Memory Structures

Mag. Thomas Griesmayer

RECO   PMON   SMON

System Global Area

Database Buffer Cache   **Redo Log Buffer**

User Process

User Processes

Shared Server Process

Dedicated Server Process

DBW0

ARC0

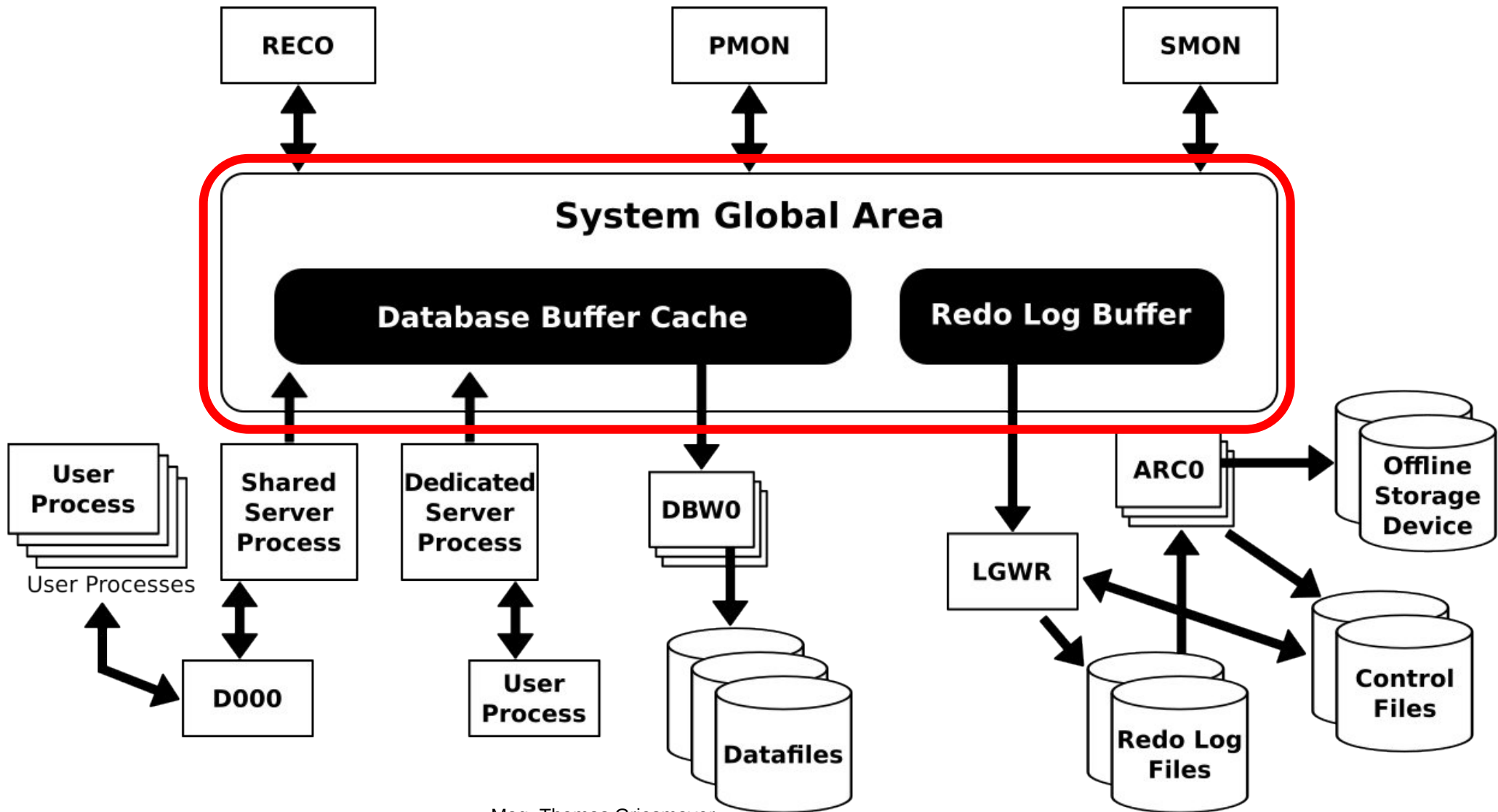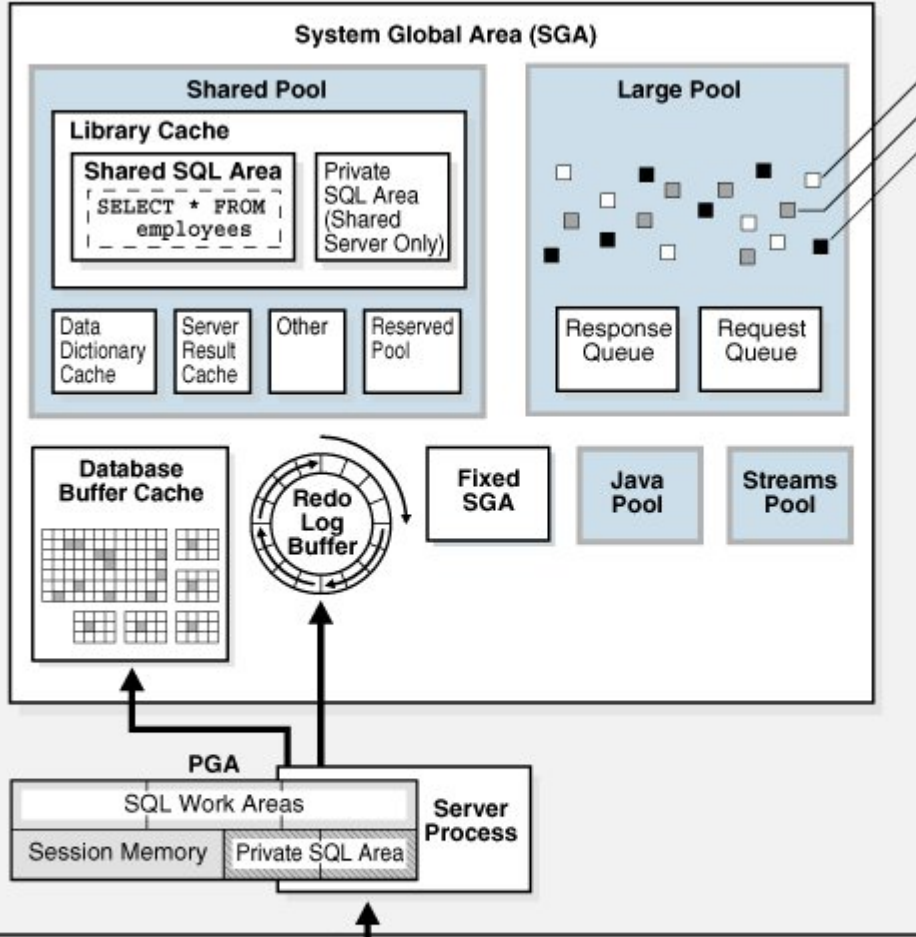Offline Storage Device

LGWR

D000

User Process

Datafiles

Redo Log Files

Control Files

Mag. Thomas Griesmayer

Oracle uses memory to store information:

- Program code

- Connected sessions

- Information about program execution (current state of a query, number of rows being fetched)

- Information for other Oracle processes (locking information)

- Cached data

- Memory structures:

  – System Global Area (SGA) - shared by all server and background processes

  – Program Global Areas (PGA) - there is one PGA for each server and background.
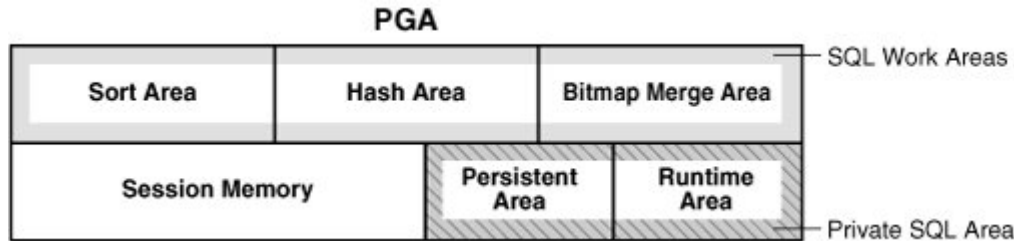
Memory structures

Instance

System Global Area (SGA)

Shared Pool

Library Cache

Shared SQL Area

```
SELECT * FROM
employees
```

Private SQL Area (Shared Server Only)

Data Dictionary Cache | Server Result Cache | Other | Reserved Pool

Large Pool

- Free Memory
- I/O Buffer Area
- UGA

Response Queue | Request Queue

Database Buffer Cache

Redo Log Buffer

Fixed SGA

Java Pool

Streams Pool

PGA

SQL Work Areas

Session Memory | Private SQL Area

Server Process

Client Process

https://docs.oracle.com/cd/E11882_01/server.112/e40540/memory.htm#CNCPT77 78 (8.9.2019)

Mag. Thomas Griesmayer

- The system global area (SGA) is a memory structure that contains data and control information for one Oracle database instance.

- When multiple users are concurrently connected to the same instance, then the data in the instance is shared among the users.

- A private SQL area holds information about a parsed SQL statement and other session-specific information for processing.

- A work area is a private allocation of PGA memory used for memory-intensive operations. For example, a sort operator uses the sort area to sort a set of rows. Similarly, a hash join operator uses a hash area to build a hash table from its left input, whereas a bitmap merge uses the bitmap merge area to merge data retrieved from scans of multiple bitmap indexes.

PGA

SQL Work Areas		
Sort Area	Hash Area	Bitmap Merge Area

Private SQL Area		
Session Memory	Persistent Area	Runtime Area

https://docs.oracle.com/cd/E11882_01/server.112/e40540/memory.htm#CNCPT7778 (8.9.2019)

- The database buffer cache holds copies of data blocks read from datafiles.

- All user processes connected to the instance share the database buffer cache.

- The write list contains blocks that have been modified (dirty) but have not yet been written to disk.

- The dirty buffers list contains dirty blocks that have not yet been moved to the write list.

- Free buffers do not contain any useful data.

- Pinned buffers are currently being accessed.

Buffer cache

- When an Oracle process accesses a buffer, the process moves the buffer to the most recently used (MRU) end of the LRU list. As more buffers are continually moved to the MRU end of the LRU list, dirty buffers age toward the LRU end of the LRU list.

- An Oracle user process requires a particular piece of data, it searches in the database buffer cache.

- cache hit

- cache miss - copy the data block from a datafile into the buffer

SGA

0
```
4 Hans   €300
6 Susi   €270
```

1
```
1 Alex   €800
3 Franz  €900
```

2
```

```

Harddisk

9
```
1 Alex   €800
3 Franz  €900
```

A
```
4 Hans   €300
6 Susi   €270
```

B
```
9 Paul   €300
0 Peter  €560
```

C
```
5 Karl   €280
8 Thom   €560
```

D
```
7 Max    €880
2 Kurt   €870
```

E
```

```

LRU: 1   0
Dirty:
Free: 2

SGA

€350

0
```
4 Hans    €300
6 Susi    €270
```

1
```
1 Alex    €800
3 Franz   €900
```

2
```
9 Paul    €350
0 Peter   €560
```

Harddisk

9
```
1 Alex    €800
3 Franz   €900
```

A
```
4 Hans    €300
6 Susi    €270
```

B
```
9 Paul    €300
0 Peter   €560
```

C
```
5 Karl    €280
8 Thom    €560
```

D
```
7 Max     €880
2 Kurt    €870
```

E
```

```

```
UPDATE employee
SET     salary = salary+50
WHERE   employee_id = 9;
COMMIT;
```

LRU: **1**   **0**   **2**

Dirty: **2**
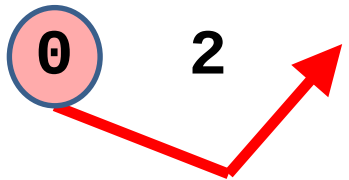
Free: ❌

Mag. Thomas Griesmayer

SGA

0
```
4 Hans    €300
6 Susi    €270
```

1
```
1 Alex    €800
3 Franz   €900
```

2
```
9 Paul    €350
0 Peter   €560
```

Harddisk

9
```
1 Alex    €800
3 Franz   €900
```

A
```
4 Hans    €300
6 Susi    €270
```

B
```
9 Paul    €300
0 Peter   €560
```

C
```
5 Karl    €280
8 Thom    €560
```

D
```
7 Max     €880
2 Kurt    €870
```

E
```

```

```
SELECT  *
FROM    employee
WHERE   employee_id = 4;
COMMIT;
```

LRU: **1**   **0**   **2**
Dirty: 2
Free:

SGA

| 0 | 4 Hans €300
6 Susi €270 | 1 | ~~1 Alex €800~~
~~3 Franz €900~~ | 2 | 9 Paul €350
0 Peter €560 |

Harddisk

| 9 | 1 Alex €800
3 Franz €900 | A | 4 Hans €300
6 Susi €270 | B | 9 Paul €300
0 Peter €560 |
| C | 5 Karl €280
8 Thom €560 | D | 7 Max   €880
2 Kurt €870 | E | |

FREE ONE BLOCK

LRU: ~~1~~   **2**   **0**
Dirty: 2
Free: 1

SGA

0
```
4 Hans    €400
6 Susi    €270
```

1

2
```
9 Paul   €350
0 Peter  €560
```

Harddisk

9
```
1 Alex    €800
3 Franz   €900
```

A
```
4 Hans    €300
6 Susi    €270
```

B
```
9 Paul   €300
0 Peter  €560
```

C
```
5 Karl    €280
8 Thom    €560
```

D
```
7 Max     €880
2 Kurt    €870
```

E

```
UPDATE employee
SET     salary = salary+100
WHERE   employee_id = 4;
COMMIT;
```

LRU: **2**   **0**
Dirty: **2**   **0**
Free: **1**

SGA

0
4 Hans	€400
6 Susi	€270

1

2
~~9 Paul~~	~~€350~~
~~0 Peter~~	~~€560~~

Harddisk

9
1 Alex	€800
3 Franz	€900

A
4 Hans	€300
6 Susi	€270

B   **€350**
9 Paul	€~~350~~
0 Peter	€560

C
5 Karl	€280
8 Thom	€560

D
7 Max	€880
2 Kurt	€870

E

FREE ONE BLOCK

LRU: ~~2~~ 0

Dirty: ~~2~~ 0

Free: 1 **2**

SGA

0
```
4 Hans    €400
6 Susi    €270
```

1
```
9 Paul    €350
0 Peter   €560
```
(0 Peter €560 struck through)

2
(empty)

Harddisk

9
```
1 Alex    €800
3 Franz   €900
```

A
```
4 Hans    €300
6 Susi    €270
```

B
```
9 Paul    €350
0 Peter   €560
```

C
```
5 Karl    €280
8 Thom    €560
```

D
```
7 Max     €880
2 Kurt    €870
```

E
(empty)

```
DELETE FROM employee
WHERE  employee_id = 0;
COMMIT;
```

LRU: **0**   **1**
Dirty: **0**   **1**
Free: ~~**1**~~   **2**

- The redo log holds information about changes made to the database.
- The background process LGWR writes the redo log buffer to the active redo log file on disk.

- The shared pool portion of the SGA contains the library cache, the dictionary cache, buffers for parallel execution messages, and control structures.

- The shared SQL area contains the parse tree and execution plan for a given SQL statement.

- Oracle saves memory and CPU by using one shared SQL area for SQL statements run multiple times.

system
user

```
SELECT *
FROM    V$SGA

NAME                        VALUE
-------------------- ----------
Fixed Size              1344840
Variable Size         369101496
Database Buffers       79691776
Redo Buffers            6008832
```