

Course code : **CSE2007**
Course title : **Database Management System**
Module : **5**
Topic : **5**

Recovery Concepts

Objectives

This session will give the knowledge about

- Recovery Concepts
- Deferred update techniques
- Immediate update techniques

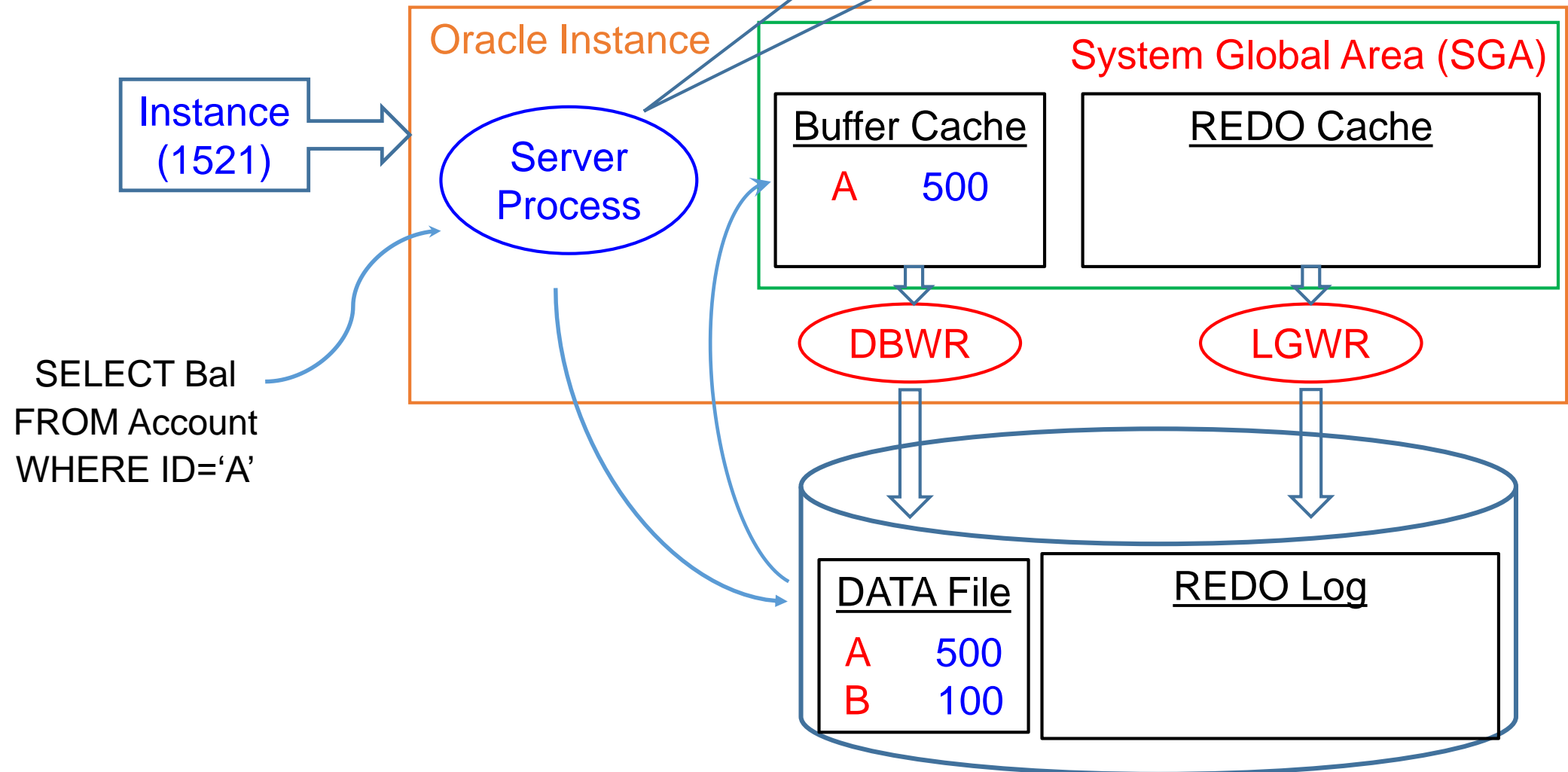
Introduction to Recovery

Recovery process restores database to most recent consistent state before time of failure with the information kept in system log

Typical recovery strategies

- Restore backed-up copy of database
- Identify any changes that may cause inconsistency
- Some operations may require redo
- Deferred update techniques
- Immediate update techniques
- Shadow Paging

Query Execution Process



Query Execution process

- A system global area (SGA) is a group of shared memory structures that contain data and control information for one instance in Oracle DB.
- If multiple users, multiple instance will be created so that multiple SGA will be created.
- Oracle automatically allocated the memory for an SGV when we start an instance.

Caching (buffering)

- DBMS cache: a collection of in-memory buffers
- Cache directory keeps track of which database items are in the buffers
- Cache buffers replaced (flushed) to make space for new items
- Dirty bit associated with each buffer in the cache Indicates whether the buffer has been modified
- Contents written back to disk before flush if dirty bit equals one

Type of Failures

- A Computer Failure (System Crash) - A hardware, software or network error happens during the execution of a transaction
- Transaction Error - Exceptions such as integer overflow or division by zero can occur and interrupt a transaction
- Exceptions detected by transaction - Insufficient balance and hence cancellation of withdrawal
- Concurrency Control enforcement - Transactions are in a deadlock
- Disk failure - Some disk blocks lose data because of disk failure
- Physical problems - fire accidents, flood etc

Deferred Update Technique

- Do not physically update the database until after transaction commits
- Undo is not needed; redo may be needed

Deferred Update Technique

CASE-1

T1

- 1 BEGIN T1
- 2 READ(A)
- 3 A=A-100
- 4 WRITE(A)
- 5 READ(B)
- 6 B=B+100
- 7 WRITE(B)
- 8 COMMIT

CASE-2

T1

- 1 BEGIN T1
- 2 READ(A)
- 3 A=A-100
- 4 WRITE(A)
- 5 ABORT

CASE-3

T1

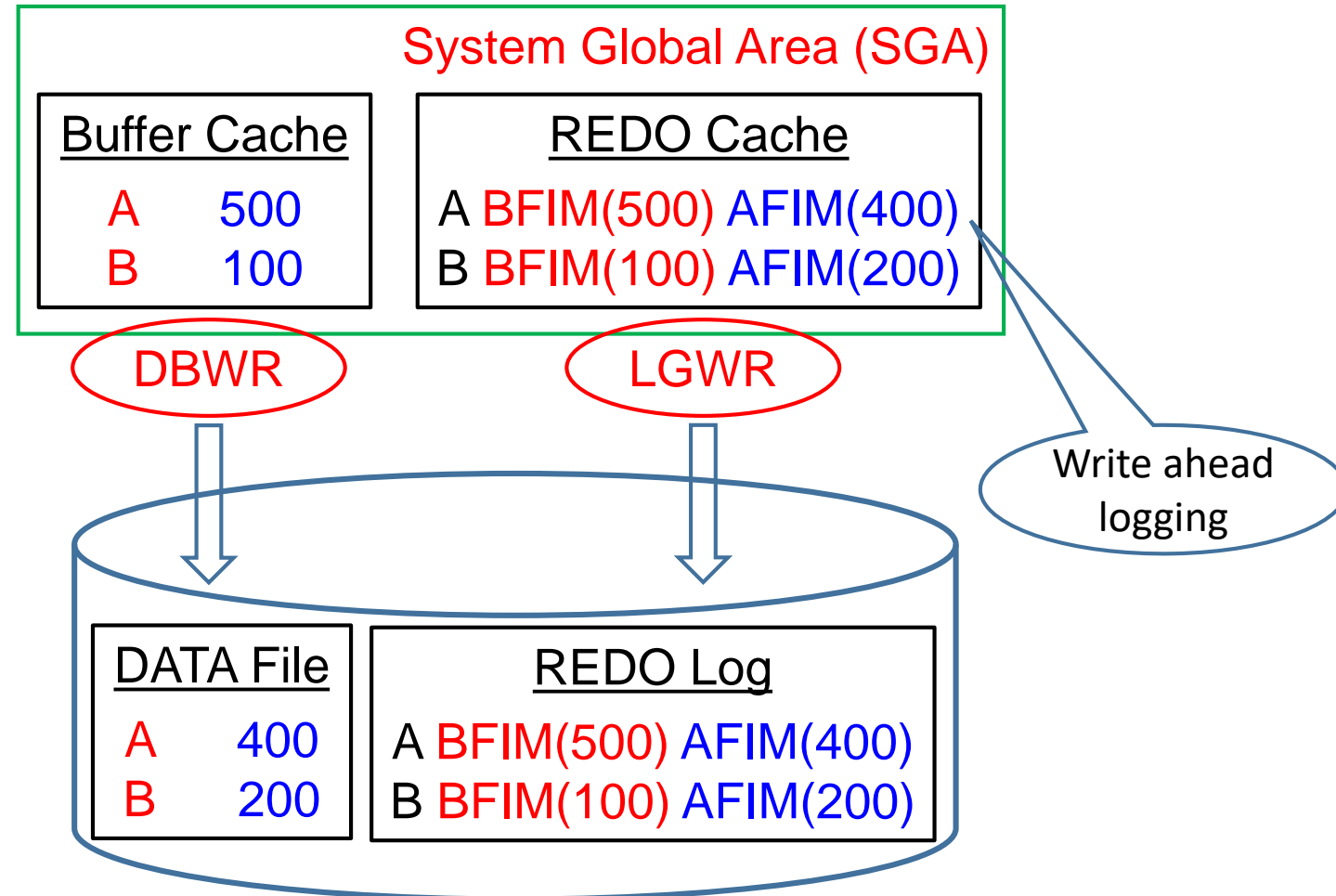
- 1 BEGIN T1
- 2 READ(A)
- 3 A=A-100
- 4 WRITE(A)
- 5 READ(B)
- 6 B=B+100
- 7 WRITE(B)
- 8 ABORT

Deferred Update Technique

```

2  READ(A)
3  A=A-100
4  WRITE(A)
5  READ(B)
6  B=B+100
7  WRITE(B)
8  COMMIT
  
```

BFIM(A) - Before Image (A) – Old value
 AFIM(A) - After Image (A) – New value



Deferred Update Technique

Write-ahead logging

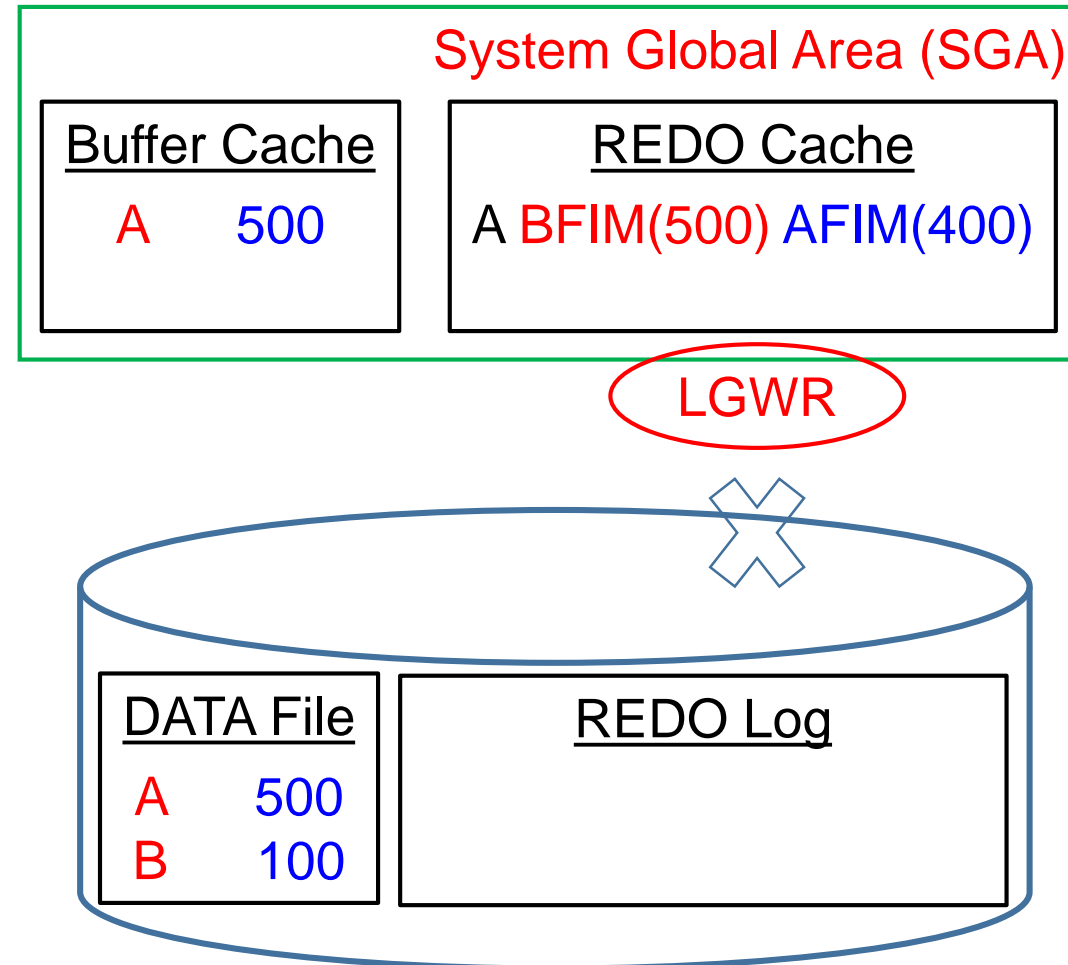
- Ensure the before-image (BFIM) is recorded
- Appropriate log entry flushed to disk
- Necessary for UNDO operation if needed

UNDO-type log entries

REDO-type log entries

Deferred Update Technique with NO UNDO

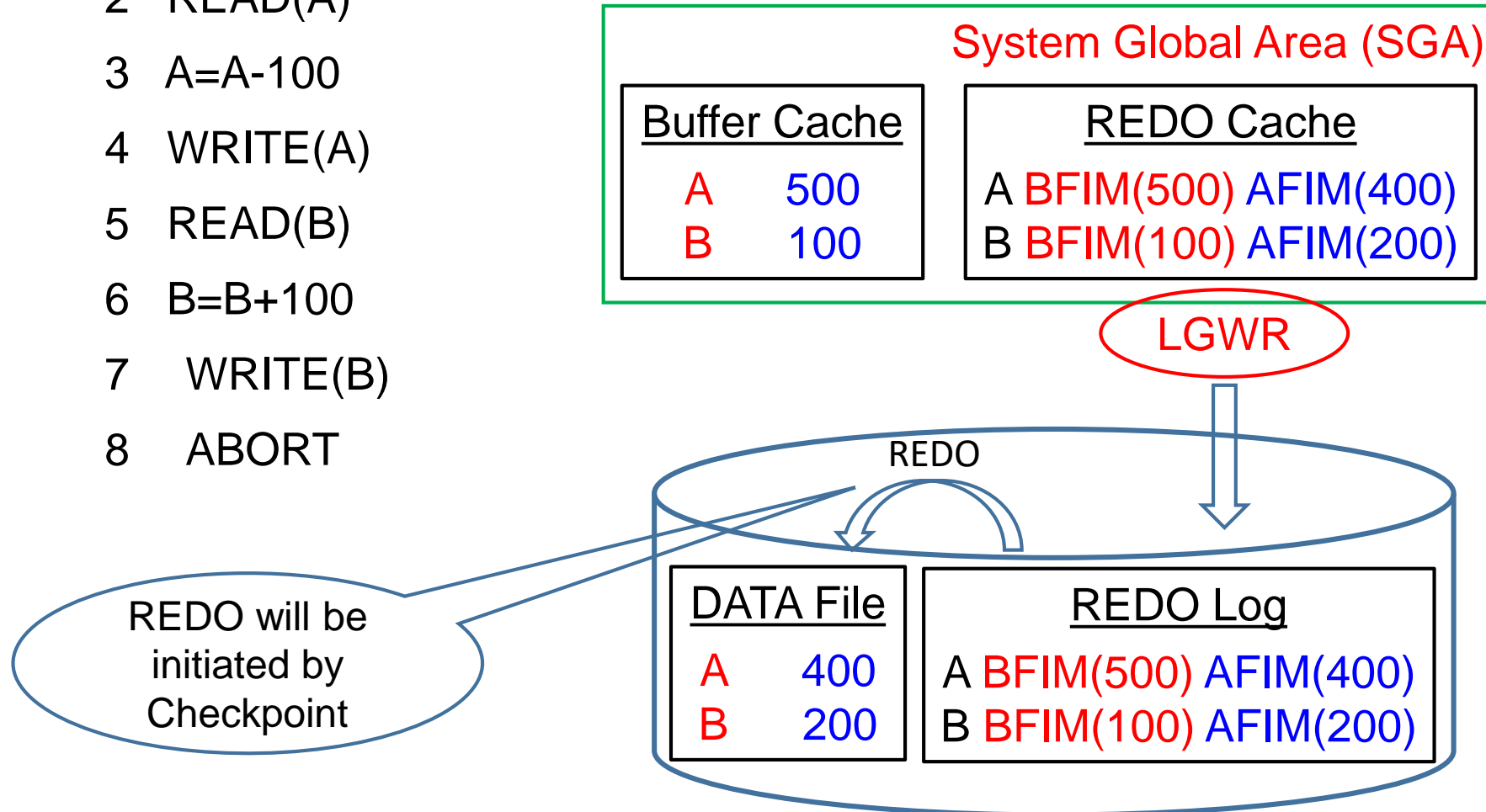
- 2 READ(A)
- 3 A=A-100
- 4 WRITE(A)
- 5 ABORT



Deferred Update Technique with REDO

```

2  READ(A)
3  A=A-100
4  WRITE(A)
5  READ(B)
6  B=B+100
7  WRITE(B)
8  ABORT
  
```



Steal/no-steal and force/no-force

Specify rules that govern when a page from the database cache can be written to disk

No-steal approach

- Cache buffer page updated by a transaction cannot be written to disk before the transaction commits

Steal approach

- Recovery protocol allows writing an updated buffer before the transaction commits

Steal/no-steal and force/no-force

Force approach

- All pages updated by a transaction are immediately written to disk before the transaction commits

Otherwise, **no-force approach**

Typical **database systems employ a steal/no-force strategy**

- Avoids need for very large buffer space
- Reduces disk I/O operations for heavily updated pages

Checkpoint

Taking a checkpoint

- Suspend execution of all transactions temporarily
- Force-write all main memory buffers that have been modified to disk
- Write a checkpoint record to the log, and forcewrite the log to the disk
- Resume executing transactions

DBMS recovery manager decides on checkpoint interval

Deferred update concept

- Postpone updates to the database on disk until the transaction completes successfully and reaches its commit point
- Redo-type log entries are needed
- Undo-type log entries not necessary
- Can only be used for short transactions and transactions that change few items
- Buffer space an issue with longer transactions

Deferred update concept

Deferred update protocol

- Transaction cannot change the database on disk until it reaches its commit point
- All buffers changed by the transaction must be pinned until the transaction commits (no-steal policy)
- Transaction does not reach its commit point until all its REDO-type log entries are recorded in log and log buffer is force-written to disk

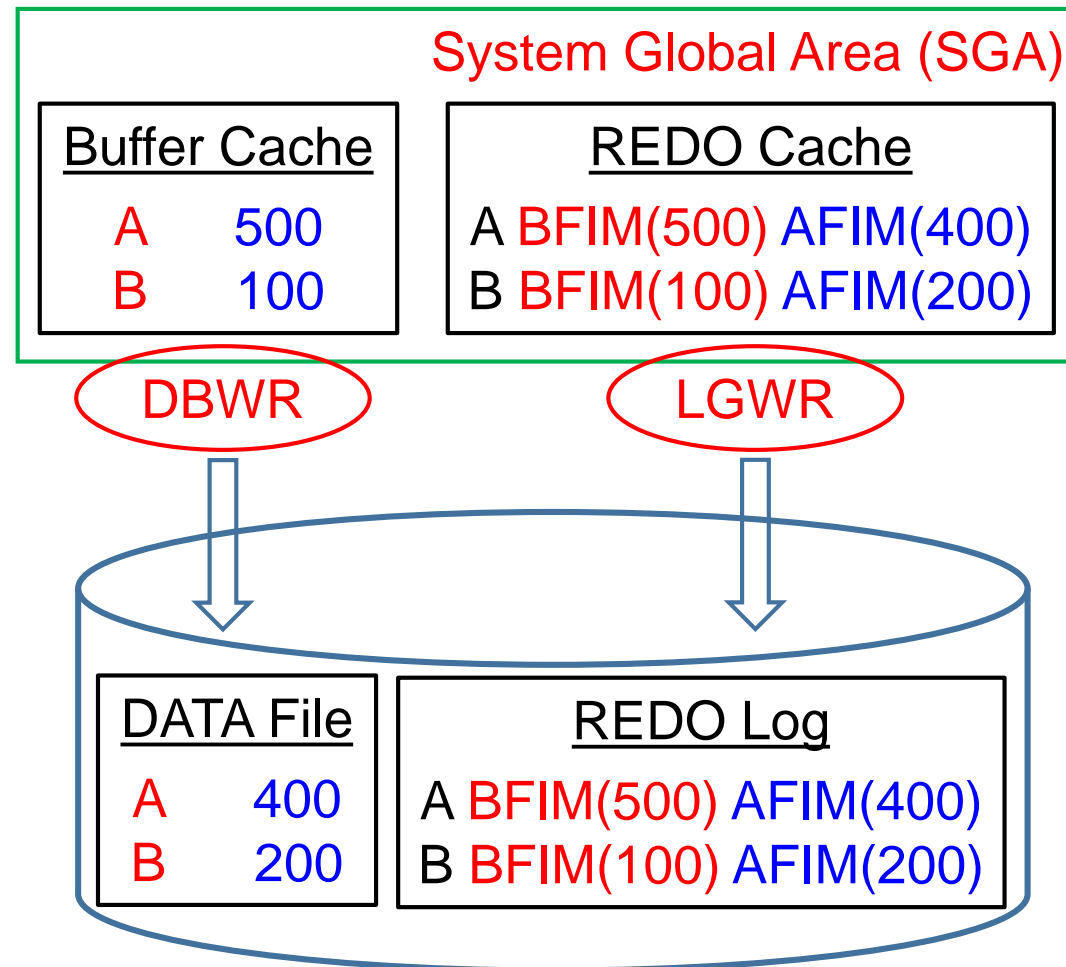
Immediate Update Technique

- Database may be updated by some operations of a transaction before it reaches commit point
- Operations also recorded in log
- Recovery still possible

Immediate Update Technique

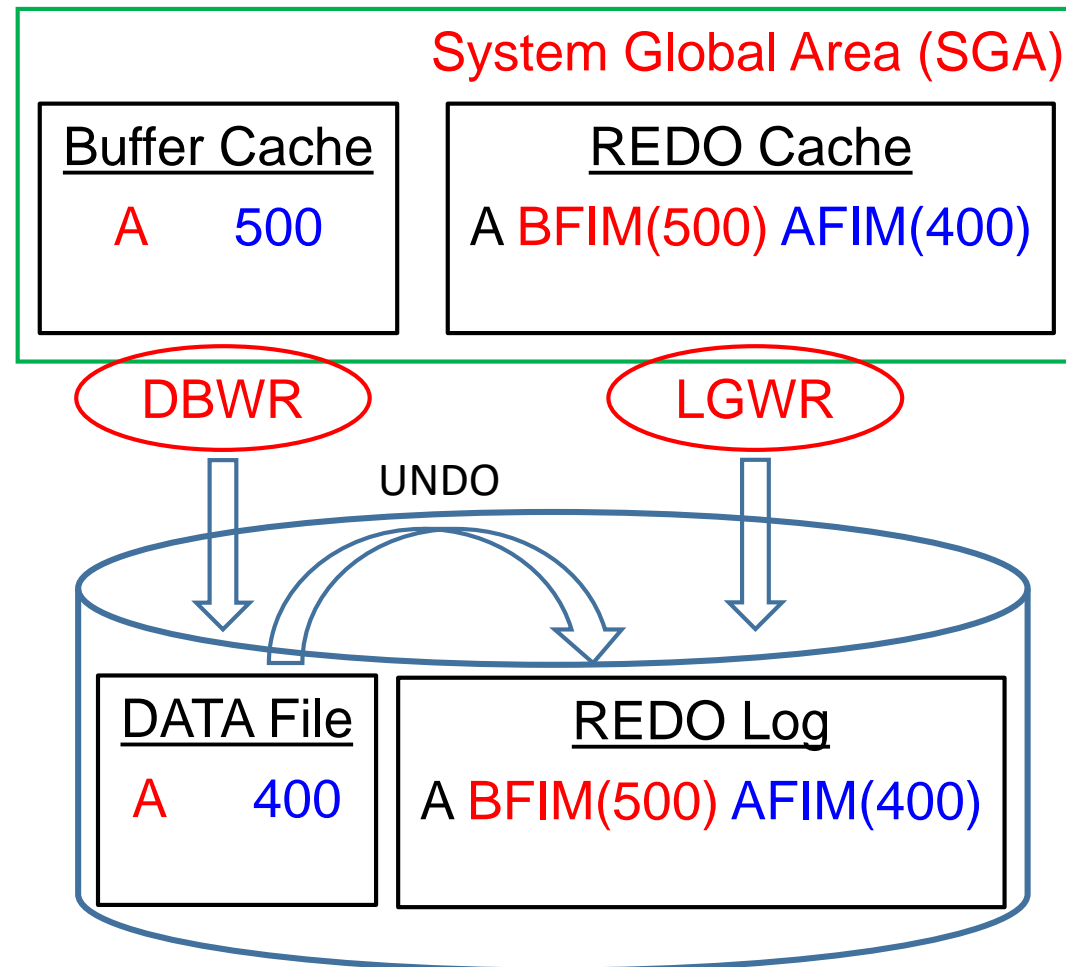
```

2  READ(A)
3  A=A-100
4  WRITE(A)
5  READ(B)
6  B=B+100
7  WRITE(B)
8  COMMIT
  
```



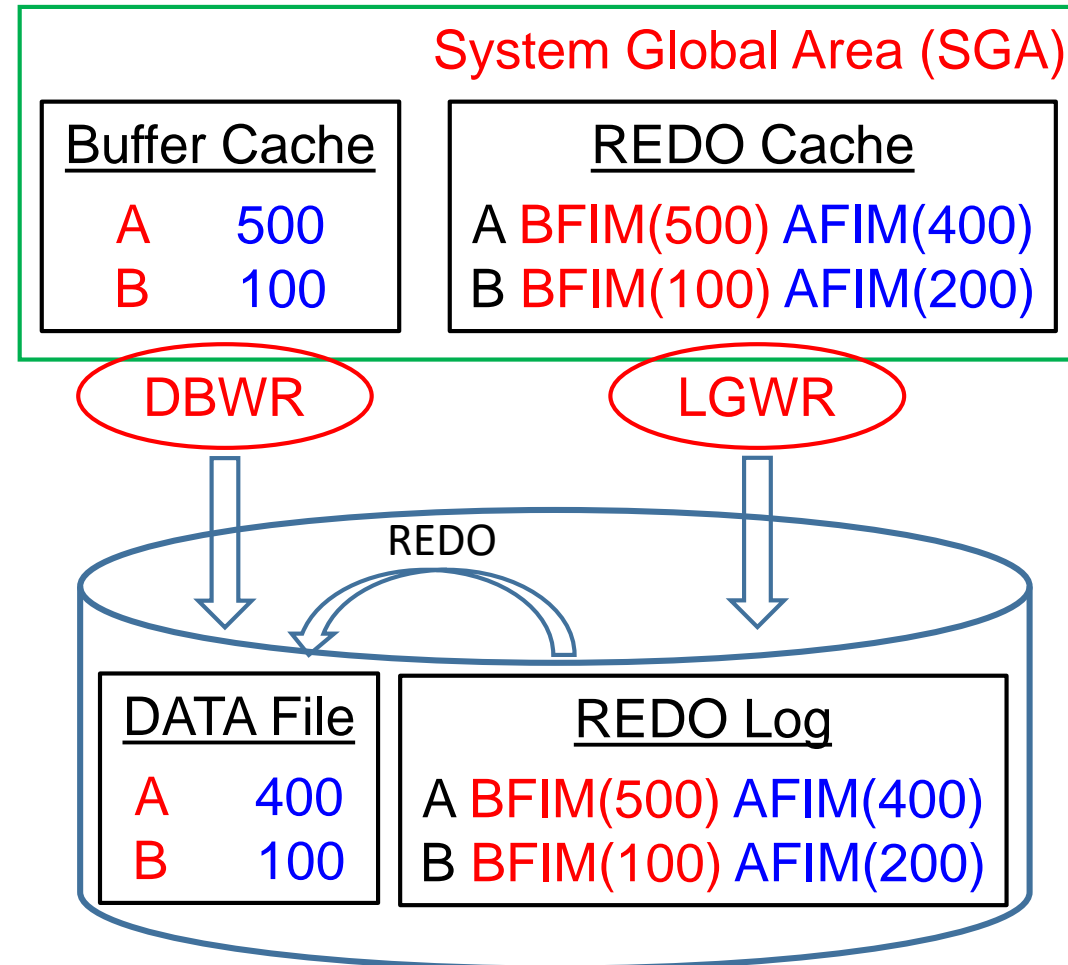
Immediate Update Technique with UNDO

- 2 READ(A)
- 3 A=A-100
- 4 WRITE(A)
- 5 ABORT



Immediate Update Technique with REDO

2 READ(A)
 3 A=A-100
 4 WRITE(A)
 5 READ(B)
 6 B=B+100
 7 WRITE(B)
 8 ABORT



Immediate Update Technique

Database can be updated immediately

- No need to wait for transaction to reach commit point
- Not a requirement that every update be immediate

UNDO-type log entries must be stored

Recovery algorithms

- UNDO/NO-REDO (steal/force strategy)
- UNDO/REDO (steal/no-force strategy)

Summary

This session will give the knowledge about

- Recovery Concepts
- Deferred update techniques
- Immediate update techniques