Use the transaction log in Table 10.16 to trace a simple database recovery process. To make sure you understand the recovery process, the simple transaction log includes three transactions and one checkpoint. This transaction log includes the transaction components used earlier in the chapter, so you should already be familiar with the basic process. Given the transaction, the transaction log has the following characteristics:

- Transaction 101 consists of two UPDATE statements that reduce the quantity on hand for product 54778-2T and increase the customer balance for customer 10011 for a credit sale of two units of product 54778-2T.
- Transaction 106 is the same credit sales event you saw in Section 10-1a. This transaction represents the credit sale of one unit of product 89-WRE-Q to customer 10016 for $277.55. This transaction consists of five SQL DML statements: three INSERT statements and two UPDATE statements.
- Transaction 155 represents a simple inventory update. This transaction consists of one UPDATE statement that increases the quantity on hand of product 2232/QWE from 6 units to 26 units.
- A database checkpoint writes all updated database buffers to disk. The checkpoint event writes only the changes for all previously committed transactions. In this case, the checkpoint applies all changes made by transaction 101 to the database data files.

Using Table 10.16, you can now trace the database recovery process for a DBMS using the deferred update method as follows:

1. Identify the last checkpoint—in this case, TRL ID 423. This was the last time database buffers were physically written to disk.
2. Note that transaction 101 started and finished before the last checkpoint. Therefore, all changes were already written to disk, and no additional action needs to be taken.
3. For each transaction committed after the last checkpoint (TRL ID 423), the DBMS will use the transaction log data to write the changes to disk, using the "after" values. For example, for transaction 106:
   a. Find COMMIT (TRL ID 457).
   b. Use the previous pointer values to locate the start of the transaction (TRL ID 397).
   c. Use the next pointer values to locate each DML statement, and apply the changes to disk using the "after" values. (Start with TRL ID 405, then 415, 419, 427, and 431.) Remember that TRL ID 457 was the COMMIT statement for this transaction.
   d. Repeat the process for transaction 155.
4. Any other transactions will be ignored. Therefore, for transactions that ended with ROLLBACK or that were left active (those that do not end with a COMMIT or ROLLBACK), nothing is done because no changes were written to disk.

TABLE 10.16

A TRANSACTION LOG FOR TRANSACTION RECOVERY EXAMPLES

TRL ID	TRX NUM	PREV PTR	NEXT PTR	OPERATION	TABLE	ROW ID	ATTRIBUTE	BEFORE VALUE	AFTER VALUE
341	101	Null	352	START	****Start Transaction				
352	101	341	363	UPDATE	PRODUCT	54778-2T	PROD_QOH	45	43
363	101	352	365	UPDATE	CUSTOMER	10011	CUST_BALANCE	615.73	675.62
365	101	363	Null	COMMIT	**** End of Transaction				
397	106	Null	405	START	****Start Transaction				
405	106	397	415	INSERT	INVOICE	1009			1009,10016, …
415	106	405	419	INSERT	LINE	1009,1			1009,1, 89-WRE-Q,1, .
419	106	415	427	UPDATE	PRODUCT	89-WRE-Q	PROD_QOH	12	11
423				CHECKPOINT					
427	106	419	431	UPDATE	CUSTOMER	10016	CUST_BALANCE	0.00	277.55
431	106	427	457	INSERT	ACCT_TRANSACTION	10007			1007,18-JAN-2016, …
457	106	431	Null	COMMIT	**** End of Transaction				
521	155	Null	525	START	****Start Transaction				
525	155	521	528	UPDATE	PRODUCT	2232/QWE	PROD_QOH	6	26
528	155	525	Null	COMMIT	**** End of Transaction				
* * * * * C *R*A* S* H * * * *									