Migration Report: CSV/JSON Storage to SQLite

Project: App-sensibilisation-video

November 7, 2025

Contents

1 Purpose

This living document records each step of migrating the project from ad-hoc CSV (`data.csv`) and JSON (`data.json`) storage to a structured SQLite database using the `better-sqlite3` Node.js library. It will be updated after every migration task (schema changes, data import, service refactors, etc.).

2 Baseline State (Pre-Migration)

- User session data appended to `data.csv` with 17 columns (user choices, resolutions, metadata).
- Aggregated score/time data persisted in `data.json` (nested objects for scores, times, precisions).
- No database; all queries implied linear scans / in-memory aggregation.

3 Step 1: Environment Setup (Completed)

3.1 Objective

Add a performant, synchronous SQLite driver to project dependencies.

3.2 Actions

1.1 Installed dependency: `better-sqlite3@î1.10.0`.
Added to `package.json` under `dependencies`.

1.2 Updated `package-lock.json` (36 packages added transitively).[1]

1.3 No other scripts modified at this step.

3.3 Result

Project is now capable of creating and interacting with a local SQLite database file. No runtime code yet consumes the DB.

4 Step 2: Database Initialization (Completed)

4.1 Objective

Introduce a reproducible schema creation script and persistent database file.

4.2 Artifacts Created

- Directory: `db/`
- Script: `db/init-database.js`
- Database file (after execution): `db/database.db`
- Added npm script: `d̈b:init:̈  n̈ode db/init-database.js¨`

4.3 Schema Defined

Two tables established:

[1] As reported by npm: "added 36 packages, audited 115 packages, 0 vulnerabilities".

sessions Mirrors original CSV structure; adds surrogate primary key for internal references.

```
CREATE TABLE IF NOT EXISTS sessions (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  user TEXT NOT NULL,
  category1 TEXT,
  videoName1 TEXT,
  videoPath1 TEXT,
  resolution1 TEXT,
  category2 TEXT,
  videoName2 TEXT,
  videoPath2 TEXT,
  resolution2 TEXT,
  QO1 TEXT,
  QO2 TEXT,
  QO3 TEXT,
  QO4 TEXT,
  QO5 TEXT,
  comments TEXT,
  screenType TEXT,
  timestamp TEXT NOT NULL
);
CREATE INDEX IF NOT EXISTS idx_sessions_user ON sessions(user);
CREATE INDEX IF NOT EXISTS idx_sessions_timestamp ON sessions(
    timestamp);
```

users Stores aggregate metrics previously in JSON.

```
CREATE TABLE IF NOT EXISTS users (
  pseudo TEXT PRIMARY KEY,
  totalScore INTEGER DEFAULT 0,
  totalTime INTEGER DEFAULT 0,
  sessionCount INTEGER DEFAULT 0
);
```

4.4 Pragmas

Enabled Write-Ahead Logging and foreign key enforcement:

```
PRAGMA journal_mode = WAL;
PRAGMA foreign_keys = ON;
```

4.5 Verification

Execution output confirmed creation of tables: `sessions`, `users`, plus `sqlite_sequence` (auto-increment bookkeeping).

5 Change Log

Step 1
Added dependency `better-sqlite3`. (No code paths yet updated.)

Step 2

Created schema initialization script; introduced `sessions` and `users` tables; added npm script `db:init`.

6 How to Reproduce Current DB State

Run:

```
npm install
npm run db:init
```

This will (re)create the database with the current schema.

This document will be updated after each subsequent migration task.