An overview of SQL, Relational Database Management Systems, and object-relational mapping

(as much as fits in an hour)

Andrew Walker 14/5/2026

Structured

Query

Language

a tool to extract data from relational databases

SQL

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	4	2001	5

Staff table

SQL tables are a lot like data frames in Pandas or R: each column has a name and fixed type, each row represents a “thing”, the number of rows can increase or decrease and there can be a kind of index column so rows are unique.

relational databases

SQL

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	4	2001	5

Staff table

SELECT name, grade FROM staff;

Bob	5
Sally	8
Jim	4

relational databases

SQL

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	4	2001	5

Staff table

```
SELECT name, department FROM staff WHERE grade < 7;
```

Bob	5
Jim	5

relational databases

SQL

Primary Key

Foreign Key

Department number	Name	Department head
5	IT	2001
1	HR	1002
3	Stuff	1202

Department table

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	4	2001	5

Staff table

Relational databases can have multiple tables. These have names and can be related via keys

relational databases

SQL

Department number	Name	Department head
5	IT	2001
1	HR	1002
3	Stuff	1202

Department table

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	4	2001	5

Staff table

```
SELECT staff.name, department.name
FROM staff JOIN department
ON staff.department = department.department_number
WHERE department.name = “IT” AND staff.grade < 7;
```

Bob	IT
Jim	IT

relational databases

Examples and tutorial

Greg Wilson’s material: https://third-bit.com/sql/

Online install instructions:
https://compearthsci.github.io/meetings/2026/05/11/sql.html

SQL

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	4	2001	5

Staff table

UPDATE staff SET grade = 5 WHERE name = “Jim”;

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	5	2001	5

Staff table

relational databases

SQL

Staff number	Name	Grade	Supervisor	Department
1000	Bob	5	2001	5
2001	Sally	8	1002	5
2010	Jim	5	2001	5
4999	Tim	4	2001	5
5000	Liz	5	2001	5

Staff table

```
INSERT INTO staff (staff_number, name, grade, supervisor, department)
    VALUES
    (4999, ‘Tim’, 4, 2001, 5),
    (5000, ‘Liz’, 5, 2001, 5);
```

relational databases

SQL is declarative:

We specify what to do, not how to do it

Databases have schemas:

What is permitted in each table.
What relations are enforced.

SQL transactional:

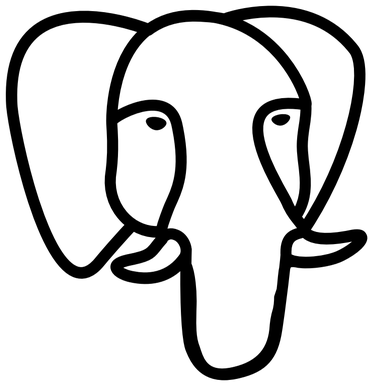
Queries that change things happen in one go.
Queries that read things happen before or after each transaction.

Examples and tutorial

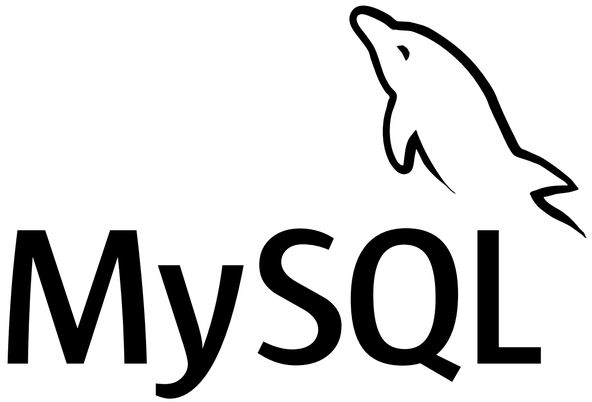
Greg Wilson’s material: https://third-bit.com/sql/

Online install instructions:
https://compearthsci.github.io/meetings/2026/05/11/sql.html

Database Management Systems (DBMS)

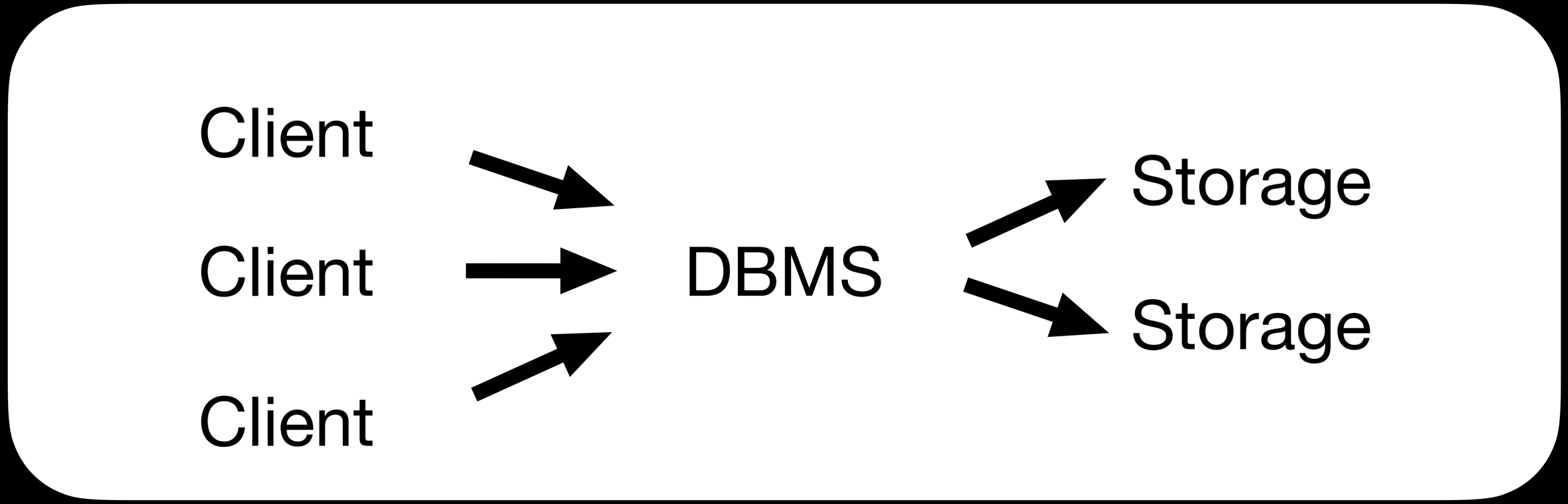


PostgreSQL



SQLite

Database Management Systems (DBMS)



DBMS clients

Command line (e.g. sqlite3) - a REPL where you connect to a database and type SQL commands

GUI / web based (e.g. marino notebook) - maybe you still type SQL, maybe it’s all point and click

Library in programming language (e.g. sqlite3 or dbm in the python standard library) - SQL in strings, results in some kind of iterable

An object relational mapper (e.g. SQLAlchemy) - defines relationships between tables and objects in a language (so database used to persist information)

DBMS are:

Atomicity **C**onsistency **I**solation **D**urability

Update is done or not done (never half done)

Schema rules are enforced

Multiple updates don’t break things (appear to be sequential)

Once committed, a change is persistent

When to use SQL?

When to use SQL?

Multiple people updating data (maybe using different tools)

“Easily” scale things like we still think Web2.0 is cool

Lots of people reading / searching data