

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