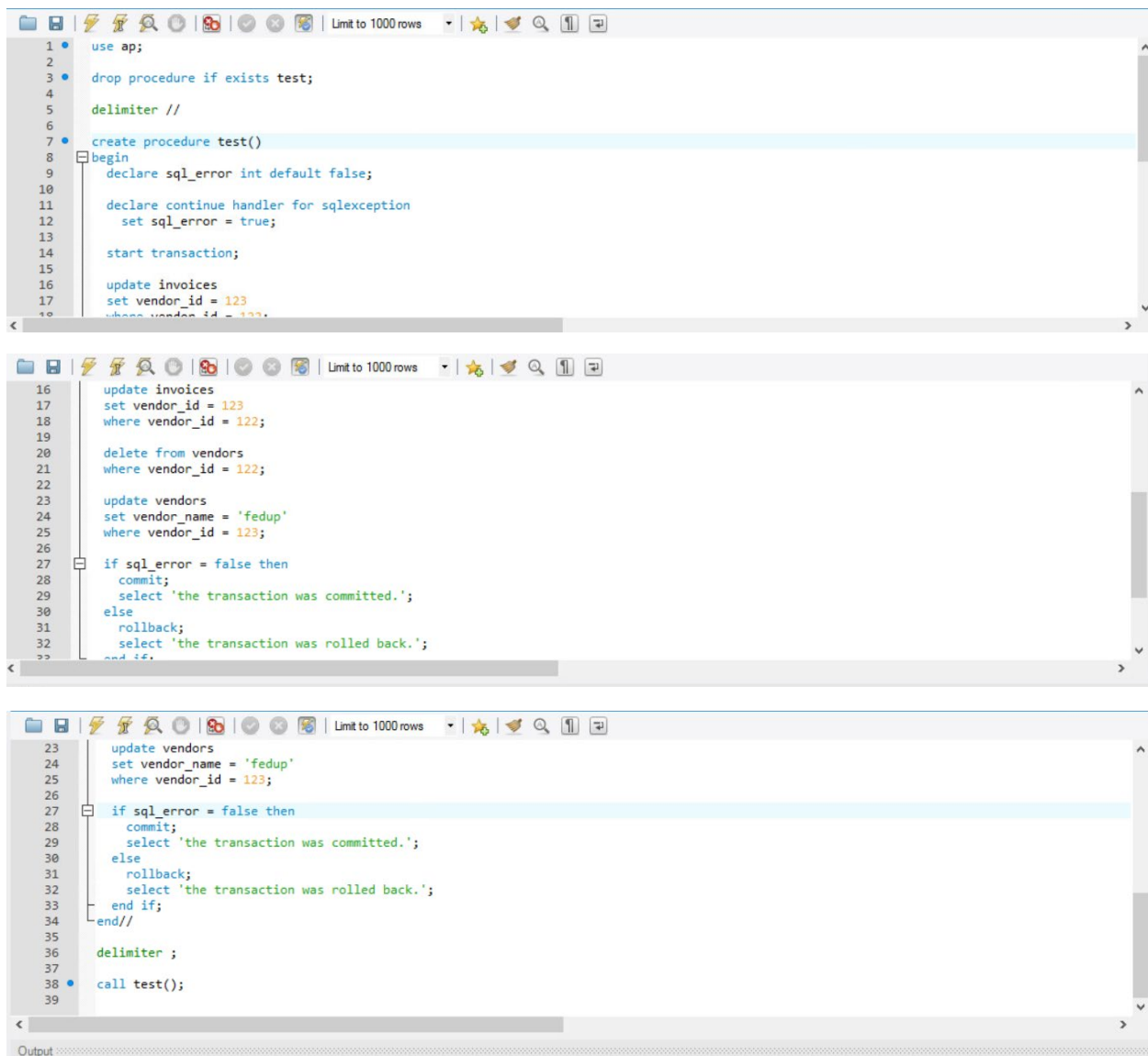


Chapter 14

1.)



```
1 use ap;
2
3 drop procedure if exists test;
4
5 delimiter //
6
7 create procedure test()
8 begin
9     declare sql_error int default false;
10
11     declare continue handler for sqlexception
12         set sql_error = true;
13
14     start transaction;
15
16     update invoices
17     set vendor_id = 123
18     where vendor_id = 122;
19
20     delete from vendors
21     where vendor_id = 122;
22
23     update vendors
24     set vendor_name = 'fedup'
25     where vendor_id = 123;
26
27     if sql_error = false then
28         commit;
29         select 'the transaction was committed.';
30     else
31         rollback;
32         select 'the transaction was rolled back.';
33     end if;
34 end//
35
36 delimiter ;
37
38 call test();
39
```

Output

The screenshot shows the SQL Developer interface. The code editor contains the following PL/SQL block:

```
33   end if;  
34   end//  
35  
36   delimiter ;  
37  
38   call test();  
39
```

Below the code editor, the 'Result Grid' shows two rows of data:

the transaction was committed.
the transaction was committed.

The 'Action Output' window is open, displaying the following table:

#	Time	Action	Message	Duration / Fetch
1	07:52:05	use ap;	0 row(s) affected	0.000 sec
2	07:52:05	drop procedure if exists test;	0 row(s) affected	0.000 sec
3	07:52:05	create procedure test() begin declare sql_error int default false; declare continue handler for sqlexception ...	0 row(s) affected	0.000 sec
4	07:52:05	call test();	1 row(s) returned	0.047 sec / 0.000 sec

2.)

The first screenshot shows the beginning of the PL/SQL code:

```
1   use ap;  
2  
3   drop procedure if exists test;  
4  
5   delimiter //  
6  
7   create procedure test()  
8   begin  
9       declare sql_error int default false;  
10  
11       declare continue handler for sqlexception  
12           set sql_error = true;  
13  
14       start transaction;  
15  
16       delete from invoice_line_items  
17       where invoice_id = 114;  
18
```

The second screenshot shows the continuation of the code:

```
19   delete from invoices  
20   where invoice_id = 114;  
21  
22   commit;  
23  
24   if sql_error = false then  
25       commit;  
26       select 'the transaction was committed.';  
27   else  
28       rollback;  
29       select 'the transaction was rolled back.';  
30   end if;  
31 end//  
32  
33 delimiter ;  
34  
35 call test();  
36
```

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a SQL script with the following code:

```
30 end if;  
31 end//  
32  
33 delimiter ;  
34  
35 call test();  
36
```

The middle pane shows the results of the script execution, which are two rows of text:

the transaction was committed.
the transaction was committed.

The bottom pane shows the Action Output, which is a table with the following data:

#	Time	Action	Message	Duration / Fetch
1	08:06:17	use ap	0 row(s) affected	0.000 sec
2	08:06:17	drop procedure if exists test	0 row(s) affected	0.000 sec
3	08:06:17	create procedure test() begin declare sql_error int default false; declare continue handler for sqlexception ...	0 row(s) affected	0.000 sec
4	08:06:17	call test()	1 row(s) returned	0.125 sec / 0.000 sec