

“САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ” (СПбГУ)

Математико-механический факультет

“Математика и компьютерные науки”

Мотовилова Анна Олеговна

“ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ”

Итоговый проект по

“Основам баз данных”

студента группы 23.Б16-мм

Преподаватель:

Шевнин Лев Ярославович

Санкт-Петербург

2024

Содержание

1. Проектирование базы данных.....	2
2. Создание базы данных (Изменение, удаление внесенных значений)	5
3. Запросы на создание, удаление и изменение данных в таблицах	9
4. Индексы, запросы и ограничения	14
5. Триггеры и функции	16

2

1. Проектирование базы данных

Схематическое изображение базы данных (Рисунок1)

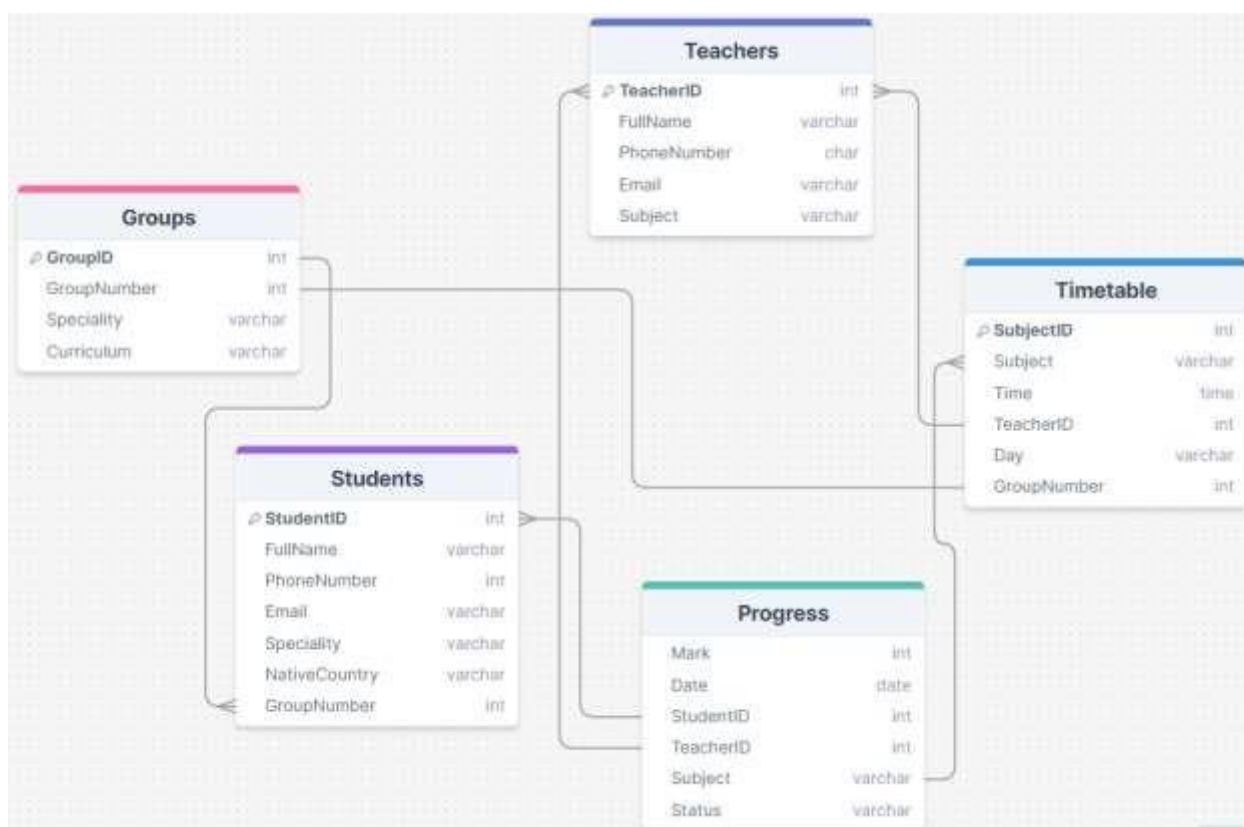


Рисунок1 – Схематическое изображение **Описание:**

Схема представляет пять связанных между собой таблиц.

1. Таблица «Students»

Данная таблица содержит данные студентов. У каждого учащегося есть свой личный StudentID, также таблица содержит такие личные данные, как ФИО, контактные данные, учебные специальности, родные страны, еще есть GroupNumber, благодаря которым будет известны группы, в которых они обучаются. (Связь с таблицей Progress: Многие к 1, связь с таблицей Groups: Многие к 1)

2. Таблица «Groups»

Данная таблица содержит информацию о группах. У каждой группы есть свой GroupID, номер группы, направление обучения этой группы, а также план обучения. (Связь с таблицей Students: 1 к Многим, связь с таблицей Timetable: Многие ко Многим)

3. Таблица «Teachers»

Данная таблица содержит информацию о преподавателях. У каждого преподавателя есть свой личный TeacherID, также в таблице можно найти ФИО, контактные данные и предмет, который ведет данный преподаватель. (Связь с таблицей Progress и таблицей Timetable: Многие к 1)

4. Таблица «Timetable»

Данная таблица содержит информацию о изучаемом предмете. В таблице представлена такая информация, как время и день, в которое проходит лекция, группа, у которой пара и преподаватель, который ее ведет. (Связь с таблицей Progress и таблицей Timetable: Многие к 1)

5. Таблица «Progress»

Данная таблица содержит информацию об успеваемости студентов. В таблице отображается StudentID, предмет, который он сдавал, оценка, которую получил, ID преподавателя, который принимал экзамен, дата проведения и статус стипендии студента после сессии. (Связь с таблицами Students, Teachers, Timetable: 1 к Многим)

2. Создание базы данных (Изменение, удаление внесенных значений)

2.1) Для начала создаем пустые таблицы, в которых указываем название колонки, её тип данных, а потом заполняем их данными. Для этого пишем следующие код с выводом результата (Создаем таблицы с помощью «**CREATE TABLE** table_name», заполняем таблицы значениями с помощью «**INSERT INTO** table_name **VALUES**»):

2.1 Код для таблицы «Groups»

```
CREATE TABLE groups
(
  GroupID integer PRIMARY KEY,
  GroupNumber integer,
  Speciality varchar (255),
  Curriculum varchar (255)
);
INSERT INTO groups
VALUES (156, 16, 'Software Engineering', 'bachelors degree'),
(157, 18, 'Artificial Intelligence', 'bachelors degree'),
(158, 25, 'Machine Learning', 'masters degree'),
(159, 30, 'Data Science', 'PhD'),
(160, 21, 'Big Data', 'bachelors degree');
SELECT * FROM groups
```

	groupid [PK] integer	groupnumber integer	speciality character varying (255)	curriculum character varying (255)
1	156	16	Software Engineering	bachelors degree
2	157	18	Artificial Intelligence	bachelors degree
3	158	25	Machine Learning	masters degree
4	159	30	Data Science	PhD
5	160	21	Big Data	bachelors degree

Рисунок2 – Таблица «Groups» со значениями

2.2 Код для таблицы «Students»

```
CREATE TABLE students
(
  StudentsID integer PRIMARY KEY,
  FullName varchar (255),

  PhoneNumber integer,
  Email varchar (255),
  Speciality varchar (255),
  NativeCountry varchar (255),
  GroupNumber integer
);

INSERT INTO students
VALUES (245, 'Name Surname1', 899933355, 'email1@', 'Software Engineering', 'Russia', 16),
(267, 'Name Surname2', 899922255, 'email2@', 'Artificial Intelligence', 'Belarus', 18),
(268, 'Name Surname3', 778811223, 'email3@', 'Machine Learning', 'Kazakhstan', 25),
(269, 'Name Surname4', 667700122, 'email4@', 'Data Science', 'USA', 30),
(270, 'Name Surname5', 556699011, 'email5@', 'Big Data', 'Russia', 21);

SELECT * FROM students
```

	studentsid [PK] integer	fullname character varying (255)	phonenumber integer	email character varying (255)	speciality character varying (255)	nativecountry character varying (255)	groupnumber integer
1	245	Name Surname1	899933355	email1@	Software Engineering	Russia	16
2	267	Name Surname2	899922255	email2@	Artificial Intelligence	Belarus	18
3	268	Name Surname3	778811223	email3@	Machine Learning	Kazakhstan	25
4	269	Name Surname4	667700122	email4@	Data Science	USA	30
5	270	Name Surname5	556699011	email5@	Big Data	Russia	21

Рисунок3 – Таблица «Students» со значениями

2.3 Код для таблицы «Teachers»

```
CREATE TABLE teachers
(
  TeacherID integer PRIMARY KEY,
  FullName varchar (255),
  PhoneNumber integer,
  Email varchar (255),
  Subject varchar (255)
);

INSERT INTO teachers
VALUES (897, 'Name Surname9', 89992345, 'email@9', 'Algebra'),
(345, 'Name Surname10', 89994567, 'email@10', 'Mathematical Analysis'),
(346, 'Name Surname11', 770811223, 'email@11', 'Machine Learning'),
(347, 'Name Surname12', 660700122, 'email@12', 'Data Science'),
(368, 'Name Surname13', 550699011, 'email@13', 'Big Data');

SELECT * FROM teachers
```

	teacherid [PK] integer	fullname character varying (255)	phonenumber integer	email character varying (255)	subject character varying (255)
1	897	Name Surname9	89992345	email@9	Algebra
2	345	Name Surname10	89994567	email@10	Mathematical Analysis
3	346	Name Surname11	770811223	email@11	Machine Learning
4	347	Name Surname12	660700122	email@12	Data Science
5	368	Name Surname13	550699011	email@13	Big Data

Рисунок4 – Таблица «Teachers» со значениями

2.4 Код для таблицы «Timetable»

```
CREATE TABLE timetable
```

```
(
```

```
    SubjectID integer PRIMARY KEY,
```

```
    Subject varchar (255),
```

```
    Times time,
```

```
    TeacherID integer,
```

```
    Day varchar,
```

```
    GroupNumber integer
```

```
);
```

```
INSERT INTO timetable
```

```
VALUES (765,'Algebra','13:00:00', 897, 'Monday', 16),
```

```
(987,'Mathematical Analysis','11:00:00',345, 'Tuesday', 18),
```

```
(978,'Machine Learning','17:00:00',346, 'Friday', 25 ),
```

```
(959,'Data Science','9:00:00',346, 'Saturday', 21 );
```

```
SELECT * FROM timetable
```

	subjectid [PK] integer	subject character varying (255)	times time without time zone	teacherid integer	day character varying	groupnumber integer
1	765	Algebra	13:00:00	897	Monday	16
2	987	Mathematical Analysis	11:00:00	345	Tuesday	18
3	978	Machine Learning	17:00:00	346	Friday	25
4	959	Data Science	09:00:00	346	Saturday	21

Рисунок5 – Таблица «Timetable» со значениями

2.5 Код для таблицы «Progress»

```
CREATE TABLE progress
```

```
(  
    Mark integer,  
    Dates date,  
    StudenID integer,  
    TeacherID integer,  
    Subject varchar (255),  
    Status varchar (255)  
);
```

```
INSERT INTO progress
```

```
VALUES (5,'2024-06-04',245,897,'Algebra','You have grant'),  
(4,'2024-06-17',267,345,'Mathematical Analysis', 'You have grant'),  
(3,'2024-06-19',268,346,'Machine Learning', 'You can lose grant'),  
(2,'2024-06-18',270,368,'Big Data', 'You have lost grant'),  
(4,'2024-06-02',269,347,'Data Sience', 'You have grant');
```

```
SELECT * FROM progress
```

	mark integer	dates date	studenid integer	teacherid integer	subject character varying (255)	status character varying (255)
1	5	2024-06-04	245	897	Algebra	You have grant
2	4	2024-06-17	267	345	Mathematical Analysis	You have grant
3	3	2024-06-19	268	346	Machine Learning	You can lose grant
4	2	2024-06-18	270	368	Big Data	You have lost grant
5	4	2024-06-02	269	347	Data Sience	You have grant

Рисунок6 – Таблица «Progress» со значениями

2.2) Изменение внесенных значений

```
UPDATE timetable
```

```
SET times = '09:30:00'
```

```
WHERE times = '09:00:00';
```

	subjectid integer	subject character varying (255)	times time without time zone	teacherid integer	day character varying	groupnumber integer
1	765	Algebra	13:00:00	897	Monday	16
2	987	Mathematical Analysis	11:00:00	345	Tuesday	18
3	978	Machine Learning	17:00:00	346	Friday	25
4	959	Data Science	09:30:00	346	Saturday	21

Рисунок7 – Таблица «Timetable» с измененным значением times в 4 строке

2.3) Удаление значений из таблицы

```
DELETE FROM progress
WHERE studenID = 267;
SELECT * FROM progress
```

	mark integer	dates date	studenid integer	teacherid integer	subject character varying (255)	status character varying (255)
1	5	2024-06-04	245	897	Algebra	You have grant
2	3	2024-06-19	268	346	Machine Learning	You can lose grant
3	2	2024-06-18	270	368	Big Data	You have lost grant
4	4	2024-06-02	269	347	Data Sience	You have grant

Рисунок8 – Таблица «Progress» после удаления строки, где StudenID = 267

3. Запросы на создание, удаление и изменение данных в таблицах

3.1) Выборка данных (Однотабличная):

(Выбираем данные одной из таблиц по какому-то параметру)

```
SELECT * FROM progress
WHERE status = 'You have grant'
AND mark > 4;
```

	mark integer	dates date	studenid integer	teacherid integer	subject character varying (255)	status character varying (255)
1	5	2024-06-04	245	897	Algebra	You have grant

Рисунок9 – Данные из таблицы «Progress» с нашим запросом

3.2) Выборка данных(Многотабличная):

(Выбираем данные из двух таблиц по какому-то одному параметру)

```
SELECT * FROM progress
INNER JOIN timetable ON progress.subject = timetable.subject
```

	mark integer	date date	studentid integer	teacherid integer	subject character varying (255)	status character varying (255)	subjectid integer	subject character varying (255)	times time without time zone	teacherid integer	day character varying	groupnumber integer
1	5	2024-05-04	245	897	Algebra	You have grant	765	Algebra	13:00:00	897	Monday	16
2	3	2024-05-19	258	346	Machine Learning	You can lose grant	978	Machine Learning	17:00:00	346	Friday	25

Рисунок10– Данные из таблицы «Progress» и «Timetable»с нашим запросом

3.3) Запрос на создание столбца и изменения его значений

(Создаем столбец year в таблице Groups и изменяем его значения с нуля на 2023 год)

```
ALTER TABLE groups
ADD COLUMN year int;
UPDATE groups
SET year = '2023';
SELECT * FROM groups
```

	groupid [PK] integer	groupnumber integer	speciality character varying (255)	curriculum character varying (255)	year integer
1	156	16	Software Engineering	bachelors degree	2023
2	157	18	Artificial Intelligence	bachelors degree	2023
3	158	25	Machine Learning	masters degree	2023
4	159	30	Data Science	PhD	2023
5	160	21	Big Data	bachelors degree	2023

Рисунок11– Данные из таблицы «Groups» с новым столбцом и с новыми его значениями

3.4) Запрос на создание таблицы

(Создаем новую таблицу allID и добавляем значения)

```
CREATE TABLE allID
(GroupID int, StudentsID int, TeacherID int, SubjectID int);
INSERT INTO allID
VALUES (349, 467, 875, 236);
SELECT * FROM allID
```

9

	groupid integer	studentsid integer	teacherid integer	subjectid integer
1	349	467	875	236

Рисунок12 – Данные новой таблицы allID

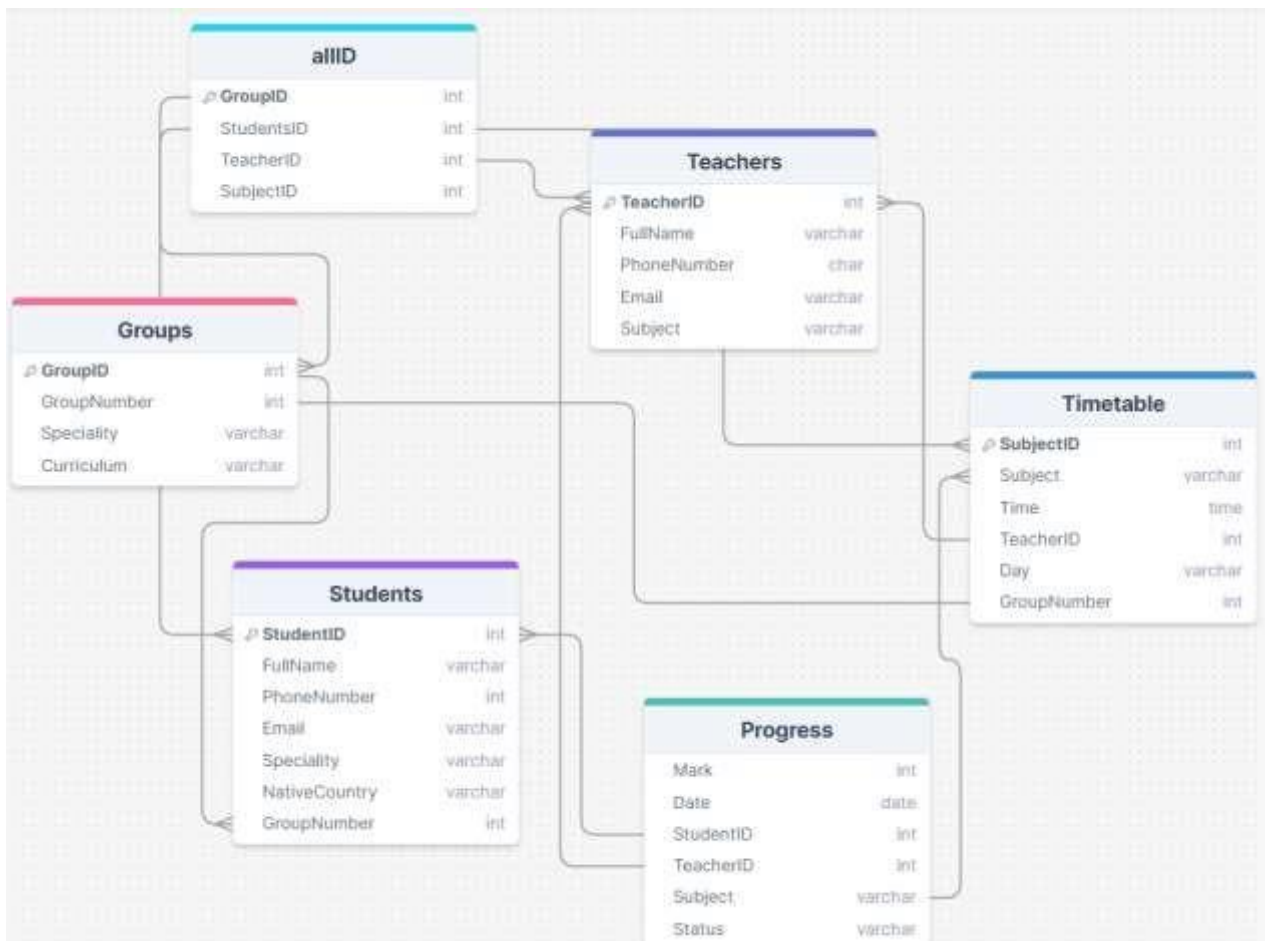


Рисунок13 – Схема после добавления новой таблицы

3.5) Запрос на добавление значения

(Например, нам требуется добавить нового студента в таблицу Students, для этого мы должны заполнить все значения в этой строке, а также добавить это значение в таблицу Progress)

```

INSERT INTO students
VALUES (271, 'NameSurname15', 87654321, 'email@15', 'Big Data', 'Belarus', 23);
INSERT INTO progress
VALUES (4, '2024-06-03', 271, 368, 'Big Data', 'You have grant');
SELECT * FROM students
JOIN progress ON students.studentsID = progress.studenID;
  
```

	studentsid	fullname	phonenumber	email	speciality	nativecountry	groupnumber	mark	dates	studentid	teacherid	subject	status
	integer	character varying (255)	integer	character varying (255)	character varying (255)	character varying (255)	integer	integer	date	integer	integer	character varying (255)	character varying (255)
1	245	Name Surname1	899933333	email1@	Software Engineering	Russia	16	5	2024-06-04	245	317	Algebra	You have grant
2	268	Name Surname2	778811223	email2@	Machine Learning	Kazakhstan	25	3	2024-06-10	268	348	Machine Learning	You can use grant
3	270	Name Surname5	556699011	email5@	Big Data	Russia	21	2	2024-06-18	270	368	Big Data	You have not grant
4	269	Name Surname4	667700122	email4@	Data Science	USA	30	4	2024-06-02	269	347	Data Science	You have grant
5	271	NameSurname15	87654321	email@15	Big Data	Belarus	23	4	2024-06-03	271	368	Big Data	You have grant

Рисунок14 – Данные таблиц «Students» и «Progress» после добавления нового значения

3.6) Запрос на добавление нескольких значений

(Делаем запрос для добавления нескольких новых значений в таблицу allID)

```
INSERT INTO allID
VALUES (156,245, 897,765),
(158,268, 346,978),
(159,270, 368,111),
(160,269, 347,959),
(157,271, 368,111);
SELECT * FROM allID
```

	groupid integer	studentsid integer	teacherid integer	subjectid integer
1	349	467	875	236
2	156	245	897	765
3	158	268	346	978
4	159	270	368	111
5	160	269	347	959
6	157	271	368	111

Рисунок15 – Данные таблицы «allID» после добавления новых значений

3.7) Запрос на удаление значений

(Делаем запрос на удаление значения из таблицы allID)

```
DELETE FROM allID
WHERE groupid = 349;
SELECT * FROM allID ;
```

	groupid integer	studentsid integer	teacherid integer	subjectid integer
1	156	245	897	765
2	158	268	346	978
3	159	270	368	111
4	160	269	347	959
5	157	271	368	111

Рисунок16 – Данные таблицы «allID» после удаления одного из значений

3.8) Вложенные запросы

(Задаем такой запрос, чтобы у нас данные таблицы Subject группировались с данными таблицы Teachers при каком-то условии)

```
SELECT Subject
FROM teachers
WHERE FullName = 'Name Surname9'
GROUP BY subject
```

	subject character varying (255)
1	Algebra

Рисунок17 – Результат вложенного запроса

3.9) Запрос на обновлении записей

(Обновим запись в таблице, заменив одно из значений столбца на другое)

```
UPDATE allID
SET subjectID = 244
WHERE subjectID = 111;
SELECT * FROM allID
```

	groupid integer	studentsid integer	teacherid integer	subjectid integer
1	156	245	897	765
2	158	268	346	978
3	160	269	347	959
4	159	270	368	222
5	157	271	368	222

Рисунок18 – Данные таблицы «allID» после обновления значения столбца

4. Индексы, запросы и ограничения

4.1) Вложенные запросы

4.1.1 (Используем вложенный запрос, чтобы выбрать groupID из таблицы allID, где найдены studentsID, равные 245.)

```
SELECT groupID
FROM allID
WHERE groupID IN (
    SELECT groupID
    FROM allID
    WHERE studentsID = 245
);
```

	groupid integer	
1		156

Рисунок19 – Результат вложенного запроса

4.1.2 (Используем вложенный запрос для вывода ФИО студентов, у которых есть стипендия)

```
SELECT fullname
FROM students
WHERE studentsid IN
(
    SELECT studenid
    FROM progress
    WHERE mark > 3
);
```

	fullname character varying (255)	
1	Name Surname1	
2	Name Surname4	
3	NameSurname15	

Рисунок20 – Результат вложенного запроса

4.2) Индексация

(Проиндексируем 6 столбцов, которые используют чаще, чем остальные, чтобы максимально оптимизировать их)

4.2.1

```
CREATE INDEX idx_studentsID ON students(studentsID);  
CREATE INDEX idx_teachersID ON teachers(teacherID);  
CREATE INDEX idx_groupID ON groups (groupID);  
CREATE INDEX idx_SubjectID ON timetable (SubjectID);
```

13

Эти столбцы индексируем, так как самыми частыми запросами являются вызовы ID, так как по ним можно получить больше всего информации о человеке или предмете.

4.2.2

```
CREATE INDEX idx_FullName ON teachers (FullName);  
CREATE INDEX idx_FullName2 ON students (FullName)
```

Эти столбцы индексируем, так как самая важная информация о человеке, это его имя, но оно уступает по частоте вызова ID, так как могут совпадать.

4.3) Добавление ограничений первичных и внешних ключей или описание существующих

(Создаем функцию FOREIGN KEY для таблицы, у которой повторяется значение со значением другой таблицы, которое как раз является PRIMARY KEY)

4.3.1

```
ALTER TABLE timetable  
ADD CONSTRAINT func_teacherID FOREIGN KEY (teacherID)  
REFERENCES teachers(teacherID )  
ON DELETE CASCADE  
ON UPDATE CASCADE
```

В каждой записи в таблице timetable значения в колонке teacherID должны существовать как идентификаторы в колонке teacherID в таблице teachers.

4.3.2

```
ALTER TABLE progress  
ADD CONSTRAINT func_studentID FOREIGN KEY (studentID)  
REFERENCES students (studentsID )  
ON DELETE CASCADE  
ON UPDATE CASCADE
```

4.3.3

```
ALTER TABLE progress  
ADD CONSTRAINT func_teacherID FOREIGN KEY (teacherID)  
REFERENCES teachers(teacherID )  
ON DELETE CASCADE  
ON UPDATE CASCADE
```

5. Триггеры и функции

Требуется:

1. Триггер, устанавливающий статус (готово/в процессе).
2. Триггер, который обновляет данные при добавлении новых строк или изменении количества.
3. Триггер, который проверяет ограничение на количество (или численное выражение).
4. Вычислительную функцию (принимает аргументы, выдает результат).

(1 и 2 пункт сделан в один код)

5.1,2) Триггер, устанавливающий статус (готово/в процессе), который обновляет данные при добавлении новых строк или изменении количества. Создаем таблицу Exams, в которую позже будут внесены значения.

```
CREATE TABLE exams
(
    studenID integer,
    exam_status varchar
);
```

```
CREATE OR REPLACE FUNCTION update_ex_status()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.mark = 2 THEN
        UPDATE exams SET exam_status = 'не сдан' WHERE studenID = NEW.studenID;
    ELSE
        UPDATE exams SET exam_status = 'сдан' WHERE studenID = NEW.studenID;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER update_ex_status_trigger
AFTER INSERT ON progress
FOR EACH ROW
EXECUTE FUNCTION update_ex_status();
```



```
INSERT INTO progress (mark, studenID)
VALUES (3, 268);
SELECT * FROM exams
```



	studenid 	exam_status 
1	268	сдан

Рисунок21 – Результат триггера, устанавливающего статус

15

5.3) Триггер, который проверяет ограничение на количество (или численное выражение)

```
RETURNS TRIGGER AS
$$
BEGIN
    IF (SELECT COUNT(*) FROM students) >= 100 THEN
        RAISE EXCEPTION 'Row limit exceeded';
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER row_limit_trigger
BEFORE INSERT ON students
FOR EACH ROW
EXECUTE FUNCTION check_row_limit();
```

5.4)

```
CREATE OR REPLACE FUNCTION count_grants (numbers INT)
RETURNS INT AS $$
DECLARE
    c_grants INT;
BEGIN
    SELECT COUNT(*)
    INTO c_grants
    FROM progress
    WHERE status = 'You have grant' ;

    RETURN c_grants;
END;
$$ LANGUAGE plpgsql;

SELECT count_grants(5);
```


	count_grants	
	integer	
1		3

Рисунок21 – Результат после использования триггера