MODULE 2

ADVANCED DATABASE CONCEPTS

“Transaction Management and Concurrency Control”

Catanduanes State University
College of Information and Communications Technology
ITP731-Advanced Database Systems

PRE-TEST

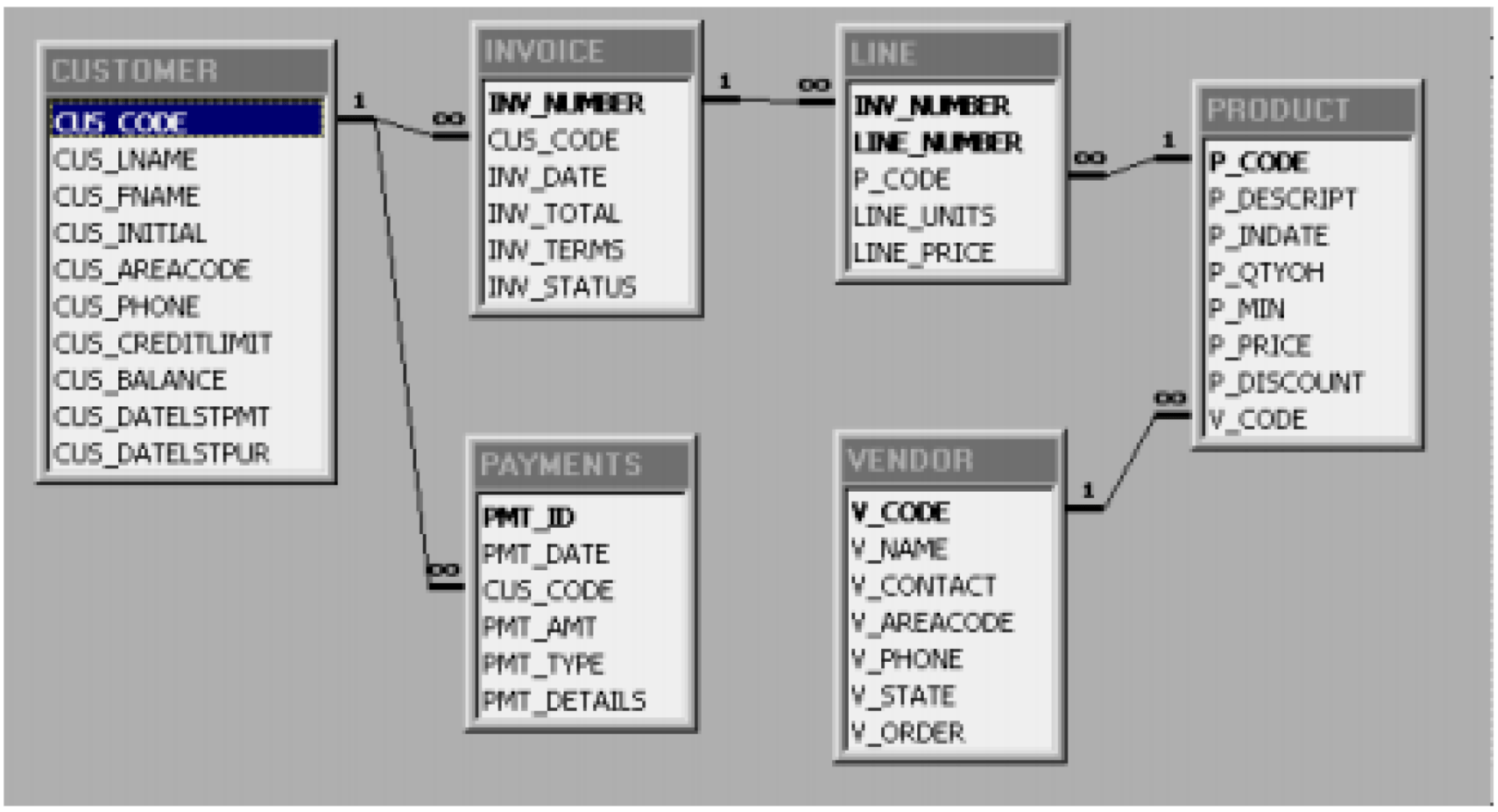
Name: _____ Student No. _____ Score: ____

Case Analysis: Discuss the following statement given below:

1. What is transaction log, and what is its function? 5 pts

2. What is concurrency control, and what is its objective? 5 pts

3. Using this database, write the SQL code to represent each one of the following transactions. Use BEGIN TRANSACTION and COMMIT to group the SQL statements in logical transactions. 10pts



Problem:

On May 11, 2006, customer '10010' makes a credit purchase (30 days) of one unit of product '11QER/31' with a unit price of $110.00; the tax rate is 8 percent. The invoice number is 10983, and this invoice has only one product line.

MODULE MAP

Core Coverage

- Database System
- Data Models
- Introduction to Structures Query Language (SQL)

Database Design and Implementation Focus

- Advanced Data Modeling
- Advanced SQL
- Database Design

Database Management Focus

- Database Performance Tuning and Query Optimization
- Business Intelligence and Data Warehouse
- Object Oriented Database

Supplementary Reading

DEFINITION OF KEY TERMS

Transaction	A sequence of database requests that accesses the database
Consistent Database State	A database state in which all data integrity constraints are satisfied
Database request	The equivalent of a single SQL statement in an application program or a transaction
Consistency	A database condition in which all data integrity constraints are satisfied.
Transaction Log	A feature used by the DBMS to keep track of all transactions that update the database.
Concurrency control	A DBMS feature that coordinates the simultaneous execution of transactions in a multiprocessing database system while preserving data integrity.
Atomicity	The transaction property that requires all parts of a transaction to be treated as a single, indivisible, logical unit of work.
Concurrency control	A DBMS feature that coordinates the simultaneous execution of transactions in multi-processing database system while preserving data integrity.
Lost update	A concurrency control problem in which a data update is lost during the concurrent execution of transactions.
Inconsistent retrievals	A concurrency control problem that arises when a transaction calculating summary functions over a set of data while other transactions are updating the data, yielding erroneous results.
Isolation	A database transaction property in which a data item used one transaction is not available to other transactions until the first one ends.
Durability	The transaction property that ensures that once transaction changes are done and committed, they cannot be undone or lost, even in the event of a system failure.
Uncommitted Data	A concurrency control problem in which a transaction accesses uncommitted data from another transaction

Lesson 2

B. Advanced Database Concept

Transaction Management and Concurrency Control

A. What is Transaction

a. Evaluating Transaction Result
b. Transaction Properties
c. Transactions and Schedules

B. Concurrency Control for database management

C. Database Recovery Management

ADVANCED DATABASE CONCEPT:

Transaction Management and Concurrency Control

Database transactions reflect real-world transactions that are triggered by events such as buying a product, registering for a course, or making a deposit into a checking account. Transactions are likely to contain many parts, such as updating a customer account, adjusting product inventory, and updating the seller's account receivable. To prevent data integrity all parts of a transaction must be successfully completed.

Transaction - A sequence of database requests that accesses the database. It is a logical unit of work; that is, it must be entirely completed or aborted. All transactions must have the properties of atomicity, consistency, isolation and durability.

To understand the concept of a transaction, suppose that you sell a product to a customer. Furthermore, suppose that the customer may charge the purchase to his or her account. Given that scenario, your sales transaction consists of at least the following parts.

- You must write a new customer invoice.
- You must reduce the quantity on hand in the product's inventory.
- You must update the account transactions.
- You must update the customer balance.

OBJECTIVES: *At the end of this lesson, you will be able to:*

- ✓ Understand about database transactions and their properties
- ✓ Define what concurrency control is a what role it plays in maintaining the database's integrity
- ✓ Discuss what are the locking methods and how they work
- ✓ Understand stamping and optimistic methods are used for concurrency control
- ✓ Understand how database recovery management is used to maintain database integrity

The preceding sales transaction must be reflected in the database. In database terms, a transaction is any action that reads from or writes to a database. A transaction may consist of the following.

- A simple SELECT statement to generate a list of table contents
- A series of related UPDATE statements to change the values of attributes in various tables.
- A series of INSERT statements to add rows to one or more tables.
- A combination of SELECT, UPDATE, and INSERT statements

Transaction

- Logical unit of work that must be entirely completed or aborted
- Consists of:
  - SELECT statement
  - Series of related UPDATE statements
  - Series of INSERT statements
  - Combination of SELECT, UPDATE, and INSERT statements

Consistent database state: All data integrity constraints are satisfied

- Must begin with the database in a known consistent state to ensure consistency

Formed by two or more database requests

- Database requests: Equivalent of a single SQL statement in an application program or transaction

Consists of a single SQL statement or a collection of related SQL statements

Evaluating Transaction Results

- Not all transactions update database
  - SQL code represents a transaction because it accesses a database
- Improper or incomplete transactions can have devastating effect on database integrity
  - Users can define enforceable constraints based on business rules
  - Other integrity rules are automatically enforced by the DBMS

Transaction Properties

There are four key properties of transactions that a DBMS must ensure to maintain data in the face of concurrent access and system failures:

The term **ACID** refer to the four properties of transactions presented: (*atomicity, consistency, isolation and durability*

Atomicity - Users should be able to regard the execution of each transaction.

- All operations of a transaction must be completed
- If not, the transaction is aborted

Consistency - Each transaction, run by itself with no current execution of other transactions must preserve the consistency of the database.

- Permanence of database's consistent state
- Ensuring that property of a transaction is the responsibility of the user.

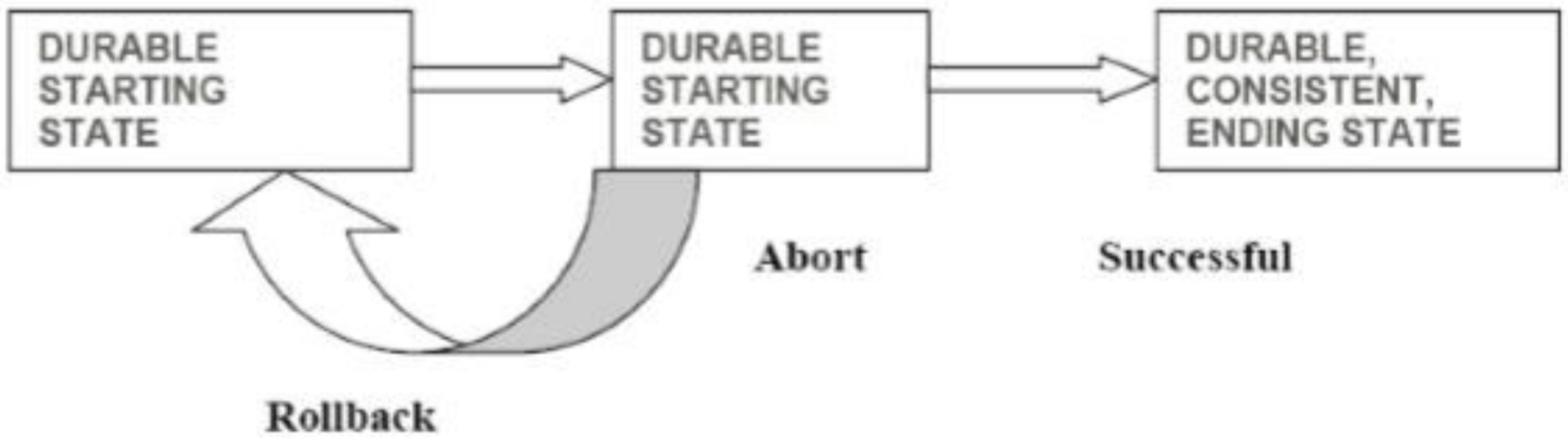


Figure 1: Database Transaction States

Isolation – Users should be able to understand a transaction without considering the effect of other concurrently executing transactions, even if the DBMS interleaves the actions of several transactions for performance reasons.

- Transactions are isolated, or protected, from the effects of concurrently scheduling other transactions.
- Data used during transaction cannot be used by second transaction until the first is completed

Durability – Informs the user that a transaction has been successfully completed, its effects should persist even if the system crashes before all its changes are reflected on disk

- Ensures that once transactions are committed, they cannot be undone or lost

Transactions and Schedules

A transaction is seen by the DBMS as a series, or list, of actions.

The actions that can be executed by a transaction include reads and writes of database objects. A transaction can also be defined as a set of actions that are partially ordered. That is, the relative order of some of the actions may not be important.

Abort *T* denotes the action of ***T*** aborting, and ***Commit T*** denotes ***T*** committing.

Schedule is a list of actions (reading, writing, aborting, or committing) from a set of transactions, in which two actions of a transaction T appear in a schedule must be the same as the order in which they appear in T.

Concurrency control for database management

Systems permits many users. Control is needed to coordinate concurrent accesses to a DBMS so that the overall correctness of the database is maintained.

For example, users A and B both may wish to read and update the same record in the database at about the same time.

The relative timing of the two transactions may have an impact on the state of the database at the end of the transactions.

The result may be an inconsistent database.

Several Functions of Concurrent Control

- **Lost update problem**
  It occurs when two transactions that access the same database items have their operations interleaved.

- **Temporary update problem**
  It occurs when one transaction updates a database item and then the transaction fails for some reason.

- **Incorrect summary problem**
  If one transaction is calculating on a number of records while other transaction is updating some records, the function may calculate some values before they are updated and others after they are updated

Problems created due to the concurrent execution of the transactions:

1. Multiple update problems
2. Incorrect Analysis Problem
3. Inconsistent Retrievals
4. Uncommitted Dependency
5. Serializability

Concurrency Control

Coordination of the simultaneous transactions execution in a multiuser database system
Objective - Ensures serializability of transactions in a multiuser database environment

Problems in Concurrency Control

Lost update
• Occurs in two concurrent transactions when:
  • Same data element is updated
• One of the updates is lost

Uncommitted data

Occurs when:
• Two transactions are executed concurrently
• First transaction is rolled back after the second transaction has already accessed uncommitted data

Inconsistent retrievals
• Occurs when a transaction accesses data before and after one or more other transactions finish working with such data

Concurrency Control with Locking Methods

Locking methods are one of the most common techniques used in concurrency control because they facilitate the isolation of data items used in concurrently executing transactions. A lock guarantees exclusive use of a data item to a current transaction. In other words, transaction T2 does not have access to a data item that is currently being used by transaction T1.

A transaction acquires a lock prior to data access:

- the lock is released (unlocked) when the transaction is completed so that another transaction can lock the data item for its exclusive use.

- this series of locking actions assumes that concurrent transactions might attempt to manipulate the same data item at the same time.

- the use of locks based on the assumption that conflict between transactions is likely is usually referred to as pessimistic locking.

Database Recovery Management

- Database recovery: Restores database from a given state to a previously consistent state
- The recovery manager of a DBMS is responsible for ensuring atomicity and durability.
- Recovery transactions are based on the atomic transaction property
  - **Atomic transaction property:** All portions of a transaction must be treated as a single logical unit of work
  - If transaction operation cannot be completed:

Transaction must be aborted
Changes to database must be rolled back

Concepts that Affect Transaction Recovery

Deferred-write technique or deferred update
- Ensures that transaction logs are always written before the data are updated

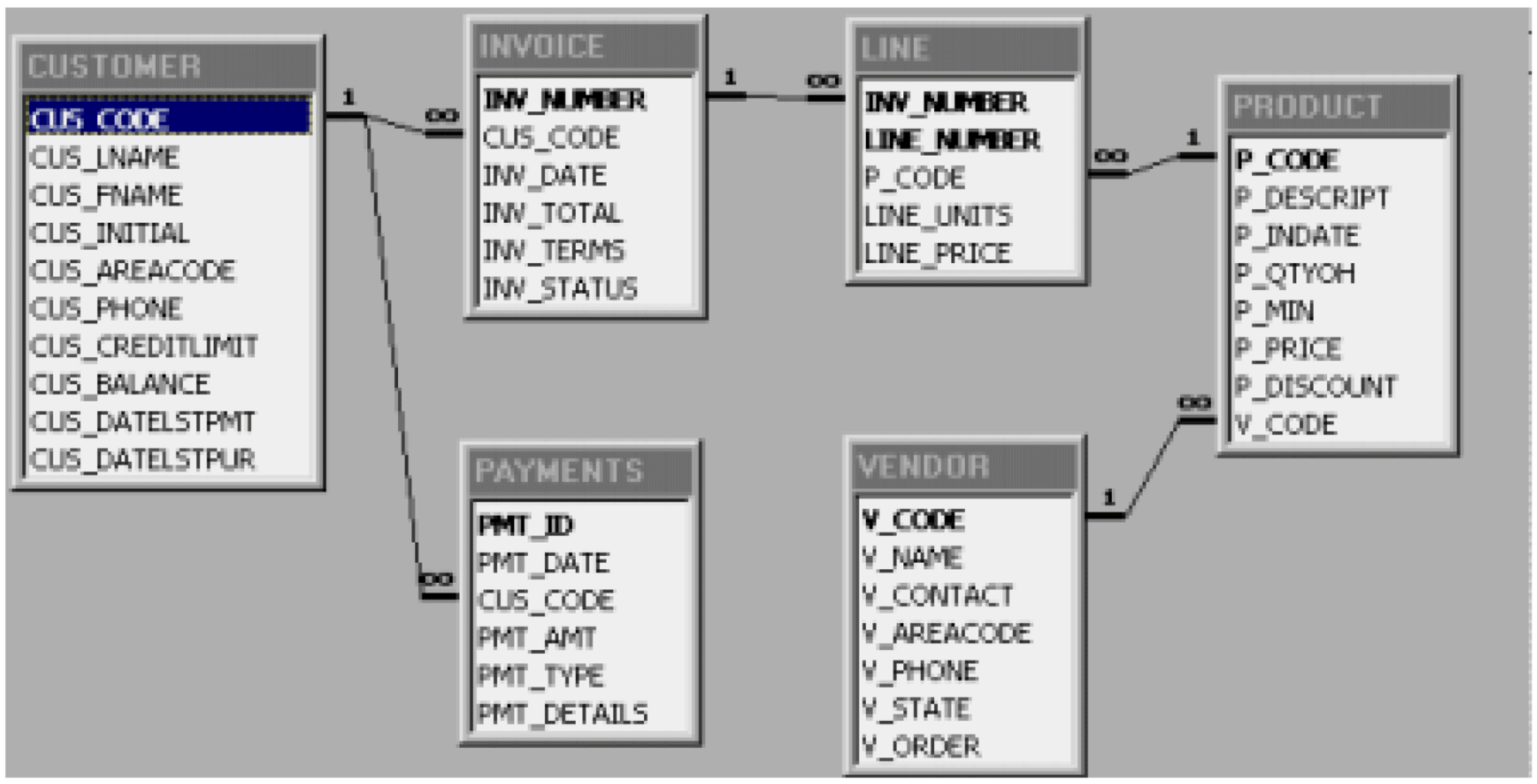
Redundant transaction logs
- Ensure that a physical disk failure will not impair the DBMS's ability to recover data

Buffers
- Temporary storage areas in a primary memory

Checkpoints
- Allows DBMS to write all its updated buffers in memory to disk

SAQ: Self-Assessment Question

Using this database, write the SQL code to represent each one of the following transactions. Use BEGIN TRANSACTION and COMMIT to group the SQL statements in logical transactions.



Sample Problem with Solution

A. On May 11, 2006, customer '10010' makes a credit purchase (30 days) of one unit of product '11QER/31' with a unit price of $110.00; the tax rate is 8 percent. The invoice number is 10983, and this invoice has only one product line.

```
 BEGIN TRANSACTION
INSERT INTO INVOICE
        VALUES (10983, '10010', '11-May-2006', 118.80, '30', 'OPEN');
        INSERT INTO LINE
        VALUES (10983, 1, '11QER/31', 1, 110.00);
        UPDATE PRODUCT
        SET P_QTYOH = P_QTYOH – 1
        WHERE P_CODE = '11QER/31';
        UPDATE CUSTOMER
        SET CUS_DATELSTPUR = '11-May-2006', CUS_BALANCE = CUS_BALANCE +118.80
        WHERE CUS_CODE = '10010';
        COMMIT;
```

DO YOUR PART:

Answer the problem given below:

A. On June 3, 2006, customer '10010' makes a payment of $100 in cash. The payment ID is 3428.

B. Describe the three most common concurrent transaction execution problems. Explain how concurrency control can be used to avoid those problems.

C. What DBMS component is responsible for concurrency control? How is this feature used to resolve conflicts?

SUMMARY

A transaction is a sequence of database operations that access the database. A transaction is a logical unit of work; that is, all parts are executed, or the transaction is aborted. A transaction takes a database from one consistent state to another. A consistent database state is one in which all data integrity constraints are satisfied.

Transactions have four main properties: atomicity, consistency, isolation, and durability. Atomicity means that all parts of the transaction must be executed; otherwise the transaction is aborted. Consistency means that the database's consistent state is maintained. Isolation means that data used by one transaction cannot be accessed by another transaction until the first one is completed; Durability means that changes made by a transaction cannot be rolled back once the transaction is committed.

SQL provides support for transaction using two statements: COMMIT, which saves changes to disk, and ROLLBACK, which restores the previous database state. SQL transactions are formed by several SQL statements or database request. Each database requests originates several I/O database operations. The transaction log keeps track of all transactions that modify the database.

SAQ 1.1 - SELF-ASSESSMENT QUESTIONS

1. Explain the following statement: A transaction is a logical unit of work.

Answer:

A transaction is a logical unit of work that must be entirely completed of aborted; no intermediate states are accepted. In other words, a transaction, composed of several database requests, is treated by the DBMS as a unit of work in which all transaction steps must be fully completed if the transaction is to be accepted by the DBMS.

Acceptance of an incomplete transaction will yield an inconsistent database state. To avoid such a state, the DBMS ensures that all of a transaction's database

operations are completed before they are committed to the database. For example, a credit sale requires a minimum of three database operations:

1. An invoice is created for the sold product.
2. The product's inventory quantity on hand is reduced.
3. The customer accounts payable balance is increased by the amount listed on the invoice.

If only parts 1 and 2 are completed, the database will be left in an inconsistent state. Unless all three parts (1, 2, and 3) are completed, the entire sales transaction is canceled.

2. **What is a consistent database state, and how it is achieved?**

   A consistent database state is one in which all data integrity constraints are satisfied. To achieve a consistent database state, a transaction must take the database from one consistent state to another.

3. **The DBMS does not guarantee that the semantic meaning of the transactions truly represents the real-world event. What are the possible consequences of that limitation? Give an example.**

   The database is designed to verify the syntactic accuracy of the database commands given by the user to be executed by the DBMS. The DBMS will check that the database exists, that the referenced attributes exist in the selected tables, that the attribute data types are correct, and so on. Unfortunately, the DBMS is not designed to guarantee that the syntactically correct transaction accurately represents the real-world event.

4. **List and discuss the five transaction properties.**

   The five transaction properties are:

   Atomicity requires that all parts of a transaction must be completed or the transaction is aborted. This property ensures that the database will remain in a consistent state.

   Consistency Indicates the permanence of the database consistent state.

   Isolation means that the data required by an executing transaction cannot be accessed by any other transaction until the first transaction finishes. This property ensures data consistency for concurrently executing transactions.

   Durability indicates that the database will be in a permanent consistent state after the execution of a transaction. In other words, once a consistent state is reached, it cannot be lost.

   Serializability means that a series of concurrent transactions will yield the same result as if they were executed one after another.

SUPPLEMENTARY LEARNING RESOURCES:

Chapter 11 database performance tuning and query optimization. (n.d.). StuDocu. https://www.studocu.com/en-au/document/university-of-new-south-wales/enterprise-database-management/lecture-notes/chapter-11-database-performance-tuning-and-query-optimization/4296791/view

Coronel, Carlos / Morris Steven (2017) DATABASE SYSTEMS: Design, Implementation, and Management 12e, Boston, MA: Cengage

DBMS - Concurrency control. (n.d.). RxJS, ggplot2, Python Data Persistence, Caffe2, PyBrain, Python Data Access, H2O, Colab, Theano, Flutter, KNime, Mean.js, Weka, Solidity. https://www.tutorialspoint.com/dbms/dbms_concurrency_control.htm

Database (2019, May 15). Transaction and concurrency control. Medium. https://medium.com/@jinghua.shih/database-transaction-and-concurrency-control-abb7fd16c507

Retrieved from http://solr.bccampus.ca:8001/bcc/file/2819cb7e-46a3-4e07-a767-68d85f0656c0/1/D2LExport_ICT216.zip/Week%2010%20Transaction%20Mana