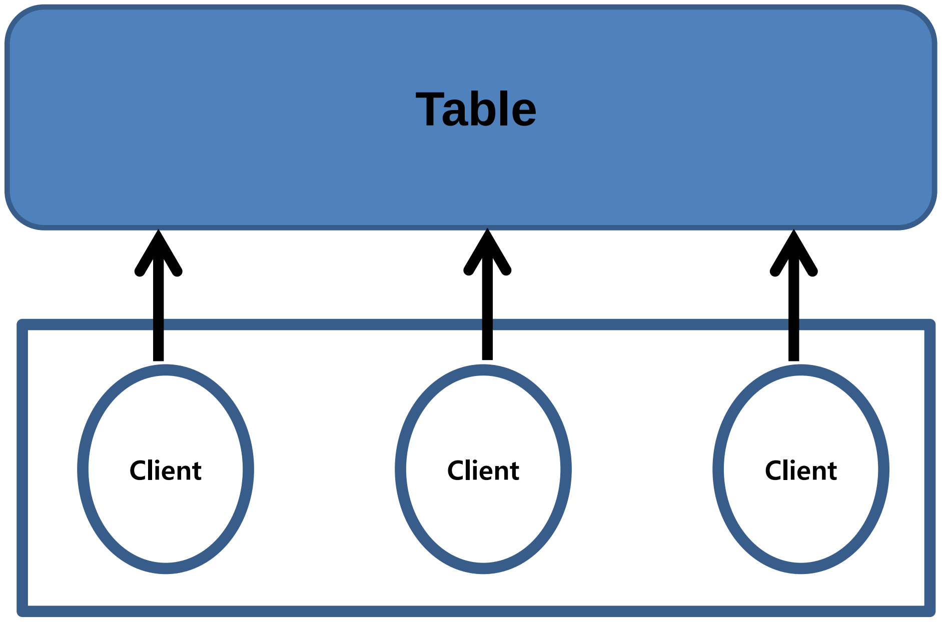
CONCURRENCY CONTROL

CONCURRENCY CONTROL

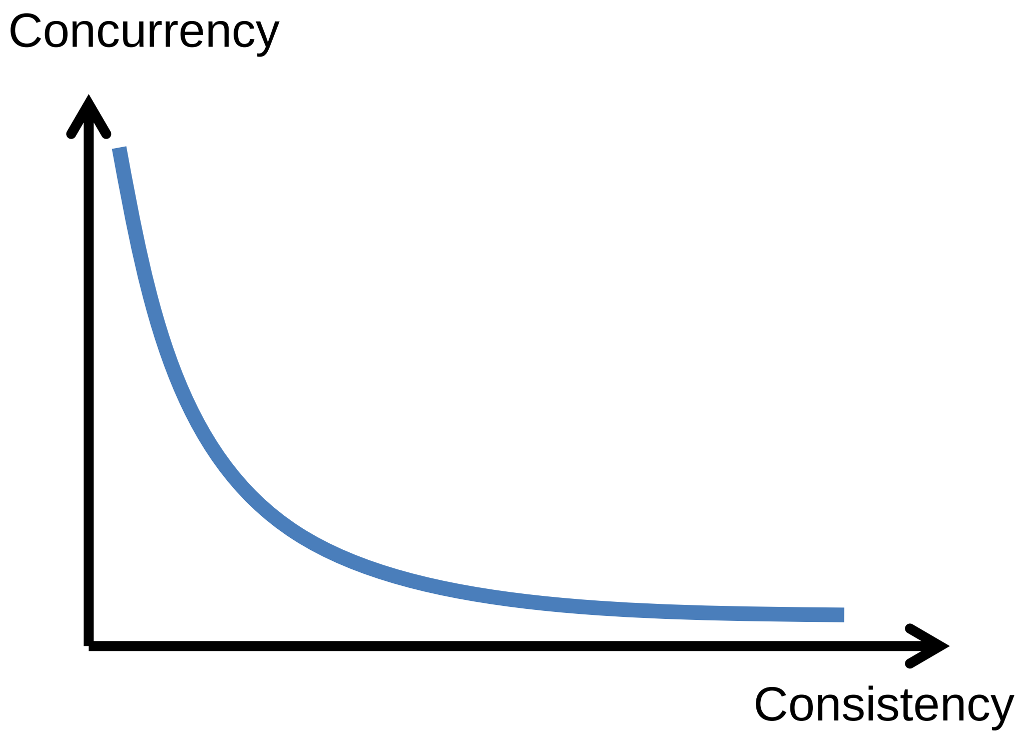
Concurrency Control

- Method that prevent the inconsistency, loss and the changes while processing multiple transactions in parallel
- Lock is provided for concurrency control to let multiple transactions change different data from a same table while they are not affecting each other.



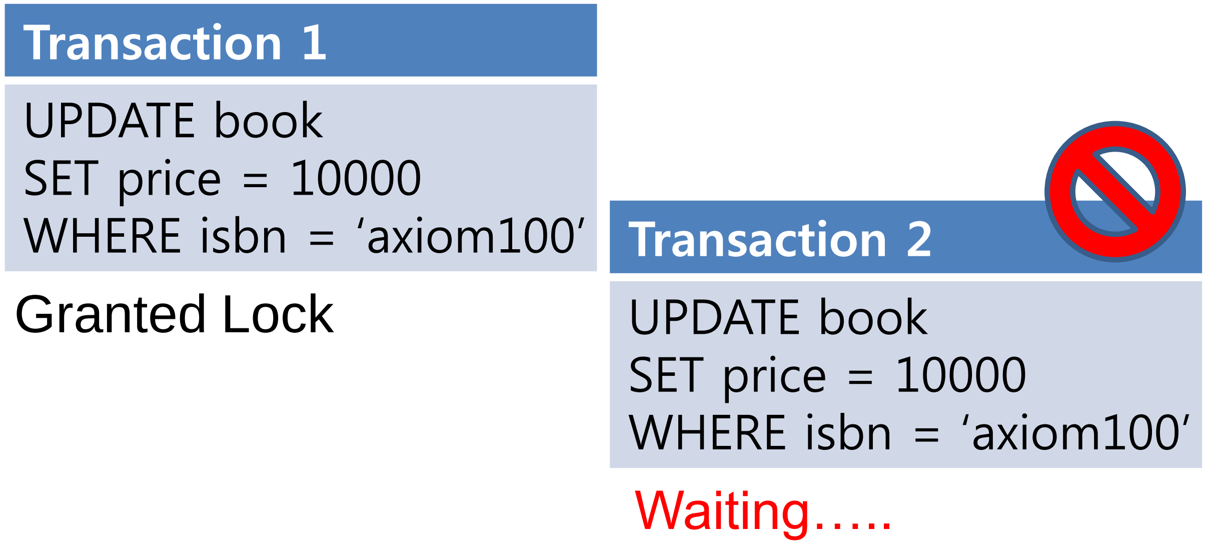
CONCURRENCY CONTROL

- **Concurrency**
  - Multiple users or sessions accessing same data at same time
- **Consistency**
  - Restricting multiple users or sessions to access the same data at same time



LOCK

❖ **Transaction Lock**



Transaction 1

```
UPDATE book
SET price = 10000
WHERE isbn = 'axiom100'
```

Granted Lock

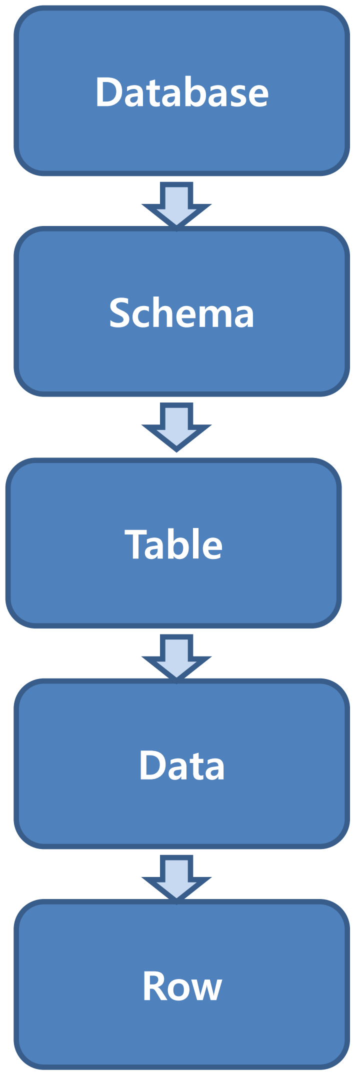
Transaction 2

```
UPDATE book
SET price = 10000
WHERE isbn = 'axiom100'
```

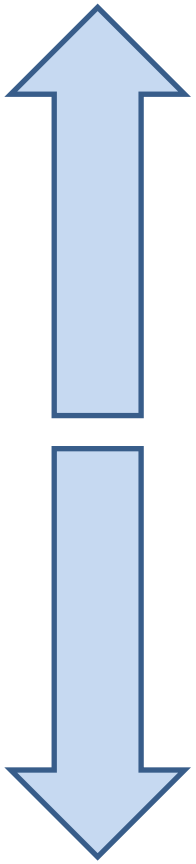
Waiting…..

- Prevents accessing same data in a multi-user environment
- Automatically grants the lock for DML(INSERT, DELETE and UPDATE)
- Automatically grants the lock for DDL(ALTER,CREATE,DROP and etc..)
- To provide maximum consistency and concurrency
- Two different locks: Row level lock & Table level lock

LOCK



Bigger level lock (Low Concurrency Control)



Smaller level lock (High Concurrency Control)

LOCK

- **Lock Mode: Table level & Row level**
- **Row level lock Mode**

Lock Mode	Description	Functions
S	Shared Lock	User with lock can select data
X	Exclusive Lock	User with lock can select and update and other transaction cannot do any of statements

LOCK

Table level lock Mode

Lock Mode	Description	Functions
S	Shared Lock	User with lock can select data but other transaction can only select the data as well
X	Exclusive Lock	User with lock can select and update but other transaction cannot do any of statements
IS	Intent Shared Lock	User with this lock can select only, once Shared Lock(S) is obtained first
IX	Intent Exclusive Lock	User with lock can select and update once Exclusive Lock(X) is obtained first. Other transaction can also select and update same data at same time
SIX	Shared with Intent Exclusive Lock	User with lock can select and once Exclusive Lock is obtained then it can update as well. Other transaction can only select not update

LOCK CONFLICTS

- **Not Committed Transaction**
  - COMMIT or ROLLBACK statement is not executed

- **Long Time Transaction**
  - Execution time is long due to processing enormous large amount of data

- **Unnecessary High Level Object Lock**
  - When high level lock such as table lock is given, then the transaction that tries to access data of table that has high level lock has to wait. For example, when Transaction with DDL such as ALTER TABLE is executed then all the transactions have to wait because of DDL statements.

CONCURRENCY CONTROL

- **SVCC vs. MVCC**
  - Methods for controlling simultaneous read / modify operations on the same records

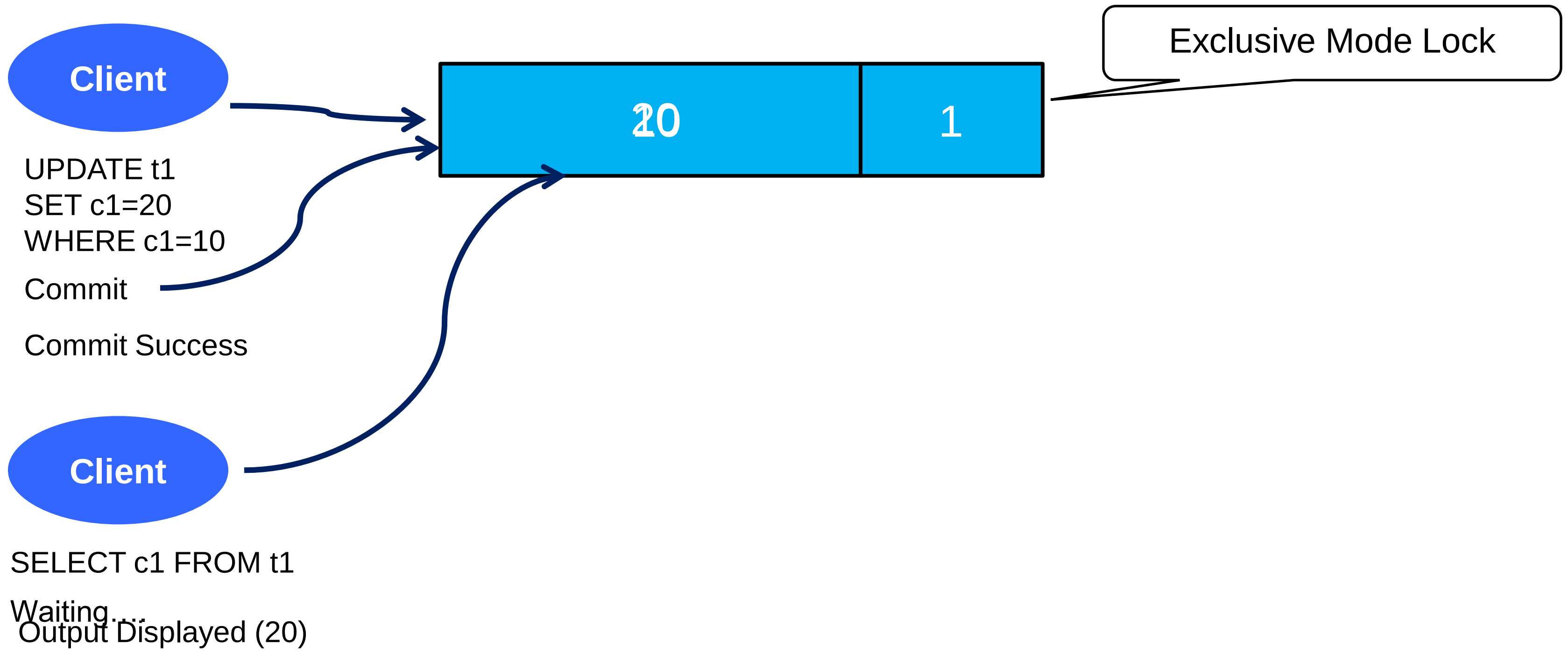
- **SVCC (Single-Version Concurrency Control)**
  - With this method, only one version (or “image”) of a record exists.
  - Even search operations can affect other transactions.

- **MVCC (Multi-Version Concurrency Control)**
  - With this technique, when a record is changed, the original version of the record is maintained and unchanged. A new changed version of the record is created, so that even if one transaction is performing an operation on a record, it will have no effect on another transaction that reads the same record.

SVCC

SVCC

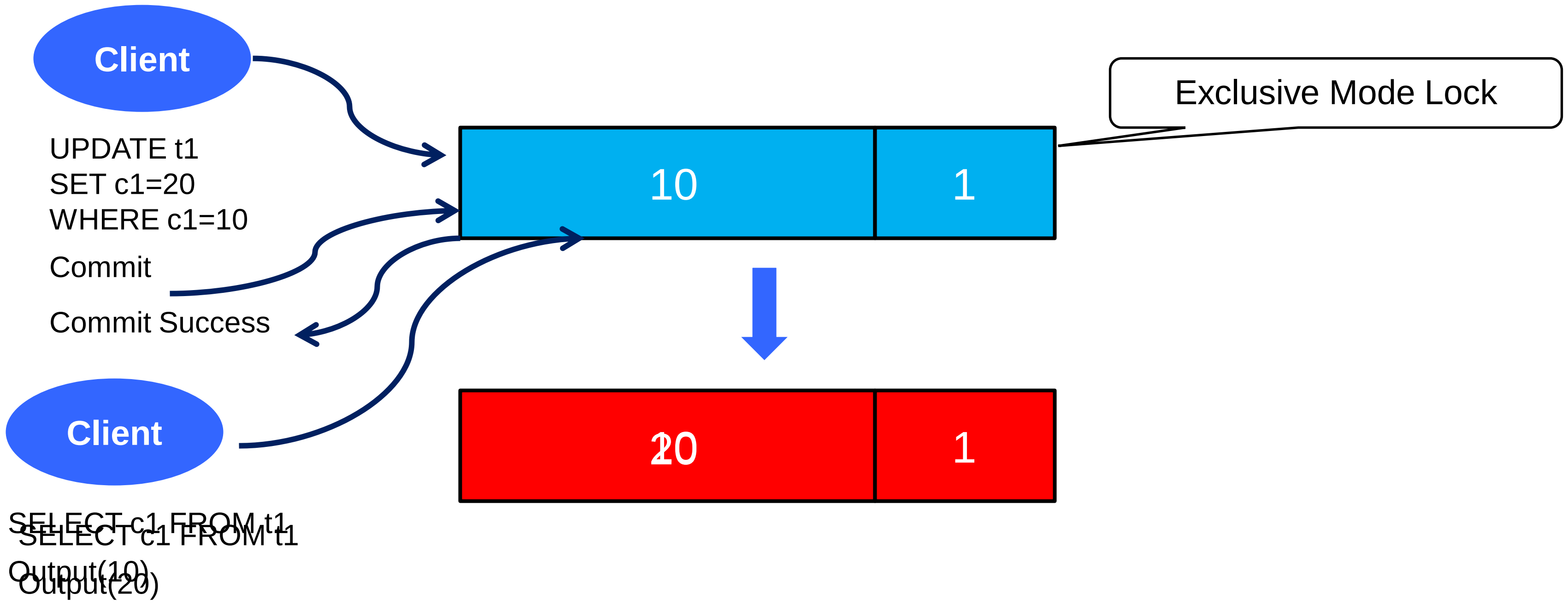
- Because a record is locked when it is modified, conflicts can occur between read and modifying operations, thus there is a high possibility of reduced performance in environments characterized by frequent transactions.



MVCC

MVCC

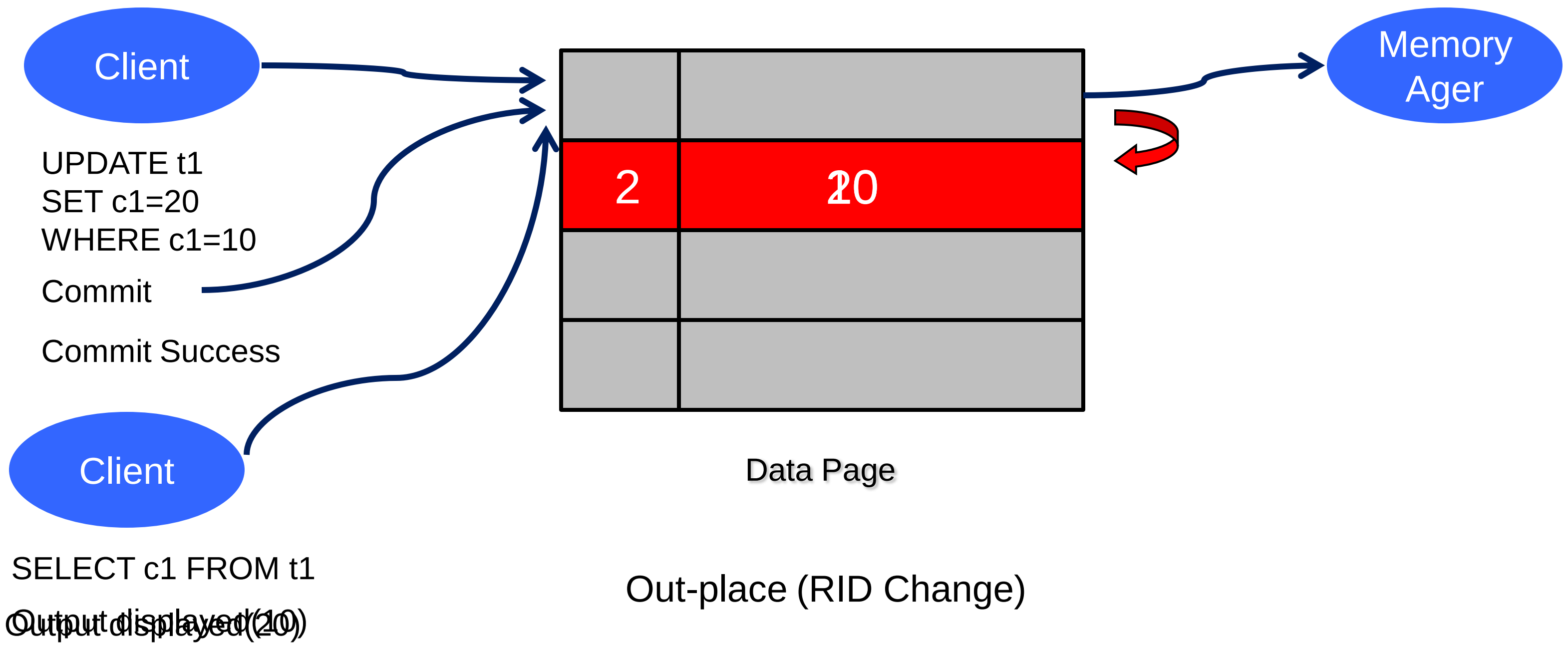
- When a record is modified the lock is granted and a new version of the record is created, so conflicts do not occur between read and modifying operations.
- Good performance is guaranteed, even in environments characterized by frequent transactions.



OUT-PLACE & IN-PLACE

Memory Table: Out-Place

- New version of a record is saved in the data page with a different Row ID thus the high performance is guaranteed but there is a possibility that the page usage rate might decrease so when a version of particular record is bigger than the configured size in property, it is replaced to In-place Update.

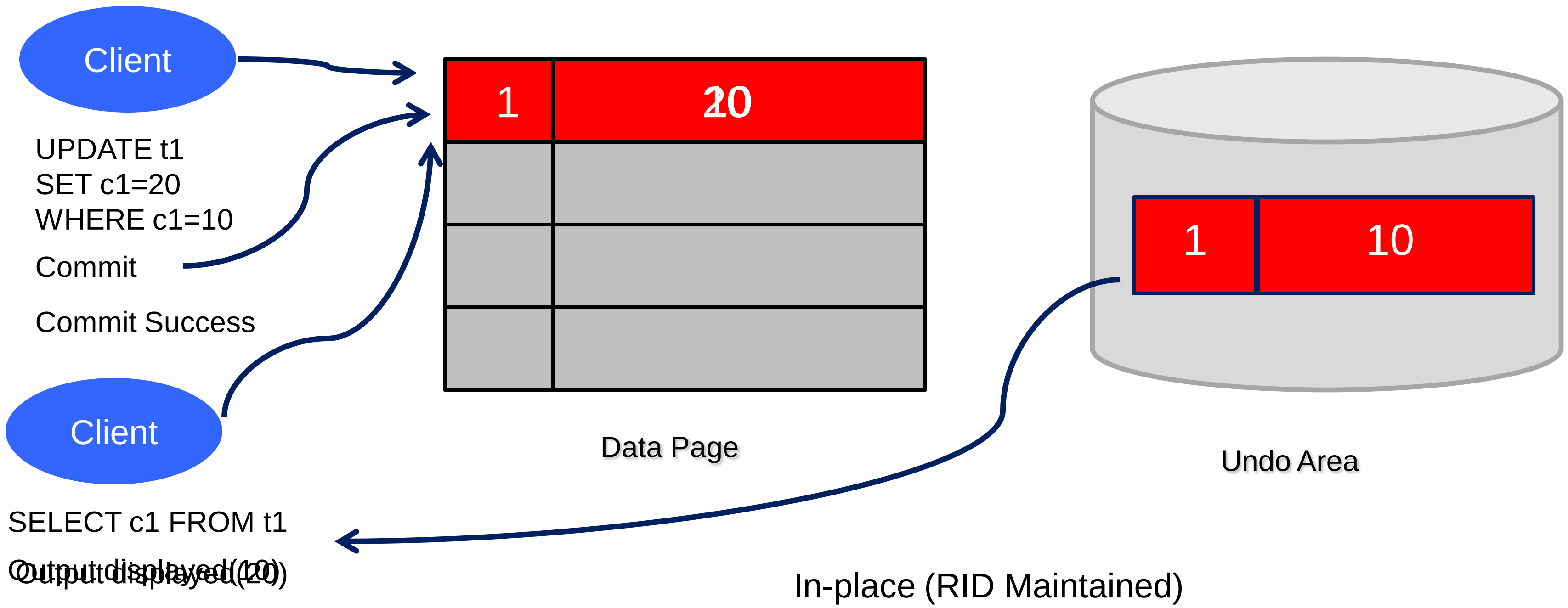


Out-place (RID Change)

OUT-PLACE & IN-PLACE

Disk Table: In-Place

- The modified columns that belong to the original record are written as undo log records to the Undo tablespace, and the modified values are written to the location of the original record.



In-place (RID Maintained)

Thank you!

Altibase Education Center

Tel : 02-2082-1451
Fax : 02-2082-1459
E-mail : education@altibase.com
Homepage : http://edu.altibase.com