



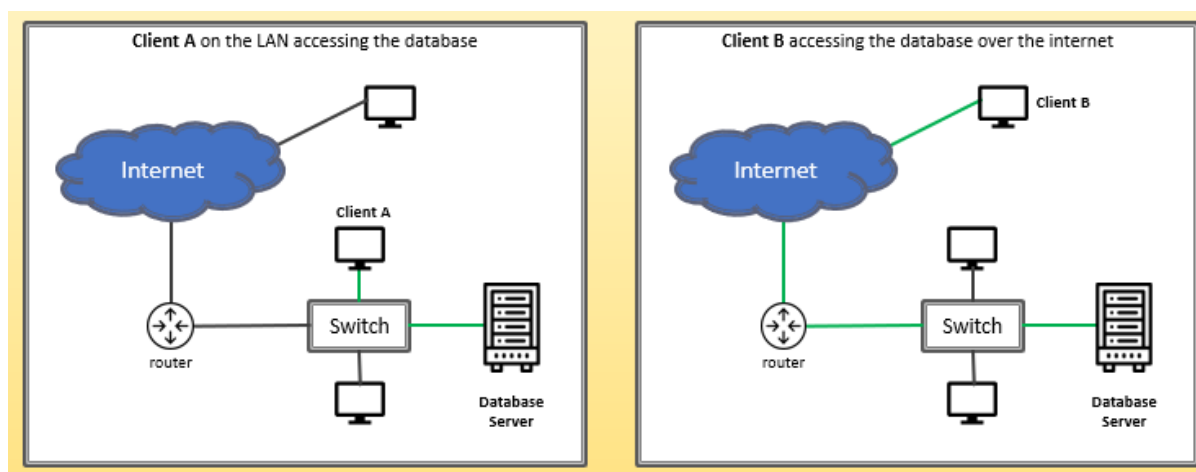
4.10.5 Client Server Database

Client-server Database

A **client-server database** provides simultaneous access to a database stored on a server for multiple clients.

A client could be a device on the LAN network with the database server or a device accessing the database server over the internet.

A Database Management System (DBMS) on the server processes database requests from the clients.



Advantages:

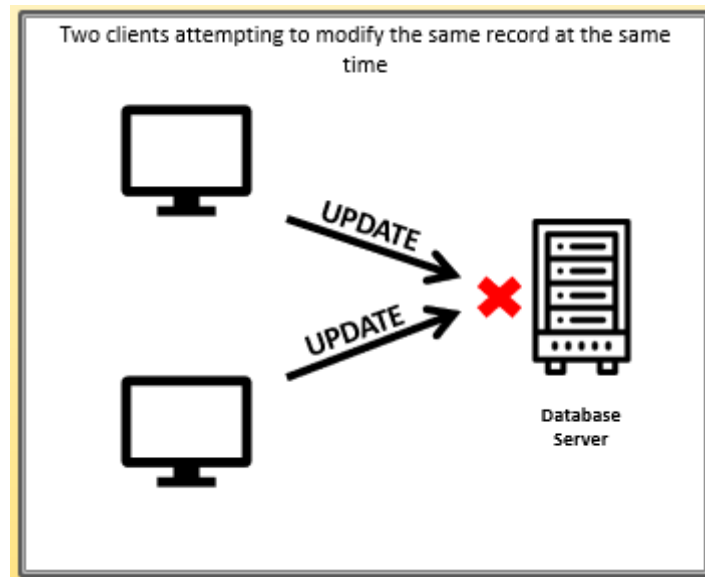
- Database is consistent as there is only one copy on the database server.
- Many clients can access a database stored on a powerful central server.





Concurrent Access

Multiple clients may try to access or modify the same data at the same time. The database must control these transactions to prevent conflicts and maintain data integrity.



Potential solutions for controlling concurrent access are:

- Record locking
- Serialisation
 - Timestamp ordering
 - Commitment ordering





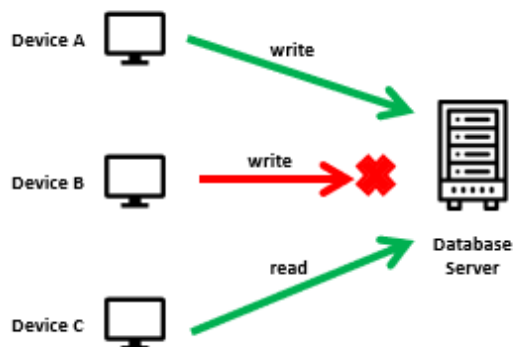
Record Locking

- A concurrency control method where a record can only be modified by one client at a time.
- When a transaction on a record starts, a lock is set on the record.
- Other transactions cannot **edit** the record until the lock is released (they may still be allowed read access).

Device A has accessed the database with the intent to update a record.

Device B is blocked for accessing the same record with the intent to also edit the record.

Device C is allowed to access the record as its intent is only to read the record.



Serialisation

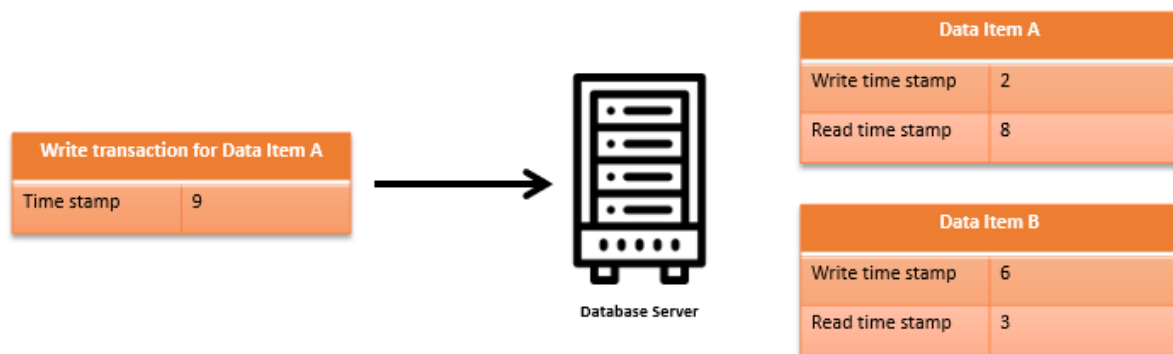
- A method of controlling transactions by processing them in a specific order rather than allowing conflicting transactions to occur simultaneously.
- Database requests are placed into a queue and processed in order.
- If transactions conflict, one transaction may be delayed or aborted so that the transactions do not interfere with each other.





Timestamp Ordering

- A serialization method where **timestamps** are generated for each transaction.
- The database records timestamps for the last read and the last write operations for each data item.
- These timestamps are used to determine whether a transaction can proceed without causing a conflict.
- A transaction may be rejected/aborted if it would violate the required ordering.



Commitment Ordering

- Transactions are created before being sent to the server for execution.
- The database assesses the transactions to identify potential conflicts.
- The database applies an algorithm to create a commit order which avoids conflicts between transactions.

