## Задача 1.

# 1.1 создаем таблицы в DBEAUER при помощи sql-скрипта

<b>⊞</b> Статистика 1 ×				
Name	Value			
Queries	2			
Updated Rows	0			
Execute time (ms)	119			
Fetch time (ms)	0			
Total time (ms)	119			
Start time	2023-09-01 16:50:38.745			
Finish time	2023-09-01 16:50:38.868			

```
1.2 создаем функцию update_total_score() и триггер
update_total_score_trigger
CREATE OR REPLACE FUNCTION update_total_score()
 RETURNS trigger AS
$$
BEGIN
 UPDATE students SET total_score = (SELECT sum(score) FROM activity_scores WHERE
student_id = NEW.student_id) WHERE id = NEW.student_id;
 RETURN NULL;
END;
$$
LANGUAGE 'plpgsql';
CREATE OR REPLACE TRIGGER update_total_score_trigger
 AFTER UPDATE OR INSERT OR DELETE
 ON "activity_scores"
 FOR EACH ROW
 EXECUTE FUNCTION update_total_score();
```

В Статистика 1 >	7	
Name	Value	
Name	value	
Queries	2	
Updated Rows	0	
Execute time (ms)	104	
Fetch time (ms)	0	
Total time (ms)	104	
Start time	2023-09-04 19:34:24.073	
Finish time	2023-09-04 19:34:24.180	

### Задача 2.

## 2. Создаем функцию calculate\_scholarship() и триггер update\_scholarship\_trigger

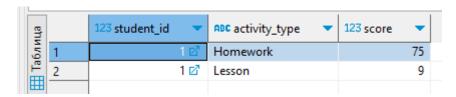
```
CREATE OR REPLACE FUNCTION calculate_scholarship()
 RETURNS trigger AS
BEGIN
 CASE WHEN NEW.total_score >= 90 THEN UPDATE students SET scholarship = 1000 WHERE
id = NEW.id;
   WHEN NEW.total_score < 90 AND NEW.total_score >= 80 THEN UPDATE students SET
scholarship = 500 WHERE id = NEW.id;
    ELSE UPDATE students SET scholarship = 0 WHERE id = NEW.id;
 END CASE;
RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';
create or replace TRIGGER update_scholarship_trigger
 AFTER UPDATE OF total_score OR INSERT OR DELETE
 ON students
 FOR EACH ROW
  EXECUTE FUNCTION calculate_scholarship();
```

В Статистика 1 ×		
Value		
2		
0		
8		
0		
8		
2023-09-05 15:33:22.463		
2023-09-05 15:33:22.483		
	Value 2 0 8 0 8 2023-09-05 15:33:22.463	

# 3. Проверяем, что всё работает

# SELECT \*

FROM activity\_scores



# SELECT \*

**FROM** students

