

Question 1:

You are managing a database for a hospital information system. The Patients table in the system has the following structure:

- PatientID: A unique identifier for each patient.
- Name: The name of the patient.
- Age: The age of the patient.
- MedicalCondition: The medical condition of the patient.
- RoomNumber: The room number assigned to the patient.
- Version: An integer representing the version of the patient record.
- LastUpdated: A timestamp indicating the last time the record was updated.

Tasks:

1. Two different healthcare staff members are trying to update the room number of a patient at the same time. One is updating it due to a room change for patient comfort, and another is updating it for medical isolation purposes. Implement MVCC to demonstrate how both versions of the room assignment can be maintained in the database.
2. Write SQL queries to simulate MVCC in this scenario and demonstrate how different versions of the patient's room assignment can exist in the database simultaneously.

Input Table:

PatientID	Name	Age	MedicalCondition	RoomNumber	Version	LastUpdated
1	John Doe	45	Hypertension	101	1	2023-09-26 12:00:00

Title for Question 1: Tracking Patient Room Changes with Versioned Records

Solution:

```
-- Insert the initial version of the patient record
INSERT INTO Patients (PatientID, Name, Age, MedicalCondition, RoomNumber,
VALUES (1, 'John Doe', 45, 'Hypertension', 101, 1, '2023-09-26 12:00:00')

-- Staff Member 1 starts updating the room number for patient comfort and
INSERT INTO Patients (PatientID, Name, Age, MedicalCondition, RoomNumber,
VALUES (1, 'John Doe', 45, 'Hypertension', 102, 2, CURRENT_TIMESTAMP);

-- Staff Member 2 also starts updating the room number for medical isolat
INSERT INTO Patients (PatientID, Name, Age, MedicalCondition, RoomNumber,
VALUES (1, 'John Doe', 45, 'Hypertension', 103, 3, CURRENT_TIMESTAMP);
```

TestCases:

S.No	Inputs	Outputs
------	--------	---------

1		PatientID   Name   Age   MedicalCondition   RoomNumber   Version   LastUpdated -----   -----   ---   -----   -----   -----   ----- 1   John Doe   45   Hypertension   101   1   2023-09-26 12:00:00 1   John Doe   45   Hypertension   102   2   2024-02-23 05:42:49 1   John Doe   45   Hypertension   103   3   2024-02-23 05:42:49
2		
3		
4		
5		
6		

White List:

Black List:

Question 2:

You are managing a database for an e-commerce website. The Products table in the system has the following structure:

- ProductID: A unique identifier for each product.
- ProductName: The name of the product.
- Price: The price of the product.
- Stock: The number of available items in stock.
- Version: An integer representing the version of the product record.
- LastUpdated: A timestamp indicating the last time the record was updated.

Tasks:

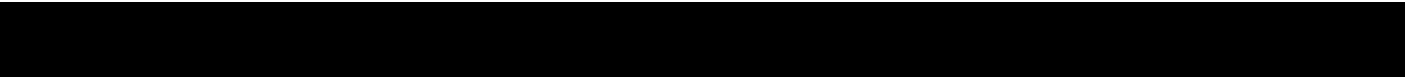
1. Two different employees are trying to update the stock of a product at the same time. One is updating the stock due to new items being added to the inventory, and another is updating it due to a customer placing an order. Implement MVCC to demonstrate how both versions of the product stock can be maintained in the database.
2. Write SQL queries to simulate MVCC in this scenario and demonstrate how different versions of the product stock can exist in the database simultaneously.

Input Table:

ProductID	ProductName	Price	Stock	Version	LastUpdated
1	Wireless Mouse	25	100	1	2023-09-26 12:00:00

Title for Question 2: Tracking Product Stock Updates with Versioned Records

Solution:



TestCases:

S.No	Inputs	Outputs
1		ProductID   ProductName   Price   Stock   Version   LastUpdated -----   -----   ----   ----   -----   ----- 1   Wireless Mouse   25.00   100   1   2023-09-26 12:00:00 1   Wireless Mouse   25.00   120   2   2024-02-23 05:47:49 1   Wireless Mouse   25.00   119   3   2024-02-23 05:47:49
2		
3		
4		
5		
6		

White List:

Black List:

Question 3:

You are managing a database for an online grocery store. The Items table in the system has the following structure:

- ItemID: A unique identifier for each item.
- ItemName: The name of the item.
- Price: The price of the item.
- QuantityAvailable: The quantity of the item currently available.
- Version: An integer representing the version of the item record.
- LastUpdated: A timestamp indicating the last time the record was updated.

Tasks:

1. Two different staff members are trying to update the quantity available of an item at the same time. One is updating the quantity available due to a new delivery, and another is updating it due to an item being sold. Implement MVCC to demonstrate how both versions of the item availability can be maintained in the database.
2. Write SQL queries to simulate MVCC in this scenario and demonstrate how different versions of the item availability can exist in the database simultaneously.

Input Table:

ItemID	ItemName	Price	QuantityAvailable	Version	LastUpdated
1	Organic Apples	1.5	300	1	2023-09-26 12:00:00

Title for Question 3: Simulating MVCC for Concurrent Updates in an Online Grocery Store

Database

Solution:

```
-- Insert the initial version of the item record
INSERT INTO Items (ItemID, ItemName, Price, QuantityAvailable, Version, LastUpdated)
VALUES (1, 'Organic Apples', 1.50, 100, 1, '2023-09-26 12:00:00');

-- Staff Member 1 starts updating the QuantityAvailable due to a new delivery
INSERT INTO Items (ItemID, ItemName, Price, QuantityAvailable, Version, LastUpdated)
VALUES (1, 'Organic Apples', 1.50, 120, 2, CURRENT_TIMESTAMP);

-- Staff Member 2 also starts updating the QuantityAvailable due to an item price change
INSERT INTO Items (ItemID, ItemName, Price, QuantityAvailable, Version, LastUpdated)
VALUES (1, 'Organic Apples', 1.50, 119, 3, CURRENT_TIMESTAMP);
```

TestCases:

S.No	Inputs	Outputs
1		ItemID   ItemName   Price   QuantityAvailable   Version   LastUpdated -----   -----   ----   -----   -----   ----- 1   Organic Apples   1.50   100   1   2023-09-26 12:00:00 1   Organic Apples   1.50   120   2   2024-02-23 05:52:25 1   Organic Apples   1.50   119   3   2024-02-23 05:52:25
2		
3		
4		
5		
6		

White List:

Black List:

Question 4:

You are managing a database for a flight booking system. The system has a Flights table with the following structure:

- FlightID: An integer representing the unique identifier for each flight.
- FlightNumber: A string representing the flight number.
- SeatsAvailable: An integer representing the number of available seats.
- LastUpdated: A timestamp indicating the last time the record was updated.

Tasks:

1. Two different agents are trying to update the SeatsAvailable for a flight at the same time, one for booking and another for cancellation. Simulate a Timestamp-based Scheduler to ensure that the

- transaction with the older timestamp gets committed first.
2. Write SQL queries to simulate this scenario and demonstrate how Timestamp-based Scheduler ensures that the transaction with the older timestamp gets committed first.

Input Tabl:

FlightID	FlightNumber	SeatsAvailable	LastUpdated
1	AI101	50	2024-02-22 07:00:00

Title for Question 4: Flight Seat Availability Management and Data Retrieval in a Booking System

Solution:

```
-- Start the first transaction (Agent 1 - Booking)
START TRANSACTION;

-- Update the SeatsAvailable for Flight AI101 by Agent 1
UPDATE Flights
SET SeatsAvailable = SeatsAvailable - 1, LastUpdated = CURRENT_TIMESTAMP
WHERE FlightID = 1 AND SeatsAvailable > 0;

-- Commit Transaction 1

-- Start the second transaction (Agent 2 - Cancellation)
START TRANSACTION;

-- Attempt to update the SeatsAvailable for Flight AI101 by Agent 2
-- This transaction should check the LastUpdated timestamp before updating
UPDATE Flights
SET SeatsAvailable = SeatsAvailable + 1, LastUpdated = CURRENT_TIMESTAMP
WHERE FlightID = 1
  AND (LastUpdated < CURRENT_TIMESTAMP OR LastUpdated IS NULL);

-- End the second transaction

COMMIT; -- Commit Transaction 2
```

TestCases:

S.No	Inputs	Outputs
1		FlightID   FlightNumber   SeatsAvailable   LastUpdated -----   -----   -----   ----- 1   AI101   49   2024-02-22 06:18:34
2		
3		
4		
5		
6		

White List:

## Black List:

---

### Question 5:

You are managing a database for an online voting system. The system has a Candidates table with the following structure:

- CandidateID: An integer representing the unique identifier for each candidate.
- CandidateName: A string representing the name of the candidate.
- Votes: An integer representing the number of votes a candidate has received.
- LastUpdated: A timestamp indicating the last time the record was updated.

### Tasks:

1. Two different volunteers are trying to update the Votes of a candidate at the same time after receiving new ballot counts. Simulate a Timestamp-based Scheduler to ensure that the transaction with the older timestamp gets committed first.
2. Write SQL queries to simulate this scenario and demonstrate how Timestamp-based Scheduler ensures that the transaction with the older timestamp gets committed first.

### Input Table:

CandidateID	CandidateName	Votes	LastUpdated
1	Jane Doe	500	2023-09-25 10:00:00

**Title for Question 5:** Transaction Handling and Data Retrieval in a Candidate Vote Count System

### Solution:

```
-- Start the first transaction (Volunteer 1)
START TRANSACTION;
-- Update the Votes of Jane Doe by Volunteer 1
UPDATE Candidates
SET Votes = Votes + 100, LastUpdated = CURRENT_TIMESTAMP
WHERE CandidateID = 1;
-- Commit Transaction 1
-- Start the second transaction (Volunteer 2)
-- Attempt to update the Votes of Jane Doe by Volunteer 2
-- This transaction should check the LastUpdated timestamp before updating
UPDATE Candidates
SET Votes = Votes + 200, LastUpdated = CURRENT_TIMESTAMP
WHERE CandidateID = 1;
--   AND LastUpdated < CURRENT_TIMESTAMP;
-- Retrieve the updated data for CandidateID 1
-- End the second transaction
COMMIT;
```

TestCases:

S.No	Inputs	Outputs
1		CandidateID   CandidateName   Votes   LastUpdated -----   -----   ----   ----- 1   Jane Doe   1300   2024-02-23 04:46:36
2		
3		
4		
5		
6		

White List:

Black List:

Question 6:

You are managing a database for a hospital. The hospital has a Patients table with the following structure:

- PatientID: An integer representing the unique identifier for each patient.
- PatientName: A string representing the name of the patient.
- Age: An integer representing the age of the patient.
- AdmissionDate: A date indicating when the patient was admitted.
- LastUpdated: A timestamp indicating the last time the record was updated.

Tasks:

1. Two different nurses are trying to update the AdmissionDate of a patient at the same time due to a system error. Simulate a Timestamp-based Scheduler to ensure that the transaction with the older timestamp gets committed first.
2. Write SQL queries to simulate this scenario and demonstrate how Timestamp-based Scheduler ensures that the transaction with the older timestamp gets committed first.

Input Table:

PatientID	PatientName	Age	AdmissionDate	LastUpdated
1	John Doe	30	2023-09-20	2023-09-25 15:00:00

Title for Question 6: Patient Admission Date Update with Transaction Handling in a Healthcare Database

Solution:

```
-- Check if a record with PatientID 1 already exists, and if not, insert
INSERT IGNORE INTO Patients (PatientID, PatientName, Age, AdmissionDate,
VALUES (1, 'John Doe', 30, '2023-09-20', '2023-09-25 15:00:00');
```

```
-- Start the first transaction (Nurse 1)
START TRANSACTION;

-- Update the AdmissionDate of John Doe by Nurse 1
UPDATE Patients
SET AdmissionDate = '2023-09-21', LastUpdated = CURRENT_TIMESTAMP
WHERE PatientID = 1;

-- Commit Transaction 1
COMMIT;

-- Start the second transaction (Nurse 2)
START TRANSACTION;

-- Attempt to update the AdmissionDate of John Doe by Nurse 2
UPDATE Patients
SET AdmissionDate = '2023-09-22', LastUpdated = CURRENT_TIMESTAMP
WHERE PatientID = 1;

-- Check if any rows were updated in the second transaction
-- If no rows were updated, it means another transaction modified the data

-- End the second transaction
COMMIT;
```

TestCases:

S.No	Inputs	Outputs
1		PatientID   PatientName   Age   AdmissionDate   LastUpdated -----   -----   ---   -----   ----- 1   John Doe   30   2023-09-22   2024-02-23 04:13:05
2		
3		
4		
5		
6		

White List:

Black List:

---