ScopeTreasure Courses

SQL COURSES · LESSON DB.1 · CODE SCOPE SERIES

Database Kya Hai? + SQL Ka Taaruf

What a database is and why apps need one: tables, rows and columns, your first SELECT — and the SQL vs NoSQL comparison, with when each one wins.

Introduction

A school keeps its students in a paper register. To find one student's marks, someone flips pages for ten minutes. To find **every student above 90 marks**, someone reads the whole register, page by page. Now imagine the same school with 50,000 students — the register idea dies. This is exactly why **databases** exist: a database is data stored in an organised way on a computer, so questions get answered in less than a second.

The language we use to ask those questions is **SQL** (Structured Query Language). In this first lesson you will meet the three words every database lesson uses — **table, row, column** — run your first real query, and settle the famous question: **SQL or NoSQL — which one, and when?**

(Roman Urdu: Register mein dhoondhna sust hai; database wohi register hai jo computer par hai aur pal bhar mein jawab deta hai.)

Key points

- **A database is organised data, stored safely:** names, marks, products, orders — kept in one place, searchable in milliseconds, and still there when the app restarts. Files in a folder cannot do this job well; a database can.
  (Roman Urdu: Database data ko mehfooz aur tartib se rakhta hai — app band ho jaye, data waisa ka waisa.)
- **A table is one topic:** the `students` table holds students, the `products` table holds products. One table, one topic — never mix two topics in one table.
- **A row is one record:** one row = one student. **A column is one fact type:** the `marks` column holds only marks, for every row. Rows go across, columns go down — like a register page, but strict.
- **SQL is a language, not a database:** SQL is how you *talk*. The software that stores the data and listens is called a **DBMS** — MySQL, PostgreSQL and SQLite are famous ones. Same SQL language, slightly different accents.
- **SQL vs NoSQL — the shape:** SQL databases store data in **fixed-shape tables** (every row has the same columns). NoSQL databases (like MongoDB) store **flexible documents** — one record can have fields another record does not.
- **SQL vs NoSQL — relations:** SQL is built for **connected data** (a student linked to his fees, an order linked to its items) and answers multi-table questions with JOINs. NoSQL prefers each record to carry everything it needs inside itself.
- **SQL vs NoSQL — when each wins:** choose **SQL** when the shape is clear and the data is connected — school records, shops, banks, bookings. Choose **NoSQL** when the shape keeps changing or is unknown

— social feeds, product catalogues with different features per item, logs. Most real apps start with SQL, and many use SQL for the core data even when NoSQL helps at the edges.

(Roman Urdu: Saaf shape aur judaa hua data = SQL; badalti shape = NoSQL. Yeh muqabla har database course ki bunyad hai.)

- **Your first SQL word is SELECT:** `SELECT` means "show me". Almost every question you will ever ask a database starts with it.

Code examples

Example 1 — Your first query: show me everything

```
SELECT * FROM students;
```

What it does: `SELECT *` means "show all columns", and `FROM students` names the table. This one line asks the database for the complete register page.

What you will see:

```
+----+-------------+--------+-------+----------+
| id | name        | class  | marks | city     |
+----+-------------+--------+-------+----------+
|  1 | Ali Raza    | 10-A   |    88 | Sahiwal  |
|  2 | Sara Khan   | 10-A   |    92 | Lahore   |
|  3 | Bilal Ahmed | 10-B   |    76 | Multan   |
|  4 | Hira Shah   | 10-B   |    95 | Sahiwal  |
|  5 | Usman Tariq | 9-A    |    64 | Karachi  |
+----+-------------+--------+-------+----------+
5 rows in set
```

Five students, five rows — and the answer came instantly. This same `students` table will travel with us through the whole series, so you can check every later result by hand.

Example 2 — Ask a real question: the top students

```
SELECT name, marks FROM students
WHERE marks >= 90;
```

What it does: Now we choose only two columns — `name` and `marks` — and add a condition with `WHERE`: only rows where marks are 90 or more. Finding these names in a paper register would take a full page-by-page read; here it is one line.

What you will see:

```
+-----------+-------+
| name      | marks |
+-----------+-------+
| Sara Khan |    92 |
| Hira Shah |    95 |
+-----------+-------+
2 rows in set
```

Only Sara and Hira pass the 90 mark. Notice the result is itself a small table — every SQL answer comes back as rows and columns, even when the answer is a single number.

Example 3 — The same student, the NoSQL way

```
-- SQL shape: one row, fixed columns (from Example 1)
-- id = 1, name = 'Ali Raza', class = '10-A', marks = 88, city = 'Sahiwal'
```

```
NoSQL shape: one document, flexible fields
{
  "name": "Ali Raza",
  "class": "10-A",
  "marks": 88,
  "city": "Sahiwal",
  "hobby": "cricket"
}
```

What it does: This is not a query — it is the comparison made visible. In the SQL row, every student has exactly the same five columns; Ali cannot have a `hobby` unless the whole table gets that column. In the NoSQL document, Ali's record carries a `hobby` field that Sara's record may not have at all. Neither shape is "better" — they answer different needs, which is why the key points above give you the *when*, not just the *what*.

What you will see: nothing runs here — and that is the lesson. SQL thinks in strict tables; NoSQL thinks in flexible documents. Code Scope teaches SQL first (this series) and MongoDB next; by the end of both, you will choose between them like a professional, not by fashion.

Quick recap

- A **database** stores organised data so questions are answered in milliseconds, not page-flips.
- **Table** = one topic, **row** = one record, **column** = one fact type.
- **SQL** is the language; **MySQL / PostgreSQL / SQLite** are the DBMS software that listens.
- **SQL vs NoSQL:** fixed tables + strong relations (SQL) vs flexible documents (NoSQL). Connected, clear-shaped data → SQL; changing shapes → NoSQL.
- `SELECT ... FROM ... WHERE ...` is the shape of almost every question.

Common mistakes

- **Thinking SQL is a database:** SQL is the language you speak; the database software is MySQL, PostgreSQL or SQLite. Mixing these up makes every later lesson confusing.
  (Roman Urdu: SQL zubaan hai, software nahi — zubaan se baat hoti hai, software sunta hai.)
- **Putting two topics in one table:** students and their fees in a single wide table creates copies of the same fact in many rows. One table, one topic — JOINs (DB.5) reconnect them when you ask.
- **Choosing NoSQL "because it is modern":** if your data has a clear shape and real relations — marks, fees, orders — SQL is the safer, simpler home. Choose by the data's shape, not by trend.
- **Forgetting the semicolon:** a query ends with `;`. In most tools the query simply does not run without it, and beginners lose minutes hunting for a "big" error that is one missing character.

Practice

1. Write the query that shows only the `name` and `city` of every student. (Two columns, no condition — use Example 2 as your model, minus the WHERE line.)
2. Write the query that shows the full record of students who live in Sahiwal. Check by hand against Example 1: you should get exactly 2 rows — Ali Raza and Hira Shah.
3. Decide SQL or NoSQL, and say why in one line each: (a) a bank's accounts and transfers, (b) a clothing shop whose every product has different extra details (size chart, fabric, care notes in free text), (c) a school result system where every student must link to subjects and fees.

Practice quiz

1. In a database, what is a table, a row, and a column?

- Table = one record, row = one topic, column = one database
- Table = one topic, row = one record, column = one fact type
- Table = one fact, row = one topic, column = one record
- They are three names for the same thing

Answer: Table = one topic, row = one record, column = one fact type

2. Is SQL itself a database?

- Yes, SQL is the database software
- No, SQL is the language you talk; the DBMS software that stores data and listens is MySQL, PostgreSQL or SQLite
- Yes, but only on Windows
- No, SQL is a programming language for websites

Answer: No, SQL is the language you talk; the DBMS software that stores data and listens is MySQL, PostgreSQL or SQLite

3. When do you choose SQL over NoSQL?

- When the shape keeps changing every week
- When the shape is clear and the data is connected, like school records, shops, banks and bookings; changing shapes point to NoSQL
- When you want flexible documents
- Always, NoSQL is never the right choice

Answer: When the shape is clear and the data is connected, like school records, shops, banks and bookings; changing shapes point to NoSQL

Watch the Code Scope channel: https://www.youtube.com/@CodeScopePK — this lesson's own video joins the channel when it is recorded.

Read online: https://scopetreasure.com/notes/sql-database-intro · More free tools and notes at https://scopetreasure.com