

1. To identify an account is a seller or user from the login tables

**Program:**

```
DECLARE
    c_username login.username%type;
    c_password login.password%type;
    c_user_type login.user_type%type;

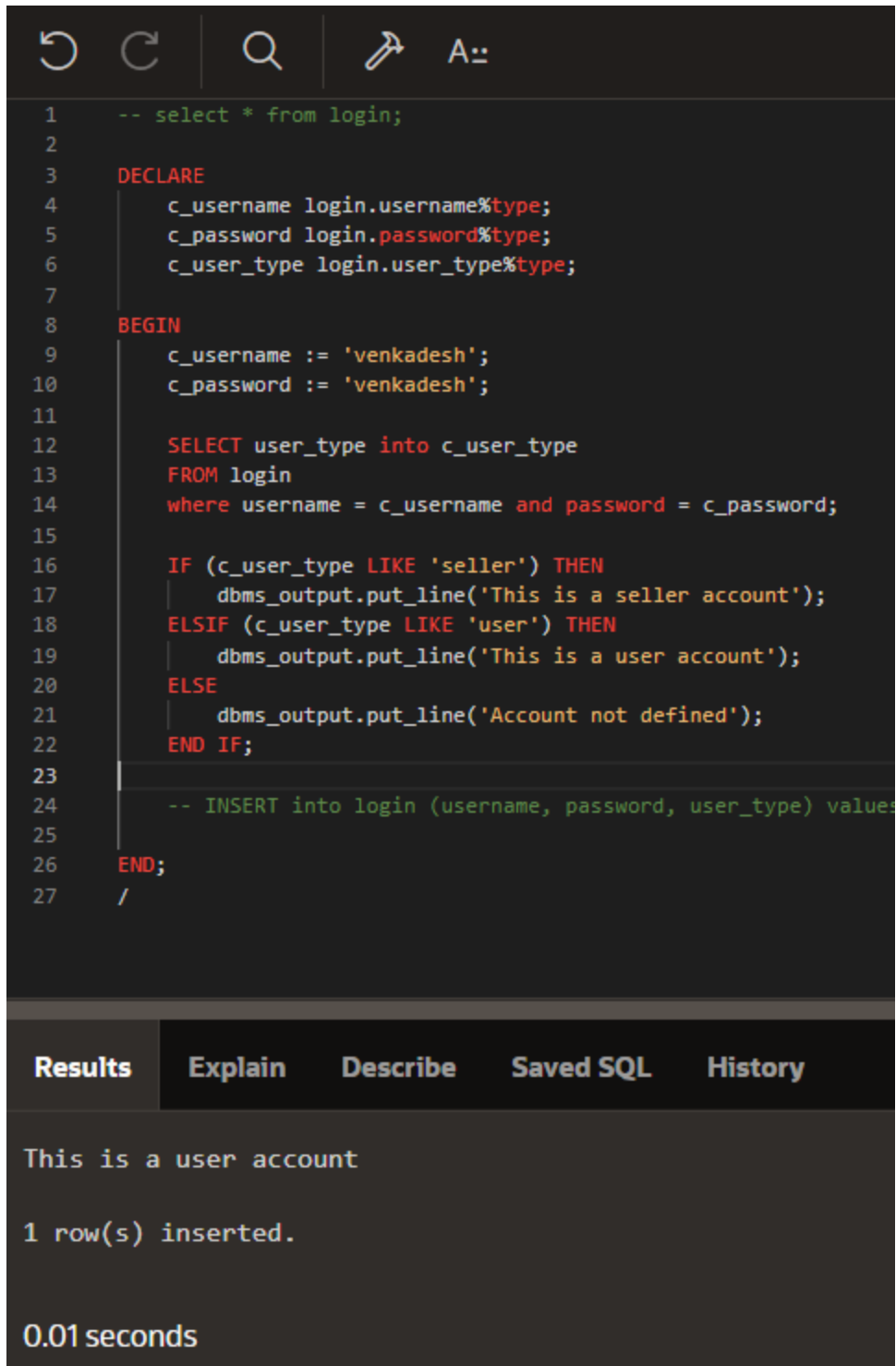
BEGIN
    c_username := 'venkadesh';
    c_password := 'venkadesh';

    SELECT user_type into c_user_type
    FROM login
    where username = c_username and password = c_password;

    IF (c_user_type LIKE 'seller') THEN
        dbms_output.put_line('This is a seller account');
    ELSIF (c_user_type LIKE 'user') THEN
        dbms_output.put_line('This is a user account');
    ELSE
        dbms_output.put_line('Account not defined');
    END IF;

END;
/
```

**Output:**



The image shows a SQL IDE interface with a dark theme. At the top is a toolbar with icons for undo, redo, search, and edit. Below the toolbar is a text editor containing a PL/SQL script. The script starts with a comment line, followed by a DECLARE section with three variables, a BEGIN section with assignments and a SELECT statement, an IF-ELSEIF-ELSE block for conditional output, and an END IF statement. Line 24 has a comment for an INSERT statement. The script ends with END; and a slash. Below the editor is a results pane with tabs for Results, Explain, Describe, Saved SQL, and History. The Results tab is active, showing the output 'This is a user account', '1 row(s) inserted.', and '0.01seconds'.

```
1  -- select * from login;
2
3  DECLARE
4      c_username login.username%type;
5      c_password login.password%type;
6      c_user_type login.user_type%type;
7
8  BEGIN
9      c_username := 'venkadesh';
10     c_password := 'venkadesh';
11
12     SELECT user_type into c_user_type
13     FROM login
14     where username = c_username and password = c_password;
15
16     IF (c_user_type LIKE 'seller') THEN
17         dbms_output.put_line('This is a seller account');
18     ELSEIF (c_user_type LIKE 'user') THEN
19         dbms_output.put_line('This is a user account');
20     ELSE
21         dbms_output.put_line('Account not defined');
22     END IF;
23
24     -- INSERT into login (username, password, user_type) values
25
26 END;
27 /
```

**Results**   Explain   Describe   Saved SQL   History

This is a user account

1 row(s) inserted.

0.01seconds

2. From the array of users find who the sellers and users

**Program:**

DECLARE

```

type c_username IS VARRAY(5) OF login.username%type;
users_list c_username;
total_count integer;
c_user_type login.user_type%type;

BEGIN
  users_list := c_username('venkadesh', 'rithick', 'sabarish', 'dinesh', 'prasanna');
  total_count := users_list.count;

  dbms_output.put_line('Total users: ' || total_count);

  FOR i in 1 .. total_count LOOP
    SELECT user_type INTO c_user_type
    FROM login
    WHERE username = users_list(i);

    IF (c_user_type = 'seller') THEN
      dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');
```

```

    ELSIF (c_user_type = 'user') THEN
      dbms_output.put_line('The user ' || users_list(i) || ' is a User.');
```

```

    ELSE
      dbms_output.put_line('The user ' || users_list(i) || ' is Unknown.');
```

```

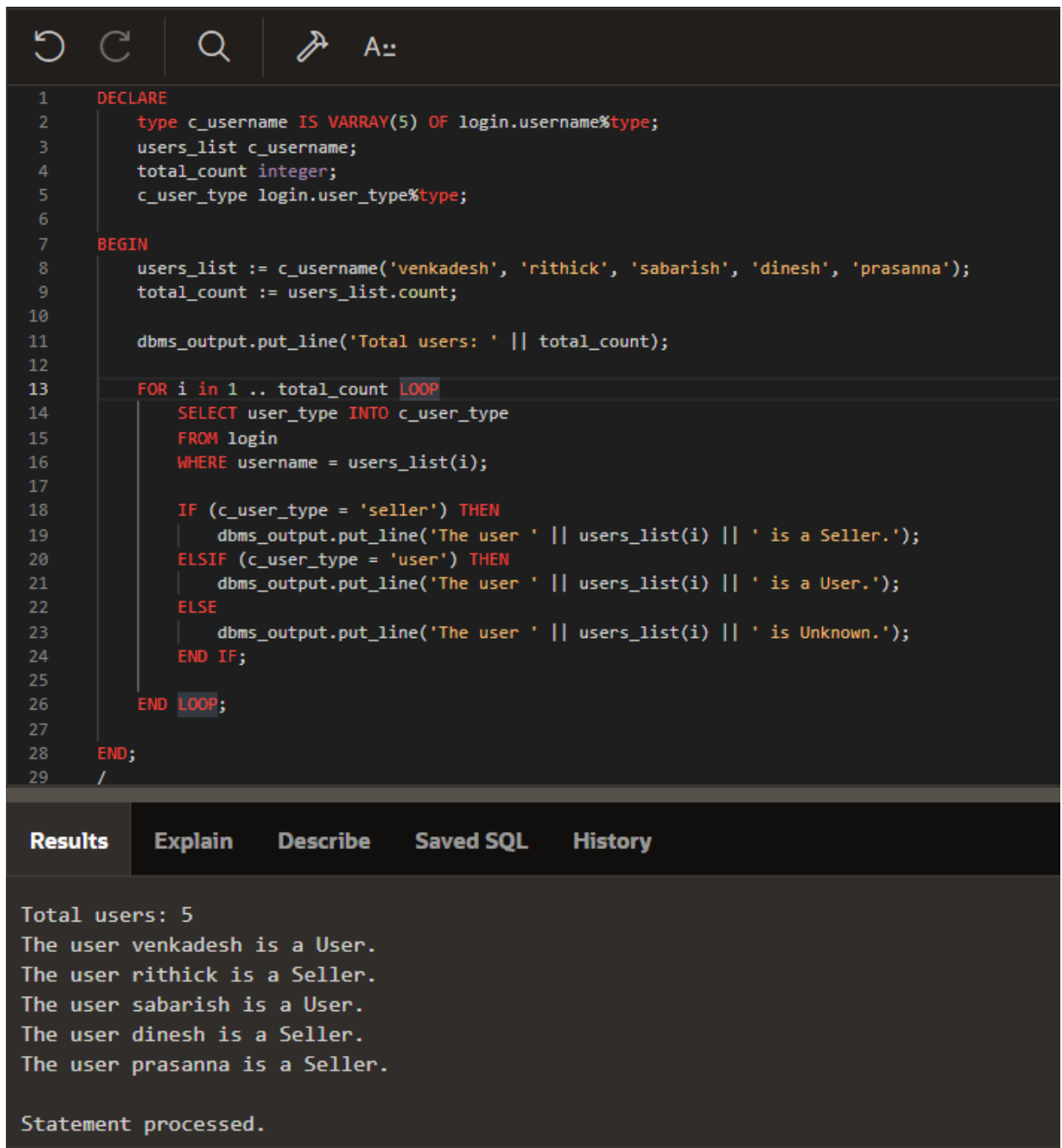
    END IF;

  END LOOP;

END;
/

```

**Output:**



The screenshot shows a SQL IDE with a dark theme. At the top, there is a toolbar with icons for undo, redo, search, and a command prompt. Below the toolbar, a PL/SQL script is written in a text editor. The script declares a variable `c_username` as a VARRAY of type `login.username%type`, initializes it with five usernames, and then loops through each username to determine the user type from the `login` table. The results are printed using `dbms_output.put_line`. The script ends with `END;` and a forward slash `/`.

```
1 DECLARE
2     type c_username IS VARRAY(5) OF login.username%type;
3     users_list c_username;
4     total_count integer;
5     c_user_type login.user_type%type;
6
7 BEGIN
8     users_list := c_username('venkadesh', 'rithick', 'sabarish', 'dinesh', 'prasanna');
9     total_count := users_list.count;
10
11     dbms_output.put_line('Total users: ' || total_count);
12
13     FOR i in 1 .. total_count LOOP
14         SELECT user_type INTO c_user_type
15         FROM login
16         WHERE username = users_list(i);
17
18         IF (c_user_type = 'seller') THEN
19             dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');

Below the script, there is a tabbed interface with five tabs: Results, Explain, Describe, Saved SQL, and History. The Results tab is selected, showing the output of the script:



```
Total users: 5
The user venkadesh is a User.
The user rithick is a Seller.
The user sabarish is a User.
The user dinesh is a Seller.
The user prasanna is a Seller.

Statement processed.
```


```

3. From the array of users find those who are users and print their full name and email from the users table

**Program:**

```
DECLARE
    type c_username IS VARRAY(5) OF login.username%type;
    users_list c_username;
```

```

total_count integer;
c_user_type login.user_type%type;

fullname users.name%type;
user_email users.email%type;

BEGIN
  users_list := c_username('venkadesh', 'rithick', 'sabarish', 'dinesh', 'prasanna');
  total_count := users_list.count;

  dbms_output.put_line('Total users: ' || total_count);

  FOR i in 1 .. total_count LOOP
    SELECT user_type INTO c_user_type
    FROM login
    WHERE username = users_list(i);

    IF (c_user_type = 'seller') THEN
      dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');
```

```

    ELSIF (c_user_type = 'user') THEN
      dbms_output.put_line('The user ' || users_list(i) || ' is a User.');
```

```

      SELECT name, email INTO fullname, user_email
      FROM users
      where username = users_list(i);

      dbms_output.put_line('The user Full name is ' || fullname || ', and his email is ' ||
user_email);
    ELSE
      dbms_output.put_line('The user ' || users_list(i) || ' is Unknown.');
```

```

    END IF;

  END LOOP;

END;
/

```

**Output:**

↶ ↷ 🔍 📌 A::

```

1 DECLARE
2     type c_username IS VARRAY(5) OF login.username%type;
3     users_list c_username;
4     total_count integer;
5     c_user_type login.user_type%type;
6     fullname users.name%type;
7     user_email users.email%type;
8 BEGIN
9     users_list := c_username('venkadesh', 'rithick', 'sabarish', 'dinesh', 'prasanna');
10    total_count := users_list.count;
11    dbms_output.put_line('Total users: ' || total_count);
12    FOR i in 1 .. total_count LOOP
13        SELECT user_type INTO c_user_type
14        FROM login
15        WHERE username = users_list(i);
16
17        IF (c_user_type = 'seller') THEN
18            dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');

Results
Explain
Describe
Saved SQL
History



```

Total users: 5
The user venkadesh is a User.
The user Full name is venkadesh, and his email is venkadesh@gmail.com
The user rithick is a Seller.
The user sabarish is a User.
The user Full name is Sabarish, and his email is sabari@gmail.com
The user dinesh is a Seller.
The user prasanna is a Seller.

```


```

4. Convert the following above program with the help of Procedures

**Program:**

```

DECLARE
type c_username IS VARRAY(5) OF login.username%type;
users_list c_username;
total_count integer;

```

```

    c_user_type login.user_type%type;
    fullname users.name%type;
    user_email users.email%type;

PROCEDURE findType(username_val IN login.username%type, c_user_type OUT
login.user_type%type) IS
BEGIN
    SELECT user_type INTO c_user_type
    FROM login
    WHERE username = username_val;
END;

PROCEDURE getData(username_val IN login.username%type) IS
BEGIN
    SELECT name, email INTO fullname, user_email
    FROM users
    where username = username_val;
    dbms_output.put_line('The user Full name is ' || fullname || ', and his email is ' ||
user_email);
END;

BEGIN
    users_list := c_username('venkadesh', 'rithick', 'sabarish', 'dinesh', 'prasanna');
    total_count := users_list.count;
    dbms_output.put_line('Total users: ' || total_count);
    FOR i in 1 .. total_count LOOP
        findType(users_list(i), c_user_type);

        IF (c_user_type = 'seller') THEN
            dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');
```

```

        ELSIF (c_user_type = 'user') THEN
            dbms_output.put_line('The user ' || users_list(i) || ' is a User.');
```

```

            getData(users_list(i));
        ELSE
            dbms_output.put_line('The user ' || users_list(i) || ' is Unknown.');
```

```

        END IF;

    END LOOP;

END;
/

```

**Output:**

```
1 DECLARE
2     type c_username IS VARRAY(5) OF login.username%type;
3     users_list c_username;
4     total_count integer;
5     c_user_type login.user_type%type;
6     fullname users.name%type;
7     user_email users.email%type;
8
9 PROCEDURE findType(username_val IN login.username%type, c_user_type OUT login.user_type%type) IS
10 BEGIN
11     SELECT user_type INTO c_user_type
12     FROM login
13     WHERE username = username_val;
14 END;
15
16 PROCEDURE getData(username_val IN login.username%type) IS
17 BEGIN
18     SELECT name, email INTO fullname, user_email
19     FROM users
20     where username = username_val;
21     dbms_output.put_line('The user Full name is ' || fullname || ', and his email is ' || user_email);
22 END;
23
24 BEGIN
25     users_list := c_username('venkadesh', 'rithick', 'sabarish', 'dinesh', 'prasanna');
26     total_count := users_list.count;
27     dbms_output.put_line('Total users: ' || total_count);
28     FOR i in 1 .. total_count LOOP
29         findType(users_list(i), c_user_type);
30
31         IF (c_user_type = 'seller') THEN
32             dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');
```

5. To fetch the users in the login table using Cursors

### Program:

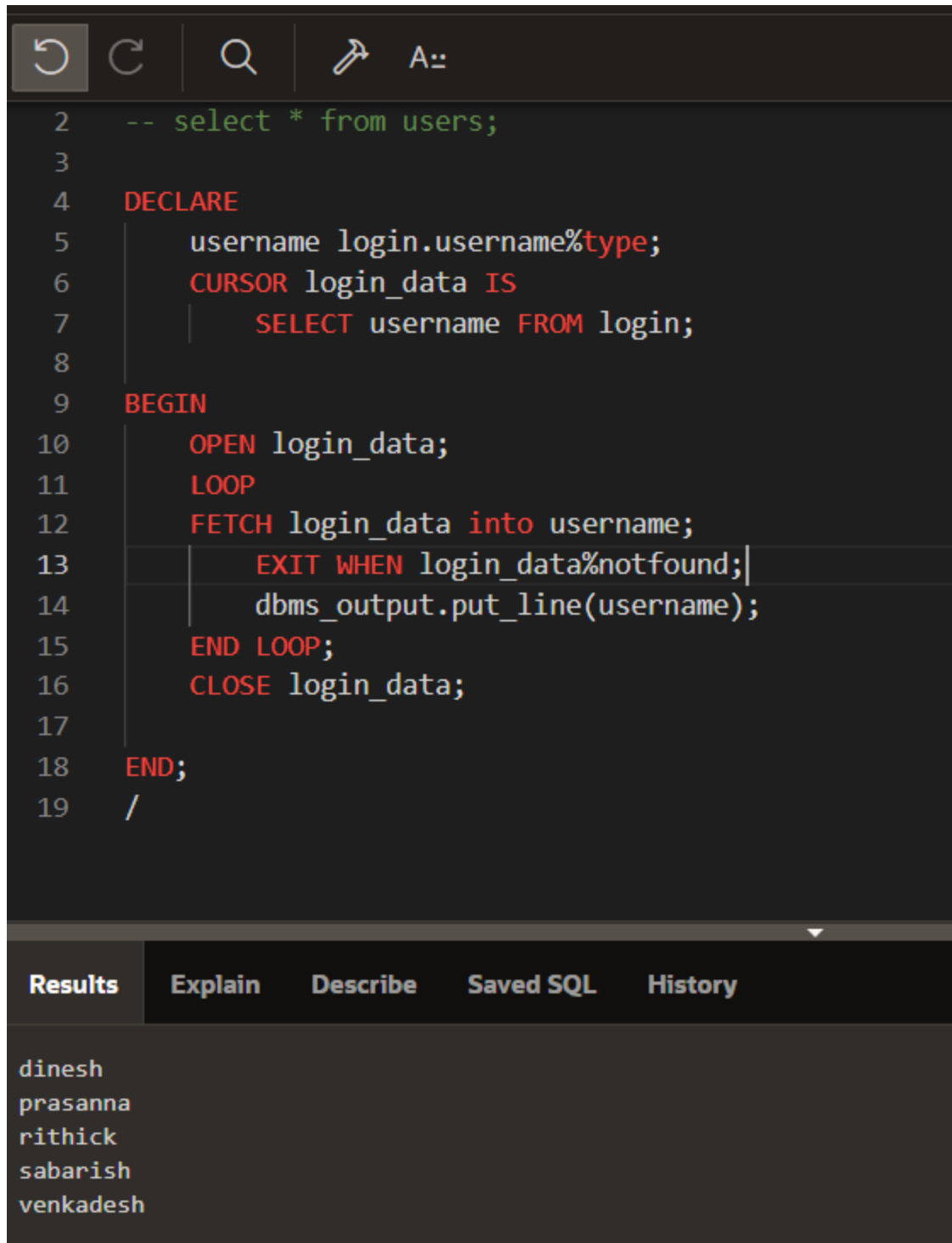
```
DECLARE
    username login.username%type;
    CURSOR login_data IS
        SELECT username FROM login;

BEGIN
```



```
OPEN login_data;  
LOOP  
  FETCH login_data into username;  
    EXIT WHEN login_data%notfound;  
    dbms_output.put_line(username);  
END LOOP;  
CLOSE login_data;  
  
END;  
/
```

**Output:**



The image shows a screenshot of a SQL IDE. The top toolbar contains icons for undo, redo, search, and save, along with a text input field containing 'A:'. The main editor area displays a PL/SQL program with line numbers 2 through 19. The program includes a comment, a DECLARE section for a cursor and a variable, a BEGIN section with an OPEN, LOOP, FETCH, EXIT WHEN, dbms\_output.put\_line, END LOOP, and CLOSE statements, and an END; statement. The bottom panel shows the 'Results' tab with a list of usernames: dinesh, prasanna, rithick, sabarish, and venkadesh.

```
2  -- select * from users;
3
4  DECLARE
5      username login.username%type;
6      CURSOR login_data IS
7          SELECT username FROM login;
8
9  BEGIN
10     OPEN login_data;
11     LOOP
12         FETCH login_data into username;
13         EXIT WHEN login_data%notfound;
14         dbms_output.put_line(username);
15     END LOOP;
16     CLOSE login_data;
17
18 END;
19 /
```

**Results**   Explain   Describe   Saved SQL   History

dinesh  
prasanna  
rithick  
sabarish  
venkadesh

6. For the 4th problem handle the exception on no data found

**Program:**

DECLARE

```

type c_username IS VARRAY(5) OF login.username%type;
users_list c_username;
total_count integer;
c_user_type login.user_type%type;
fullname users.name%type;
user_email users.email%type;

```

```

PROCEDURE getData(username_val IN login.username%type) IS
BEGIN
    SELECT name, email INTO fullname, user_email
    FROM users
    where username = username_val;
    dbms_output.put_line('The user Full name is ' || fullname || ', and his email is ' || user_email);
END;

```

```

BEGIN
    users_list := c_username('venkadesh', 'rithick', 'sabarish', 'dinesh', 'new');
    total_count := users_list.count;
    dbms_output.put_line('Total users: ' || total_count);
    FOR i in 1 .. total_count LOOP
        dbms_output.put_line(users_list(i));

        SELECT user_type INTO c_user_type
        FROM login
        WHERE username = users_list(i);

        IF (c_user_type = 'seller') THEN
            dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');
```

```

        ELSIF (c_user_type = 'user') THEN
            dbms_output.put_line('The user ' || users_list(i) || ' is a User.');
```

```

            getData(users_list(i));
        ELSE
            dbms_output.put_line('The user ' || users_list(i) || ' is Unknown.');
```

```

        END IF;
    END LOOP;

```

```

EXCEPTION
    WHEN no_data_found THEN
        dbms_output.put_line('Data not found');
    WHEN others THEN
        dbms_output.put_line('Error!');

```

```

END;
/

```

## Output:

```
18 users_list := c_username('venkatesh', 'richick', 'sudarshan', 'ganesan', new);
19 total_count := users_list.count;
20 dbms_output.put_line('Total users: ' || total_count);
21 FOR i in 1 .. total_count LOOP
22     dbms_output.put_line(users_list(i));
23
24     SELECT user_type INTO c_user_type
25     FROM login
26     WHERE username = users_list(i);
27
28     IF (c_user_type = 'seller') THEN
29         dbms_output.put_line('The user ' || users_list(i) || ' is a Seller.');
```

```
30     ELSIF (c_user_type = 'user') THEN
31         dbms_output.put_line('The user ' || users_list(i) || ' is a User.');
```

```
32         getData(users_list(i));
33     ELSE
34         dbms_output.put_line('The user ' || users_list(i) || ' is Unknown.');
```

```
35     END IF;
36 END LOOP;
37
38 EXCEPTION
39     WHEN no_data_found THEN
40         dbms_output.put_line('Data not found');
```

```
41     WHEN others THEN
42         dbms_output.put_line('Error!');
```

```
43
44 END;
45 /
```

Results	Explain	Describe	Saved SQL	History
new				
Data not found				
Statement processed.				
0.01 seconds				