Task 1: In the figures below, please check whether the implementation of the Conversation (User1 raising the salary of the Guide[id=2] to 3000) is described truly.

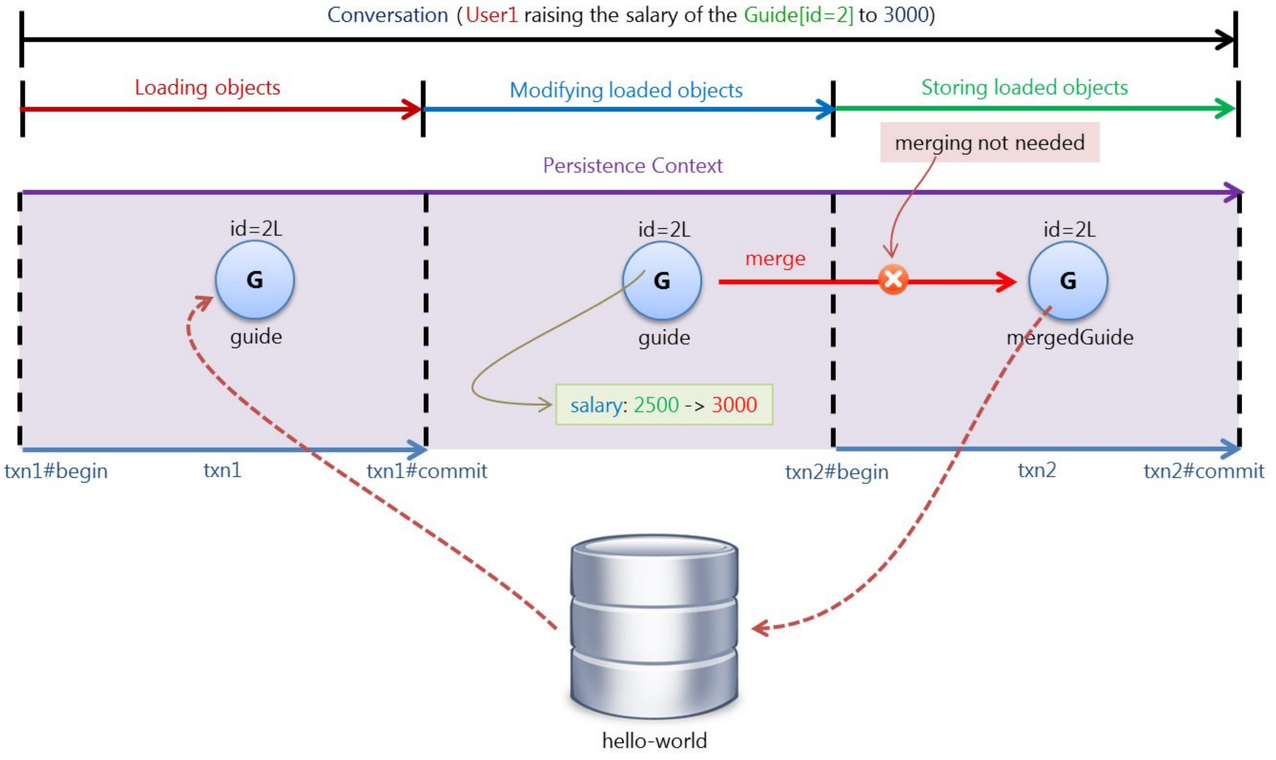

Figure 1.1 Conversation implementation with an extended persistence context - TRUE OR FALSE?

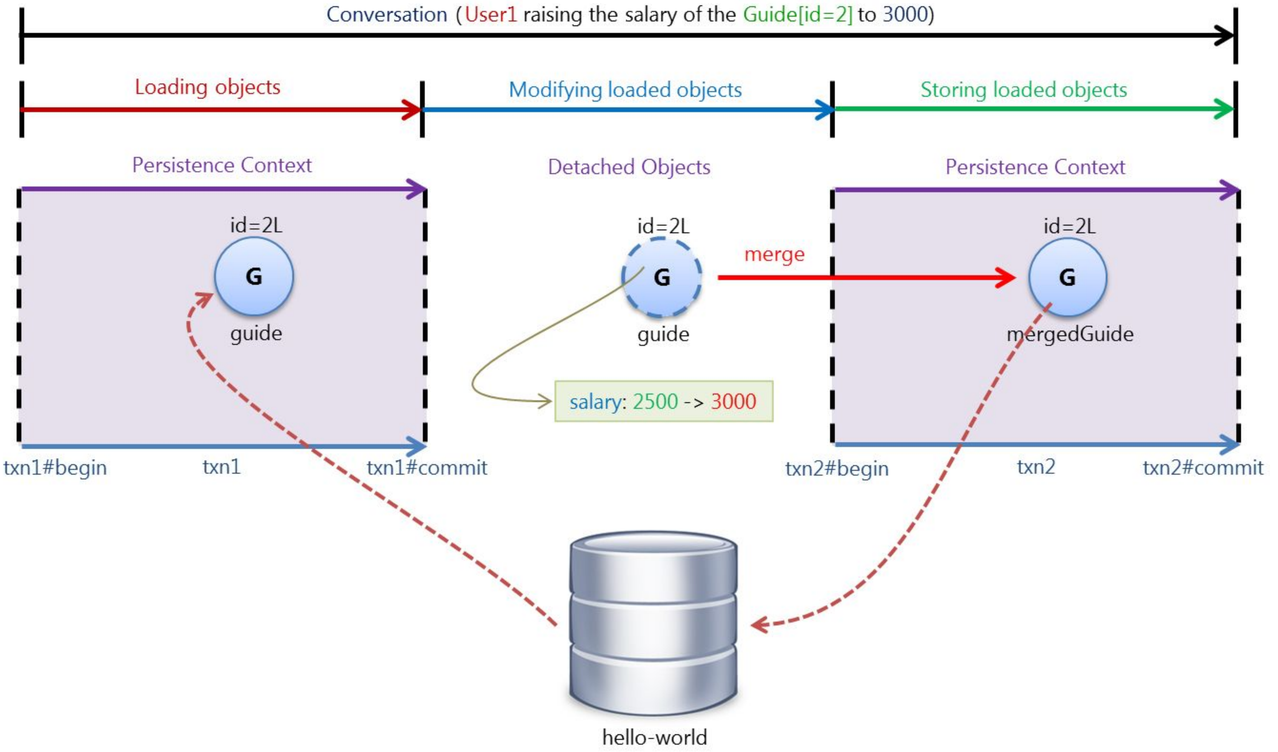

Figure 1.2 Conversation implementation with detached object state - TRUE OR FALSE?

Task 2: As explain in the lecture on Optimistic Locking and Versioning, simulate a Lost Update situation while implementing the concurrent conversations of two users, User1 and User2, where User1 wants to modify the salary of the Guide[id=2] to 3000, whereas User2 wants to modify the salary of the same Guide to 4000.

Consider the following to be the starting state of the guide table:

guide

id	name	salary	staff_id
1	Mike Lawson	1000	2000MO10789
2	Ian Lamb	2500	2000IM10901
3	David Crow	3000	2000DO10777

Task 3: While simulating the Lost Update situation in Task 2, please check whether the last transaction commit won (usually referred as Last Commit Wins) or not, as it was explained in the lecture?

Task 4: Solve the Lost Update situation while performing the Task 2 using the Versioning strategy (Optimistic Locking).

Hint: Add the version column in the guide table using the following SQL:

```
ALTER TABLE `guide`
ADD `version` INT(11) NOT NULL DEFAULT '0';
```

***The source code files for the lecture on "Optimistic Locking and Versioning" are available to be downloaded with this lab exercise. You could use them to complete the given tasks successfully.