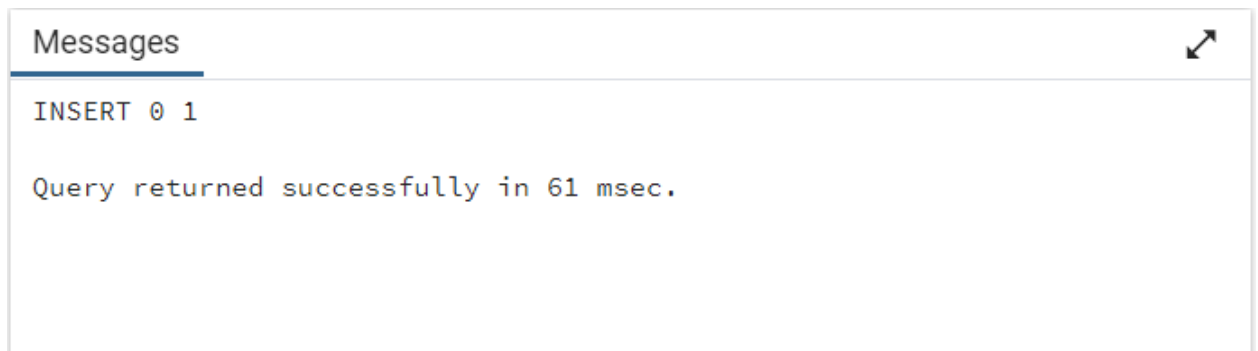


Задание 1. Примените пожалуйста [скрипт](#) для базы данных PostgreSQL. Если есть необходимость, то вы можете применить его заново, предварительно удалив таблицы командой

DROP TABLE *имя_таблицы*;

Применить скрипт - это значит выполнить его через ваш используемый IDE (DataGrip, DBeaver , pgAdmin)

Результат работы:



Задание 2.

- а.** Напишите SQL запрос который создаст таблицу с именем “*student_on_course_kn*” ,содержащую информацию по успеваемости студентов из Казани и Новосибирска (как часть данных таблицы “*student_on_course*” со всеми атрибутами). Используйте паттерн “**CREATE ... AS SELECT ...**”.

Пожалуйста приложите команду создания таблицы и результат запроса

SELECT * FROM student_on_course_kn **ORDER BY** id;

в виде скриншота к заданию.

Код к запросу:

```
CREATE TABLE student_on_course_kn AS
SELECT soc.*
FROM student_on_course soc
JOIN student s ON soc.student_id = s.id
WHERE s.city IN ('Казань', 'Новосибирск');
```

Результат работы:

Messages
SELECT 8
Query returned successfully in 51 msec.

Рис. 1. Результат выполнения запроса на создание таблицы

	id bigint	student_id bigint	course_id bigint	student_rating integer	finished_date date
1	10	10	10	75	[null]
2	20	10	20	83	[null]
3	30	10	40	40	[null]
4	50	30	30	76	2022-12-12
5	60	30	40	42	[null]
6	70	40	10	76	[null]
7	80	40	20	83	[null]
8	90	40	50	96	[null]

Рис. 2. Таблица student_course_kn

- b. Обновите баллы всех студентов в новой созданной таблице “student_on_course_kn” одной **UPDATE** командой с учетом следующих правил
- уменьшаем на 5 баллов *student_rating*, если значение *student_rating* лежит между 0 и 60 баллами включительно
 - уменьшаем на 10 баллов *student_rating*, если значение *student_rating* лежит между 61 и 80 баллами включительно
 - уменьшаем на 15 баллов *student_rating*, если значение *student_rating* лежит между 81 и 90 баллами включительно
 - во всех остальных случаях вычитаем 20 баллов из значения *student_rating*

Пожалуйста приложите команду обновления таблицы и результат запроса

SELECT * FROM student_on_course_kn ORDER BY id;

в виде скриншота к заданию.

Код запроса:

```
--SELECT * FROM student_on_course_kn  
  
UPDATE student_on_course_kn  
  
SET student_rating = CASE  
  
    WHEN student_rating >= 0 and student_rating <= 60 THEN student_rating -5  
  
    WHEN student_rating >= 61 and student_rating <= 80 THEN student_rating -10  
  
    WHEN student_rating >= 81 and student_rating <= 90 THEN student_rating -15  
  
    ELSE student_rating -20  
  
END;
```

Результат работы:

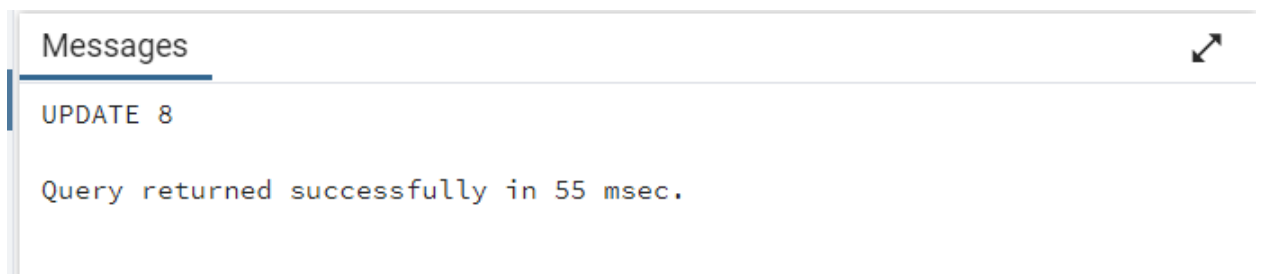


Рис. 1. Результат выполнения запроса на обновление

	id bigint	student_id bigint	course_id bigint	student_rating integer	finished_date date
1	10	10	10	65	[null]
2	20	10	20	68	[null]
3	30	10	40	35	[null]
4	50	30	30	66	2022-12-12
5	60	30	40	37	[null]
6	70	40	10	66	[null]
7	80	40	20	68	[null]
8	90	40	50	76	[null]

Рис. 2. Измененная таблица student_course_kn

- с. Удалите запись о студентах из таблицы “*student_on_course_kn*”, родной город которых Новосибирск, посещающие курс “Цифровая трансформация” Пожалуйста приложите команду удаления строк таблицы и результат запроса

```
SELECT * FROM student_on_course_kn ORDER BY id;
```

в виде скриншота к заданию.

Код запроса:

```
--SELECT * FROM student_on_course_kn ORDER BY id;
```

```
DELETE FROM student_on_course_kn
```

```
WHERE student_id IN (
```

```
    SELECT student.id
```

```
    FROM student
```

```
    JOIN student_on_course_kn ON student.id = student_on_course_kn.student_id
```

```
    JOIN course ON student_on_course_kn.course_id = course.id
```

```
    WHERE student.city = 'Новосибирск'
```

```
    AND course.name = 'Цифровая трансформация'
```

```
);
```

Результат выполнения:


Messages		
DELETE 2		
Query returned successfully in 52 msec.		

Рис. 1. Результат выполнения запроса на удаление

	id bigint	student_id bigint	course_id bigint	student_rating integer	finished_date date
1	10	10	10	65	[null]
2	20	10	20	68	[null]
3	30	10	40	35	[null]
4	70	40	10	66	[null]
5	80	40	20	68	[null]
6	90	40	50	76	[null]

Рис. 2. Измененная таблица student_course_kn

- d. Напишите SQL запрос который к существующим записям таблицы “*student_on_course_kn*” добавит всех студентов из города Москва, **НО** с учетом вычитания баллов из задания b). Те баллы студентов должны соответствовать формуле. Пожалуйста приложите команду добавления строк и результат запроса

```
SELECT * FROM student_on_course_kn ORDER BY id;
```

в виде скриншота к заданию.

Код запроса:

```
INSERT TABLE student_on_course_kn AS  
  
SELECT soc.*  
  
FROM student_on_course soc  
  
JOIN student s ON soc.student_id = s.id  
  
WHERE s.city = 'Москва';
```

Результат работы:

Рис. 1. Результат выполнения запроса на добавление.

	id bigint	student_id bigint	course_id bigint	student_rating integer	finished_date date
1	10	10	10	65	[null]
2	20	10	20	68	[null]
3	30	10	40	35	[null]
4	40	20	50	95	2022-12-12
5	70	40	10	66	[null]
6	80	40	20	68	[null]
7	90	40	50	76	[null]
8	100	50	10	12	[null]
9	110	50	20	21	[null]
10	120	50	30	56	[null]
11	130	50	40	92	[null]

Рис. 2. Обновленная таблица student_on_course_kn

- e. Выведите имена студентов, рейтинг по курсу, учебное заведение студента, соответствующее имя курса, учебное заведение, которое проводит курс одним SQL запросом, используя новую таблицу “*student_on_course_kn*”. Результат отсортируйте по имени студента и названию курса. Обратите внимание на имена

столбцов ниже, которые должны быть указаны как атрибуты вывода вашего SQL запроса.

student_name	student_rating	student_college_name	course_name	course_college_name
Анна Потапова	66	МФТИ	Нейронные сети	МФТИ
Екатерина Андреева	75	МГУ	Актерское мастерство	Сколково
...

Код запроса:

```
SELECT
    student.name AS student_name,
    student_on_course_kn.student_rating AS student_rating,
    college.name AS student_college_name,
    course.name AS course_name,
    college.name AS student_college_name
FROM
    student_on_course_kn
    JOIN student ON student.id = student_on_course_kn.student_id
    JOIN college ON college.id = student.college_id
    JOIN course ON course.id = student_on_course_kn.course_id
ORDER BY
    student_name, course_name;
```

Результат запроса:

	student_name character varying	student_rating integer	student_college_name character varying	course_name character varying	student_college_name character varying
1	Екатерина Андреева	95	МГУ	Актерское мастерство	МГУ
2	Иван Иванов	68	КФУ	Data Mining	КФУ
3	Иван Иванов	65	КФУ	Введение в РСУБД	КФУ
4	Иван Иванов	35	КФУ	Цифровая трансформация	КФУ
5	Ильяс Мухаметшин	68	Иннополис	Data Mining	Иннополис
6	Ильяс Мухаметшин	76	Иннополис	Актерское мастерство	Иннополис
7	Ильяс Мухаметшин	66	Иннополис	Введение в РСУБД	Иннополис
8	Сергей Петров	21	Сколково	Data Mining	Сколково
9	Сергей Петров	12	Сколково	Введение в РСУБД	Сколково
10	Сергей Петров	56	Сколково	Нейронные сети	Сколково
11	Сергей Петров	92	Сколково	Цифровая трансформация	Сколково