

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**ФГБОУ ВО «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

ПРАКТИЧЕСКАЯ РАБОТА №3

**ВЫПОЛНИЛА: Юрьева Марина
ГРУППА: УБ-02**

**ВОРОНЕЖ
2023**

Оператор ALTER TABLE используется для изменения определения существующей таблицы, такой как изменение структуры таблицы, изменение типов данных столбцов, добавление ограничений.

Так, при помощи данного оператора поставим ограничение NOT NULL у нужных столбцов, используя команду:

“ALTER TABLE ‘Название таблицы’ ALTER COLUMN ‘Название столбца’ SET NOT NULL”.

```
apтека=# ALTER TABLE Lekarstva ALTER COLUMN name_lekarstva SET NOT NULL;
ALTER TABLE
apтека=# ALTER TABLE Lekarstva ALTER COLUMN price SET NOT NULL;
ALTER TABLE
apтека=# \d Lekarstva
          Таблица "public.lekarstva"
+-----+-----+-----+-----+-----+
| Столбец | Тип | Правило сортировки | Допустимость NULL | По умолчанию |
+-----+-----+-----+-----+-----+
| id_lekarstva | integer | | not null | |
| name_lekarstva | character varying(30) | | not null | |
| price | character varying(20) | | not null | |
| expiration_date | date | | not null | |
Индексы:
    "lekarstva_pkey" PRIMARY KEY, btree (id_lekarstva)
Ссылки извне:
    TABLE "buyers" CONSTRAINT "buyers_id_lekarstva_fkey" FOREIGN KEY (id_lekarstva) REFERENCES lekarstva(id_lekarstva)
    TABLE "registration_log" CONSTRAINT "registration_log_id_lekarstva_fkey" FOREIGN KEY (id_lekarstva) REFERENCES lekarstva(id_lekarstva)
```

```
apтека=# ALTER TABLE Buyers ALTER COLUMN fio SET NOT NULL;
ALTER TABLE
apтека=# ALTER TABLE Buyers ALTER COLUMN sale SET NOT NULL;
ALTER TABLE
apтека=# \d Buyers
          Таблица "public.buyers"
+-----+-----+-----+-----+-----+
| Столбец | Тип | Правило сортировки | Допустимость NULL | По умолчанию |
+-----+-----+-----+-----+-----+
| id_buyers | integer | | not null | |
| fio | character varying(60) | | not null | |
| sale | character varying(15) | | not null | |
| time_of_purchase | date | | not null | |
| id_lekarstva | integer | | not null | |
Индексы:
    "buyers_pkey" PRIMARY KEY, btree (id_buyers)
Ограничения внешнего ключа:
    "buyers_id_lekarstva_fkey" FOREIGN KEY (id_lekarstva) REFERENCES lekarstva(id_lekarstva)
Ссылки извне:
    TABLE "registration_log" CONSTRAINT "registration_log_id_buyers_fkey" FOREIGN KEY (id_buyers) REFERENCES buyers(id_buyers)
```

```
apтека=# ALTER TABLE Workers ALTER COLUMN full_name SET NOT NULL;
ALTER TABLE
apтека=# ALTER TABLE Workers ALTER COLUMN passport SET NOT NULL;
ALTER TABLE
apтека=# ALTER TABLE Workers ALTER COLUMN phone SET NOT NULL;
ALTER TABLE
apтека=# ALTER TABLE Workers ALTER COLUMN work_experience SET NOT NULL;
ALTER TABLE
apтека=# \d Workers
          Таблица "public.workers"
+-----+-----+-----+-----+-----+
| Столбец | Тип | Правило сортировки | Допустимость NULL | По умолчанию |
+-----+-----+-----+-----+-----+
| id_workers | integer | | not null | |
| full_name | character varying(60) | | not null | |
| passport | character varying(20) | | not null | |
| phone | character varying(20) | | not null | |
| work_experience | character varying(20) | | not null | |
Индексы:
    "workers_pkey" PRIMARY KEY, btree (id_workers)
Ссылки извне:
    TABLE "registration_log" CONSTRAINT "registration_log_id_workers_fkey" FOREIGN KEY (id_workers) REFERENCES workers(id_workers)
```

Также, при помощи данного оператора можно добавлять (удалять) новые столбцы следующей командой:

“ALTER TABLE ‘Название таблицы’ ADD (DROP) COLUMN ‘Название столбца’ ‘Тип данных’”.

```
apтека=# ALTER TABLE Lekarstva ADD COLUMN month varchar(15);
ALTER TABLE
apтека=# \d lekarstva
```

Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
id_lekarstva	integer		not null	
name_lekarstva	character varying(30)		not null	
price	character varying(20)		not null	
expiration_date	date		not null	
month	character varying(15)			

```
apтека=# ALTER TABLE Lekarstva DROP COLUMN month;
ALTER TABLE
apтека=# \d lekarstva
```

Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
id_lekarstva	integer		not null	
name_lekarstva	character varying(30)		not null	
price	character varying(20)		not null	
expiration_date	date		not null	

Для того, чтобы вывести выполнить запрос для выборки используем оператор **SELECT**. Так, выведем ID покупателя и его ФИО покупающие лекарства под id ‘1542’ из таблицы Buyers.

```
apтека=# SELECT id_buyers, fio FROM Buyers WHERE id_lekarstva = '1542';
 id_buyers |          fio
-----+-----
          2 | Аистова Елизавета Андреевна
          3 | Моисеев Вадим Олегович
(2 строки)
```

Используя **ORDER BY ‘Название колонки’ ASC (DESC)**, можно вывести результаты в порядке возрастания (убывания). Так, выведем ID работников и их стаж заезда в период от 1 до 5 лет в порядке возрастания в таблице Workers.

```

аптека=# SELECT id_workers, work_experience FROM Workers WHERE work_experience BETWEEN '1' AND '5' ORDER BY work_experience ASC;
 id_workers | work_experience
-----+-----
          2 | 2 года
          3 | 3 года
          1 | 4 года
(3 строки)

```

Чтобы узнать имена этих ID выполним следующую команду:

```

аптека=# SELECT id_workers, full_name FROM Workers WHERE id_workers in (2, 3, 1);
 id_workers |      full_name
-----+-----
          1 | Сапунов Никита Сергеевич
          2 | Магелад Максим Олегови
          3 | Паринова Нелли Владимировна
(3 строки)

```

Для того, чтобы вывести всех гостей, чья фамилия начинается с определённой буквы, можно использовать следующую команду, где “%” обозначает последовательность из нескольких произвольных символов:

```

аптека=# SELECT * FROM BUYERS WHERE fio LIKE 'C%';
 id_buyers |      fio      | sale | time_of_purchase | id_lekarstva
-----+-----+-----+-----+-----
          1 | Соболева Елизавета Сергеевна | 2%   | 2023-03-13       | 1324
          5 | Самцова Елизавета Михайловна | 0%   | 2023-03-14       | 1324
          7 | Старухин Александр Геннадьевич | 3%   | 2023-03-15       | 1213
(3 строки)

```

Также, используя функция **COUNT**, можно вывести количество строк из определённой таблицы, а функции **MIN** и **MAX** выводят минимальные и максимальные значения из таблицы.

```

аптека=# SELECT COUNT(*) FROM Lekarstva;
count
-----
          5
(1 строка)

```

```

аптека=# SELECT MIN(price), MAX(price) FROM Lekarstva;
 min | max
-----+-----
111rub | 572rub
(1 строка)

```

Для вставки данных в таблицу, можно использовать оператор **INSERT**, указав имена столбцов, которым будут присвоены значения:

```

аптека=# INSERT INTO Lekarstva (
аптека=# id_lekarstva, name_lekarstva, price, expiration_date)
аптека=# VALUES ('1', 'Rinza', '349rub', '13-05-2026');
INSERT 0 1
аптека=# SELECT * FROM Lekarstva;
 id_lekarstva | name_lekarstva | price  | expiration_date
-----+-----+-----+-----
      1122 | Aveloks        | 554rub | 2028-02-24
      1213 | Trekrezan      | 572rub | 2025-01-14
      1486 | Nize           | 246rub | 2026-10-06
      1542 | Terafly        | 293rub | 2025-08-21
      1324 | Tizin          | 111rub | 2028-12-01
           1 | Rinza          | 349rub | 2026-05-13
(6 строк)

```

Используя **UPDATE** можно изменить данные как одного столбца, так и сразу нескольких:

```

аптека=# UPDATE Lekarstva SET price = '700rub' WHERE id_lekarstva = 1;
UPDATE 1
аптека=# SELECT * FROM Lekarstva;
 id_lekarstva | name_lekarstva | price  | expiration_date
-----+-----+-----+-----
      1122 | Aveloks        | 554rub | 2028-02-24
      1213 | Trekrezan      | 572rub | 2025-01-14
      1486 | Nize           | 246rub | 2026-10-06
      1542 | Terafly        | 293rub | 2025-08-21
      1324 | Tizin          | 111rub | 2028-12-01
           1 | Rinza          | 700rub | 2026-05-13
(6 строк)

```

```

аптека=# UPDATE Lekarstva SET price = '999rub', name_lekarstva = 'Ingavirin' WHERE id_lekarstva = 1;
UPDATE 1
аптека=# SELECT * FROM Lekarstva;
 id_lekarstva | name_lekarstva | price  | expiration_date
-----+-----+-----+-----
      1122 | Aveloks        | 554rub | 2028-02-24
      1213 | Trekrezan      | 572rub | 2025-01-14
      1486 | Nize           | 246rub | 2026-10-06
      1542 | Terafly        | 293rub | 2025-08-21
      1324 | Tizin          | 111rub | 2028-12-01
           1 | Ingavirin      | 999rub | 2026-05-13
(6 строк)

```

Для удаления данных из таблицы используется оператор **DELETE**:

```

аптека=# DELETE FROM Lekarstva WHERE id_lekarstva = 1;
DELETE 1
аптека=# SELECT * FROM Lekarstva;
 id_lekarstva | name_lekarstva | price  | expiration_date
-----+-----+-----+-----
      1122 | Aveloks        | 554rub | 2028-02-24
      1213 | Trekrezan      | 572rub | 2025-01-14
      1486 | Nize           | 246rub | 2026-10-06
      1542 | Terafly        | 293rub | 2025-08-21
      1324 | Tizin          | 111rub | 2028-12-01
(5 строк)

```