SQL: ACID and Isolation levels

ACID

- In computer science, **ACID** (*Atomicity, Consistency, Isolation, Durability*) is a set of properties of database transactions intended to guarantee validity even in the event of errors, power failures, etc.
- In the context of databases, a sequence of database operations that satisfies the ACID properties, and thus can be perceived as a single logical operation on the data, is called a transaction.
- For example, a transfer of funds from one bank account to another, even involving multiple changes such as debiting one account and crediting another, is a single transaction.

ACID Properties of Transactions

Atomicity	• All or nothing
Consistency	• Data integrity when complete
Isolation	• Modification isolated from other transactions
Durability	• Changes are permanent

Atomicity

- **Definition**:
  means that each step in the transaction must complete successfully, or they're all rolled back.
- **Example**:
  Imagine transferring money from your checking to savings account, if the system crashes in between the transfer, you want both accounts rolled back to the state they started in, so you don't lose money.

Consistency

- **Definition**:
  in database systems refers to the requirement that any given database transaction must change affected data only in allowed ways.
- Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof.

Isolation

- **Definition**:
  means that data in a transaction that's in process shouldn't be visible to another transaction until it completes.
- **Goal**:
  To improve performance. Providing isolation is the main goal of concurrency control.

Durability

- **Definition**:
  means that the result of the committed transaction is a permanent modification to the system, meaning that even if the system were to crash and reboot, the data will remain.
- **Example**:
  if a flight booking reports that a seat has successfully been booked, then the seat will remain booked even if the system crashes.

Questions

- What is the Atomicity?
- What is the Consistency?
- What is the Isolation?
- What is the Durability?

Read phenomena

Dirty Read

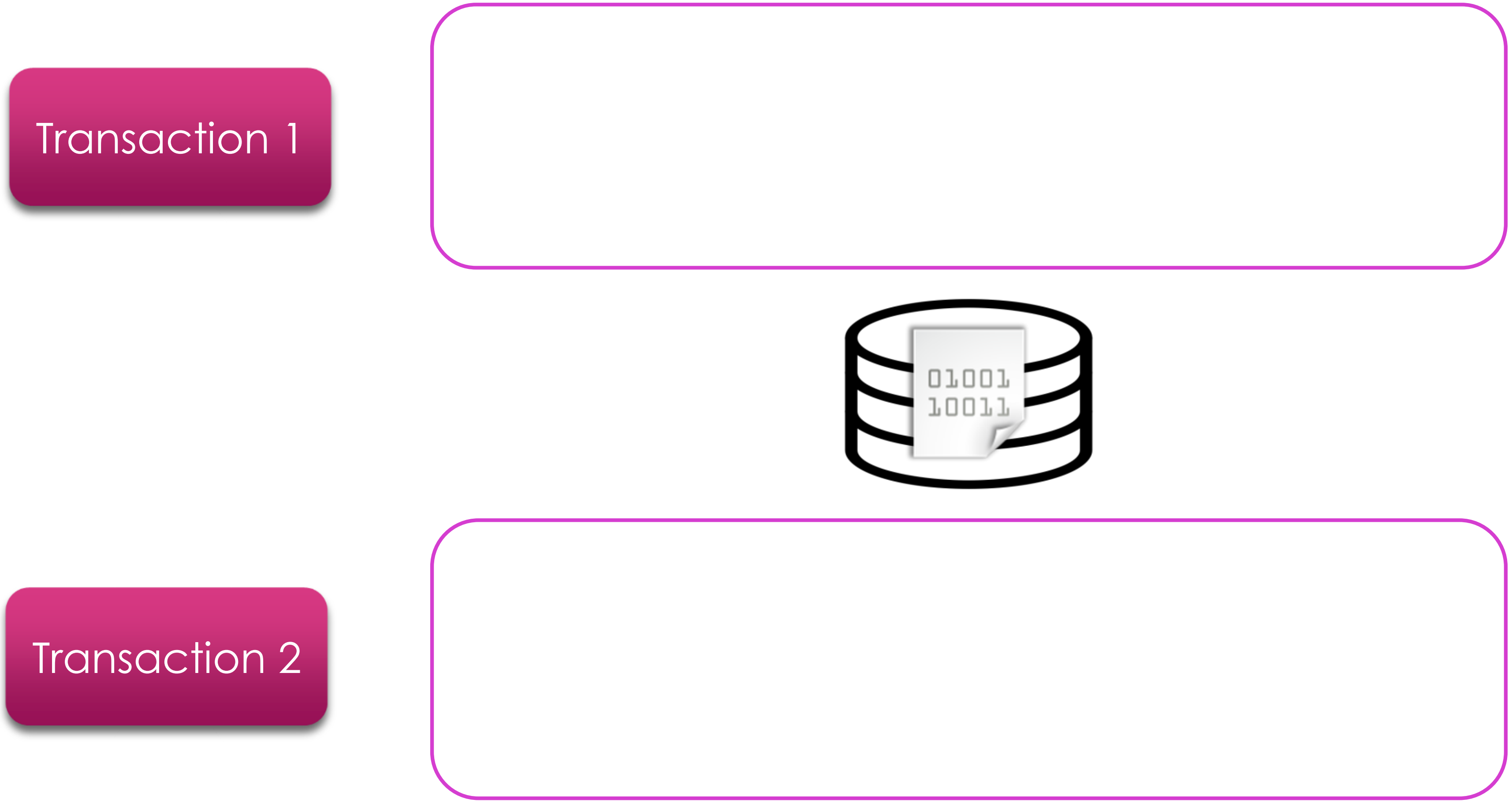
- A dirty read (aka uncommitted dependency) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.

Dirty Read

Transaction 1

Transaction 2

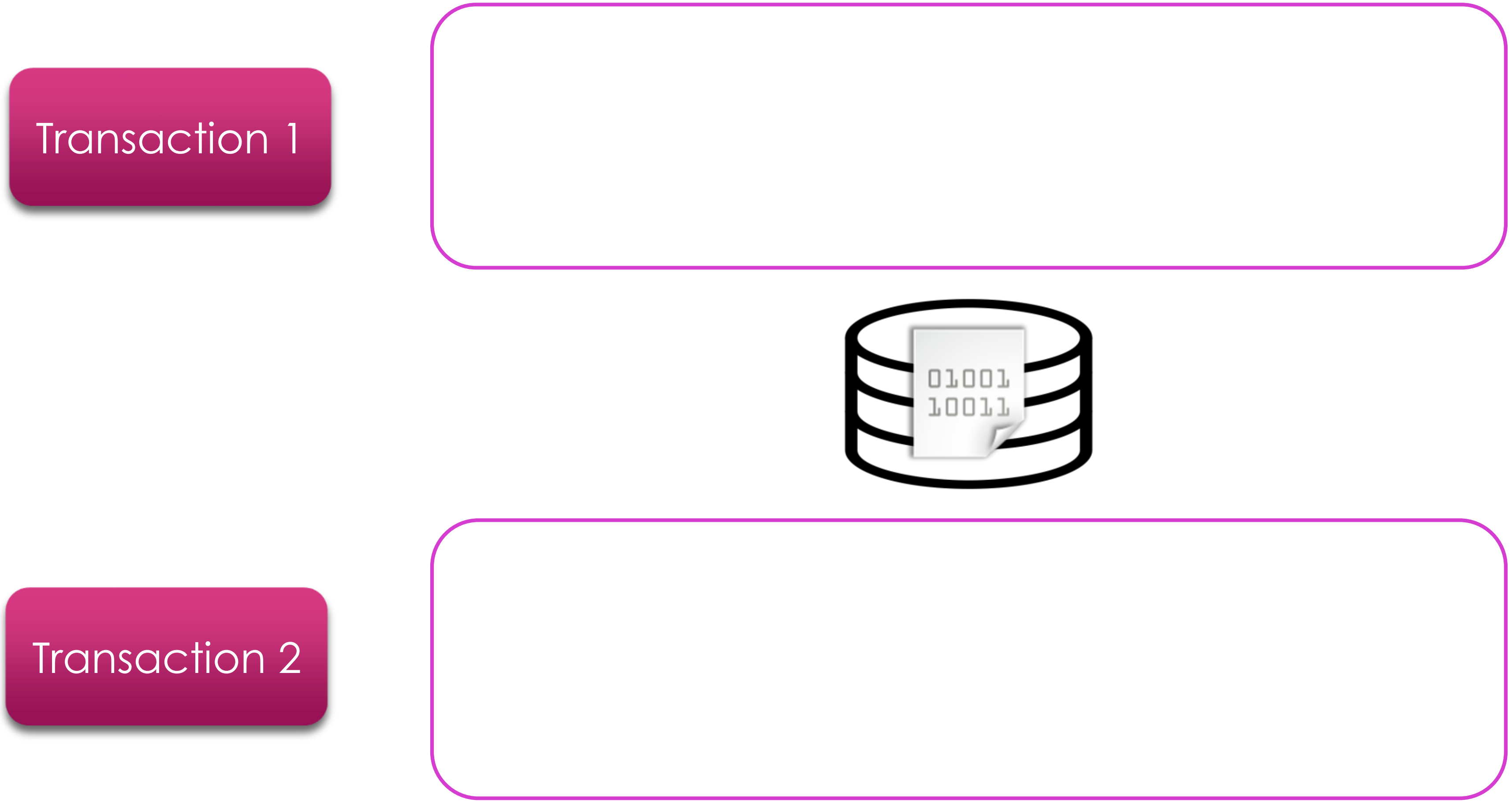
Dirty Read



Transaction 1

Transaction 2

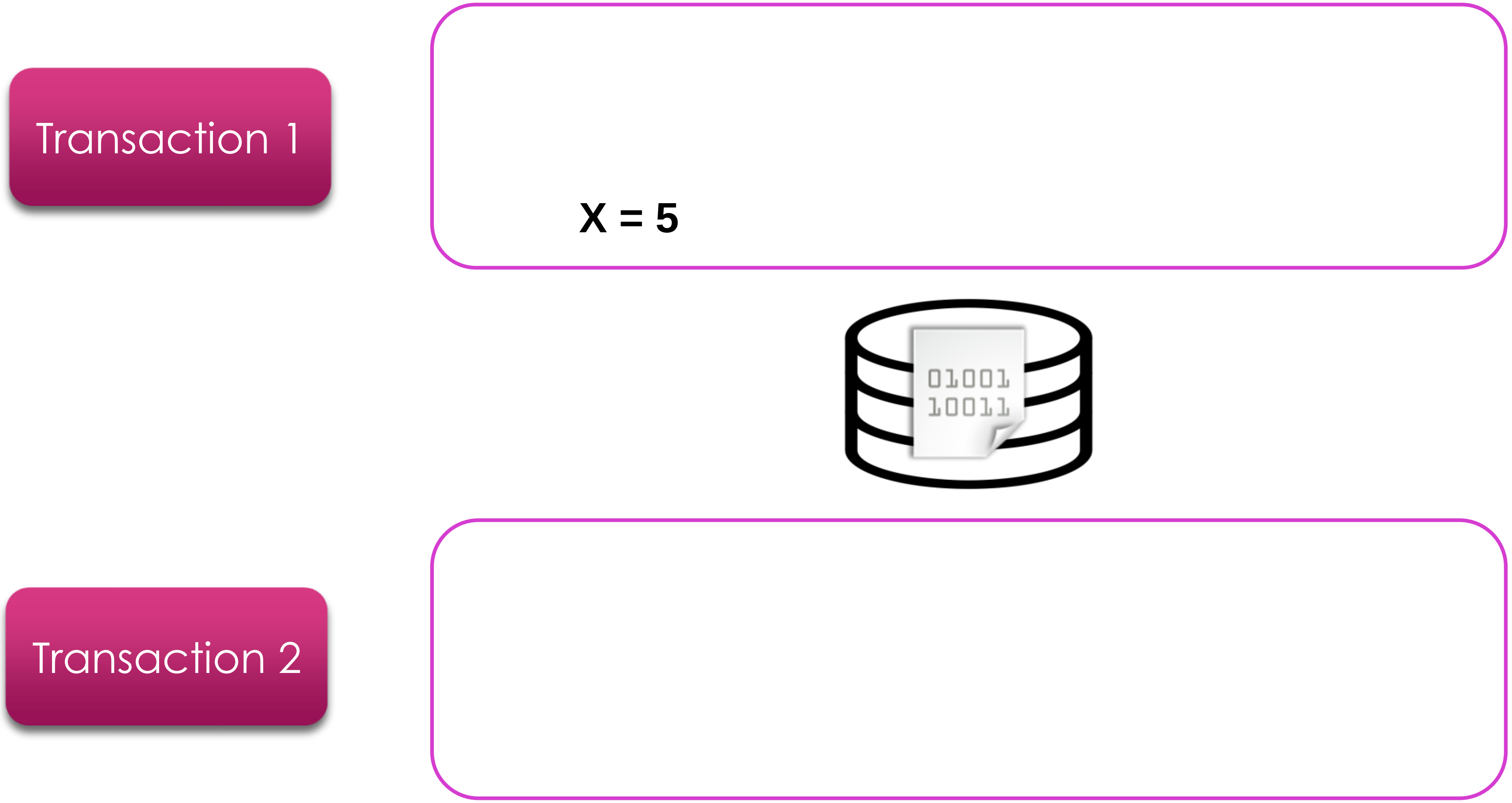
Dirty Read



Transaction 1

Transaction 2

Dirty Read

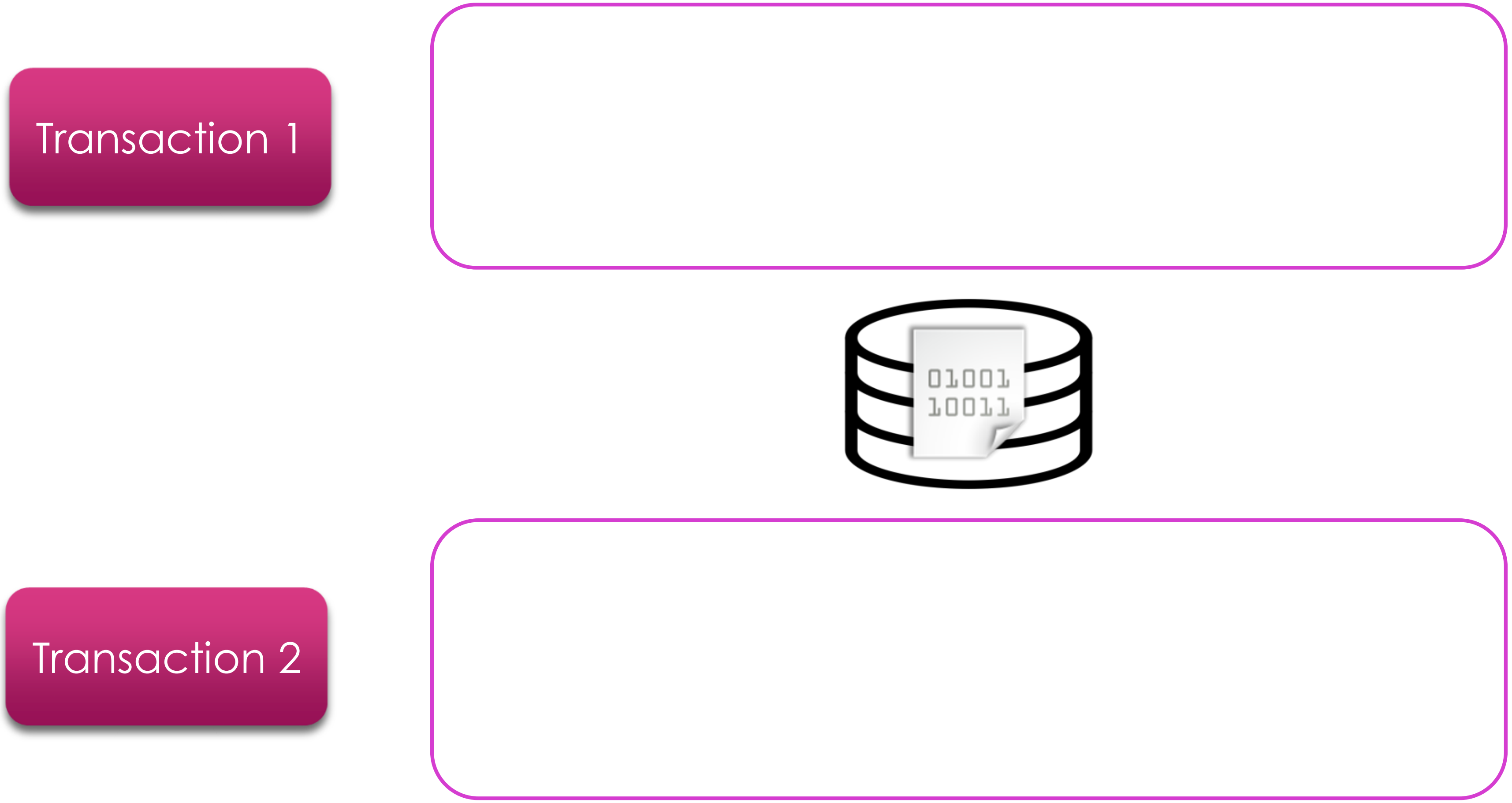


Transaction 1

X = 5

Transaction 2

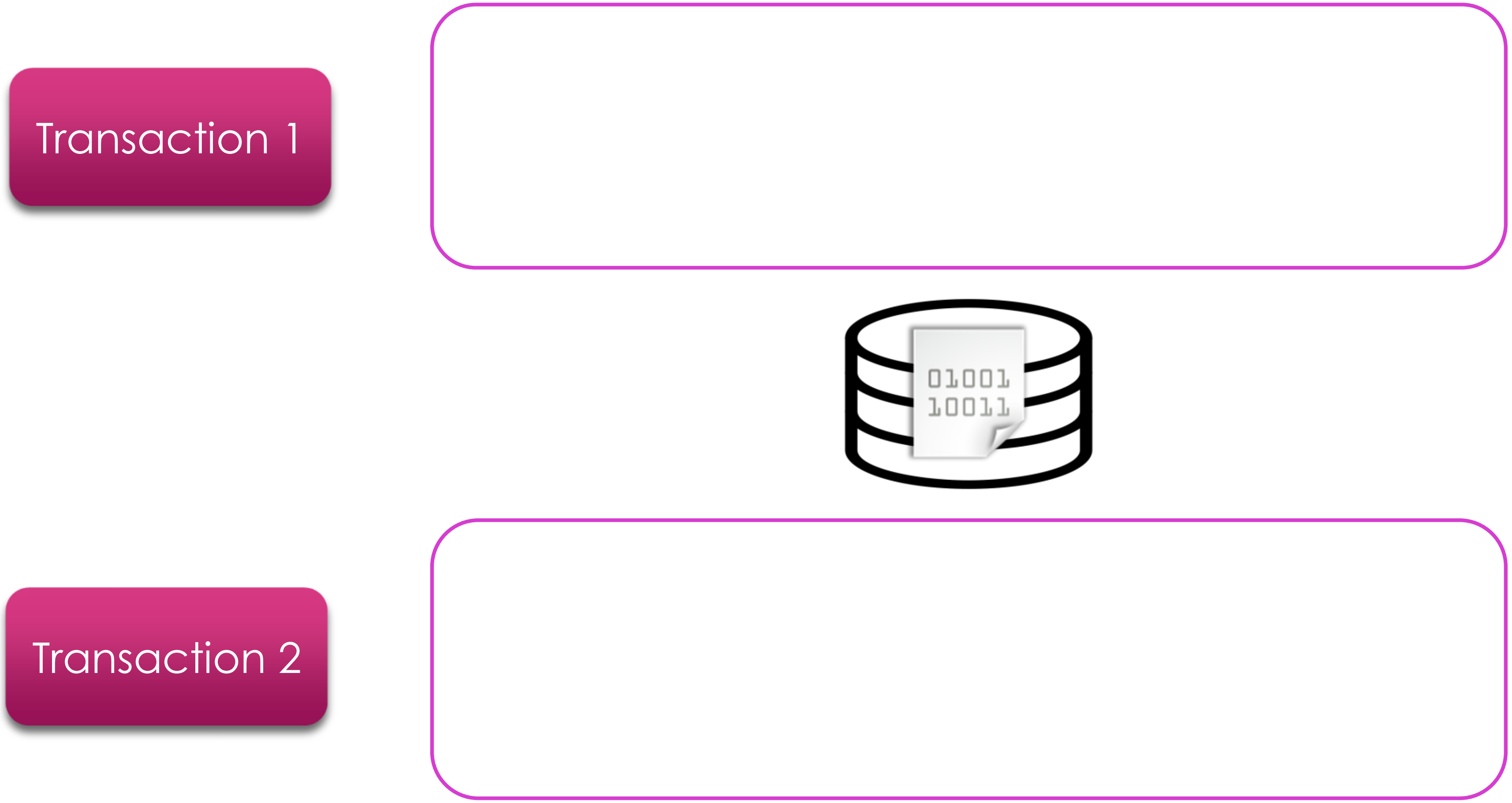
Dirty Read



Transaction 1

Transaction 2

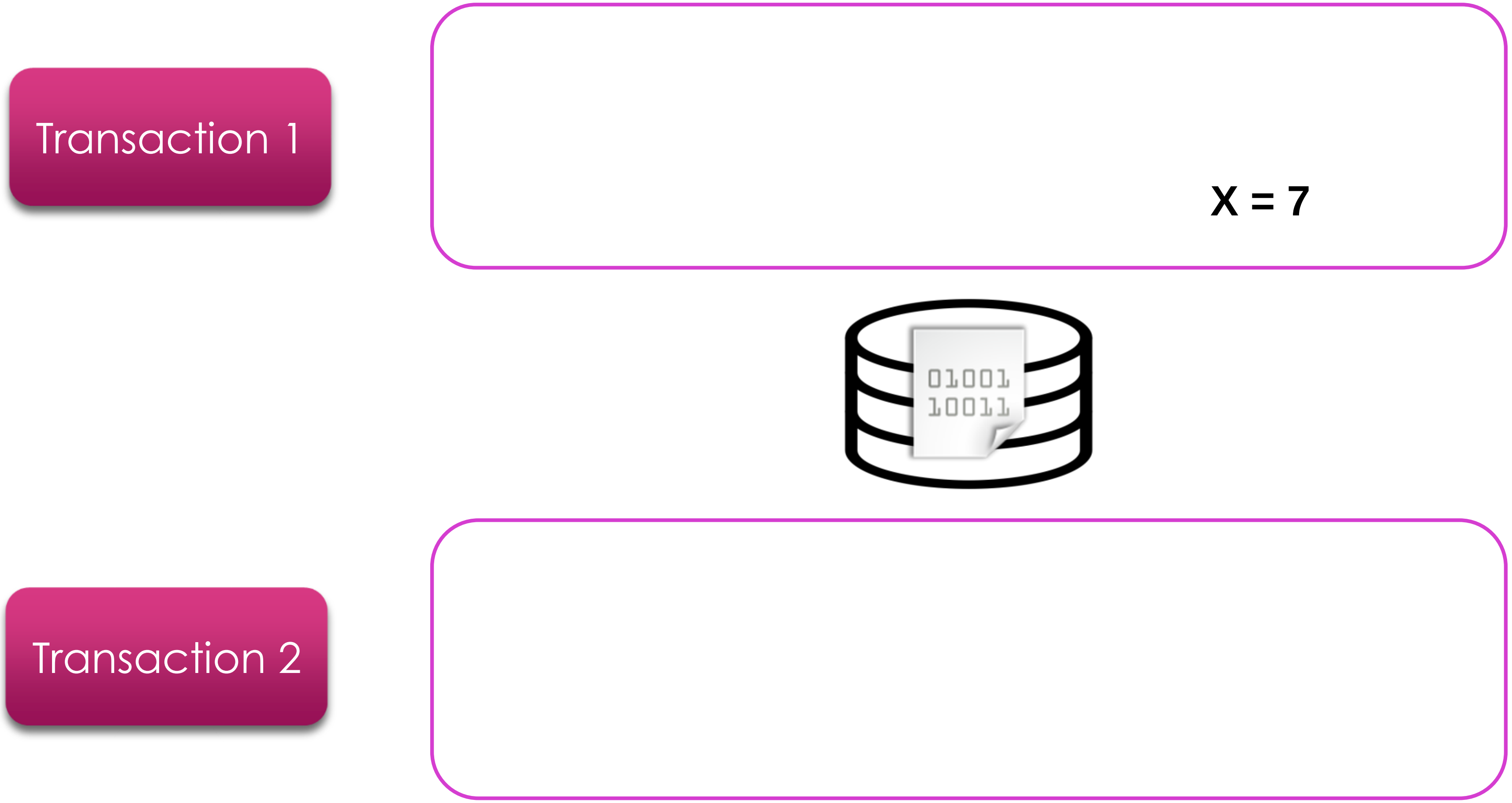
Dirty Read



Transaction 1

Transaction 2

Dirty Read

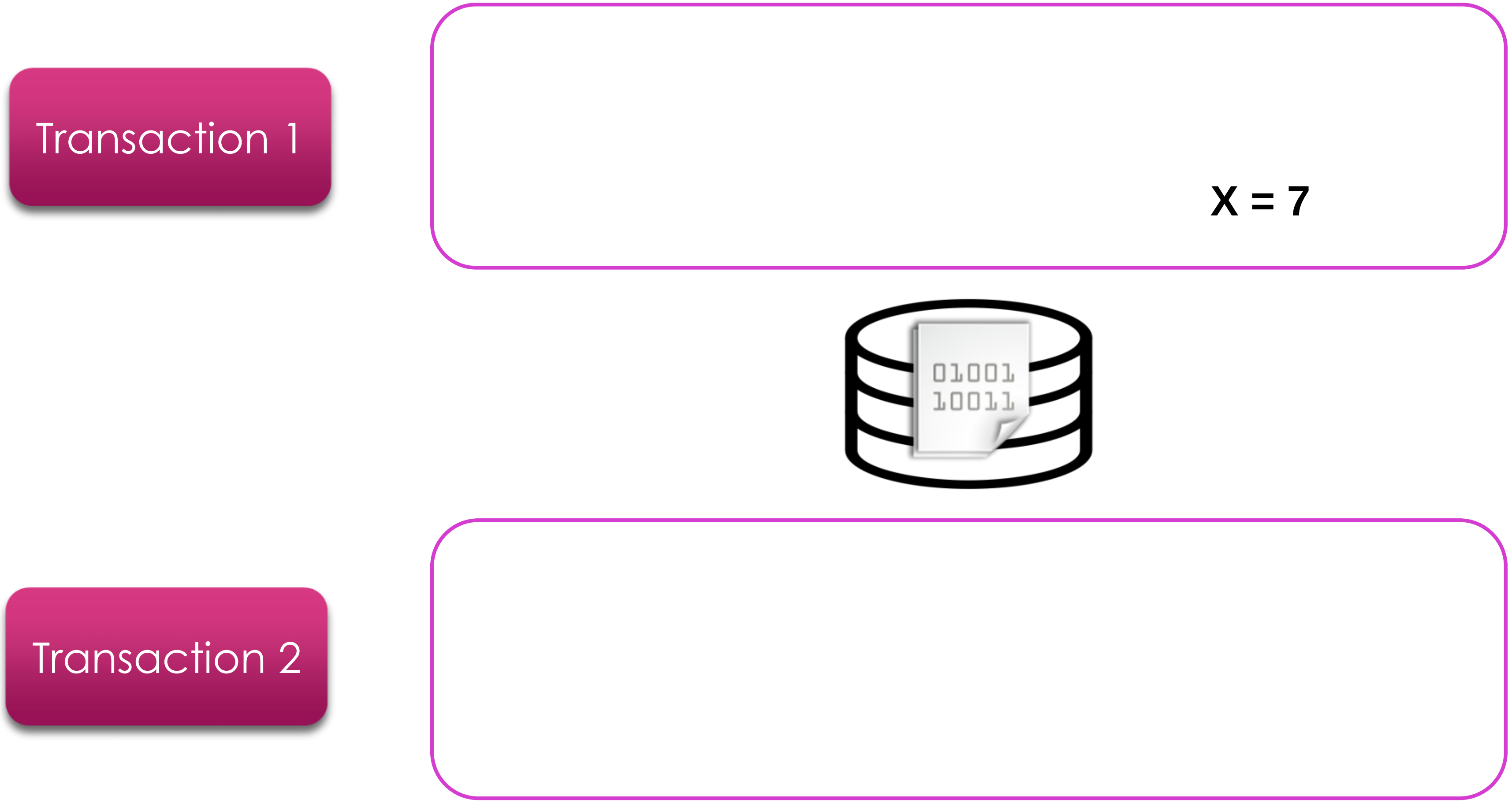


Transaction 1

X = 7

Transaction 2

Dirty Read

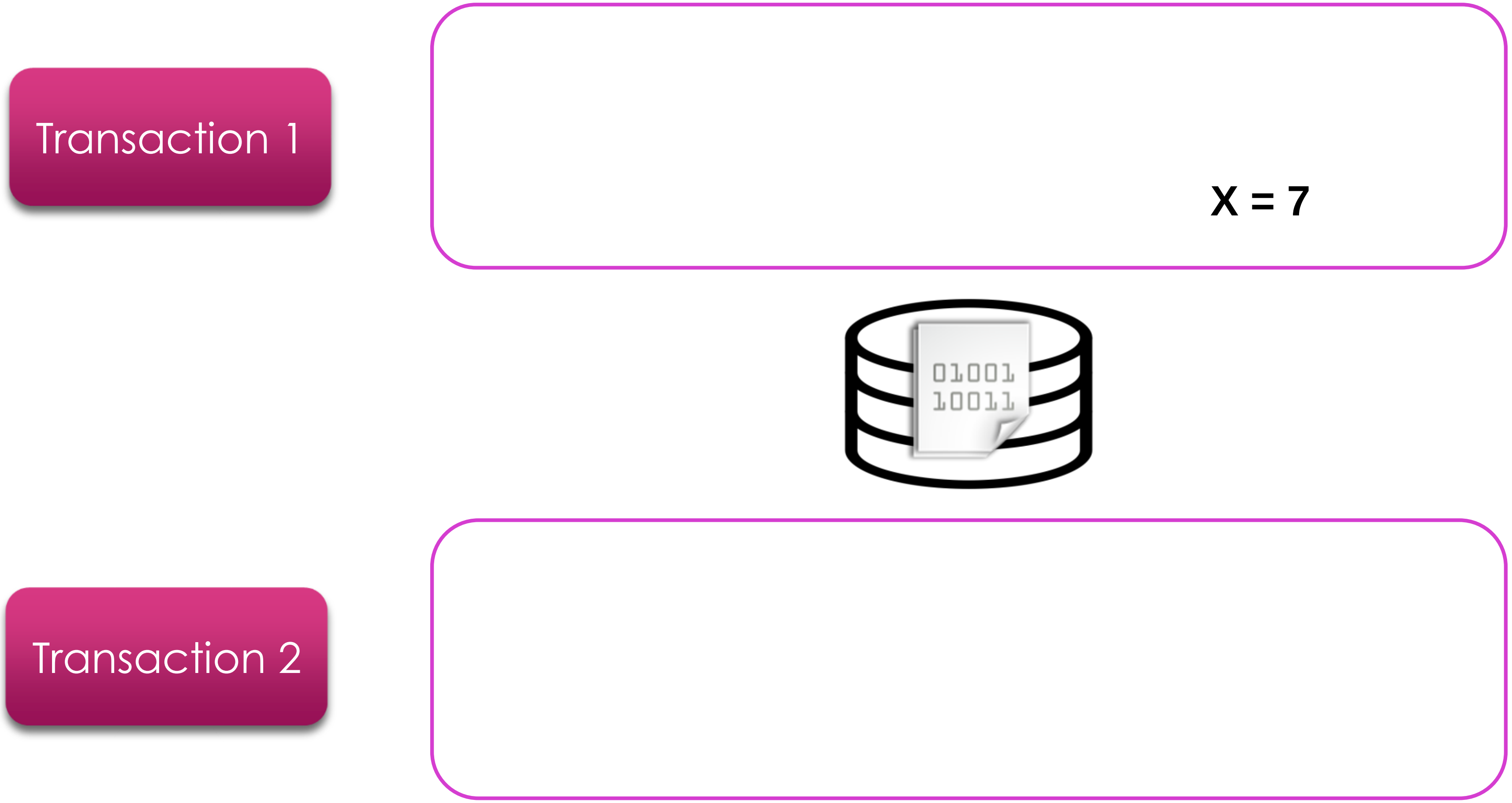


Transaction 1

X = 7

Transaction 2

Dirty Read

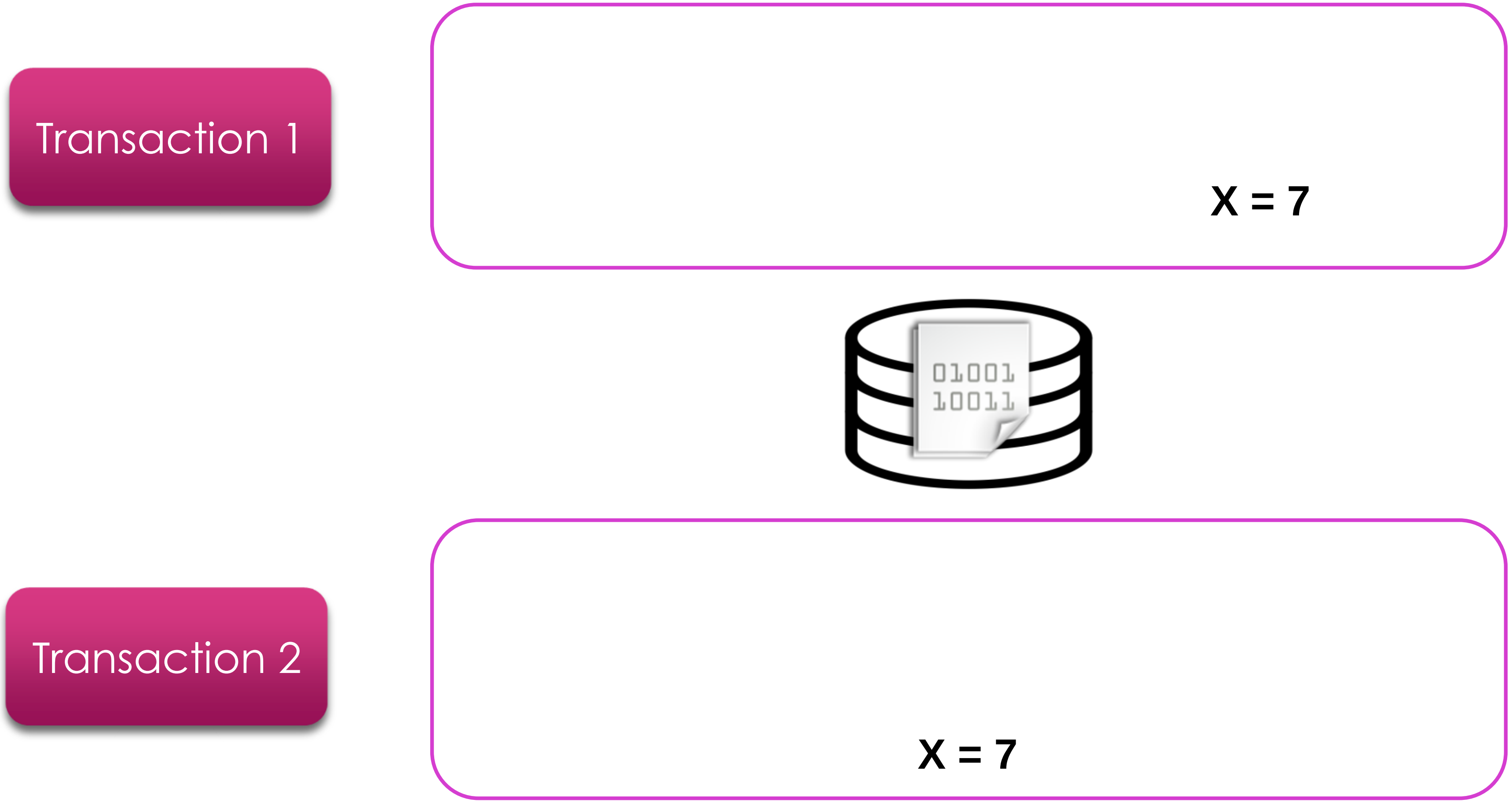


Transaction 1

X = 7

Transaction 2

Dirty Read



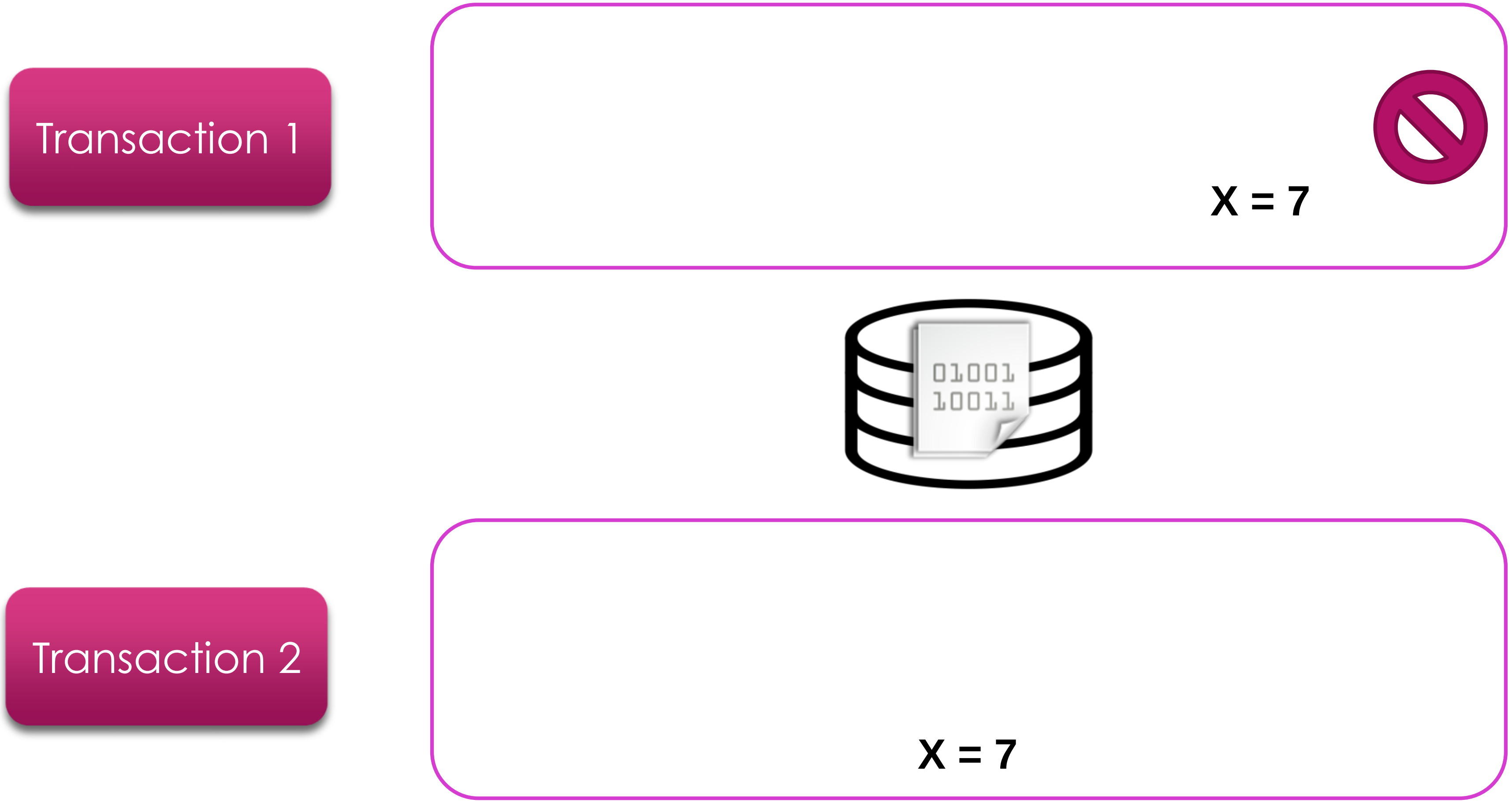
Transaction 1

X = 7

Transaction 2

X = 7

Dirty Read



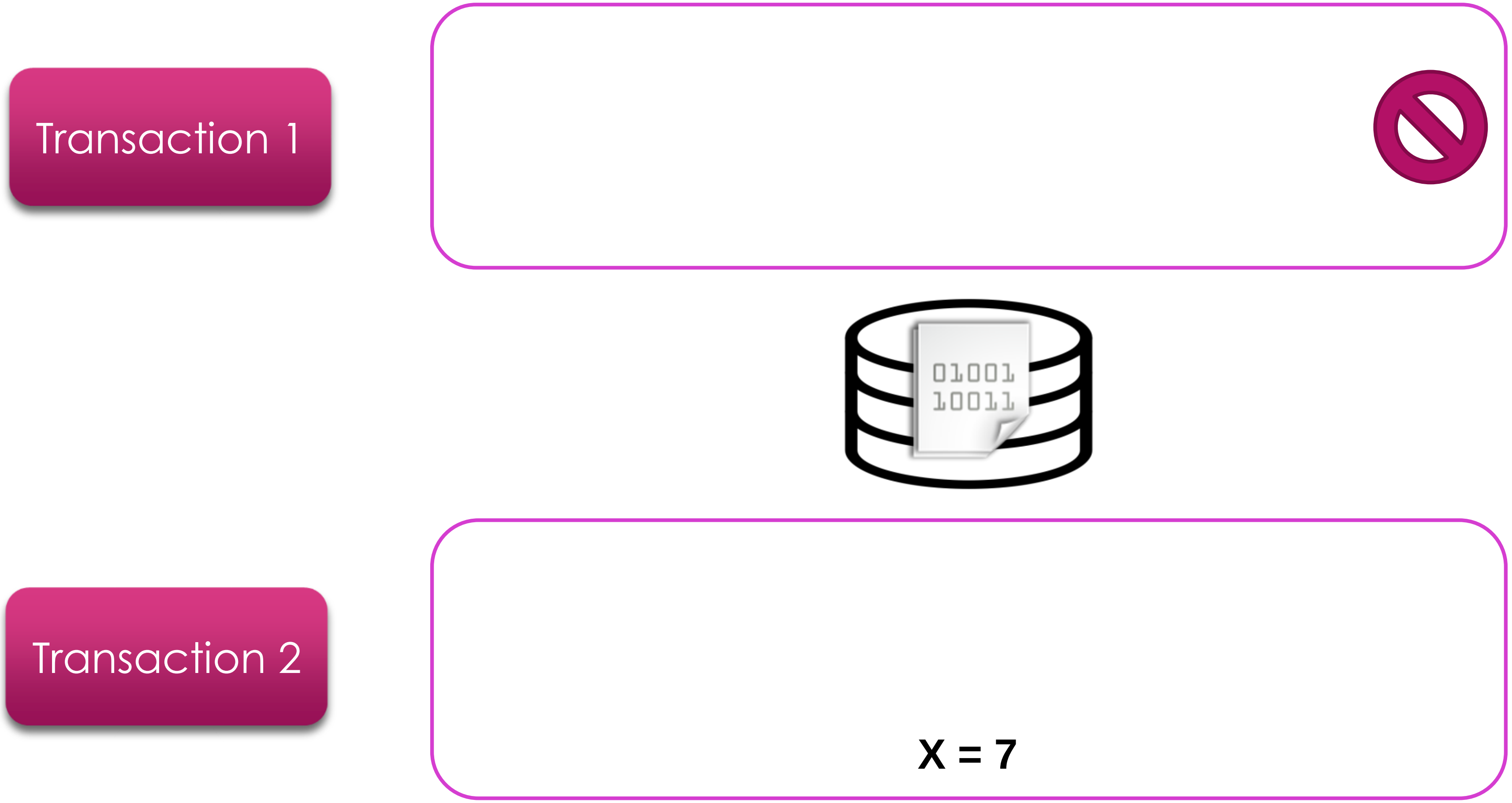
Transaction 1

X = 7

Transaction 2

X = 7

Dirty Read

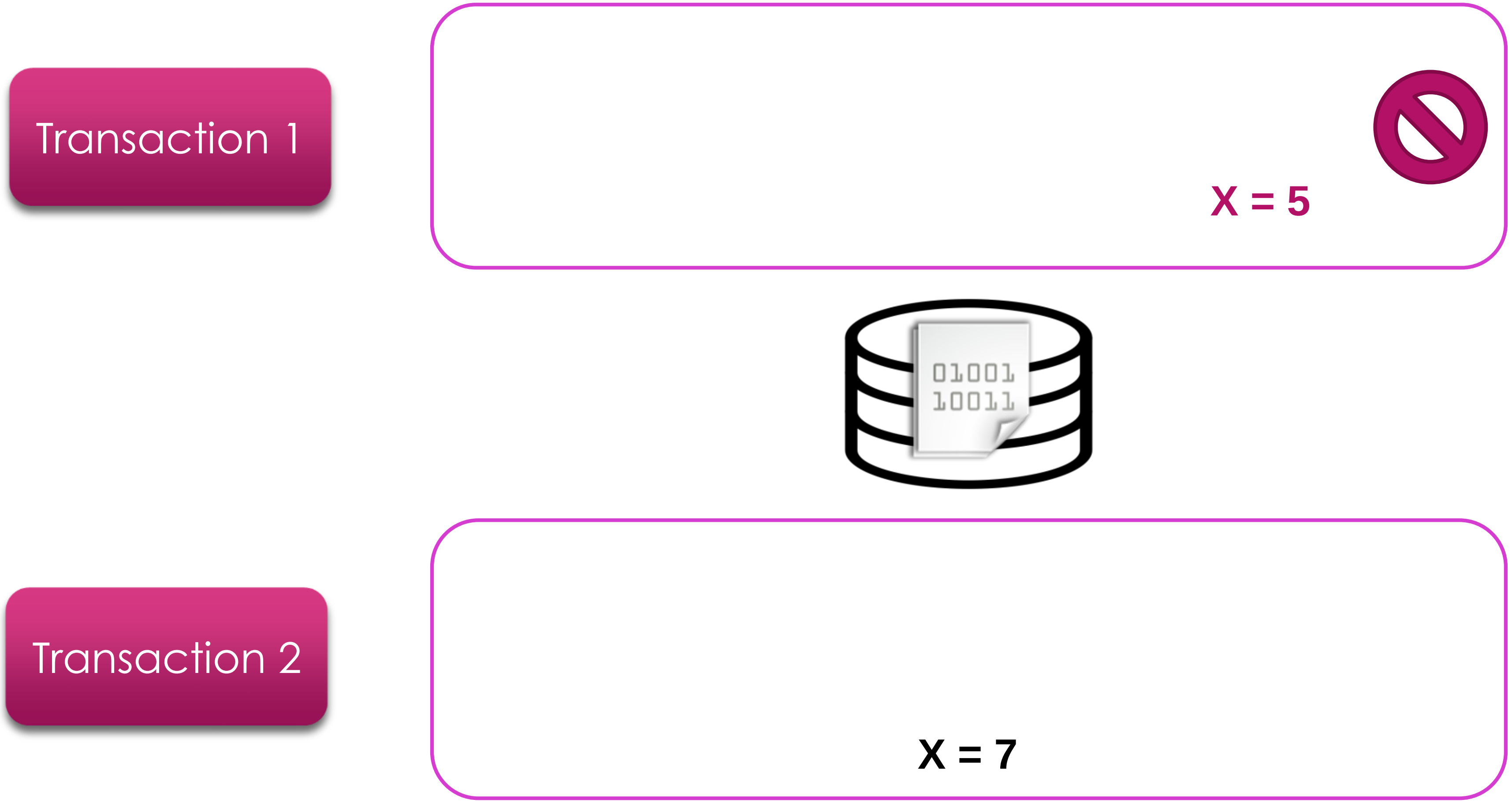


Transaction 1

Transaction 2

X = 7

Dirty Read



Transaction 1

X = 5

Transaction 2

X = 7

Non-repeatable Read

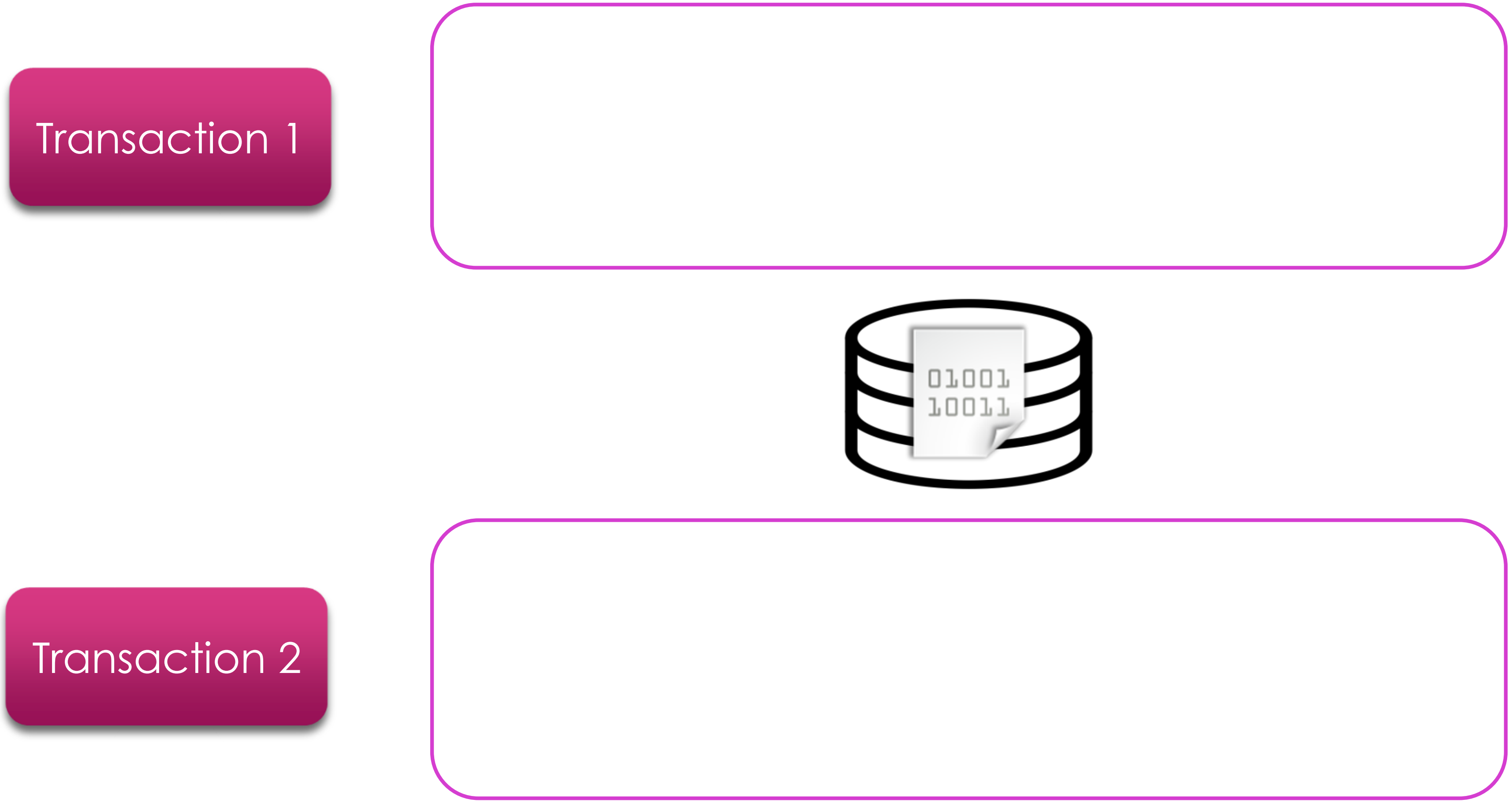
- A non-repeatable read occurs, when during the course of a transaction, a row is retrieved twice and the values within the row differ between reads.

Non-repeatable Read

Transaction 1

Transaction 2

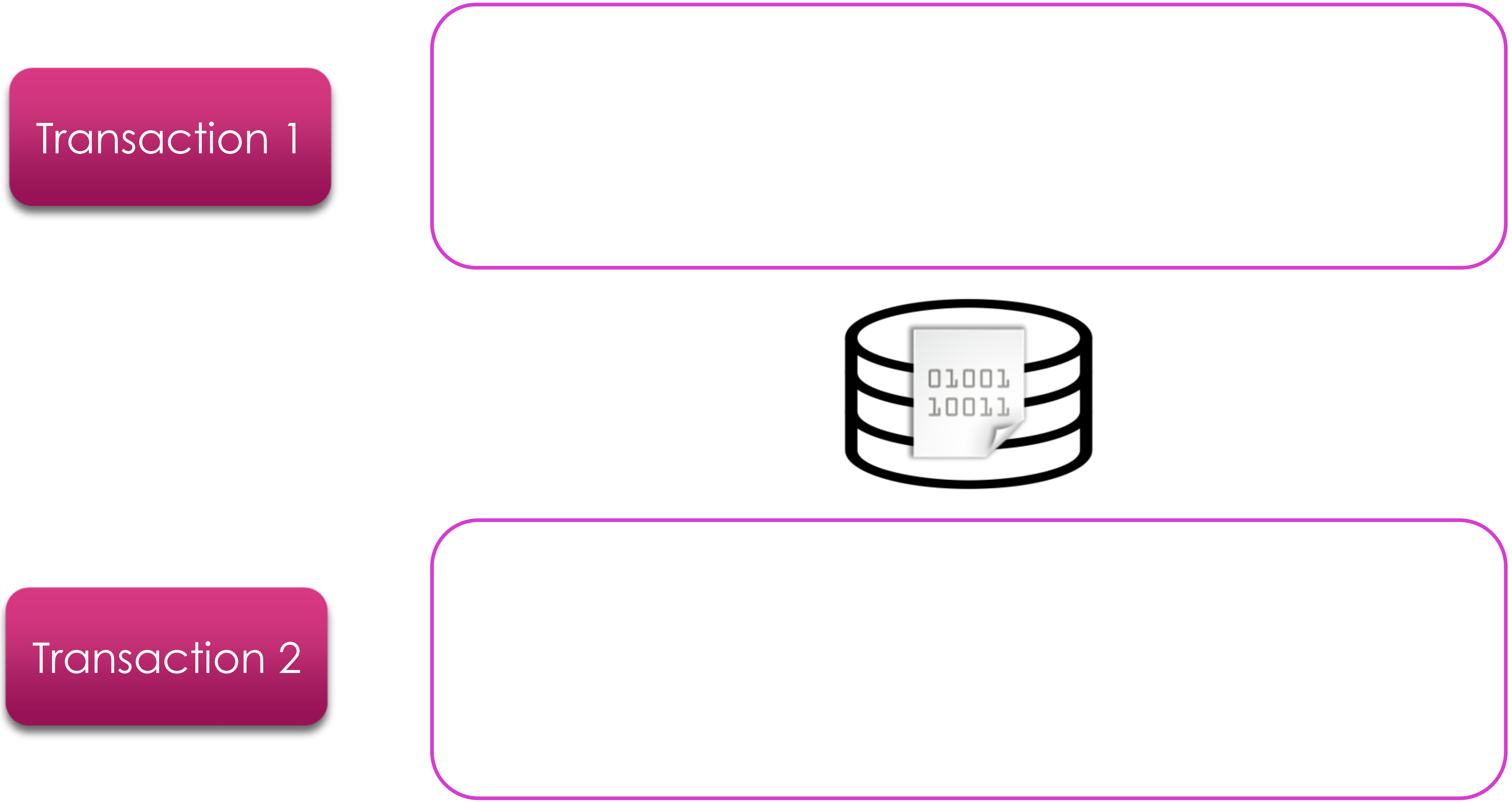
Non-repeatable Read



Transaction 1

Transaction 2

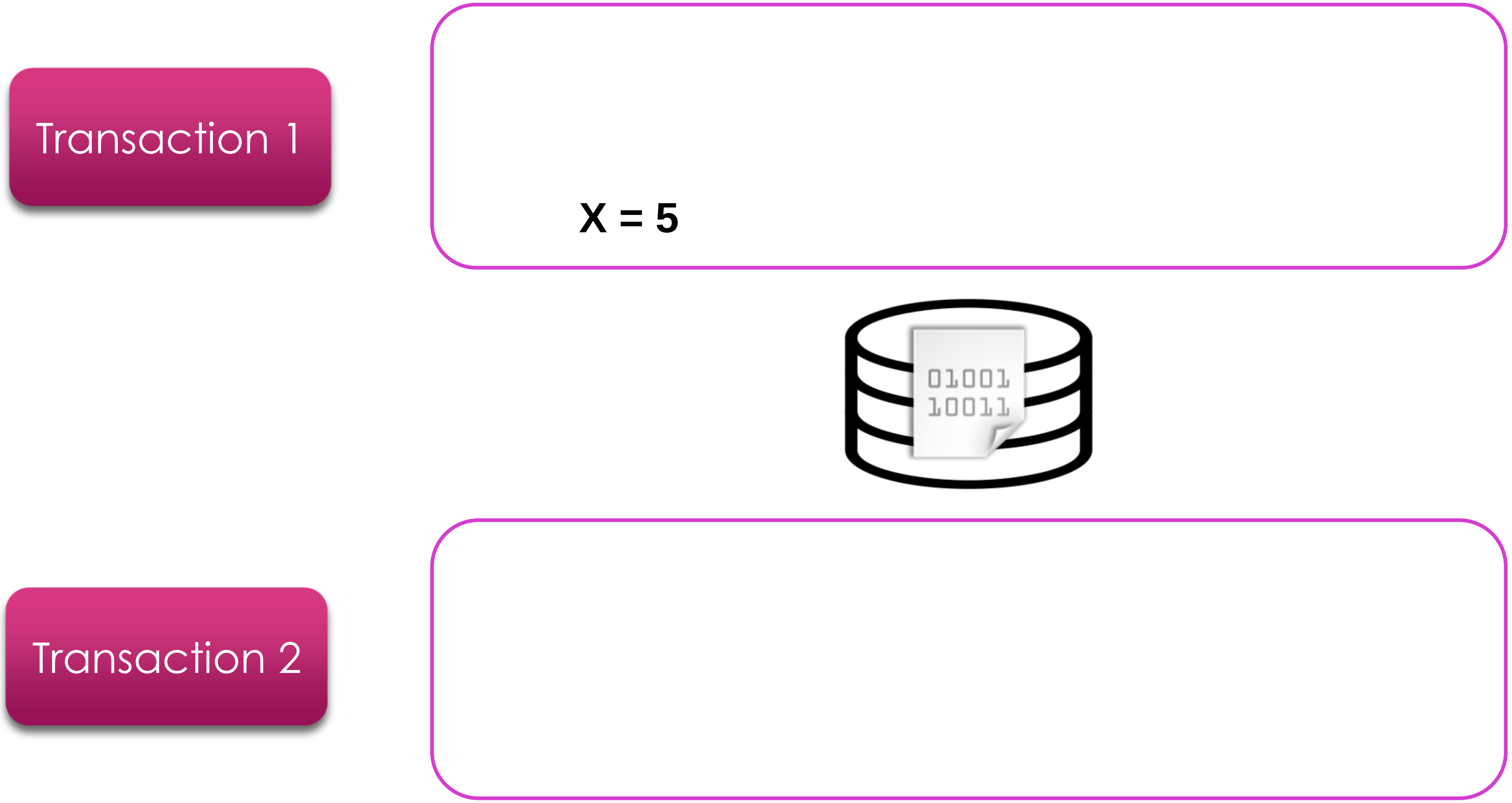
Non-repeatable Read



Transaction 1

Transaction 2

Non-repeatable Read

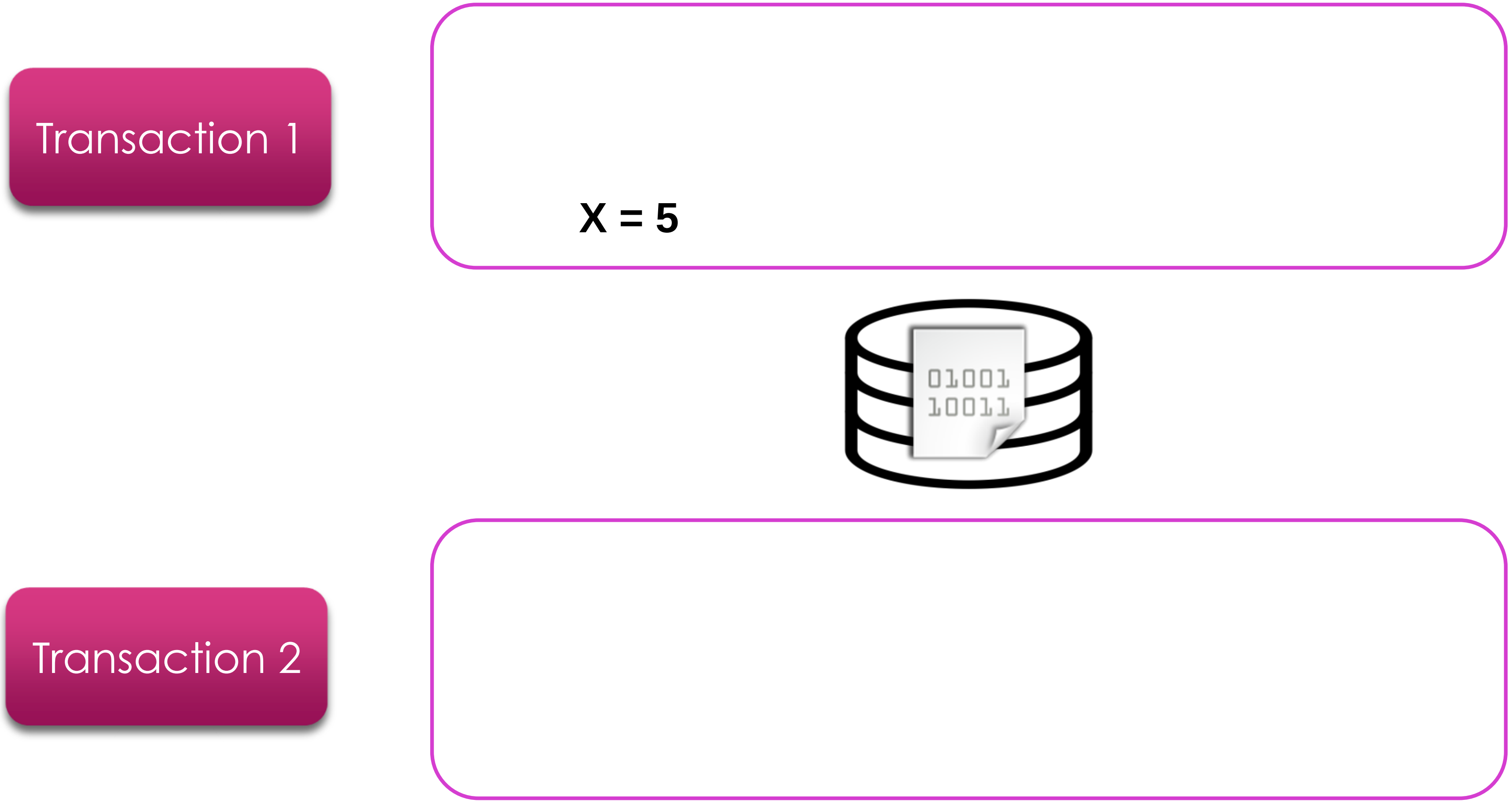


Transaction 1

X = 5

Transaction 2

Non-repeatable Read

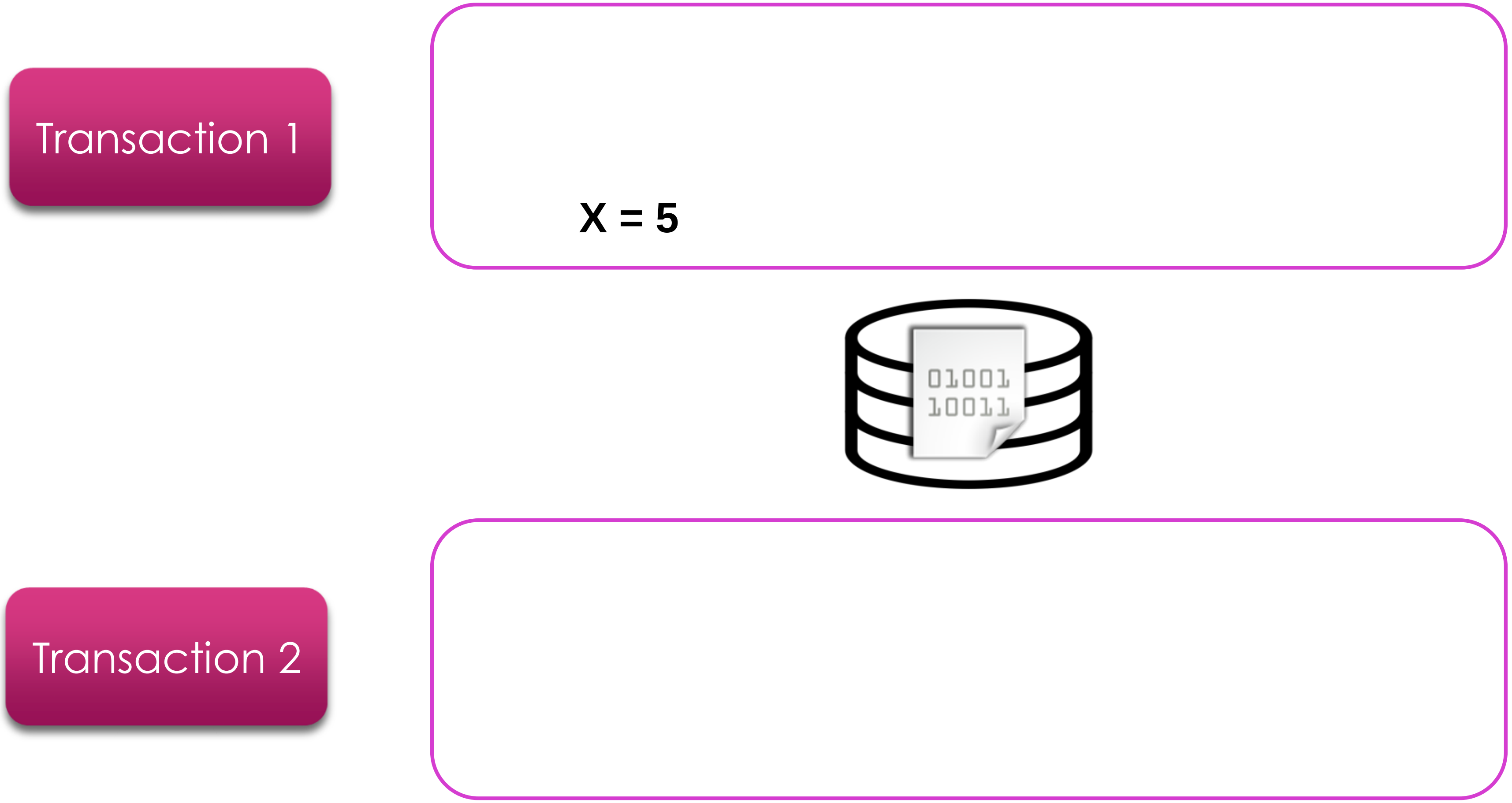


Transaction 1

X = 5

Transaction 2

Non-repeatable Read

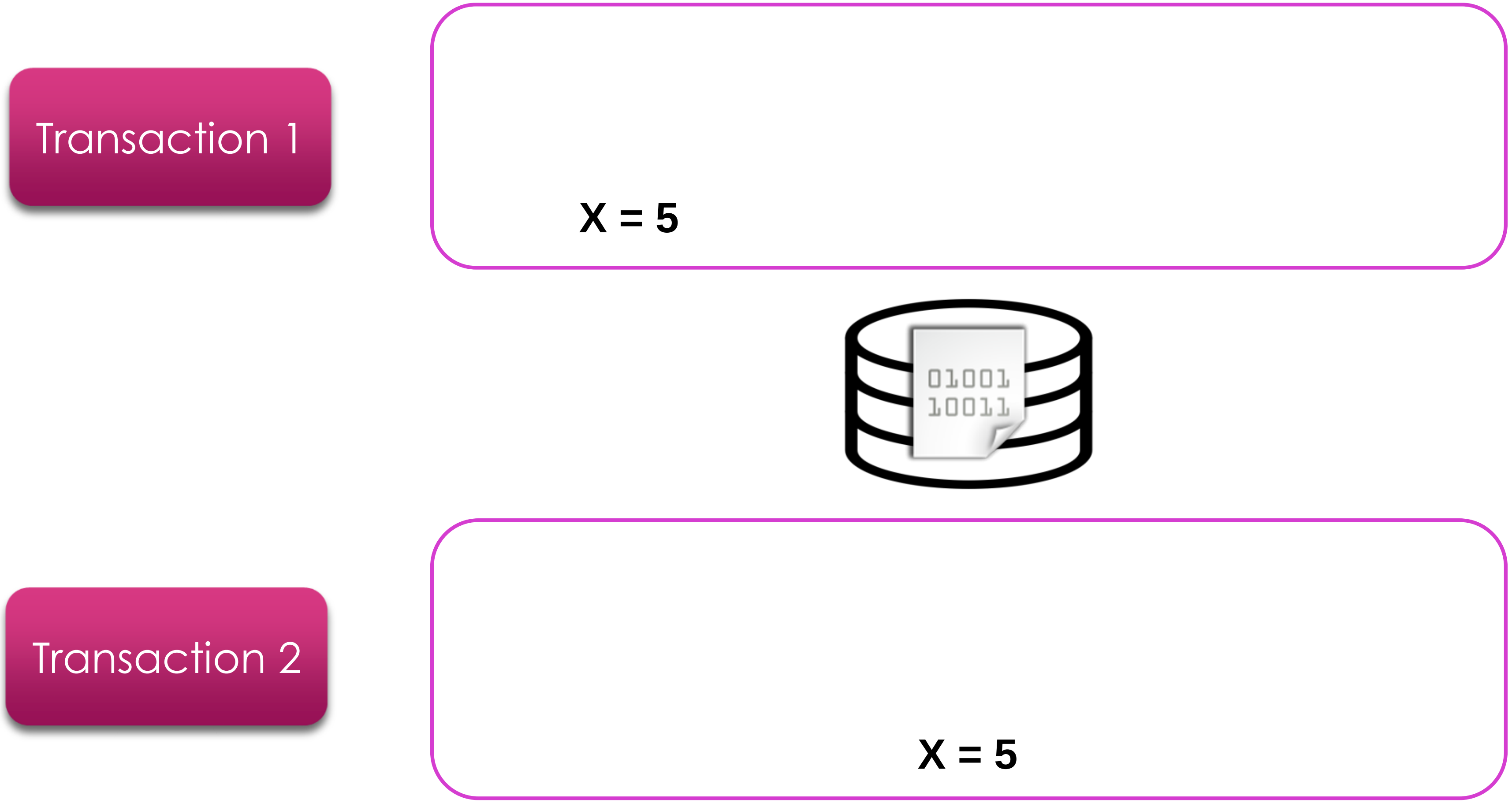


Transaction 1

X = 5

Transaction 2

Non-repeatable Read



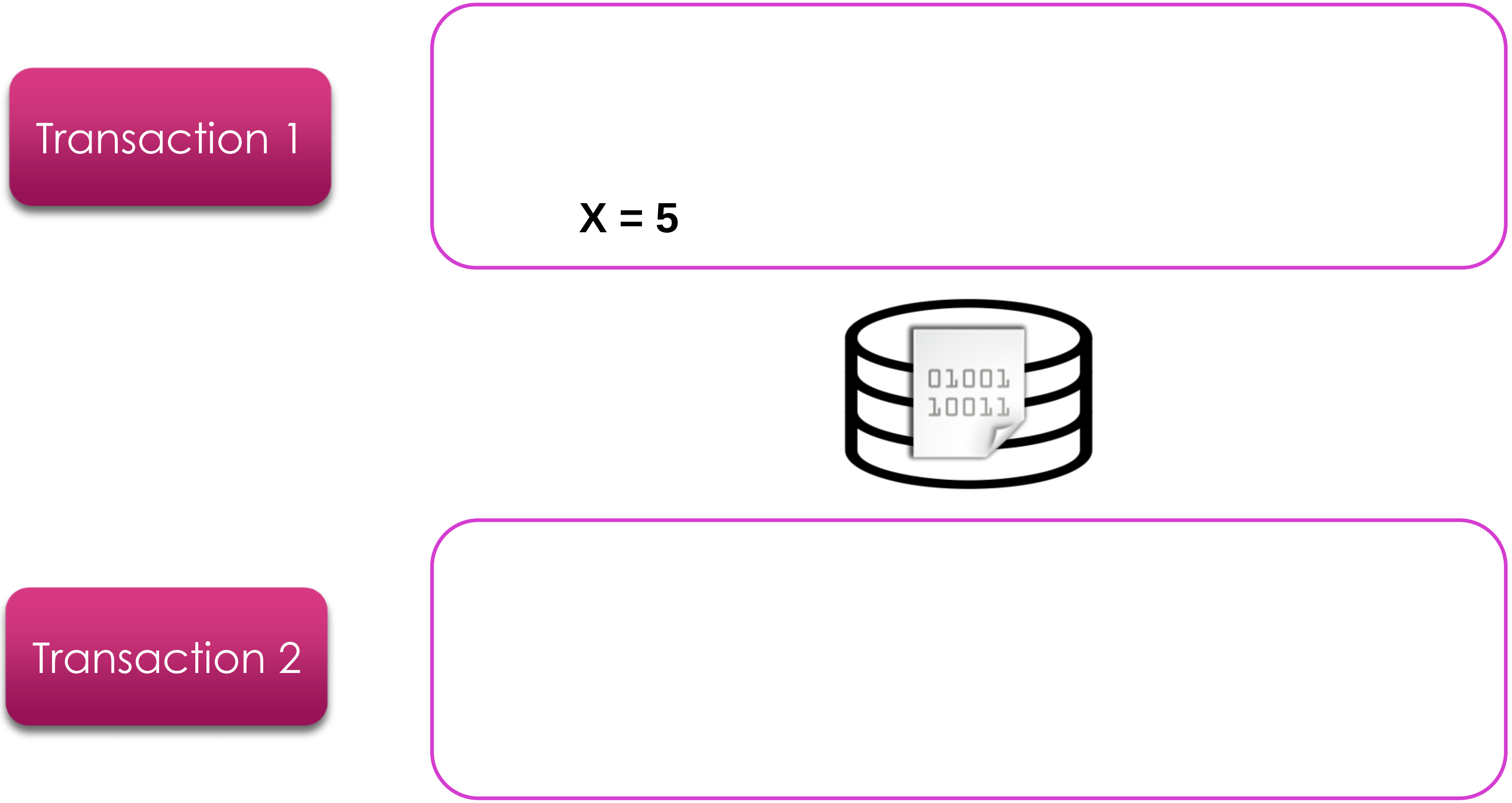
Transaction 1

X = 5

Transaction 2

X = 5

Non-repeatable Read

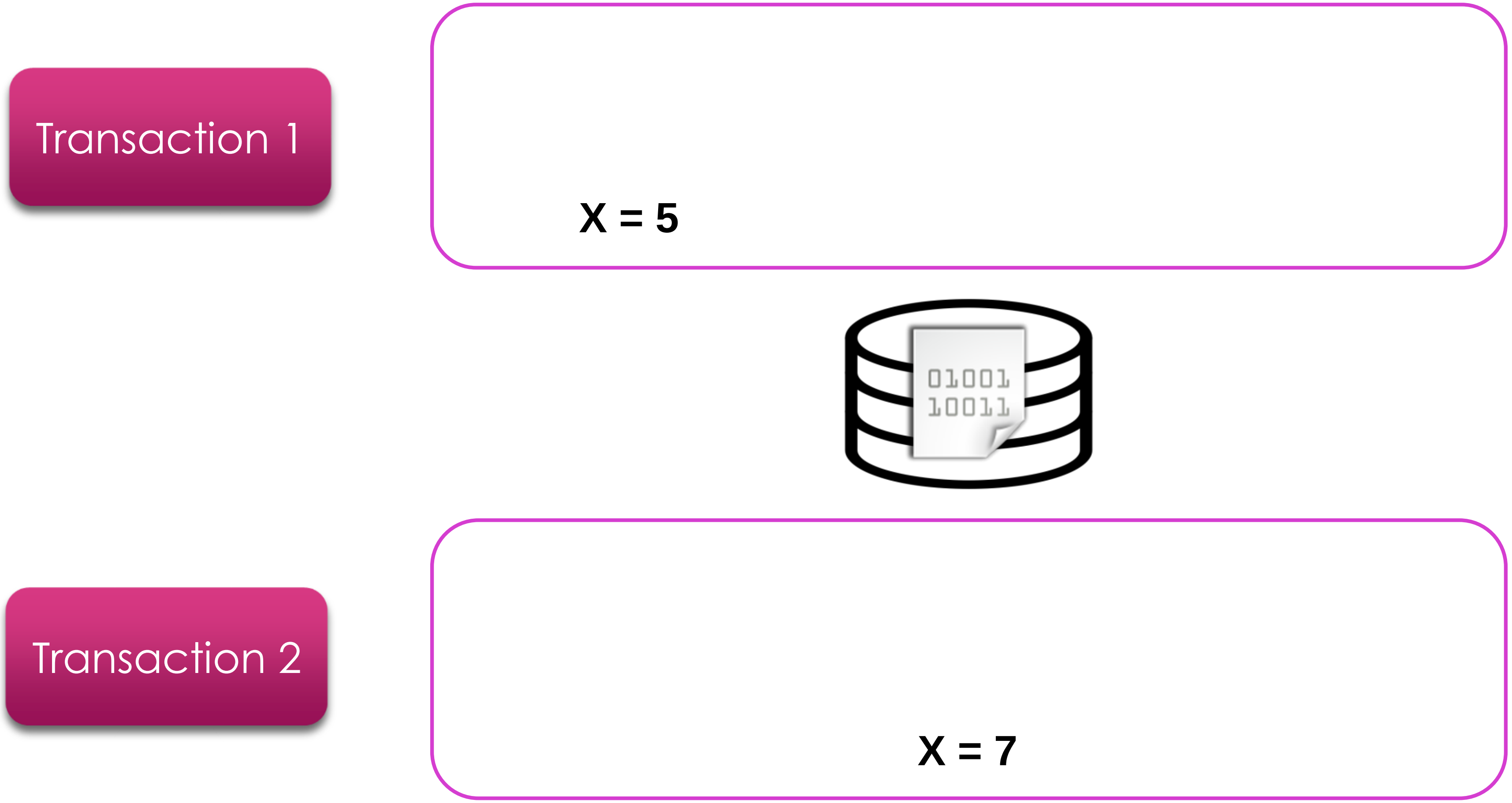


Transaction 1

X = 5

Transaction 2

Non-repeatable Read



Transaction 1

X = 5

Transaction 2

X = 7

Non-repeatable Read

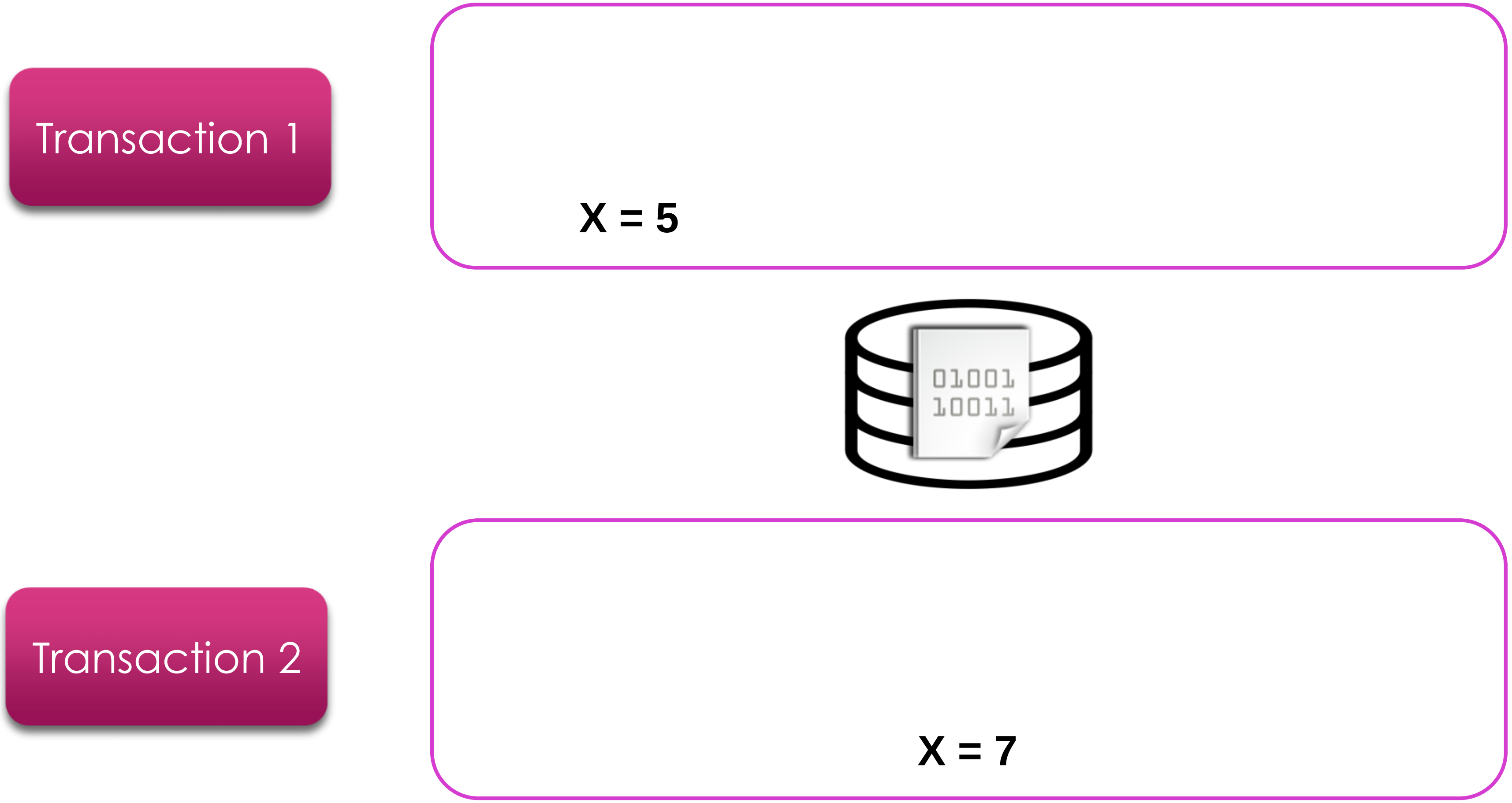
Transaction 1

X = 5

Transaction 2

X = 7

Non-repeatable Read



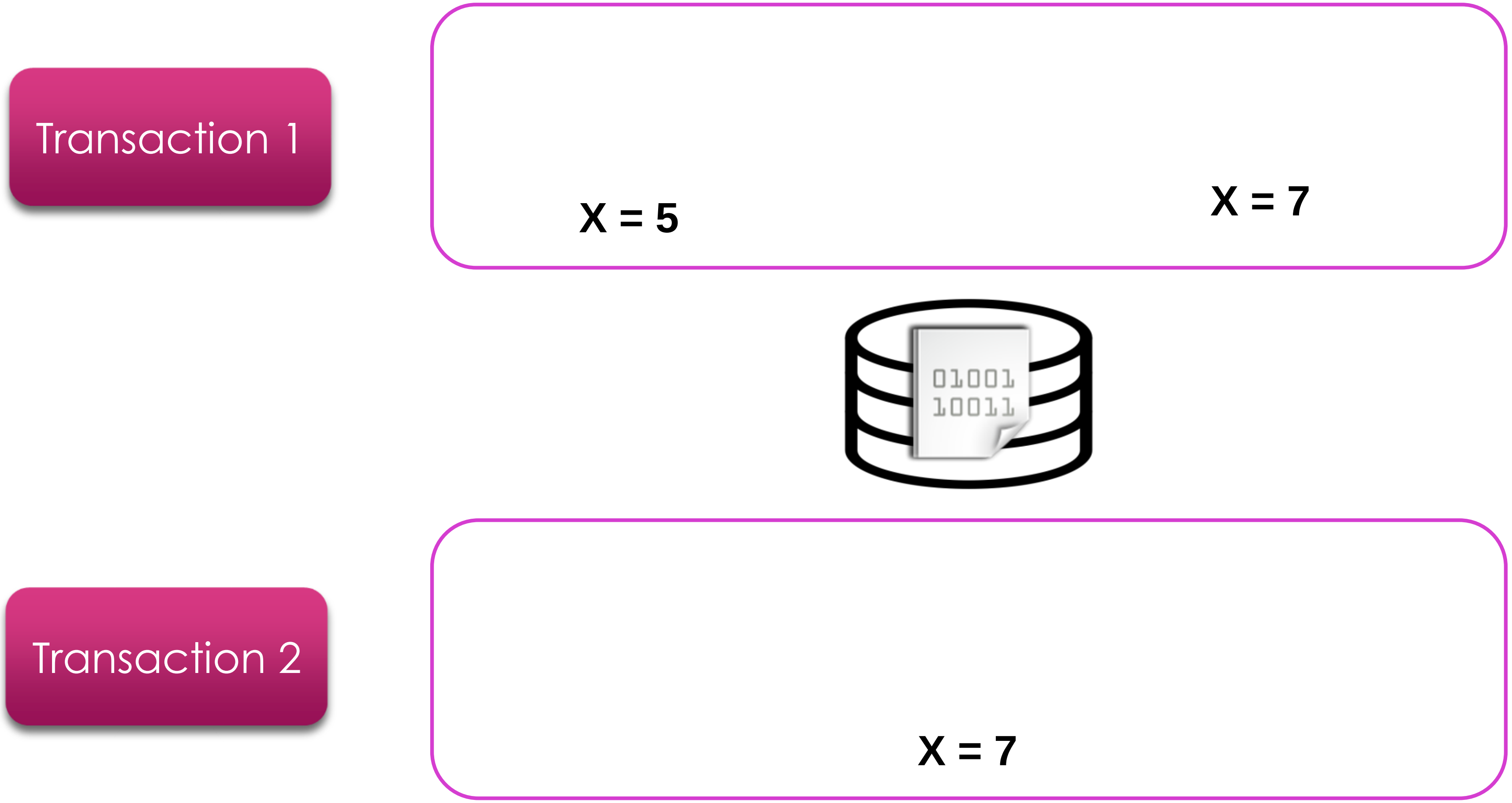
Transaction 1

X = 5

Transaction 2

X = 7

Non-repeatable Read



Transaction 1

X = 5

X = 7

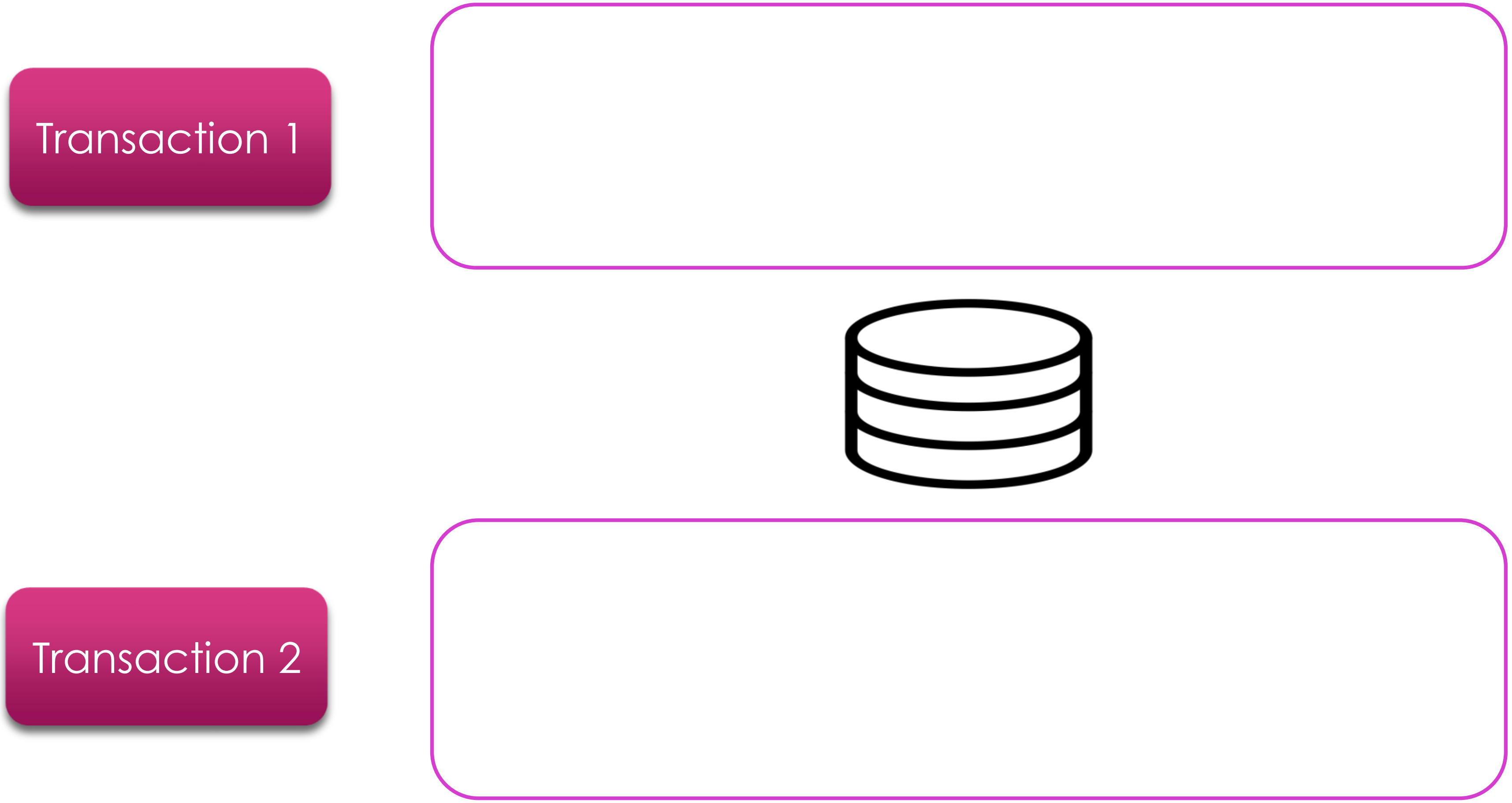
Transaction 2

X = 7

Phantom Read

- A phantom read occurs when, in the course of a transaction, new rows are added by another transaction to the records being read.

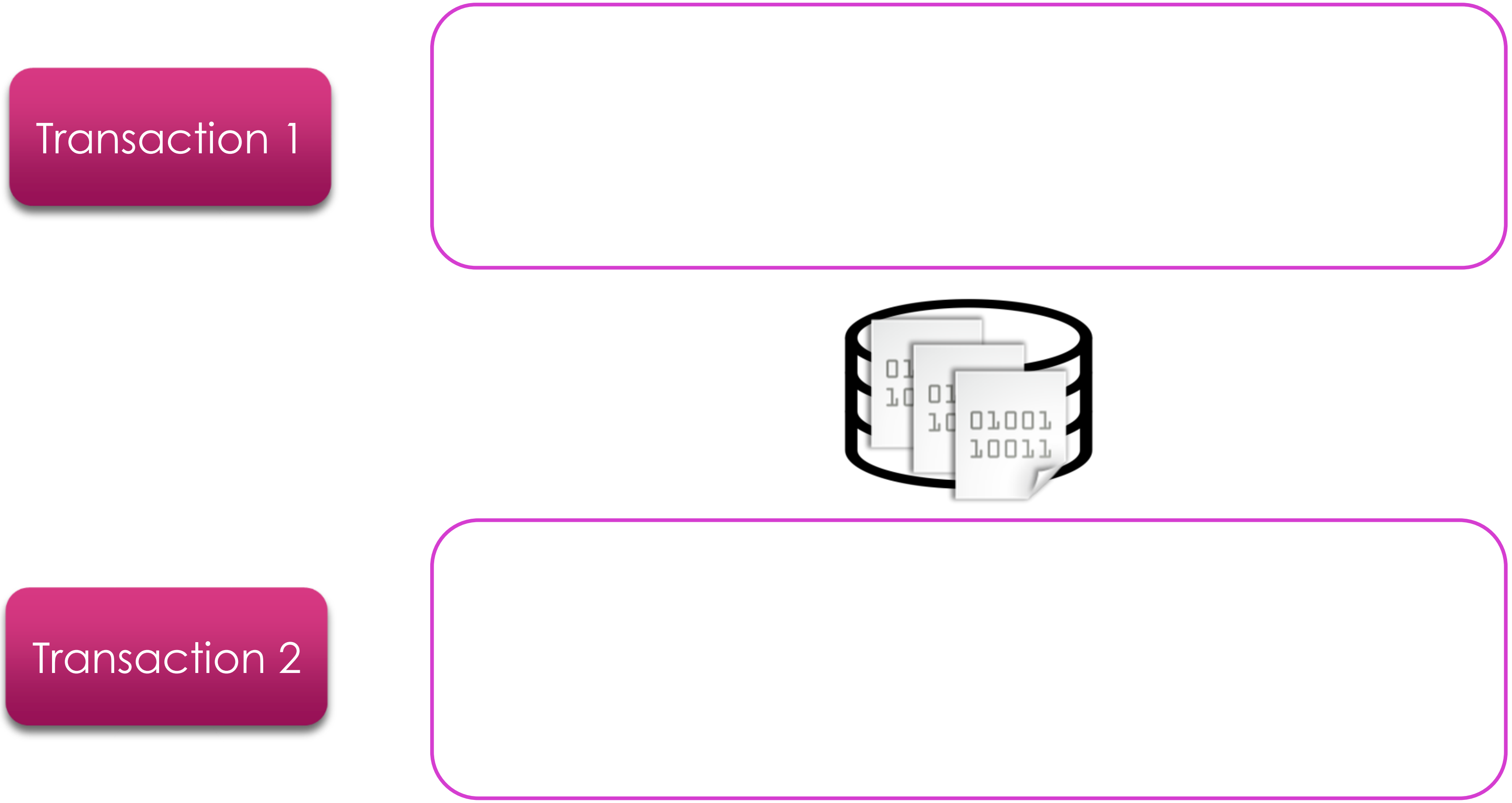
Phantom Read



Transaction 1

Transaction 2

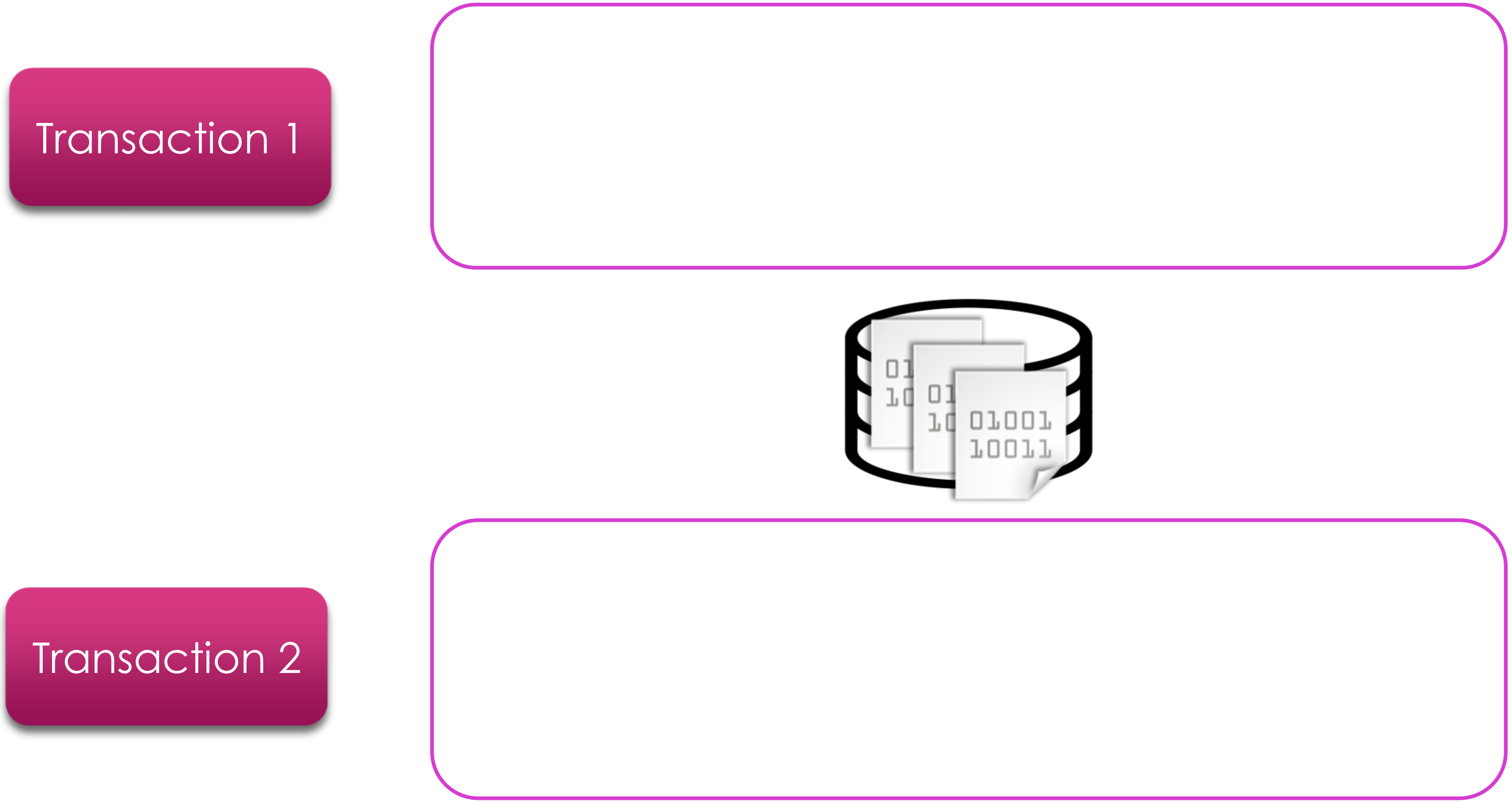
Phantom Read



Transaction 1

Transaction 2

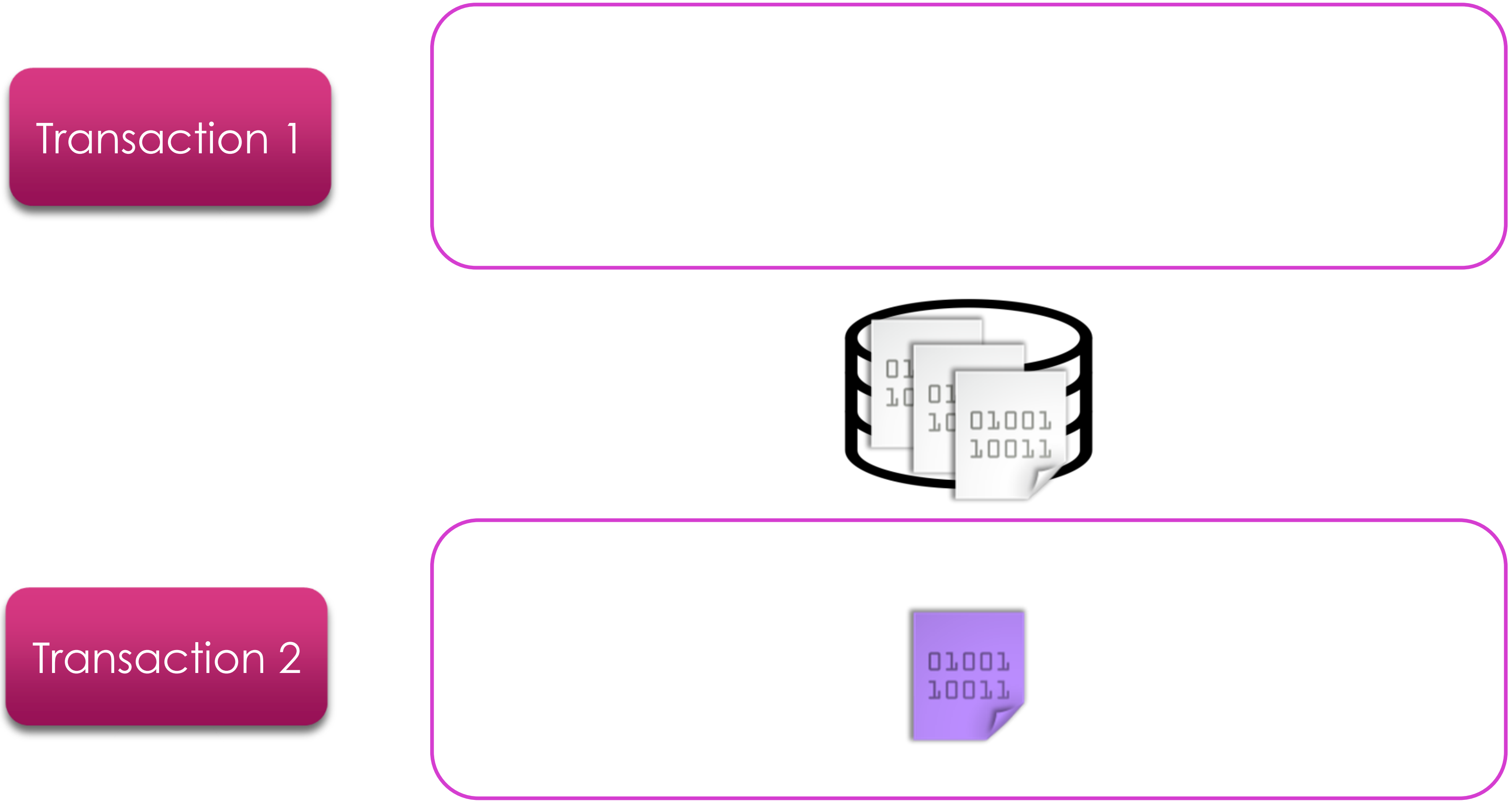
Phantom Read



Transaction 1

Transaction 2

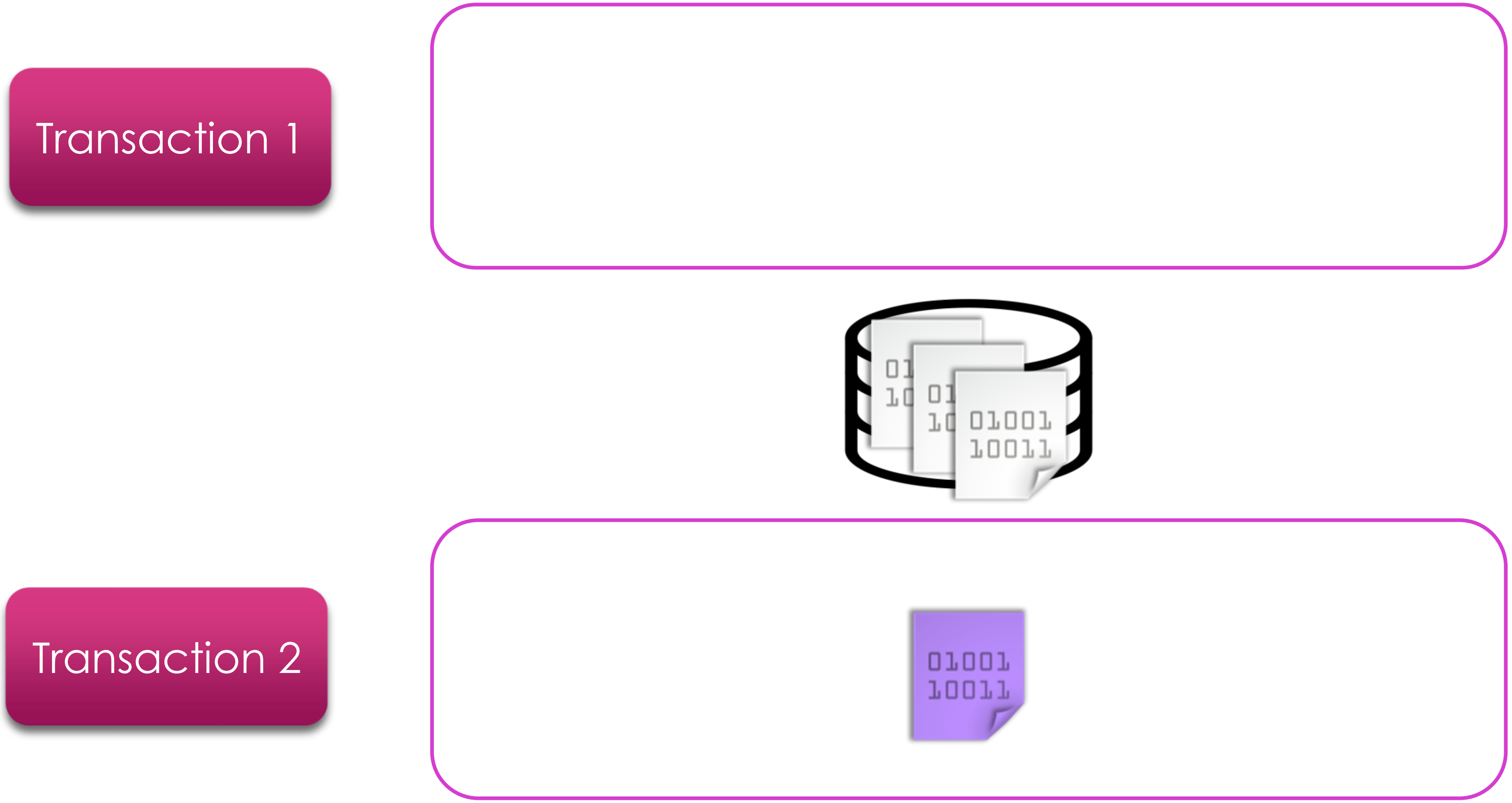
Phantom Read



Transaction 1

Transaction 2

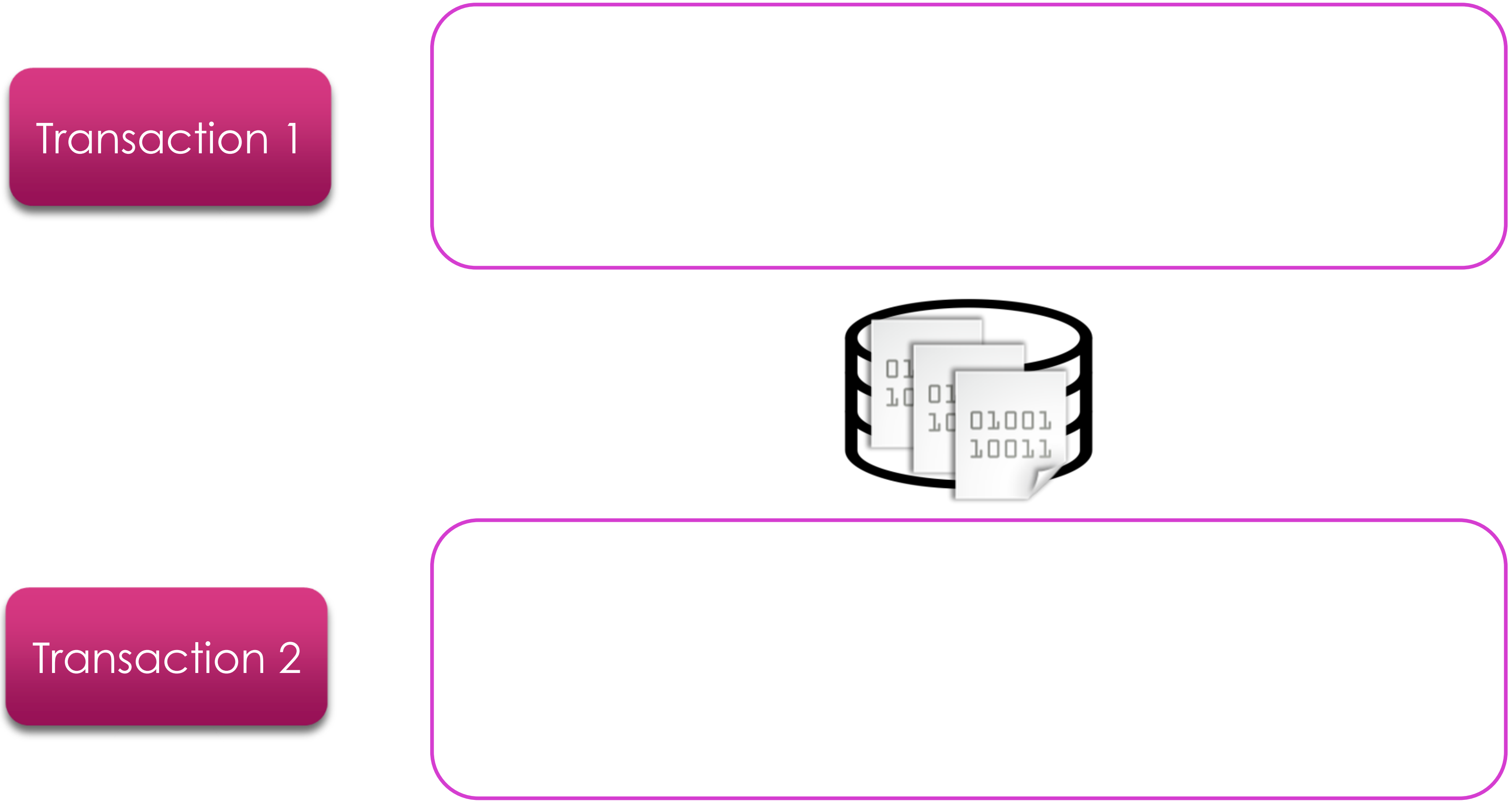
Phantom Read



Transaction 1

Transaction 2

Phantom Read



Phantom Read

Transaction 1

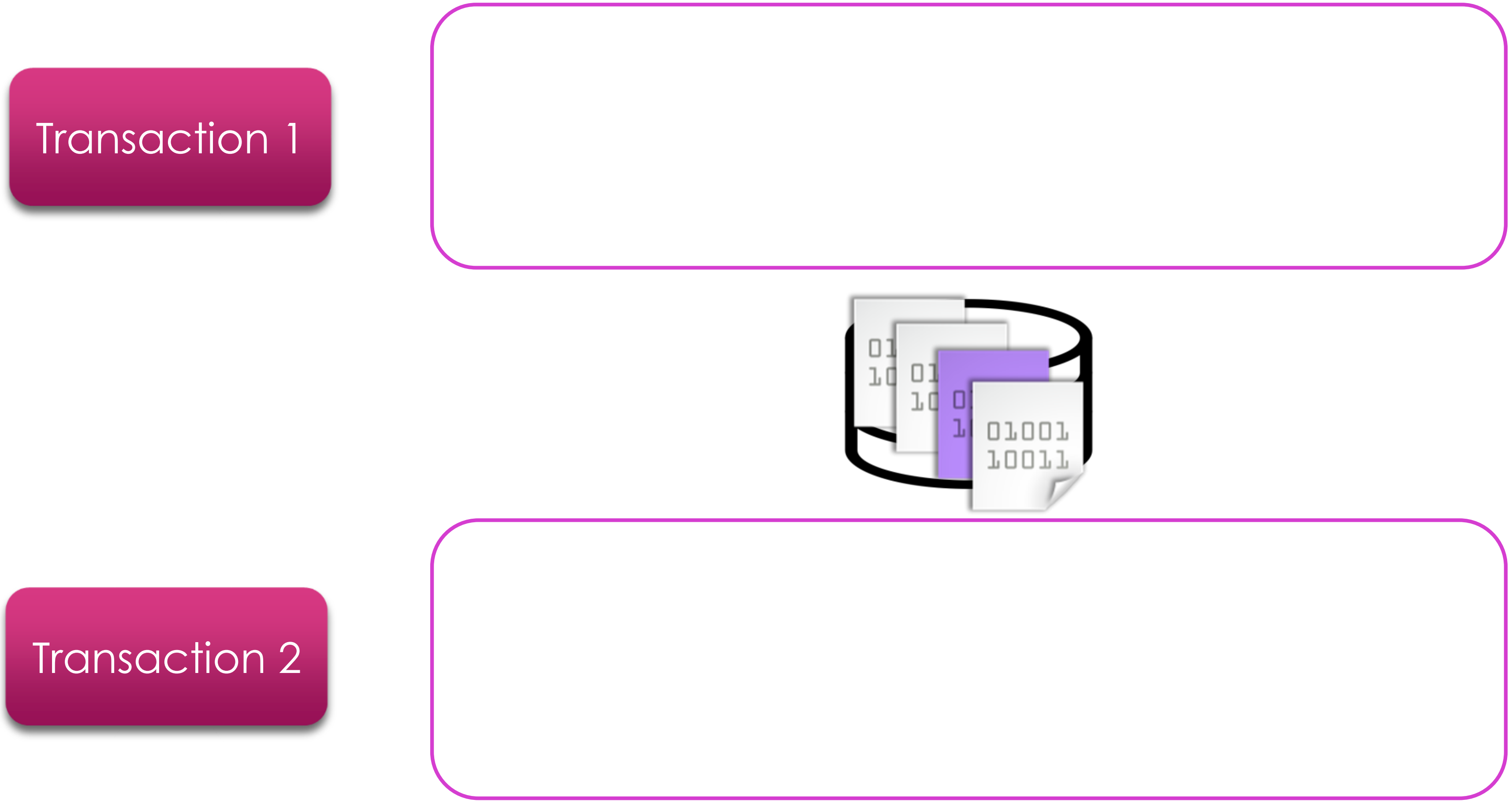
Transaction 2

Phantom Read

Transaction 1

Transaction 2

Phantom Read



Transaction 1

Transaction 2

Lost Update

- A lost update can be interpreted in one of two ways. In the first scenario, a lost update is considered to have taken place when data that has been updated by one transaction is overwritten by another transaction, before the first transaction is either committed or rolled back.

Lost Update

Transaction 1

ID	X
1	

Transaction 2

Lost Update

Transaction 1

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

X = 5

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

X = 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	5

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	

Transaction 2

Lost Update

Transaction 1

X = 5

ID	X
1	2

Transaction 2

Lost Update

Transaction 1

ID	X
1	2

Transaction 2

Lost Update

Transaction 1

ID	X
1	2

Transaction 2

Lost Update

Transaction 1

X = 7

ID	X
1	2

Transaction 2

Lost Update

Transaction 1

ID	X
1	2

Transaction 2

Lost Update

Transaction 1

ID	X
1	2

Transaction 2

Lost Update

Transaction 1

ID	X
1	

Transaction 2

Lost Update

Transaction 1

ID	X
1	7

Transaction 2

Lost Update

Transaction 1

ID	X
1	7

Transaction 2

Missing or Double Reads

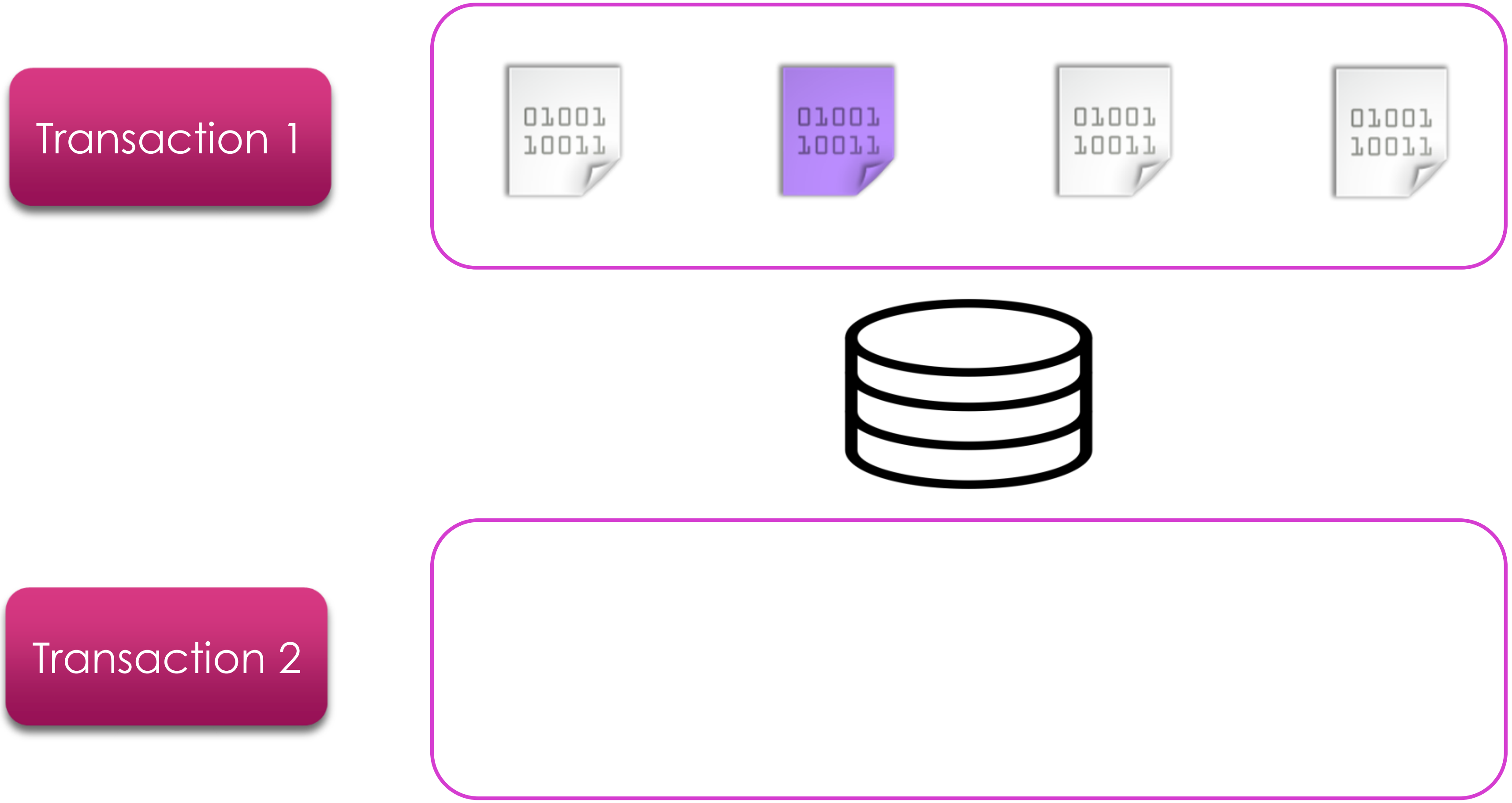
- Missing a updated row or seeing an updated row multiple times
- Missing one or more rows that were not the target of update

Missing or Double Reads

Transaction 1

Transaction 2

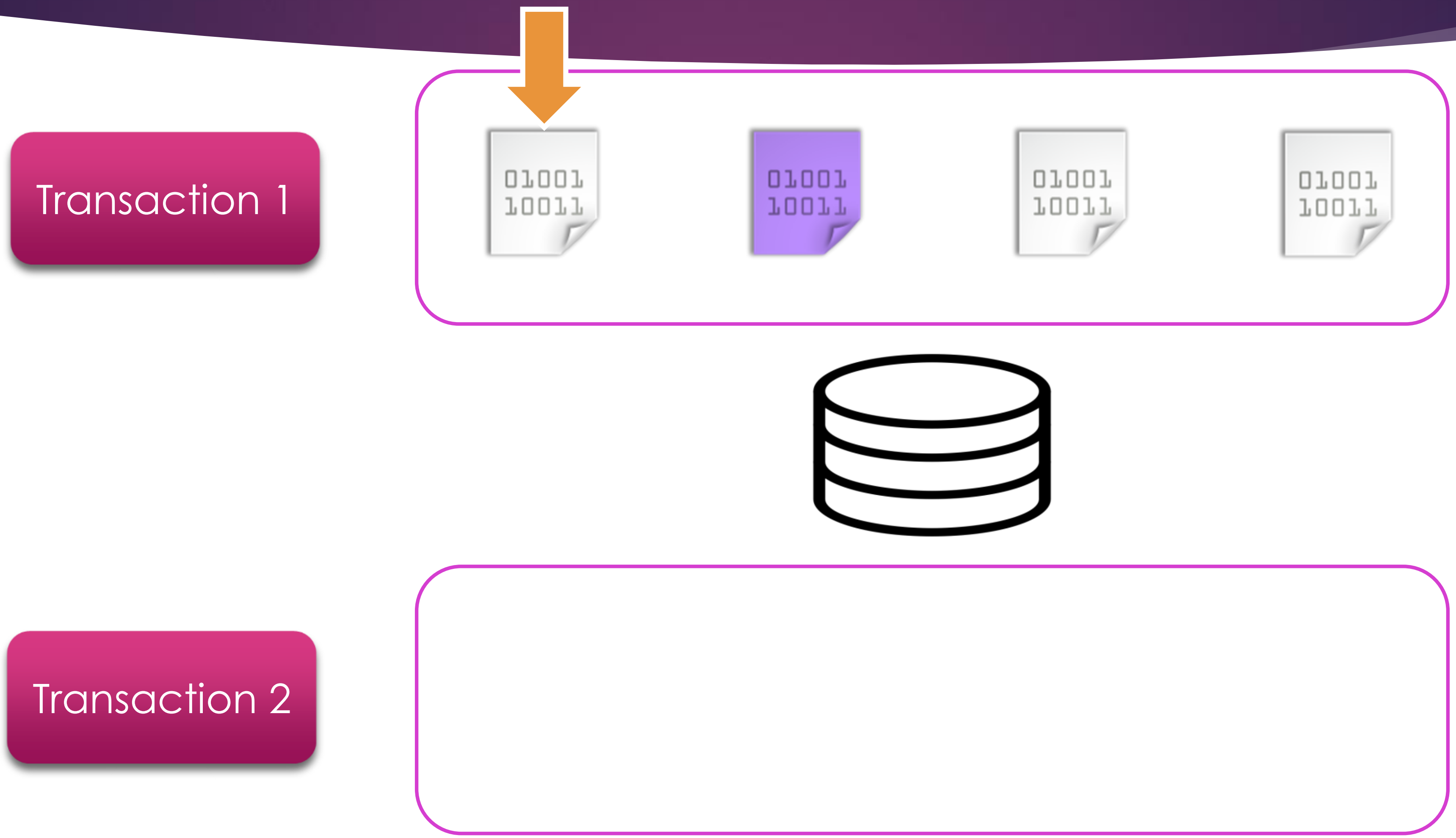
Missing or Double Reads



Transaction 1

Transaction 2

Missing or Double Reads



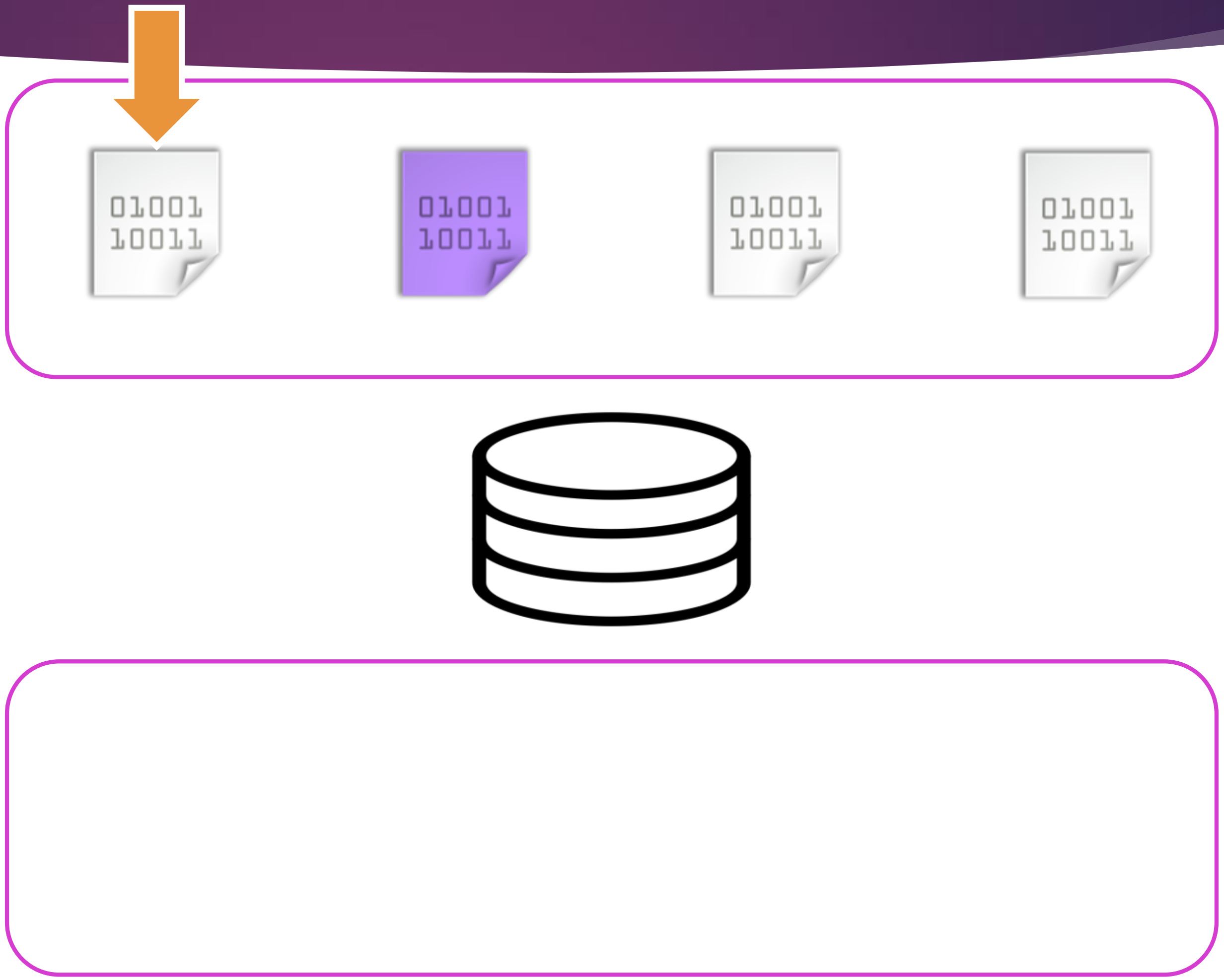
Transaction 1

Transaction 2

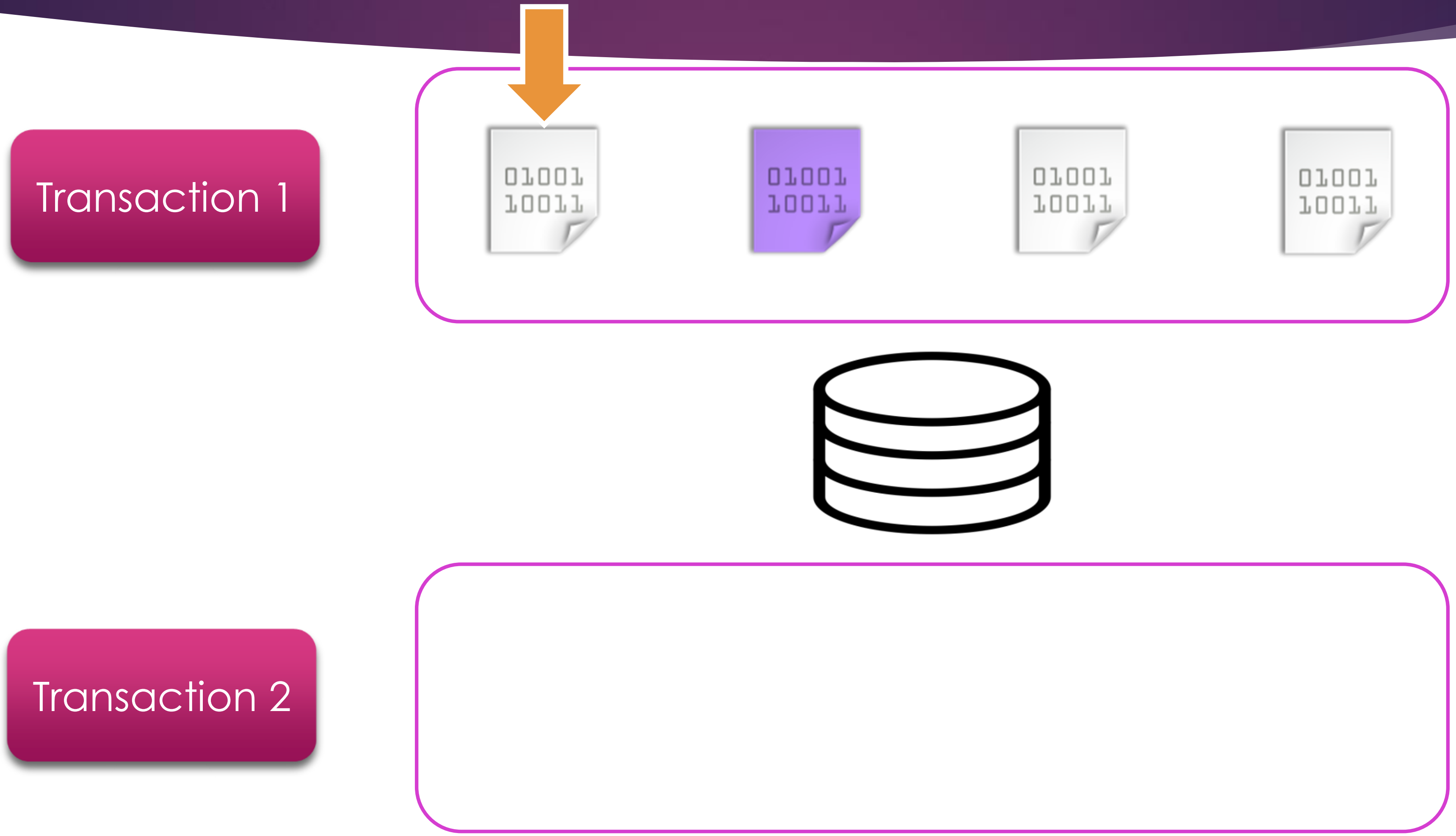
Missing or Double Reads

Transaction 1

Transaction 2



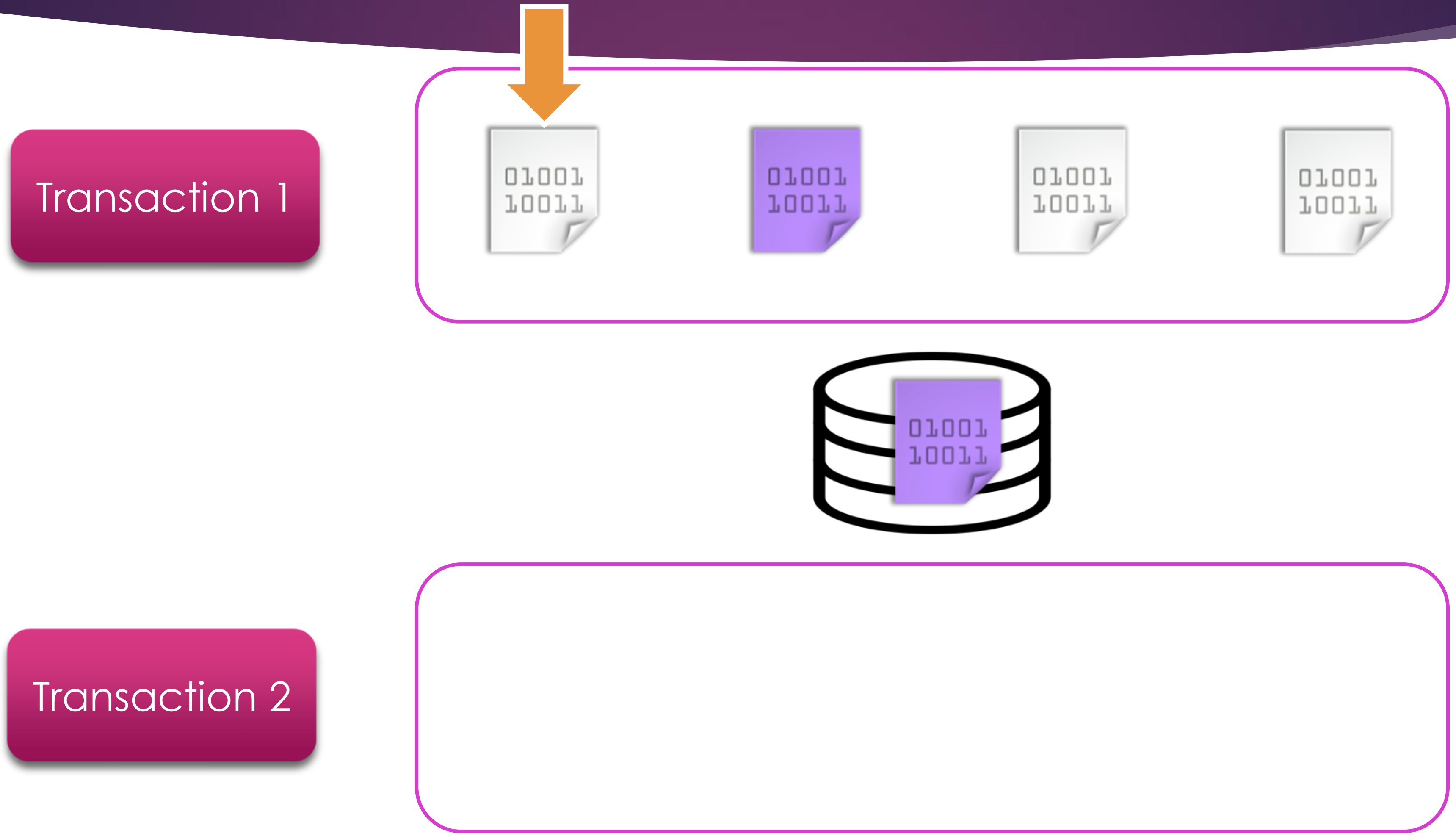
Missing or Double Reads



Transaction 1

Transaction 2

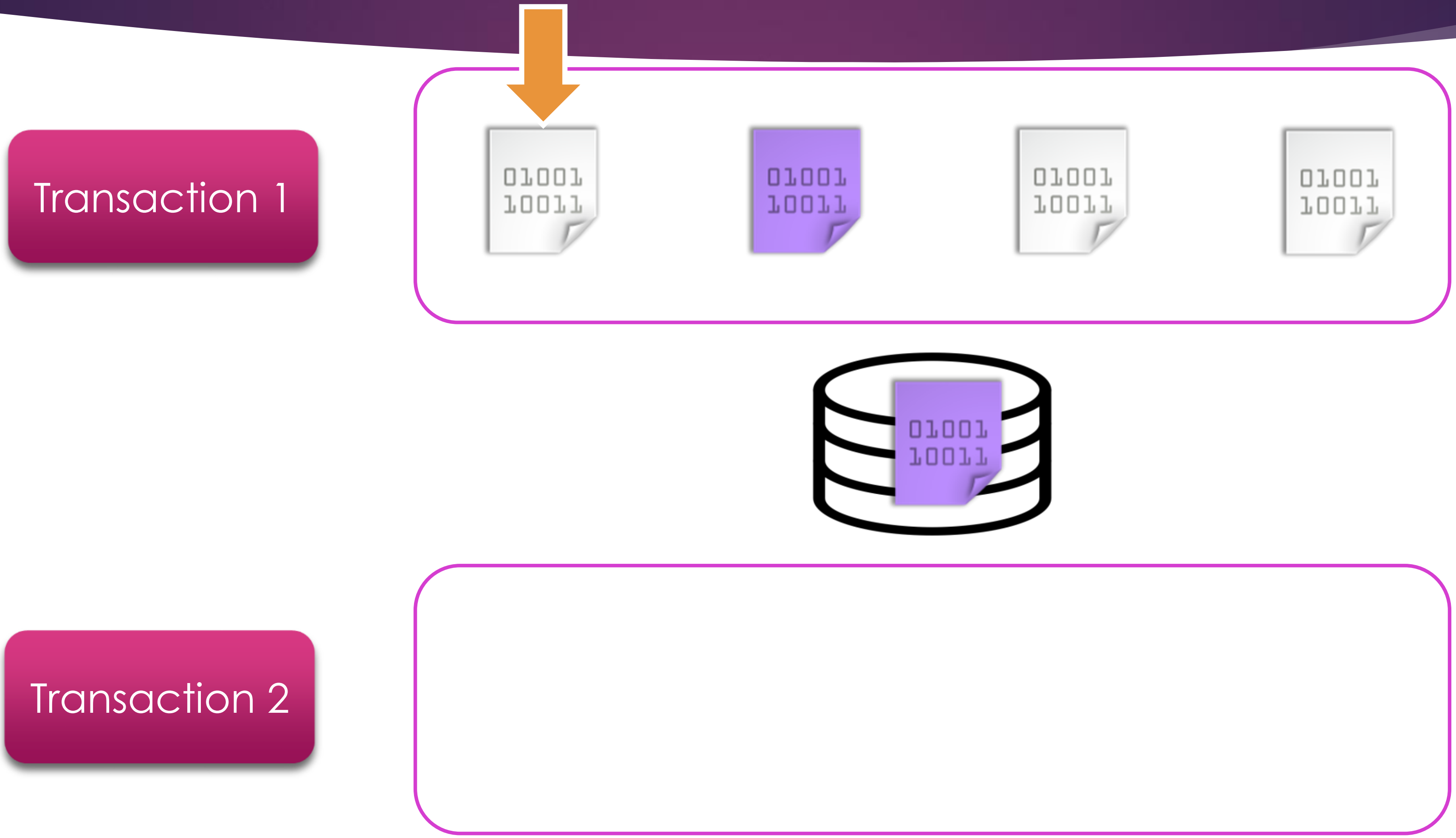
Missing or Double Reads



Transaction 1

Transaction 2

Missing or Double Reads

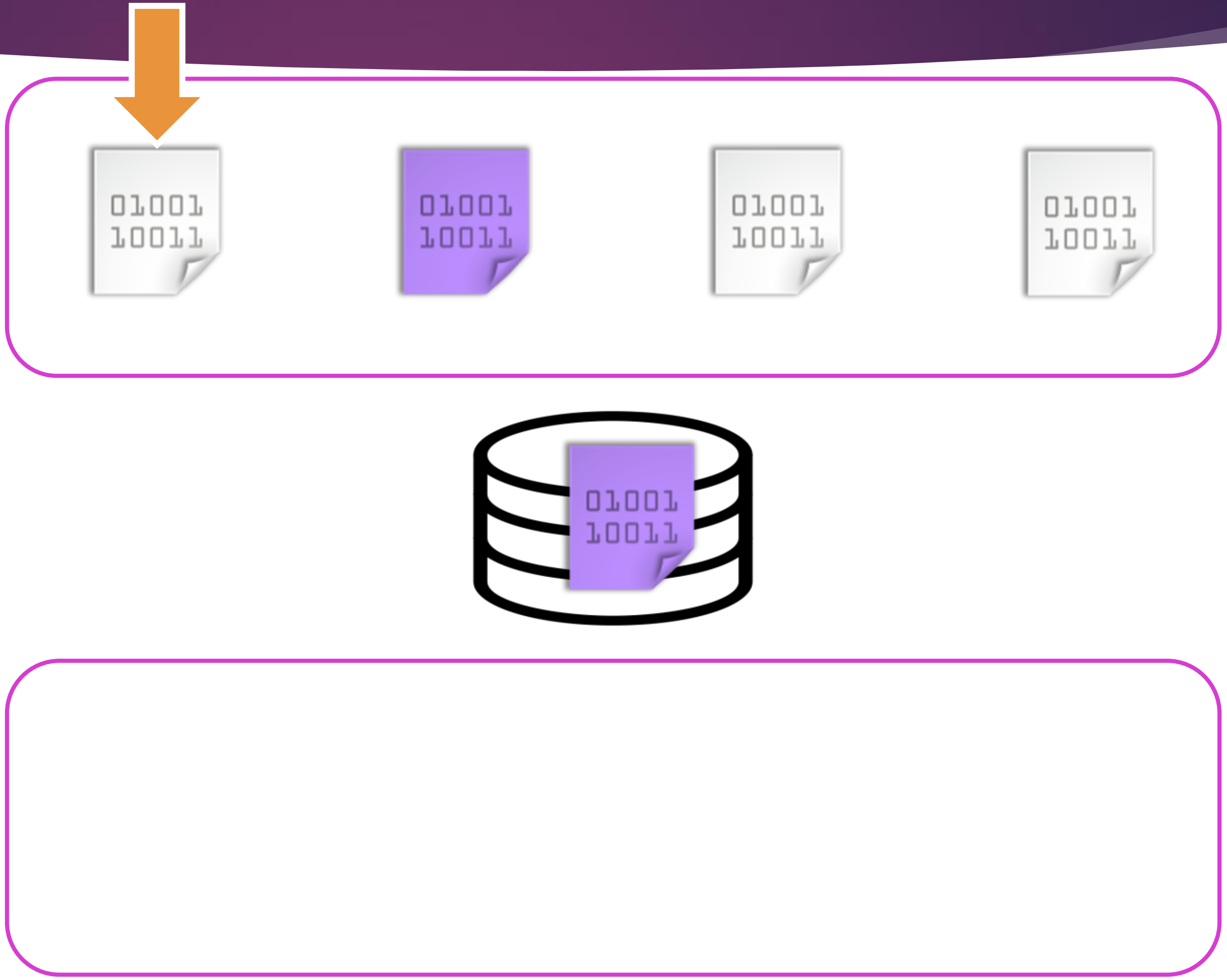


Transaction 1

Transaction 2

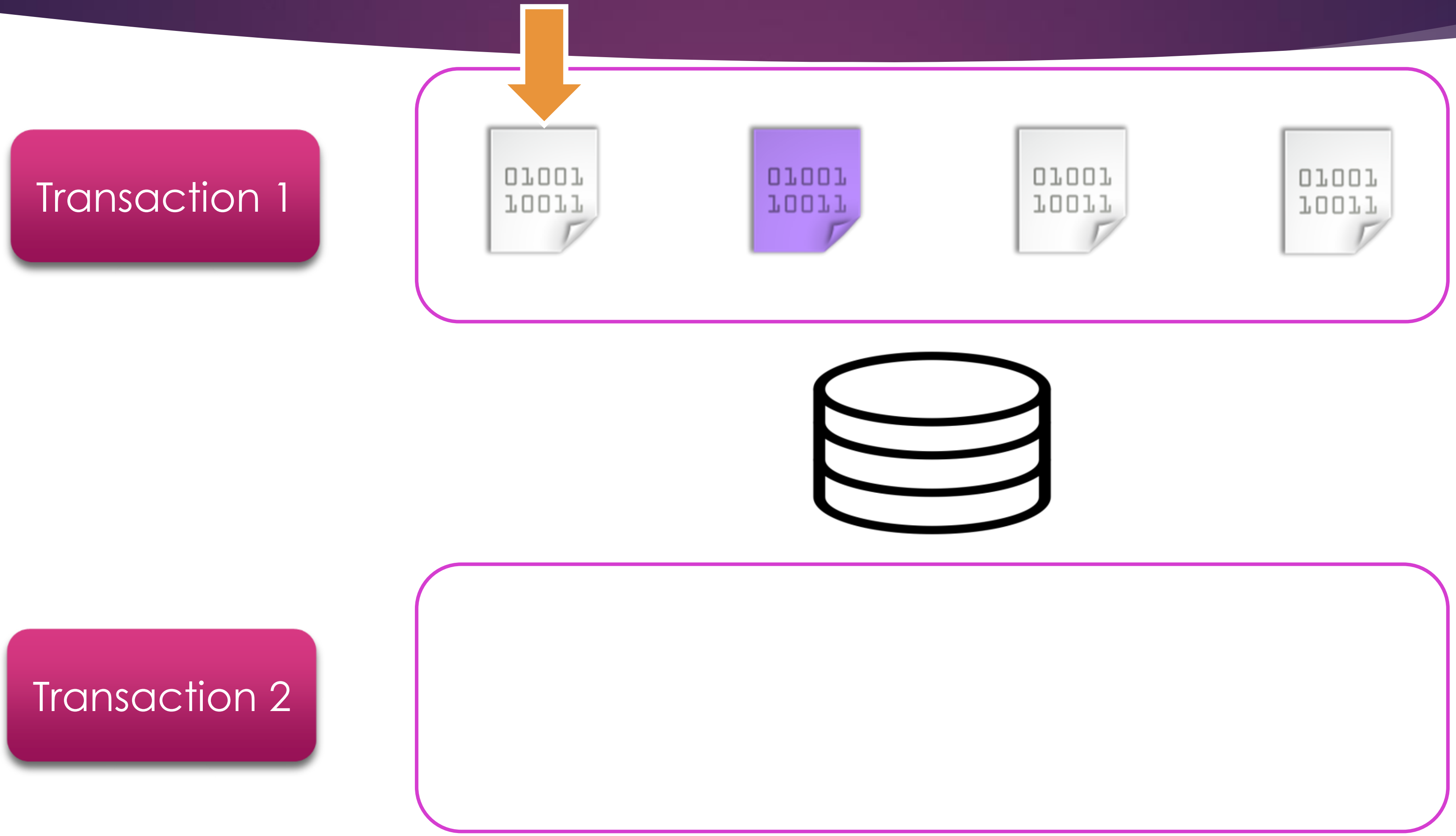
Missing or Double Reads

Transaction 1



Transaction 2

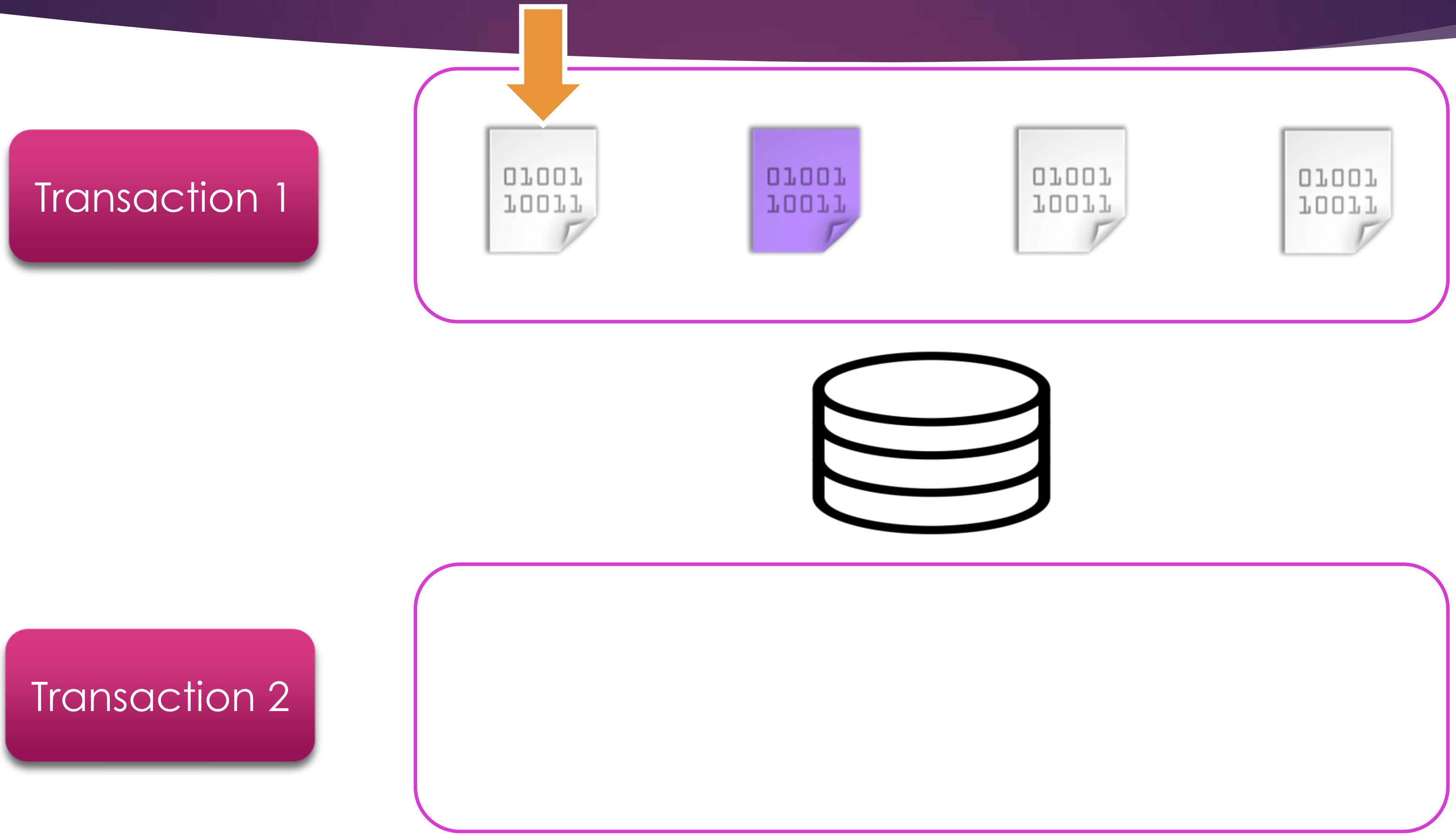
Missing or Double Reads



Transaction 1

Transaction 2

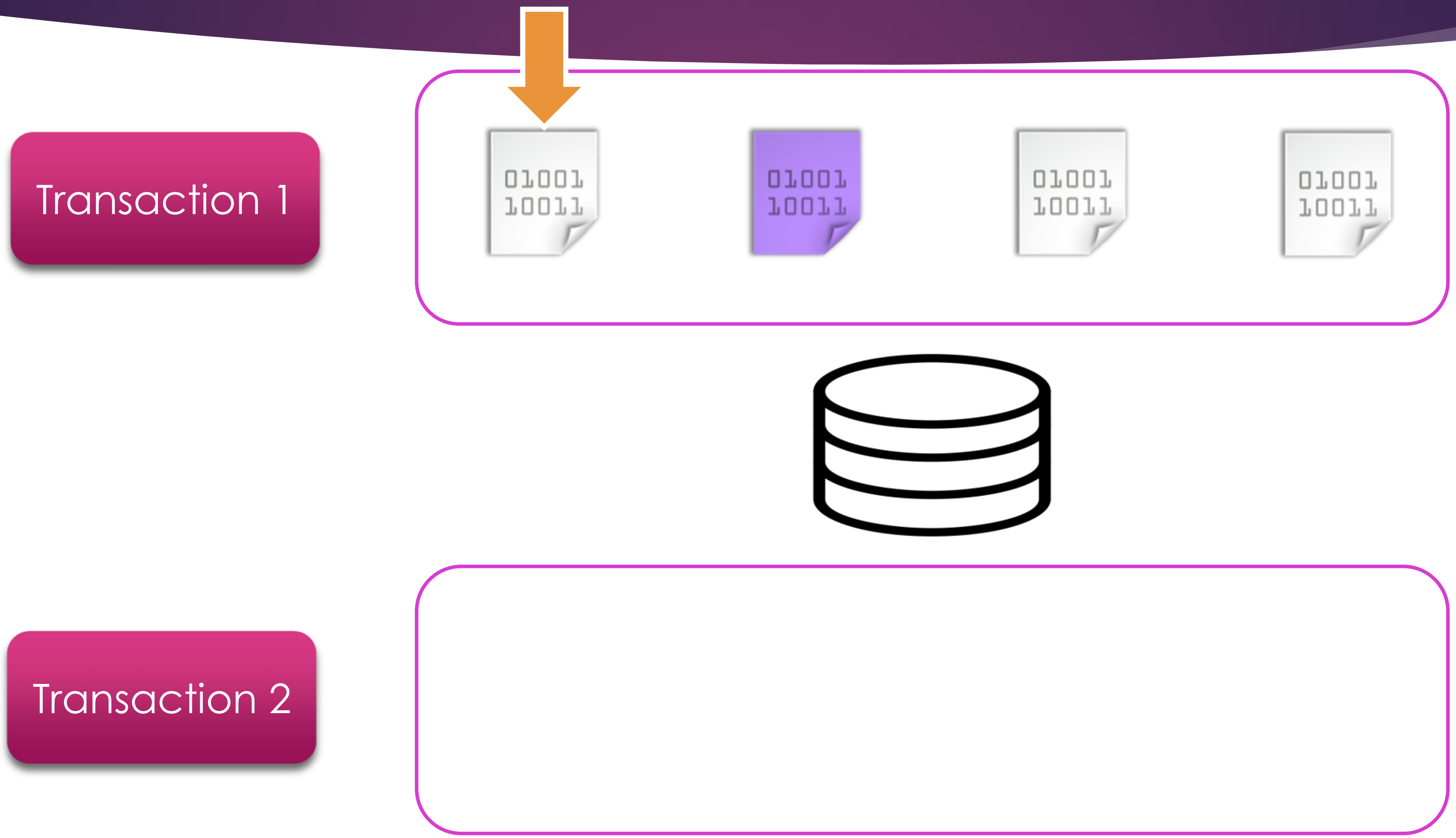
Missing or Double Reads



Transaction 1

Transaction 2

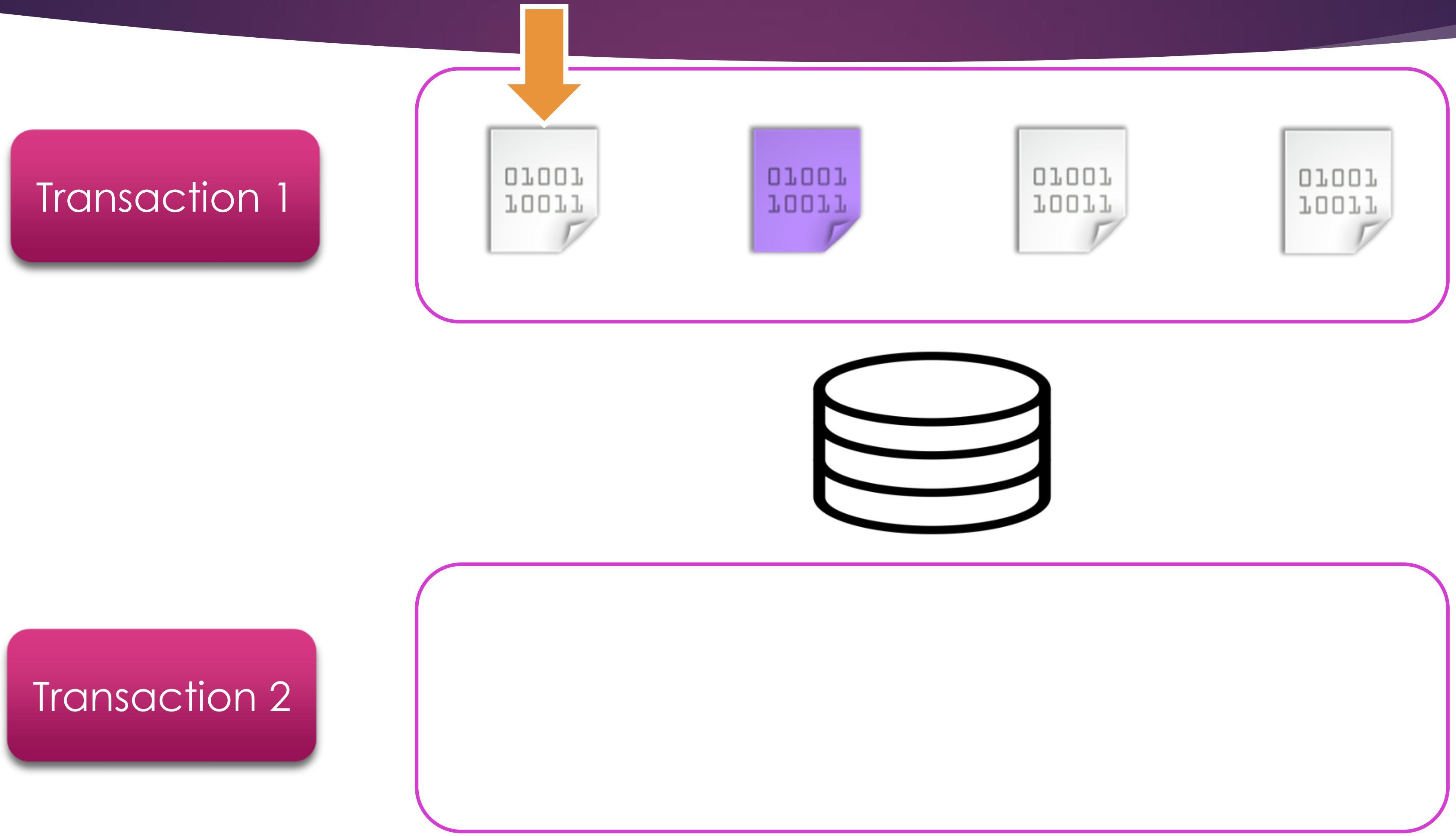
Missing or Double Reads



Transaction 1

Transaction 2

Missing or Double Reads



Transaction 1

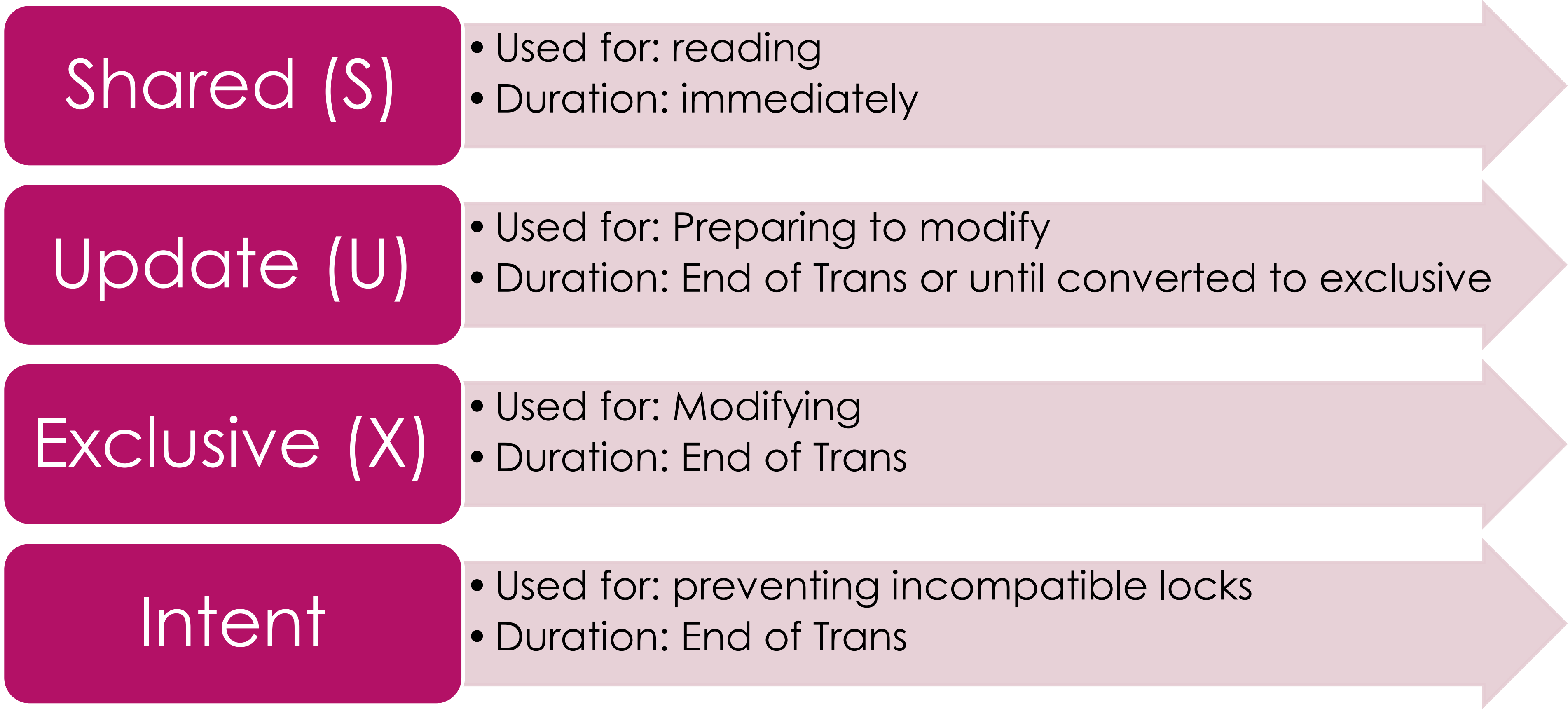
Transaction 2

Questions

- What is the Non-repeatable reads?
- What is the difference between Dirty reads and Phantom reads?

Database Isolation levels

Common Lock Types



Shared (S)

- Used for: reading
- Duration: immediately

Update (U)

- Used for: Preparing to modify
- Duration: End of Trans or until converted to exclusive

Exclusive (X)

- Used for: Modifying
- Duration: End of Trans

Intent

- Used for: preventing incompatible locks
- Duration: End of Trans

Isolation Levels (ANSI)

Read uncommitted

- ✓ Dirty Reads
- ✓ Non-repeatable
- ✓ Phantom Read

Read committed

- 🛇 Dirty Reads
- ✓ Non-repeatable
- ✓ Phantom Read

Repeatable read

- 🛇 Dirty Reads
- 🛇 Non-repeatable
- ✓ Phantom Read

Serializable

- 🛇 Dirty Reads
- 🛇 Non-repeatable
- 🛇 Phantom Read

Read uncommitted

- This is the lowest isolation level. In this level, dirty reads are allowed, so one transaction may see not-yet-committed changes made by other transactions.

Read committed

- In this isolation level, a lock-based concurrency control DBMS implementation keeps write locks (acquired on selected data) until the end of the transaction, but read locks are released as soon as the SELECT operation is performed (so the non-repeatable reads phenomenon can occur in this isolation level).
- As in the previous level, range-locks are not managed.

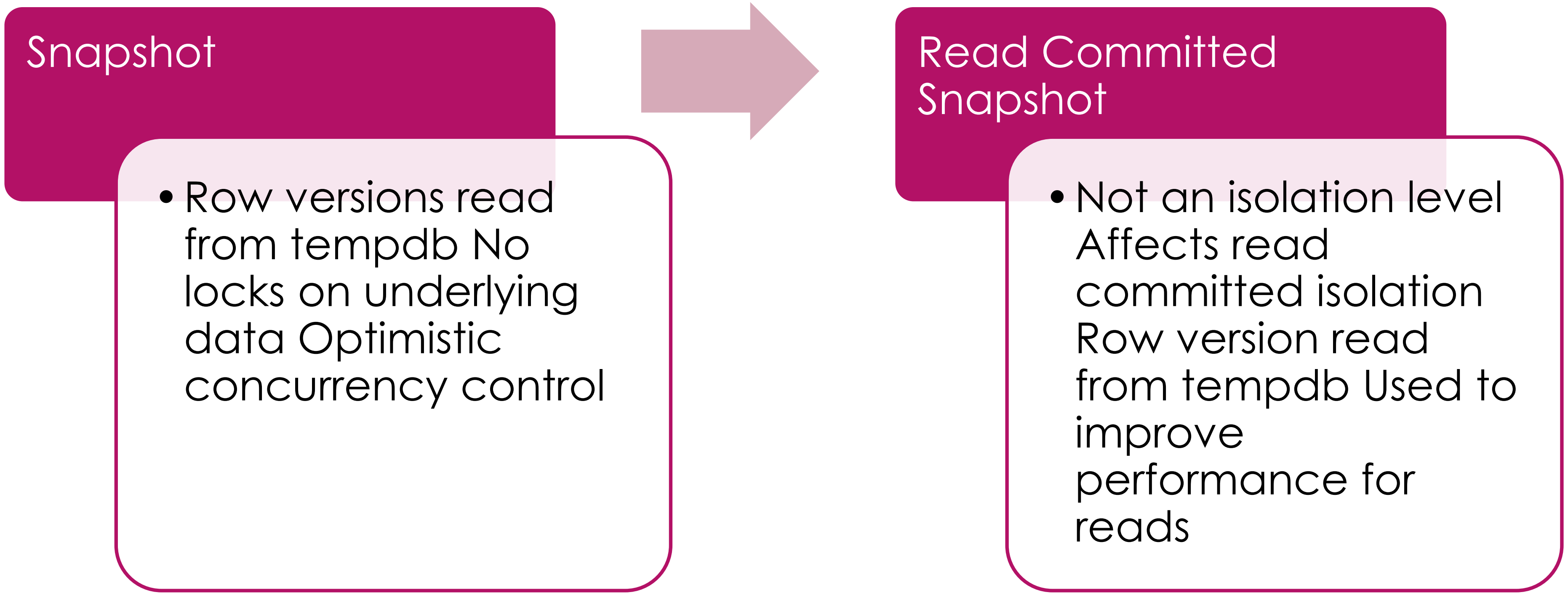
Repeatable reads

- In this isolation level, a lock-based concurrency control DBMS implementation keeps read and write locks (acquired on selected data) until the end of the transaction.
- However, range-locks are not managed, so phantom reads can occur.

Serializable

- This is the highest isolation level.
- With a lock-based concurrency control DBMS implementation, serializability requires read and write locks (acquired on selected data) to be released at the end of the transaction.
- Also range-locks must be acquired when a SELECT query uses a ranged WHERE clause, especially to avoid the phantom reads phenomenon.

Additional SQL Server Isolation Levels



Snapshot

- Row versions read from tempdb No locks on underlying data Optimistic concurrency control

Read Committed Snapshot

- Not an isolation level Affects read committed isolation Row version read from tempdb Used to improve performance for reads

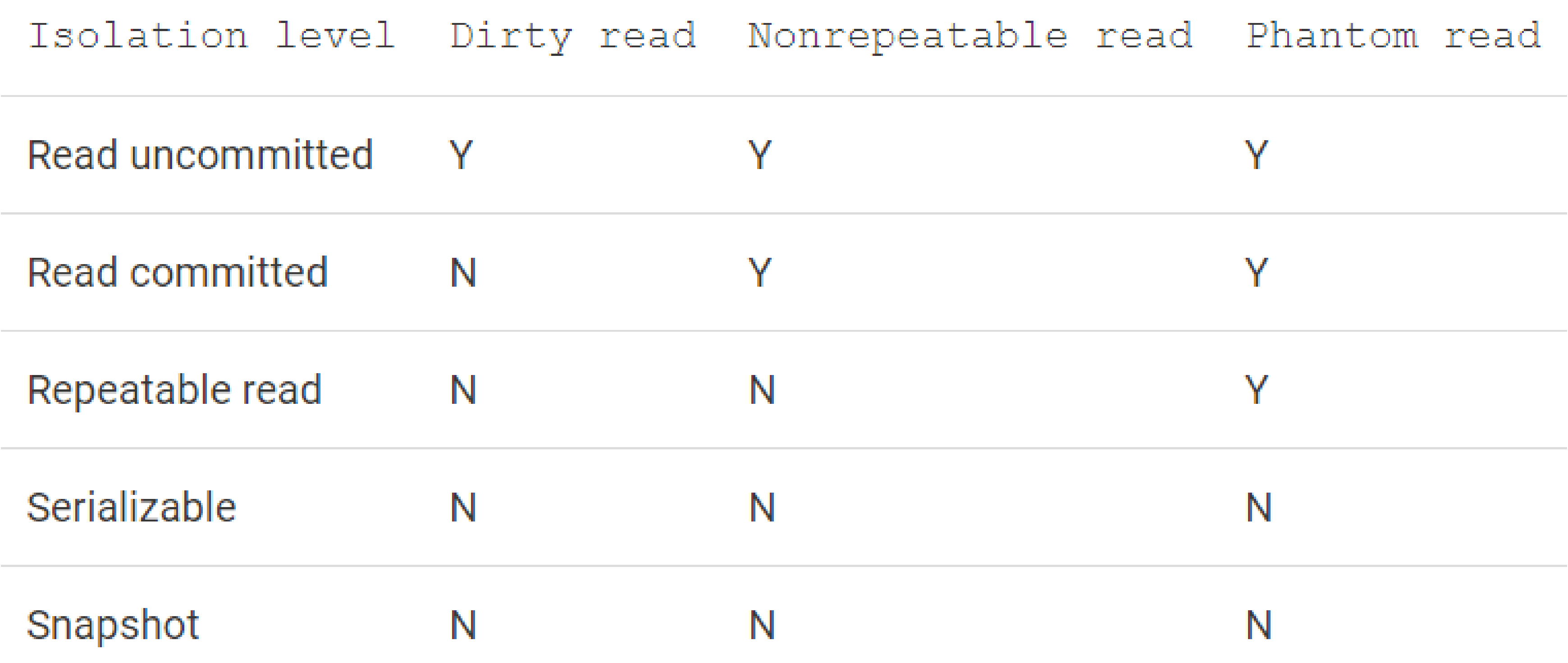
Snapshot Isolation (SI)

- This is new proprietary isolation level that provides non-blocking access for read operations.
- The transactions under SI see the snapshot of the database as of the beginning of the transaction.
- There is no automatic mapping of transaction isolation levels to SI so you must change your application to access data under SI isolation level.

Read-Committed-Snapshot (RCSI)

- This is not a new isolation level but a new implementation of read committed isolation level that does not take any S lock on the data.
- The word snapshot stems from the fact that query under RCSI sees the snapshot of the database as of the beginning of the statement.
- It is a better alternative for applications that must access only committed data but without taking locks.

The concurrency issues that each isolation level is susceptible to:



Isolation level	Dirty read	Nonrepeatable read	Phantom read
Read uncommitted	Y	Y	Y
Read committed	N	Y	Y
Repeatable read	N	N	Y
Serializable	N	N	N
Snapshot	N	N	N

Choosing an Isolation Levels

Lower Isolation Level

- Better concurrent access
- More change of concurrency effects

Higher Isolation Level

- Reduced change of concurrency effects
- Require more system resources
- Transactions may block waiting for locks to be released
- Deadlocks more likely

Questions

- What are the differences between READ-COMMITTED and REPEATABLE-READ transaction isolation levels?
- What’s the difference between using the NOLOCK table hint and the Read Uncommitted transaction level?