

Домашняя работа №4

Попов Матвей, М8О-114СВ-24

Задание 2

Проверьте, как будет работать новое ограничение в модифицированной таблице progress. Для этого выполните команды INSERT, как удовлетворяющие ограничению, так и нарушающие его.

В таблице уже было ограничение на допустимые значения атрибута mark. Как вы думаете, не будет ли оно конфликтовать с новым ограничением? Проверьте эту гипотезу. Если ограничения конфликтуют, тогда удалите старое ограничение и снова попробуйте добавить строки в таблицу.

Запрос

```
ALTER TABLE progress
  ADD COLUMN test_form text,
  ADD CHECK (
    ( test_form = 'экзамен' AND mark IN ( 3, 4, 5 ))
    OR
    ( test_form = 'зачет' AND mark IN ( 0, 1 ))
  );

INSERT INTO progress (record_book, subject, acad_year, term,
mark, test_form)
VALUES (12345, 'МАТАН', '2020', 2, 3, 'экзамен');

INSERT INTO progress (record_book, subject, acad_year, term,
mark, test_form)
VALUES (54321, 'БД', '2024', 1, 4, 'зачет');

INSERT INTO progress (record_book, subject, acad_year, term,
mark, test_form)
VALUES (54321, 'ИИ', '2024', 1, 1, 'зачет');

ALTER TABLE progress
DROP CONSTRAINT progress_mark_check;

ALTER TABLE progress
  ADD CHECK (
    ( test_form = 'экзамен' AND mark IN ( 3, 4, 5 ))
    OR
```

```
( test_form = 'зачет' AND mark IN ( 0, 1 ))  
);
```

```
INSERT INTO progress (record_book, subject, acad_year, term,  
mark, test_form)  
VALUES (54321, 'ИИ', '2024', 1, 1, 'зачет');
```

Результат

Результаты запросов на вставку:

```
demo: bookings, public> INSERT INTO progress (record_book, subject, acad_year, term, mark, test_form)  
VALUES (12345, 'МАТАН', '2020', 2, 3, 'экзамен')
```

```
[23514] ERROR: new row for relation "progress" violates check constraint "progress_check"  
Detail: Failing row contains (54321, БД, 2024, 1, 4, зачет).
```

```
[23514] ERROR: new row for relation "progress" violates check constraint "progress_mark_check"  
Detail: Failing row contains (54321, ИИ, 2024, 1, 1, зачет).
```

```
demo: bookings, public> INSERT INTO progress (record_book, subject, acad_year, term, mark, test_form)  
VALUES (54321, 'ИИ', '2024', 1, 1, 'зачет')
```

Задание 9

Добавив ограничение, попробуйте теперь вставить в таблицу `students` строку (row), в которой значение атрибута `name` было бы пустой строкой (string).

Давайте продолжим эксперименты и предложим в качестве значения атрибута `name` строку, содержащую сначала один пробел, а потом — два пробела.

```
INSERT INTO students VALUES ( 12346, ' ', 0406, 112233 );  
INSERT INTO students VALUES ( 12347, '  ', 0407, 112234 );
```

Для того чтобы «увидеть» эти пробелы в выборке, сделаем так:

```
SELECT *, length( name ) FROM students;
```

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. Если вы еще не изучили команду `ALTER TABLE`, то удалите таблицу `students` и создайте ее заново с учетом нового ограничения, а если уже познакомились с ней, то сделайте так:

```
ALTER TABLE students ADD CHECK (...);
```

Запрос

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

```
INSERT INTO students VALUES ( 12346,' ', 0406, 112233 );  
INSERT INTO students VALUES ( 12347,'  ', 0407, 112234 );
```

```
SELECT *, length( name ) FROM students;
```

```
TRUNCATE TABLE students;
```

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

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

```
SELECT *, length( name ) FROM students;
```

Результат

Результаты запросов на выборку:

	record_book	name	doc_ser	doc_num	length
1	12346		406	112233	1
2	12347		407	112234	1

record_book	name	doc_ser	doc_num	length

Задание 17

Представления могут быть, условно говоря, *вертикальными* и *горизонтальными*. При создании вертикального представления в список его столбцов включается лишь часть столбцов базовой таблицы (таблиц). Например:

```
CREATE VIEW airports_names AS
  SELECT airport_code, airport_name, city
  FROM airports;
```

```
SELECT * FROM airports_names;
```

В горизонтальное представление включаются не все строки базовой таблицы (таблиц), а производится их отбор с помощью фраз WHERE или HAVING.

Например:

```
CREATE VIEW siberian_airports AS
  SELECT * FROM airports
  WHERE city = 'Новосибирск' OR city = 'Кемерово';
```

```
SELECT * FROM siberian_airports;
```

Конечно, вполне возможен и смешанный вариант, когда ограничивается как список столбцов, так и множество строк при создании представления.

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

Результат

	aircraft_code	range
1	773	11100
2	763	7900
3	320	5700
4	321	5600
5	319	6700
6	733	4200
7	CN1	1200
8	CR2	2700
9	SU9	6000

	aircraft_code	model	range
1	773	{"en": "Boeing 777-300", "ru": "Боинг 777-300"}	11100
2	763	{"en": "Boeing 767-300", "ru": "Боинг 767-300"}	7900
3	319	{"en": "Airbus A319-100", "ru": "Айрбас А319-100"}	6700
4	SU9	{"en": "Sukhoi Superjet-100", "ru": "Сухои Суперджет-100"}	6000

Задание 18

Задание. 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 airports_data
ADD COLUMN location jsonb;
```

```
UPDATE airports_data
SET location =
  '{"city": "Москва",
    "country": "Россия",
    "coordinates": {
      "latitude": 37.4,
      "longitude": 56.0
    }}'
```

```
WHERE airport_code = 'SVO';
```

```
SELECT location FROM airports_data  
WHERE airport_code = 'SVO';
```

```
SELECT location->'city' FROM airports_data  
WHERE airport_code = 'SVO';
```

```
SELECT location #> '{coordinates, longitude}' FROM airports_data  
WHERE airport_code = 'SVO';
```

Результат

	location ▾
1	{"city": "Москва", "country": "Россия", "coordinates": {"latitude": 37.4, "longitude": 56.0}}

	?column? ▾
1	"Москва"

	?column? ▾
1	56.0