

**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 18, 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

```
2. DROP TABLE IF EXISTS Users;
3. CREATE TABLE Users(
4.     id SERIAL PRIMARY KEY,
5.     username VARCHAR(20) NOT NULL,
6.     email VARCHAR(50),
7.     first_name VARCHAR(20) NOT NULL,
8.     last_name VARCHAR(20),
9.     dob DATE NOT NULL,
10.    age INT
11.);
```

```

23 -- Insert some data in table
24 INSERT INTO Users(username, email, first_name, last_name, dob)
25 VALUES
26 ('Chandan','cmahato2000@gmail.com','Chandan','Mahato','2001-03-01'),
27 ('Dipendra','dipendra@gmail.com','Dipendra','Mahato','1998-01-01'),
28 ('Dipak','dipak@gmail.com','Dipak','Mahato','199-01-01'),
29 ('Gokul','gokul@gmail.com','Gokul','Mahato','2016-04-21'),
30 ('Gauri','gauri@gmail.com','Gauri','Mahato','2020-02-14');
31
32 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:

```

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
)

```

```

34 -- 3. Create the following functions
35 -- 3.1. Function that returns the full name of the user with the given ID
36 CREATE OR REPLACE FUNCTION full_name(uid INTEGER)
37 RETURNS VARCHAR
38 AS
39 $$
40 DECLARE fullname VARCHAR;
41 BEGIN
42     SELECT first_name || ' ' || last_name INTO fullname FROM Users WHERE ID = uid;
43     RETURN fullname;
44 END;
45 $$ LANGUAGE plpgsql;
46 SELECT full_name(4);

```

Data Output Explain Messages Notifications

	full_name character varying
1	Gokul Mahato

```

48 -- 3.2. Function that returns the number of users
49 CREATE OR REPLACE FUNCTION no_of_users()
50     RETURNS INT
51     AS
52     $$
53     DECLARE no_users INT;
54     BEGIN
55         SELECT COUNT(id) INTO no_users FROM Users;
56         RETURN no_users;
57     END;
58     $$ LANGUAGE plpgsql;
59 SELECT no_of_users();

```

Data Output Explain Messages Notifications

	no_of_users	
	integer	
1	5	

```

61 -- 3.3. Function that returns the age of the user with the given ID
62 CREATE OR REPLACE FUNCTION age_of_given_user(uid INTEGER)
63     RETURNS INTEGER
64     AS
65     $$
66     DECLARE age_of_user INT;
67     DECLARE dob_of_user DATE;
68     BEGIN
69         SELECT dob INTO dob_of_user FROM Users WHERE ID = uid;
70         age_of_user = DATE_PART('year', now()) - DATE_PART('year', dob_of_user);
71         return age_of_user;
72     END;
73     $$ LANGUAGE plpgsql;
74
75 SELECT age_of_given_user(1);

```

Data Output Explain Messages Notifications

	age_of_given_user	
	integer	
1	21	

```

78 -- 4 Create the following stored procedures
79 -- 4.1. SP to update the full name of the user with the given ID
80 DROP PROCEDURE set_fullname;
81 ALTER TABLE Users ADD full_name VARCHAR(60);
82 CREATE PROCEDURE set_fullname(uid INTEGER)
83 AS
84 $$
85 DECLARE fullname VARCHAR;
86 BEGIN
87     SELECT first_name || ' ' || last_name INTO fullname FROM Users WHERE id = uid;
88     UPDATE Users SET full_name = fullname WHERE ID = uid;
89 END;
90 $$LANGUAGE plpgsql;
91 CALL set_fullname(3);
92 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

```

94 -- 4.2. SP to update the age of the user with the give ID
95 DROP PROCEDURE set_user_age;
96 CREATE PROCEDURE set_user_age(uid INTEGER)
97 AS
98 $$
99 DECLARE age_of_user INTEGER;
100 DECLARE dob_of_user DATE;
101 BEGIN
102     SELECT dob INTO dob_of_user FROM Users WHERE ID = uid;
103     age_of_user = DATE_PART('year', now()) - DATE_PART('year', dob_of_user);
104     UPDATE Users SET age = age_of_user WHERE ID = uid;
105 END;
106 $$ LANGUAGE plpgsql;
107
108 CALL set_user_age(1);
109 SELECT * FROM Users;

```

```

111 -- 5 Create the following triggers
112 -- 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]

```

112 -- 5.1. Trigger that populates full name on adding a new user
113 DROP TRIGGER IF EXISTS calculate_fullname ON Users;
114 CREATE OR REPLACE FUNCTION fill_fullname()
115     RETURNS trigger
116     AS
117     $$
118     DECLARE fullname varchar;
119     BEGIN
120         SELECT first_name || ' ' || last_name INTO fullname FROM Users WHERE id = new.id;
121         UPDATE Users SET full_name = fullname WHERE ID = NEW.id;
122         RETURN NEW;
123     END;
124     $$ LANGUAGE plpgsql;
125
126 CREATE TRIGGER calculate_fullname AFTER INSERT
127 ON Users FOR EACH ROW
128 EXECUTE FUNCTION fill_fullname();
129
130 INSERT INTO Users(username, email, first_name, last_name, dob)
131 VALUES
132 ('Durgendra', 'mahatodurgendra@gmail.com', 'Durgendra', 'Mahato', '1982-02-26');
133 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

```

135 -- 5.2. Trigger that populates age of on adding a new user
136 DROP TRIGGER IF EXISTS calculate_age ON Users;
137 CREATE OR REPLACE FUNCTION fill_age()
138     RETURNS trigger
139     AS
140     $$
141     DECLARE
142         age_of_user INTEGER;
143         dob_of_user DATE;
144     BEGIN
145         SELECT dob INTO dob_of_user FROM Users WHERE ID = NEW.id;
146         age_of_user = DATE_PART('year', now()) - DATE_PART('year', dob_of_user);
147         UPDATE Users SET age = age_of_user WHERE ID = NEW.id;
148         RETURN NEW;
149     END;
150     $$ LANGUAGE plpgsql;
151 CREATE TRIGGER calculate_age AFTER INSERT
152 ON Users FOR EACH ROW
153 EXECUTE FUNCTION fill_age();
154 INSERT INTO Users(username, email, first_name, last_name, dob)
155 VALUES('Yashodah', 'yashodha@gmail.com', 'Yashodha', 'Mahato', '1992-04-24');
156 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

```

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 Users;
237
238

```

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	7	Yashodah	yashodha@gmail.com	Yashodha	Mahato	1992-04-24	30	Yashodha Mahato
7	8	Yashodah	yashodha@gmail.com	Yashodha	Mahato	1992-04-24	30	Yashodha Mahato
8	1	chandanmahato123	cmahato@gmail.com	Chandan123	Mahato123	2001-02-28	21	[null]