

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

```

1  -- Create a table that stores the following information about users:
2  -- ID
3  -- Username
4  -- Email
5  -- First name
6  -- Last name
7  -- Date of birth
8  -- Age
9
10 CREATE TABLE usersInfo(
11     id INT NOT NULL,
12     username CHARACTER VARYING(20),
13     email CHARACTER VARYING(50),
14     first_name CHARACTER VARYING(20),
15     last_name CHARACTER VARYING(20),
16     full_name CHARACTER VARYING(20),
17     DOB DATE NOT NULL,
18     Age INT,
19     primary key (id)
20 );
21
22

```

Data Output Explain Messages Notifications

CREATE TABLE

Query returned successfully in 319 msec.

```

21
22 -- Inserting Values in the above table UsersInfo
23 INSERT INTO usersInfo(id,username,email,first_name,last_name,DOB)
24 VALUES
25 (1, 'Saugat','saugat32456@gmail.com','Saugat','Poudel','2002-01-03'),
26 (2, 'Saskar','saskar.khadka@gmail.com','Saskar','Khadka','2001-04-25'),
27 (3, 'Rohan','dhakal.rohan@gmail.com','Rohan','Dhakal','2002-04-10'),
28 (4, 'Nischal','nischalthapaliya229@gmail.com','Nischal','Thapaliya','1999-04-10');
29
30 SELECT * FROM usersInfo;
31
32

```

Data Output Explain Messages Notifications

INSERT 0 4

Query returned successfully in 51 msec.

29

30

31

32

SELECT * FROM usersInfo;

Data Output

Explain

Messages

Notifications

	id	username	email	first_name	last_name	full_name	dob	age
	[PK] integer	character varying (20)	character varying (50)	character varying (20)	character varying (20)	character varying (20)	date	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]

```

32 -- Create a table named user_logs with the following columns:
33 -- ID
34 -- Old_value
35 -- New_value
36 -- Description
37 -- Log_time
38 CREATE TABLE userLogs(
39     logId SERIAL NOT NULL,
40     old_value CHARACTER VARYING(25),
41     new_value CHARACTER VARYING(25),
42     description CHARACTER VARYING(25),
43     log_time TIMESTAMP,
44     PRIMARY KEY (logId)
45 );
46
47 -- Create the following functions
48 -- Function that returns the full name of the user with the given I
49
50 CREATE OR REPLACE FUNCTION fullname(uid INTEGER)
51     RETURN VARCHAR(100)

```

[Data Output](#)
[Explain](#)
[Messages](#)
[Notifications](#)

CREATE TABLE

Query returned successfully in 46 msec.

```

47 -- Create the following functions
48 -- Function that returns the full name of the user with the given ID
49
50 CREATE OR REPLACE FUNCTION fullname(uid INTEGER)
51 RETURNS CHARACTER VARYING
52 AS
53 $$
54 DECLARE
55     fname CHARACTER VARYING;
56     lname CHARACTER VARYING;
57 BEGIN
58     SELECT first_name, last_name INTO fname, lname FROM usersInfo WHERE id = uid;
59
60     RETURN fname || ' ' || lname;
61 END;
62 $$ LANGUAGE plpgsql;
63
64 SELECT username, email, first_name, last_name, fullname (usersInfo.id) FROM UsersInfo;
65
66 -- Function that returns the number of users

```

Data Output Explain Messages Notifications

CREATE FUNCTION

Query returned successfully in 37 msec.

```

63
64 SELECT username, email, first_name, last_name, fullname (usersInfo.id) FROM UsersInfo;
65
66 -- Function that returns the number of users

```

Data Output Explain Messages Notifications

	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

```
66 -- Function that returns the number of users
67 CREATE OR REPLACE FUNCTION countuser()
68     RETURNS integer
69     AS
70     $$
71     DECLARE
72         total integer;
73     BEGIN
74         SELECT COUNT(*) INTO total FROM usersInfo;
75
76         RETURN total;
77     END;
78     $$ LANGUAGE plpgsql;
79
80 SELECT countuser() AS total_user;
81
82 -- Function that returns the age of the user with the given ID
83
84
```

Data Output Explain Messages Notifications

CREATE FUNCTION

Query returned successfully in 45 msec.

```

79
80 SELECT countuser() AS total_user;
81
82 -- Function that returns the age of
83
84

```

Data Output Explain Messages Notifications

	total_user	
	integer	
1	4	

```

81
82 -- Function that returns the age of the user with the given ID
83
84 CREATE OR REPLACE FUNCTION calculateAge(uid INTEGER)
85 RETURNS int
86 AS
87 $$
88 DECLARE
89     users_age int;
90 BEGIN
91     SELECT date_part('year',AGE(dob)) Into users_age FROM usersInfo WHERE id = uid;
92
93     RETURN users_age;
94 END;
95 $$ LANGUAGE plpgsql;
96
97 SELECT fullname (usersInfo.id) , DOB , calculateAge (usersInfo.id) FROM usersInfo;
98
99

```

Data Output Explain Messages Notifications

	fullname	dob	calculateage	
	character varying	date	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	


```
100 -- Create the following stored procedures
101 -- SP to update the full name of the user with the given ID
102
103 CREATE OR REPLACE PROCEDURE updatefullName (uid INTEGER)
104 AS
105 $$
106 BEGIN
107     UPDATE usersInfo SET full_name = first_name || ' ' || last_name WHERE id = uid;
108 END;
109 $$ LANGUAGE plpgsql;
110
111
```

Data Output Explain Messages Notifications

CREATE PROCEDURE

Query returned successfully in 47 msec.

```
110
111 CALL updatefullName(1);
112 SELECT * FROM usersInfo;
113
114
115
```

Data Output Explain Messages Notifications								
id	username	email	first_name	last_name	full_name	dob	age	
[PK] integer	character varying (20)	character varying (50)	character varying (20)	character varying (20)	character varying (20)	date	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]

```

115 -- SP to update the age of the user with the given ID
116
117 CREATE OR REPLACE PROCEDURE updateAge(uid INTEGER)
118 AS
119 $$
120 BEGIN
121     UPDATE usersInfo SET age = date_part('year',AGE(dob)) WHERE id = uid;
122 END;
123 $$ LANGUAGE plpgsql;
124
125 CALL updateAge (1);
126 CALL updateAge (2);
127

```

Data Output Explain Messages Notifications

CREATE PROCEDURE

Query returned successfully in 41 msec.

```

124
125 CALL updateAge (1);
126 CALL updateAge (2);
127 CALL updateAge (3);
128 CALL updateAge (4);
129 SELECT * FROM usersInfo;
130

```

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)	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

```

131 -- Create the following triggers
132 -- Trigger that populates full name on adding a new user
133
134 CREATE OR REPLACE FUNCTION populatefullName() RETURNS TRIGGER
135 AS
136 $$
137 DECLARE full_name CHARACTER VARYING;
138 BEGIN
139     full_name = NEW.first_name || ' ' || NEW.last_name;
140     NEW.full_name = full_name;
141
142     RETURN NEW;
143 END;
144 $$ LANGUAGE plpgsql;
145
146 CREATE OR REPLACE TRIGGER triggerfullName BEFORE INSERT
147 ON
148 usersInfo
149 FOR EACH ROW
150 EXECUTE PROCEDURE populatefullName();
151

```

Data Output Explain Messages Notifications

CREATE TRIGGER

Query returned successfully in 40 msec.

```

152 INSERT INTO usersInfo(id,username,email,first_name,last_name,DOB)
153 VALUES
154 (5,'sundar', 'sundarpangeni@gmail.com', 'Sundar', 'Pangeni', '1996-04-12');
155 SELECT * FROM usersInfo;
156
157 -- Trigger that populates age on adding a new user
158
159 -- Trigger that inserts a new row in user_logs if any value is updated in users table. If last name of a user
160 -- <old last name>, <new last name>, 'Last name updated', current time
161
162

```

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)	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]

```

157 -- Trigger that populates age on adding a new user
158 -- Trigger that inserts a new row in user_logs if any value is updated in users table. If last name of a use
159 -- <old last name>, <new last name>, 'Last name updated', current time
160 CREATE OR REPLACE FUNCTION populateUserLogs() RETURNS TRIGGER
161 AS $$
162 BEGIN
163     IF OLD.username <> NEW.username THEN
164         INSERT INTO userLogs(old_value, new_value, description, log_time)
165         VALUES (OLD.username, NEW.username, 'username updated', NOW());
166     END IF;
167
168     IF OLD.email <> NEW.email THEN
169         INSERT INTO userLogs(old_value, new_value, description, log_time)
170         VALUES (OLD.email, NEW.email, 'email updated', NOW());
171     END IF;
172
173     IF NEW.first_name <> OLD.first_name THEN
174         INSERT INTO userLogs(old_value, new_value, description, log_time)
175         VALUES(OLD.first_name, NEW.first_name, 'first_name updated', NOW());
176     END IF;
177
178     IF NEW.last_name <> OLD.last_name THEN
179         INSERT INTO userLogs(old_value, new_value, description, log_time)
180         VALUES(OLD.last_name, NEW.last_name, 'last_name updated', NOW());
181     END IF;
182

```

```

182
183     IF NEW.DOB <> OLD.DOB THEN
184         INSERT INTO userLogs(old_value, new_value, description, log_time)
185         VALUES(OLD.DOB, NEW.DOB, 'DOB updated', NOW());
186     END IF;
187     RETURN NEW;
188 END
189 $$ LANGUAGE plpgsql;
190

```

```

191 CREATE OR REPLACE TRIGGER triggerUserLogs AFTER UPDATE
192 ON usersInfo
193 FOR EACH ROW
194 EXECUTE PROCEDURE populateUserLogs();
195

```

```

196 SELECT * FROM usersInfo;
197
198 UPDATE usersInfo SET email = 'sundarpangeni@gmail.com' WHERE id = 5;
199 UPDATE usersInfo SET first_name = 'Sundar' WHERE id = 5;
200 UPDATE usersInfo SET last_name = 'Pangeni' WHERE id = 5;

```

Query returned successfully in 53 msec.

Data Output

id	username	email	first_name	last_name	full_name	dob	age
[PK] integer	character varying (20)	character varying (50)	character varying (20)	character varying (20)	character varying (20)	date	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]

```

198 UPDATE usersInfo SET username = 'prajwal' WHERE id = 1;
199 UPDATE usersInfo SET email = 'sushantbasyal@gmail.com' WHERE id = 2;
200 UPDATE usersInfo SET first_name = 'Prajwal' WHERE id = 3;
201 UPDATE usersInfo SET last_name = 'Dahal' WHERE id = 4;
202 UPDATE usersInfo SET DOB = '2000-12-28' WHERE id = 2;
203 UPDATE usersInfo SET DOB = '2004-12-28' WHERE id = 5;
204
205 |
206 SELECT * FROM usersInfo;
207

```

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]

```

204
205 CALL updateAge (5);
206 SELECT * FROM usersInfo;
207

```

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