

УДК:004.4

ЯЗЫК ЗАПРОСОВ SQL: НАЗНАЧЕНИЕ, ОСНОВНЫЕ КОМАНДЫ, ФОРМИРОВАНИЕ ПРОСТЫХ И СЛОЖНЫХ ЗАПРОСОВ

Евсюков Д.С., Негребецкая В.И.

ФГБОУ ВО «Курский государственный университет», колледж коммерции, технологий и сервиса, Россия, Курск, e-mail: dimon2002wasa@yandex.ru, violetta-negrebel@mail.ru.

В статье описывается актуальность использования языка запросов SQL, дается характеристика СУБД (PostgreSQL, MySQL, SQLite). Кроме того рассматриваются основные команды языка SQL, а также формирование простых и сложных запросов.

Ключевые слова: база данных, СУБД, PostgreSQL, MySQL, SQLite, язык SQL, запрос, выборка, группировка, объединение.

SQL QUERY LANGUAGE: PURPOSE, BASIC COMMANDS, FORMATION OF SIMPLE AND COMPLEX QUERIES

Evsyukov D.S, Negrebetskaya V. I.

Kursk State University, College of Commerce, Technology and Service, Kursk, Russia, e-mail: dimon2002wasa@yandex.ru , violetta-negrebel@mail.ru .

The article describes the relevance of using the SQL query language, gives the characteristics of the DBMS (PostgreSQL, MySQL, SQLite). In addition, the main commands of the SQL language are considered, as well as the formation of simple and complex queries.

Keywords: database, DBMS, PostgreSQL, MySQL, SQLite, SQL language, query, selection, grouping, union.

Без компьютера не обходится жизнь любого человека, будь это специалист или обычный пользователь. В условиях рыночной экономики, перехода на цифровую экономику, предприятия автоматизируют свою деятельность, внедряя автоматизированные информационные системы, в которых основой являются базы данных.

Для того чтобы создать базу данных необходимо можно использовать Microsoft Access, но чтобы сделать, по настоящему хорошую базу данных потребуется язык SQL.

SQL — это структурированный язык запросов, созданный для того, чтобы получать из базы данных необходимую информацию. Если описать схему работы SQL

простыми словами, то специалист формирует запрос и направляет его в базу. Та в свою очередь обрабатывает эту информацию, «понимает», что именно нужно специалисту, и отправляет ответ [1]. Данные хранятся в виде таблиц, они структурированы и разложены по строкам и столбцам, чтобы ими легче было оперировать. Такой способ хранения информации называют реляционными базами данных (от англ. relation — «отношения»). Название указывает на то, что объекты в такой базе связаны определенными отношениями.

В настоящее время есть множество СУБД, которые используют различные организации [2]:

1. PostgreSQL - объектно-ориентированная система, то есть она обрабатывает данные как абстрактные объекты. Каждый объект, в отличие от простых табличных значений, может иметь собственные характеристики и уникальные методы взаимодействия с другими объектами. Это позволяет PostgreSQL обрабатывать более сложные структуры данных и выполнять более сложные процедуры. Например, Яндекс. Почта в свое время перешла на эту систему, чтобы поддерживать стабильное соединение десятков тысяч пользователей к одной базе.

2. MySQL. Простая в изучении и функциональная система, которая работает с сайтами и веб-приложениями. Чаще всего используется в системах управления контентом сайтов (CMS), на сайтах с возможностью регистрации пользователей, в корпоративных системах CRM, в планировщиках, чатах и форумах. MySQL считается одним из самых безопасных и высокоскоростных решений, которое существует на рынке.

3. SQLite - облегченная встраиваемая версия СУБД. В ней нет возможности поделиться правами доступа, как во многих других системах, но благодаря своему устройству эта система быстрая и мощная. SQLite подходит для обработки запросов на сайтах с низким и средним трафиком, а также в однопользовательских мобильных приложениях и играх. Преимущество такой системы — файловая структура, то есть база в SQLite состоит из одного файла, поэтому ее очень легко переносить. [5]

Основные команды, с помощью которых выстраивается SQL-запрос следующие:

Show database-команда отвечает за просмотр доступных баз данных.

Create database- команда для создания баз данных.

Use- эта команда отвечает за выбор той или иной базы данных, с которой пользователь будет работать.

Source- позволит выполнить сразу несколько команд которые присутствуют в базе данных.

Drop database- позволит удалить базу данных [3].

Еще в базе данных обязательно присутствуют запросы. Среди них есть как и простые, так и сложные запросы:

Приведем простые примеры использования оператора SELECT.

Синтаксис:

```
> SELECT <fields1> FROM <table> [JOIN <table2>] [ WHERE <conditions> ORDER BY <fields2> LIMIT <count> ]
```

где fields1 — поля для выборки через запятую, также можно указать все поля знаком;

table — имя таблицы, из которой вытаскиваем данные;

conditions — условия выборки;

fields2 — поле или поля через запятую, по которым выполнить сортировку;

count — количество строк для выгрузки.

Запрос в квадратных скобках не является обязательным для выборки данных.

1. Обычная выборка данных

```
> SELECT * FROM users
```

В данном примере можно получить список всех записей из таблицы users.

2. Выборка данных с объединением двух таблиц (JOIN)

```
SELECT u.name, r.* FROM users u JOIN users_rights r ON r.user_id=u.id
```

В данном примере идет выборка данных с объединением таблиц users и users_rights. Объединяются они по полям user_id (в таблице users_rights) и id (users). Извлекается поле name из первой таблицы и все поля из второй.

Рассмотрим примеры более сложных запросов или используемых редко.

1. Объединение с группировкой выбранных данных в одну строку (GROUP_CONCAT)

```
> SELECT GROUP_CONCAT(DISTINCT CONVERT(id USING 'utf8') SEPARATOR ', ') as ids FROM users
```

Из таблицы users извлекаются данные по полю id, все они помещаются в одну строку, значения разделяются запятыми.

2. Группировка данных по двум и более полям

```
> SELECT * FROM users GROUP BY CONCAT(title, '::', birth)
```

В данном примере делается выгрузка данных из таблицы users и они группируются по полям title и birth. Перед группировкой выполняется объединение полей в одну строку с разделителем ::.

3. Объединение результатов из двух и более таблиц (UNION)

а) простой вариант использования UNION:

```
> (SELECT id, fio, address, 'Пользователи' as type FROM users)
UNION (SELECT id, fio, address, 'Покупатели' as type FROM customers). [4]
```

Таким образом, можно сделать вывод, что язык запросов SQL представляет собой язык программирования, предназначенный для работы с наборами фактов и отношениями между ними. В программах управления реляционными базами данных, таких как Microsoft Office Access, язык SQL используется для работы с данными. Он помогает создать отсортированную, обработанную базу данных, но в свою очередь язык не простой и не каждому пользователю он подойдет.

Список литературы:

1. SQL основное понятие и принцип [Электронный ресурс] / URL: <https://blog.skillfactory.ru/glossary/sql/> (дата обращения: 20.03.2022).
2. Актуальные СУБД которые используются на предприятии [Электронный ресурс] / URL: <https://itvdn.com/ru/blog/article/m-sql> (дата обращения: 20.03.2022).
3. Вывод про язык SQL [Электронный ресурс] / URL: <https://support.microsoft.com/ru-ru/office/access-sql-основные-понятия-лексика-и-синтаксис-444d0303-cde1-424e-9a74-e8dc3e460671#:~:text=SQL%20—%20это%20язык%20программирования%2C,используется%20для%20работы%20с%20дан%20ными> (дата обращения: 20.03.2022).
4. Основные понятия, лексика и синтаксис [Электронный ресурс] / URL: <https://support.microsoft.com/ru-ru/office/access-sql-основные-понятия-лексика-и-синтаксис-444d0303-cde1-424e-9a74-e8dc3e460671> (дата обращения: 20.03.2022).
5. Простые и более сложные запросы в СУБД [Электронный ресурс] / URL: <https://www.dmosk.ru/miniiinstruktions.php?mini=sql-mysql#complex> (дата обращения: 20.03.2022).