Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavrepalanchowk

A lab report on

"Lab 05"

Sub Code: COMP 232

Submitted by:

Rohan Dhakal (Roll No. 14)

Submitted to:

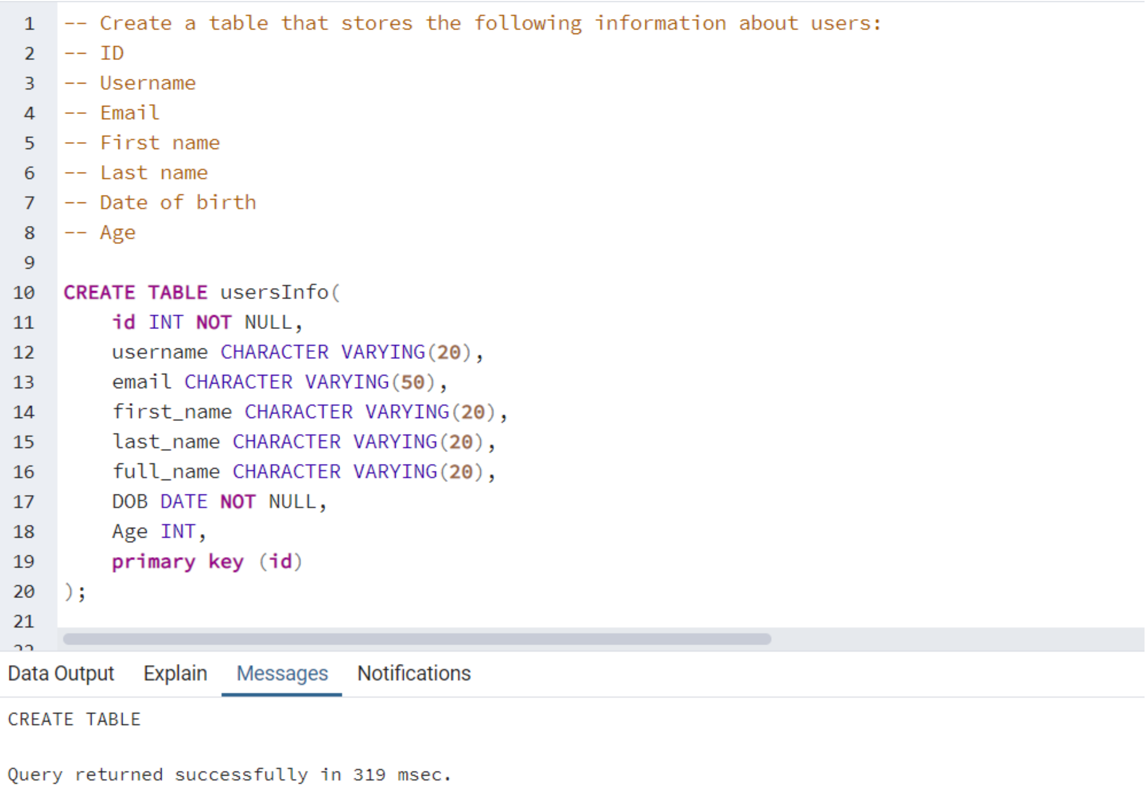
Dr. Rajani Chulyadyo

Department of Computer Science and Engineering

Submission Date: 2022/04/19

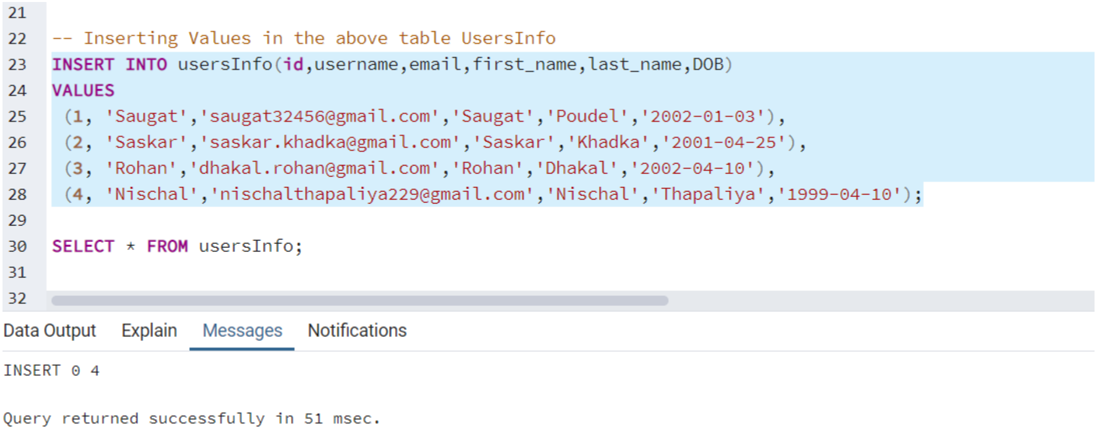
Tasks

1. Create a table that stores the following information about users:
    1. ID
    2. Username
    3. Email
    4. First name
    5. Last name
    6. Date of birth
    7. Age
2. Create a table named user_logs with the following columns:
    1. ID
    2. Old_value
    3. New_value
    4. Description
    5. Log_time
3. Create the following functions
    1. Function that returns the full name of the user with the given ID
    2. Function that returns the number of users
    3. Function that returns the age of the user with the given ID
4. Create the following stored procedures
    1. SP to update the full name of the user with the given ID
    2. SP to update the age of the user with the given ID
5. Create the following triggers
    1. Trigger that populates full name on adding a new user
    2. Trigger that populates age on adding a new user
    3. Trigger that inserts a new row in user_logs if any value is updated in users table. If last name of a user is updated, the following values must be inserted into the user_logs table:
       <old last name>, <new last name>, 'Last name updated', current time



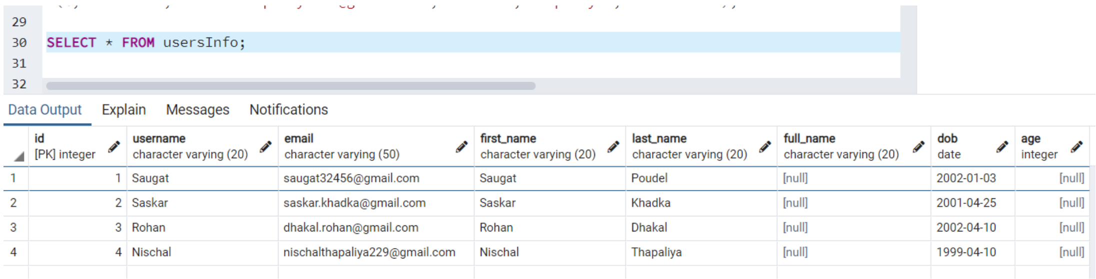
```sql
-- Create a table that stores the following information about users:
-- ID
-- Username
-- Email
-- First name
-- Last name
-- Date of birth
-- Age

CREATE TABLE usersInfo(
    id INT NOT NULL,
    username CHARACTER VARYING(20),
    email CHARACTER VARYING(50),
    first_name CHARACTER VARYING(20),
    last_name CHARACTER VARYING(20),
    full_name CHARACTER VARYING(20),
    DOB DATE NOT NULL,
    Age INT,
    primary key (id)
);

```

```
CREATE TABLE

Query returned successfully in 319 msec.
```



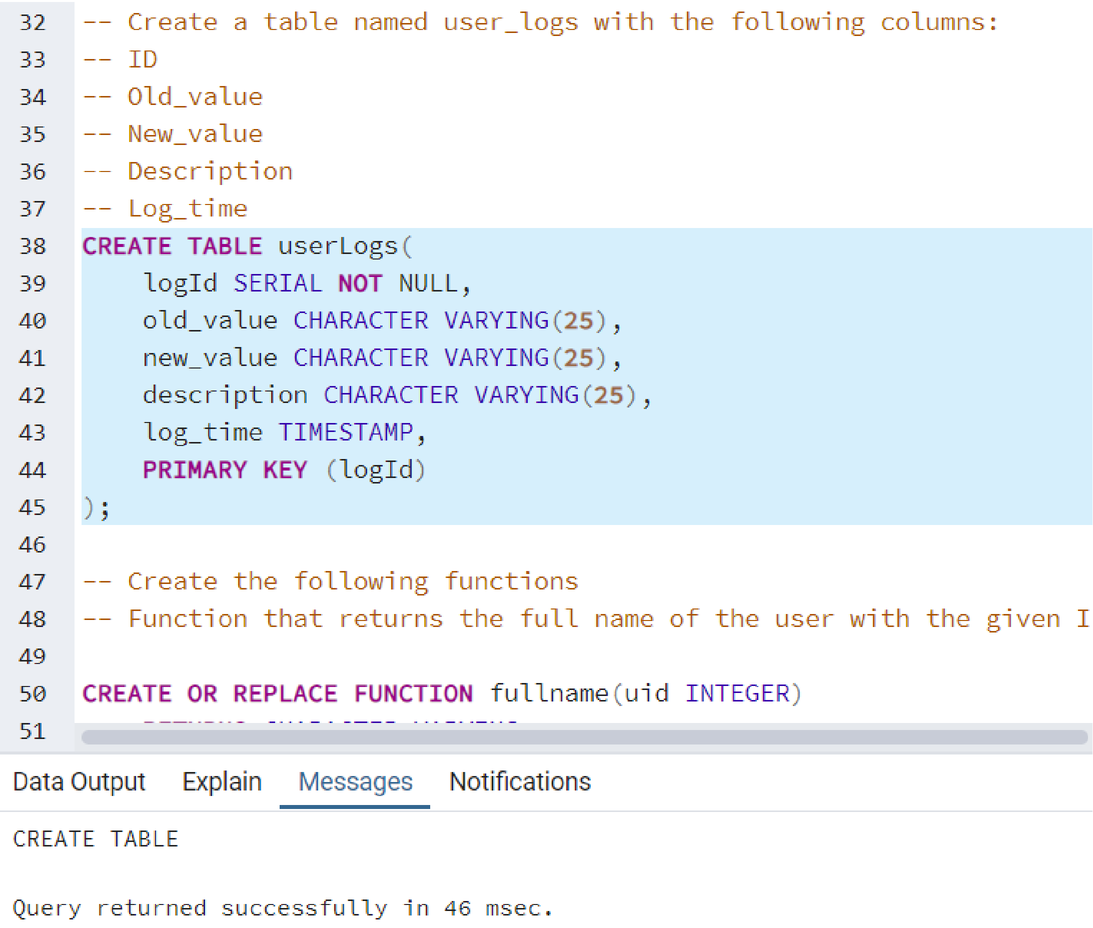
```sql

-- Inserting Values in the above table UsersInfo
INSERT INTO usersInfo(id,username,email,first_name,last_name,DOB)
VALUES
 (1, 'Saugat','saugat32456@gmail.com','Saugat','Poudel','2002-01-03'),
 (2, 'Saskar','saskar.khadka@gmail.com','Saskar','Khadka','2001-04-25'),
 (3, 'Rohan','dhakal.rohan@gmail.com','Rohan','Dhakal','2002-04-10'),
 (4, 'Nischal','nischalthapaliya229@gmail.com','Nischal','Thapaliya','1999-04-10');

SELECT * FROM usersInfo;

```

```
INSERT 0 4

Query returned successfully in 51 msec.
```



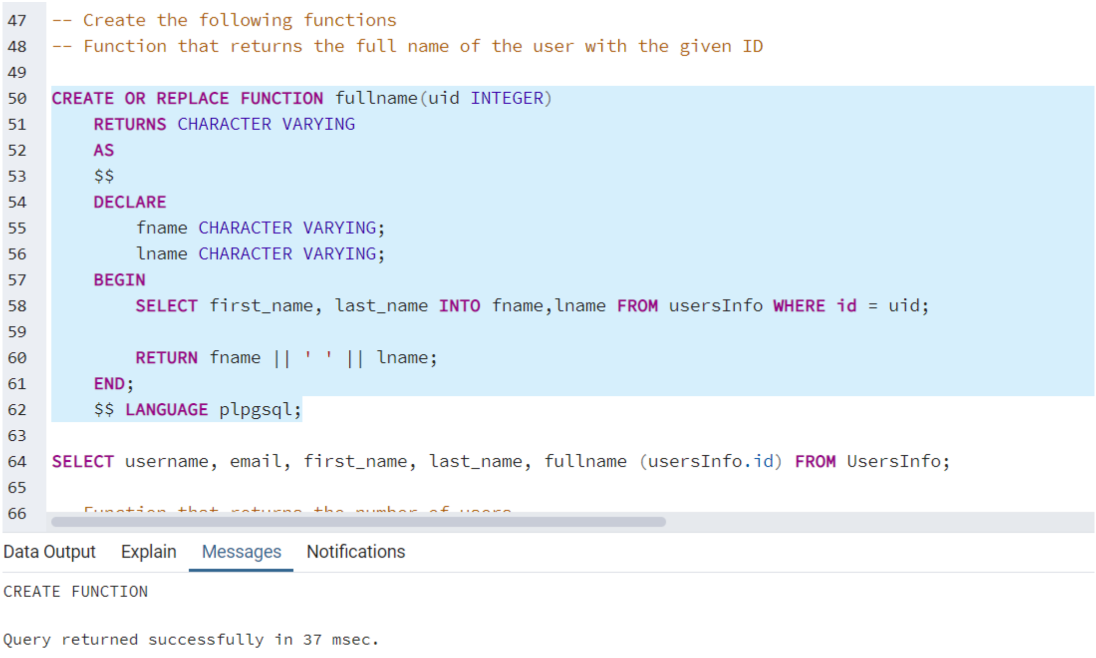
```sql
SELECT * FROM usersInfo;
```

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	full_name character varying (20)	dob date	age integer
1	1	Saugat	saugat32456@gmail.com	Saugat	Poudel	[null]	2002-01-03	[null]
2	2	Saskar	saskar.khadka@gmail.com	Saskar	Khadka	[null]	2001-04-25	[null]
3	3	Rohan	dhakal.rohan@gmail.com	Rohan	Dhakal	[null]	2002-04-10	[null]
4	4	Nischal	nischalthapaliya229@gmail.com	Nischal	Thapaliya	[null]	1999-04-10	[null]



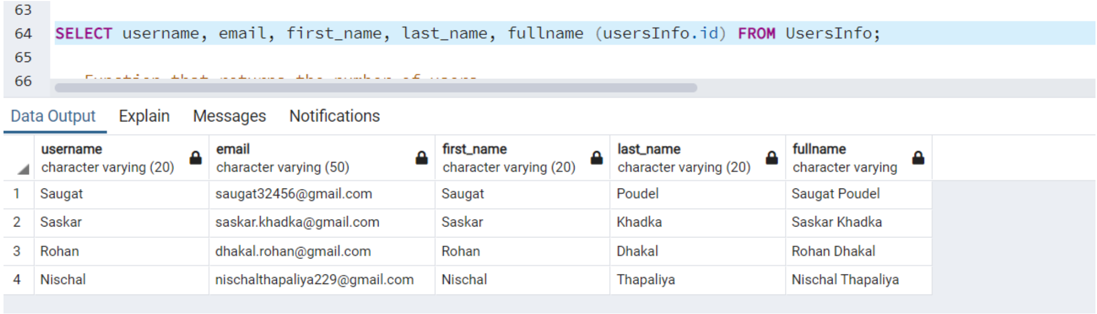
```sql
-- Create a table named user_logs with the following columns:
-- ID
-- Old_value
-- New_value
-- Description
-- Log_time
CREATE TABLE userLogs(
    logId SERIAL NOT NULL,
    old_value CHARACTER VARYING(25),
    new_value CHARACTER VARYING(25),
    description CHARACTER VARYING(25),
    log_time TIMESTAMP,
    PRIMARY KEY (logId)
);

-- Create the following functions
-- Function that returns the full name of the user with the given I

CREATE OR REPLACE FUNCTION fullname(uid INTEGER)
```

```
CREATE TABLE

Query returned successfully in 46 msec.
```



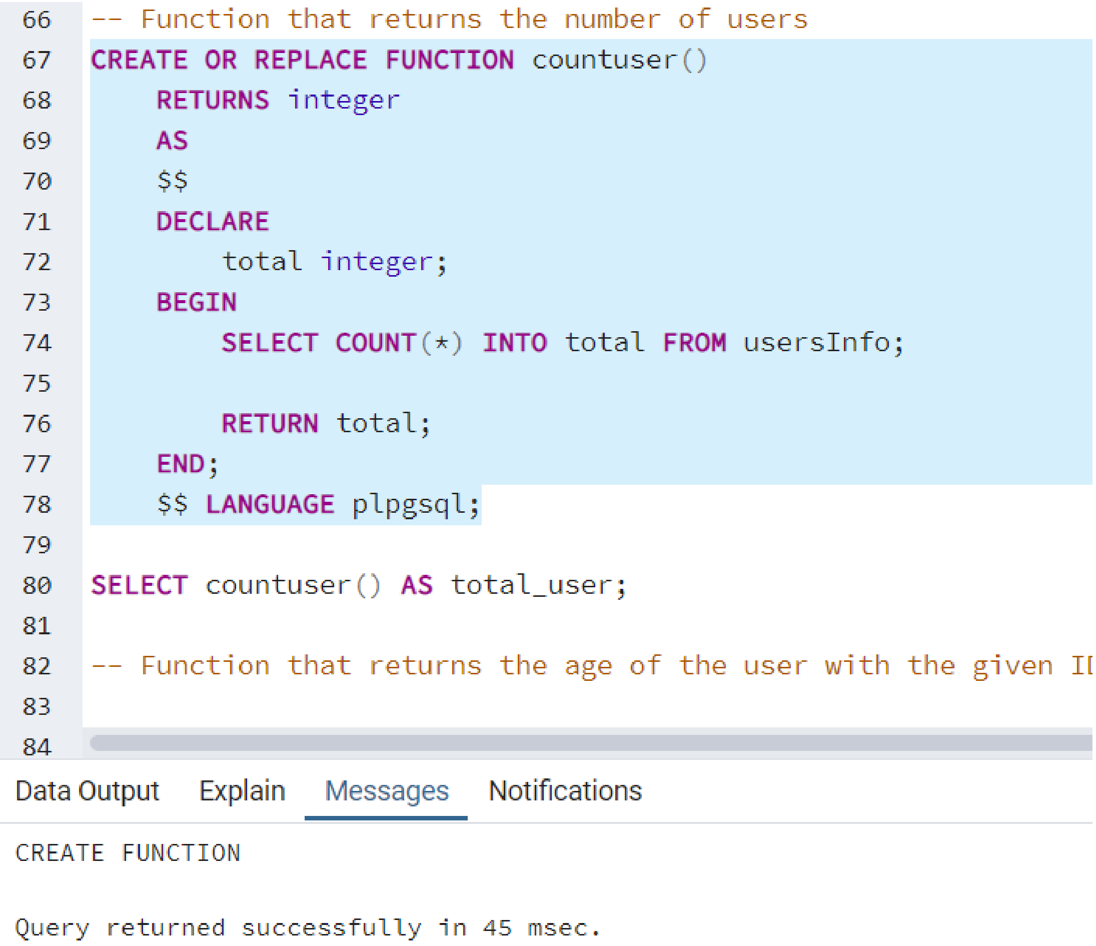
```sql
-- Create the following functions
-- Function that returns the full name of the user with the given ID

CREATE OR REPLACE FUNCTION fullname(uid INTEGER)
    RETURNS CHARACTER VARYING
    AS
    $$
    DECLARE
        fname CHARACTER VARYING;
        lname CHARACTER VARYING;
    BEGIN
        SELECT first_name, last_name INTO fname,lname FROM usersInfo WHERE id = uid;

        RETURN fname || ' ' || lname;
    END;
    $$ LANGUAGE plpgsql;

SELECT username, email, first_name, last_name, fullname (usersInfo.id) FROM UsersInfo;

    -- Function that returns the number of users
```

```
CREATE FUNCTION

Query returned successfully in 37 msec.
```



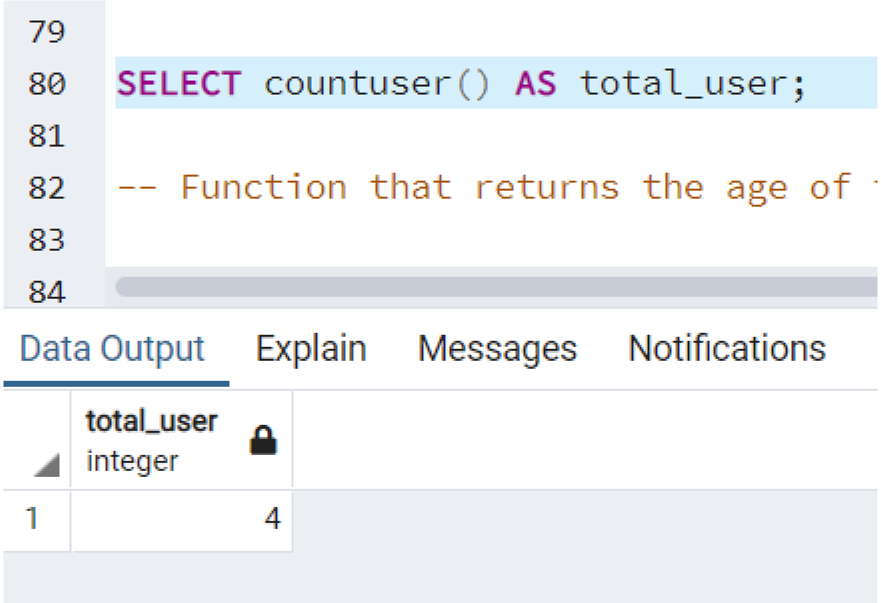
```sql

SELECT username, email, first_name, last_name, fullname (usersInfo.id) FROM UsersInfo;

    -- Function that returns the number of users
```

	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	fullname character varying
1	Saugat	saugat32456@gmail.com	Saugat	Poudel	Saugat Poudel
2	Saskar	saskar.khadka@gmail.com	Saskar	Khadka	Saskar Khadka
3	Rohan	dhakal.rohan@gmail.com	Rohan	Dhakal	Rohan Dhakal
4	Nischal	nischalthapaliya229@gmail.com	Nischal	Thapaliya	Nischal Thapaliya



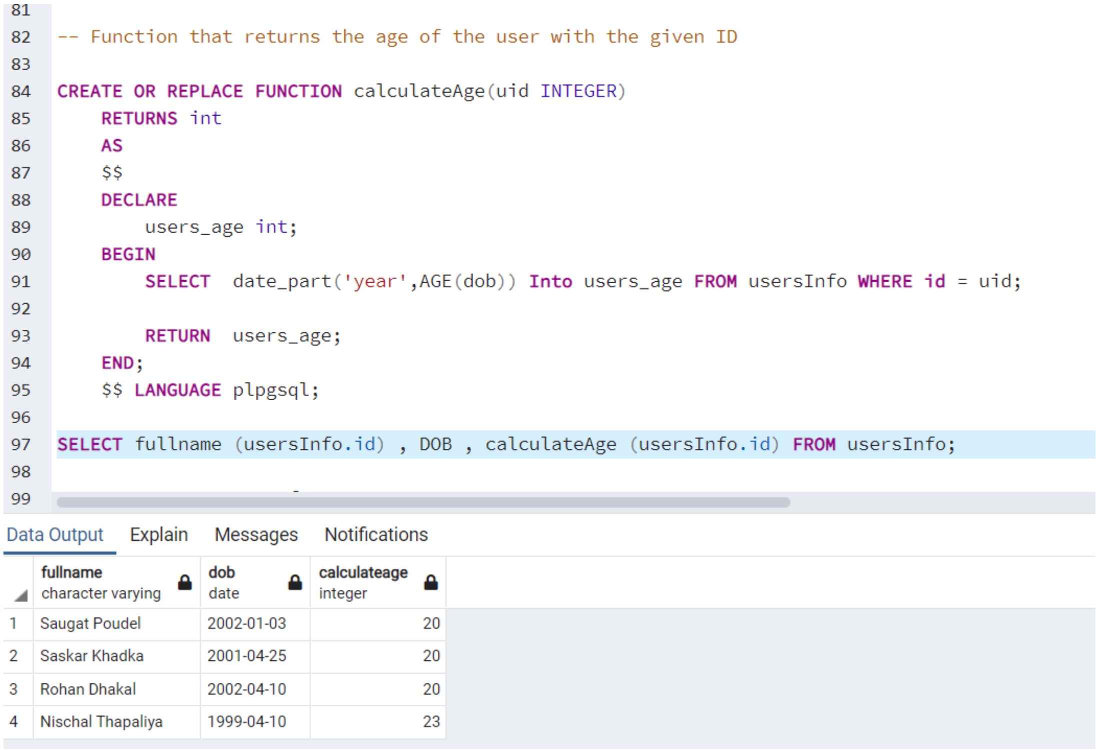
```
-- Function that returns the number of users
CREATE OR REPLACE FUNCTION countuser()
    RETURNS integer
    AS
    $$
    DECLARE
        total integer;
    BEGIN
        SELECT COUNT(*) INTO total FROM usersInfo;

        RETURN total;
    END;
    $$ LANGUAGE plpgsql;

SELECT countuser() AS total_user;

-- Function that returns the age of the user with the given ID
```

```
CREATE FUNCTION

Query returned successfully in 45 msec.
```



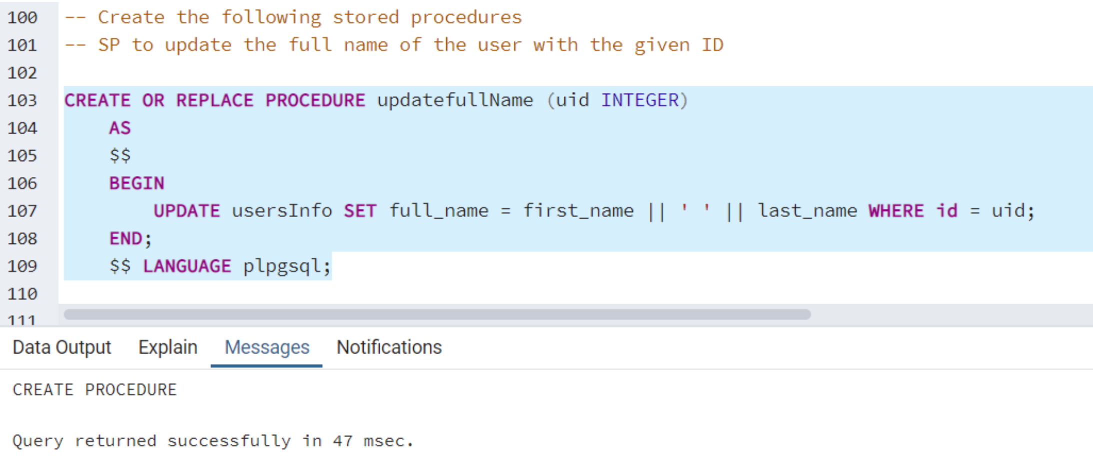
```sql
SELECT countuser() AS total_user;

-- Function that returns the age of
```

	total_user integer
1	4



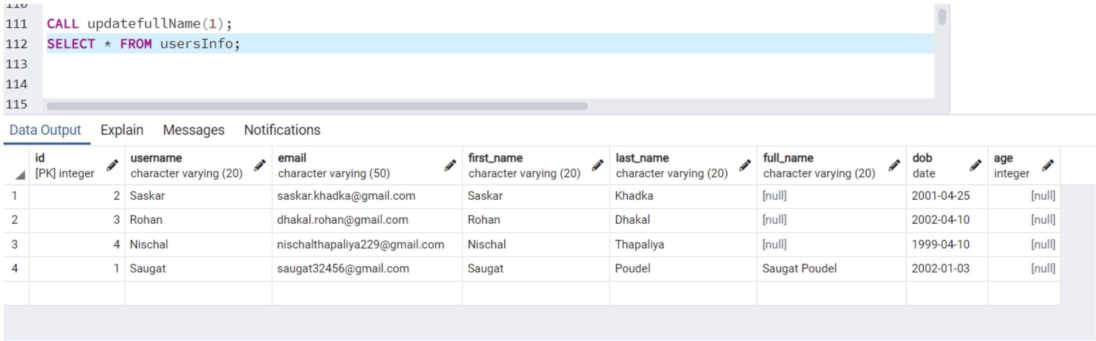
```sql
-- Function that returns the age of the user with the given ID

CREATE OR REPLACE FUNCTION calculateAge(uid INTEGER)
    RETURNS int
    AS
    $$
    DECLARE
        users_age int;
    BEGIN
        SELECT  date_part('year',AGE(dob)) Into users_age FROM usersInfo WHERE id = uid;

        RETURN  users_age;
    END;
    $$ LANGUAGE plpgsql;

SELECT fullname (usersInfo.id) , DOB , calculateAge (usersInfo.id) FROM usersInfo;
```

	fullname character varying	dob date	calculateage integer
1	Saugat Poudel	2002-01-03	20
2	Saskar Khadka	2001-04-25	20
3	Rohan Dhakal	2002-04-10	20
4	Nischal Thapaliya	1999-04-10	23



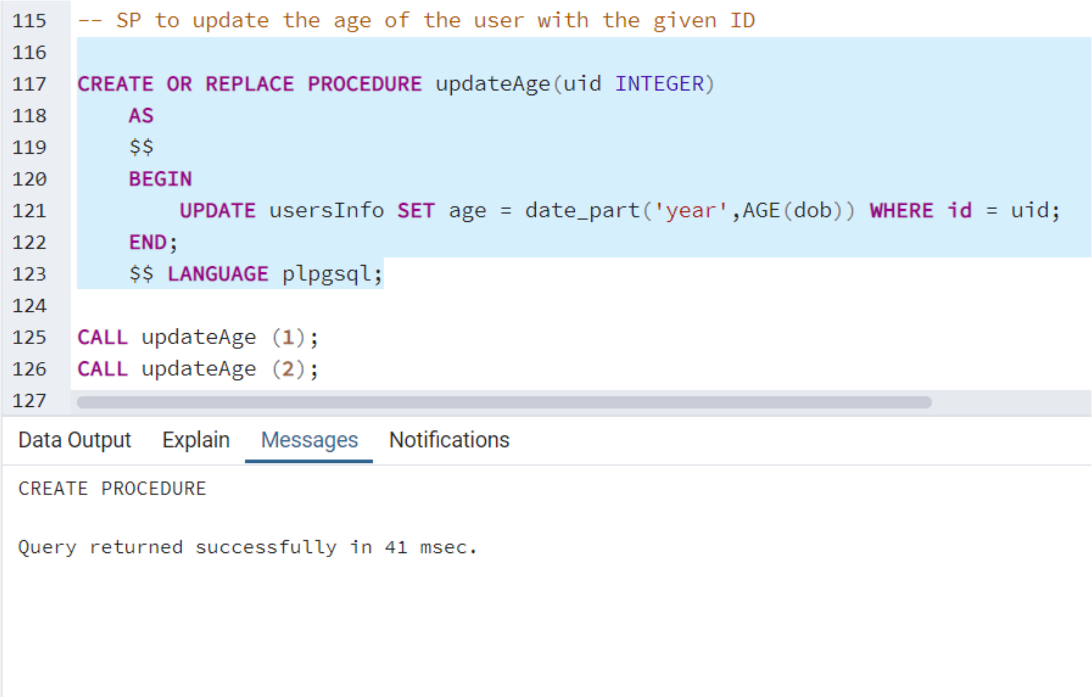
```
-- Create the following stored procedures
-- SP to update the full name of the user with the given ID

CREATE OR REPLACE PROCEDURE updatefullName (uid INTEGER)
    AS
    $$
    BEGIN
        UPDATE usersInfo SET full_name = first_name || ' ' || last_name WHERE id = uid;
    END;
    $$ LANGUAGE plpgsql;
```

```
CREATE PROCEDURE

Query returned successfully in 47 msec.
```



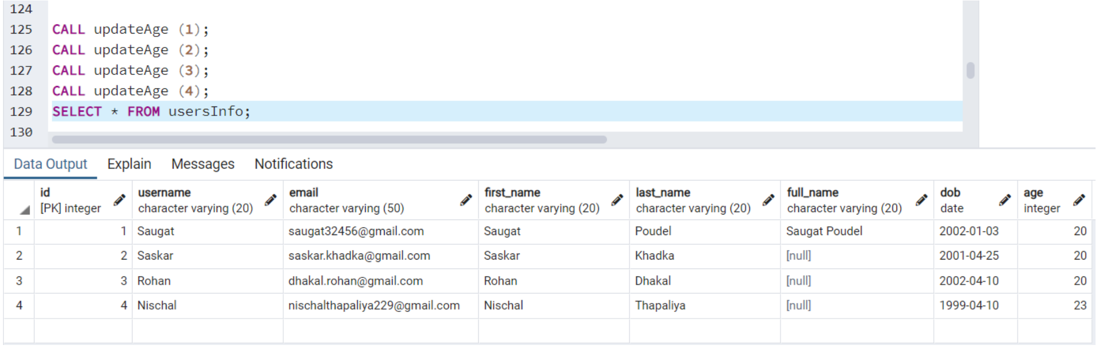
```
CALL updatefullName(1);
SELECT * FROM usersInfo;
```

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	full_name character varying (20)	dob date	age integer
1	2	Saskar	saskar.khadka@gmail.com	Saskar	Khadka	[null]	2001-04-25	[null]
2	3	Rohan	dhakal.rohan@gmail.com	Rohan	Dhakal	[null]	2002-04-10	[null]
3	4	Nischal	nischalthapaliya229@gmail.com	Nischal	Thapaliya	[null]	1999-04-10	[null]
4	1	Saugat	saugat32456@gmail.com	Saugat	Poudel	Saugat Poudel	2002-01-03	[null]



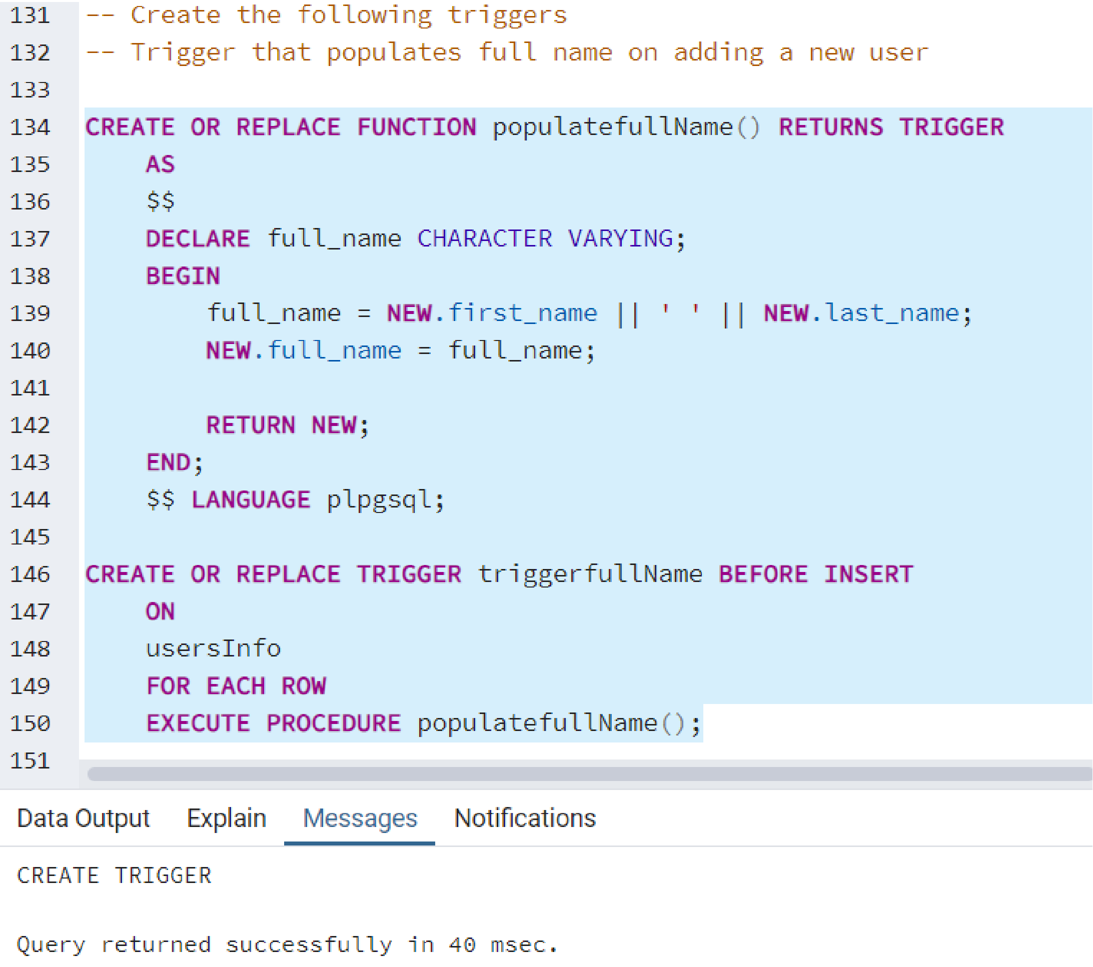
```
-- SP to update the age of the user with the given ID

CREATE OR REPLACE PROCEDURE updateAge(uid INTEGER)
    AS
    $$
    BEGIN
        UPDATE usersInfo SET age = date_part('year',AGE(dob)) WHERE id = uid;
    END;
    $$ LANGUAGE plpgsql;

CALL updateAge (1);
CALL updateAge (2);

```

```
CREATE PROCEDURE

Query returned successfully in 41 msec.
```



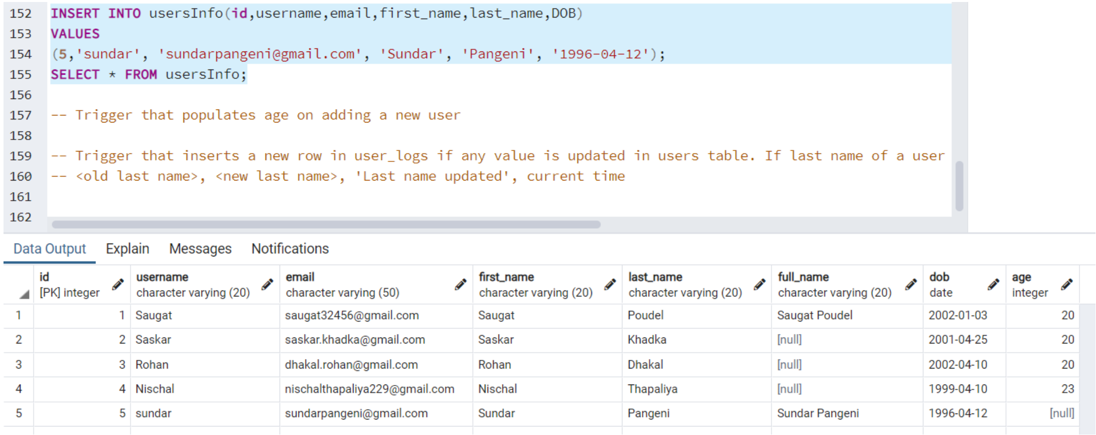
```

CALL updateAge (1);
CALL updateAge (2);
CALL updateAge (3);
CALL updateAge (4);
SELECT * FROM usersInfo;

```

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	full_name character varying (20)	dob date	age integer
1	1	Saugat	saugat32456@gmail.com	Saugat	Poudel	Saugat Poudel	2002-01-03	20
2	2	Saskar	saskar.khadka@gmail.com	Saskar	Khadka	[null]	2001-04-25	20
3	3	Rohan	dhakal.rohan@gmail.com	Rohan	Dhakal	[null]	2002-04-10	20
4	4	Nischal	nischalthapaliya229@gmail.com	Nischal	Thapaliya	[null]	1999-04-10	23



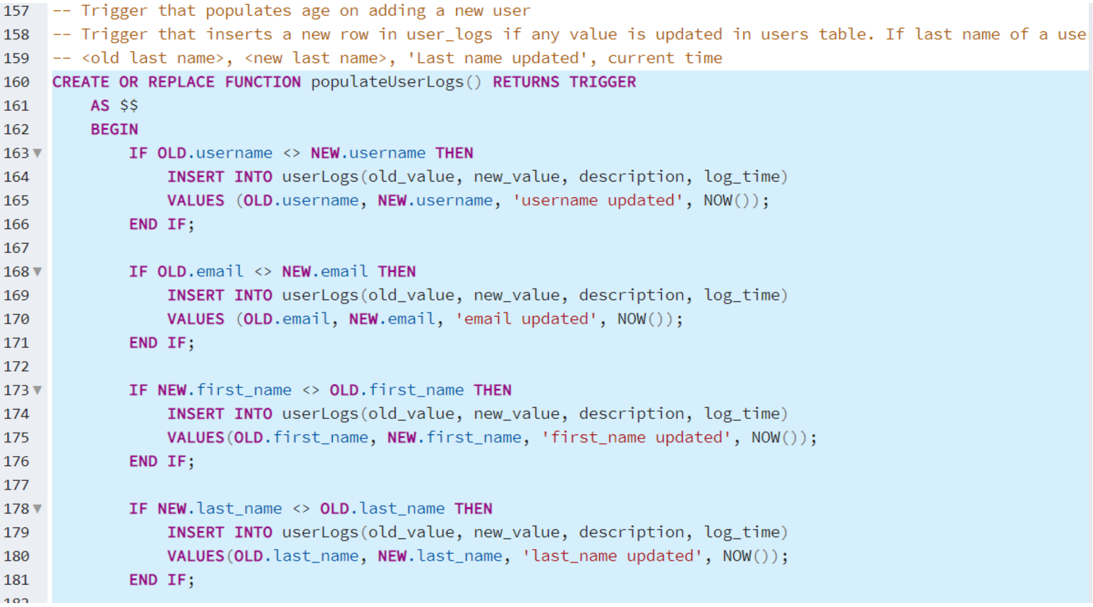
```sql
-- Create the following triggers
-- Trigger that populates full name on adding a new user

CREATE OR REPLACE FUNCTION populatefullName() RETURNS TRIGGER
    AS
    $$
    DECLARE full_name CHARACTER VARYING;
    BEGIN
        full_name = NEW.first_name || ' ' || NEW.last_name;
        NEW.full_name = full_name;

        RETURN NEW;
    END;
    $$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER triggerfullName BEFORE INSERT
    ON
    usersInfo
    FOR EACH ROW
    EXECUTE PROCEDURE populatefullName();

```

```
CREATE TRIGGER

Query returned successfully in 40 msec.
```

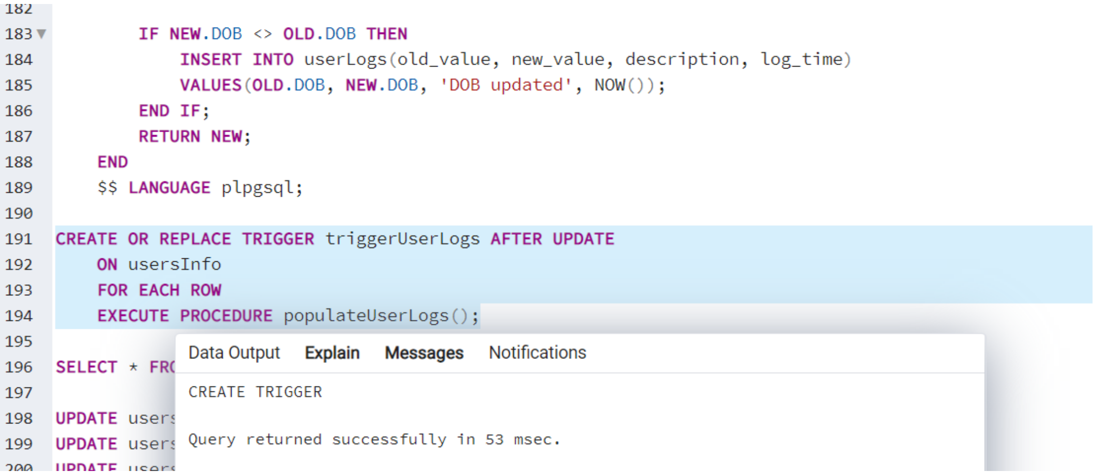


```sql
INSERT INTO usersInfo(id,username,email,first_name,last_name,DOB)
VALUES
(5,'sundar', 'sundarpangeni@gmail.com', 'Sundar', 'Pangeni', '1996-04-12');
SELECT * FROM usersInfo;

-- Trigger that populates age on adding a new user

-- Trigger that inserts a new row in user_logs if any value is updated in users table. If last name of a user
-- <old last name>, <new last name>, 'Last name updated', current time


```

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	full_name character varying (20)	dob date	age integer
1	1	Saugat	saugat32456@gmail.com	Saugat	Poudel	Saugat Poudel	2002-01-03	20
2	2	Saskar	saskar.khadka@gmail.com	Saskar	Khadka	[null]	2001-04-25	20
3	3	Rohan	dhakal.rohan@gmail.com	Rohan	Dhakal	[null]	2002-04-10	20
4	4	Nischal	nischalthapaliya229@gmail.com	Nischal	Thapaliya	[null]	1999-04-10	23
5	5	sundar	sundarpangeni@gmail.com	Sundar	Pangeni	Sundar Pangeni	1996-04-12	[null]

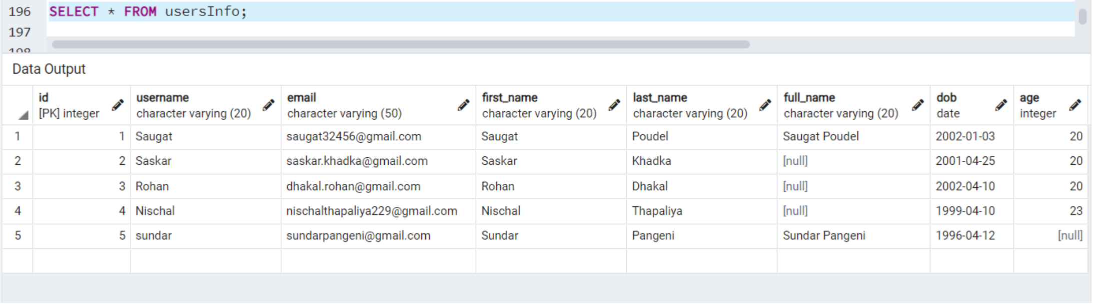


```
-- Trigger that populates age on adding a new user
-- Trigger that inserts a new row in user_logs if any value is updated in users table. If last name of a use
-- <old last name>, <new last name>, 'Last name updated', current time
CREATE OR REPLACE FUNCTION populateUserLogs() RETURNS TRIGGER
    AS $$
    BEGIN
        IF OLD.username <> NEW.username THEN
            INSERT INTO userLogs(old_value, new_value, description, log_time)
            VALUES (OLD.username, NEW.username, 'username updated', NOW());
        END IF;

        IF OLD.email <> NEW.email THEN
            INSERT INTO userLogs(old_value, new_value, description, log_time)
            VALUES (OLD.email, NEW.email, 'email updated', NOW());
        END IF;

        IF NEW.first_name <> OLD.first_name THEN
            INSERT INTO userLogs(old_value, new_value, description, log_time)
            VALUES(OLD.first_name, NEW.first_name, 'first_name updated', NOW());
        END IF;

        IF NEW.last_name <> OLD.last_name THEN
            INSERT INTO userLogs(old_value, new_value, description, log_time)
            VALUES(OLD.last_name, NEW.last_name, 'last_name updated', NOW());
        END IF;
```



```

        IF NEW.DOB <> OLD.DOB THEN
            INSERT INTO userLogs(old_value, new_value, description, log_time)
            VALUES(OLD.DOB, NEW.DOB, 'DOB updated', NOW());
        END IF;
        RETURN NEW;
    END
    $$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER triggerUserLogs AFTER UPDATE
    ON usersInfo
    FOR EACH ROW
    EXECUTE PROCEDURE populateUserLogs();

SELECT * FRO

UPDATE users
UPDATE users
UPDATE users
```

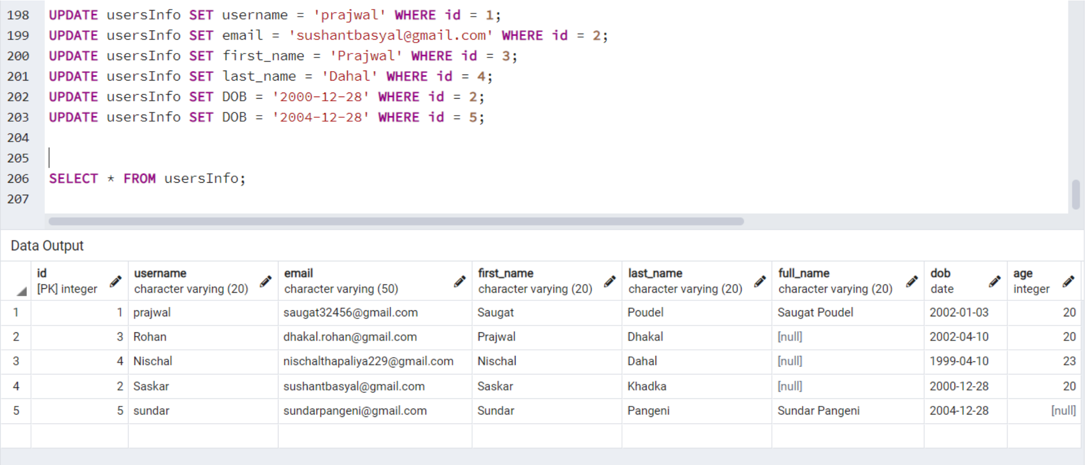
```
CREATE TRIGGER

Query returned successfully in 53 msec.
```



```
SELECT * FROM usersInfo;
```

Data Output

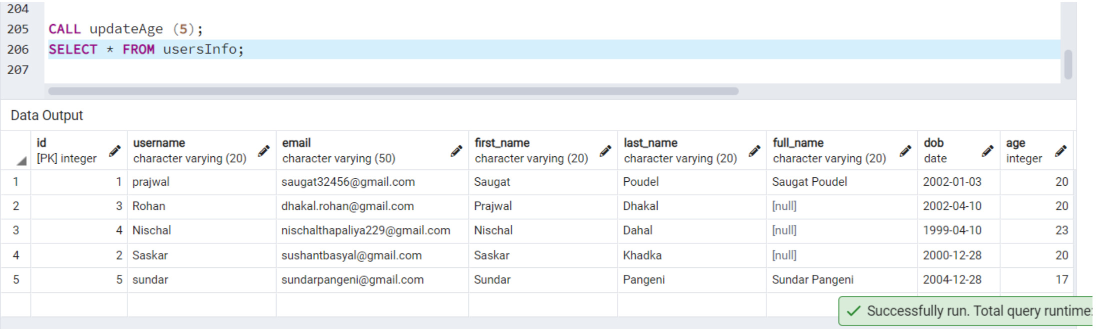
	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	full_name character varying (20)	dob date	age integer
1	1	Saugat	saugat32456@gmail.com	Saugat	Poudel	Saugat Poudel	2002-01-03	20
2	2	Saskar	saskar.khadka@gmail.com	Saskar	Khadka	[null]	2001-04-25	20
3	3	Rohan	dhakal.rohan@gmail.com	Rohan	Dhakal	[null]	2002-04-10	20
4	4	Nischal	nischalthapaliya229@gmail.com	Nischal	Thapaliya	[null]	1999-04-10	23
5	5	sundar	sundarpangeni@gmail.com	Sundar	Pangeni	Sundar Pangeni	1996-04-12	[null]



```sql
UPDATE usersInfo SET username = 'prajwal' WHERE id = 1;
UPDATE usersInfo SET email = 'sushantbasyal@gmail.com' WHERE id = 2;
UPDATE usersInfo SET first_name = 'Prajwal' WHERE id = 3;
UPDATE usersInfo SET last_name = 'Dahal' WHERE id = 4;
UPDATE usersInfo SET DOB = '2000-12-28' WHERE id = 2;
UPDATE usersInfo SET DOB = '2004-12-28' WHERE id = 5;


SELECT * FROM usersInfo;

```

Data Output

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	full_name character varying (20)	dob date	age integer
1	1	prajwal	saugat32456@gmail.com	Saugat	Poudel	Saugat Poudel	2002-01-03	20
2	3	Rohan	dhakal.rohan@gmail.com	Prajwal	Dhakal	[null]	2002-04-10	20
3	4	Nischal	nischalthapaliya229@gmail.com	Nischal	Dahal	[null]	1999-04-10	23
4	2	Saskar	sushantbasyal@gmail.com	Saskar	Khadka	[null]	2000-12-28	20
5	5	sundar	sundarpangeni@gmail.com	Sundar	Pangeni	Sundar Pangeni	2004-12-28	[null]



```sql

CALL updateAge (5);
SELECT * FROM usersInfo;

```

Data Output

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	full_name character varying (20)	dob date	age integer
1	1	prajwal	saugat32456@gmail.com	Saugat	Poudel	Saugat Poudel	2002-01-03	20
2	3	Rohan	dhakal.rohan@gmail.com	Prajwal	Dhakal	[null]	2002-04-10	20
3	4	Nischal	nischalthapaliya229@gmail.com	Nischal	Dahal	[null]	1999-04-10	23
4	2	Saskar	sushantbasyal@gmail.com	Saskar	Khadka	[null]	2000-12-28	20
5	5	sundar	sundarpangeni@gmail.com	Sundar	Pangeni	Sundar Pangeni	2004-12-28	17

✓ Successfully run. Total query runtime: