Oracle Tablespaces, etc.: Managing the Disk Resource

CS634
Lecture 7

These slides are not based on "Database Management Systems" 3rd ed, Ramakrishnan and Gehrke

Look at disks we have to work with on dbs3

```
[oracle@dbs3 ~]$ df
Filesystem            1K-blocks      Used  Available Use% Mounted on
/dev/mapper/ol-root    52403200  20489020   31914180  40% /
devtmpfs               32789692         0   32789692   0% /dev
tmpfs                  32800144         0   32800144   0% /dev/shm
tmpfs                  32800144   3244840   29555304  10% /run
tmpfs                  32800144         0   32800144   0% /sys/fs/cgroup
/dev/sda1                508588    217012     291576  43% /boot
/dev/mapper/ol-home 1845857688 234579576 1611278112  13% /home
tmpfs                   6560032         0    6560032   0% /run/user/54321
```

- df stands for "disk free", useful UNIX/Linux utility
- This df output shows
  - Several "tmpfs" filesystems that use physical memory for data,
  - Two /dev/mapper volumes, using real disk under the LVM (Logical Volume Manager) disk mapping manager,
  - One small disk partition for /boot, i.e., helping boot up the OS.
  - No networked disks on other systems (high security here)
  - /home has **`1845857688 1K blocks = 1845 GB = 1.8 TB`**

Linux LVM is managing the main disk resource

- Utility dmsetup (low level logical volume management) can give us more info:

```
[oracle@dbs3 ~]$ sudo dmsetup deps -o devname
ol-home: 1 dependencies : (sda2)
ol-swap: 1 dependencies : (sda2)
ol-root: 1 dependencies : (sda2)
```

- This shows that the LVM home and LVM root volumes both mapped to the single disk partition sda2, partition 2 of disk sda, where sd stands for "SCSI/SATA disk" and the a means the first such disk.
- The swap space is on this disk too.
- This one "disk" sda is actually a hardware RAID set.

Use “sudo fdisk –l” to find out about disks

```
Disk /dev/sda: 1979.1 GB, 1979119828992 bytes, 3865468416 sectors
…
Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048     1026047      512000   83  Linux
/dev/sda2         1026048  3865466879  1932220416   8e  Linux LVM

Disk /dev/sdb: 1018.2 GB, 1018230472704 bytes, 1988731392 sectors
…
Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048  1988728831   994363392   83  Linux
```

- This shows two disks, sda and sdb, where sdb is only about half as big as sda, and is not in active use (not in df output)

Disks on dbs3

- The "fdisk –l" output of last slide shows two disks, sda and sdb, where sda is 2TB and sdb is 1TB and is not in active use (not in df output)
- System has 7 500GB SATA disks (7200 rpm), total of 3.5 TB of disk, in RAID5 config.
- The 2TB of sda corresponds to 4 disks, and one more is needed for redundancy info, so 5 are involved in sda's RAID.
- That means about 5*100 io/s read bandwidth for sda, about ¼ of that for writing (read and write 2 blocks).
- 1 TB in sdb, not is use. Apparently 2 spares configured as one disk.

Compare to pe07

```
pe07$ df
Filesystem                     1K-blocks     Used Available Use% Mounted on
udev                            65970424        0  65970424   0% /dev
tmpfs                           13198528   730328  12468200   6% /run
/dev/mapper/pe07--vg-root      348419444 25334044 305363648   8% /
tmpfs                           65992620        0  65992620   0% /dev/shm
tmpfs                               5120        0      5120   0% /run/lock
tmpfs                           65992620        0  65992620   0% /sys/fs/cgroup
/dev/sda1                         482922   115992    341996  26% /boot
tmpfs                           13198528        0  13198528   0% /run/user/1188
vm61:/vol/fac2/home/eoneil     132112384 55867392  69533696  45% /home/eoneil
vm61:/vol/mail/spool/cs        131981312  6961152 118293504   6% /spool/mail
vm62:/vol/fac1/courses/cs637   131981312 76761088  48493568  62% /courses/cs637
vm62:/vol/fac2/data/htdocs     131981312 81536000  43718656  66% /data/htdocs
vm62:/vol/fac1/courses/cs634   131981312 76761088  48493568  62% /courses/cs634
```

- See similar tmpfs, boot, one /dev/mapper entry for the main filesystem, 5 mounted NFS (network file system) partitions at the moment
- The main filesystem has **`348419444 1K blocks = 348 GB,`** so almost certainly on just one disk.

pe07 disks, continued

```
pe07$  sudo dmsetup deps -o devname
pe07--vg-swap_1: 1 dependencies : (sda5)
pe07--vg-root: 1 dependencies   : (sda5)
```

This shows both the main filesystem and swap space are carved out of partition 5 of disk sda of this system.

From sudo fdisk –l: shows lots more disk in sda5:

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	999423	997376	487M	83	Linux
/dev/sda2		1001470	1952446463	1951444994	930.5G	5	Extended
/dev/sda5		1001472	1952446463	1951444992	930.5G	8e	Linux LVM

I asked Rick Martin, our head system administrator and network architect, why the filesystem in use (348GB, shown on last slide) was so small compared to the partition it resides in (930GB, shown above)…

Explanation from Rick Martin

- "I sized the root logical volume at 1/3 TB, arbitrarily but much as topcat is/was, leaving the rest of the volume group free. We have several degrees of freedom in using the rest of the disk."
  - Make another logical volume and mount it somewhere convenient, or
  - Grow the root lv (logical volume) and the filesystem within it, though I've never tried that.
- So he is keeping some of the disk in reserve, for possible use on this system, or, through networked filesystems, on another system
- He also shows use of other sudo commands, lvdisplay and vgdisplay for looking at lvs and vgs (volume groups) of LVM.
  - Lvdisplay shows that the swap space is 128GB, equal to size of memory on this system. We had seen that this was also in this partition.
  - Vgdisplay shows there is still more than 400GB unallocated, i.e. approximately 930 - 348 – 128
- However, still no info on what type of disks are in the RAID, or how many

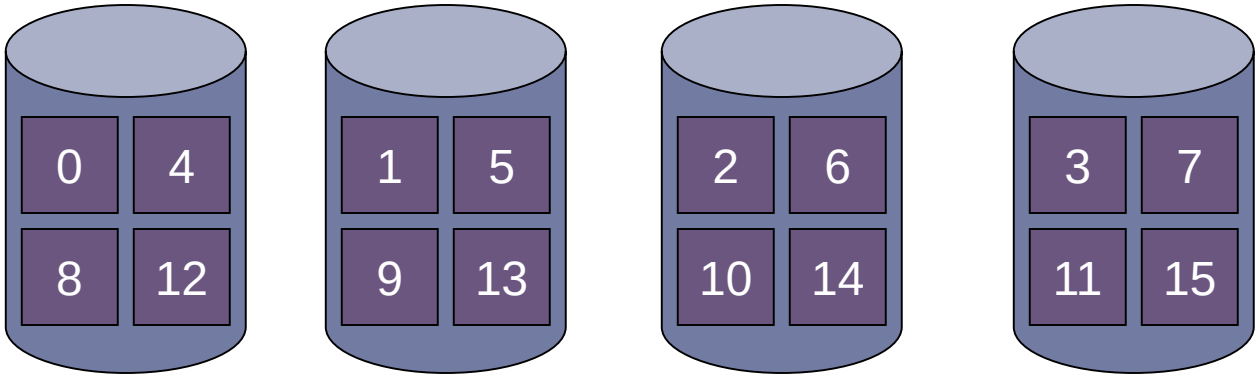
Explanation from Rick Martin, cont.

- For deeper system info, Rick says we need to talk to the DRAC "daughter board controller", which has its own embedded web server (password-protected) at pe07-drac.cs.umb.edu
  - He sent some output from it including the Dell service tag 8S3SR22.
  - I used this to look up the original system configuration at Dell (no privs needed once you have the tag) and finally found the disk info: one disk described as
    - 1TB 7.2K RPM Near-Line SAS 6Gb ps 2.5in Hot-plug Hard Drive
- So we have a RAID controller running a single disk!
- It's 7200rpm, SAS, with 6Gb/s bus compatibility (not its top transfer rate, which would be more like 100MB/s = 0.8Gb/s)
- The DRAC output also gave info on the two processors on the system, each with 8 cores.

Partitions of a Disk

- A disk can be split up into partitions
- A partition is a consecutive sequence of cylinders of the disk.
  - Thus it limits seek time for files within it.
- Partitions are created before file systems. Each partition may have its own filesystem.
- Under UNIX/Linux (including MacOS), file systems can be pasted together by "mounting" one filesystem on a directory of another already in use.
  - The first filesystem to be put in use has the root directory of the final filesystem.
  - You can tell what partition (or at least LVM volume) your current directory is part of by using the "df ." command.
- This describes local disks and partitions. It is also possible to mount a remote filesystem via NFS (network file system).
  - However, for database use, we want local disk.

RAID LBNs and partitions



- Stripes for RAID 0 (striping) cycle around disks and then along tracks on a cylinder and then along cylinders, like next-LBNs for a simple disk. RAID 5 is similar but more complicated.
- The stripe# = LBN/stripe-size, so each stripe has a consecutive sequence of LBNs.
- Thus as you fill a RAID with data, you are filling up cylinder 0 on all disks, then cylinder1 on all disks, etc.
- You can partition the RAID, and then certain cylinders of all disks belong to a certain partition.

Oracle Data Files: *.dbf: where are they?

```
SQL> select file_name from dba_data_files;

FILE_NAME
-----------------------------------------------
/home/oracle/app/oracle/oradata/dbs3/system01.dbf
/home/oracle/app/oracle/oradata/dbs3/sysaux01.dbf
/home/oracle/app/oracle/oradata/dbs3/undotbs01.dbf
/home/oracle/app/oracle/oradata/dbs3/users01.dbf
/home/oracle/app/oracle/oradata/dbs3/eoneil1.dbf
/home/oracle/app/oracle/oradata/dbs3/eoneil_setq.dbf
/home/oracle/app/oracle/oradata/dbs3/eoneil_setq250.dbf
```

- So they are all in one directory, /home/oracle/app/oracle/oradata/dbs3
- This is in /home, i.e., in the big hardware RAID
- This query can be done from your "x" account (using privileges of select_catalog_role)

Look at .dbf files in the filesystem

```
[oracle@dbs3 ~]$ ls -l /home/oracle/app/oracle/oradata/dbs3
total 172863640
-rw-r-----. 1 oracle oinstall     10534912 Feb 10 14:37 control01.ctl
-rw-r-----. 1 oracle oinstall    104865792 Feb 10 14:11 eoneil1.dbf
-rw-r-----. 1 oracle oinstall 161061281792 Feb 10 14:11 eoneil_setq250.dbf
-rw-r-----. 1 oracle oinstall   1073750016 Feb 10 14:11 eoneil_setq.dbf
-rw-r-----. 1 oracle oinstall     52429312 Feb 10 10:06 redo01.log
-rw-r-----. 1 oracle oinstall     52429312 Feb 10 14:06 redo02.log
-rw-r-----. 1 oracle oinstall     52429312 Feb 10 14:36 redo03.log
-rw-r-----. 1 oracle oinstall   1971331072 Feb 10 14:35 sysaux01.dbf
-rw-r-----. 1 oracle oinstall    996155392 Feb 10 14:36 system01.dbf
-rw-r-----. 1 oracle oinstall   6042951680 Feb 10 14:06 temp01.dbf
-rw-r-----. 1 oracle oinstall   3523223552 Feb 10 14:36 undotbs01.dbf
-rw-r-----. 1 oracle oinstall   2070945792 Feb 10 14:11 users01.dbf
```

- **By far biggest is setq250's tablespace**

Tablespaces are created from OS files

- Oracle, simple case:

```
create tablespace setq
  datafile
'/home/oracle/app/oracle/oradata/dbs3/eoneil_setq.dbf'
  size 1 G;
```

- Don't need SIZE if file already exists
- These files need to be as contiguous on disk as possible for best performance
- Suggest reinitializing the filesystem before creating the file, if it has been used for many files of limited lifetime ("churned")
- Our disk is "fresh", never filled with files, then cleaned up.

Tablespaces in other products

- Create tablespace command exists in mysql 5.7, but not our v 5.6.
- Even in mysql 5.7, not in general use yet.
- For mysql v 5.1-5.6, can only set up the one and only all-inclusive system tablespace at initialization. You can add a file to it later under some conditions.
- DB2 has tablespaces much like Oracle.
- MS Sql Server has “file groups”

Files to Tablespaces on dbs3

```
SQL> column tablespace_name format a12
SQL> column file_name format a30
SQL> select file_name, tablespace_name, blocks from dba_data_files;

FILE_NAME                      TABLESPACE_N     BLOCKS
------------------------------ ------------ ----------
/home/oracle/app/oracle/oradat SYSTEM           121600
a/dbs3/system01.dbf

/home/oracle/app/oracle/oradat SYSAUX           240640
a/dbs3/sysaux01.dbf

/home/oracle/app/oracle/oradat UNDOTBS1         430080
a/dbs3/undotbs01.dbf

/home/oracle/app/oracle/oradat USERS            252800
a/dbs3/users01.dbf
```

Shows original tablespaces of the installed system

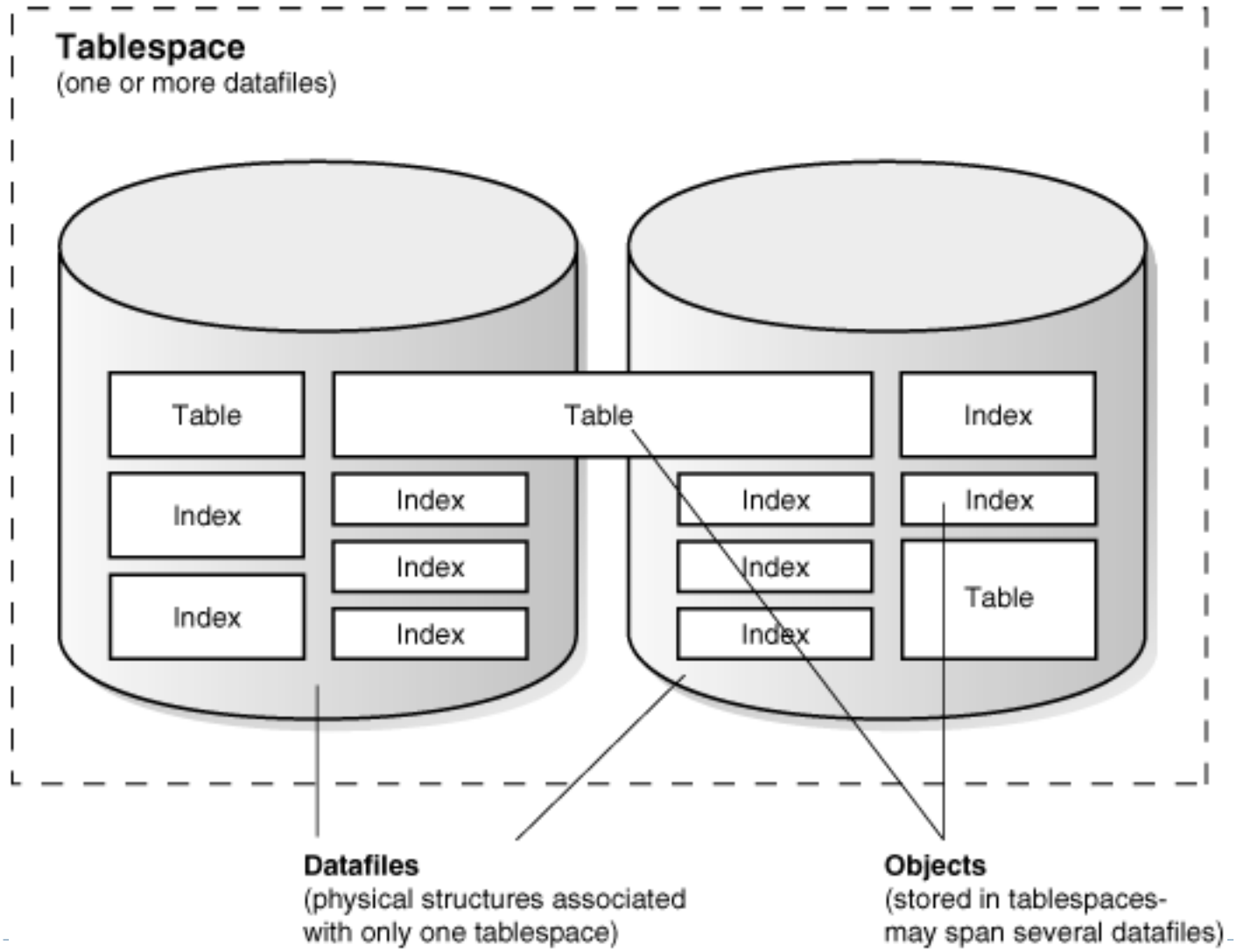
Files to Tablespaces on dbs3: added ones

FILE_NAME	TABLESPACE_N	BLOCKS
/home/oracle/app/oracle/oradata/dbs3/eoneil1.dbf	EONEIL1	12800
/home/oracle/app/oracle/oradata/dbs3/eoneil_setq.dbf	SETQ	131072
/home/oracle/app/oracle/oradata/dbs3/eoneil_setq250.dbf	SETQ250	19660800

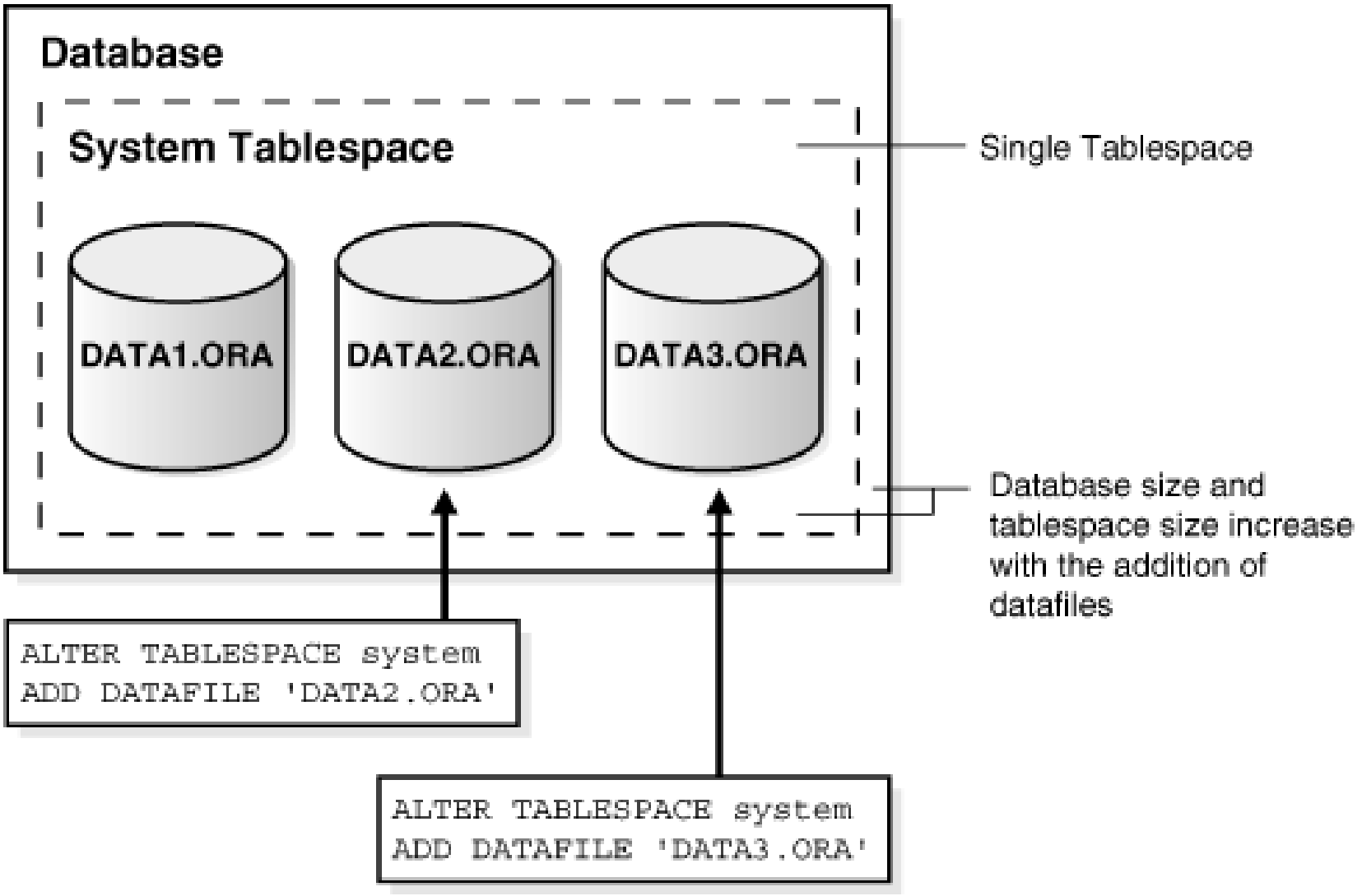
The SYSTEM tablespace

- Every Oracle database contains a tablespace named SYSTEM, which Oracle creates automatically when the database is created.
- The SYSTEM tablespace is always online when the database is open.
- The SYSTEM tablespace always contains the data dictionary tables for the entire database.

From Oracle Docs



Enlarging a Database by Adding a Datafile to a Tablespace

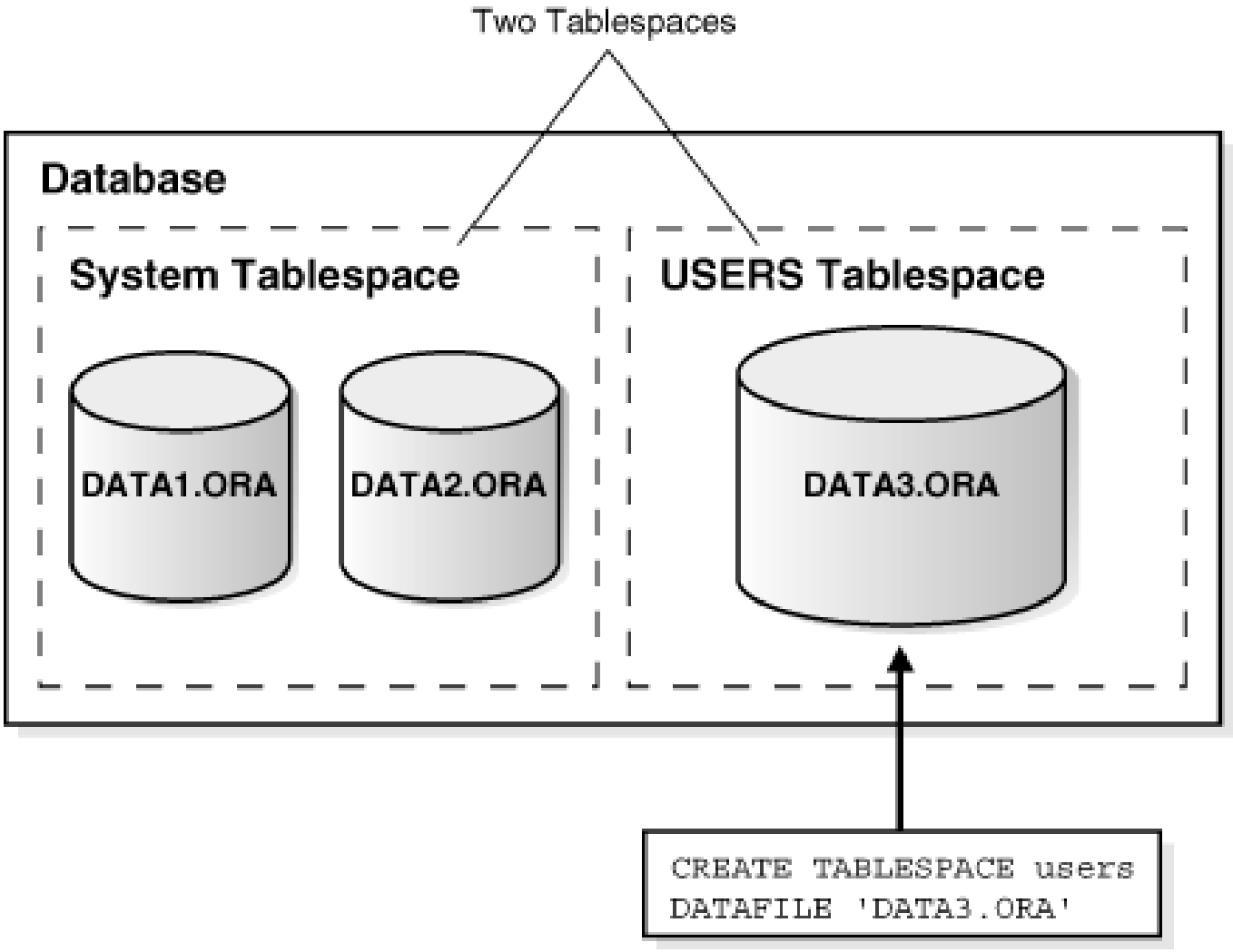


Example of use of Alter Tablespace

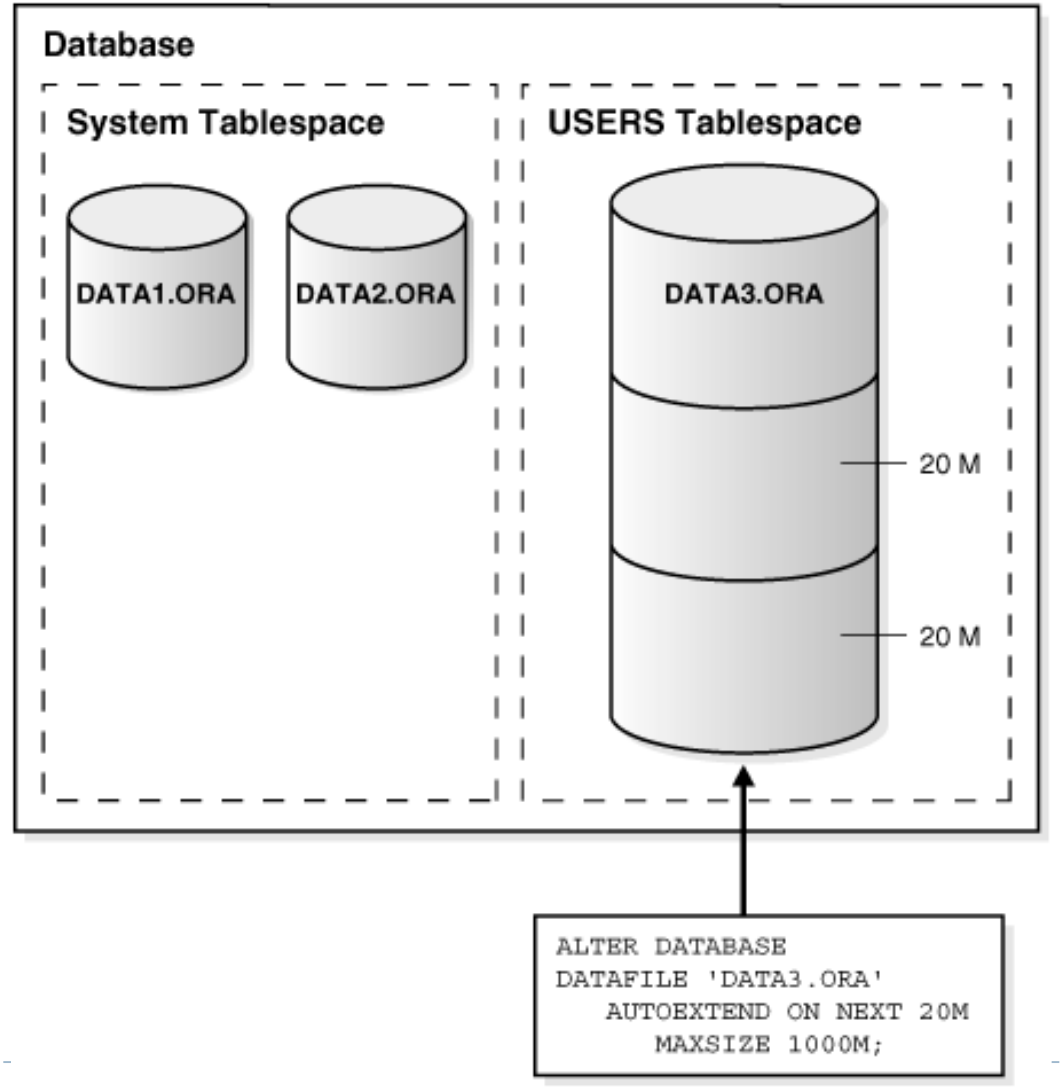
- Command used to expand our USERS tablespace on dbs2, where we had multiple disks instead of one huge RAID, so when one disk filled up we needed to expand to another one:

```
alter tablespace users
   add datafile
      '/disk/sd1f/data/oracle-10.1/dbs2/users02.dbf'
   size 4G;
```

Enlarging a Database by Adding a New Tablespace



Enlarging a Database by Dynamically Sizing Datafiles



Tables and indexes are in a particular tablespace

user_tables: catalog table for this user's (eoneil's) tables

```
SQL> select table_name, tablespace_name from user_tables;
TABLE_NAME                     TABLESPACE_NAME
------------------------------ ------------------------
ACCOUNT                        USERS
AGENTS                         USERS
APERF_RESULT                   USERS
```

…

user_indexes: catalog table for this user's tables

```
SQL> select index_name, tablespace_name from user_indexes;
INDEX_NAME                     TABLESPACE_NAME
------------------------------ ------------------------
BITS1                          USERS
BITS2                          USERS
K100X                          USERS
```

- … not an accident: account eoneil has default tablespace USERS

Create table can specify tablespace

- CREATE TABLE [schema.]tablename
- (coldef | table_constraint}
- {, coldef | table_constraint, . . .}
- [TABLESPACE tblspname]
- [STORAGE…] ← will cover later today
- [PCTFREE n] [PCTUSED n] ← for pages of table
- [other clauses] ←partitioning support is in here
- [AS subquery]
- This tablespace will override the default for the user
- Create index is similar

PCTFREE and PCTUSED for table

- PCTFREE n, n goes from 0 to 99, default 10.
- PCTUSED n, n goes from 1 to 99, default 40.
- The PCTUSED n clause specifies a condition where if page gets empty enough, inserts will start again!
- Require PCTFREE + PCTUSED < 100, or invalid.
- Example, if PCTFREE 10 PCTUSED 80, then stop inserts when >90% full, start again when <80% full.

Uses of tablespaces: control over disk resources

- In a two-disk system, can use one disk for table, other for index to speed up range searches
- Put table in tablespace USERS, composed of files on one disk, create tablespace USERIND for indexes, composed of file(s) on other disk.
- In a shared system, put one project on high-end disks made into one tablespace using RAID, another project on cheap disks made into another tablespace, also using RAID.
- With RAID, can mix tables and indexes pretty freely.

Block Size (i.e., page size)

- "Oracle recommends smaller Oracle Database block sizes (2 KB or 4 KB) for online transaction processing (OLTP) or mixed workload environments and larger block sizes (8 KB, 16 KB, or 32 KB) for decision support system (DSS) workload environments" from Burleson
- How is this block size specified by the DBA?
- You might expect it to be specified by the tablespace, but it's more central than that:
- The block size determines the page buffer size in the all-important database page buffer
- So most Oracle installations have a single page size

Finding the block size of an Oracle DB

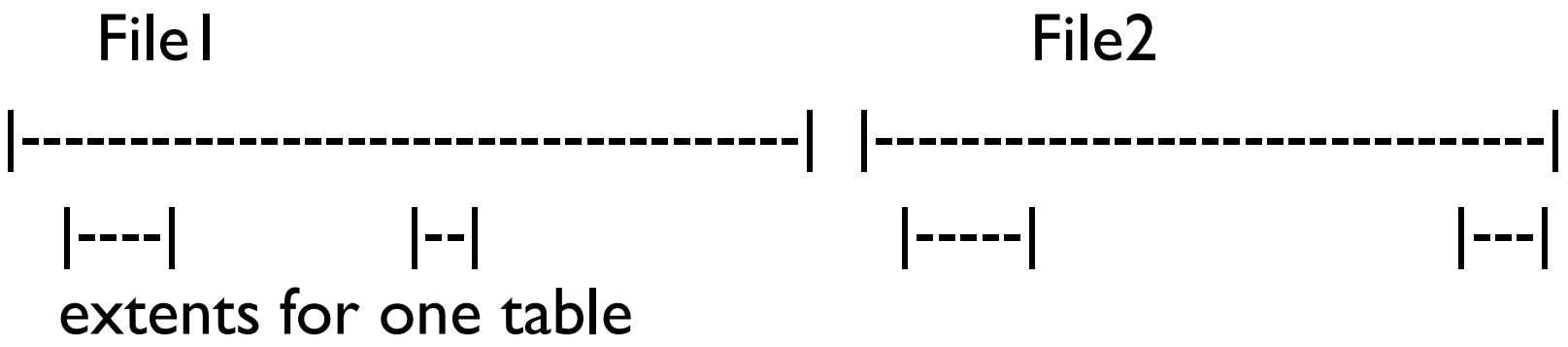
```
SQL> select tablespace_name, block_size from
dba_tablespaces;

TABLESPACE_NAME                BLOCK_SIZE
------------------------------ ----------
SYSTEM                               8192
UNDOTBS1                             8192
SYSAUX                               8192
TEMP                                 8192
USERS                                8192
CASPAR                               8192
```

- So we see it's 8KB, larger than recommended for OLTP,
- But small for DSS, i.e., a compromise.

Extents of disk in Oracle

An extent is a (hopefully contiguous) part of a file, composed of a whole number of blocks/pages.

One tablespace made of two files:



extents for one table

Note extents can be of different sizes—by default they get bigger and bigger as the table grows.

Goal: less seeking because lots of related data is close by on disk

STORAGE clause of Create Table

[STORAGE ([INITIAL n [K|M|G]] [NEXT n [K|M|G]] [MINEXTENTS n] [PCTINCREASE n])]

INITIAL n: size in bytes of initial extent (default 5 pages)

NEXT n: size in bytes of next extent (default 5 pages)

PCTINCREASE n: increase from one extent to next, starting from third one. (default 50%)

- MINEXTENTS n: start at creation with this number of extents; used when know initial use will be very large

DEFAULT STORAGE clause of Create Tablespace

[DEFAULT STORAGE ([INITIAL n [K|M|G]] [NEXT n [K|M|G]] [MINEXTENTS n] [PCTINCREASE n])]

- Sets defaults for create table and create index in that tablespace
- Example: tablespace for warehouse tables should have larger extents by default
- DEFAULT STORAGE (INITIAL 10M NEXT 10M)
- Downside: a little side table takes 10M
- But 10M in a warehouse is trivial.

Other Database Files

- So far, considered the files holding pages of data for tables and indexes
- Other important files: saw redo*.dbf, undotbs01.dbf
- Redo log files: information that allows for crash recovery
  - The current such file is appended to constantly as the DB is changed, read only in crash recovery
  - The system cuts over to another of these files periodically
  - For a serious database, should be mirrored, since otherwise is a single point of failure
  - Striping not helpful for its sequential i/o, so use simple mirroring here
- Undo tablespace: information that allows for rollbacks and also snapshots for efficient reads
  - This data is written and read, more like the DB data, so held in a tablespace, unlike the redo log

Example: 1TB Database with 2000 ops/s

- Burleson says: ***Size first for IO capacity, then for volume.***
- 2000 ops/sec means 20 7200 rpm disks or 10 15Krpm disks, roughly, not counting parity disks or mirrors or spares
- So say 12 15Krpm disks in a RAID1+0, plus 12 mirrors for data
- 2 disks for mirrored log, RAID 1, plus 5 spares.
  - Smart RAID controller with memory cache best here
- 1TB/12 = 83 GB, so 143GB disks are fine for data.

1TB example

- Build RAID for data
  - End up with new empty filesystem /disk/raida
- Build RAID for redo log
  - End up with new empty filesystem /disk/raidb
- Create tablespace DBDATA and let Oracle create one huge file /disk/raida/dbdata.dbf
- Change database to use redo logs on /disk/raidb:
  - alter database add logfile group 5 (‘/disk/raid/redo05a.log',
  - ‘/disk/raid/redo05b.log') size 500m;
- Create tables and indexes in tablespace DBDATA

Oracle Project Account

- Create an Oracle account for the project, and make its default tablespace be DBDATA

  create user myproject identified by pw default tablespace dbdata
  temporary tablespace temp;

- This simplifies the createdb.sql, etc.
- Makes it less likely that someone accidentally makes a table in tablespace USERS for the project, off on wrong disks.
- Make a project rule that DBA actions are done as this user
- If user already exists:

  alter user myproject default tablespace dbdata;

Summary

- Hierarchy of data containers:
- Files containing blocks/pages 8KB each on dbs3
- Tablespace: some number of files ganged together
- Extent: some number of blocks in a certain file and thus in a certain tablespace, by default, bigger and bigger as a table grows
- Table or Index: some number of extents all in the same tablespace
- Separately: redo log file, no page structure, just append records describing DB changes.