Answer to the Questions

Question:

We want to know what state the database was in at a specific date in the past (including what records it contained). Therefore, we need to:

- Insert Operation: For each record inserted, we must know exactly when it was added to the data warehouse. We can add a column to each table for this purpose (the record with this column is considered equal to the corresponding record in the source database that does not have this column).
- Update or Delete Operation: For each record updated or deleted in the source database, this record must be preserved in the data warehouse but transferred to another table that is considered a history table. It must be clear when the record was removed from the main table and why (deleted from the main table or updated).

These two conditions are sufficient to travel back to the past. Explain how you can travel back in time in the data warehouse using these two conditions.

Answer:

To maintain the historical state of the database and travel back to a specific point in time, we need to implement two primary attributes:

1. Insert Timestamp: This attribute records the exact time when a record is inserted into the data warehouse. By adding a column for this timestamp in each table, we can identify when each record was added.
2. Update/Delete History: For each record that is updated or deleted in the source database, it must be preserved in the data warehouse and transferred to a history table. This table will contain the timestamp indicating when the record was removed from the main table and the reason (whether it was deleted or updated).

By implementing these two conditions, we can accurately travel back in time to view the state of the database at any given point.

Steps to Implement

1. Insert Operation:

- When a record is inserted into the source database, we record the insertion time in a specific column in the data warehouse.

- Example:

```sql
INSERT INTO warehouse_table (column1, column2, ..., insert_timestamp)
VALUES (value1, value2, ..., CURRENT_TIMESTAMP);
```

2. Update Operation:

- When a record is updated in the source database, the corresponding record in the data warehouse is moved to a history table with a timestamp indicating the update time.

- Example:

```sql
-- Move to history table
INSERT INTO warehouse_history (column1, column2, ..., update_timestamp)
SELECT column1, column2, ..., CURRENT_TIMESTAMP
FROM warehouse_table
WHERE primary_key = value;

-- Update the main table
UPDATE warehouse_table
```

SET column1 = new_value1, column2 = new_value2, ...

WHERE primary_key = value;

```

**3. Delete Operation:**

- When a record is deleted from the source database, the corresponding record in the data warehouse is moved to a history table with a timestamp indicating the deletion time.

- Example:

```sql

-- Move to history table

INSERT INTO warehouse_history (column1, column2, ..., delete_timestamp)

SELECT column1, column2, ..., CURRENT_TIMESTAMP

FROM warehouse_table

WHERE primary_key = value;

-- Delete from the main table

DELETE FROM warehouse_table

WHERE primary_key = value;

```

By following these steps, we can ensure that every change (insert, update, delete) in the source database is accurately reflected in the data warehouse, and we can travel back to any point in time to see the state of the database.
```