МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №  43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

### Старший преподаватель                                                               Путилова Н. В.

должность, уч. Степень, звание   подпись, дата                    инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

Триггеры. Обеспечение активной целостности данных базы данных.

по курсу: ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

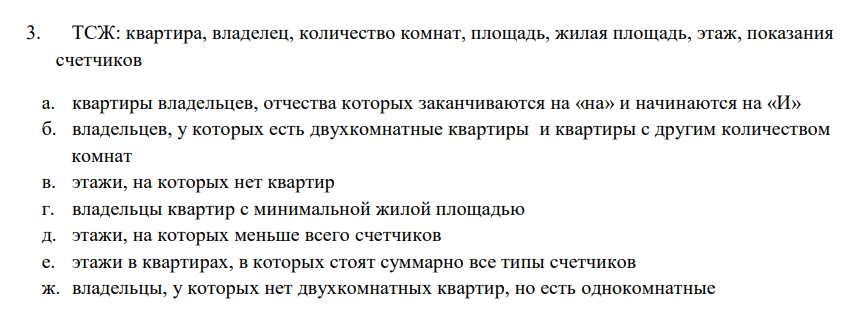
РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 4136                                                                        Бобрович Н. С.

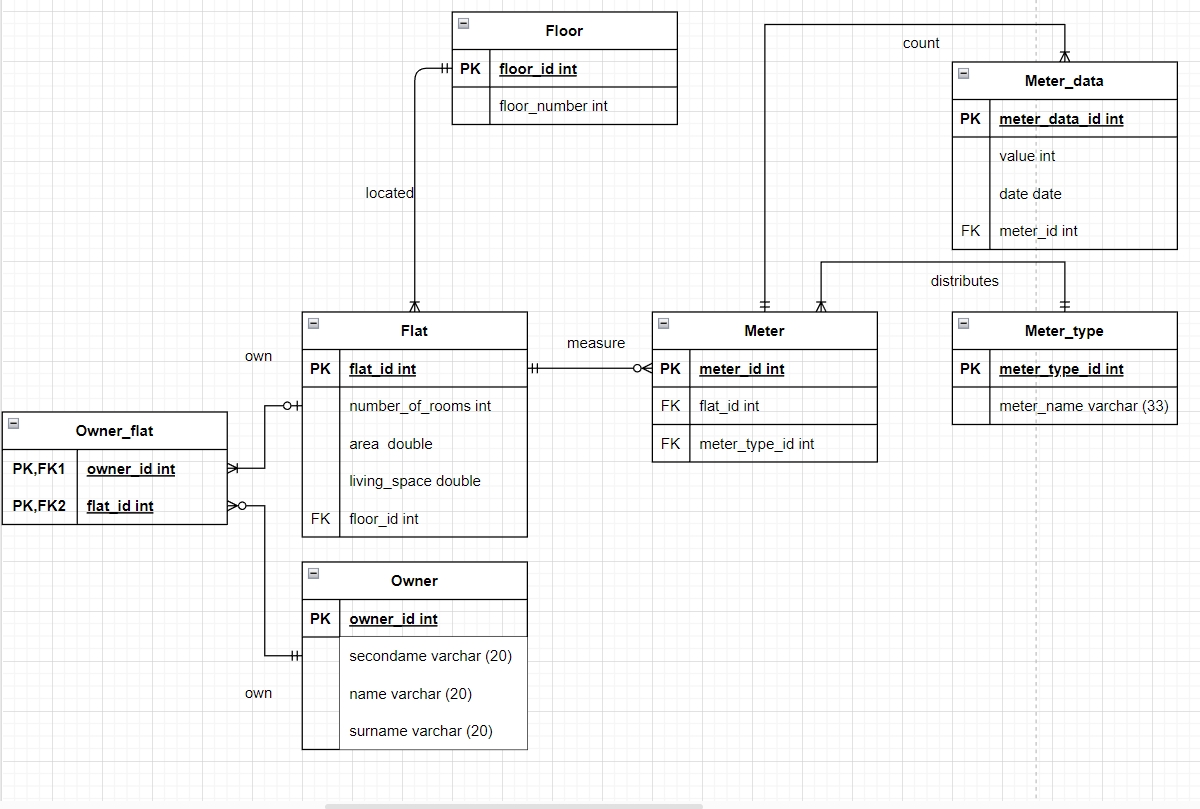
                                                                         подпись, дата                      инициалы, фамилия

Санкт-Петербург 2023

Вариант 3:



Физическая модель БД:



Назначение и тексты триггеров и SQL операторы и наборы данных, иллюстрирующие работу триггеров:

1. **Триггер BEFORE DELETE:**

delimiter //

CREATE TRIGGER t1\_del\_m\_with\_m\_data

BEFORE DELETE ON meter

FOR EACH ROW

BEGIN

IF EXISTS (SELECT \* FROM meter\_data WHERE meter\_id = OLD.meter\_id AND value = 0) THEN

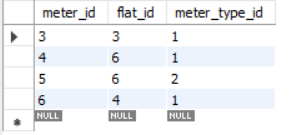
SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Нельзя удалять счётчик с ненулевыми показаниями';

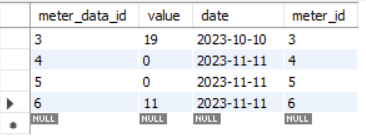
END IF;

END//

delimiter ;

**ДО:**





DELETE from meter where meter\_id = 5;

**ПОСЛЕ:**



**2) Триггер AFTER INSERT:** (Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей)

delimiter //

CREATE TRIGGER t2

AFTER INSERT ON owner\_flat

FOR EACH ROW

BEGIN

UPDATE owner

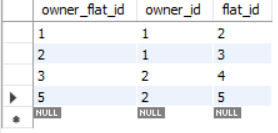
SET flat\_count = flat\_count + 1

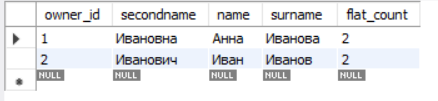
WHERE owner\_id = NEW.owner\_id;

END//

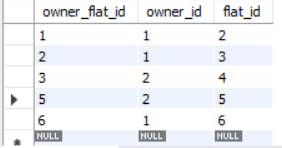
delimiter ;

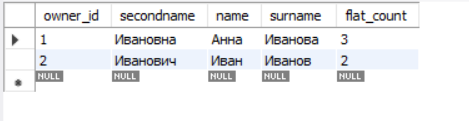
**ДО:**





INSERT INTO owner\_flat VALUES (6, 1, 6);

**ПОСЛЕ:**  




**3) Триггер Before insert:** (обеспечение безопасности данных)

delimiter //

CREATE TRIGGER t3

BEFORE INSERT ON flat

FOR EACH ROW

BEGIN

IF NOT EXISTS (SELECT \* FROM floor WHERE floor\_id = NEW.floor\_id) THEN

SIGNAL SQLSTATE '45000'

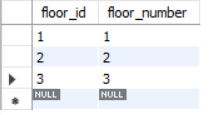
SET MESSAGE\_TEXT = 'Квартира с указанным этажом не существует';

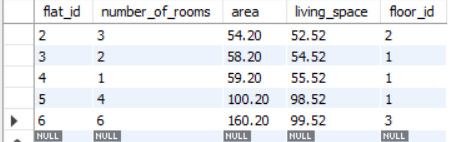
END IF;

END//

delimiter ;

**ДО:**





INSERT INTO flat VALUES (7, 7, 170.20, 99.55, 4);

**ПОСЛЕ:**



**4) Триггер after delete: (**Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей))

delimiter //

CREATE TRIGGER t4

AFTER DELETE ON owner\_flat

FOR EACH ROW

BEGIN

UPDATE owner

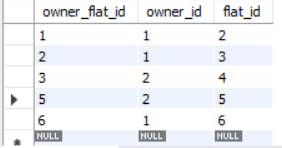
SET flat\_count = flat\_count - 1

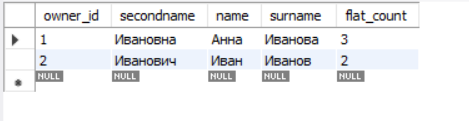
WHERE owner\_id = OLD.owner\_id;

END//

delimiter ;

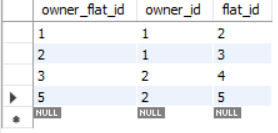
**ДО:**

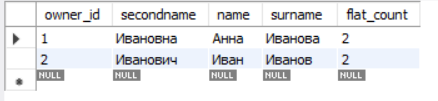




delete from owner\_flat where owner\_flat\_id = 6;

**ПОСЛЕ:**





**5) Триггер BEFORE UPDATE:** (обеспечение безопасности данных)

delimiter //

CREATE TRIGGER t5

BEFORE UPDATE ON flat

FOR EACH ROW

BEGIN

IF NOT EXISTS (SELECT 1 FROM floor WHERE floor\_id = NEW.floor\_id) THEN

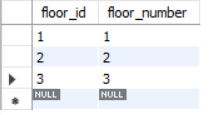
SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Квартира с данным этажом не существует';

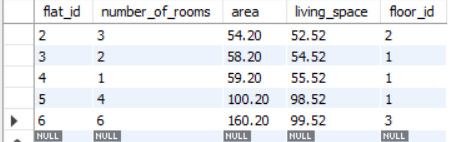
END IF;

END//

delimiter ;

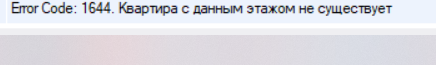
**ДО:**





UPDATE flat SET floor\_id = 4 WHERE flat\_id = 6;

**ПОСЛЕ:**



1. **Триггер AFTER UPDATE:** (Вычисление/поддержание в актуальном состоянии вычислимых (производных) атрибутов (полей))

delimiter //

CREATE TRIGGER t6

AFTER UPDATE ON owner\_flat

FOR EACH ROW

BEGIN

IF NEW.owner\_id <> OLD.owner\_id THEN

UPDATE owner

SET flat\_count = flat\_count - 1

WHERE owner\_id = OLD.owner\_id;

UPDATE owner

SET flat\_count = flat\_count + 1

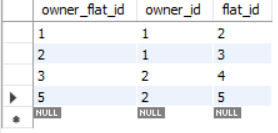
WHERE owner\_id = NEW.owner\_id;

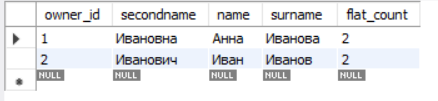
END IF;

END//

delimiter ;

**ДО:**





UPDATE owner\_flat SET owner\_id = 2 WHERE owner\_flat\_id = 3;

**ПОСЛЕ:**

