

60. Структуризированный язык запросов. Язык SQL. Построение запросов на создание и удаление таблиц. Построение запросов на извлечение данных. Использование критериев отбора. Вложенные запросы. Построение запросов на занесение, удаление и обновление данных.

SQL (structured query language — «**структурированный язык запросов**») — формальный не процедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД).

SQL является прежде всего информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных.

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций:

- создание в базе данных новой таблицы;
- добавление в таблицу новых записей;
- изменение записей;
- удаление записей;
- выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
- изменение структур таблиц.

Со временем SQL усложнился — обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) — и стал приобретать черты, свойственные языкам программирования.

При всех своих изменениях SQL остаётся единственным механизмом связи между прикладным программным обеспечением и базой данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов.

Каждое предложение SQL — это либо запрос данных из базы, либо обращение к базе данных, которое приводит к изменению данных в базе. В соответствии с тем, какие изменения происходят в базе данных, различают следующие типы запросов:

- запросы на создание или изменение в базе данных новых или существующих объектов (при этом в запросе описывается тип и структура создаваемого или изменяемого объекта);
- запросы на получение данных;
- запросы на добавление новых данных (записей);
- запросы на удаление данных;

Основным объектом хранения реляционной базы данных является таблица, поэтому все SQL-запросы — это операции над таблицами. В соответствии с этим, запросы делятся на:

- запросы, оперирующие самими таблицами (создание и изменение таблиц);
- запросы, оперирующие с отдельными записями (или строками таблиц) или наборами записей.

Каждая таблица описывается в виде перечисления своих полей (столбцов таблицы) с указанием

- типа хранимых в каждом поле значений;
- связей между таблицами (задание первичных и вторичных ключей);
- информации, необходимой для построения индексов.

Запросы первого типа в свою очередь делятся на запросы, предназначенные для создания в базе данных новых таблиц, и на запросы, предназначенные для изменения уже существующих таблиц. Запросы второго типа оперируют со строками, и их можно разделить на запросы следующего вида:

- вставка новой строки;
- изменение значений полей строки или набора строк;
- удаление строки или набора строк.

Самый главный вид запроса — это запрос, возвращающий (пользователю) некоторый набор строк, с которым можно осуществить одну из трёх операций:

- просмотреть полученный набор;
- изменить все записи набора;
- удалить все записи набора.

Таким образом использование SQL сводится, по сути, к формированию всевозможных выборок строк и совершению операций над всеми записями, входящими в набор.

Построение запросов на создание и удаление таблиц

Для создания объекта БД (таблица, представление, пользователь и др.) в SQL используется оператор CREATE. При создании таблицы необходимо описать ее структуру (поля и типы данных полей).

Синтаксис создания таблицы: CREATE TABLE <ИМЯ> (Структура таблицы)

Пример:

```
CREATE TABLE Student (  
    Code INTEGER NOT NULL,  
    Name CHAR (30) NOT NULL ,  
    Address CHAR (50),  
    Mark DECIMAL  
);
```

Для удаления объекта используется оператор DROP.

Синтаксис удаления таблицы DROP TABLE <ИМЯ>

Пример:

```
DROP TABLE STUDENT;
```

Построение запросов на извлечение данных

Для извлечения данных используется оператор SELECT. При формировании запроса SELECT пользователь описывает ожидаемый набор данных: его вид (набор столбцов) и его содержимое (критерий попадания записи в набор, группировка значений, порядок вывода записей и т. п.).

Синтаксис: SELECT список полей FROM список таблиц WHERE условия

В случае отсутствия условий WHERE можно опустить. Простейший пример, выборка всех полей из одной таблицы:

```
SELECT * FROM Student;
```

В случае, когда нужно получить не все записи, можно указать **критерии отбора** после ключевого слова WHERE. Все записи, для которых значение условия будет истинно, будут возвращены. SQL также поддерживает логические операции AND, OR, NOT, LIKE и др.

Пример:

```
SELECT * from Students WHERE mark > 4.5
```

Вложенные запросы

Вложенный запрос - это запрос, заключенный в круглые скобки и вложенный в предложение WITH, SELECT, FROM, WHERE или HAVING основного (внешнего) запроса SELECT или других запросов, использующих эти предложения. Вложенный запрос также называют подзапросом. Вложенный запрос в своих предложениях может содержать другой вложенный запрос и т. д.

Например, мы хотим получить имя пользователя, создавшего определенную тему на форуме. Это можно сделать, например, так

```
SELECT name FROM users WHERE user_id = (SELECT author_id FROM topics WHERE topic_id = 1)
```

Построение запросов на занесение, удаление и обновление данных.

Для **занесения** данных в таблицу используется оператор INSERT

Синтаксис INSERT INTO <название таблицы> ([<Имя столбца>, ...]) VALUES (<Значение>,...)

Имена столбцов можно опустить.

Пример: INSERT INTO PEOPLE(ID, NM, FM, OT) VALUES(1, 'John', 'Godwin', 'Petrovich')

Для **удаления** записей из таблицы используется оператор DELETE

DELETE FROM <Имя Таблицы> WHERE <Условие отбора записей>

Пример:

```
DELETE FROM PEOPLE WHERE name="BarackObama"
```

Для **обновления** данных используется оператор UPDATE

Синтаксис UPDATE <объект>

SET <[присваивание1 , присваивание2, ...]>

[WHERE <условие>];

Пример: UPDATE persons SET street = 'Nissestien 67', city = 'Sandnes' WHERE lastname = 'Tjessem' AND firstname = 'Jakob';

Доп. Ссылки

<http://ru.wikipedia.org/wiki/SQL> - тут все круто написано, в том числе преимущества и недостатки SQL кому надо

<http://www.pmbk.ru/lister/010/41/index.shtml> - подробнее про вложенные запросы