

**CREATING SIMPLE QUERIES FOR DATA MANIPULATION.**

Bekmurodov Lazizbek Hamdam o'g'li,

Qarshi State Technical University,

Computer engineering student

Annotation. *This article discusses the creation of simple queries for data manipulation, focusing on the essential techniques for interacting with databases. Data manipulation involves tasks like retrieving, updating, and deleting data, all of which are achieved through SQL (Structured Query Language) queries. The article introduces key SQL operations, such as the SELECT, INSERT, UPDATE, and DELETE statements, and explains their usage with practical examples. It also covers the application of aggregation functions like COUNT, SUM, AVG, and how to use JOINS for combining data from multiple tables. This foundational knowledge of SQL queries is crucial for those working with databases to analyze and transform raw data into actionable insights.*

Keywords. *Data manipulation, SQL queries, SELECT statement, INSERT statement, UPDATE statement, DELETE statement, Aggregation functions, COUNT, SUM, AVG, JOIN operations, Database management.*

Аннотация. *В этой статье обсуждается создание простых запросов для манипулирования данными, с упором на основные методы взаимодействия с базами данных. Манипулирование данными включает в себя такие задачи, как извлечение, обновление и удаление данных, все из которых достигаются с помощью запросов SQL (язык структурированных запросов). В статье представлены ключевые операции SQL, такие как операторы SELECT, INSERT, UPDATE и DELETE, и объясняется их использование на практических примерах. В ней также рассматривается применение функций агрегирования, таких как COUNT, SUM, AVG, и как использовать JOIN для объединения данных из нескольких таблиц. Эти фундаментальные знания SQL-запросов*



имеют решающее значение для тех, кто работает с базами данных для анализа и преобразования необработанных данных в действенные идеи.

Ключевые слова. Манипулирование данными, SQL-запросы, оператор *SELECT*, оператор *INSERT*, оператор *UPDATE*, оператор *DELETE*, функции агрегирования, операции *COUNT*, *SUM*, *AVG*, *JOIN*, управление базами данных.

Data manipulation is a crucial aspect of data analysis, where raw data is transformed into meaningful insights through various processes. One of the first steps in data manipulation is creating simple queries that allow users to interact with databases and retrieve, update, or delete information. In this article, we will explore how to create basic queries for data manipulation, focusing on the fundamental components of SQL (Structured Query Language) and how they are applied to real-world data.

Data manipulation refers to the process of adjusting, editing, or transforming data into a desired format for analysis. It involves tasks such as cleaning, transforming, sorting, filtering, and summarizing data. For this purpose, queries are used to interact with databases, extracting relevant data for further processing or reporting.

In databases, SQL is the most common language used for data manipulation. With SQL, users can create queries to perform tasks like retrieving data (using *SELECT*), updating data (using *UPDATE*), and deleting data (using *DELETE*). These operations are essential for maintaining and analyzing large datasets.

Understanding Basic SQL Queries

a) *SELECT* Query

The *SELECT* statement is the most fundamental SQL query used to retrieve data from a database. It allows users to select specific columns or all columns from a table.

Syntax:

SELECT column1, column2, ...

FROM table_name;

If you want to select all columns, you can use the * symbol.

**Example:**

```
SELECT first_name, last_name, age  
FROM employees;
```

This query retrieves the first name, last name, and age of all employees from the "employees" table.

b) WHERE Clause

The WHERE clause is used to filter records based on specific conditions. It helps in retrieving only the rows that meet the specified criteria.

Syntax:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Example:

```
SELECT first_name, last_name  
FROM employees  
WHERE age > 30;
```

This query retrieves the names of employees who are older than 30 years.

c) INSERT Query

The INSERT statement is used to add new records to a table. You can insert data into specific columns or all columns.

Syntax:

```
INSERT INTO table_name (column1, column2, ...)  
VALUES (value1, value2, ...);
```

Example:

```
INSERT INTO employees (first_name, last_name, age)  
VALUES ('John', 'Doe', 28);
```

This query adds a new employee, John Doe, aged 28, into the employees table.

d) UPDATE Query

The UPDATE statement is used to modify existing records in a table. It requires a WHERE clause to specify which rows need to be updated.

**Syntax:**

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Example:

```
UPDATE employees  
SET age = 29  
WHERE first_name = 'John' AND last_name = 'Doe';
```

This query updates the age of John Doe to 29 in the employees table.

e) DELETE Query

The DELETE statement is used to remove records from a table. It is crucial to include a WHERE clause to specify which rows should be deleted, otherwise, all rows will be removed.

Syntax:

```
DELETE FROM table_name  
WHERE condition;
```

Example:

```
DELETE FROM employees  
WHERE age < 30;
```

This query deletes all employees who are younger than 30 years old.

Using Aggregation Functions in Queries

SQL also allows the use of aggregation functions to perform calculations on multiple rows. Common aggregation functions include COUNT(), SUM(), AVG(), MIN(), and MAX().

a) COUNT()

The COUNT() function returns the number of rows that match a specified condition.

Example:

```
SELECT COUNT(*)  
FROM employees
```



WHERE age > 30;

This query counts how many employees are older than 30 years.

b) SUM()

The SUM() function calculates the total of a numeric column.

Example:

```
SELECT SUM(salary)
```

```
FROM employees;
```

This query calculates the total salary of all employees in the database.

c) AVG()

The AVG() function calculates the average value of a numeric column.

Example:

```
SELECT AVG(age)
```

```
FROM employees;
```

This query calculates the average age of all employees.

d) GROUP BY Clause

The GROUP BY clause groups rows that have the same values into summary rows, often used with aggregate functions.

Example:

```
SELECT department, AVG(salary)
```

```
FROM employees
```

```
GROUP BY department;
```

This query calculates the average salary for each department.

Combining Multiple Queries with Joins

In many cases, data is spread across multiple tables, and you may need to retrieve information from more than one table. This is where JOINS come in.

a) INNER JOIN

An INNER JOIN returns records that have matching values in both tables.

Example:

```
SELECT          employees.first_name,          employees.last_name,  
departments.department_name
```



```
FROM employees  
INNER JOIN departments ON employees.department_id =  
departments.department_id;
```

This query retrieves the first and last names of employees along with their department names.

b) LEFT JOIN

A LEFT JOIN returns all records from the left table, along with matching records from the right table. If no match is found, NULL values are returned for columns of the right table.

Example:

```
SELECT employees.first_name, employees.last_name,  
departments.department_name  
FROM employees  
LEFT JOIN departments ON employees.department_id =  
departments.department_id;
```

This query retrieves all employees, even if they are not assigned to a department.

Creating simple queries for data manipulation is essential for working with databases. SQL offers a variety of powerful commands that allow users to retrieve, modify, and delete data with ease. Understanding the basics of SQL, such as the use of SELECT, INSERT, UPDATE, DELETE, and aggregate functions, is critical for anyone working in data analysis, database management, or software development.

By mastering these fundamental queries, users can effectively manipulate and analyze data to gain valuable insights and support decision-making processes. As you progress, you can explore more advanced techniques like subqueries, indexing, and optimization to further enhance your data manipulation skills.

REFERENCES:

1. Zarif o'g'li K. F. CREATING A TEST FOR SCHOOL EDUCATIONAL PROCESSES IN THE ISPRING SUITE PROGRAM //BOSHLANG 'ICH SINFLARDA O 'ZLASHTIRMOVCHILIKNI. – C. 84.



2. O'G'Li K. F. Z. CREATING A TEST FOR SCHOOL EDUCATIONAL PROCESSES IN THE ISPRING SUITE PROGRAM //Yosh mutaxassislari. – 2023. – T. 1. – №. 8. – С. 84-87.
3. Kaynarov F. Z. THEORETICAL FOUNDATIONS FOR THE CREATION OF ELECTRONIC TEXTBOOKS FOR DISTANCE EDUCATION //Экономика и социум. – 2024. – №. 2-2 (117). – С. 169-175.
4. Kaynarov F. APPLICATION OF MODERN INFORMATION TECHNOLOGIES IN MEDICINE //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.
5. Кайнаров Ф. З. ИННОВАЦИОННЫЕ МЕТОДЫ ПРЕПОДАВАНИЯ ПРИКЛАДНОЙ МАТЕМАТИКИ //Экономика и социум. – 2023. – №. 1-2 (104). – С. 619-622.
6. Daminova B. ACTIVATION OF COGNITIVE ACTIVITY AMONG STUDENTS IN TEACHING COMPUTER SCIENCE //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2023. – T. 2. – №. 1. – С. 68-71.
7. Esanovna D. B. Modern Teaching Aids and Technical Equipment in Modern Educational Institutions //International Journal of Innovative Analyses and Emerging Technology. – T. 2. – №. 6.
8. Рахимов Н., Эсановна Б., Примкулов О. Ахборот тизимларида мантикий хулосалаш самарадорлигини ошириш ёндашуви //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023
9. Даминова Б. Э. СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И ТЕНДЕНЦИИ ЕГО ИЗМЕНЕНИЯ ПОД ВЛИЯНИЕМ НОВЫХ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ УСЛОВИЙ //Yosh mutaxassislari. – 2023. – T. 1. – №. 8. – С. 72-77.
10. Кувандиков Ж., Даминова Б., Хафизадинов У. АВТОМАТЛАШТИРИЛГАН ЭЛЕКТРОН ТАЪЛИМ ТИЗИМИНИ ЛОЙИХАЛАШДА ЎҚУВ ЖАРАЁНИНИ МОДЕЛЛАШТИРИШ //International



Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.

11. Даминова Б. Э. Сравнительный анализ состояния организации многоуровневых образовательных процессов //Экономика и социум. – 2023. – №. 1-2 (104). – С. 611-614.

12. Daminova B. Algorithm of education quality assessment system in secondary special education institution (on the example of guzor industrial technical college) //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.

13. Daminova B. FORMATION OF THE MANAGEMENT STRUCTURE OF EDUCATIONAL PROCESSES IN THE HIGHER EDUCATION SYSTEM //Science and innovation. – 2023. – Т. 2. – №. А6. – С. 317-325.

14. Даминова Б. Э., Якубов М. С. Развития познавательной и творческой активности слушателей //Международная конференция" Актуальные проблемы развития инфокоммуникаций и информационного общества. – 2012. – С. 26-27.06.

15. Якубов М., Даминова Б., Юсупова С. Формирование и повышение качества образования с помощью образовательных информационных технологий //International Scientific and Practical Conference on Algorithms and Current Problems of Programming.-2023.

16. Даминова Б. Э. и др. ОБРАБОТКА ВИДЕОМАТЕРИАЛОВ ПРИ РАЗРАБОТКЕ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ //Экономика и социум. – 2024. – №. 2-2. – С. 117.