

SQL



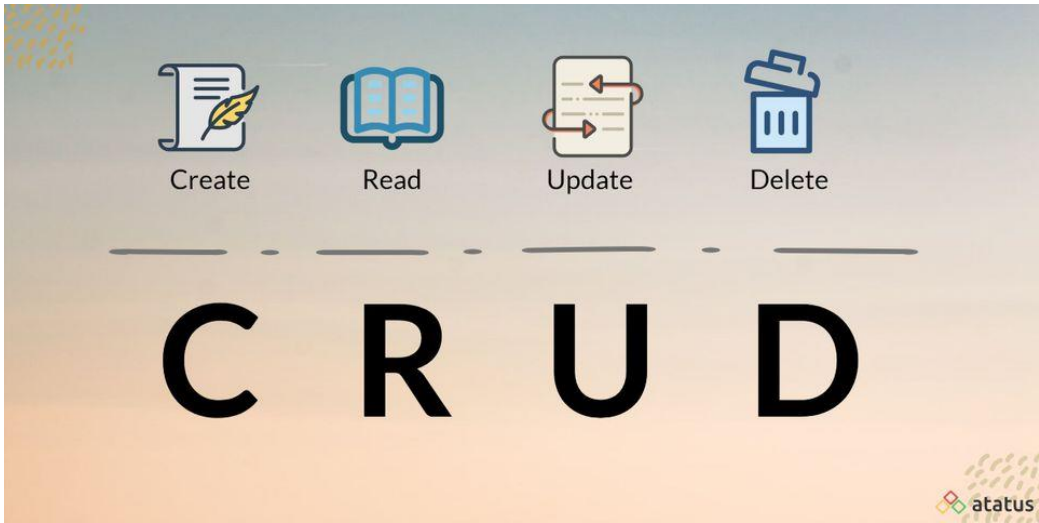
Chananel Perel - 2024

SQL (Structured Query Language) היא שפת מחשב הצהרתית לטיפול, תשאול ועיבוד מידע בבסיסי נתונים יחסיים, שפותחה על ידי IBM, והתבססה במקור על אלגברה רלציונית. השפה מאפשרת שליפת נתונים ועדכונים ויצירת טבלה ושינויה.

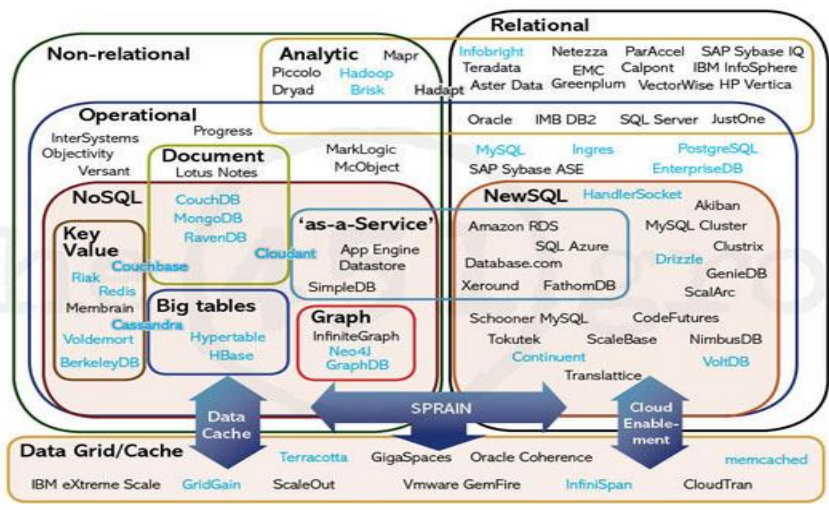
SQL הייתה אחת השפות הראשונות המיועדת לבסיס נתונים יחסי. הושקה בשנת 1974 לפני 48 שנה. (!היא כיום השפה הנפוצה ביותר לתשאול בבסיסי נתונים יחסיים.

SQL היא שפת מחשב הצהרתית, כלומר, היא מתארת רק מה לבצע ולא כיצד לבצע, ומתמקדת במהות ולא בטכניקה. מכיוון שכך, ניתן להשתמש באותו משפט SQL בבסיסי נתונים שונים, הפועלים בדרכים שונות לחלוטין. אי תלותה של SQL במבנה הנתונים, בנוסף על עוצמתה של השפה, המאפשרת לאחזר חתכים מורכבים בפשטות רבה, דבר המביא להגדלת הפריון של מתכנתי השפה, והתפשטותה למשתמשים שאינם מתכנתים, הפכו אותה לשפה המקובלת ביותר כיום בניהול בסיסי נתונים.

CRUD



סוגי בסיס נתונים



Cheat Sheet

- בסלנג "שליף" הוא דף המכיל גרסה מקוצרת של הערות אשר נועדו לעיון מהיר
- מאד מקובל בתור דף חזרה ורענון על נושא מסוים
- נחפש בגוגל SQL Cheat Sheet: אפשר להוסיף PDF למשל)
- מאגר גדול/ <https://www.cheat-sheets.org/>
- <https://learnsql.com/blog/sql-basics-cheat-sheet/>

SQL CHEAT SHEET <http://www.sqltutorial.org>

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
Query data in columns c1, c2 from a table.

SELECT * FROM t;
Query all rows and columns from a table.

SELECT c1, c2 FROM t
WHERE condition;
Query data and filter rows with a condition.

SELECT DISTINCT c1 FROM t
WHERE condition;
Query distinct rows from a table.

SELECT c1, c2 FROM t
ORDER BY c1 ASC (DESC);
Sort the result set in ascending or descending order.

SELECT c1, c2 FROM t
LIMIT n OFFSET offset;
Skip offset rows and return the next n rows.

SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
Group rows using an aggregate function.

SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
Filter groups using HAVING clause.
```

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
Inner join t1 and t2.

SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
Left join t1 and t2.

SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
Right join t1 and t2.

SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
Perform full outer join.

SELECT c1, c2
FROM t1
CROSS JOIN t2;
Produce a Cartesian product of rows in tables.

SELECT c1, c2
FROM t1
ANOTHER JOIN t2 ON condition;
Another way to perform cross join.
```

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION (ALL)
SELECT c1, c2 FROM t2;
Combine rows from two queries.

SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
Return the intersection of two queries.

SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
Subtract a result set from another result set.

SELECT c1, c2 FROM t1
WHERE c1 (NOT) LIKE pattern;
Query rows using pattern matching "%...".

SELECT c1, c2 FROM t1
WHERE c1 (NOT) IN value_list;
Query rows in a list.

SELECT c1, c2 FROM t1
WHERE c1 BETWEEN low AND high;
Query rows between two values.

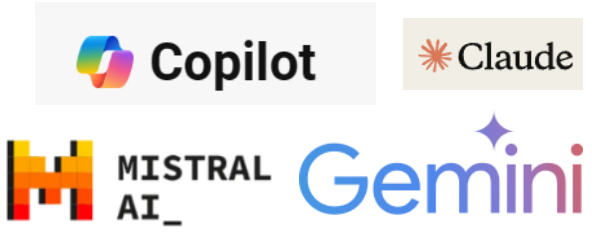
SELECT c1, c2 FROM t1
WHERE c1 IS (NOT) NULL;
Check if values in a table is NULL or not.
```

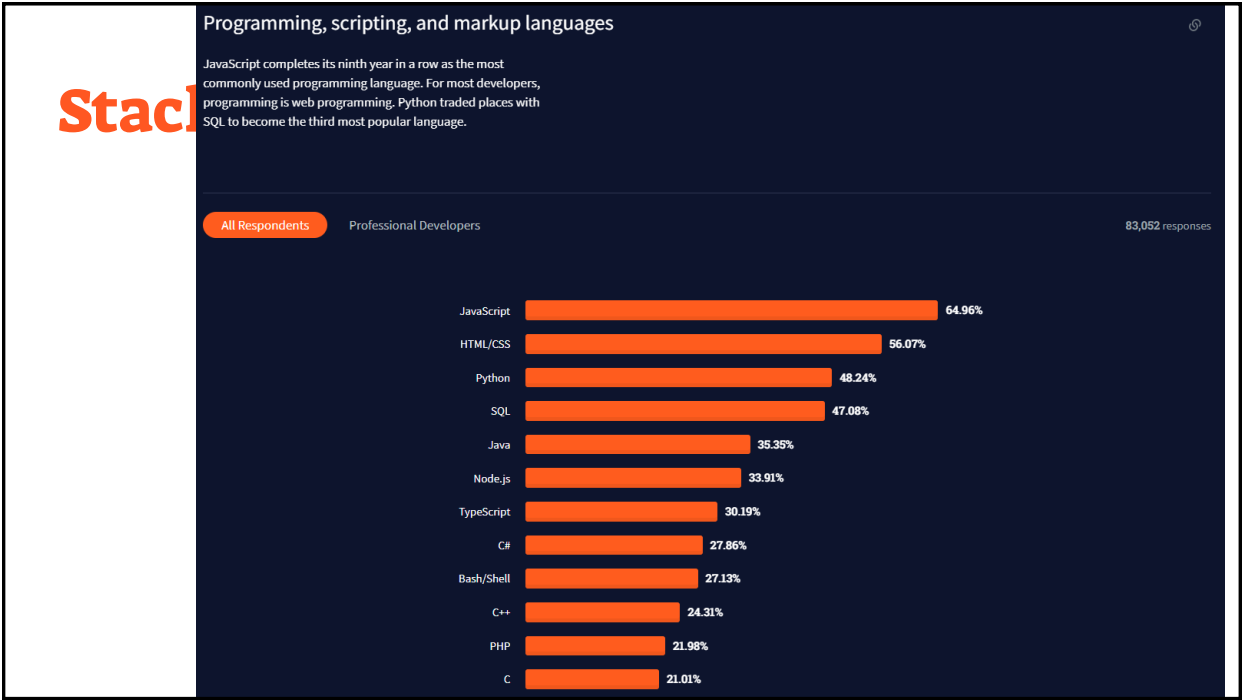
Stack Overflow stackoverflow

- <https://stackoverflow.com>
- אתר שאלות ותשובות בנושא תוכנה הגדול בעולם .
- כל אחד יכול לשאול וכל אחד יכול לענות, נותנים ניקוד אחד לשני.
- מאד מאד שימושי בכל תחומי התוכנה.
- עושים סקר שנתי בנושא (עשרות אלפי עונים) תוכנה ושם ניתן לראות טרנדים.

כלים של AI

- <https://chat.openai.com/>
- <https://copilot.microsoft.com/>
- <https://github.com/features/copilot>
- <https://gemini.google.com/app>
- <https://claude.ai/chats>
- <https://llama.meta.com/>
- <https://mistral.ai/>

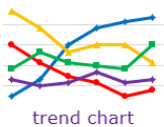




DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



416 systems in ranking, November 2023

Rank			DBMS	Database Model	Score		
Nov 2023	Oct 2023	Nov 2022			Nov 2023	Oct 2023	Nov 2022
1.	1.	1.	Oracle	Relational, Multi-model	1277.03	+15.61	+35.34
2.	2.	2.	MySQL	Relational, Multi-model	1115.24	-18.07	-90.30
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	911.42	+14.54	-1.09
4.	4.	4.	PostgreSQL	Relational, Multi-model	636.86	-1.96	+13.70
5.	5.	5.	MongoDB	Document, Multi-model	428.55	-2.87	-49.35
6.	6.	6.	Redis	Key-value, Multi-model	160.02	-2.95	-22.03
7.	7.	7.	Elasticsearch	Search engine, Multi-model	139.62	+2.48	-10.70
8.	8.	8.	IBM Db2	Relational, Multi-model	136.00	+1.13	-13.56
9.	9.	10.	SQLite	Relational	124.58	-0.56	-10.05
10.	10.	9.	Microsoft Access	Relational	124.49	+0.18	-10.53

SQLITE



- <https://sqlite.org/index.html>
- <https://github.com/sqlite/sqlite>
- **SQLite is a open source, C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.**
- **Most of SQL, not all:** <https://sqlite.org/omitted.html>
- **Function:** https://sqlite.org/lang_corefunc.html

SQL Clients

- **Small and Basic - but good for us now:**
 - **DB Browser for SQLite** - <https://sqlitebrowser.org/>
 - **SQLiteSpy** - <https://yunqa.de/delphi/apps/sqlitespy/index>
 - **SQLite Viewer (web based)** - <https://inloop.github.io/sqlite-viewer/>
- **Big with many Features:**
 - **DBeaver** - <https://dbeaver.io/>
 - **HeidiSQL** - <https://www.heidisql.com/>

Python connection to SQLITE

- **Built into python - sqlite3**
 - <https://docs.python.org/3/library/sqlite3.html>

```
import sqlite3
con = sqlite3.connect("data.db") # or ":memory:"
cur = con.cursor()
res = cur.execute("SELECT * FROM names_table")
data = res.fetchall() # or can return only one: cur.fetchone()
con.close()
```
- **Pandas DF - pandas**
 - https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_sql.html

```
import pandas as pd
import sqlite3
con = sqlite3.connect("data.db") # or ":memory:"
df = pd.read_sql_query("SELECT * FROM names_table", con) # or: read_sql_table
con.close()
## or even:
df = pd.read_sql_query("SELECT * FROM names_table", 'sqlite:///data.db')
```
- **SQLAlchemy - Python SQL toolkit that gives application developers the full power and flexibility of SQL** - <https://www.sqlalchemy.org/>

DATA BASE

בסיס נתונים

TABLE

טבלה

ROW / TUPLE / RECORD
שורה / רשומה

COLUMN / ATTRIBUTE
עמודה / תכונה

CELL / FIELD
תא / שדה

SCHEMA
סכמה

הערות כלליות

- בסיס נתונים הוא אוסף טבלאות
- בטבלה אין לדעת באיזה סדר נשמרים העמודות והשורות
- תוצאה של שאילתה על טבלאות היא טבלה ..
- (לפעמים זו טבלה עם עמודה אחת ושורה אחת ולפעמים זו טבלה ריקה) ..
- חשיבות התוצאות היא בכמות המדויקת (לא יותר ולא פחות) של השורות (ונכונותם) שם חשובה הלוגיקה, (ובהצגת המידע המדויק.
- בתצוגה של העמודות המבוקשות ניתן להציג עמודות נוספות ולרוב זה לא מהווה בעיה או טעות) זו רק דרך הצגה.

DATA TYPES

- **NULL** - The value is a NULL value.
- **INTEGER** - The value is a signed integer, stored in 0, 1, 2, 3, 4, 6, or 8 bytes depending on the magnitude of the value.
- **REAL** - The value is a floating point value, stored as an 8-byte IEEE floating point number.
- **TEXT** - The value is a text string, stored using the database encoding (UTF-8, UTF-16BE or UTF-16LE).
- **BLOB** - The value is a blob of data, stored exactly as it was input.

DATA TYPES - notes

- **Boolean Datatype**
SQLite does not have a separate Boolean storage class. Instead, Boolean values are stored as integers 0 (false) and 1 (true). SQLite recognizes the keywords "TRUE" and "FALSE", those keywords are really just alternative spellings for the integer literals 1 and 0 respectively.
- **Date and Time Datatype**
SQLite does not have a storage class set aside for storing dates and/or times. Instead, the built-in Date And Time Functions of SQLite are capable of storing dates and times as TEXT, REAL, or INTEGER values.

MISC

Quotes - The SQL standard requires double-quotes around identifiers and single-quotes around string literals.

- "this is a legal SQL column name"
- 'this is an SQL string literal'
 - Note: inside string literals the case IS important!

CASE INSENSITIVE - all keywords, table and columns names can be written in any case and will refer to same value (SELECT = Select = sELeT = select ← all will work)

Convention: Language keyword - ALL UPPERCASE
 Table name - prefer one word lowercase
 Column name - lowercase with underscore separator (_) [no space]

<https://www.sqlstyle.guide/>

```
SELECT first_name AS fn
FROM staff AS s1
JOIN students AS s2
ON s2.mentor_id = s1.staff_id
```

MISC

- The **LIKE** operator does a pattern matching comparison. The operand to the right of the **LIKE** operator contains the pattern and the left hand operand contains the string to match against the pattern.
 - A percent symbol ("%") in the **LIKE** pattern matches any sequence of zero or more characters in the string.
 - An underscore ("_") in the **LIKE** pattern matches any single character in the string.
 - Any other character matches itself or its lower/upper case equivalent (i.e. case-insensitive matching)
- **SELECT ***
FROM city
WHERE name LIKE 'k%' *-- including Kabul (case INSENSITIVE)*
WHERE name IN ('kabul') *-- NOT including Kabul (case SENSITIVE)*
WHERE name = 'kabul' *-- NOT including Kabul (case SENSITIVE)*

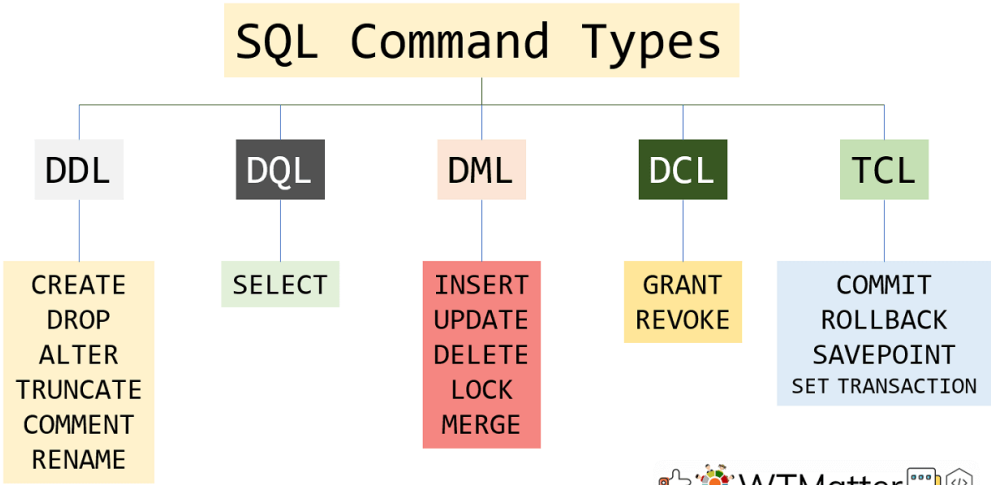
SQL Parts

SQL commands are mainly categorized into five categories as:

- **DDL – Data Definition Language**
- **DQL – Data Query Language**
- **DML – Data Manipulation Language**
- **DCL – Data Control Language**
- **TCL – Transaction Control Language**

<https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/>

SQL Parts



SQL Quirks and Caveats

- **SQL Features That SQLite Does Not Implement**
<https://www.sqlite.org/omitted.html>
- **Quirks, Caveats, and Gotchas In SQLite**
<https://www.sqlite.org/quirks.html>
 - **Flexible Typing**
SQLite is flexible and forgiving with regard to datatypes. Datatypes are advisory rather than mandatory. Some say that SQLite is "weakly typed" and that other SQL databases are "strongly typed".
 - **Double-quoted String Literals Are Accepted**
The SQL standard requires double-quotes around identifiers and single-quotes around string literals. SQLite accepts both of the above. But SQLite will also interpret a double-quotes string as string literal if it does not match any valid identifier.
 - **Aggregate Queries Can Contain Non-Aggregate Result Columns That Are Not In The GROUP BY Clause**
In most SQL implementations, output columns of an aggregate query may only reference aggregate functions or columns named in the GROUP BY clause. It does not make good sense to reference an ordinary column in an aggregate query because each output row might be composed from two or more rows in the input table(s).

Now we will learn the SQL language!

תרגול בביתה

- | | |
|---|-----|
| להציג את כל הערים | 1. |
| להציג את כל המדינות | 2. |
| להציג את הערים במדינה שהקוד שלה NLD | 3. |
| הצג את המדינות שהקוד שלהם הוא LBR או IOT או TKL | 4. |
| להציג את כל הערים עם אוכלוסיה מעל 4 מיליון | 5. |
| להציג את כל הערים עם אוכלוסיה מעל 3 מיליון וגם בברזיל (קוד BRA) | 6. |
| להציג את כל הערים עם אוכלוסיה בין 150,000 ל 170,000 | 7. |
| להציג את המדינות ששנת עצמאות 1970 או 1980 או 1990 | 8. |
| להציג את המדינות ששנת עצמאות 1980 וגם 1990 | 9. |
| להציג את המדינות ששנת עצמאות בין 1980 ל 1990 | 10. |
| להציג את המדינות באפריקה וגם שנת העצמאות 1964 | 11. |
| להציג את המדינות באפריקה או שנת העצמאות 1964 | 12. |
| להציג את המדינות שהם באסיה | 13. |
| להציג את המדינות שהם לא באסיה | 14. |
| להציג את המדינות שהם לא באסיה ולא באירופה | 15. |
| להציג את הערים שמתחילים באות H | 16. |
| להציג את הערים שאין בהם את האות e | 17. |