Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Report on ‘Lab Work 5’ [COMP 232]

Submitted by:

Chandan Kumar Mahato (31)

II-year, II semester

Submitted to:

Dr. Rajani Chulyadyo

Department of Computer Science and Engineering

Submission Date: April 19, 2022

1. Create a table that stores the following information about users:
    0. ID
    1. Username
    2. Email
    3. First name
    4. Last name
    5. Date of birth
    6. Age
2. Create a table named user_logs with the following columns:
    0. ID
    1. Old_value
    2. New_value
    3. Description
    4. Log_time
3. Create the following functions
    0. Function that returns the full name of the user with the given ID
    1. Function that returns the number of users
    2. Function that returns the age of the user with the given ID
4. Create the following stored procedures
    0. SP to update the full name of the user with the given ID
    1. SP to update the age of the user with the given ID
5. Create the following triggers
    0. Trigger that populates full name on adding a new user
    1. Trigger that populates age on adding a new user
    2. 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

Solution:

Queries SQL (structured query language) for the following questions:

1. Create a table that stores the following information about users

```
DROP TABLE IF EXISTS Users;
CREATE TABLE Users(
    id SERIAL PRIMARY KEY,
username VARCHAR(20) NOT NULL,
email VARCHAR(50),
first_name VARCHAR(20) NOT NULL,
last_name VARCHAR(20),
dob DATE NOT NULL,
age INT
);
```

```sql
-- Insert some data in table
INSERT INTO Users(username, email, first_name, last_name, dob)
VALUES
('Chandan','cmahato2000@gmail.com','Chandan', 'Mahato','2001-03-01'),
('Dipendra', 'dipendra@gmail.com', 'Dipendra', 'Mahato', '1998-01-01'),
('Dipak', 'dipak@gmail.com', 'Dipak', 'Mahato', '199-01-01'),
('Gokul', 'gokul@gmail.com', 'Gokul', 'Mahato', '2016-04-21'),
('Gauri', 'gauri@gmail.com', 'Gauri', 'Mahato', '2020-02-14');

SELECT * FROM Users;
```

Data Output | Explain | Messages | Notifications

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	dob date	age integer
1	1	Chandan	cmahato2000@gmail.com	Chandan	Mahato	2001-03-01	[null]
2	2	Dipendra	dipendra@gmail.com	Dipendra	Mahato	1998-01-01	[null]
3	3	Dipak	dipak@gmail.com	Dipak	Mahato	0199-01-01	[null]
4	4	Gokul	gokul@gmail.com	Gokul	Mahato	2016-04-21	[null]
5	5	Gauri	gauri@gmail.com	Gauri	Mahato	2020-02-14	[null]

2. Create a table named user_logs with the following columns:

```sql
DROP TABLE IF EXISTS user_logs;
CREATE TABLE user_logs(
    id SERIAL PRIMARY KEY,
    old_value VARCHAR (255),
    new_value VARCHAR (255),
    description varchar (50),
    log_time TIMESTAMP
)
```

```sql
-- 3. Create the following functions
-- 3.1. Function that returns the full name of the user with the given ID
CREATE OR REPLACE FUNCTION full_name(uid INTEGER)
    RETURNS VARCHAR
    AS
    $$
    DECLARE fullname VARCHAR;
    BEGIN
        SELECT first_name || ' ' || last_Name INTO fullname FROM Users WHERE ID = uid;
        RETURN fullname;
    END;
    $$ LANGUAGE plpgsql;
SELECT full_name(4);
```

Data Output | Explain | Messages | Notifications

	full_name character varying
1	Gokul Mahato

```sql
-- 3.2. Function that returns the number of users
CREATE OR REPLACE FUNCTION no_of_users()
    RETURNS INT
    AS
    $$
    DECLARE no_users INT;
    BEGIN
        SELECT COUNT(id) INTO no_users FROM Users;
        RETURN no_users;
    END;
    $$ LANGUAGE plpgsql;
SELECT no_of_users();
```

Data Output | Explain | Messages | Notifications

	no_of_users integer
1	5

```sql
-- 3.3. Function that returns the age of the user with the given ID
CREATE OR REPLACE FUNCTION age_of_given_user(uid INTEGER)
    RETURNS INTEGER
    AS
    $$
    DECLARE age_of_user INT;
    DECLARE dob_of_user DATE;
    BEGIN
        SELECT dob INTO dob_of_user FROM Users WHERE ID = uid;
        age_of_user = DATE_PART('year', now()) - DATE_PART('year', dob_of_user);
        return age_of_user;
    END;
    $$ LANGUAGE plpgsql;

SELECT age_of_given_user(1);
```

Data Output | Explain | Messages | Notifications

	age_of_given_user integer
1	21

```sql
-- 4 Create the following stored procedures
-- 4.1. SP to update the full name of the user with the given ID
DROP PROCEDURE set_fullname;
ALTER TABLE  Users ADD full_name VARCHAR(60);
CREATE PROCEDURE set_fullname(uid INTEGER)
    AS
    $$
    DECLARE fullname VARCHAR;
    BEGIN
        SELECT first_name || ' ' || last_name INTO fullname FROM Users WHERE id = uid;
        UPDATE Users set full_name = fullname WHERE ID = uid;
    END;
    $$LANGUAGE plpgsql;
CALL set_fullname(3);
SELECT * FROM Users;
```

Data Output | Explain | Messages | Notifications

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	dob date	age integer	full_name character varying (60)
1	1	Chandan	cmahato2000@gmail.com	Chandan	Mahato	2001-03-01	[null]	[null]
2	2	Dipendra	dipendra@gmail.com	Dipendra	Mahato	1998-01-01	[null]	[null]
3	4	Gokul	gokul@gmail.com	Gokul	Mahato	2016-04-21	[null]	[null]
4	5	Gauri	gauri@gmail.com	Gauri	Mahato	2020-02-14	[null]	[null]
5	3	Dipak	dipak@gmail.com	Dipak	Mahato	0199-01-01	[null]	Dipak Mahato

```sql
-- 4.2. SP to update the age of the user with the give ID
DROP PROCEDURE set_user_age;
CREATE PROCEDURE set_user_age(uid INTEGER)
    AS
    $$
    DECLARE age_of_user INTEGER;
    DECLARE dob_of_user DATE;
    BEGIN
        SELECT dob into dob_of_user FROM Users WHERE ID = uid;
        age_of_user = DATE_PART('year', now()) - DATE_PART('year', dob_of_user);
        UPDATE Users SET age = age_of_user WHERE ID = uid;
    END;
    $$ LANGUAGE plpgsql;

CALL set_user_age(1);
SELECT * FROM Users;

-- 5 Create the follwoing triggers
-- 5.1. Trigger that populates full name on adding a new user
```

Data Output | Explain | Messages | Notifications

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	dob date	age integer	full_name character varying (60)
1	2	Dipendra	dipendra@gmail.com	Dipendra	Mahato	1998-01-01	[null]	[null]
2	4	Gokul	gokul@gmail.com	Gokul	Mahato	2016-04-21	[null]	[null]
3	5	Gauri	gauri@gmail.com	Gauri	Mahato	2020-02-14	[null]	[null]
4	3	Dipak	dipak@gmail.com	Dipak	Mahato	0199-01-01	[null]	Dipak Mahato
5	1	Chandan	cmahato2000@gmail.com	Chandan	Mahato	2001-03-01	21	[null]

```sql
-- 5.1. Trigger that populates full name on adding a new user
DROP TRIGGER IF EXISTS calculate_fullname ON Users;
CREATE OR REPLACE FUNCTION fill_fullname()
    RETURNS trigger
    AS
    $$
    DECLARE fullname varchar;
    BEGIN
        SELECT first_name || ' ' || last_name INTO fullname FROM Users WHERE id = new.id;
        UPDATE Users SET full_name = fullname WHERE ID = NEW.id;
        RETURN NEW;
    END;
    $$ LANGUAGE plpgsql;

CREATE TRIGGER calculate_fullname AFTER INSERT
ON Users FOR EACH ROW
EXECUTE FUNCTION fill_fullname();

INSERT INTO Users(username, email, first_name, last_name, dob)
VALUES
('Durgendra', 'mahatodurgendra@gmail.com', 'Durgendra', 'Mahato', '1982-02-26');
SELECT * FROM Users;
```

Data Output | Explain | Messages | Notifications

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	dob date	age integer	full_name character varying (60)
3	5	Gauri	gauri@gmail.com	Gauri	Mahato	2020-02-14	[null]	[null]
4	3	Dipak	dipak@gmail.com	Dipak	Mahato	0199-01-01	[null]	Dipak Mahato
5	1	Chandan	cmahato2000@gmail.com	Chandan	Mahato	2001-03-01	21	[null]
6	6	Durgendra	mahatodurgendra@gmail.c...	Durgendra	Mahato	1982-02-26	[null]	Durgendra Mahato

```sql
-- 5.2. Trigger that populates age of on adding a new user
DROP TRIGGER IF EXISTS calculate_age ON Users;
CREATE OR REPLACE FUNCTION fill_age()
    RETURNS trigger
    AS
    $$
    DECLARE
    age_of_user INTEGER;
    dob_of_user DATE;
    BEGIN
        SELECT dob INTO dob_of_user FROM Users WHERE ID = NEW.id;
        age_of_user = DATE_PART('year', now()) - DATE_PART('year', dob_of_user);
        UPDATE Users SET age = age_of_user WHERE ID = NEW.id;
        RETURN NEW;
    END;
    $$ LANGUAGE plpgsql;
CREATE TRIGGER calculate_age AFTER INSERT
ON Users FOR EACH ROW
EXECUTE FUNCTION fill_age();
INSERT INTO Users(username, email, first_name, last_name, dob)
VALUES('Yashodah', 'yashodha@gmail.com', 'Yashodha', 'Mahato', '1992-04-24');
SELECT * FROM Users;
```

Data Output | Explain | Messages | Notifications

	id [PK] integer	username character varying (20)	email character varying (50)	first_name character varying (20)	last_name character varying (20)	dob date	age integer	full_name character varying (60)
6	6	Durgendra	mahatodurgendra@gmail.c...	Durgendra	Mahato	1982-02-26	[null]	Durgendra Mahato
7	7	Yashodah	yashodha@gmail.com	Yashodha	Mahato	1992-04-24	30	Yashodha Mahato
8	8	Yashodah	yashodha@gmail.com	Yashodha	Mahato	1992-04-24	30	Yashodha Mahato

```sql
-- 5.3
CREATE OR REPLACE FUNCTION update_logs()
    RETURNS TRIGGER
    AS
    $$
```

```
BEGIN
    IF OLD.first_name != NEW.first_name THEN
        INSERT INTO user_logs(old_value, new_value, description, log_time)
        VALUES
        (
        OLD.first_name,
        NEW.first_name,
        'First Name Changed',
        NOW()
        );
    END IF;
    IF OLD.last_name != NEW.last_name THEN
        INSERT INTO user_logs(old_value, new_value, description, log_time)
        VALUES
        (
        OLD.last_name,
        NEW.last_name,
        'Last Name Changed',
        NOW()
        );
    END IF;
    IF OLD.username != NEW.username THEN
        INSERT INTO user_logs(old_value, new_value, description, log_time)
        VALUES
        (
        OLD.username,
        NEW.username,
        'Username Changed',
        NOW()
        );
    END IF;

    IF OLD.email != NEW.email THEN
        INSERT INTO user_logs(old_value, new_value, description, log_time)
        VALUES
        (
        OLD.email,
        NEW.email,
        'Email Changed',
        NOW()
        );
    END if;
    IF OLD.dob != NEW.dob THEN
        INSERT INTO user_logs(old_value, new_value, description, log_time)
        VALUES
```

```
            (
            cast(OLD.dob AS VARCHAR),
            cast(NEW.dob AS VARCHAR),
            'DOB Changed',
            NOW()
            );
        END IF;

        RETURN NEW;
    END;
    $$ LANGUAGE plpgsql;
CREATE TRIGGER update_logs AFTER update
ON Users FOR EACH ROW
EXECUTE FUNCTION update_logs();
UPDATE Users SET first_name = 'Chandan123' WHERE ID = 1;
UPDATE Users SET last_name = 'Mahato123' WHERE ID = 1;
UPDATE Users SET dob = '2001-02-28' WHERE ID = 1;
UPDATE Users SET email = 'cmahato@gmail.com' WHERE ID = 1;
UPDATE Users SET username = 'chandanmahato123' WHERE ID = 1;
SELECT * FROM user_logs;
```

Query Editor | Query History

```
219             NOW()
220             );
221         END IF;
222
223         RETURN NEW;
224     END;
225     $$ LANGUAGE plpgsql;
226
227 CREATE TRIGGER update_logs AFTER update
228 ON Users FOR EACH ROW
229 EXECUTE FUNCTION update_logs();
230
231 UPDATE Users SET first_name = 'Chandan123' WHERE ID = 1;
232 UPDATE Users SET last_name = 'Mahato123' WHERE ID = 1;
233 UPDATE Users SET dob = '2001-02-28' WHERE ID = 1;
234 UPDATE Users SET email = 'cmahato@gmail.com' WHERE ID = 1;
235 UPDATE Users SET username = 'chandanmahato123' WHERE ID = 1;
236 SELECT * FROM user_logs;
```

Data Output | Explain | Messages | Notifications

	id [PK] integer	old_value character varying (255)	new_value character varying (255)	description character varying (50)	log_time timestamp without time zone
1	1	Chandan	Chandan123	First Name Changed	2022-04-18 17:40:28.702471
2	2	Mahato	Mahato123	Last Name Changed	2022-04-18 17:40:28.702471
3	3	2001-03-01	2001-02-28	DOB Changed	2022-04-18 17:40:28.702471
4	4	cmahato2000@gmail.com	cmahato@gmail.com	Email Changed	2022-04-18 17:40:28.702471
5	5	Chandan	chandanmahato123	Username Changed	2022-04-18 17:40:28.702471