Oracle Database Transactions

Oracle Database Transactions

- A database transaction consists of one of the following:
  - DML statements that constitute one consistent change to the data
  - One DDL statement
  - One data control language (DCL) statement

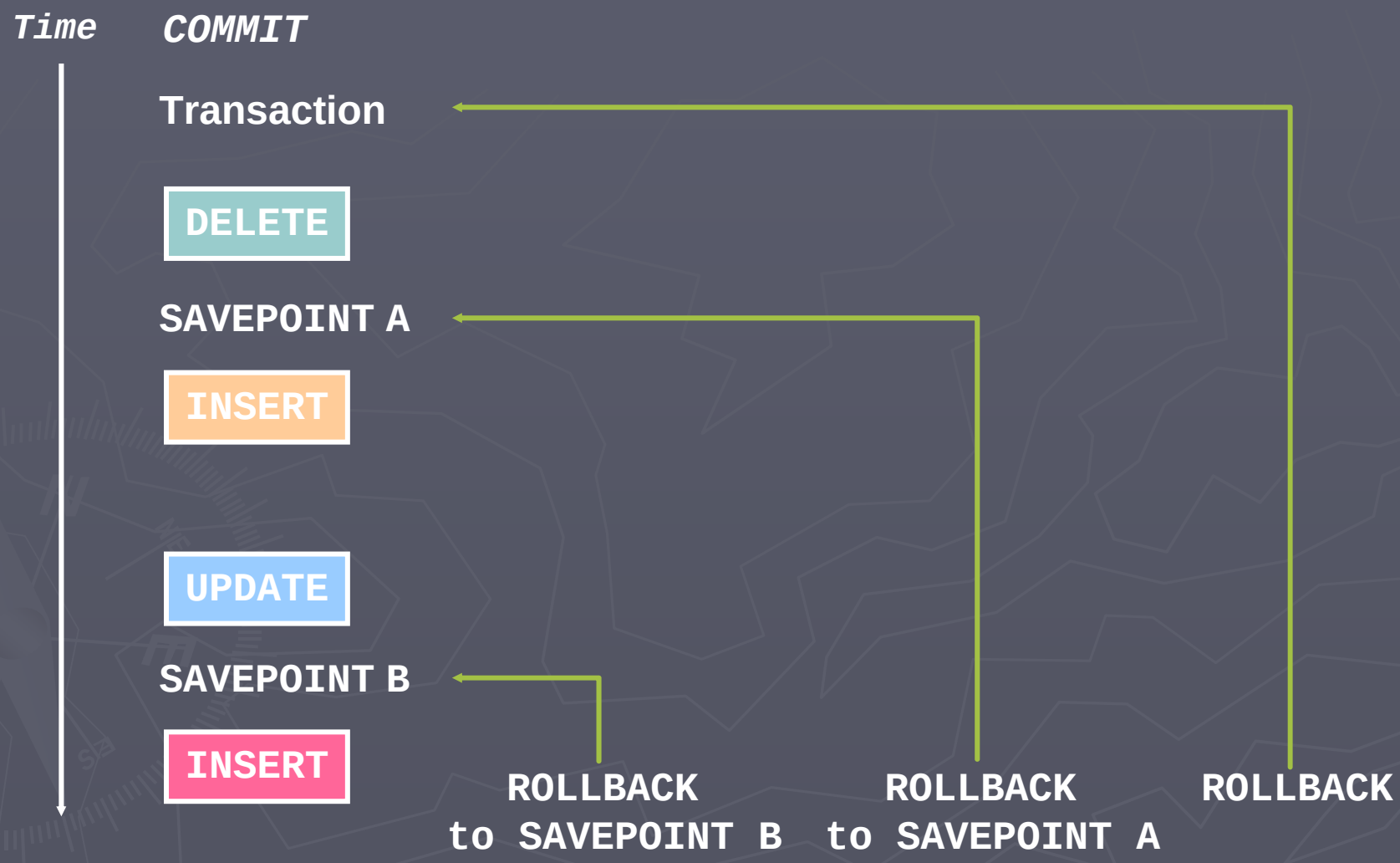
Oracle Database Transactions

- Begin when the first DML SQL statement is executed.
- End with one of the following events:
  - A `COMMIT` or `ROLLBACK` statement is issued.
  - A DDL or DCL statement executes (automatic commit).
  - The system crashes.

Advantages of COMMIT and ROLLBACK Statements

- With COMMIT and ROLLBACK statements, you can:
  - Ensure data consistency
  - Preview data changes before making changes permanent
  - Group logically related operations

Controlling Transactions



Rolling Back Changes to a Marker

- Create a marker in a current transaction by using the SAVEPOINT statement.
- Roll back to that marker by using the ROLLBACK TO SAVEPOINT statement.

```
UPDATE...
SAVEPOINT update_done;
Savepoint created.
INSERT...
ROLLBACK TO update_done;
Rollback complete.
```

Implicit Transaction Processing

- An automatic commit occurs under the following circumstances:
  - DDL statement is issued
  - DCL statement is issued
- An automatic rollback occurs under an abnormal termination of *i*SQL*Plus or a system failure.

State of the Data Before COMMIT or ROLLBACK

- The previous state of the data can be recovered.
- The current user can review the results of the DML operations by using the SELECT statement.
- Other users *cannot* view the results of the DML statements by the current user.
- The affected rows are *locked*; other users cannot change the data in the affected rows.

State of the Data After COMMIT

- Data changes are made permanent in the database.
- The previous state of the data is permanently lost.
- All users can view the results.
- Locks on the affected rows are released; those rows are available for other users to manipulate.
- All savepoints are erased.

Committing Data

- Make the changes:

```
DELETE FROM employees
WHERE  employee_id = 99999;
1 row deleted.

INSERT INTO departments
VALUES (290, 'Corporate Tax', NULL, 1700);
1 row created.
```

- Commit the changes:

```
COMMIT;
Commit complete.
```

State of the Data After ROLLBACK

- Discard all pending changes by using the ROLLBACK statement:
  - Data changes are undone.
  - Previous state of the data is restored.
  - Locks on the affected rows are released.

```
DELETE FROM copy_emp;
22 rows deleted.
ROLLBACK ;
Rollback complete.
```

State of the Data After ROLLBACK

```
DELETE FROM test;
25,000 rows deleted.

ROLLBACK;
Rollback complete.

DELETE FROM test WHERE  id = 100;
1 row deleted.

SELECT * FROM   test WHERE  id = 100;
No rows selected.

COMMIT;
Commit complete.
```

Statement-Level Rollback

- If a single DML statement fails during execution, only that statement is rolled back.
- The Oracle server implements an implicit savepoint.
- All other changes are retained.
- The user should terminate transactions explicitly by executing a `COMMIT` or `ROLLBACK` statement.

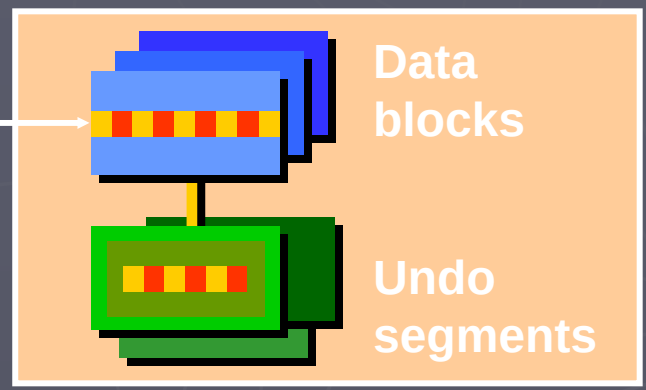
Read Consistency

- Read consistency guarantees a consistent view of the data at all times.
- Changes made by one user do not conflict with changes made by another user.
- Read consistency ensures that on the same data:
  - Readers do not wait for writers
  - Writers do not wait for readers

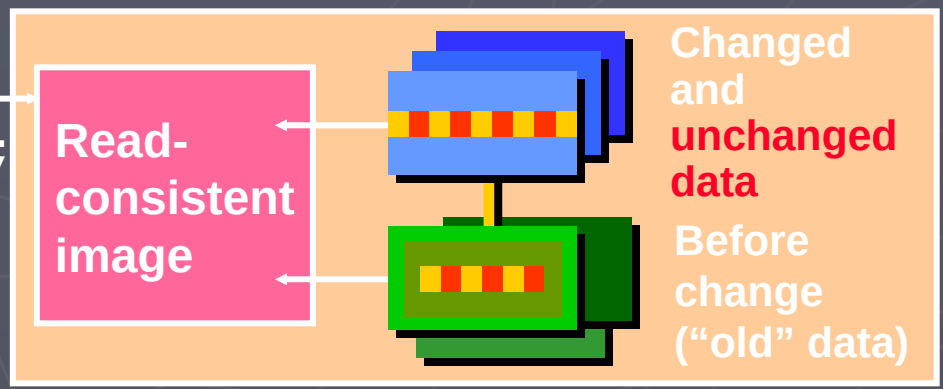
Implementation of Read Consistency

User A

```
UPDATE employees
SET    salary = 7000
WHERE  last_name = 'Grant';
```



```
SELECT  *
FROM userA.employees;
```



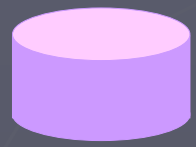
User B

Oracle Architectural Components

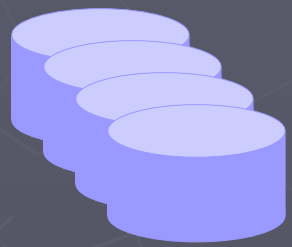
Oracle DB Architecture: Overview

- The Oracle database consists of two main components:
  - The database or the physical structures
  - The instance or the memory structures

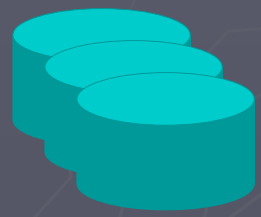
Database Physical Architecture



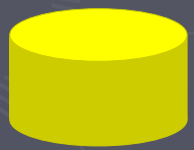
Control files



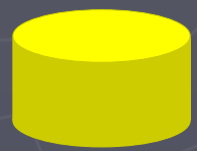
- Data files



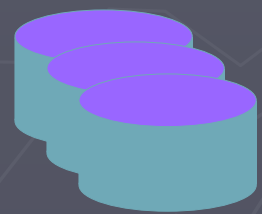
- Online redo log files



- Parameter file



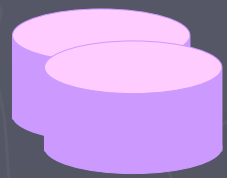
- Password file



- Archive log files

Control Files

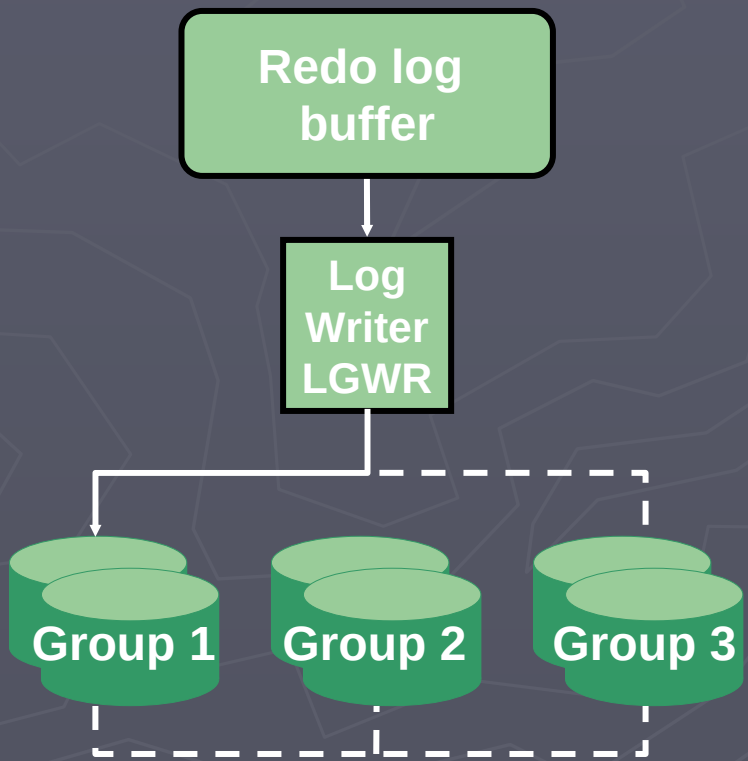
- Contains physical database structure information
- Multiplexed to protect against loss
- Read at mount stage



Control files

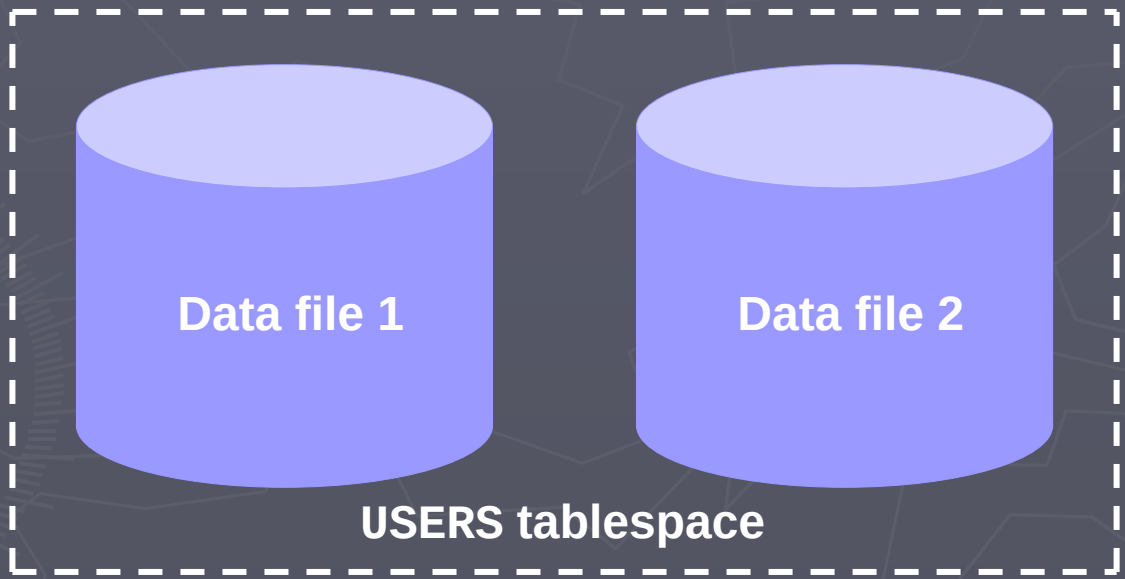
Redo Log Files

- Record changes to the database
- Multiplexed to protect against loss



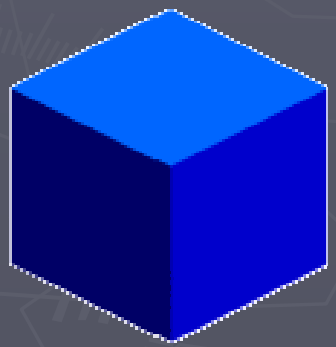
Tablespaces and Data Files

- Tablespaces consist of one or more data files.
- Data files belong to only one tablespace.

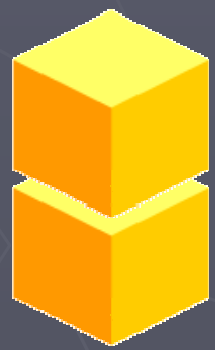


Segments, Extents, and Blocks

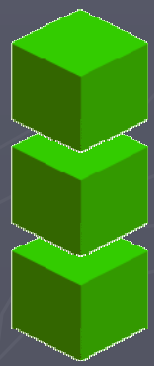
- Segments exist within a tablespace.
- Segments consist of a collection of extents.
- Extents are a collection of data blocks.
- Data blocks are mapped to OS blocks.



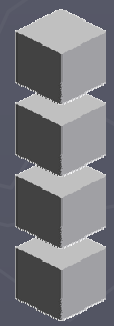
Segment



Extents

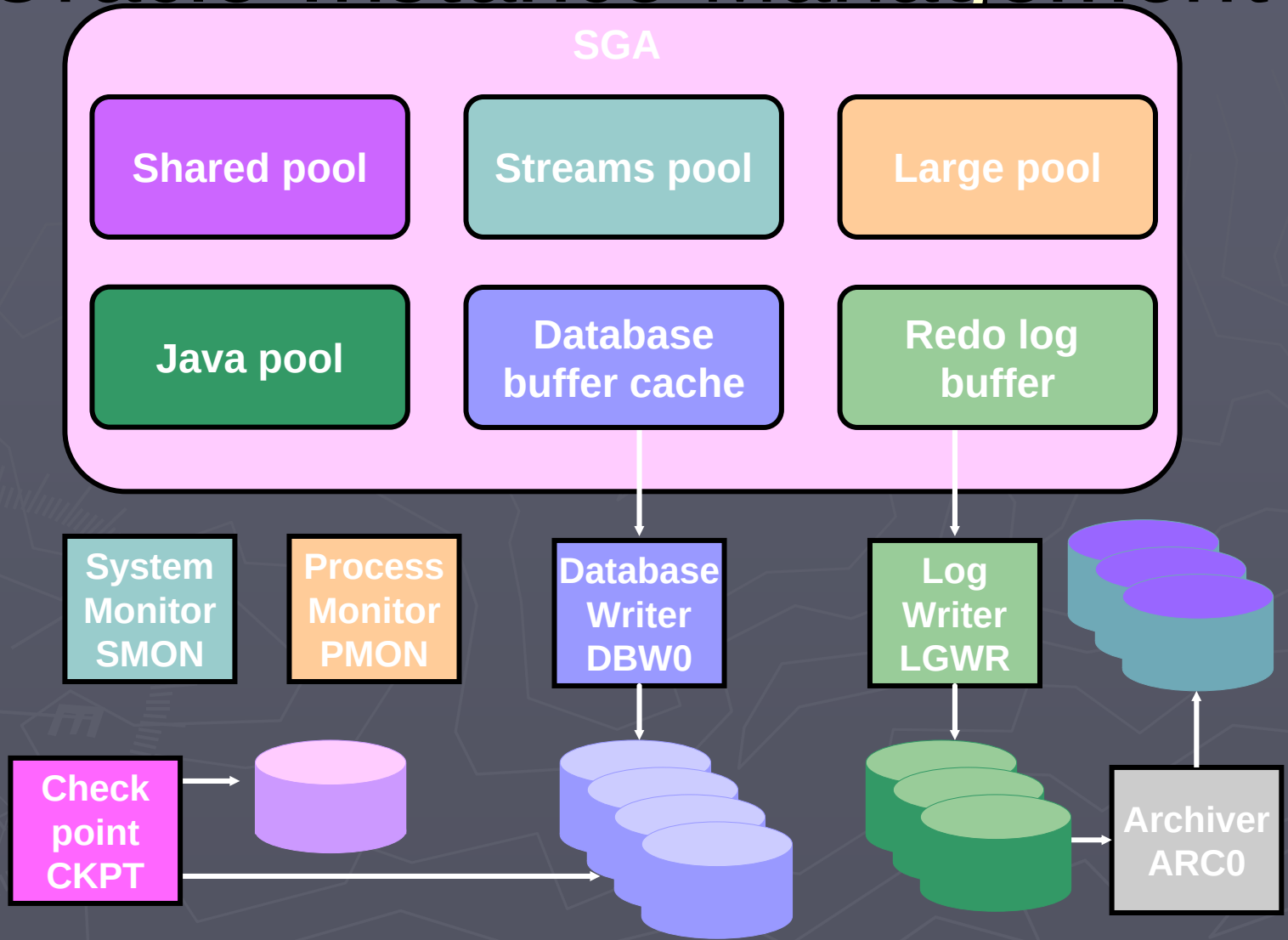


Data blocks

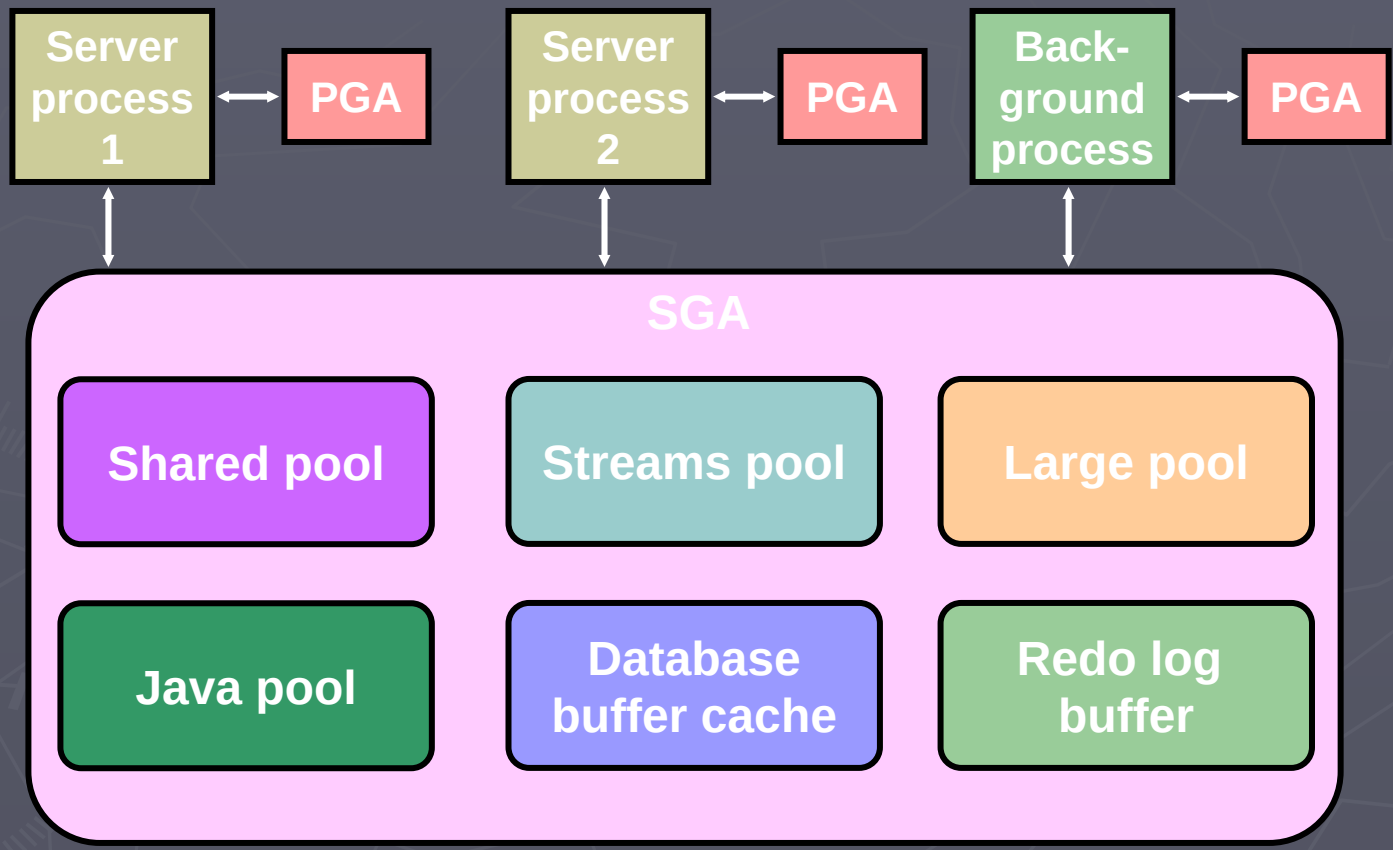


OS blocks

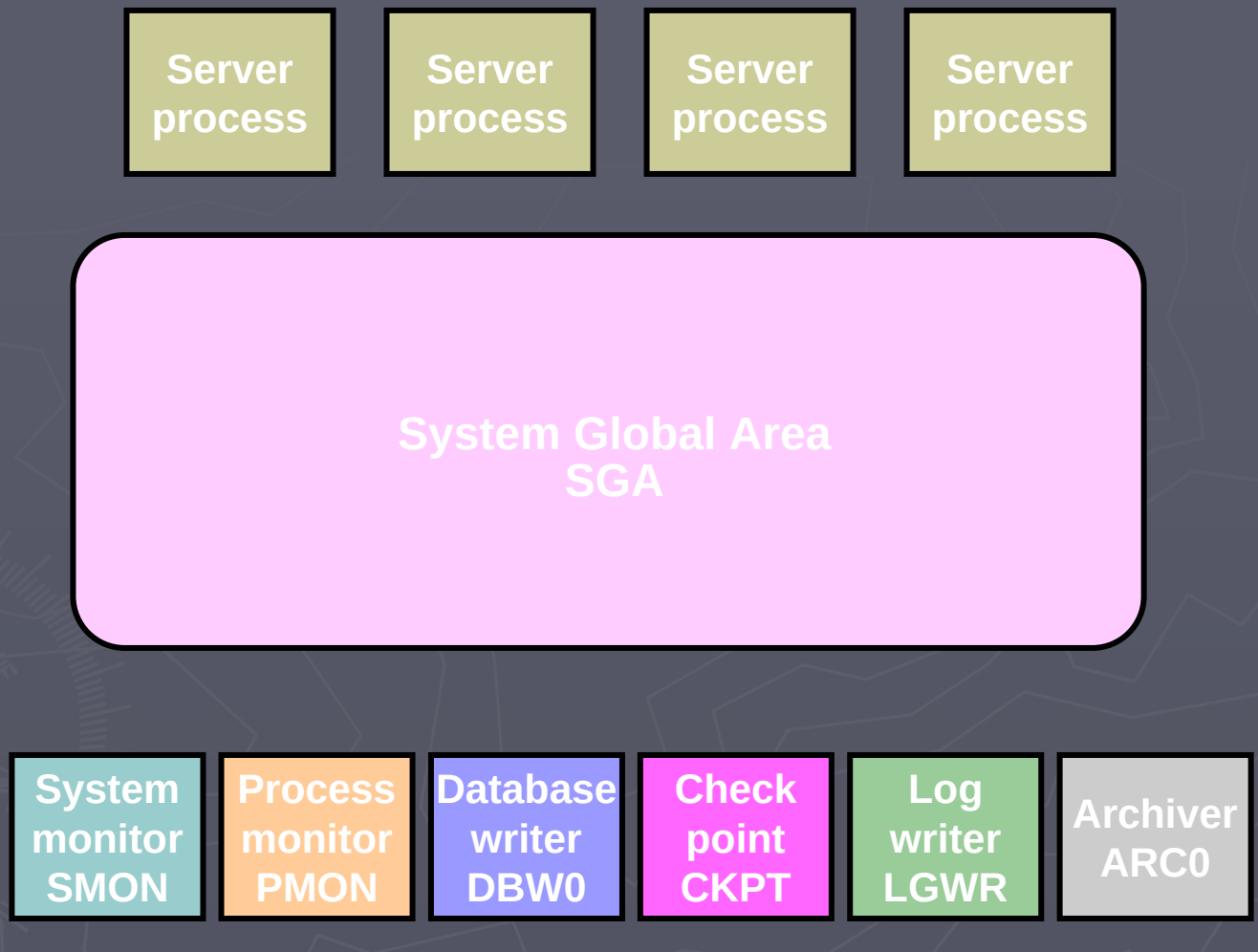
Oracle Instance Management



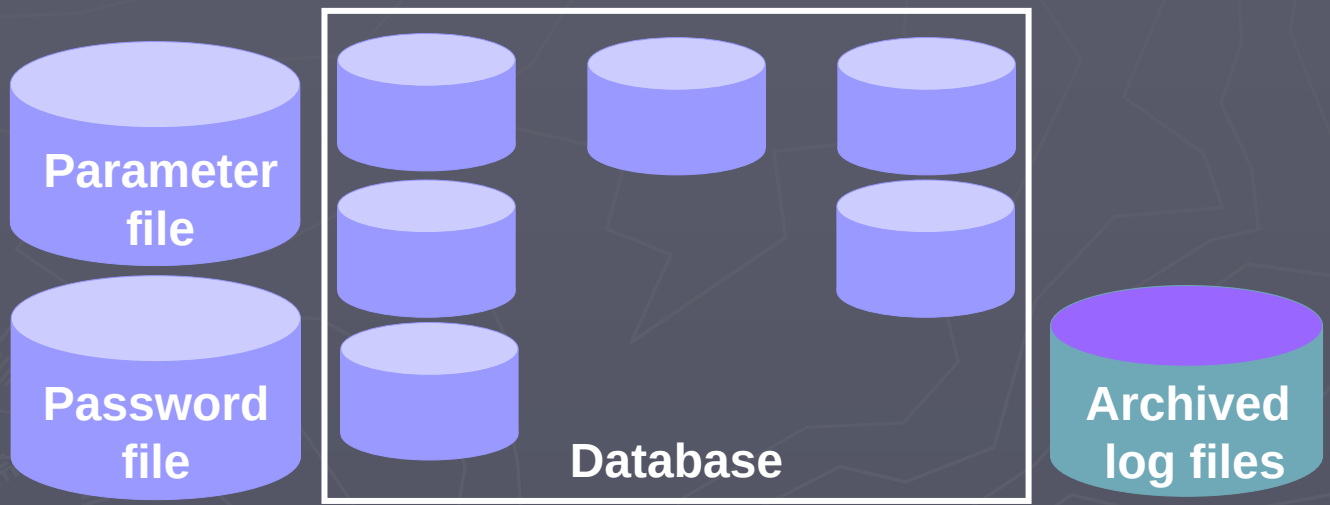
Oracle Memory Structures



Oracle Processes



Other Key Physical Structures



Processing a SQL Statement

- Connect to an instance using:
  - The user process
  - The server process
- The Oracle server components that are used depend on the type of SQL statement:
  - Queries return rows
  - DML statements log changes
  - Commit ensures transaction recovery
- Some Oracle server components do not participate in SQL statement processing.

Processing a Query

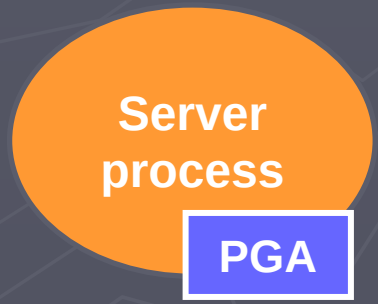
- Parse:
  - Search for identical statement
  - Check syntax, object names, and privileges
  - Lock objects used during parse
  - Create and store execution plan
- Execute: Identify rows selected
- Fetch: Return rows to user process

The Shared Pool

- The library cache contains the SQL statement text, parsed code, and execution plan.
- The data dictionary cache contains table, column, and other object definitions and privileges.
- The shared pool is sized by `SHARED_POOL_SIZE`.

Program Global Area (PGA)

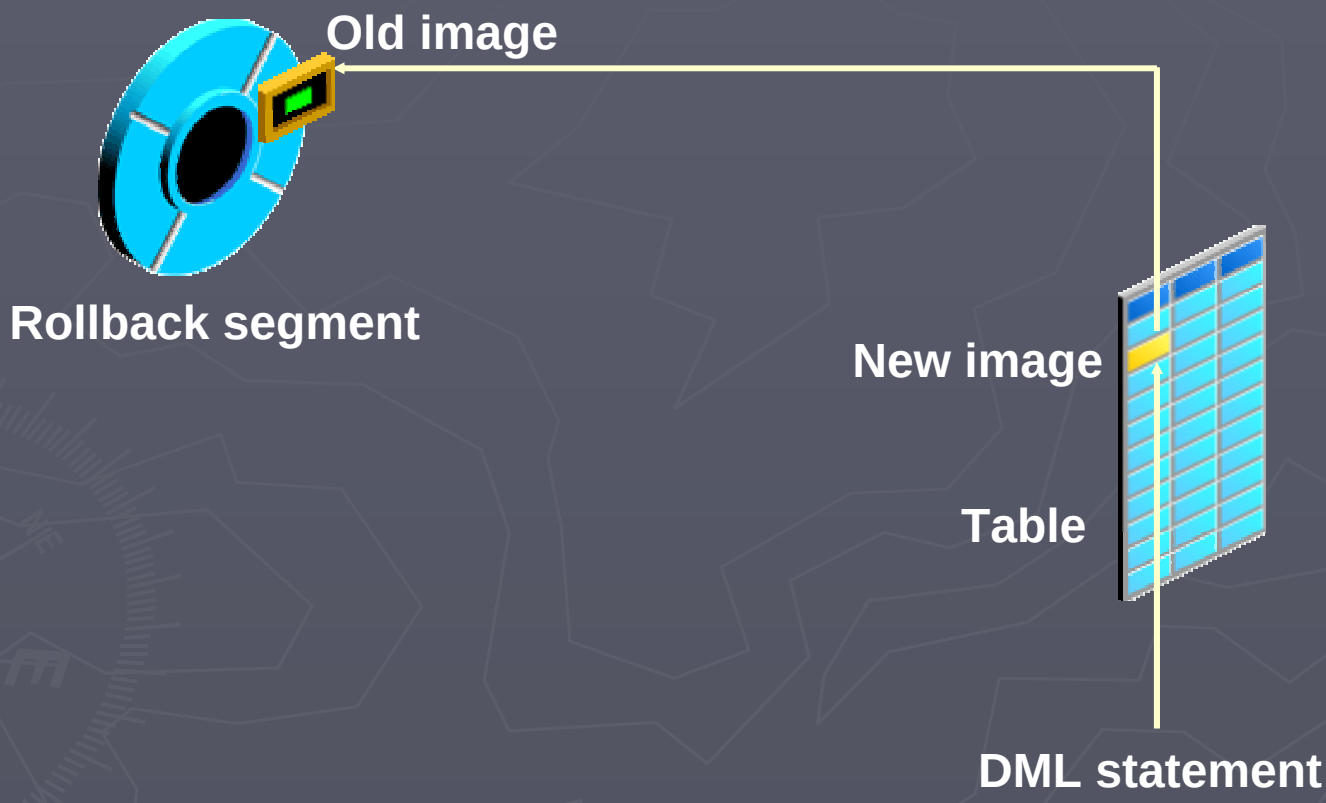
- Not shared
- Writable only by the server process
- Contains:
  - Sort area
  - Session information
  - Cursor state
  - Stack space



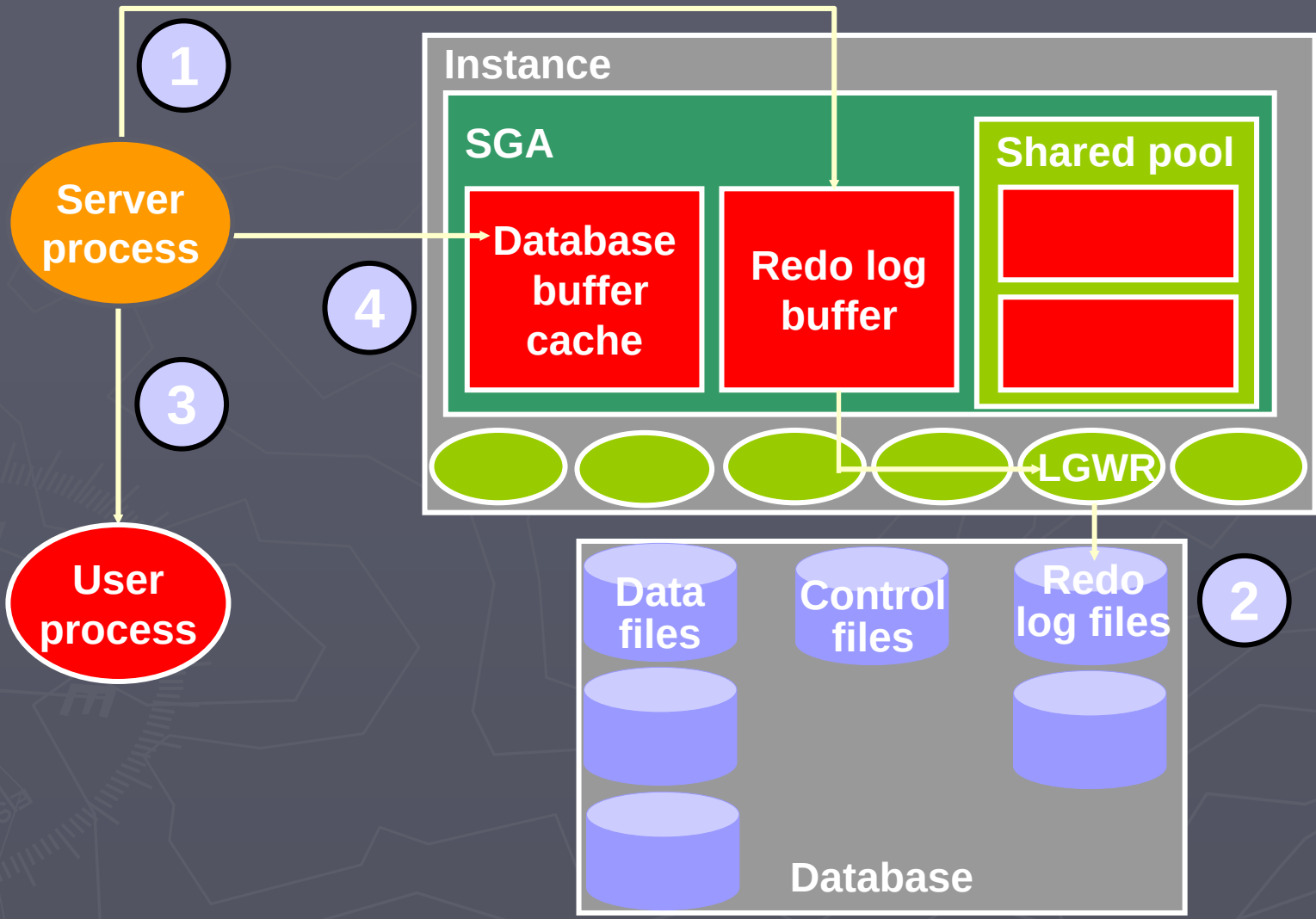
Redo Log Buffer

- Has its size defined by LOG_BUFFER
- Records changes made through the instance
- Is used sequentially
- Is a circular buffer

Rollback Segment



COMMIT Processing



CAP Theorem

- Three properties of a system
  - Consistency (all copies have same value)
  - Availability (system can run even if parts have failed)
  - Partitions (network can break into two or more parts)
- Brewer's CAP "Theorem": You can have at most two of these three properties for any system