

Islamic University of Technology

RDBMS

CSE 4508

Lab Report 8

Name : Amina

Student ID : 200041155

Section : 1

Lab Group : 1B (Shifted from 1A)

Date of Performance : 01/11/23

Date of Submission : 02/11/23

Task 1

In task 1, I need to create a trigger that will automatically insert user id after inserting user name. For generating user id sequentially I used sequence function. And then I made a trigger which would insert user id before inserting a row in the table.

Then I inserted according to the helper file and generated results.

Table creation and PL/SQL procedure:

```
CREATE TABLE USERINFO(  
    USERID INT,  
    NAME VARCHAR(10)  
);  
  
DROP SEQUENCE USERIDSEQ;  
  
CREATE SEQUENCE USERIDSEQ  
MINVALUE 1  
MAXVALUE 999999  
START WITH 1  
INCREMENT BY 1  
CACHE 20;  
  
DROP TRIGGER BEFOREINSERTINUSER;  
  
CREATE OR REPLACE TRIGGER BEFOREINSERTINUSER BEFORE  
    INSERT ON USERINFO FOR EACH ROW  
BEGIN  
    IF :NEW.USERID IS NULL THEN  
        :NEW.USERID := USERIDSEQ.NEXTVAL;  
    END IF;  
END;  
/
```

Result:

```
SQL> select * from UserInfo;
```

USERID	NAME
1	monir
2	shabnur
3	nargis
4	jalil

Task 2

In this task, I made one table FrndList using a procedure makeFriends. I made another procedure to check if any id is invalid and then inserts only the valid ids in the result table.

Then I made the makeFriends procedure to simply insert into the table.

Then I inserted data from helper file and generated output.

In the created procedure, I can generate error message for wrong input.

Below is the implementation of the procedure :

```
CREATE OR REPLACE TRIGGER INVALIDIDCHECK BEFORE
  INSERT ON FRNDLIST FOR EACH ROW
DECLARE
  ID1 INT;
  ID2 INT;
BEGIN
  SELECT
    USERID INTO ID1
  FROM
    USERINFO
  WHERE
    USERID = :NEW.USERID1;
  SELECT
    USERID INTO ID2
  FROM
    USERINFO
  WHERE
    USERID = :NEW.USERID2;
  IF ID1 IS NULL OR ID2 IS NULL OR ID1 = ID2 THEN
    RAISE_APPLICATION_ERROR(-1718, 'Invalid user ID(s)');
  END IF;
END;
/

CREATE OR REPLACE PROCEDURE MAKEFRIENDS(
  USER_ID1 INT,
  USER_ID2 INT
) AS
BEGIN
  INSERT INTO FRNDLIST VALUES (
    USER_ID1,
    USER_ID2
  );
  INSERT INTO FRNDLIST VALUES (
    USER_ID2,
    USER_ID1
  );
END MAKEFRIENDS;
/
```

Result:

```
SQL> select * from FrndList;
```

USERID1	USERID2
1	2
2	1
3	2
2	3
1	3
3	1
4	2
2	4

```
8 rows selected.
```

Task 3

In this task, I made one table Message using a procedure FRNDCHECK which checks if two person trying to send message are friends or not.

Then I inserted data from helper file and generated output.

In the created procedure, I can generate error message for wrong input.

Below is the implementation of the procedure :

```
CREATE OR REPLACE TRIGGER INVALIDIDCHECK BEFORE
  INSERT ON FRNDLIST FOR EACH ROW
DECLARE
  ID1 INT;
  ID2 INT;
BEGIN
  SELECT
    USERID INTO ID1
  FROM
    USERINFO
  WHERE
    USERID = :NEW.USERID1;
  SELECT
    USERID INTO ID2
  FROM
    USERINFO
  WHERE
    USERID = :NEW.USERID2;
  IF ID1 IS NULL OR ID2 IS NULL OR ID1 = ID2 THEN
    RAISE_APPLICATION_ERROR(-1718, 'Invalid user ID(s)');
  END IF;
END;
/

CREATE OR REPLACE PROCEDURE MAKEFRIENDS(
  USER_ID1 INT,
  USER_ID2 INT
) AS
BEGIN
  INSERT INTO FRNDLIST VALUES (
    USER_ID1,
    USER_ID2
  );
  INSERT INTO FRNDLIST VALUES (
    USER_ID2,
    USER_ID1
  );
END MAKEFRIENDS;
/
```

Result:

```
SQL>
SQL> select * from Message;

SENDERID RECEIVERID TEXT
-----
3          1 Hi
1          3 Hello
3          1 How are you ?
4          2 Give me the solution of todays task.
1          3 Fine.
2          1 Jalil is asking for task again. Should I give it to him?
3          1 Do you have a girlfriend?
1          2 No.
1          3 Yes, her name is Shabnur.

9 rows selected.
```

Task 4

For this task, I created a procedure to delete message from the message table. But before deleting it also checks if the provided ids are friends or not.

Also I created a trigger for a correct deletion.

Then I inserted data from helper file and generated output.

In the created procedure, I can generate error message for wrong input.

Below is the implementation of the procedure :

```
CREATE OR REPLACE PROCEDURE UNFRIEND(  
    USER_ID1 INT,  
    USER_ID2 INT  
) AS  
    COUNTING INT := 0;  
BEGIN  
    SELECT  
        COUNT(*) INTO COUNTING  
    FROM  
        FRNDLIST  
    WHERE  
        (USERID1 = USER_ID1  
        AND USERID2 = USER_ID2)  
        OR (USERID1 = USER_ID2  
        AND USERID2 = USER_ID1);  
    IF COUNTING = 0 THEN  
        RAISE_APPLICATION_ERROR(-9, 'Users are not friends (ORA-00009)');  
    ELSE  
        DELETE FROM FRNDLIST  
        WHERE  
            (USERID1 = USER_ID1  
            AND USERID2 = USER_ID2)  
            OR (USERID1 = USER_ID2  
            AND USERID2 = USER_ID1);  
    END IF;  
END UNFRIEND;  
/  
  
CREATE OR REPLACE TRIGGER MESSAGEDELETE BEFORE  
    DELETE ON FRNDLIST FOR EACH ROW  
BEGIN  
    DELETE FROM MESSAGE  
    WHERE  
        (SENDERID = :OLD.USERID1  
        AND RECEIVERID = :OLD.USERID2)  
        OR (SENDERID = :OLD.USERID2  
        AND RECEIVERID = :OLD.USERID1);  
END;  
/
```

Result:

```
ORA-21000: error number argument to raise_application_error of -9 is out of range
```

```
ORA-06512: at "AMINA3RD.UNFRIEND", line 17
```

```
ORA-06512: at line 1
```

```
SQL> Execute unfriend(1,3);
```

```
PL/SQL procedure successfully completed.
```

```
SQL>
```

```
SQL> select * from Message;
```

SENDERID	RECEIVERID	TEXT
4	2	Give me the solution of todays task.
2	1	Jalil is asking for task again. Should I give it to him?
1	2	No.

```
SQL>
```

Task 5

For this task, I created two triggers, one for handling deletion and another for handling insertion.

Then I inserted data from helper file and generated output.

In the created procedure, I can generate error message for wrong input.

Below is the implementation of the procedure :

```
CREATE OR REPLACE TRIGGER DELETEPREVIOUSPOSTS BEFORE
  INSERT ON POST FOR EACH ROW
BEGIN
  DELETE FROM POSTNOTIFICATION
  WHERE
    (USERID = :NEW.USERID);
END;
/

CREATE OR REPLACE TRIGGER UPDATEPOSTNOTIFICATION AFTER
  INSERT ON POST FOR EACH ROW
BEGIN
  FOR FRNDRCD IN (
    SELECT
      USERID2
    FROM
      FRNDLIST
    WHERE
      USERID1 = :NEW.USERID
  ) LOOP
    INSERT INTO POSTNOTIFICATION (
      USERID,
      FRNDID,
      POSTID
    ) VALUES (
      :NEW.USERID,
      FRNDRCD.USERID2,
      :NEW.POSTID
    );
  END LOOP;
END;
/
```

Result:

```
SQL> SELECT
  2      *
  3  FROM
  4      POSTNOTIFICATION;

  USERID    FRNDID    POSTID
-----
         4         2         1

SQL> -- should show one row with post id as 1
SQL> INSERT INTO POST (
  2      POSTID,
  3      USERID,
  4      TEXT
  5  ) VALUES (
  6      2,
  7      4,
  8      'I need friends.'
  9  );

1 row created.

SQL> SELECT
  2      *
  3  FROM
  4      POSTNOTIFICATION;

  USERID    FRNDID    POSTID
-----
         4         2         2

SQL>
```


