

## TIBBIYOT VA SOG'LIQNI SAQLASH TIZIMLARIDA SQL SO'ROVLAR TILI STRUKTURASINI QO'LLANILISHI

**Mamatova Zilolaxon Xabibulloxonovna**

*Farg'ona davlat universiteti dotsenti,  
pedagogika fanlari bo'yicha falsafa doktori (PhD)*

**Tojimamatov Isroil Nurmamatovich**

*Farg'ona davlat universiteti amaliy  
matematika va informatika kafedrası katta o'qituvchisi*

[israiltojimatov@gmail.com](mailto:israiltojimatov@gmail.com)

**Abdumannopova Dildora Akmaljonovna**

*Farg'ona davlat universiteti talabasi*

[iakmaljon214@gmail.com](mailto:iakmaljon214@gmail.com)

**Anottatsiya:** SQL So'rovlar Tili Strukturasi maqolasi SQL (Structured Query Language) tili yordamida ma'lumotlar bazasini boshqarish va undagi ma'lumotlarga so'rovlar yuborishning asosiy usullarini ko'rib chiqadi. SQL tili ma'lumotlar bazasidan ma'lumot olish, yangilash, qo'shish va o'chirish kabi amallarni bajarishda qo'llaniladi. Maqolada SQL so'rovlarining turli turlari, ularning sintaksisi, asosiy operatorlari va jadvallarni birlashtirish uchun ishlatiladigan JOIN operatorlari haqida batafsil tushuncha beriladi. Shuningdek, SQL so'rovlarining optimallashtirilishi va samarali ishlashi uchun tavsiyalar keltiriladi. Ushbu maqola ma'lumotlar bazasini boshqarish bilan shug'ullanuvchilar va SQL tilini o'rganishga qiziqqan foydalanuvchilar uchun mo'ljallangan.

**Kalit So'zlar:** SQL (Structured Query Language), SQL (Structured Query Language), So'rovlar tili, Ma'lumotlar bazasi, DDL (Data Definition Language), TCL (Transaction Control Language), SELECT so'rovi, INSERT so'rovi, UPDATE so'rovi, DELETE so'rovi, Join so'rovi, Inner Join, Left Join, Right Join, SQL sintaksisi,

Ma'lumotlar bazasi optimallashtirish, Index yaratish, SQL so'rovlarini samarali bajarish

**Annotation:** The Structure of SQL Query Language article discusses the main methods of managing databases and querying data within them using SQL (Structured Query Language). SQL is used to perform operations like retrieving, updating, inserting, and deleting data in a database. The article provides a detailed understanding of various types of SQL queries, their syntax, key operators, and the JOIN operators used to combine tables. It also includes recommendations for optimizing SQL queries and ensuring efficient database operation. This article is intended for database administrators and individuals interested in learning SQL.

**Key words:** SQL (Structured Query Language), Query language, Database, DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), TCL (Transaction Control Language), Select query, Insert Query, Update Query, Delete Query, Join operators, Inner Join, Left Join, Right Join, SQL Syntax, Database Optimization, Index Creation, Efficient SQL Query Execution.

**Аннотация:** Структура языка запросов SQL в данной статье рассматриваются основные методы управления базами данных и выполнения запросов к данным с использованием SQL (Structured Query Language). SQL используется для выполнения операций, таких как извлечение, обновление, добавление и удаление данных в базе данных. В статье подробно рассматриваются различные типы SQL-запросов, их синтаксис, ключевые операторы и операторы JOIN, используемые для объединения таблиц. Также приведены рекомендации по оптимизации SQL-запросов и обеспечению эффективной работы базы данных. Эта статья предназначена для администраторов баз данных и всех, кто интересуется изучением SQL.

**Ключевые слова:** SQL (Structured Query Language), Язык запросов, База данных, DDL (Data Definition Language), DML (Data Manipulation Language),

DCL (Data Control Language), TCL (Transaction Control Language), SELECT-запрос, INSERT-запрос, UPDATE-запрос, DELETE-запрос, Операторы JOIN, INNER JOIN, LEFT JOIN, Синтаксис SQL, RIGHT JOIN, Оптимизация базы данных, Создание индекса, Эффективное выполнение SQL-запросов.

Kirish. SQL So‘rovlar Tili Strukturasi (SQL Query Language Structure) tibbiyot va sog‘liqni saqlash tizimlarida keng qo‘llaniladi. Bu sohada SQL, asosan, bemorlar ma‘lumotlari, tibbiy arxiv, laboratoriya natijalari, dori-darmonlar, davolash jarayonlari va boshqa tibbiy ko‘rsatkichlarni boshqarish va tahlil qilish uchun ishlatiladi. SQL tibbiyot tizimlarida ma‘lumotlar bazasini samarali boshqarish va ma‘lumotlarni tez va aniq olishda muhim rol o‘ynaydi.

Quyida SQL so‘rovlar tili strukturasi tibbiyot va sog‘liqni saqlash tizimlarida qo‘llanilishi haqida batafsil ma‘lumot berishimiz mumkin:

### **1. Bemorlarga doir ma‘lumotlarni boshqarish**

SQL yordamida tibbiyot tizimlarida bemorlar haqidagi ma‘lumotlarni boshqarish mumkin. Bemorga oid ma‘lumotlar (ism, familiya, tug‘ilgan sana, jins, manzil, aloqa raqami, va boshqalar) alohida jadvallarda saqlanadi.

Bemorlar ro‘yxati: Bemorga oid ma‘lumotlar SELECT so‘rovi yordamida olinadi.

Misol:

Sql:

```
SELECT Name, Surname, DateOfBirth, Gender, ContactNumber  
FROM Patients  
WHERE PatientID = 12345;
```

Bemorni ro‘yxatga olish: Yangi bemorlarni tizimga qo‘shish uchun INSERT so‘rovi ishlatiladi.

Misol:

Sql:

```
INSERT INTO Patients (Name, Surname, DateOfBirth, Gender, ContactNumber)  
VALUES (“John”, “Doe”, “1980-05-15”, “Male”, “91-456-78-90”);
```

Bemorni yangilash: Agar bemorning manzili yoki aloqa ma'lumotlari o'zgarsa, UPDATE so'rovi ishlatiladi.

Misol:

Sql:

```
UPDATE Patients
```

```
SET ContactNumber = "98-654-32-10"
```

```
WHERE PatientID = 12345;
```

Bemorni tizimdan o'chirish: Agar bemor tizimdan chiqarilishi kerak bo'lsa, DELETE so'rovi ishlatiladi.

Misol:

Sql:

```
DELETE FROM Patients
```

```
WHERE PatientID = 12345;
```

## 2. Tibbiy arxivni boshqarish

Tibbiy arxivda bemorlarning kasalliklar, davolashlar, simptomlar va muolajalar bilan bog'liq ma'lumotlar saqlanadi. SQL yordamida bu ma'lumotlar tezda tahlil qilinadi va bemorning tarixini kuzatib borish mumkin.

Kasalliklar tarixi: Bemorning tibbiy tarixidagi kasalliklar haqida ma'lumot olish uchun SQL so'rovlari ishlatiladi. Bunda bemorning kasalliklari va davolanishi haqida ma'lumotlar alohida jadvallarda saqlanadi.

Misol:

Sql:

```
SELECT Disease, DiagnosisDate, Treatment
```

```
FROM MedicalHistory
```

```
WHERE PatientID = 12345;
```

Yangi kasallik qo'shish: Agar bemorga yangi kasallik qo'shilishi kerak bo'lsa, INSERT so'rovi ishlatiladi.

Misol:

Sql:

```
INSERT INTO MedicalHistory (PatientID, Disease, DiagnosisDate, Treatment)
VALUES (12345, "Hypertension", "2024-11-01", "Medication A");
```

Tibbiy tarixni yangilash: Agar davolash yoki simptomlar o'zgarsa, UPDATE so'rovi yordamida tibbiy tarix yangilanadi.

Misol:

Sql:

```
UPDATE MedicalHistory
SET Treatment = "Medication B"
WHERE PatientID = 12345 AND Disease = "Hypertension";
```

### 3.Laboratoriya natijalari va diagnostika

SQL laboratoriya natijalarini saqlash va bemorlarning diagnostikasi bilan bog'liq ma'lumotlarni boshqarish uchun ham ishlatiladi. Tahlil natijalari va laboratoriya tekshiruvlari ham SQL yordamida saqlanadi va tezda qayd qilinadi.

Laboratoriya natijalarini olish: SQL yordamida laboratoriya natijalari bemorlar bilan bog'langan holda olingan bo'lishi mumkin.

Misol:

Sql:

```
SELECT TestName, TestDate, Result
FROM LabResults
WHERE PatientID = 12345;
```

Yangi laboratoriya natijalari qo'shish: Yangi laboratoriya natijalarini qo'shish uchun INSERT so'rovi ishlatiladi.

Misol:

Sql:

```
INSERT INTO LabResults (PatientID, TestName, TestDate, Result)
VALUES (12345, "Blood Test", "2024-11-15", "Normal");
```

### 4.Dori-darmonlar va davolash jarayonlari

SQL tibbiyot tizimlarida dori-darmonlar, davolash jarayonlari va bemorning foydalanayotgan dori vositalarini boshqarish uchun ham ishlatiladi. Dori-darmonlar ro'yxati va davolash rejalarini qayd etishda ham SQLning yordamidan foydalaniladi.

Dori-darmonlar qo'shish: Bemorga dori-darmonlar buyurilganda, SQL so'rovi orqali bu ma'lumotlar tizimga qo'shiladi.

Misol:

Sql:

```
INSERT INTO Medications (PatientID, MedicationName, Dosage, StartDate)
VALUES (12345, "Aspirin", "50mg", "2024-11-10");
```

Dori-darmonlar ro'yxatini olish: Bemorga tayinlangan barcha dori-darmonlar ro'yxatini olish uchun SELECT so'rovi ishlatiladi.

Misol:

Sql:

```
SELECT MedicationName, Dosage, StartDate
FROM Medications
WHERE PatientID = 12345;
```

### **5. Hisobotlar va ma'lumotlarni tahlil qilish**

SQL yordamida sog'liqni saqlash tizimlarida tibbiy statistikalar va hisobotlar tayyorlanadi. Bemorlarning umumiy statistikasi, kasalliklar bo'yicha ma'lumotlar, davolash natijalari va boshqa ko'plab tahlillarni SQL yordamida amalga oshirish mumkin.

Kasalliklar bo'yicha hisobot: Tibbiyot tizimlarida kasalliklar sonini tahlil qilish uchun SQL so'rovlari ishlatiladi.

Misol:

Sql:

```
SELECT Disease, COUNT(*) AS PatientCount
FROM MedicalHistory
GROUP BY Disease;
```

Dori-darmonlar statistikasi: Hamma bemorlar uchun dori-darmonlar qo'llanilishini tahlil qilishda SQL ishlatiladi.

Misol:

Sql:

```
SELECT MedicationName, COUNT(*) AS UsageCount  
FROM Medications  
GROUP BY MedicationName;
```

## 6. Tibbiy tizimlar boshqaruvi va xavfsizlik

SQL, tibbiyot tizimlarida ma'lumotlarni xavfsiz boshqarish uchun zarur bo'lgan vositadir. DCL (Data Control Language) buyruqlari orqali ma'lumotlar bazasining xavfsizligi ta'minlanadi, foydalanuvchilarga ma'lum ruxsatlar (o'qish, yozish, o'chirish) beriladi. TCL (Transaction Control Language) yordamida ma'lumotlar uzilishsiz va to'liq saqlanishi ta'minlanadi.

Quyidagi misollarni keltirishimiz mumkin:

### 1-misol. Bemor Ma'lumotlarini Kiritish va Yangilash

Faraz qilaylik, sizda bemorlar haqidagi ma'lumotlarni saqlash uchun `patients` nomli jadval mavjud. Bu jadvalda har bir bemorning ismi, tug'ilgan sanasi, jinsiyati va kontakt ma'lumotlari saqlanadi. Yangi bemorni qo'shish va undan keyin uning ma'lumotlarini yangilash uchun quyidagi SQL so'rovlari ishlatiladi.

#### 1. Bemorni qo'shish (INSERT):

Yangi bemorni qo'shish uchun **INSERT** so'rovini ishlatamiz.

Sql:

Yangi bemor qo'shish

```
INSERT INTO patients (patient_id, name, date_of_birth, gender, contact_info)  
VALUES (1, "John Doe", "1985-07-14", "Male", "123-456-7890");
```

Bu so'rov `patients` jadvaliga yangi bemor, ya'ni **John Doe** ni qo'shadi.

#### 2. Bemor ma'lumotlarini yangilash (UPDATE):

Endi, agar bemor **John Doe** ning telefon raqami o'zgartirilgan bo'lsa, uni yangilash uchun **UPDATE** so'rovi ishlatiladi.

Sql:

Bemorning telefon raqamini yangilash

```
UPDATE patients
```

```
SET contact_info = "987-654-3210"
```

```
WHERE patient_id = 1;
```

Bu so'rov bemorning telefon raqamini yangilaydi.

### 3. Bemor ma'lumotlarini olish (SELECT):

Agar siz bemorning barcha ma'lumotlarini olishni istasangiz, **SELECT** so'rovini ishlatasiz.

Sql:

Bemor ma'lumotlarini olish

```
SELECT * FROM patients WHERE patient_id = 1;
```

Bu so'rov bemor **John Doe** ning barcha ma'lumotlarini chiqaradi.

### 2-misol. Kasalliklar tarixi va davolash ma'lumotlari

Faraz qilaylik, sizda *diagnoses* (kasalliklar tarixi) va *treatments* (davolashlar) nomli jadvallar mavjud. Har bir bemorning kasalliklari va davolashlari haqida ma'lumotlar shu jadvallarda saqlanadi. Misol uchun, bemor **John Doe** ning kasallik tarixini va davolanish jarayonini qo'shamiz.

### 1. Kasallikni qo'shish (INSERT):

Bemorning kasallik tarixini qo'shish uchun **INSERT** so'rovi ishlatiladi.

Sql:

Kasallik qo'shish

```
INSERT INTO diagnoses (patient_id, diagnosis_code, diagnosis_date)
```

```
VALUES (1, "I10", "2024-02-25");
```

Bu so'rov bemor **John Doe**ga "**Hypertension**" (yuqori qon bosimi) tashxisini qo'shadi.



### 1. Davolashni qo'shish (INSERT):

Davolash jarayonini qo'shish uchun yana **INSERT** so'rovini ishlatamiz.

Sql:

Davolash qo'shish

```
INSERT INTO treatments (patient_id, treatment_type, start_date, end_date)
```

```
VALUES (1, "Chemotherapy", "2024-03-01", "2024-06-01");
```

Bu so'rov bemor **John Doe** ga "**Chemotherapy**" (kimyoterapiya) davolashni qo'shadi.

### 2. Kasalliklar va davolashlarni olish (SELECT):

Agar siz bemorning kasallik tarixini va davolashlarini olishni istasangiz, **SELECT** so'rovlarini ishlatasiz. Buning uchun **JOIN** operatoridan foydalanamiz, chunki kasalliklar va davolashlar alohida jadvallarda saqlanadi.

Sql:

Bemorning kasalliklar va davolashlar tarixini olish

```
SELECT p.name, d.diagnosis_code, d.diagnosis_date, t.treatment_type, t.start_date,  
t.end_date
```

```
FROM patients p
```

```
JOIN diagnoses d ON p.patient_id = d.patient_id
```

```
JOIN treatments t ON p.patient_id = t.patient_id
```

```
WHERE p.patient_id = 1;
```

Bu so'rov bemor **John Doe** ga tegishli kasalliklar va davolashlar haqida barcha ma'lumotlarni beradi.

### 3-misol. Laboratoriya natijalari va tahlillar

Tibbiyot tizimlarida laboratoriya natijalari ham muhim ahamiyatga ega. Faraz qilaylik, bemorning laboratoriya tahlillari `lab_results` nomli jadvalda saqlanadi. SQL yordamida laboratoriya natijalarini qo'shish va olish mumkin.

### 1. Laboratoriya natijalarini qo'shish (INSERT):

Bemorning laboratoriya natijasini qo'shish uchun **INSERT** so'rovi ishlatiladi.

Sql:

Laboratoriya natijalarini qo'shish

```
INSERT INTO lab_results (patient_id, test_type, result_value, test_date)
```

```
VALUES (1, "Blood Test", "Normal", "2024-03-10");
```

Bu so'rov bemor **John Doe** ning **qon tekshiruvi** natijasini qo'shadi.

## 2.Laboratoriya natijalarini olish (SELECT):

Bemorning laboratoriya natijalarini olish uchun quyidagi **SELECT** so'rovini ishlatish mumkin.

Sql:

Laboratoriya natijalarini olish

```
SELECT test_type, result_value, test_date
```

```
FROM lab_results
```

```
WHERE patient_id = 1;
```

Bu so'rov bemor **John Doe** ning barcha laboratoriya natijalarini chiqaradi.

## 4-misol. Sog'liqni saqlash hisobotlarini yaratish

Tibbiyot tizimlarida keng qamrovli hisobotlar yaratish hamda statistika tahlillari olib borish uchun SQLdan foydalanish mumkin. Masalan, barcha kasalliklar bo'yicha statistika yaratish uchun quyidagi so'rovni yozish mumkin:

Sql:

Kasalliklar bo'yicha statistika (kasallik kodi va soni)

```
SELECT diagnosis_code, COUNT(patient_id) AS diagnosis_count
```

```
FROM diagnoses
```

```
GROUP BY diagnosis_code;
```

Bu so'rov turli kasalliklar (kasallik kodi bo'yicha) va ularning nechta bemorda mavjudligini hisoblab beradi.

## Xulosa

SQL so'rovlar tili tibbiyot va sog'liqni saqlash tizimlarida ma'lumotlar bazasini samarali boshqarish va tahlil qilishda asosiy vosita hisoblanadi. Bemorlar

ma'lumotlarini saqlash, tibbiy tarixni boshqarish, laboratoriya natijalarini tahlil qilish va dori-darmonlarni nazorat qilishda SQLning o'rni juda muhimdir. SQL yordamida sog'liqni saqlash tizimlari yanada samarali ishlaydi, ma'lumotlar tezda tahlil qilinadi va barcha tibbiy ma'lumotlar to'g'ri va aniq saqlanadi.

#### Foydalanilgan Adabiyotlar:

1. **Casteel, R. (2016).** *SQL For Dummies*. John Wiley & Sons, Inc.  
Bu kitob SQLni yangi boshlovchilar uchun tushunarli tilda tushuntiradi va turli SQL so'rovlarini misollar bilan ko'rsatadi.
2. **Date, C. J. (2004).** *An Introduction to Database Systems* (8th ed.). Addison-Wesley.  
C. J. Date tomonidan yozilgan ushbu kitob ma'lumotlar bazasining nazariy asoslari va SQL so'rovlar tili haqida chuqur bilim beradi.
3. **Elmasri, R., & Navathe, S. B. (2015).** *Fundamentals of Database Systems* (7th ed.). Addison-Wesley.  
Ma'lumotlar bazasi tizimlarining asosiy konseptlari va SQL sintaksisi bo'yicha to'liq tushuncha beradi.
4. **Golan, G. (2017).** *SQL Queries for Mere Mortals: A Hands-On Guide to Data Manipulation in SQL*. Addison-Wesley.  
Bu kitob SQL so'rovlarini yozish va ma'lumotlar bazasi bilan ishlashni o'rganish uchun amaliy qo'llanma sifatida xizmat qiladi.
5. **Garcia-Molina, H., Ullman, J. D., & Widom, J. (2009).** *Database Systems: The Complete Book* (2nd ed.). Prentice Hall.  
Ma'lumotlar bazalari tizimlari va SQLni chuqur o'rganish uchun bir nechta misollarni o'z ichiga olgan keng qamrovli manba.
6. **Ben-Gan, I. (2007).** *T-SQL Fundamentals*. Microsoft Press.  
T-SQL sintaksisi va SQL Server muhitida ishlashni o'rganish uchun foydali resurs.
7. **SQL.org. (2024).** *SQL Basics and Tutorials*. <https://www.sql.org>  
SQLning asosiy buyruqlari, operatorlari va strukturalari bo'yicha onlayn ma'lumot.

8. **W3Schools.** (2024). *SQL Tutorial.* <https://www.w3schools.com/sql/>  
SQL tili bo'yicha keng tarqalgan onlayn manba bo'lib, unda so'rovlar va SQL sintaksisiga oid to'liq ma'lumotlar taqdim etiladi.
9. Tojimatov, I. N., Topvoldiyeva, H., Karimova, N., & Inomova, G. (2023). **ГРАФИК МА'LUMOTLAR BAZASI.** Евразийский журнал технологий и инноваций, 1(4), 75-84.
10. Nurmamatovich, T. I., & Azizjon o'g, N. A. Z. (2024). The SQL server language and its structure. *American Journal of Open University Education*, 1(1), 11-15.
11. Nurmamatovich, T. I. (2024). **MY SQL MISOLIDA LOYIHA YARATISH.** Ta'limda raqamli texnologiyalarni tadbiq etishning zamonaviy tendensiyalari va rivojlanish omillari, 31(2), 82-90.
12. Ro'zimatov, J. I., & Nurmamatovich, T. I. (2024). SQL tili tarixi, vazifasi, turlari va rejimlari.
13. Nurmamatovich, T. I. (2024). **NORMALLASHTIRISH. NORMAL FORMALAR.** worldly knowledge conferens, 7(2), 597-599.
14. Isroil, T. (2023). **NOSQL MA'LUMOTLAR BAZASI: TANQIDIY TAHLIL VA TAQQOSLASH.** IJODKOR O'QITUVCHI, 3(28), 134-146.