МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №4**

**по курсу «Проектирование баз данных»**

Выполнила: Прудникова А. А.

Группа: М8О-114СВ-24

Преподаватель: Моргунов Е. П.

Москва, 2024

## Задание 2

Посмотрите, какие ограничения уже наложены на атрибуты таблицы «Успеваемость» (progress). Воспользуйтесь командой **\d** утилиты psql. А теперь предложите для этой таблицы ограничение уровня таблицы.

### Запрос

CREATE TABLE students

( record\_book numeric( 5 ),

doc\_ser numeric( 4 ),

doc\_num numeric( 6 ),

name TEXT NOT NULL,

PRIMARY KEY ( record\_book )

);

CREATE TABLE progress

( record\_book numeric( 5 ) NOT NULL,

subject text NOT NULL,

acad\_year text NOT NULL,

term numeric( 1 ) NOT NULL CHECK ( term = 1 OR term = 2 ),

mark numeric( 1 ) NOT NULL CHECK ( mark >= 3 AND mark <= 5 )

DEFAULT 5,

FOREIGN KEY ( record\_book )

REFERENCES students ( record\_book )

ON DELETE CASCADE

ON UPDATE CASCADE

);

ALTER TABLE progress

ADD COLUMN test\_form TEXT NOT NULL

CHECK (test\_form IN ('экзамен', 'зачет')),

ADD CHECK (

(test\_form = 'экзамен' AND mark IN (3, 4, 5))

OR (test\_form = 'зачет' AND mark IN (0, 1))

);

INSERT INTO students (record\_book, doc\_ser, doc\_num, name)

VALUES (12345, 1234, 567890, 'Иван Иванов');

### Запрос

INSERT INTO progress (record\_book, subject, acad\_year, term, mark, test\_form)

VALUES (12345, 'Линейная алгебра и геометрия', '2024', 1, 4, 'экзамен');

INSERT INTO progress (record\_book, subject, acad\_year, term, mark, test\_form)

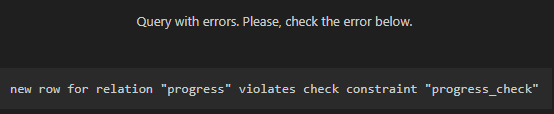
VALUES (12345, 'Основы программирования', '2024', 2, 2, 'экзамен');

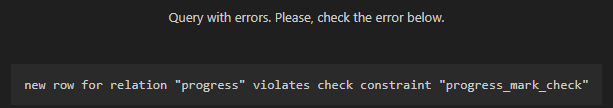
INSERT INTO progress (record\_book, subject, acad\_year, term, mark, test\_form)

VALUES (12345, 'Философия', '2024', 2, 1, 'зачет');

### Результат







Кажется, возникает конфликт с ограничением на оценки зачетов. На оценку одновременно наложено требование, что она равна 0 или 1, и требование, что она равна какому-либо из чисел 3, 4, 5. Теперь удалим изначальное ограничение на оценку и введем команды вставки заново.

### Запрос

ALTER TABLE progress

DROP CONSTRAINT progress\_mark\_check;

INSERT INTO progress (record\_book, subject, acad\_year, term, mark, test\_form)

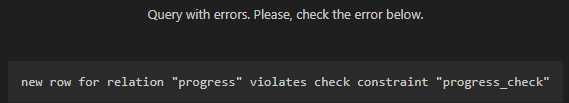
VALUES (12345, 'Философия', '2024', 2, 1, 'зачет');

INSERT INTO progress (record\_book, subject, acad\_year, term, mark, test\_form)

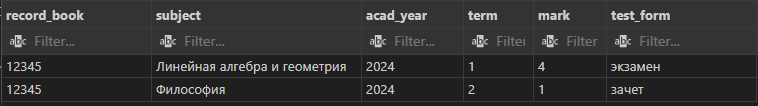
VALUES (12345, 'История', '2024', 1, 3, 'зачет');

### Результат





Проверим итоговое содержание таблицы progress:



## Задание 9

В таблице «Студенты» (students) есть текстовый атрибут name, на который наложено ограничение NOT NULL. Как вы думаете, что будет, если при вводе новой строки в эту таблицу дать атрибуту name в качестве значения пустую строку? Оказывается, эти невидимые значения имеют ненулевую длину. Что делать, чтобы не допустить таких значений-невидимок? Один из способов: возложить проверку таких ситуаций на прикладную программу. А что можно сделать на уровне определения таблицы students? Какое ограничение нужно предложить? В разделе 9.4 документации «Строковые функции и операторы» есть функция trim. Попробуйте воспользоваться ею.

Есть ли подобные слабые места в таблице «Успеваемость» (progress)?

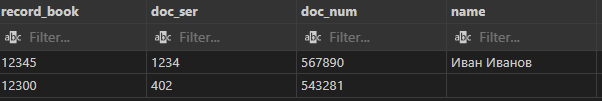
### Запрос

INSERT INTO students ( record\_book, name, doc\_ser, doc\_num )

VALUES ( 12300, '', 0402, 543281 );

### Результат





### Запрос

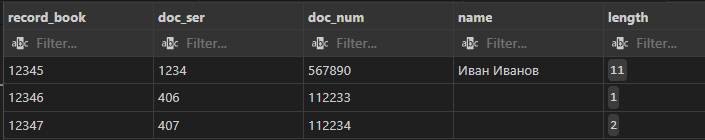
DELETE FROM students WHERE name = '';

INSERT INTO students VALUES ( 12346, 0406, 112233, ' ' );

INSERT INTO students VALUES ( 12347, 0407, 112234, ' ' );

SELECT \*, length( name ) FROM students;

### Результат



### Запрос

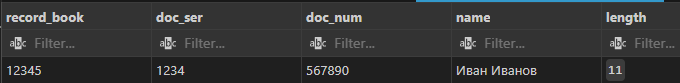
DELETE FROM students

WHERE trim(name) = '';

ALTER TABLE students ADD CHECK (trim(name) <> '');

SELECT \*, length( name ) FROM students;

### Результат



Что касается таблицы progress, проверка на пустые и пробельные значения в столбце с названием предмета может быть полезна и там. В данной таблице есть столбец subject, который также может потребовать аналогичной проверки.

### Запрос

ALTER TABLE progress ADD CHECK (trim(subject) <> '');

### Результат



## Задание 17

Подумайте, какие представления было бы целесообразно создать для нашей базы данных «Авиаперевозки». Необходимо учесть наличие различных групп пользователей, например: пилоты, диспетчеры, пассажиры, кассиры.

Создайте представления и проверьте их в работе.

### Запрос

-- Для пилотов: запланированные или текущие рейсы

CREATE VIEW pilot\_flights AS

SELECT flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport, arrival\_airport

FROM flights

WHERE status IN ('Scheduled', 'In Flight');

SELECT \* FROM pilot\_flights;

-- Для диспетчеров с группировкой по статусу рейсов

CREATE VIEW dispatcher\_flights AS

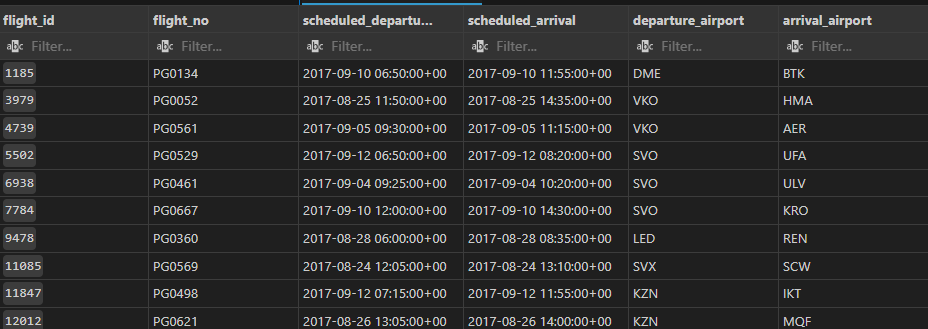
SELECT status, COUNT(flight\_id) AS flight\_count

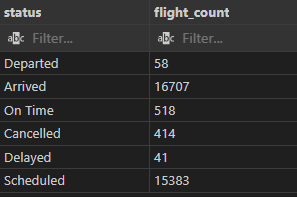
FROM flights

GROUP BY status;

SELECT \* FROM dispatcher\_flights;

### Результат





## Задание 18

Подумайте, какие еще таблицы было бы целесообразно дополнить столбцами типа json/jsonb. Вспомните, что, например, в таблице «Билеты» (tickets) уже есть столбец такого типа — contact\_data. Выполните модификации таблиц и измените в них одну-две строки для проверки правильности ваших решений.

### Запрос

ALTER TABLE aircrafts\_data ADD COLUMN specifications jsonb;

UPDATE aircrafts\_data

SET specifications =

'{ "crew": 2,

"engines": { "type": "IAE V2500",

"num": 2

}

}'::jsonb

WHERE aircraft\_code = '320';

### Результат



### Запрос

ALTER TABLE pilots ADD COLUMN qualifications jsonb;

UPDATE pilots

SET qualifications =

'{

"licenses": ["ATPL", "CPL"],

"experience\_years": 10,

"rating": "Captain"

}'::jsonb

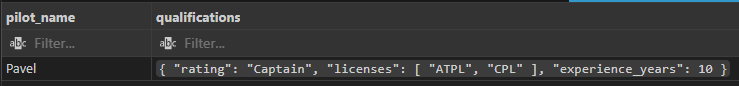
WHERE pilot\_name = 'Pavel';

SELECT pilot\_name, qualifications

FROM pilots

WHERE pilot\_name = 'Pavel';

### Результат



### Запрос

ALTER TABLE flights ADD COLUMN weather\_conditions jsonb;

UPDATE flights

SET weather\_conditions =

'{

"departure": { "temperature": -5, "wind\_speed": 15, "condition": "snow" },

"arrival": { "temperature": 2, "wind\_speed": 10, "condition": "clear" }

}'::jsonb

WHERE flight\_id = 24;

SELECT flight\_no, weather\_conditions

FROM flights

WHERE flight\_id = 24;

### Результат

