

## Лабораторная работа №2

Тема: проектирование реляционной базы данных PostgreSQL

### Вариант 3

Цель работы: Получение практических навыков работы с СУБД и языком SQL (создание и изменения таблиц).

Задание:

- 1) В созданной на предыдущей лабораторной работе базе данных дополните таблицы ограничениям CHECK, DEFAULT, NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY;
- 2) внести изменения в схему базы, используя операторы ALTER TABLE; и DROP TABLE;
- 3) создайте новую таблицу (не менее трех полей);
- 4) добавьте в нее новый столбец;
- 5) удалите второй столбец из новой таблицы;
- 6) удалите все таблицу;

Чтобы добавить ограничения в ранее созданную базу данных, следует создать новую таблицу путем использования оператора CREATE TABLE и воспользоваться оператором ALTER TABLE и DROP TABLE.

ALTER TABLE – оператор, меняющий определение таблицы путем добавлением, переопределением, удалением столбца.

DROP TABLE – оператор, удаляющий таблицу.

### Шаг 1. Создание таблицы.

Создаем новую таблицу с помощью оператора CREATE TABLE. Имя таблицы: price\_leck – таблица, хранящая в себе цену каждой платной книги в библиотеках. Состоит из следующих столбцов: id, id\_leckp, name\_leck, price.

Код:

```
create table price_leck
(
    id integer,
    id_leckp int,
    name_leck varchar(30),
    price integer
);
```

```
Query  Query History
1  create table price_leck
2  (
3      id integer,
4      id_lickp int,
5      name_leck varchar(30),
6      price integer
7  );
8

Data Output  Messages  Notifications
CREATE TABLE
Query returned successfully in 53 msec.
```

## Шаг 2. Ограничение CHECK.

Для того, чтобы добавить ограничение CHECK воспользовался оператором ALTER TABLE и синтаксисом написания ограничения CHECK.

Код:

```
ALTER TABLE price_leck
ADD CHECK (price > 100);
```

```
Query  Query History  Sc
1  ALTER TABLE price_leck
2  ADD CHECK (price > 100);
3

Data Output  Messages  Notifications
ALTER TABLE
Query returned successfully in 55 msec.
```

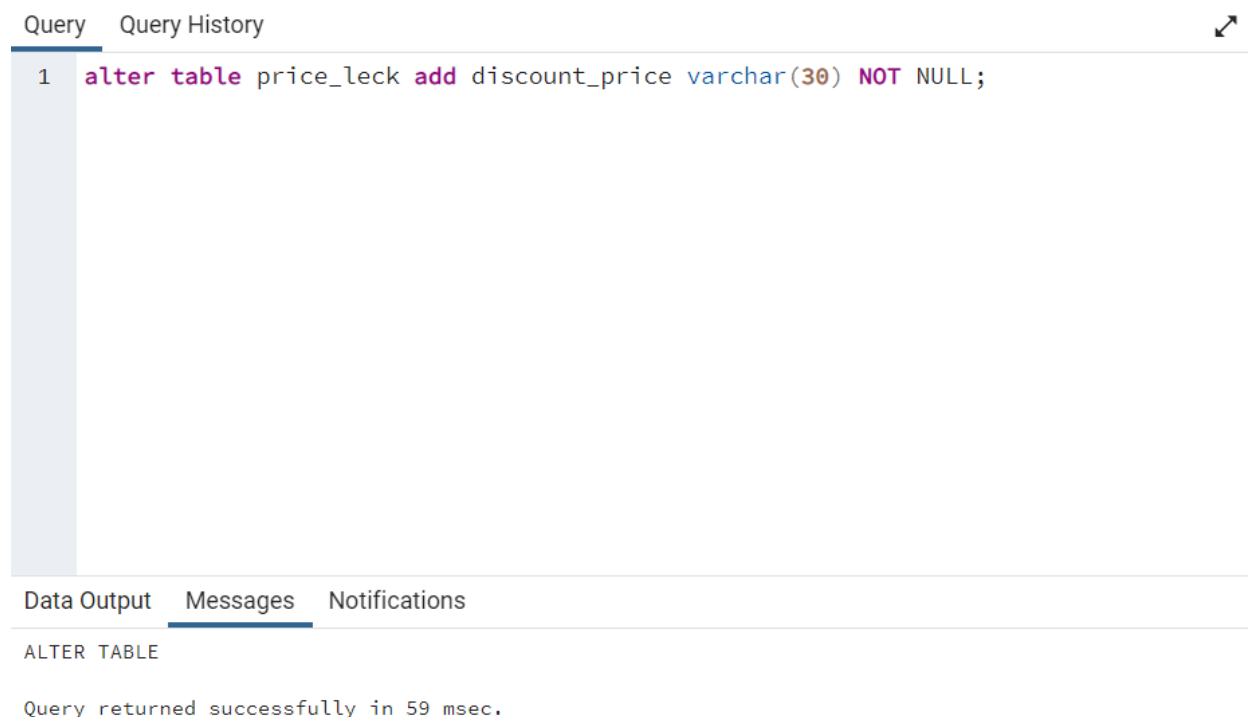
### Шаг 3. NOT NULL + DEFAULT.

Чтобы указать, может ли столбец принимать значение NULL, при определении столбца ему можно задать атрибут NULL или NOT NULL. Если этот атрибут не будет использован, то по умолчанию столбец будет допускать значение NULL. Исключением является, когда столбец выступает в роли первичного ключа - в этом случае по умолчанию столбец имеет значение NOT NULL.

Для добавления ограничения DEFAULT, так же воспользуемся оператором ALTER TABLE и синтаксисом написания ограничения DEFAULT. DEFAULT определяет значение по умолчанию для столбца. Если при добавлении данных для столбца не будет предусмотрено значение, то для него будет использоваться значение по умолчанию.

Код:

```
alter table price_leck add discount_price varchar(30) NOT NULL;
```



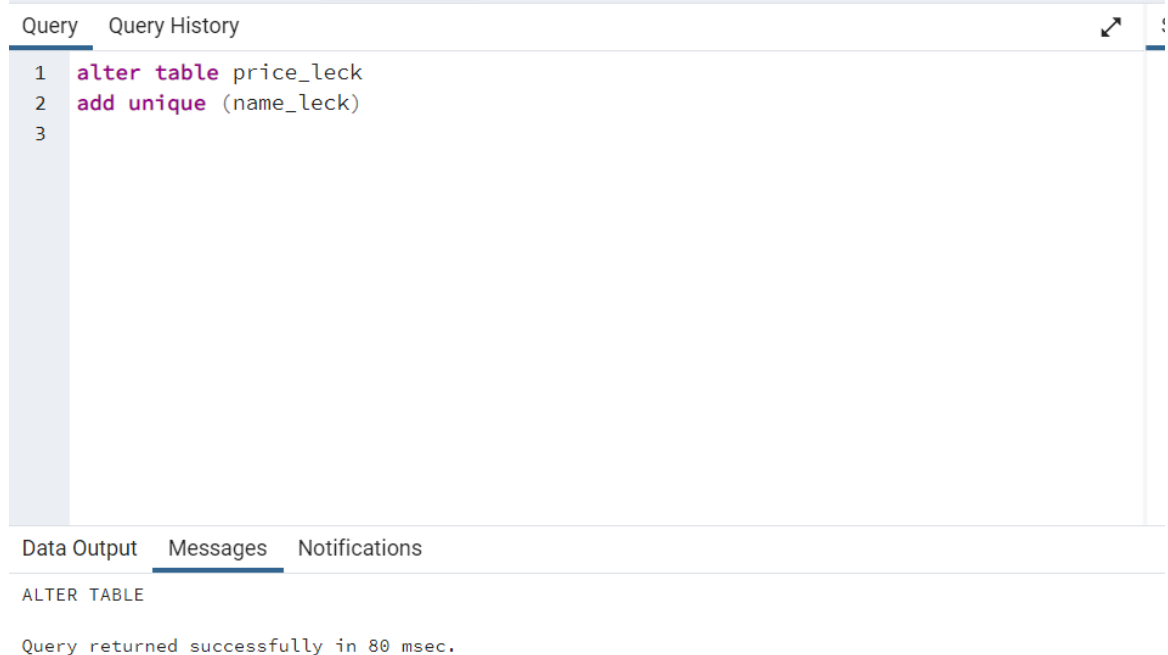
The screenshot shows a database query editor interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a single SQL command: `1 alter table price_leck add discount_price varchar(30) NOT NULL;`. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the output: `ALTER TABLE` and `Query returned successfully in 59 msec.`

### Шаг 4. UNIQUE.

UNIQUE – присвоение уникального значения к столбцу. Для добавления ограничения воспользуюсь оператором ALTER TABLE и синтаксисом написания ограничения UNIQUE.

Код:

```
alter table price_leck  
add unique (name_leck)
```



The screenshot shows a database query interface with a 'Query' tab selected. The query editor contains three lines of SQL code: '1 alter table price\_leck', '2 add unique (name\_leck)', and '3'. Below the editor, the 'Data Output' tab is selected, displaying the text 'ALTER TABLE'. At the bottom, a status message reads 'Query returned successfully in 80 msec.'.

```
Query Query History
```

```
1 alter table price_leck  
2 add unique (name_leck)  
3
```

```
Data Output Messages Notifications
```

```
ALTER TABLE
```

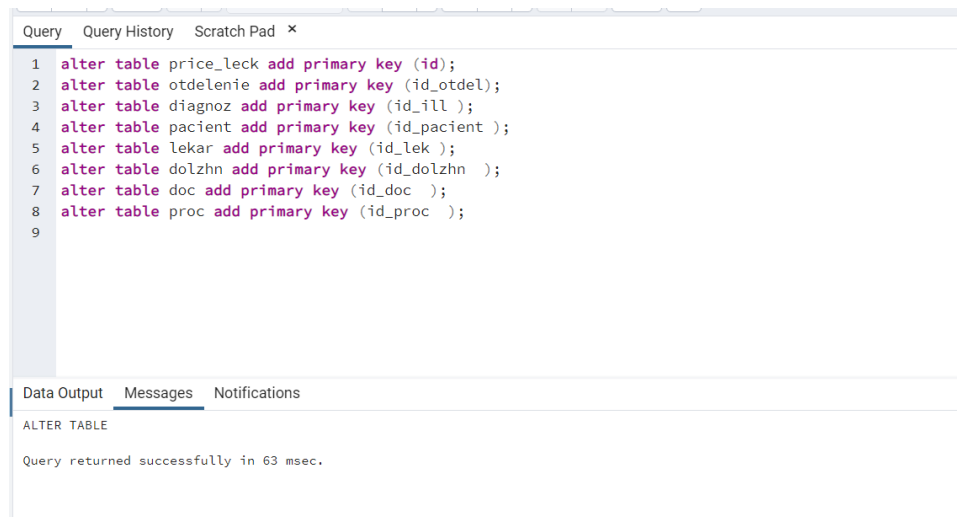
```
Query returned successfully in 80 msec.
```

### Шаг 5. PRIMARY KEY, FOREIGN KEY.

PRIMARY KEY - первичный ключ, ограничение, позволяющее однозначно идентифицировать каждую запись в таблице SQL. Для добавления ограничения воспользуюсь оператором ALTER TABLE и синтаксисом написания ограничения PRIMARY KEY.

Код:

```
alter table price_leck add primary key (id);  
  
alter table otделение add primary key (id_otdel);  
  
alter table diagnoz add primary key (id_ill );  
  
alter table pacient add primary key (id_pacient );  
  
alter table lekar add primary key (id_lek );  
  
alter table dolzhn add primary key (id_dolzhn );  
  
alter table doc add primary key (id_doc );  
  
alter table proc add primary key (id_proc );
```



```
Query    Query History    Scratch Pad x
1 alter table price_leck add primary key (id);
2 alter table otделение add primary key (id_otdel);
3 alter table diagnoz add primary key (id_ill );
4 alter table pacient add primary key (id_pacient );
5 alter table lekar add primary key (id_lek );
6 alter table dolzhn add primary key (id_dolzhn );
7 alter table doc add primary key (id_doc );
8 alter table proc add primary key (id_proc );
9

Data Output    Messages    Notifications
ALTER TABLE
Query returned successfully in 63 msec.
```

FOREIGN KEY используется для ограничения по ссылкам. Когда все значения в одном поле таблицы представлены в поле другой таблицы, говорится, что первое поле ссылается на второе. Это указывает на прямую связь между значениями двух полей. Для добавления ограничения воспользуюсь оператором ALTER TABLE и синтаксисом написания ограничения FOREIGN KEY.

Код:

```
alter table diagnoz
add foreign key (id_otdel ) references otделение(id_otdel);

alter table pacient
add foreign key (id_otdel) references otделение(id_otdel);

alter table pacient
add foreign key (id_ill) references diagnoz(id_ill);

alter table diagnoz_lekar
add foreign key (fk_id_ill ) references diagnoz(id_ill);

alter table diagnoz_lekar
add foreign key (fk_id_lek) references lekar(id_lek);

alter table doc
add foreign key (id_dolzhn) references dolzhn(id_dolzhn) ;

alter table doc
add foreign key (id_otdel) references otделение(id_otdel);

alter table diagnoz_proc
add foreign key (fk_id_proc) references proc(id_proc);

alter table diagnoz_proc add foreign key (fk_id_ill) references diagnoz(id_ill);
```

```
Query Query History Scratch Pad ✕
1 alter table diagnoz
2 add foreign key (id_otdel ) references otделение(id_otdel);
3
4 alter table pacient
5 add foreign key (id_otdel) references otделение(id_otdel);
6
7 alter table pacient
8 add foreign key (id_ill) references diagnoz(id_ill);
9
10 alter table diagnoz_lekar
11 add foreign key (fk_id_ill ) references diagnoz(id_ill);
12
13 alter table diagnoz_lekar
14 add foreign key (fk_id_lek) references lekar(id_lek);
15
Data Output Messages Notifications
ALTER TABLE
Query returned successfully in 48 msec.
```

## Шаг 6. Добавления строки в таблицу price\_book.

Для того чтобы добавить новый столбец воспользуюсь оператором ALTER TABLE и ADD

Код:

```
alter table price_leck
add data_of_creation varchar(30) null;
```

```
Query Query History Scratch Pad ✕
1 alter table price_leck
2 add data_of_creation varchar(30) null;
3
Data Output Messages Notifications
ALTER TABLE
Query returned successfully in 45 msec.
```

Шаг 7. Удаление строки + удаление всей таблицы price\_book.  
Вторым столбцом является id\_book.

Код:

```
alter table price_leck  
drop column id_lickp;
```

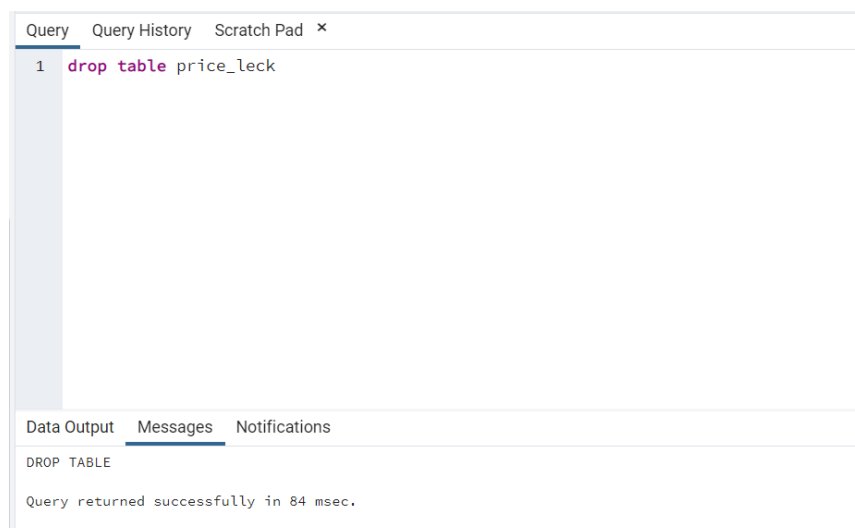


The screenshot shows a SQL query editor with a tab labeled 'Query'. The query text is: `1 alter table price_leck`  
`2 drop column id_lickp;`  
`3`. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the output: `ALTER TABLE`  
`Query returned successfully in 53 msec.`

Чтобы удалить всю таблицу, нужно воспользоваться оператором DROP TABLE.

Код:

```
drop table price_leck
```



The screenshot shows a SQL query editor with a tab labeled 'Query'. The query text is: `1 drop table price_leck`. Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the output: `DROP TABLE`  
`Query returned successfully in 84 msec.`