Database Project

Agile Sprint Planning Guide

6 Sprints to Success

Sprint 0: Idea & Planning
Sprint 1: Database Design
Sprint 2: Frontend Architecture
Sprint 3: Build & Populate
Sprint 4: Queries & Features
Sprint 5: Testing & Submission

What is Agile?

Agile breaks big projects into smaller chunks called "sprints".
Each sprint has a clear goal and deliverables.

At the end of each sprint, you have something working to show.
This helps you track progress and catch problems early.

How This Works

You'll work through 6 sprints. Each sprint builds on the previous one.

Sprint	Focus	Key Output
0	Idea & Planning	Business idea document
1	Database Design	ERD + table specifications
2	Frontend Architecture	UI wireframes + tech choices
3	Build & Populate	Working database with data
4	Queries & Features	All SQL queries working
5	Testing & Submission	Final tested submission

Sprint Ceremonies

Each sprint follows this pattern:

Ceremony	When	What Happens
Sprint Planning	Start of sprint	Review tasks, set goals for the sprint
Daily Stand-up	Each work session	Quick check: What did I do? What's next? Any blockers?
Sprint Review	End of sprint	Demo what you built, get feedback
Retrospective	End of sprint	What went well? What could improve?

Working Solo?

Even working alone, these ceremonies help!

- Planning: Write down your sprint goals
- Stand-up: Quick self-check at start of each session
- Review: Test your work, show someone if possible
- Retro: Note what slowed you down for next time

SPRINT 0

Idea Generation & Documentation

Recommended: 1 session

Sprint Goal

Choose your business idea and document what your system needs to do.

Tasks

✓	Task	Status
☐	Brainstorm 3-5 possible business ideas	
☐	Choose ONE idea to develop	
☐	Write a brief description of the business (2-3 sentences)	
☐	List the main things your business does (sells products, takes bookings, etc.)	
☐	Identify who uses the system (customers, staff, admin?)	
☐	List 5-10 questions your database should answer	
☐	Document any assumptions you're making	

📦 Sprint Deliverables

- ☐ Business Idea Document (1 page)
- ☐ List of business questions the database must answer
- ☐ Initial thoughts on what data you'll need to store

Questions to Answer

- What does my business sell or provide?
- Who are the customers?
- What information do I need about each customer?
- What information do I need about each product/service?
- How do orders/bookings work?
- What reports would the business owner want to see?

SPRINT 1

Database Design

Recommended: 1-2 sessions

Sprint Goal

Design your database structure before writing any code.

Tasks

✓	Task	Status
☐	Identify all entities (things you need to store data about)	
☐	List attributes (columns) for each entity	
☐	Identify primary keys for each table	
☐	Identify relationships between tables	
☐	Identify foreign keys needed	
☐	Draw an Entity Relationship Diagram (ERD)	
☐	Write out table specifications (name, columns, types)	
☐	Review: Does your design support all your business questions?	

📦 Sprint Deliverables

- ☐ Entity Relationship Diagram (ERD)
- ☐ Table specification document
- ☐ List of relationships with foreign keys identified

ERD Tips

- Use boxes for tables, lines for relationships
- Label relationships (one-to-many, many-to-many)
- Mark primary keys (PK) and foreign keys (FK)
- Keep it simple - 4 tables is enough!

Tools: Draw on paper, use draw.io, or Lucidchart

Table Specification Template

Document each table like this:

Column Name	Data Type	Constraints	Notes
customer_id	INTEGER	PK, AUTO	Primary key
first_name	TEXT	NOT NULL	
email	TEXT	NOT NULL	Must be unique
created_date	TEXT		Date registered

Create a table like this for each of your 4 tables.

Relationship Documentation

Document how your tables connect:

From Table	To Table	Relationship	Foreign Key
customers	orders	One to Many	orders.customer_id
orders	order_items	One to Many	order_items.order_id
products	order_items	One to Many	order_items.product_id

Don't write any SQL yet! Design first, code later.

SPRINT 2

Frontend Architecture

Recommended: 1-2 sessions

Sprint Goal

Plan your user interface and choose your technology stack.

Tasks

✓	Task	Status
☐	Choose your frontend technology (see options below)	
☐	Sketch wireframes for main screens	
☐	Plan user flow: How will users navigate?	
☐	Decide what features each screen will have	
☐	Plan how frontend will connect to database	
☐	Set up your development environment	
☐	Create project folder structure	

📦 Sprint Deliverables

- ☐ Technology choice documented with reasons
- ☐ Wireframe sketches (hand-drawn is fine!)
- ☐ User flow diagram
- ☐ Development environment ready

Frontend Options

Option	Difficulty	Best For
Command Line (no UI)	Easiest	Focus purely on SQL skills
Python + Tkinter	Easy	Simple desktop GUI
HTML + JavaScript	Medium	Web-based interface
Python + Flask	Medium	Web app with backend
React / Vue	Harder	Modern web app experience

DB Browser for SQLite	Easiest	Visual database tool (no coding)

Keep It Simple!

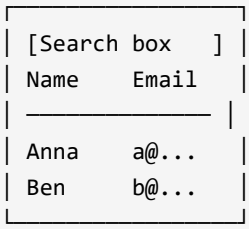
The frontend is NOT the main focus of this project.
A simple interface that works is better than a fancy one that doesn't.

If in doubt, use DB Browser for SQLite - it lets you run
queries and see results without building any UI.

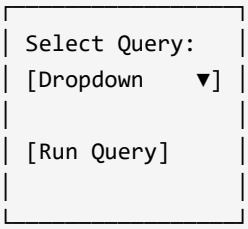
Wireframe Examples

Sketch simple screens showing what the user will see:

Screen: Customer List

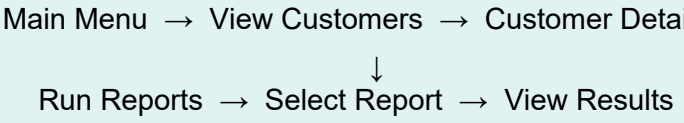


Screen: Run Query



User Flow Diagram

Show how users move between screens:



SPRINT 3

Build & Populate

Recommended: 2-3 sessions

Sprint Goal

Create your database tables and fill them with test data.

Tasks

✓	Task	Status
☐	Write CREATE TABLE statement for customers	
☐	Write CREATE TABLE statement for products	
☐	Write CREATE TABLE statement for orders	
☐	Write CREATE TABLE statement for order_items	
☐	Run all CREATE statements - fix any errors	
☐	Plan your test data on paper first	
☐	Write INSERT statements for customers (include some with no orders)	
☐	Write INSERT statements for products (include some never ordered)	
☐	Write INSERT statements for orders (vary dates and statuses)	
☐	Write INSERT statements for order_items	
☐	Run all INSERT statements - screenshot affected rows	
☐	Verify data with simple SELECT * queries	

📦 Sprint Deliverables

- ☐ SQL script: create_tables.sql
- ☐ SQL script: insert_data.sql
- ☐ Screenshots showing successful data insertion
- ☐ Screenshots of SELECT * for each table

Data Planning Tip

Before writing INSERTs, sketch your data on paper:

Customers: Anna (will have orders), Ben (will have orders),

Chloe (NO orders), Dave (NO orders)...

This helps ensure you have the right test scenarios.

Sprint 3: Verification Checklist

Before moving on, verify your database is set up correctly:

Check	How to Test	Expected Result
Tables exist	SELECT name FROM sqlite_master WHERE type='table';	See all 4 table names
Primary keys work	Try inserting duplicate ID	Should get error
Foreign keys set up	Check table structure	FK columns present
Data inserted	SELECT COUNT(*) FROM each table	Expected row counts
Relationships work	Simple JOIN query	Returns combined data

Quick Test Query

Run this to verify your relationships work:

```
-- Test: Can I join customers to orders?
SELECT c.first_name, o.order_id
FROM customers c
INNER JOIN orders o ON c.customer_id = o.customer_id
LIMIT 5;
```

If this works, your database structure is correct!

SPRINT 4

Queries & Features

Recommended: 2-3 sessions

Sprint Goal

Write all required SQL queries and implement key features.

Tasks - Basic Queries

✓	Task	Status
☐	Write query: Count records	
☐	Write query: Filter by numeric value	
☐	Write query: Filter by text/status	
☐	Write query: Sort results	
☐	Write query: Filter by date	

Tasks - JOIN Queries

✓	Task	Status
☐	Write query: INNER JOIN two tables	
☐	Write query: INNER JOIN three tables	
☐	Write query: LEFT JOIN with GROUP BY and COUNT	
☐	Write query: Find records with no match (LEFT JOIN + IS NULL)	
☐	Write query: Another "find missing" query	

Tasks - Aggregation

✓	Task	Status
☐	Write query: SUM with GROUP BY	
☐	Write query: Ranking (ORDER BY aggregate DESC)	
☐	Write query: Group by date (month or year)	

📦 Sprint Deliverables

☐ SQL script: queries.sql (all queries with comments)

- ☐ Screenshot of each query result
- ☐ Brief explanation of what each query does

Sprint 4: Data Maintenance

Tasks - UPDATE, DELETE, VIEW

✓	Task	Status
☐	Write and test an UPDATE statement	
☐	Screenshot: Before and after the UPDATE	
☐	Write and test a DELETE statement	
☐	Screenshot: Before and after the DELETE	
☐	Design a useful VIEW for your business	
☐	Write CREATE VIEW statement	
☐	Test the VIEW with SELECT	
☐	Screenshot: VIEW in action	

📦 Sprint Deliverables

- ☐ SQL script: maintenance.sql
- ☐ Before/after screenshots for UPDATE
- ☐ Before/after screenshots for DELETE
- ☐ VIEW creation and usage screenshots

Testing Tip: Use Transactions

When testing UPDATE and DELETE, you can use transactions to safely test without permanently changing data:

```
BEGIN TRANSACTION;
DELETE FROM customers WHERE ...;
SELECT * FROM customers;  -- Check the result
ROLLBACK;  -- Undo the delete
```

This lets you test destructive operations safely!

SPRINT 5

Testing & Submission

Recommended: 1-2 sessions

Sprint Goal

Test everything, fix issues, and prepare final submission.

Tasks - Testing

✓	Task	Status
☐	Run ALL queries again from scratch	
☐	Verify all screenshots are clear and readable	
☐	Check all SQL scripts run without errors	
☐	Test on a fresh database (delete and recreate)	
☐	Have someone else try your queries if possible	
☐	Fix any bugs or issues found	

Tasks - Documentation

✓	Task	Status
☐	Write/finalise business idea description	
☐	Complete table relationship explanations	
☐	Add comments to all SQL scripts	
☐	Write brief explanation for each query	
☐	Answer any required written questions	
☐	Proofread all documentation	

Tasks - Submission

✓	Task	Status
☐	Organise files into clear folder structure	
☐	Check all required files are included	
☐	Name files clearly	
☐	Final check against submission checklist	
☐	Submit before deadline!	

📦 Sprint Deliverables

- ☐ Final tested SQL scripts
- ☐ All screenshots organised
- ☐ Written documentation complete
- ☐ Everything submitted on time

Final Submission Checklist

Before you submit, make sure you have EVERYTHING:

Category	Item	✓
Scripts	CREATE TABLE statements	☐
	INSERT statements	☐
	All SELECT queries	☐
	UPDATE statement(s)	☐
	DELETE statement(s)	☐
	CREATE VIEW statement(s)	☐
Screenshots	Data insertion evidence	☐
	Each query result	☐
	UPDATE before/after	☐
	DELETE before/after	☐
	VIEW working	☐
Documentation	Business idea description	☐
	ERD diagram	☐
	Table specifications	☐
	Query explanations	☐
	Written questions answered	☐

Folder Structure Suggestion

```
📁 YourName_DatabaseProject/
   📁 sql/
      📄 01_create_tables.sql
      📄 02_insert_data.sql
      📄 03_queries.sql
      📄 04_maintenance.sql
   📁 screenshots/
      📷 (all your screenshots)
   📁 docs/
      📄 business_idea.docx
      📷 erd_diagram.png
   📁 frontend/ (if applicable)
```

Sprint Retrospective Template

Fill this in at the end of each sprint to track your learning:

✓ What went well?

_____

_____

_____

⚠ What was challenging?

_____

_____

_____

💡 What will I do differently next sprint?

_____

_____

_____

Good luck with your project! Work sprint by sprint and you'll get there.