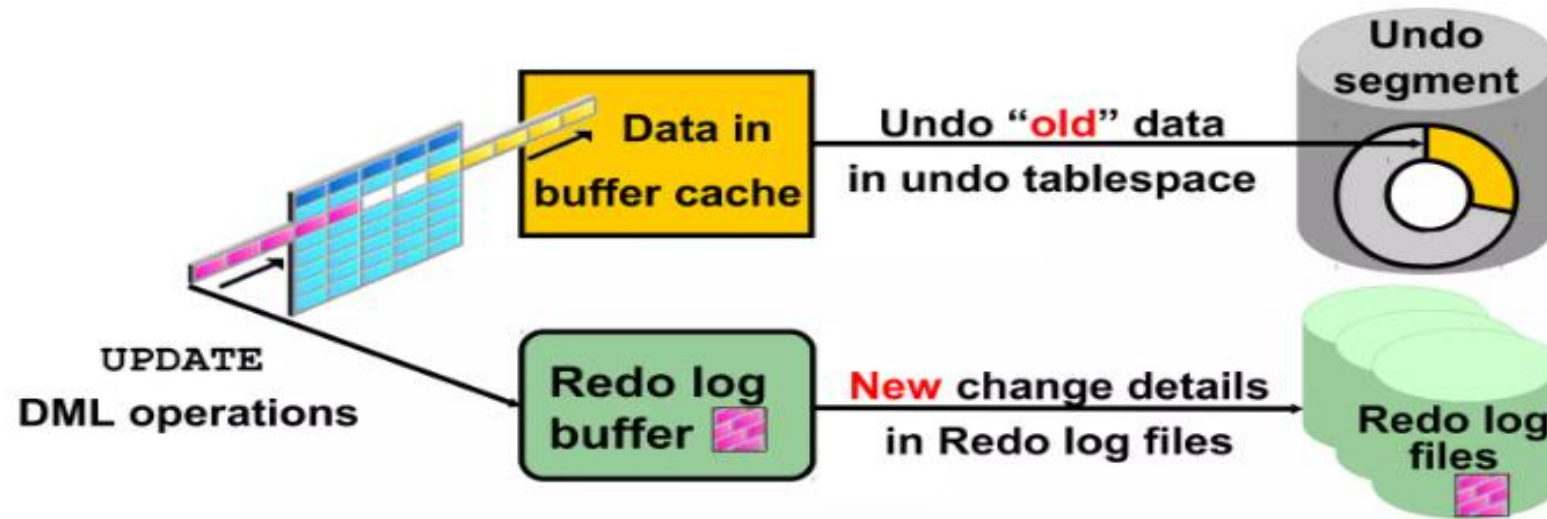
- Rollback operations
- Read consistent queries
- Oracle flashback queries, Oracle flashback transactions and oracle flashback table
- Recovery from failed transactions

To findout the current undo tablespace

SQL> show parameter *undo_tablespace* (OR)

SQL> *select VALUE from v\$parameter where NAME='undo_tablespace';*

Undo Management - TRANSACTIONS AND UNDO DATA



- Each transaction assigned to only one segment.
- An undo segment can service more than one transaction at a time.

Undo Management

UNDO_RETENTION:

- Committed undo information normally is lost when its undo space is overwritten by a newer transaction.
- However, for consistent read purposes, long-running queries sometimes require old undo information for undoing changes and producing older images of data blocks.
- The success of several Flashback features can also depend upon older undo information.
- The default value for the UNDO_RETENTION parameter is 900.
- Retention is specified in units of seconds.
- This value specifies the amount of time, undo is kept in the tablespace.
- The system retains undo for at least the time specified in this parameter.

You can set the UNDO_RETENTION in the parameter file:

`UNDO_RETENTION = 1800`

Undo Management

UNDO_RETENTION

You can change the UNDO_RETENTION value at any time using:

```
SQL> ALTER SYSTEM SET UNDO_RETENTION = 2400;
```

- The effect of the UNDO_RETENTION parameter is immediate, but it can only be honored if the current undo tablespace has enough space.
- If an active transaction requires undo space and the undo tablespace does not have available space, then the system starts reusing unexpired undo space (if retention is not guaranteed).
- This action can potentially cause some queries to fail with the ORA-01555 "snapshot too old" error message.
- UNDO_RETENTION applies to both committed and uncommitted transactions since the introduction of flashback query feature in Oracle needs this information to create a read consistent copy of the data in the past.

Undo Management

View	Description
DBA_TABLESPACES	Tablespace names and characteristics, including the CONTENTS column, which can be PERMANENT, TEMPORARY, or UNDO; the undo RETENTION column is NOT APPLY, GUARANTEE, or NOGUARANTEE.
DBA_UNDO_EXTENTS	All undo segments in the database, including their size, their extents, the tablespace where they reside, and current status(EXPIRED or UNEXPIRED).
V\$UNDOSTAT	The amount of undo usage for the database at ten-minute intervals; contains at most 1008 rows (7 days).
V\$ROLLSTAT	Rollback segment statistics, including size and status.
V\$TRANSACTION	Contains one row for each active transaction for the instance.

Undo Management - Read Consistency vs Successful DML

- For OLTP databases, generally we want DML commands to succeed at the expense of read-consistent queries.
- For a DSS environment, however, we may want long-running queries to complete without getting a “Snapshot too old” error.
- Although increasing the UNDO_RETENTION parameter or increasing the size of the undo tablespace helps to ensure that undo blocks are available for read consistent queries, undo tablespaces have another characteristic to help ensure that queries will run to completion: the RETENTION GUARANTEE setting.
- Undo retention guarantee is set at the tablespace level, and it can be altered at any time.
- Setting a retention guarantee for an undo tablespace ensures that an unexpired undo within the tablespace should be retained even if it means that DML transactions might not have enough undo space to complete successfully.

Undo Management - Read Consistency vs Successful DML



By default, a tablespace is created with NOGUARANTEE, unless you specify the GUARANTEE keyword, either when the tablespace is created or later with

ALTER TABLESPACE:

SQL> **alter tablespace undotbs1 retention guarantee;**
Tablespace altered.

SQL> **select tablespace_name, retention from dba_tablespaces where tablespace_name = 'UNDOTBS1';**

TABLESPACE_NAME	RETENTION
-----	-----
UNDOTBS1	GUARANTEE

Undo Management - Read Consistency vs Successful DML



Best Practices for Undo Tablespace/Undo Management in Oracle

This following list of recommendations will help you manage your undo space to best advantage.

- You need not set a value for the UNDO_RETENTION parameter unless your system has flashback or LOB retention requirements.
- Allow 10 to 20% extra space in your undo tablespace to provide for some fluctuation in your workload.
- Set the warning and critical alert thresholds for the undo tablespace alert properly.
- To tune SQL queries or to check on runaway queries, use the value of the SQLID column provided in the long query or in the V\$UNDOSTAT or WRH\$_UNDOSTAT views to retrieve SQL text and other details on the SQL from V\$SQL view.

User Management in Oracle



What are we going to see ?

- User Management , Creation of Users
- Assigning Privileges , Creating and Granting Roles
- Allocating Tablespace Quotas to Users
- Assigning default Tablespace for Users
- Setting Profiles to Users
- Levels of Authentication
- Schema Management

Users, Roles & Profiles in Oracle



By just creating a new user will not make the new user access the database. There are necessary roles and privileges that must be assigned to the user

To check all users inside database:

```
SQL> select username, account_status, default_tablespace from dba_users;
```

To check current user

```
SQL> show user;
```

To Lock / Unlock user

```
SQL> alter user scott account unlock;
```

```
SQL> alter user scott account lock;
```

Users, Roles & Profiles in Oracle

To Create new user

```
SQL> create user usr1 identified by usr1;
```

To create new user by assigning a default tablespace

```
SQL> create user usr2 identified by usr2 default tablespace users;
```

To change user password

```
SQL> alter user usr1 identified by oracle;
```

Check Database Default Tablespace:

When you create a new user without specifying a default tablespace, database default tablespace is assigned to the user. Use below command to find database default tablespace

```
SQL> select PROPERTY_NAME, PROPERTY_VALUE from database_properties where PROPERTY_NAME like '%DEFAULT%';
```

Users, Roles & Profiles in Oracle

Change User Default Tablespace

Use below command to change default tablespace of a user

```
SQL> alter user usr1 default tablespace example;
```

Note: The objects created in the old tablespace remain unchanged even after changing a default tablespace for a user

Tablespace Quota

You can specify a limit onto how much tablespace quota (size) a user can use

```
SQL> Alter user usr1 quota 10M on users;
```

Note: Allocating quota doesn't represent reserving the space. If 2 or more users are sharing a tablespace, quota will be filled up in first come first serve basis

Users, Roles & Profiles in Oracle

Roles in Oracle:

When you create a new user, you must at least assign CREATE SESSIONS privilege so the user can connect to the database

```
grant create session to usr1;
```

In real-time, there are more than one permission which must be assigned to a user. Sometimes the list might be very big.

For example, there is a manager who must be able to perform:

- Insert into EMP & DEPT table
- Update DEPT table
- Delete from BONUS table

Instead of giving above privileges to the user one by one, we can create a role inside the database. We then assign all privileges to the role and then assign the role to a user.

Users, Roles & Profiles in Oracle

Roles in Oracle:

Create New Role:

Use below command to create new role inside the database

```
SQL> CREATE ROLE SALES_MANAGER;
```

Grant Privileges to Role

Assign all the privileges to the role NOT THE USER

```
SQL> GRANT INSERT ON SCOTT.EMP TO SALES_MANAGER;
```

```
SQL> GRANT INSERT ON SCOTT.DEPT TO SALES_MANAGER;
```

```
SQL> GRANT UPDATE ON SCOTT.DEPT TO SALES_MANAGER;
```

```
SQL> GRANT DELETE ON SCOTT.BONUS TO SALES_MANAGER;
```

Grant Role to a User

Now that you have assigned all the necessary privileges to a role, its time to assign the role to a user

```
SQL> GRANT SALES_MANAGER TO USR1;
```

Users, Roles & Profiles in Oracle

A profile is a way to control system resource that can be used by a database user. Profile management is of two types

- Password management
- Resource management

Password Management:

The password management allows a DBA to have more control over user passwords. Some of the parameters you might be familiar in general like failed login attempts, password lock time etc..

FAILED_LOGIN_ATTEMPTS:	How many times a user can fail to login
PASSWORD_LOCK_TIME:	Users who exceed failed login attempts, their password will be locked in specific time
PASSWORD_LIFE_TIME:	Till when password is valid in days
PASSWORD_GRACE_TIME:	Grace period for user to change password, else account will be locked
PASSWORD_REUSE_TIME:	After how many days user can re-use same password
PASSWORD_REUSE_MAX:	Specify how many times old password can be used
PASSWORD_VERIFY_FUNCTION:	Defines rules for setting a new password

Users, Roles & Profiles in Oracle - Resource Management:

- Resource management helps in limiting the database abuse a user can cause. For example, if a user connects to database and never runs a query then this idle connection will take system resources like CPU.

To restrict such kind of issues, we have resource management parameters

SESSIONS_PER_USER: How many concurrent sessions user can open

IDLE_TIME: Total time user can stay inside database without doing any activity

CONNECT_TIME: Total time user can stay inside database whether idle or active

Note: resource management parameters will take in effect only if RESOURCE_LIMIT parameter is set to TRUE.

Use below command to check the RESOURCE_LIMIT parameter

```
SQL> show parameter resource_limit;
```

By default, the parameter is set to FALSE. You can change it via below

```
SQL> alter system set resource_limit=TRUE scope=both;
```

Users, Roles & Profiles in Oracle - Resource Management:

To create a new user profile

```
SQL> create profile my_profile limit  
failed_login_attempts 3  
password_lock_time 1  
sessions_per_user 1  
idle_time 5;
```

Note: password lock time by default is for 1 day. You can specify it in minutes (n/1440) or even in seconds (n/86400)

To assign profile to a user

```
SQL> alter user scott profile my_profile;
```

To check profiles assigned to a user

```
SQL> SELECT USERNAME, PROFILE FROM DBA_USERS WHERE USERNAME='SCOTT';
```

To check profile parameter values

```
SQL> SELECT * FROM DBA_PROFILES WHERE PROFILE='&PROFILE_NAME';
```

Users, Roles & Profiles in Oracle - Find User Permissions :

To check system privileges granted to a user

```
select privilege from dba_sys_privs where grantee='SCOTT';
```

To check object level privileges granted to a user or role

```
SQL> select owner, table_name, privilege from dba_tab_privs where grantee='SALES_CLERK';
```

To check roles assigned to a user

```
select granted_role from dba_role_privs where grantee='SCOTT';
```

To check permissions assigned to role

```
select privilege from role_sys_privs where role='SALES_CLERK';
```

```
select owner, table_name, privilege from role_tab_privs where role='SALES_CLERK';
```

To check roles granted to another role

```
SQL> select granted_role from role_role_privs where role='SALES_CLERK';
```