

# Using Transaction Control Statements

## What Will I Learn?

In this lesson, you will learn to:

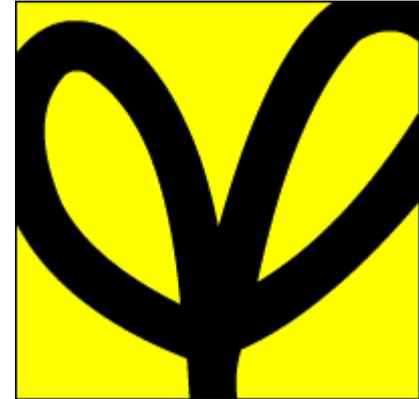
- Define a transaction and provide an example
- Construct and execute a transaction control statement in PL/SQL





## Why Learn It?

In this lesson, you learn how to include transaction control statements such as COMMIT, ROLLBACK, and SAVEPOINT in PL/SQL.



# Tell Me/Show Me



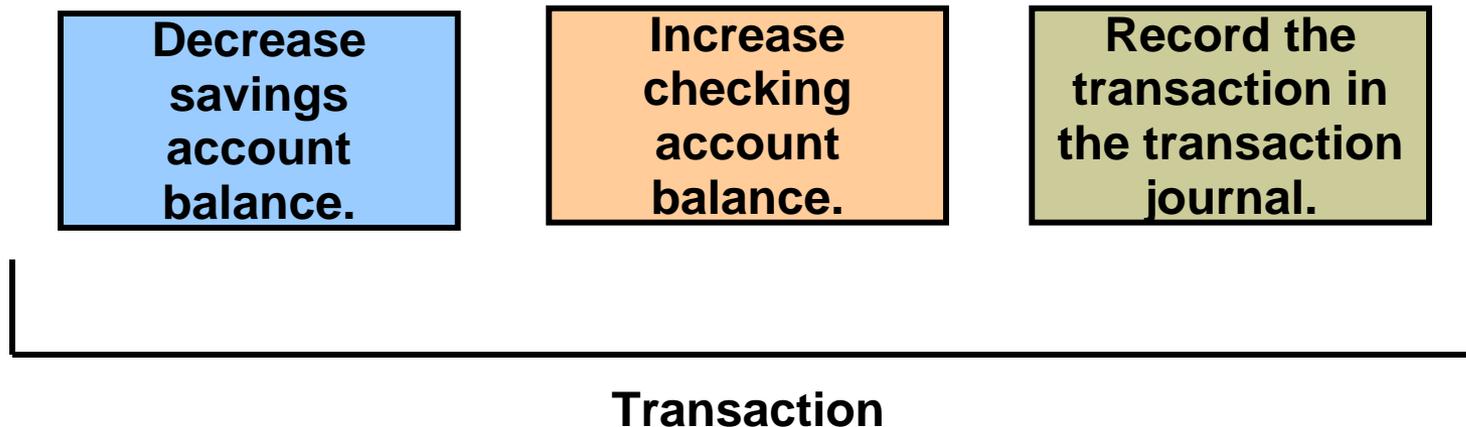
## Database Transaction

A transaction is an inseparable list of database operations that must be executed either in its entirety or not at all. Transactions maintain data integrity and guarantee that the database is always in a consistent state.

## Tell Me/Show Me

### Example of a Transaction

To illustrate the concept of a transaction, consider a banking database. When a bank customer transfers money from a savings account to a checking account, the transaction can consist of three separate operations:





## Tell Me/Show Me

### Example of a Transaction (continued)

What would happen if there were insufficient funds in the savings account? Would the funds still be added to the checking account? Would an entry be logged in the transaction journal? What do you think *should* happen?

Decrease  
savings  
account.

```
UPDATE savings_accounts  
SET balance = balance - 500  
WHERE account = 3209;
```

Increase  
checking  
account.

```
UPDATE checking_accounts  
SET balance = balance + 500  
WHERE account = 3208;
```

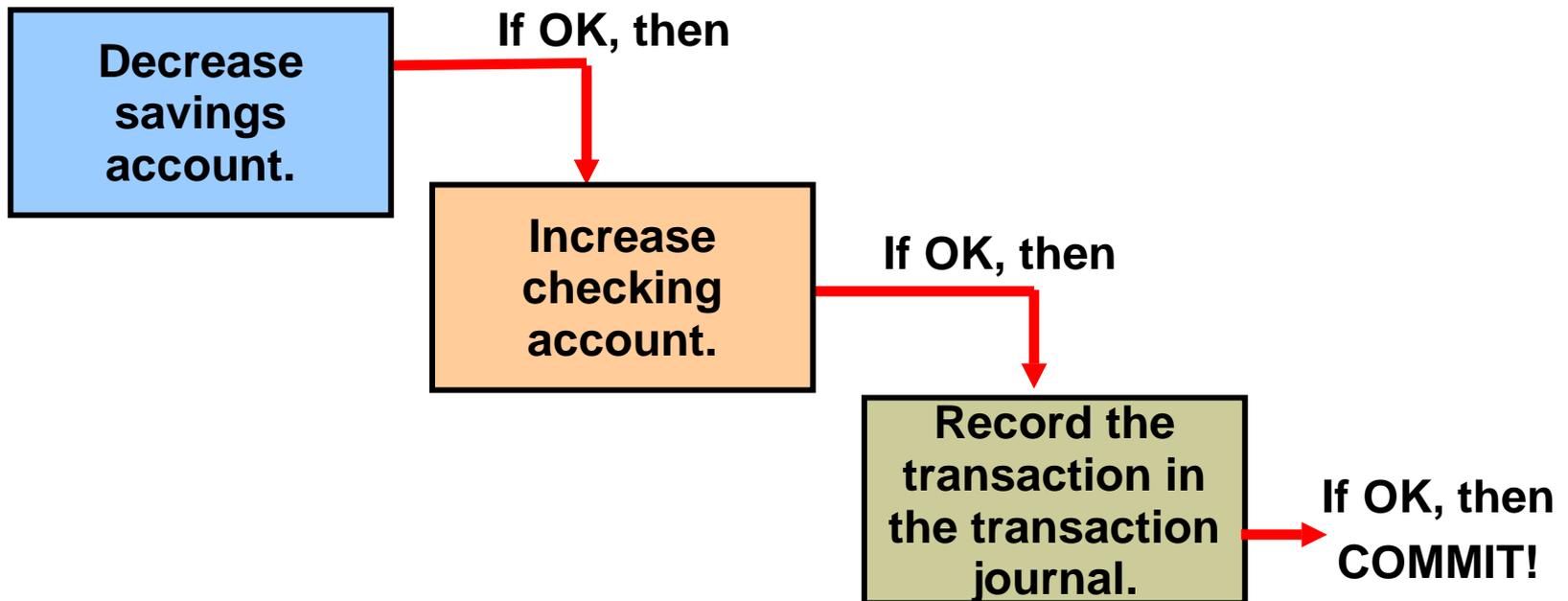
Record the  
transaction in  
the transaction  
journal.

```
INSERT INTO journal VALUES  
(journal_seq.NEXTVAL, '1B'  
3209, 3208, 500);
```

## Tell Me/Show Me

### Example of a Transaction (continued)

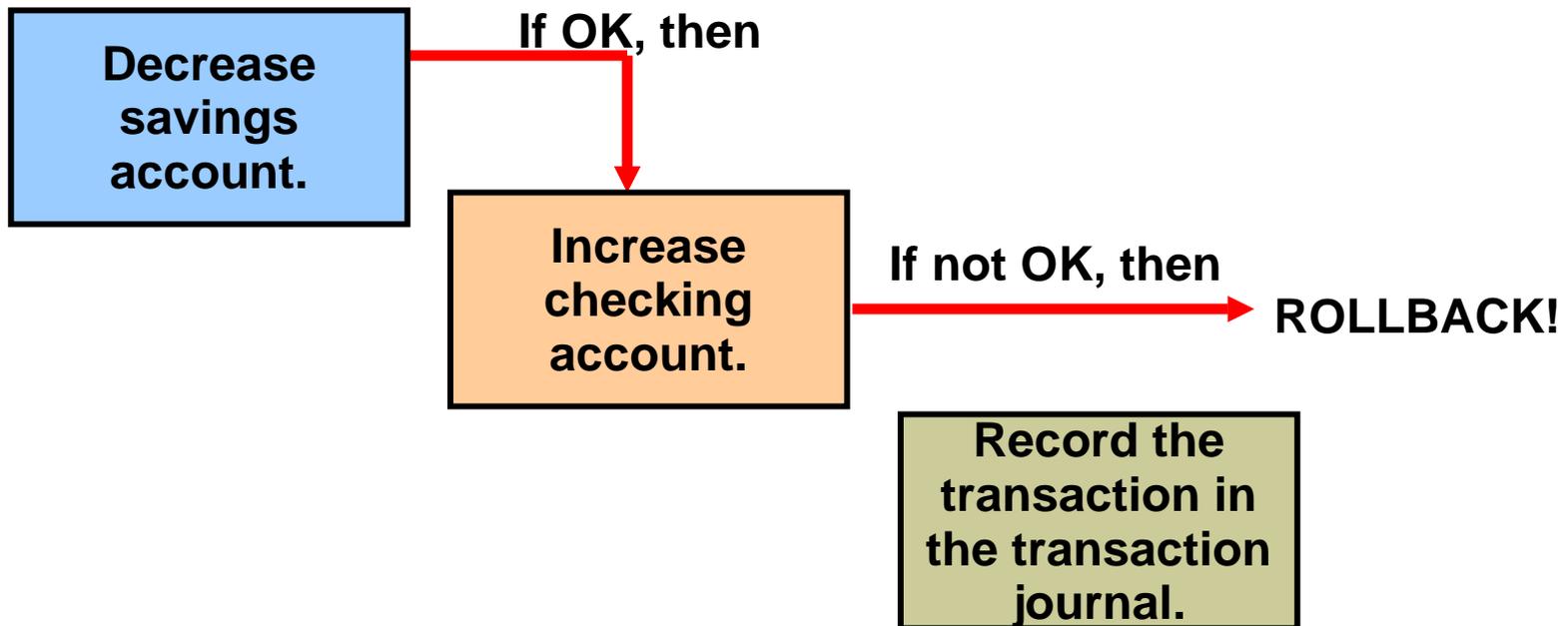
If all three SQL statements can be performed to maintain the accounts in proper balance, the effects of the transaction can be committed, or applied to the database tables.



## Tell Me/Show Me

### Example of a Transaction (continued)

However, if a problem, such as insufficient funds, invalid account number, or a hardware failure prevents one or two of the statements in the transaction from completing, the entire transaction must be rolled back (reversed out) so that the balance of all accounts is correct.



# Tell Me/Show Me

## Transaction Control Statements

You use transaction control statements to make the changes to the database permanent or to discard them. The three main transaction control statements are:

- COMMIT
- ROLLBACK
- SAVEPOINT

The transaction control commands are valid in PL/SQL and therefore can be used directly in the executable or exception section of a PL/SQL block.



## Tell Me/Show Me

### COMMIT

COMMIT is used to make the database changes permanent. If a transaction ends with a COMMIT statement, all the changes made to the database during that transaction are made permanent.

```
BEGIN
  INSERT INTO pairtable VALUES (1, 2);
  COMMIT;
END;
```

Note: The keyword END signals the end of a PL/SQL block, not the end of a transaction.



## Tell Me/Show Me

### ROLLBACK

ROLLBACK is for discarding any changes that were made to the database after the last COMMIT. If the transaction fails, or ends with a ROLLBACK, then none of the statements takes effect.

```
BEGIN
  INSERT INTO pairtable VALUES (3, 4);
  ROLLBACK;
  INSERT INTO pairtable VALUES (5, 6);
  COMMIT;
END;
```

In the example, only the second INSERT statement adds a row of data.



## Tell Me/Show Me

### SAVEPOINT

SAVEPOINT is used to mark an intermediate point in transaction processing.

```
BEGIN
  INSERT INTO pairtable VALUES (7, 8);
  SAVEPOINT my_sp_1;
  INSERT INTO pairtable VALUES (9, 10);
  SAVEPOINT my_sp_2;
  INSERT INTO pairtable VALUES (11, 12);
  ROLLBACK to my_sp_1;
  INSERT INTO pairtable VALUES (13, 14);
  COMMIT;
END;
```

Only ROLLBACK can be used to a SAVEPOINT.

# Tell Me/Show Me

## Terminology

Key terms used in this lesson include:

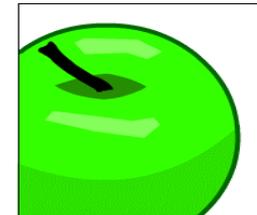
Transaction

COMMIT

END

ROLLBACK

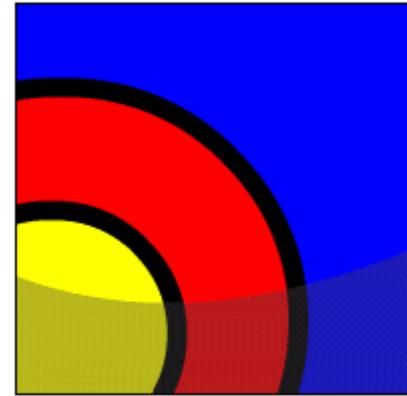
SAVEPOINT



## Summary

In this lesson, you learned to:

- Define a transaction and provide an example
- Construct and execute a transaction control statement in PL/SQL



## Try It/Solve It

The exercises in this lesson cover the following topics:

- Defining transactions and providing an example
- Constructing and executing a transaction control statement in PL/SQL

