Database Programming with PL/SQL
3-4: Using Transaction Control Statements
Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

	An inseparable list of database operations, which must be executed either in its entirety or not at all.
	Used for discarding any changes that were made to the database after the last COMMIT.
	Used to mark an intermediate point in transaction processing.
	Keyword used to signal the end of a PL/SQL block, not the end of a transaction.
	Statement used to make database changes permanent.

Try It / Solve It

Because our online version of Oracle Application Express (APEX) automatically commits changes as the code runs, the following activities will NOT work as intended unless you are using an installed/local APEX environment.

1. How many transactions are shown in the following code? Explain your reasoning.

```
BEGIN
  INSERT INTO my_savings (account_id, amount)
       VALUES (10377, 200);
  INSERT INTO my_checking (account_id, amount)
       VALUES (10378, 100);
END;
```

Because our online version of Oracle Application Express (APEX) automatically commits changes as the code runs, the following activities will NOT work as intended unless you are using an installed/local APEX environment.

2. Create the endangered species table by running the following statement in Application Express:

Unless you are using an installed/local APEX environment, there is no reason to run this code. If you are using our online version of Oracle Application Express (APEX), you should pretend this code runs successfully before you try to answer the next question.

```
CREATE TABLE endangered_species
  (species_id         NUMBER(4) CONSTRAINT es_spec_pk PRIMARY KEY,
   common_name  VARCHAR2(30) CONSTRAINT es_com_name_nn NOT NULL,
   scientific_name  VARCHAR2(30) CONSTRAINT es_sci_name_nn NOT NULL);
```

3. Examine the following block of code. If you were to run this block, what data do you think would be saved in the database?

Unless you are using an installed/local APEX environment, there is no reason to run this code. If you are using our online version of Oracle Application Express (APEX), you should pretend this code runs successfully to answer the remaining questions.

```
BEGIN
  INSERT INTO endangered_species
       VALUES (100, 'Polar Bear', 'Ursus maritimus');
  SAVEPOINT sp_100;
  INSERT INTO endangered_species
      VALUES (200, 'Spotted Owl', 'Strix occidentalis');
  SAVEPOINT sp_200;
  INSERT INTO endangered_species
      VALUES (300, 'Asiatic Black Bear', 'Ursus thibetanus');
  ROLLBACK TO sp_100;
  COMMIT;
END;
```

4. Run the block above to test your theory. Confirm your projected data was added.

Unless you are using an installed/local APEX environment, you should skip this question. The block above will NOT run as intended in our online version of Oracle Application Express (APEX) because it automatically commits changes as the code runs.

5. Examine the following block. If you were to run this block, what data do you think would be saved in the database? Run the block to test your theory.

Because our online version of Oracle Application Express (APEX) automatically commits changes as the code runs, the following block will NOT work as intended unless you are using an installed/local APEX environment.

```
BEGIN
  INSERT INTO endangered_species
    VALUES (400, 'Blue Gound Beetle', 'Carabus intricatus');
 SAVEPOINT sp_400;
 INSERT INTO endangered_species
    VALUES (500, 'Little Spotted Cat', 'Leopardus tigrinus');
 ROLLBACK;
 INSERT INTO endangered_species
    VALUES (600, 'Veined Tongue-Fern', 'Elaphoglossum nervosum');
 ROLLBACK TO sp_400;
END;
```