## Recovery techniques based on immediate update

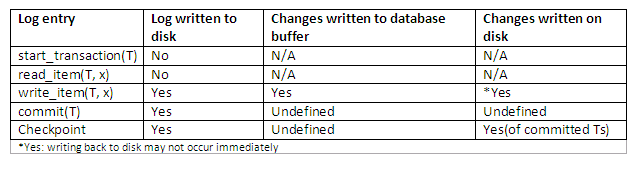
### **Immediate update**

In the immediate update techniques, the database may be updated by the operations of a transaction immediately, before the transaction reaches its commit point. However, these operations are typically recorded in the log on disk by force-writing before they are applied to the database, so that recovery is possible.

When immediate update is allowed, provisions must be made for undoing the effect of update operations on the database, because a transaction can fail after it has applied some updates to the database itself. Hence recovery schemes based on immediate update must include the capability to roll back a transaction by undoing the effect of its write operations.

1. When a transaction starts, write an entry start\_transaction(T) to the log;
2. When any operation is performed that will change values in the database, write a log entry write\_item(T, x, old\_value, new\_value);
3. Write the log to disk;
4. Once the log record is written, write the update to the database buffers;
5. When convenient, write the database buffers to the disk;
6. When a transaction is about to commit, write a log record of the form commit(T);
7. Write the log to disk.

The protocol and how different entries are affected can be best summarised below:



## Recovery techniques based on deferred update

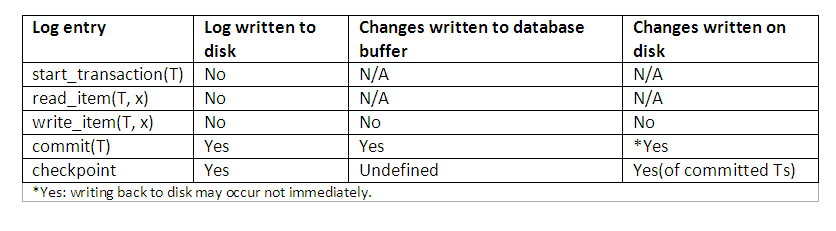
### **Deferred update**

The idea behind deferred update is to defer or postpone any actual updates to the database itself until the transaction completes its execution successfully and reaches its commit point. During transaction execution, the updates are recorded only in the log and in the transaction workspace. After the transaction reaches its commit point and the log is force-written to disk, the updates are recorded in the database itself. If a transaction fails before reaching its commit point, there is no need to undo any operations, because the transaction has not affected the database in any way.

The steps involved in the deferred update protocol are as follows:

1. When a transaction starts, write an entry start\_transaction(T) to the log.
2. When any operation is performed that will change values in the database, write a log entry write\_item(T, x, old\_value, new\_value).
3. When a transaction is about to commit, write a log record of the form commit(T); write all log records to disk.
4. Commit the transaction, using the log to write the updates to the database; the writing of data to disk need not occur immediately.
5. If the transaction aborts, ignore the log records and do not write the changes to disk.

The database is never updated until after the transaction commits, and there is never a need to UNDO any operations. Hence this technique is known as the NO-UNDO/REDO algorithm. The REDO is needed in case the system fails after the transaction commits but before all its changes are recorded in the database. In this case, the transaction operations are redone from the log entries. The protocol and how different entries are affected can be best summarised as shown:



## Shadow Paging

**Shadow Paging**is recovery technique that is used to recover [database](https://www.geeksforgeeks.org/what-is-database/). In this recovery technique, database is considered as made up of fixed size of logical units of storage which are referred as **pages.** pages are mapped into physical blocks of storage, with help of the **page table**which allow one entry for each logical page of database. This method uses two page tables named **current page table** and **shadow page table**.

The entries which are present in current page table are used to point to most recent database pages on disk. Another table i.e., Shadow page table is used when the transaction starts which is copying current page table. After this, shadow page table gets saved on disk and current page table is going to be used for transaction. Entries present in current page table may be changed during execution but in shadow page table it never get changed. After transaction, both tables become identical.

This technique is also known as **Cut-of-Place updating.**