

## SQL TILI

*Tojimamatov Israiljon Nurmatovich Farg'ona davlat universiteti*

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

*O'ktamjonova Nilufar Abdurahmon qizi Farg'ona davlat universiteti*

*[karimberdiyevanilufar625@gmail.com](mailto:karimberdiyevanilufar625@gmail.com)*

**Annotatsiya.** Ushbu maqolada SQL (Structured Query Language) tilining asosiy tushunchalari, ma'lumotlar bazalari bilan ishlash tamoyillari va zamonaviy axborot tizimlaridagi ahamiyati yoritilgan. SQL tilining asosiy buyruqlari hamda ma'lumotlarni yaratish, o'zgartirish va boshqarish jarayonlari misollar orqali tushuntirilgan. Maqola ma'lumotlar bazalari bilan ishlashni o'rganayotgan talabalar va boshlovchi dasturchilar uchun mo'ljallangan.

**Аннотация** В статье рассматриваются основные понятия языка SQL (Structured Query Language), принципы работы с базами данных и его значение в современных информационных системах. Описаны основные команды SQL, а также процессы создания, изменения и управления данными. Статья предназначена для студентов и начинающих программистов.

**Annotation.** The article discusses the basic concepts of SQL (Structured Query Language), principles of working with databases, and its importance in modern information systems. The main SQL commands and the processes of creating, modifying, and managing data are explained. The article is intended for students and beginner programmers.

**Kalit so'zlar** .SQL, ma'lumotlar bazasi, so'rov, jadval, ustun, qator, select, insert, update, delete, where, join, primary key, foreign key, indeks, trigger, view, normalizatsiya, tranzaksiya, xavfsizlik

**Ключевые слова** . SQL, база данных, запрос, таблица, столбец, строка, select, insert, update, delete, where, join, первичный ключ, внешний ключ, индекс, триггер, представление, нормализация, транзакция, безопасность

**Keywords** .SQL, database, query, table, column, row, select, insert, update,

delete, where, join, primary key, foreign key, index, trigger, view, normalization, transaction, security

### **SQL tili asoslari va sintaksisi**

SQL (Structured Query Language) – bu ma'lumotlar bazasi (MB) bilan ishlash uchun mo'ljallangan dasturlash tili. Asosiy maqsadi ma'lumotlarni yaratish, ularga murojaat qilish va ularga ishlov berishni osonlashtirishdir. SQL tili orqali ma'lumotlar bazasini yaratish (CREATE), o'zgartirish (UPDATE), o'chirish (DELETE) va undan ma'lumotlarni olish (SELECT) mumkin. Juda ko'p turdagi MB menejment tizimlari (MBMS) SQL tilini qo'llab-quvvatlaydi, masalan, MySQL, PostgreSQL, Oracle, MS SQL Server va boshqalar.

SQL tili sintaksisi va asoslari:

1. CREATE: Yangi jadval yoki MB obyektlarini yaratish uchun ishlatiladi.

- 'CREATE DATABASE' sintaksisi yangi MB yaratadi.

- 'CREATE TABLE' sintaksisi yangi jadval yaratish uchun qo'llaniladi. Unda ustunlarning turlari va cheklovlari beriladi.

Misol:

```
```sql
CREATE TABLE talabalar (
    id INT PRIMARY KEY,
    ism VARCHAR(50),
    yosh INT
);
```
```

2. SELECT: Ma'lumotlarni olish uchun ishlatiladi. Bu buyruq MB dan kataklardagi ma'lumotlarni tanlash uchun qo'llaniladi.

- Asosiy sintaksisi:

```
```sql
```

```
SELECT ustun1, ustun2 FROM jadval_nomi WHERE shartlar;  
...
```

- `SELECT \* FROM jadval\_nomi;` barcha ustunlarni tanlab oladi.

3. INSERT INTO: Jadvallariga yangi ma'lumotlar qo'shish uchun ishlatiladi.

- Asosiy sintaksisi:

```
```sql  
INSERT INTO jadval_nomi (ustun1, ustun2) VALUES (qiymat1, qiymat2);  
...
```

4. UPDATE: Jadvaldagi mavjud ma'lumotlarni yangilash uchun xizmat qiladi.

- Asosiy sintaksisi:

```
```sql  
UPDATE jadval_nomi SET ustun1 = yangi_qiymat WHERE shart;  
...
```

5. DELETE: Jadvaldan ma'lumotlarni o'chirish uchun ishlatiladi.

- Asosiy sintaksisi:

```
```sql  
DELETE FROM jadval_nomi WHERE shart;  
...
```

6. ALTER TABLE: Jadval tuzilmasini o'zgartirish uchun qo'llanadi, masalan, yangi ustun qo'shish yoki mavjud ustunni o'chirish.

- Ustun qo'shish:

```
```sql  
ALTER TABLE jadval_nomi ADD ustun_nomi tur;  
...
```

- Ustun o'chirish:

```
```sql
```

```
ALTER TABLE jadval_nomi DROP COLUMN ustun_nomi;  
...
```

7. JOIN: Bir nechta jadvaldagi ma'lumotlarni birdaniga olish va birlashtirish uchun ishlatiladi.

- 'INNER JOIN', 'LEFT JOIN', 'RIGHT JOIN', 'FULL JOIN' kabi turlari bor.

- Misol:

```
```sql  
SELECT A.ism, B.kurs_nomi  
FROM talabalar A  
INNER JOIN kurslar B ON A.kurs_id = B.id;  
...
```

8. GROUP BY: Tanlangan ma'lumotlarni guruhlash imkonini beradi.

- Misol:

```
sql  
SELECT ymax, COUNT(*) FROM talabalar GROUP BY ymax;  
...
```

9. ORDER BY: Ma'lumotlarni o'zgacha tartibda ko'rsatish uchun ishlatiladi.

- 'ASC' muayyan ustunni o'sish tartibida,

- 'DESC' esa kamayish tartibida tartiblaydi.

- Misol:

```
```sql  
SELECT ism, yosh FROM talabalar ORDER BY yosh DESC;  
...
```

10. \*\*HAVING\*\*: 'GROUP BY' bilan ishlatiladi va guruhlarining natijalariga cheklov qo'yish imkonini beradi.

- Misol:

```
sql
SELECT ymax, COUNT(*)
FROM talabalar
GROUP BY ymax
HAVING COUNT(*) > 1
```

11. DISTINCT: Takrorlanuvchi qiymatlarni chiqarib tashlashda yordam beradi.

- Misol:

```
sql
SELECT DISTINCT ism FROM talabalar;
```

12. UNION: Bir nechta `SELECT` buyruqlarining natijalarini birlashtirish uchun ishlatiladi.

- Misol:

```
sql
SELECT ism FROM talabalar_2022
UNION
SELECT ism FROM talabalar_2023;
```

Bu SQL tili va uning asosiy sintaksisi haqida umumiy tushuncha beradi. Tilning imkoniyatlari juda keng va murakkab dasturiy ta'minot yaratishda asosiy vositalardan biri hisoblanadi.

### **Ma'lumotlar bazasi va jadval yaratish (CREATE DATABASE, CREATE TABLE)**

SQL (Structured Query Language) — bu ma'lumotlar bazalari bilan ishlash va ulardan ma'lumotlar olish uchun mo'ljallangan til. SQL orqali foydalanuvchilar ma'lumotlar bazasini yaratish, saqlash, yangilash va o'chirish, shuningdek, ma'lumotlar ustida turli operatsiyalarni bajarishlari mumkin.

2. Ma'lumotlar bazasi va jadval yaratish (CREATE DATABASE, CREATE

TABLE)

Ma'lumotlar bazasini yaratish

Ma'lumotlar bazasini yaratish uchun SQL tilida `CREATE DATABASE` buyrug'i ishlatiladi. Bu buyruq quyidagi sintaksisga ega:

```
CREATE DATABASE ma'lumotlar_bazasi_nomi;
```

Bu yerda `ma'lumotlar\_bazasi\_nomi` foydalanuvchi yaratmoqchi bo'lgan ma'lumotlar bazasining nomi hisoblanadi. Misol uchun, agar "maktab" nomli ma'lumotlar bazasini yaratmoqchi bo'lsangiz, buyruq quyidagicha bo'ladi:

```
CREATE DATABASE maktab;
```

Ma'lumotlar bazasi yaratib bo'lgandan so'ng, unda jadval va boshqa obyektlarni yaratish mumkin bo'ladi.

Jadval yaratish

Ma'lumotlar bazasida ma'lumotlarni saqlash uchun jadval (table)lardan foydalaniladi. Har bir jadval satrlar (row) va ustunlar (column)dan iborat bo'lib, ular orqali ma'lumotlar tuziladi. Jadval yaratish uchun `CREATE TABLE` buyrug'i ishlatiladi. Sintaksisi quyidagicha:

```
CREATE TABLE jadval_nomi (  
    ustun_1 tur [MAJLISLAR],  
    ustun_2 tur [MAJLISLAR],  
    ...,  
    ustun_n tur [MAJLISLAR]  
);  
...
```

Misol uchun, agar talabalarning ma'lumotlarini saqlovchi `talaba` nomli jadval yaratmoqchi bo'lsangiz, asosiy ma'lumotlar ismi, familiyasi, yoshi va ID raqamini o'z ichiga olishi mumkin:

```

```
CREATE TABLE talaba (  
    id INT PRIMARY KEY,  
    ism VARCHAR(50),  
    familiya VARCHAR(50),  
    yosh INT
```

```
);
```

```

Bu yerda:

- `id` - talabaning yagona identifikatori. `INT` turi butun sonlarni ifodalaydi; `PRIMARY KEY` uni jadvalda noyob qiymat qilish uchun qo'shadigan cheklovdir.
- `ism` va `familiya` - tegishli holda, talabaning ismi va familiyasi. `VARCHAR(50)` - belgilar satri turi bo'lib, 50 ta belgigacha o'z ichiga olishi mumkin.
- `yosh` - talabaning yoshi, bu yerda ham `INT` turi ishlatiladi.

Qo'shimcha parametrlar va cheklovlar

Jadval yaratishda ustunlarga qo'shimcha parametrlar va cheklovlarni qo'llash mumkin:

- PRIMARY KEY: Jadval ustuni uchun yagona identifikatorni belgilaydi.
- FOREIGN KEY: Boshqa jadvaldagi ustun bilan bog'lanishni belgilaydi, shu orqali jadval o'rtasida aloqa o'rnatiladi.
- NOT NULL: Ustun qiymatining `NULL` bo'lishiga yo'l qo'ymaydi.

- UNIQUE: Ustun qiymatlari orasida takrorlanishni bartaraf etadi.
- DEFAULT: Ustun uchun standart qiymat belgilaydi, agar foydalanuvchi uni kiritmasa.

Misol uchun, 'CREATE TABLE' buyrug'ida qo'shimcha cheklovlar bilan jadval yaratish:

```
...  
  
CREATE TABLE o'qituvchi (  
    o'qituvchi_id INT PRIMARY KEY,  
    ism VARCHAR(50) NOT NULL,  
    fan VARCHAR(50),  
    yoshi INT CHECK (yoshi >= 0),  
    daraja VARCHAR(20) DEFAULT 'o'qituvchi'  
);  
...
```

Bu misolda:

- 'o'qituvchi\_id' — asosiy kalit sifatida ishlatiladi.
- 'ism' — 'NOT NULL' bilan yuq, shuning uchun har bir o'qituvchining ismi bo'lishi kerak.
- 'yoshi' — 'CHECK' bilan cheklangan, ushbu ustunda kiritilgan qiymatlar noldan kichik bo'lishi mumkin emas.
- 'daraja' — 'DEFAULT' bilan, standart qiymat sifatida "o'qituvchi" deb belgilangan.

### **SELECT operatori va asosiy so'rovlar**

SQL (Structured Query Language) tili bizga ma'lumotlar bazasiga kirish va ma'lumotlar bilan ishlash uchun kuchli vositalarni beradi. SQL tili orqali eng ko'p ishlatiladigan buyruqlardan biri bu SELECT operatoridir. SELECT buyrug'i ma'lumotlarni qaytarish uchun ishlatiladi, ya'ni ma'lumotlar bazasidagi kerakli



ma'lumotlarni olish uchun yoziladi.

SELECT operatori juda kuchli va moslashuvchan bo'lib, foydalanuvchilariga bir qancha imkoniyatlarni taqdim etadi:

1. ASOS ANATOMYASI: SELECT operatori odatda tanlash kerak bo'lgan ustun yoki ustunlarni ko'rsatish bilan boshlanadi, masalan:

```

```
SELECT ustun1, ustun2 FROM jadval_nomi;
```

```

Bu yerda `ustun1` va `ustun2` belgilanayotgan ustunlar bo'lib, `jadval\_nomi` esa ma'lumotlar olingan jadvalni nomini anglatadi.

2. ALL USTUN TANLASH: Ba'zan barcha ustunlar tanlanishi talab qilinishi mumkin, bu holatda \* (yulduzcha) ishlatiladi, masalan:

```

```
SELECT * FROM jadval_nomi;
```

```

Bu yondashuv jadvaldagi barcha ustunlarni tanlash imkonini beradi.

3. IZHORLASH VA SARALASH: SELECT statementiga AND, OR va XOR kabi tavsiflovchi shartlar qo'shilib, WHERE bandi yordamida ma'lumotlar filtrlash mumkin:

```

```
SELECT ustun1, ustun2 FROM jadval_nomi WHERE shart;
```

```

Misol uchun, agar yonalish `WHERE yoshi > 20` kabi berilgan bo'lsa, bu SELECT faqat yosh qiymati 20-dan katta bo'lgan satrlarni tanlaydi.

4. ORDER BY TARTIBGA SOLISH: Ma'lumotlarni o'qishdan oldin saralash

kerak bo'lsa, ORDER BY bandidan foydalanamiz. Odatda o'sish tartibida (ASC) yoki kamayish tartibida (DESC) belgilanadi:

```

```
SELECT ustun1 FROM jadval_nomi ORDER BY ustun1 ASC;
```

```

ORDER BY DESC tartibida yozilsa, u satrlarni teskari tartibda chiqaradi.

5. LIMIT BILAN CHEKLASH: Natijalarning sonini cheklash kerak bo'lsa, LIMIT bandidan foydalaniladi:

```

```
SELECT * FROM jadval_nomi LIMIT 10;
```

```

Bu yondashuv yuqoridagi ma'lumotlardan faqat 10 row ni chiqaradi.

6. JOYLASHIFNIRISH: JOINLARNI ISHLATISH: SELECT operatori boshqa jadvallardan ma'lumotlarni birgalikda qaytarish imkoniyatini beradi, bu xarakterli joinlar yordamida amalga oshiriladi. Eng ko'p ishlatiladiganlari:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

Masalan, INNER JOIN foydalanish:

```

```
SELECT a.ustun1, b.ustun2 FROM jadval_a a INNER JOIN jadval_b b ON  
a.id = b.id;
```

```

7. GRUPPALASH VA IQTLAR: Ma'lumotlarni guruhlash kerak bo'lsa GROUP BY va HAVING bandlari qo'llaniladi:

'''

```
SELECT ustunlar, COUNT(*) FROM jadval_nomi GROUP BY ustun;
```

'''

#### 8. SUBSELECT VA MALUMOTLARNI QO`SHIMCHA IZLASH:

Ma'lumotlarni specific tanlash uchun subquerylar foydalaniladi. Subquerylar bitta SELECT statement ichida ichki savollar yaratishga imkon beradi, masalan:

'''

```
SELECT ustun1 FROM jadval_nomi WHERE ustun2 IN (SELECT ustun2  
FROM boshqa_jadval);
```

'''

#### **Xulosa**

SELECT operatori, yukoridagi elementlar bilan birgalikda, SQL tilining eng kuchli va keng qo'llaniladigan qismi sanaladi. Har xil murakkablikdagi savollar tuzishda yordam beradi. Aniq va puxta o'ylangan SELECT so'rovlar orqali ma'lumotlarni tahlil qilish va chiqarib olish jarayonini optimallashtirish mumkin.

#### **Foydalanilgan adabiyotlar**

1. Date C. J. *An Introduction to Database Systems*. — Addison-Wesley, 2004.
2. Elmasri R., Navathe S. *Fundamentals of Database Systems*. — Pearson, 2016.
3. Silberschatz A., Korth H. F., Sudarshan S. *Database System Concepts*. — McGraw-Hill, 2019.
4. Beaulieu A. *Learning SQL*. — O'Reilly Media, 2020.
5. Oppel A. *Databases: A Beginner's Guide*. — McGraw-Hill, 2015.

#### **Internet manbalari:**

6. Oracle. *SQL Language Reference*. — <https://docs.oracle.com>
7. PostgreSQL Global Development Group. *PostgreSQL Documentation*. — <https://www.postgresql.org/docs>
8. W3Schools. *SQL Tutorial*. — <https://www.w3schools.com/sql>