

# Модификация структуры таблиц в СУБД PostgreSQL

Управление данными

Кафедра АПУ СПбГЭТУ «ЛЭТИ»

# Модификация структуры таблиц

---

Если в таблице еще нет данных, то изменить структуру легче путем пересоздания и определения структуры заново.

Но если таблицы содержат большое количество строк, то пересоздать их не всегда возможно. В этом случае используют команду ALTER TABLE.

ALTER TABLE имя\_таблицы

команда,

команда,

.....,

команда;

Основные команды:

ADD COLUMN, DROP COLUMN, ALTER COLUMN,

ADD CHECK,

ADD CONSTRAINT, DROP CONSTRAINT,

RENAME COLUMN, RENAME CONSTRAINT

# Модификация ограничения NOT NULL

*Пример:*

```
CREATE TABLE student
( id serial,
  name varchar(20) NOT NULL UNIQUE,
  mark numeric(1) CHECK (mark >= 3 AND mark <= 5) DEFAULT 3,
  code integer NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (code) REFERENCES student_group (group_code)
);
```

*--Удаление ограничения NOT NULL, заданного на уровне поля*

```
ALTER TABLE student ALTER COLUMN name DROP NOT NULL;
```

*--Установка ограничения NOT NULL на уровне поля*

```
ALTER TABLE student ALTER COLUMN name SET NOT NULL;
```

*--Установка ограничения NOT NULL на уровне таблицы*

```
ALTER TABLE student ADD CONSTRAINT not_null_name CHECK (name IS NOT NULL);
```

*--Удаление ограничения NOT NULL, заданного на уровне таблицы*

```
ALTER TABLE student DROP CONSTRAINT not_null_name;
```

# Модификация ограничения CHECK

*Пример:*

Таблица student

```
( id serial,  
  name varchar(20) NOT NULL UNIQUE,  
  mark numeric(1) CHECK (mark >= 3 AND mark <= 5) DEFAULT 3,  
  code integer NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (code) REFERENCES student_group (group_code)  
);
```

*--Изменение ограничения CHECK для поля*

Проверок CHECK для поля может быть несколько.

Каждый CHECK, являясь ограничением, имеет имя, либо явно заданное при его формировании, либо заданное системой.

Если надо добавить новый CHECK, то выполняется команда ADD CHECK.

Если надо заменить один CHECK на другой, то надо сначала удалить ненужный CHECK, используя команду DROP CONSTRAINT, а потом добавить новый CHECK командой ADD CHECK.

```
ALTER TABLE student DROP CONSTRAINT student_mark_check;      -- Имя ограничения,  
                                                                заданное системой
```

```
ALTER TABLE student ADD CHECK (mark>=2 AND mark<=5);
```

## Модификация DEFAULT и переименование поля

### Пример:

Таблица student

```
( id serial,  
  name varchar(20) NOT NULL UNIQUE,  
  mark numeric(1) CHECK (mark >= 2 AND mark <= 5) DEFAULT 3,  
  code integer NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (code) REFERENCES student_group (group_code)  
);
```

*--Удаление DEFAULT для поля*

```
ALTER TABLE student ALTER COLUMN mark DROP DEFAULT;
```

*--Установка DEFAULT для поля*

```
ALTER TABLE student ALTER COLUMN mark SET DEFAULT 4;
```

*--Переименование поля*

```
ALTER TABLE student RENAME COLUMN name TO student_name;
```

```
ALTER TABLE student RENAME COLUMN mark TO student_mark;
```

# Модификация ограничений UNIQUE и PRIMARY KEY

*Пример:*

Таблица student

```
( id serial,  
  student_name varchar(20) NOT NULL UNIQUE,  
  student_mark numeric(1) CHECK (student_mark >= 2 AND student_mark <= 5) DEFAULT 4,  
  code integer NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (code) REFERENCES student_group (group_code)  
);
```

*--Удаление ограничения UNIQUE для поля*

```
ALTER TABLE student DROP CONSTRAINT student_name_key; --Имя ограничения,  
                                                    заданного системой
```

*--Установка ограничения UNIQUE для поля*

```
ALTER TABLE student ADD CONSTRAINT student_name_unique UNIQUE (student_name);
```

*--Удаление PRIMARY KEY*

```
ALTER TABLE student DROP CONSTRAINT student_pkey; --Имя PRIMARY KEY,  
                                                    заданное системой
```

*--Установка PRIMARY KEY*

```
ALTER TABLE student ADD CONSTRAINT student_pkey PRIMARY KEY (id);
```

## Ввод и удаление поля, изменение типа данных поля

### Пример:

Таблица student

```
( id serial,  
  student_name varchar(20) NOT NULL UNIQUE,  
  student_mark numeric(1) CHECK (student_mark >= 2 AND student_mark <= 5) DEFAULT 4,  
  code integer NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (code) REFERENCES student_group (group_code)  
);
```

*--Ввод нового поля*

```
ALTER TABLE student ADD COLUMN birthday date NOT NULL;
```

*--Удаление поля*

```
ALTER TABLE student DROP COLUMN birthday;
```

*--Изменение типа данных для поля*

```
ALTER TABLE student ALTER COLUMN student_name SET DATA TYPE varchar(50);
```

Когда типы данных изменяются в пределах одной группы, например, оба символьные или оба числовые, то проблем обычно не возникает. Но если типы данных относятся к разным группам, то потребуются дополнительные действия.

## Изменение типа данных поля

*Пример:*

```
CREATE TABLE student_group  
( group_code serial PRIMARY KEY,  
  group_number varchar(4)  
);
```

*--Ввод нового ограничения CONSTRAINT для поля*

*--Ввод нового поля*

```
ALTER TABLE student_group  
  ADD CONSTRAINT student_group_unique UNIQUE (group_number),  
  ADD COLUMN group_type varchar(15),  
  ADD CONSTRAINT group_type_check CHECK (group_type IN ('бакалавриат',  
                                                         'магистратура', 'специалитет'));
```

*--Ввод данных в таблицу*

```
INSERT INTO student_group (group_number, group_type)  
  VALUES ('1111', 'магистратура'),  
         ('2222', 'бакалавриат'),  
         ('3333', 'специалитет');
```



## Изменение типа данных поля

*Пример:*

Таблица student\_group  
( group\_code serial PRIMARY KEY,  
group\_number varchar(4),  
group\_type varchar(15),

group_code	group_number	group_type
1	1111	магистратура
2	2222	бакалавриат
3	3333	специалитет

CONSTRAINT group\_type\_check CHECK (group\_type IN ('бакалавриат',  
'магистратура', 'специалитет')));

Задача – выделить типы групп в справочную таблицу.

```
CREATE TABLE group_type          --Создание справочной таблицы
( group_type_id integer PRIMARY KEY,
  group_type_name varchar(15) NOT NULL UNIQUE
);
```

```
INSERT INTO group_type          --Заполнение справочной таблицы
VALUES (1, 'бакалавриат'),
       (2, 'магистратура'),
       (3, 'специалитет');
```

## Изменение типа данных поля

*Пример:*

Таблица student\_group

```
( group_code serial PRIMARY KEY,  
  group_number varchar(4),  
  group_type varchar(15),
```

```
  CONSTRAINT group_type_check CHECK (group_type IN ('бакалавриат',  
                                                    'магистратура', 'специалитет')));
```

group_code	group_number	group_type
1	1111	магистратура
2	2222	бакалавриат
3	3333	специалитет

Таблица group\_type

```
( group_type_id integer PRIMARY KEY,  
  group_type_name varchar(15) NOT NULL UNIQUE  
);
```

group_type_id	group_type_name
1	бакалавриат
2	магистратура
3	специалитет

```
ALTER TABLE student_group      --Изменение типа данных и данных
```

```
  DROP CONSTRAINT group_type_check,
```

```
  ALTER COLUMN group_type SET DATA TYPE integer
```

```
  USING (CASE WHEN group_type = 'бакалавриат' THEN 1
```

```
            WHEN group_type = 'магистратура' THEN 2
```

```
            WHEN group_type = 'специалитет' THEN 3
```

```
  END);
```

## Изменение типа данных поля

*Пример:*

Таблица student\_group

```
( group_code serial PRIMARY KEY,  
  group_number varchar(4),  
  group_type integer
```

);

Таблица group\_type

```
( group_type_id integer PRIMARY KEY,  
  group_type_name varchar(15) NOT NULL UNIQUE
```

);

group_code	group_number	group_type
1	1111	2
2	2222	1
3	3333	3

group_type_id	group_type_name
1	бакалавриат
2	магистратура
3	специалитет

```
ALTER TABLE student_group      --Определение внешнего ключа  
  ADD FOREIGN KEY (group_type)  
    REFERENCES group_type (group_type_id);
```

```
ALTER TABLE student_group      -- Переименование столбца с типом группы  
  RENAME COLUMN group_type TO group_type_id;
```