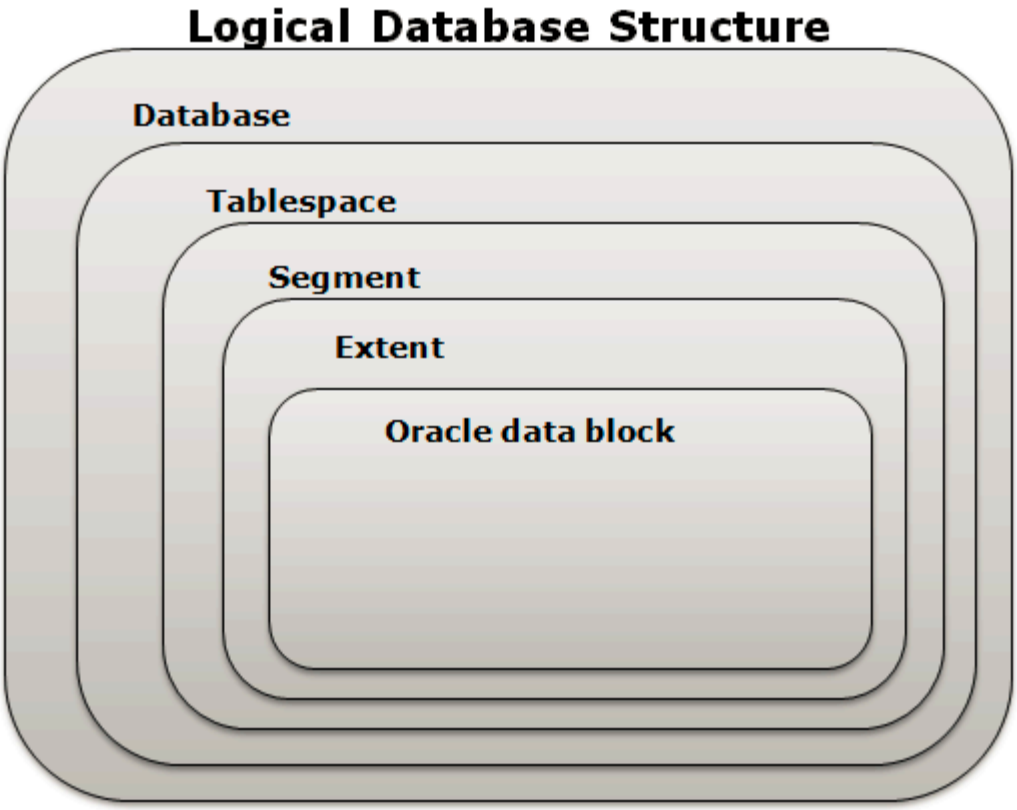
1 BLOCK :

Smallest oracle unit. Oracle stores data in data blocks. A block can be 2k to 32k but by default it is 8k. while creating the tablespace we can specify blocksize 32k;
create tablespace . . . . blocksize 32k;
show parameter block;

8K

1 EXTENT:

The next level of logical database space is called an *extent*. **1 EXTENT CAN BE ONLY IN ONE**



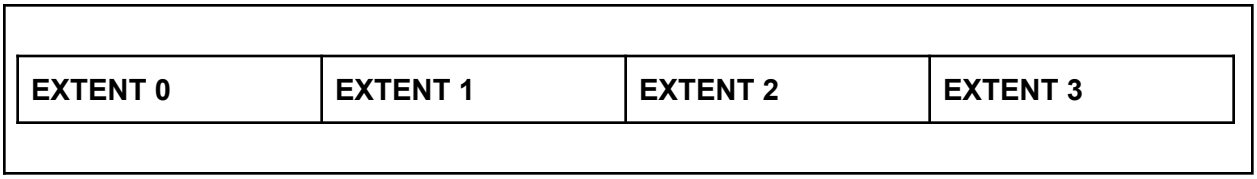
DATAFILE

8 blocks make on extent.

Block	**Block**	**Block**	**Block**
Block	**Block**	**Block**	**Block**

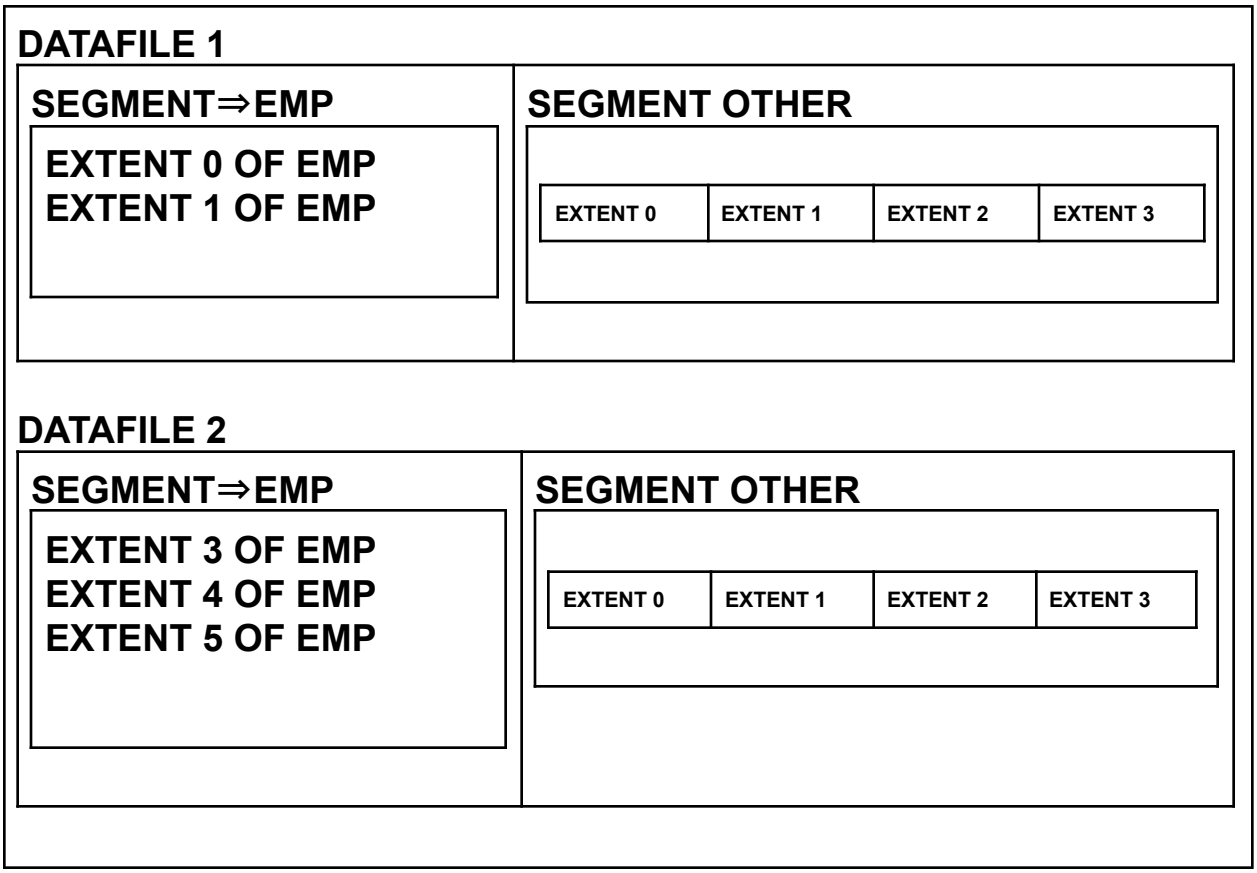
1 SEGMENT:

The level of logical database storage above an extent is called a *segment*. A segment is a set of extents that have been allocated for a specific type of data structure, and that all are stored in the same tablespace. **1 SEGMENT CAN HAVE MULTIPLE EXTENT FROM DIFFERENT DATAFILE IN A TABLESPACE.**



1 TABLESPACE:

Can have multiple datafiles. One extent can not be in multiple datafiles. But one segment can be shared among the datafiles that is their segment can have extent from multiple datafiles.



TABLESPACES:

System:

It is the engine of the database. **SYSTEM** is the name of the first compulsory tablespace containing the Oracle **data dictionary.**
A data dictionary, is repository of metadata (information about data), such as its meaning, relationships to other data, origin, usage and format. Oracle's data dictionary is stored in the SYS schema.

Sysaux:
Performance of the entire database. Keeps track of 7 days of data by default. If we want to change it to a bigger number we can do that. SYSAUX is the name of the compulsory tablespace, introduced in 10g, to support optional database components like automatic workload repository , Statspack, Oracle Streams. The view to use the sysaux tablespace is v$sysaux_occupants.

```
SQL> desc v$sysaux_occupants
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OCCUPANT_NAME                                      VARCHAR2(64)
 OCCUPANT_DESC                                      VARCHAR2(64)
 SCHEMA_NAME                                        VARCHAR2(64)
 MOVE_PROCEDURE                                     VARCHAR2(64)
 MOVE_PROCEDURE_DESC                                VARCHAR2(64)
 SPACE_USAGE_KBYTES                                 NUMBER
```

```
SQL> SELECT occupant_name, schema_name, space_usage_kbytes FROM v$sysaux_occupants;

OCCUPANT_NAME              SCHEMA_NAME          SPACE_USAGE_KBYTES
-------------------------- -------------------- ------------------
LOGMNR                     SYSTEM                             6080
LOGSTDBY                   SYSTEM                              896
STREAMS                    SYS                                 512
XDB                        XDB                               49728
AO                         SYS                               22400
XSOQHIST                   SYS                               22400
XSAMD                      OLAPSYS                           15936
SM/AWR                     SYS                               77888
SM/ADVISOR                 SYS                               11008
SM/OPTSTAT                 SYS                               62080
SM/OTHER                   SYS                                4864
STATSPACK                  PERFSTAT                              0
ODM                        DMSYS                               256
SDO                        MDSYS                             33216
WM                         WMSYS                              7040
ORDIM                      ORDSYS                              512
ORDIM/PLUGINS              ORDPLUGINS                            0
ORDIM/SQLMM                SI_INFORMTN_SCHEMA                    0
EM                         SYSMAN                            50560
TEXT                       CTXSYS                             4736
ULTRASEARCH                WKSYS                                 0
ULTRASEARCH_DEMO_USER      WK_TEST                               0
EXPRESSION_FILTER          EXFSYS                             3712
EM_MONITORING_USER         DBSNMP                             1600
TSM                        TSMSYS                              256
JOB_SCHEDULER              SYS                                 768

26 rows selected.
```

Users:

When a user is created, by default Users tablespace is assigned to that User. We can optionally specify the default tablespace and default temporary tablespace for any objects created by that user. For example:

CREATE USER ahmad IDENTIFIED BY abc123 DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE temp;

If you omit the clauses when creating the user, the user will inherit the database default tablespace USERS.

Temp:
Temp or temporary tablespaces are used to store data with short lifespan (transient data), for example: global temporary tables. Maybe objects which will not exist for a longer period of time.

Undo:
Undo tablespaces are used to store "before image" data that can be used to undo transactions, Where ROLLBACK which is an SQL command is used to undo changes made to the database (restoring data to its state prior to the user making changes).Then UNDO tablespace is used.

Often when the undo space is full. we may get the **snapshot too old error ORA-01555** when a commit happens. Someone updates and it is a long running query and UNDO is getting full now when it overwrites the older changed data and the user commits. That is when the ORA-01555 snapshot too old error will be generated.
SOLUTION:

1. Increase the number of commits so the space is again available for next updates. (BEST WAY/PREFERRED ONE)
2. Increase the undo retention which is in seconds. DEFAULT is 900 seconds. Change it to a higher number let's say 7200 seconds which will be 120 min or two hours.

Block architecture

1 Tablespace(Logical): with one or more datafile(Physical)

1 Segment: 1 or more extents

1 Extent: 8 blocks

8k	8k	8k	8k
8k	8k	8k	8k

It is a good practice to keep PCTfree a little higher if there are so many future updates like stock markets business have many changes in a day. There are two options to avoid row migration which means to find the rows from multiple blocks like searching for information from different buildings.

1. use tcp free clause while creating the table i.e create table t1 . . . . . pctfree 60 ;
2. to avoid the fragmentation, copy table such as t1.bkp and then drop the original t1 and rename the t1.bkp as t1. This way all the rows gets added into the blocks from the beginning and there is a fresh start to old row migration.

1 Extent:

BLOCK #	Status	PCT % used (percentage used)	PCT % Free (percentage Free)
Block 1	D means dirty	20%	80%
Block 2	F means free/available	20%	80%
Block 3	F means free/available	20%	80%
Block 4	F means free/available	20%	80%
Block 5	F means free/available	20%	80%
Block 6	F means free/available	20%	80%
Block 7	F means free/available	20%	80%
Block 8	F means free/available	20%	80%

Row chaining:

To avoid row chaining ideal thing to do is to have bigger block sizes. That is not have the default size of the block as 8k but have 32k as the largest size. That was the row will not have to be divided among the block and can reside in one block.

1 Block

32K max size to avoid row chaining.

Row migration:

To avoid row migration leave more space for TCP % free so that way we can have more space for future updates and row migration can be avoided.

1 Block

Division to avoid row migration leave more space for TCP free
TCP % Free80% for future updates
TCP % used 20% for current rows inserted.