

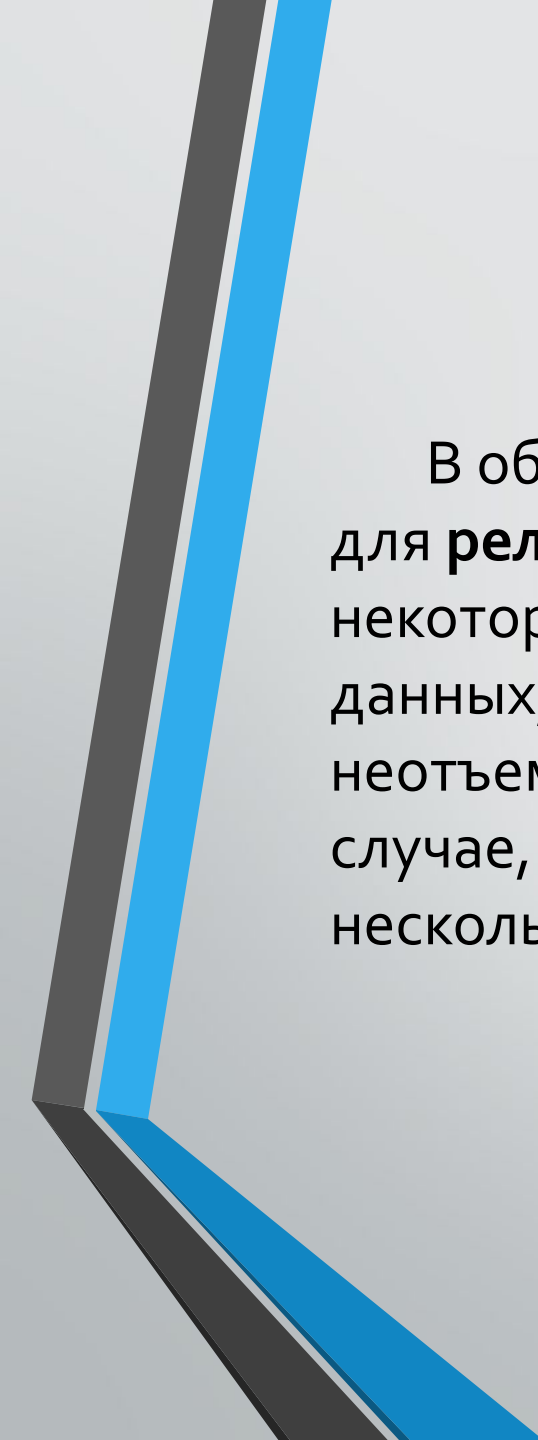


# ОСНОВЫ SQL

Запросы к базе данных

# Что такое база данных SQL?

SQL (Structured Query Language - «Структурированный язык запросов») - универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных



В общих терминах, «SQL база данных» является общим названием для **реляционной системы управления базами данных (РСУБД)**. Для некоторых систем, «база данных» также относится к группе таблиц, данных, конфигурационной информации, которые являются неотъемлемо отдельной частью от других, подобных конструкций. В этом случае, каждая инсталляция SQL базы данных может состоять из нескольких баз данных. В других системах, они упомянуты как таблицы.

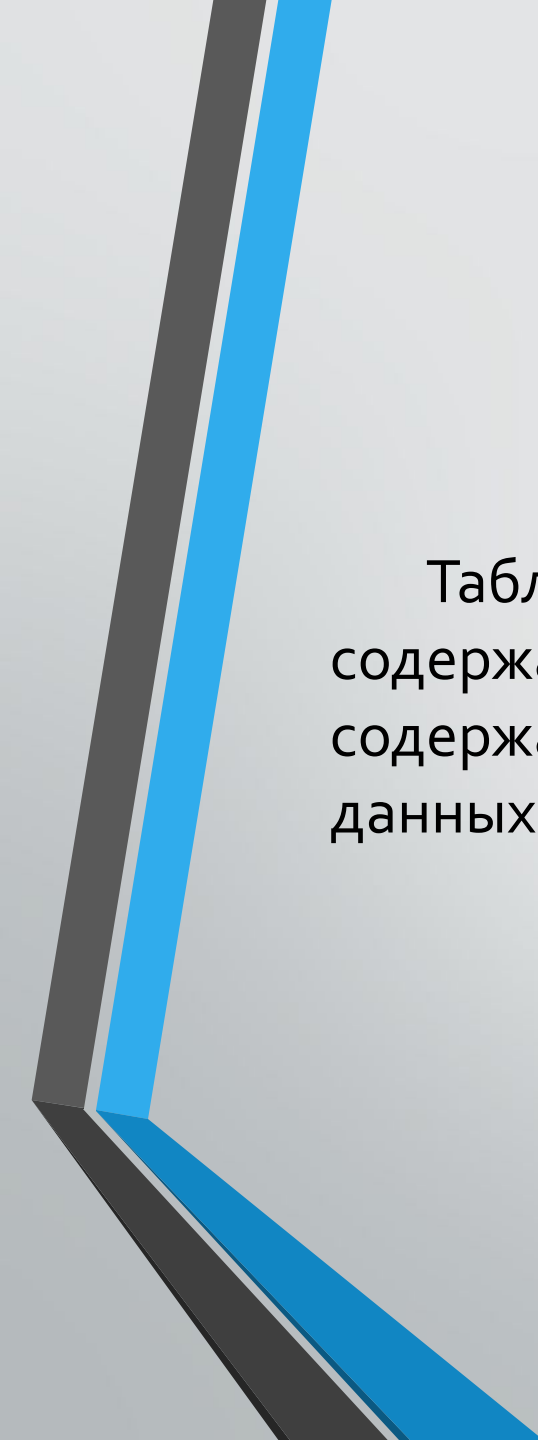
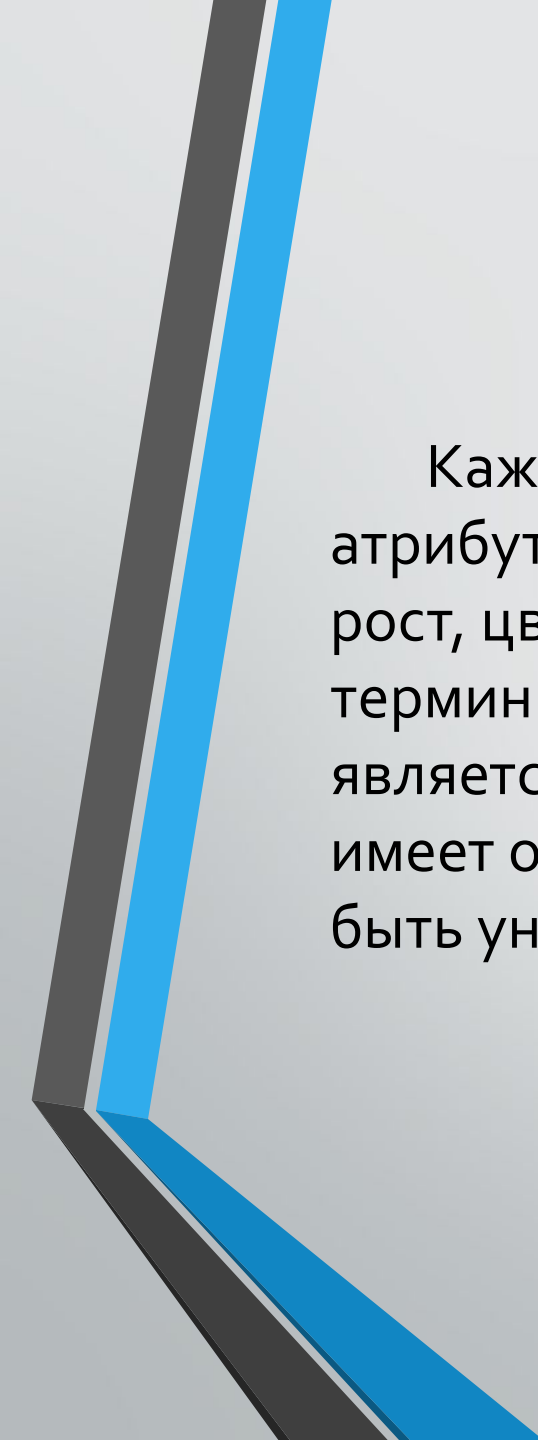
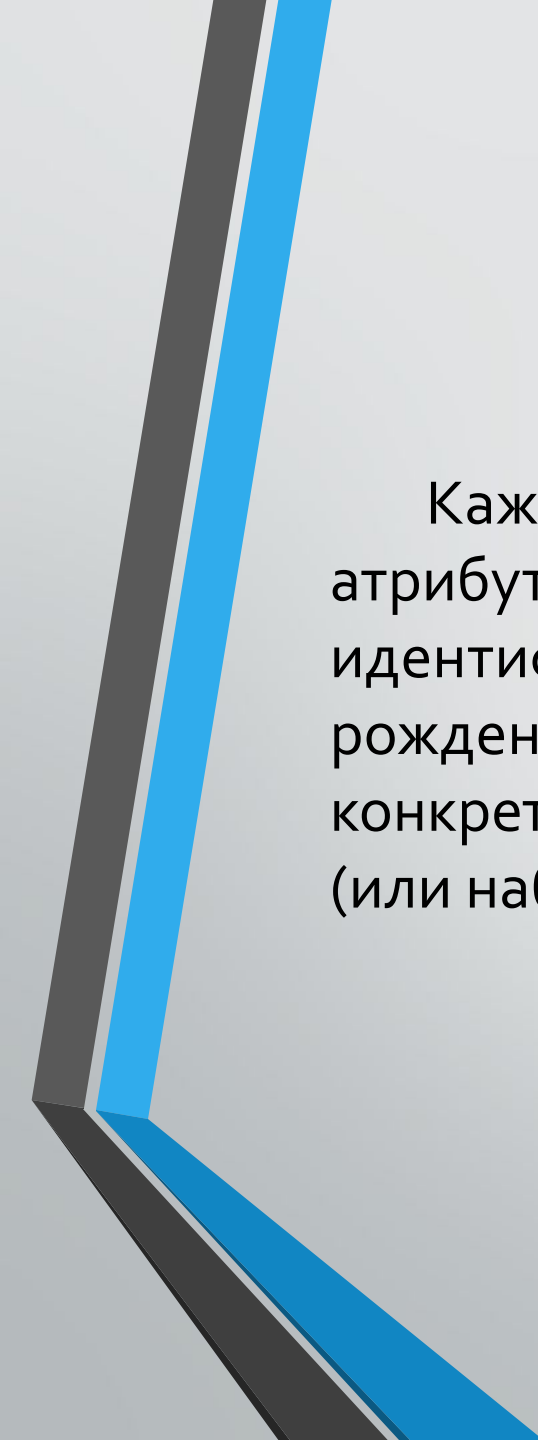


Таблица – конструкция базы данных, которая состоит из **столбцов**, содержащих **строки** данных. Обычно таблицы созданы для того, чтобы содержать связанную информацию. В пределах той же самой базы данных могут быть созданы несколько таблиц.



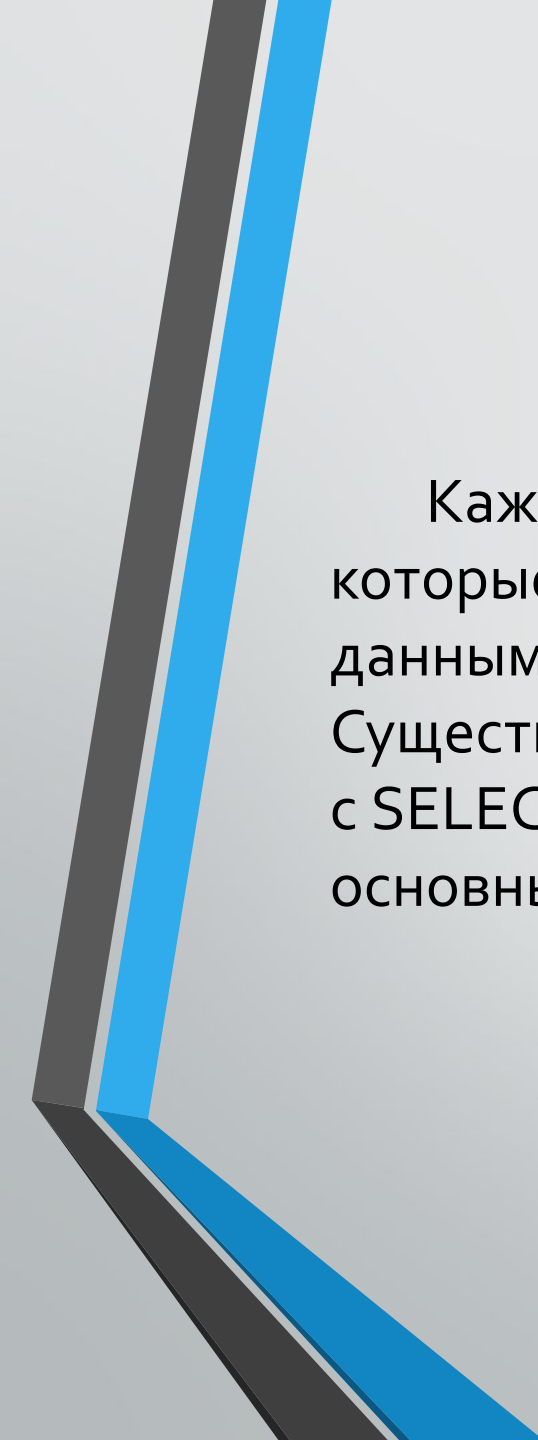
Каждый столбец представляет собой атрибут или совокупность атрибутов объектов, например идентификационные номера служащих, рост, цвет машин и т.п. Часто в отношении столбца используется термин *поле* с указанием имени, например «в поле Name». Поле строки является минимальным элементом таблицы. Каждый столбец в таблице имеет определенное имя, тип данных и размер. Имена столбцов должны быть уникальны в пределах таблицы.



Каждая строка (или запись) представляет собой совокупность атрибутов конкретного объекта, например, в строке может содержаться идентификационный номер служащего, размер его зарплаты, год его рождения и т.д. Строки таблиц не имеют названий. Чтобы обратиться к конкретной строке, пользователю необходимо указать какой-то атрибут (или набор атрибутов), уникально ее идентифицирующий.

# Типы запросов данных

- Есть четыре основных типа запросов данных в SQL, которые относятся к так называемому **языку манипулирования данными** (Data Manipulation Language или DML):
- **SELECT** – выбрать строки из таблиц;
- **INSERT** – добавить строки в таблицу;
- **UPDATE** – изменить строки в таблице;
- **DELETE** – удалить строки в таблице;



Каждый из этих запросов имеет различные операторы и функции, которые используются для того, чтобы произвести какие-то действия с данными. Запрос SELECT имеет самое большое количество опций. Существуют также дополнительные типы запросов, используемых вместе с SELECT, типа JOIN и UNION. Но пока, мы сосредоточимся только на основных запросах.



# Использование запроса SELECT для выборки нужных данных

Чтобы получить информацию, хранящуюся в базе данных используется запрос SELECT. Базовое действие этого запроса ограничено одной таблицей, хотя существуют конструкции, обеспечивающие выборку с нескольких таблиц одновременно. Для того, чтобы получить все строки данных для специфических столбцов, используется запрос такого вида:


```
SELECT column1, column2 FROM table_name;
```

Также, можно получить все столбцы из таблицы, используя подстановочный знак «\*»:

```
SELECT * FROM table_name;
```

Это может быть полезно в том случае, когда вы собираетесь выбрать данные с определенным условием WHERE. Следующий запрос возвратит все столбцы со всех строк, где «column1» содержит значение «3»:

```
SELECT * FROM table_name WHERE column1=3;
```



Кроме «=» (равно), существуют следующие условные операторы:

= Равно

<> Не равно

> Больше

< Меньше

>= Больше или равно

<= Меньше или равно

Дополнительно можно использовать условия BETWEEN и LIKE для сравнения с условием WHERE, а так же комбинации операторов AND и OR.

```
SELECT * FROM table_name WHERE ((Age >= 18) AND (LastName BETWEEN 'Иванов' AND 'Сидоров')) OR Company LIKE '%Motorola%';
```

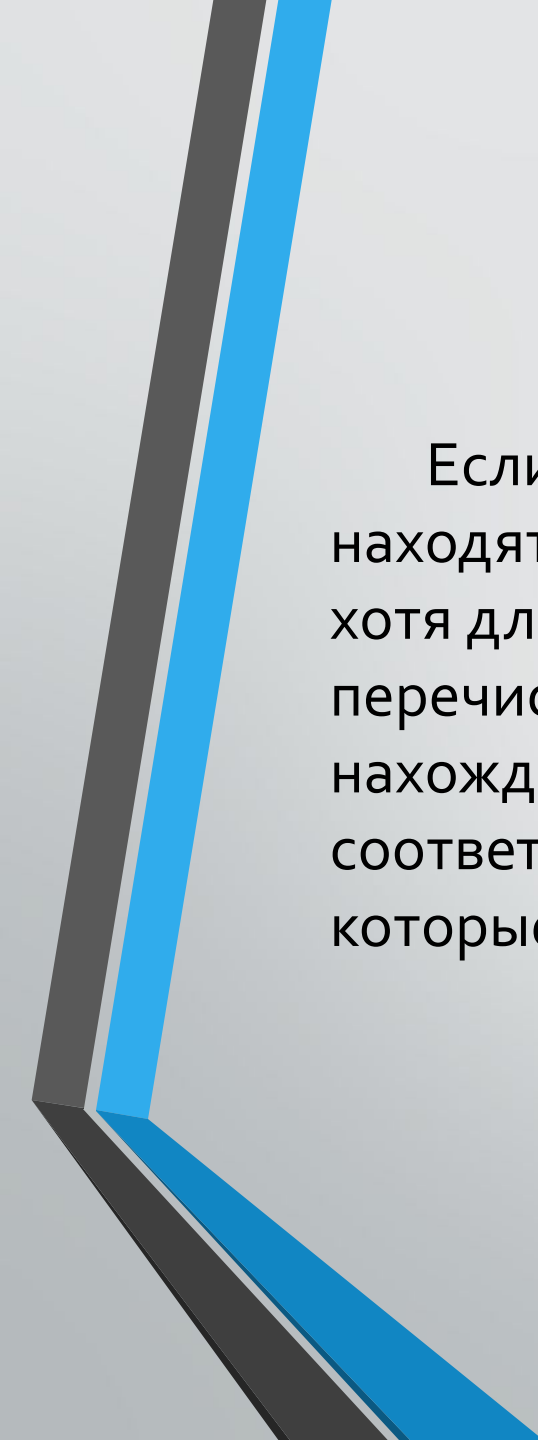
Что в переводе на русский язык означает: выбрать все столбцы из таблицы table\_name, где значение столбца age больше или равно 18, а также значение столбца LastName находится в алфавитном промежутке от Иванов до Сидоров включительно, или же значением столбца Company является Motorola

# Использование запроса INSERT для вставки НОВЫХ ДАННЫХ

Запрос INSERT используется для создания новой строки данных. Для обновления уже существующих данных или пустых полей строки нужно использовать запрос UPDATE.

Примерный синтаксис запроса INSERT:

```
INSERT INTO table_name (column1, column2, column3) VALUES  
(‘data1’, ‘data2’, ‘data3');
```

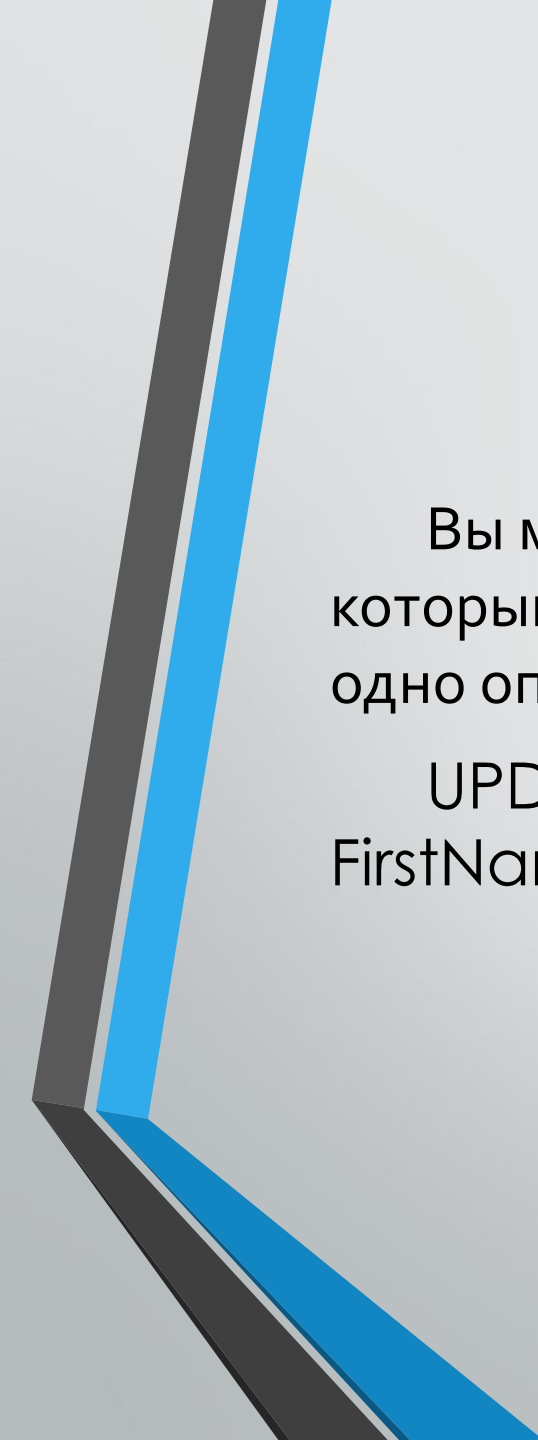


Если вы собираетесь вставлять все значения в порядке, в котором находятся столбцы таблицы, то можно и не указывать имена столбцов, хотя для удобочитаемости это предпочтительнее. Кроме того, если вы перечисляете столбцы, необязательно указывать их по порядку нахождения в базе данных, пока значения, которые вы вводите, соответствуют этому порядку. Вы не должны перечислять столбцы, в которые не вводится информация

# Запрос UPDATE и условие WHERE

UPDATE используется для того, чтобы изменить существующие значения или освободить поле в строке, поэтому новые значения должны соответствовать существующему типу данных и обеспечивать приемлемые значения. Если вы не хотите изменить значения во всех строках, то нужно использовать условие WHERE.

```
UPDATE table_name SET column1 = 'data1', column2 = 'data2'  
WHERE column3 = 'data3';
```



Вы можете использовать WHERE для любого столбца, включая тот, который хотите изменить. Это используется когда необходимо заменить одно определенное значение на другое.

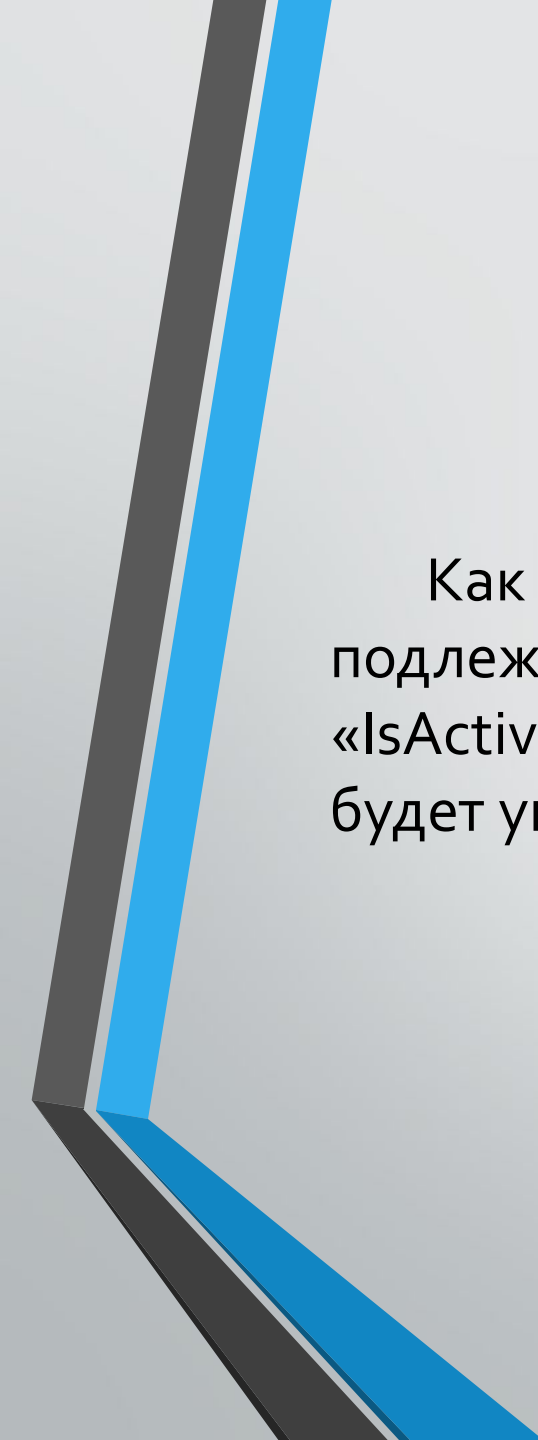
```
UPDATE table_name SET FirstName = 'Василий' WHERE  
FirstName = 'Василий' AND LastName = 'Пупкин';
```



# Будьте осторожны! Запрос DELETE удаляет целые строки

Запрос DELETE полностью удаляет строку из базы данных. Если вы хотите удалить одно единственное поле, то нужно использовать запрос UPDATE и установить для этого поля значение, которое будет являться аналогом NULL в вашей программе. Будьте внимательны, и ограничивайте ваш запрос DELETE условием WHERE, иначе вы можете потерять все содержимое таблицы.

```
DELETE FROM table_name WHERE column1 = 'data1';
```



Как только строка была удалена из вашей базы данных, она не подлежит восстановлению, поэтому желательно иметь столбец по имени «IsActive», или что-то типа того, который вы можете изменить на ноль, что будет указывать на блокировку представления данных из этой строки