Assignment7 Q2

-- the stored procedure is named test that contains three SQL statements that are UPDATE, DELETE and UPDATE that are coded as transaction--

-- to start this stored procedure declare a variable named sql\_error and sets it to FALSE to indicate that no SQL error has occurd--

-- then the second DECLARE statment creates a condition handler that sets the sql\_error variable to TRUE if sql error occurs--

-- start transaction statement identifies start of transaction and update the invoice--

-- delete the vendor id--

-- update the vendor name--

-- If statement is uses whether the sql\_error occurs when executing any of the statements--

-- If sql error did not occur this code will use the commit statement to commit the changes to the database--

-- otherwise rollback statement rollback the changes--

use ap;

DROP PROCEDURE IF EXISTS test;

DELIMITER //

CREATE PROCEDURE test()

BEGIN

DECLARE sql\_error INT DEFAULT FALSE;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION

SET sql\_error = TRUE;

START TRANSACTION;

UPDATE invoices SET vendor\_id = 123 WHERE vendor\_id = 122;

DELETE FROM vendors WHERE vendor\_id = 122;

UPDATE vendors SET vendor\_name = 'FedUP' WHERE vendor\_id = 123;

IF sql\_error = FALSE THEN

COMMIT;

SELECT 'The transaction was committed.';

ELSE

ROLLBACK;

SELECT 'The transaction was rolled back.';

END IF;

END//

DELIMITER ;

CALL test();

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application

Description automatically generated

-- Assignment7 Q3--

-- the stored procedure is named test that contains two SQL statements that are , DELETE and DELETE that are coded as transaction--

-- to start this stored procedure declare a variable named sql\_error and sets it to FALSE to indicate that no SQL error has occurd--

-- then the second DECLARE statment creates a condition handler that sets the sql\_error variable to TRUE if sql error occurs--

-- start transaction statement identifies start of transaction and update the invoice--

-- delete invoice id 114--

-- delete all line items from the invoice\_line\_items table--

-- If statement is uses whether the sql\_error occurs when executing any of the statements--

-- If sql error did not occur this code will use the commit statement to commit the changes to the database--

-- otherwise rollback statement rollback the changes--

USE ap;

DROP PROCEDURE IF EXISTS test;

DELIMITER //

CREATE PROCEDURE test()

BEGIN

DECLARE sql\_error INT DEFAULT FALSE;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION

SET sql\_error = TRUE;

START TRANSACTION;

DELETE FROM invoice\_line\_items

WHERE invoice\_id = 114;

DELETE FROM invoices

WHERE invoice\_id = 114;

COMMIT;

IF sql\_error = FALSE THEN

COMMIT;

SELECT 'The transaction was committed.';

ELSE

ROLLBACK;

SELECT 'The transaction was rolled back.';

END IF;

END//

DELIMITER ;

CALL test();

