

LAB 4

1. Create a multiply function that accepts two numbers and returns their product.

```
testdb=# CREATE FUNCTION multiply_nums(num1 integer, num2 integer)
RETURNS integer AS $$ BEGIN
RETURN num1 * num2; END;
$$ LANGUAGE plpgsql;
CREATE FUNCTION
testdb=# SELECT multiply_nums(10,3);
multiply_nums
-----
          30
(1 row)

testdb=#
```

2. Create a hello_world function that takes a name as input and returns a personalized welcome message for that name.

```
testdb=# CREATE FUNCTION hello_world (name VARCHAR(50))
RETURNS VARCHAR(50) AS $$
BEGIN
RETURN CONCAT('Welcome ', name);
END;
$$ LANGUAGE plpgsql;
CREATE FUNCTION
testdb=# SELECT hello_world(Nada);
ERROR:  column "nada" does not exist
LINE 1: SELECT hello_world(Nada);
                        ^

testdb=# SELECT hello_world('Nada');
hello_world
-----
Welcome Nada
(1 row)

testdb=#
```

3. Create a function that accepts a number and determines whether it is odd or even.

```
nada@nada-G3-3500: ~  
testdb=# CREATE OR REPLACE FUNCTION even_odd(num INTEGER)  
RETURNS VARCHAR AS $$  
BEGIN  
IF num %2 = 0 THEN  
RETURN 'even';  
ELSE  
RETURN 'odd';  
END IF;  
END;  
$$ LANGUAGE plpgsql;  
testdb=# SELECT even_odd(5);  
even_odd  
-----  
odd  
(1 row)  
  
testdb=# SELECT even_odd(2);  
even_odd  
-----  
even  
(1 row)  
  
testdb=#
```

4. Create a function that takes a Student ID as input and retrieves all information related to that student.

```
testdb=# CREATE FUNCTION stu_info(stu_id INTEGER)  
RETURNS TABLE(id INTEGER, name TEXT, email TEXT, address character(50), birth_date DATE, gender gender_type) AS $$  
BEGIN  
RETURN QUERY SELECT s.id, s.e_name, s.email, s.address, s.birth_date, s.gender FROM student s WHERE s.id=stu_id;  
END;  
$$ LANGUAGE plpgsql;  
testdb=# SELECT stu_info(3);  
stu_info  
-----  
(3,Mohamed,emily.j@university.edu,"456 Oak Ave, Somewhere",2000-05-09,Male)  
(1 row)  
  
testdb=#
```

5. Implement a function that takes the name of a subject and calculates the average grades for that subject.

```
testdb=# CREATE FUNCTION avg_grades(sname TEXT)  
RETURNS NUMERIC AS $$  
BEGIN  
RETURN (SELECT AVG(g.grade) FROM subject s JOIN grades g ON s.id = g.sub_id WHERE s.sub_name = sname);  
END;  
$$ LANGUAGE plpgsql;  
testdb=# SELECT avg_grades('html');  
avg_grades  
-----  
79.4500000000000000  
(1 row)  
  
testdb=#
```

6. Create a trigger to automatically save deleted student records from the Student table to the Deleted_Students table.

```
testdb=# CREATE TABLE deleted_student (id INTEGER, name TEXT, email TEXT, address character(50), track_id INTEGER, birth_date DATE, gender gender_type, deleted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, deleted_by TEXT DEFAULT current_user);
CREATE TABLE
testdb=# CREATE FUNCTION save_del_studs()
RETURNS TRIGGER AS $$
BEGIN
INSERT INTO deleted_student
SELECT OLD.*, CURRENT_TIMESTAMP, current_user;
DELETE FROM stu_sub WHERE stu_id = OLD.id;
RETURN OLD;
END;
$$ LANGUAGE plpgsql;
CREATE FUNCTION
testdb=# CREATE TRIGGER trigg_deleted_students
BEFORE DELETE ON student
FOR EACH ROW EXECUTE FUNCTION save_del_studs();
CREATE TRIGGER
testdb=# DELETE FROM student WHERE student.id = 2;
DELETE 1
testdb=#
testdb=# SELECT * FROM student;
testdb=# SELECT * FROM deleted_student;
-----
id | name | email | address | track_id | birth_date | gender | deleted_at | deleted_by
-----
2 | John Smith | john.smith@university.edu | 123 Main St, Anytown | 1 | 2001-05-30 | Male | 2025-05-01 22:21:16.801026 | postgres
(1 row)

testdb=# select * from stu_sub;
stu_id | sub_id
-----
1 | 2
3 | 3
(2 rows)

testdb=#
```

```
nada@nada-G3-3500: ~
-----
id | e_name | email | address | track_id | birth_date | gender
-----
3 | Mohamed | emily.j@university.edu | 456 Oak Ave, Somewhere | 2 | 2000-05-09 | Male
1 | Nada | ddds@gmail.com | Assuit | 3 | 2000-05-15 | Female
(2 rows)

(END)
```

7. Create a trigger to monitor changes made to the student table, including additions, updates, and deletions. This trigger will record the time of each action and provide a description of the action in another table.

```
Activities Terminal 1 ماي 11:39 م en 34%
nada@nada-G3-3500: ~
testdb=# CREATE TABLE student_logs (
log_id SERIAL PRIMARY KEY,
student_id INTEGER,
action TEXT CHECK (action IN ('INSERT', 'UPDATE', 'DELETE')),
action_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
old_data JSONB,
new_data JSONB
);
CREATE TABLE
```

```

testdb=# CREATE FUNCTION monitor_changes()
RETURNS TRIGGER AS $$
BEGIN
IF TG_OP = 'INSERT' THEN
INSERT INTO student_logs(student_id, action, new_data)
VALUES (NEW.id, 'INSERT', to_jsonb(NEW));
ELSIF TG_OP = 'UPDATE' THEN
INSERT INTO student_logs(student_id, action, old_data, new_data)
VALUES (OLD.id, 'UPDATE', to_jsonb(OLD), to_jsonb(NEW));
ELSIF TG_OP = 'DELETE' THEN
INSERT INTO student_logs(student_id, action, old_data)
VALUES (OLD.id, 'DELETE', to_jsonb(OLD));
END IF;
RETURN NULL;
END;
$$ LANGUAGE plpgsql;
CREATE FUNCTION
testdb=# CREATE TRIGGER log_insert_update
AFTER INSERT OR UPDATE ON student
FOR EACH ROW EXECUTE FUNCTION monitor_changes();
CREATE TRIGGER
testdb=# CREATE TRIGGER log_delete
BEFORE DELETE ON student
FOR EACH ROW EXECUTE FUNCTION monitor_changes();
CREATE TRIGGER

```

```

Activities Terminal 1 مايو 11:43 م en 37%
nada@nada-G3-3500: ~
testdb=# INSERT INTO student(id, e_name, email) VALUES (5, 'aya', 'aya@example.com');
INSERT 0 1
testdb=# UPDATE student SET gender = 'FEMALE' WHERE id = 5;
ERROR:  invalid input value for enum gender_type: "FEMALE"
LINE 1: UPDATE student SET gender = 'FEMALE' WHERE id = 5;
      ^
testdb=# UPDATE student SET gender = 'Female' WHERE id = 5;
UPDATE 1
testdb=# DELETE FROM student WHERE id = 1;
DELETE 0

```

```

nada@nada-G3-3500: ~
log_id | student_id | action |      action_time      |          new_data          |          old_data          |
-----+-----+-----+-----+-----+-----+
1 |          5 | INSERT | 2025-05-01 23:25:12.66515 | {"id": 5, "email": "aya@example.com", "e_name": "aya", "gender": null, "address": null, "track_id": null, "birth_date": null} |
2 |          5 | UPDATE | 2025-05-01 23:27:22.090549 | {"id": 5, "email": "aya@example.com", "e_name": "aya", "gender": null, "address": null, "track_id": null, "birth_date": null} | {"id": 5, "email": "aya@example.com", "e_name": "aya", "gender": "Female", "address": null, "track_id": null, "birth_date": null}
3 |          1 | DELETE | 2025-05-01 23:27:47.960164 | {"id": 1, "email": "ddsss@gmail.com", "e_name": "Nada", "gender": "Female", "address": "Assuit", "track_id": 3, "birth_date": "2000-05-15"} |
(3 rows)
(END)

```

```
Activities Terminal 1 مايو 11:44 م en 39 %
nada@nada-G3-3500: ~

testdb=# SELECT * FROM student_logs;
testdb=# SELECT * FROM student;
id | e_name | email | address | track_id | birth_date | gender
-----+-----+-----+-----+-----+-----+-----
3 | Mohamed | emily.j@university.edu | 456 Oak Ave, Somewhere | 2 | 2000-05-09 | Male
1 | Nada | dddss@gmail.com | Assuit | 3 | 2000-05-15 | Female
5 | aya | aya@example.com | | | | Female
(3 rows)

testdb=# DELETE FROM student WHERE e_name = 'Mohamed';
DELETE 0
testdb=# SELECT * FROM student_logs;
testdb=# SELECT * FROM student_logs WHERE e_name='Mohamed';
ERROR: column "e_name" does not exist
LINE 1: SELECT * FROM student_logs WHERE e_name='Mohamed';
                                         ^
testdb=# SELECT * FROM student WHERE e_name='Mohamed';
id | e_name | email | address | track_id | birth_date | gender
-----+-----+-----+-----+-----+-----+-----
3 | Mohamed | emily.j@university.edu | 456 Oak Ave, Somewhere | 2 | 2000-05-09 | Male
(1 row)

testdb=# DELETE FROM student WHERE e_name = 'Mohamed';
DELETE 0
testdb=# SELECT * FROM student;
id | e_name | email | address | track_id | birth_date | gender
-----+-----+-----+-----+-----+-----+-----
3 | Mohamed | emily.j@university.edu | 456 Oak Ave, Somewhere | 2 | 2000-05-09 | Male
1 | Nada | dddss@gmail.com | Assuit | 3 | 2000-05-15 | Female
5 | aya | aya@example.com | | | | Female
(3 rows)
```

```
Activities Terminal 1 مايو 11:46 م en 40 %
nada@nada-G3-3500: ~

log_id | student_id | action | action_time | new_data | old_data
-----+-----+-----+-----+-----+-----
1 | 5 | INSERT | 2025-05-01 23:25:12.66515 | {"id": 5, "email": "aya@example.com", "e_name": "aya", "gender": null, "address": null, "track_id": null, "birth_date": null} | null
2 | 5 | UPDATE | 2025-05-01 23:27:22.090549 | {"id": 5, "email": "aya@example.com", "e_name": "aya", "gender": null, "address": null, "track_id": null, "birth_date": null} | {"id": 5, "email": "aya@example.com", "e_name": "aya", "gender": "Female", "address": null, "track_id": null, "birth_date": null}
3 | 1 | DELETE | 2025-05-01 23:27:47.960164 | {"id": 1, "email": "dddss@gmail.com", "e_name": "Nada", "gender": "Female", "address": "Assuit", "track_id": 3, "birth_date": "2000-05-15"} | {"id": 3, "email": "emily.j@university.edu", "e_name": "Mohamed", "gender": "Male", "address": "456 Oak Ave, Somewhere", "track_id": 2, "birth_date": "2000-05-09"}
4 | 3 | DELETE | 2025-05-01 23:32:35.008007 | {"id": 3, "email": "emily.j@university.edu", "e_name": "Mohamed", "gender": "Male", "address": "456 Oak Ave, Somewhere", "track_id": 2, "birth_date": "2000-05-09"} | {"id": 3, "email": "emily.j@university.edu", "e_name": "Mohamed", "gender": "Male", "address": "456 Oak Ave, Somewhere", "track_id": 2, "birth_date": "2000-05-09"}
5 | 3 | DELETE | 2025-05-01 23:38:01.2502 | {"id": 3, "email": "emily.j@university.edu", "e_name": "Mohamed", "gender": "Male", "address": "456 Oak Ave, Somewhere", "track_id": 2, "birth_date": "2000-05-09"} | {"id": 3, "email": "emily.j@university.edu", "e_name": "Mohamed", "gender": "Male", "address": "456 Oak Ave, Somewhere", "track_id": 2, "birth_date": "2000-05-09"}
(5 rows)
```