

# Pawn to King: A Guided Exercise Tutorial

## 1. Introduction

Exercise 1.1: In your own words, explain why combining Python and SQLite can help you learn chess while coding. Write down the objectives of this project: listing the features you will implement.

*Hint: Think of Python as the engine, and SQLite as the memory that records each game and move.*

## 2. Setting up the project

Exercise 2.1: Ensure Python 3.x is installed. Hint: Check the version via your terminal command 'python --version'.

Exercise 2.2: Create a new folder named 'pawn\_to\_king'. Inside, create two empty files: one for the database schema and one for your main script. Hint: Use file explorer or 'mkdir' and 'touch'.

## 3. Designing the database schema

Exercise 3.1: Sketch on paper the tables you need: games, pieces, and moves. For each table, list the columns and their data types.

*Hint: 'games' should record metadata, 'pieces' should record type, color, position, and 'moves' should record each move's details and captures.*

Exercise 3.2: Write down the SQL commands you would use to create each table, using 'CREATE TABLE'. Do not write actual code here, but describe the structure in words.

*Hint: For each column, decide if it needs to be a primary key, foreign key, or default value.*

## 4. Initializing a new game

Exercise 4.1: Describe the steps your Python script would take to connect to the database and execute the schema. Hint: Think of opening a connection and reading the schema file.

Exercise 4.2: Outline how you would insert initial pieces into the database: list the 32 pieces with their starting positions and colors. Hint: Represent rows and columns as numbers between 1 and 8.

## 5. Displaying the board

Exercise 5.1: Imagine how to represent the board as an 8x8 grid in text. Draw an example on paper, using '.' for empty squares and letters for pieces (uppercase for white, lowercase for black).

*Hint: Label columns a-h and rows 1-8. Consider how to map letters to piece types.*

## 6. Validating moves

Exercise 6.1: For the rook piece, write in words the rules that define a valid move. List the steps to check if the path is clear.

*Hint: A rook moves in straight lines; ensure no pieces are between start and end squares.*

Exercise 6.2: Extend your description to outline rules for bishop (diagonal moves) and knight (L-shaped moves).

## 7. Recording moves

Exercise 7.1: Describe how you would log each move into the moves table, including capturing logic. Hint: Think of inserting a row with details and updating or deleting captured piece.

## 8. Putting it all together

Exercise 8.1: Outline a loop your script could use to alternate player turns, display the board, prompt for input, validate and record moves, until a quit command.

*Hint: Consider how to break the loop and display an end-of-game message.*

## 9. Advanced ideas

Exercise 9.1: List three advanced chess rules or features you could implement next (castling, en passant, promotion). For each, give a high-level description of how you'd handle it.

*Hint: Think of additional columns or flags needed in your tables or logic steps in your script.*

**Reflection: What did you learn by designing and outlining this project?**