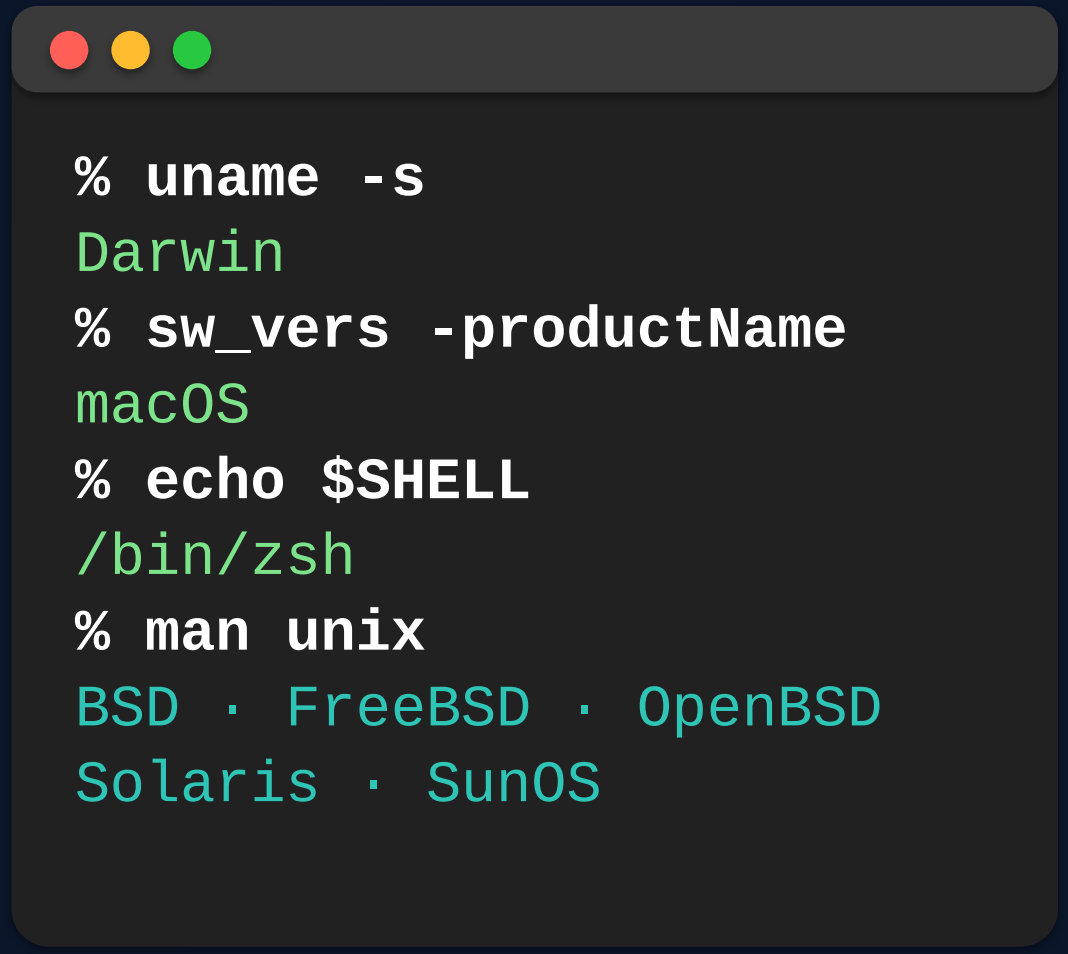


LESSON 7 OF 12

macOS UNIX

Darwin & BSD

Databases on macOS: SQLite, PostgreSQL & MySQL

Full Course · macOS 27 Golden Gate

What You Will Learn in Lesson 7

- Embedded vs client-server databases
- SQLite: built into macOS, used by macOS itself
- SQL essentials: create, insert, select, update, join
- PostgreSQL with Homebrew: install, users, databases, psql
- MySQL with Homebrew: install, secure, users, the mysql client
- Services, ports, backups, restores and GUI clients
- Databases on FreeBSD, OpenBSD and Solaris

Two Kinds of Databases

	Embedded (SQLite)	Client-server (PostgreSQL, MySQL)
Runs as	A library inside your program	A separate server process (daemon)
Data lives in	One file, e.g. school.db	A data directory managed by the server
Users & network	No users; file permissions only	Users, passwords, roles; TCP port 5432 / 3306
Best for	Apps, tools, small sites, learning	Web apps, many users at once, big data
On macOS	Built in: /usr/bin/sqlite3	Install with Homebrew

SQLite: Already on Your Mac

- sqlite3 ships with macOS in /usr/bin
- Many macOS apps store data in SQLite files
- A database is a single, portable file: copy it, email it, back it up
- Same file format on macOS, Linux, BSD, Windows, iPhone, Android
- Python includes it too: import sqlite3
- Never edit a live app's database; work on a copy

SQLite Hands-On

```
% sqlite3 ~/lab/school.db
sqlite> CREATE TABLE students (id INTEGER PRIMARY KEY, name TEXT, city TEXT);
sqlite> INSERT INTO students (name, city) VALUES ('Maryam','Kabul'), ('Ahmad','Herat');
sqlite> .headers on
sqlite> .mode column
sqlite> SELECT * FROM students;
id  name    city
1   Maryam  Kabul
2   Ahmad   Herat
sqlite> .tables      .schema students      .quit
```

SQL Essentials

```
CREATE TABLE courses (id INTEGER PRIMARY KEY, title TEXT);
CREATE TABLE enroll  (student_id INT, course_id INT);
INSERT INTO courses (title) VALUES ('macOS UNIX'), ('Python');
INSERT INTO enroll VALUES (1,1), (1,2), (2,1);

SELECT name, city FROM students WHERE city = 'Kabul' ORDER BY name;
UPDATE students SET city = 'Mazar' WHERE id = 2;
DELETE FROM enroll WHERE student_id = 2 AND course_id = 1;

SELECT s.name, c.title FROM students s
  JOIN enroll e  ON e.student_id = s.id
  JOIN courses c ON c.id = e.course_id;
```

PostgreSQL With Homebrew

```
% brew search postgresql           # choose a current version
% brew install postgresql@17
% brew services start postgresql@17 # starts now and at login
% echo 'export PATH="/opt/homebrew/opt/postgresql@17/bin:$PATH"' >> ~/.zprofile
% source ~/.zprofile

% psql postgres                    # your macOS user is the superuser
postgres=# SELECT version();
postgres=# \q
```

Version 17 is an example. Read the caveats brew prints after installing.

PostgreSQL: Users and Databases

```
% createuser --pwprompt webapp        # a login role with a password
% createdb --owner=webapp school
% psql -U webapp -d school -h localhost
school=> CREATE TABLE students (id SERIAL PRIMARY KEY, name TEXT);
school=> \dt                            -- list tables
school=> \l                             -- list databases
school=> \du                            -- list roles (users)
school=> \conninfo                      -- who and where am I?
school=> \q
```

MySQL With Homebrew

```
% brew install mysql                    # or: brew install mariadb (pick one)
% brew services start mysql
% mysql_secure_installation             # set root password, remove test users
% mysql -u root -p
mysql> SELECT VERSION();
mysql> CREATE DATABASE school;
mysql> CREATE USER 'webapp'@'localhost' IDENTIFIED BY 'Strong#Pass1';
mysql> GRANT ALL PRIVILEGES ON school.* TO 'webapp'@'localhost';
mysql> SHOW DATABASES;   exit
```

Services, Ports and Data Folders

```
% brew services list
Name           Status  User   File
mysql          started wahid  ~/Library/LaunchAgents/homebrew.mxcl.mysql.plist
postgresql@17  started wahid  ~/Library/LaunchAgents/homebrew.mxcl.postgresql@17.plist

% lsof -iTCP -sTCP:LISTEN -n -P | grep -E 'postgres|mysqld'
postgres ... TCP 127.0.0.1:5432 (LISTEN)
mysqld   ... TCP 127.0.0.1:3306 (LISTEN)

% ls /opt/homebrew/var            # data: postgresql@17/, mysql/
```

Example output.

Backup and Restore

```
# SQLite
% sqlite3 ~/lab/school.db .dump > school-sqlite.sql
% sqlite3 restored.db < school-sqlite.sql

# PostgreSQL
% pg_dump -U webapp -h localhost school > school-pg.sql
% createdb school_test && psql -d school_test -f school-pg.sql

# MySQL
% mysqldump -u root -p --databases school > school-my.sql
% mysql -u root -p < school-my.sql
```

Graphical Database Tools

Tool	Databases	Install
DB Browser for SQLite	SQLite	brew install --cask db-browser-for-sqlite
pgAdmin 4	PostgreSQL	brew install --cask pgadmin4
MySQL Workbench	MySQL	brew install --cask mysqlworkbench
DBeaver Community	All of them (and more)	brew install --cask dbeaver-community

Database Security Checklist

- Listen on localhost only unless other machines truly need access
- Run mysql_secure_installation; never leave root without a password
- One user per application, with rights to its own database only
- Strong passwords; never put them in scripts you share (use ~/.pgpass, ~/.my.cnf with chmod 600)
- Automated, tested backups — copied off the Mac
- Update regularly: brew upgrade, and read the release notes

Databases on Other UNIX Systems

Step	FreeBSD	OpenBSD	Solaris 11.4
Install	pkg install postgresql17-server	pkg_add postgresql-server	pkg install a database/postgres or mysql package
Initialize	service postgresql initdb	su _postgresql -c 'initdb ...'	per package instructions
Enable at boot	sysrc postgresql_enable=YES	rcctl enable postgresql	svcadm enable <service>
Start / status	service postgresql start / status	rcctl start / check postgresql	svcs <service>

Package names and versions change: use pkg search / pkg_info -Q / pkg search first.

Lab: A Student Database Three Ways

```
% mkdir -p ~/lab && sqlite3 ~/lab/school.db < students.sql
% sqlite3 -header -column ~/lab/school.db 'SELECT * FROM students;'

% brew install postgresql@17 && brew services start postgresql@17
% createuser --pwprompt webapp && createdb --owner=webapp school
% psql -U webapp -h localhost school -c 'CREATE TABLE t(x int); \dt'
% pg_dump -U webapp -h localhost school > ~/lab/school-pg.sql

% lsof -iTCP -sTCP:LISTEN -n -P;   brew services list
```

students.sql = the CREATE and INSERT lines from the SQLite slides.

Summary & Homework

- SQLite = one file, built in; PostgreSQL and MySQL = servers on 5432 / 3306
- SQL: CREATE, INSERT, SELECT … WHERE, UPDATE, DELETE, JOIN
- brew install + brew services start; one user per application
- Backups: .dump, pg_dump, mysqldump — and test the restore
- Homework: do the lab; add a courses table and write a JOIN
- Next: Lesson 8 — Users, Groups, sudo & Directory Services

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

Slides for this lesson: MyWebUniversity.com/presentations

YouTube: @MyWebUniversity · @MyWebUniversity-Dari

Wahidullah Lutfy — Founder and Director of IT of MyWebUniversity, and CS teacher

Book: The Direct Path to Linux Ubuntu (Amazon · Barnes & Noble)

Learn it, then pass it on.