

SQL Detective Game

Unity Seminar – Project Proposal

Members: Ofek Cofman, Ariel Vaknin

Advisor: Moshe Sulami

Workshop: Unity Seminar

Version 1.0

Introduction

Our project is a 3D educational game designed to teach SQL query-building through story-driven gameplay. The game immerses players in a dynamic virtual environment where they take on the role of an investigator navigating a data-rich world. By solving SQL-based puzzles, players will learn how to construct queries, analyse tables, and retrieve meaningful data to progress in the game.

Purpose Of The Project

In today's world, where the volume of data grows exponentially alongside our reliance on it, the need to effectively manage, retrieve, and analyse information has become a critical challenge. This creates an essential goal: enabling the effective navigation, retrieval, and analysis of data to unlock its full potential, but accessing it requires foundational skills.

Many people interested in entering the world of data lack experience and need an introduction to SQL and data management. Furthermore, teaching children to navigate and retrieve information from large datasets from an early age is essential for their personal development and crucial for society.

Our project addresses these gaps by providing an immersive, story-driven educational experience that teaches SQL query-building through gameplay. By simulating a rich data environment, players will learn to navigate databases, construct queries, and extract insights in a hands-on manner. In addition, It has already been proven that teaching a subject in the form of a fun activity such of a game has a greater impact on remembering the materials and understanding them, which is why this game has an edge over other non-interactive SQL teaching materials.

Target Audience

Our target audience includes a diverse group of people new to SQL who need a platform that immerses them in an authentic environment, allowing them to feel more connected to the data they work with, as well for children, highlighting the importance of introducing data literacy and SQL skills at an early age.

Additionally, the game appeals to casual gamers who enjoy story-driven thinking games, blending entertainment with educational value.

Other Approaches

The market offers few SQL-based educational games, most of which are 2D or text-based, some games with the same concept that can be found on the Internet :

- SQL Murder Mystery (<https://mystery.knightlab.com/>)

* More of an exercise rather than a game, is completely featured in a single web page with explanations while the user must run 2 simple given queries and write one. The Murder Mystery theme is being used solely in naming the tables and nothing else. The exercise teaches some very basic SQL verbs and can be done in 3 minutes.

- Lost at SQL (<https://lost-at-sql.therobinlord.com/>)
- A very nicely designed web, offers a small intro video for the player before entering and choosing the case to investigate. Doesn't allow to save either games or scores, drowns the

user with information when introduced to the game ; there's a lot to read on the site in order to understand the game rules before starting the actual case.

- Squid Game SQL (<https://datalemur.com/sql-game>)

An extremely well-made and friendly game, features a minimum amount of text for the user to read and even has the series' soundtrack when entering the site. In addition, there's also some short texts for the plot for the user to read and pictures from the series to keep things interesting. The game is basically reading and running a sequence of SQL queries while interacting with the web page- there's a story and the user performs queries accordingly while there is not even a 2d game to play with.

- Our game features an interactive interface that combines playing an immersive 3D game while using the mobile phone to write more queries (relatively more complicated ones). It has a plot that makes the player feel more immersed in the game and thus having a more fun SQL-learning experience. Furthermore, our game allows progress and scores savings so a player can come back at a later time to continue from where he left off or just to boast about his high-score among his friends. The player can start playing the game after a short tutorial that introduces the plotline and the mechanics. Another important thing that our game has unlike the upper-mentioned games is the ability to play in one full screen in the browser- what allows better immersion rather than scrolling down a web page and writing queries in some places.

SQL Detective sets itself apart by offering:

- A realistic 3D environment for exploration and immersion.
- Cross-platform gameplay, where players interact with the PC for clues and their phone for query-building.
- A modular framework for replayability and continuous learning.

Main Features

Dynamic Schema Evolution

As players progress, the schema dynamically expands to include new tables and relationships, teaching SQL concepts incrementally. For example, starting with CrimeEvidence and gradually introducing Witnesses, Relationships, and Locations tables allows players to tackle increasingly complex queries.

Interactive Query-Building Interface

The phone interface allows players to construct SQL queries through an intuitive drag-and-drop system. This dual approach supports both beginners and advanced users, while the PC displays the evolving schema and query results.

Cross-Platform Gameplay

The integration between PC and phone ensures seamless gameplay. Players explore the 3D environment and manage clues on the PC while using their phones to interact with the database and solve puzzles.

Replayability and Modular Design

The game's modular framework allows new puzzles to be added easily, ensuring replayability and continuous learning.

Architecture

System Components and Technologies

The system is built using the following components and technologies:

Unity Game Engine

Responsible for creating the 3D environment, gameplay mechanics, and web-based interface. Unity WebGL ensures the game runs in a browser, accessible on both PC and mobile.

C# Programming

Implements core game logic and SQL query validation.

SQLite4Unity3D

Manages the game's SQL backend by integrating SQLite with Unity. Allows local database queries for dynamic schema evolution and gameplay logic.

DB Browser for SQLite

Used during development to design and structure the game's databases. Ensures schemas are well-designed to support SQL learning.

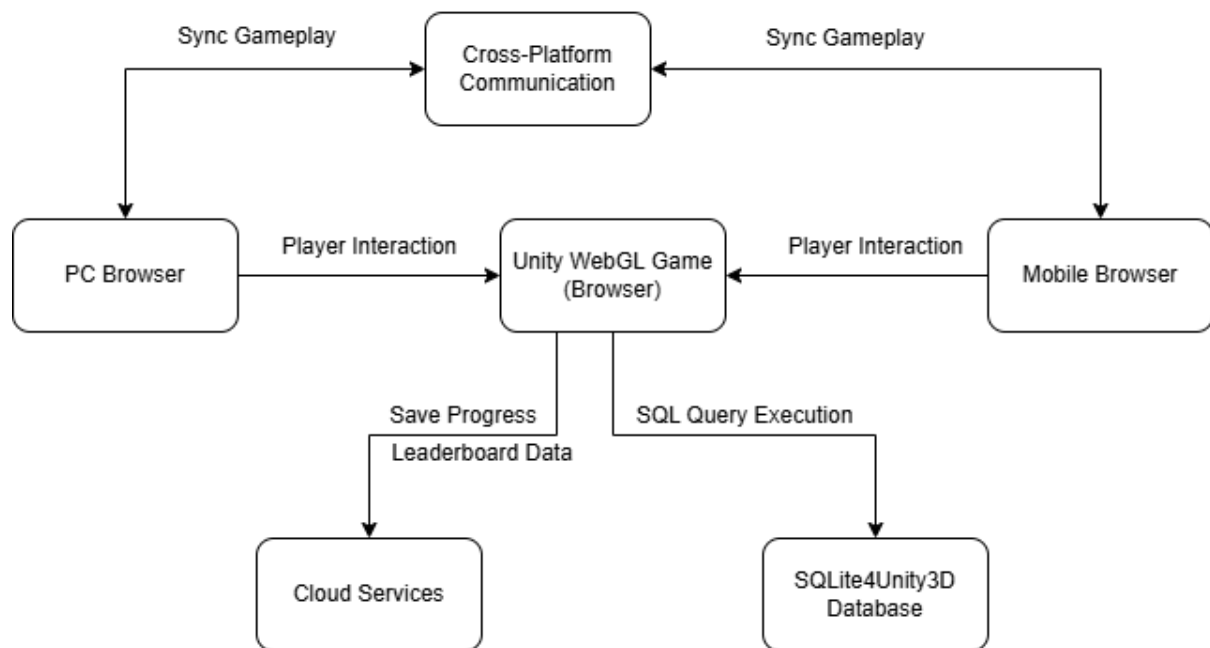
Cloud Services

Saves players' progress and supports dynamic content updates. Google's GCP is used to host both the game and the web interface for query interaction.

Cross-Platform Integration

Data transfer between devices is handled through HTTP requests. Players interact with the game world on the PC and build queries using the web interface on their phone.

High-Level Diagram



User Use Cases

The game's primary user is a learner who wants to build SQL skills.

Use Case Flow:

1. Starting the Game

- 1.1. The player launches the game in a web browser (Unity WebGL).
- 1.2. The game displays a 3D scene.

2. Opening SQL Mode

- 2.1. The player enters SQL Query Mode from his mobile device.
- 2.2. UI shows the available tables and schema on the PC screen.

3. Constructing and Executing an SQL Query

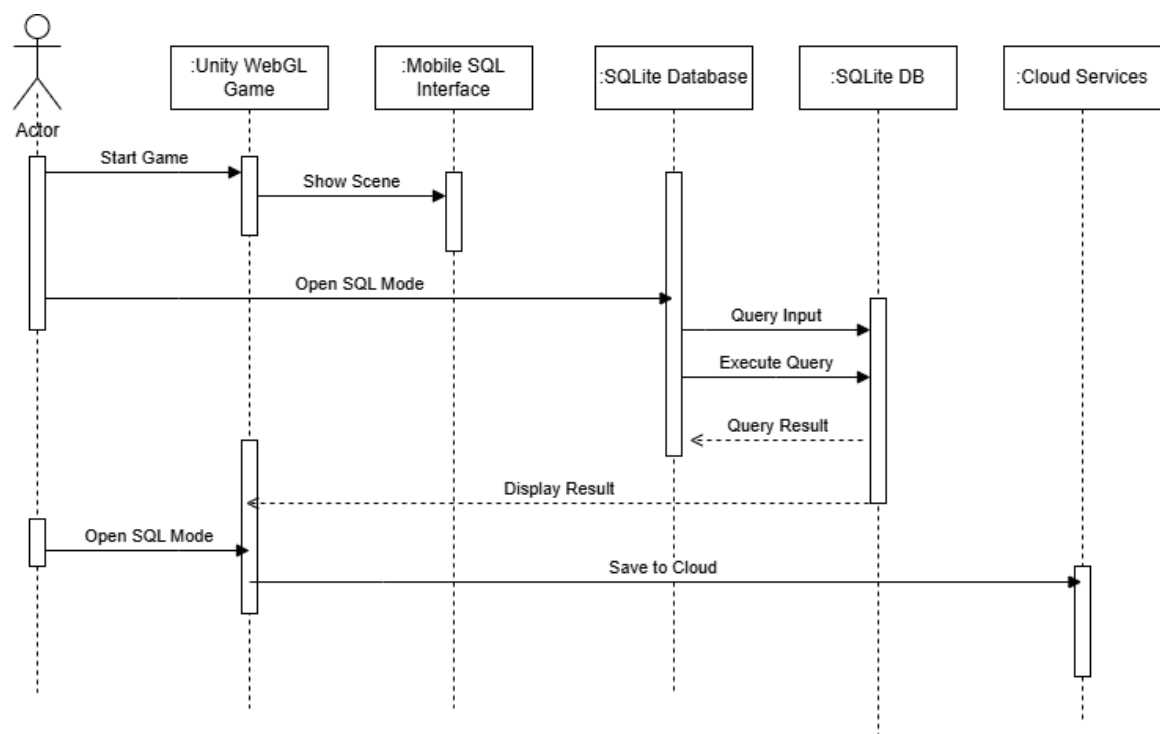
- 3.1. The player inputs a query using the mobile interface.
- 3.2. The query is sent to the SQLite Database via HTTP requests.
- 3.3. The database processes the query and executes it.

4. Retrieving and Displaying Results

- 4.1. SQLite Database returns the query results.
- 4.2. The result is sent back to the Unity WebGL Game, which displays the output on the PC screen.

5. Saving Progress to the Cloud

- 5.1. Once a task is completed, the game saves progress to Cloud Services (Google GCP).
- 5.2. the player can resume from where he left off on any device.



Testing Methods

The game will be tested across multiple phases to ensure functionality, performance, and usability:

Unit Testing

SQL query execution, game logic, and UI interactions will be verified using the Unity Test Framework.

Functional Testing

Core gameplay mechanics, including switching between PC and mobile, will be tested to confirm they work as expected.

Integration Testing

Communication between the WebGL game, SQLite database, and cloud services will be tested to ensure seamless synchronization.

Performance Testing

The Unity Profiler will analyse how fast queries run and improve game responsiveness to prevent slowdowns.

User Testing

We will test the game to assess usability, difficulty balance, and overall experience.

User Guide

Screen Structure

At the start, Main Menu will be represented and include:

1. Start Game
 - 1.1. Start
 - 1.2. Choose difficulty
2. Continue saved game
3. Tutorials
4. View Leaderboard
5. Exit



Game Flow

Exploration Mode (PC)

Players navigate a 3D world, investigate crime scenes, and gather clues. Objects and clues provide hints for constructing SQL queries.

SQL Query Mode (Mobile)

Players open the query-building interface on their mobile device. They construct SQL queries using a drag-and-drop system or by typing commands. The query is executed, and results are sent back to the PC screen.

Progression & Feedback (PC)

If the query is correct, new information is revealed, advancing the game. If incorrect, the players are prompted to adjust their approach.

Saving & Leaderboard Updates

Progress is automatically saved in the cloud. Players can check their ranking on the leaderboard.

Gameplay

Players assume the role of a detective solving intricate crime mysteries. Each session revolves around investigating a unique case, such as solving a murder or uncovering criminal activities.

Players explore a richly detailed 3D environment, collecting clues and interacting with objects to uncover vital information. The gameplay is centered around solving SQL-based puzzles to interrogate dynamic databases and progress through the story.

The databases evolve dynamically as the player progresses. At the beginning, the player is introduced to one simple table, such as CrimeEvidence, accompanied by a narrative explanation.

The player starts with basic tasks like SELECT and FROM queries to retrieve simple data, and progresses to more advanced challenges, such as JOIN operations and filtering with WHERE clauses.

As the player unlocks new tables, the schema dynamically expands, requiring increasingly complex queries and critical thinking.

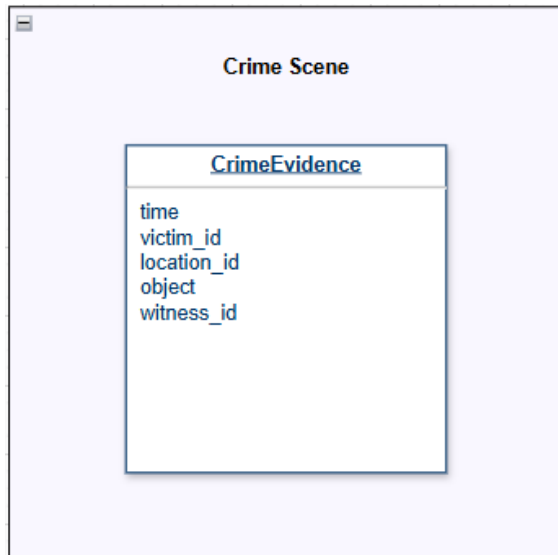
The detective is guided on what to do next, such as being directed to the police station to access the Persons table, with visual clues like an arrow pointing to the correct door. This progression ensures players develop confidence and advanced SQL skills while staying immersed in the narrative.



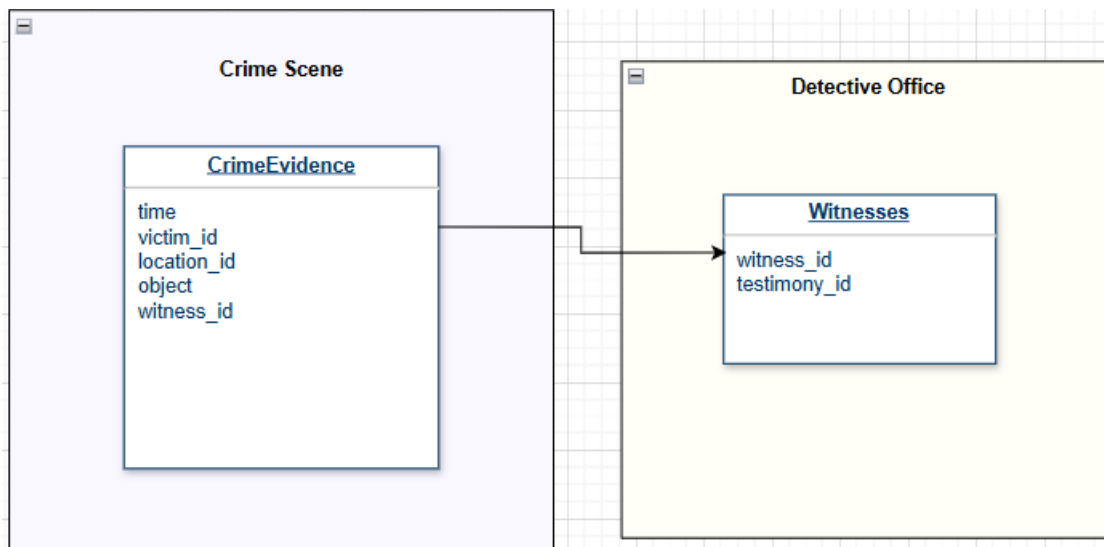
For example:

Level 1: The player starts with CrimeEvidence table and builds basic queries to retrieve data about the crime.

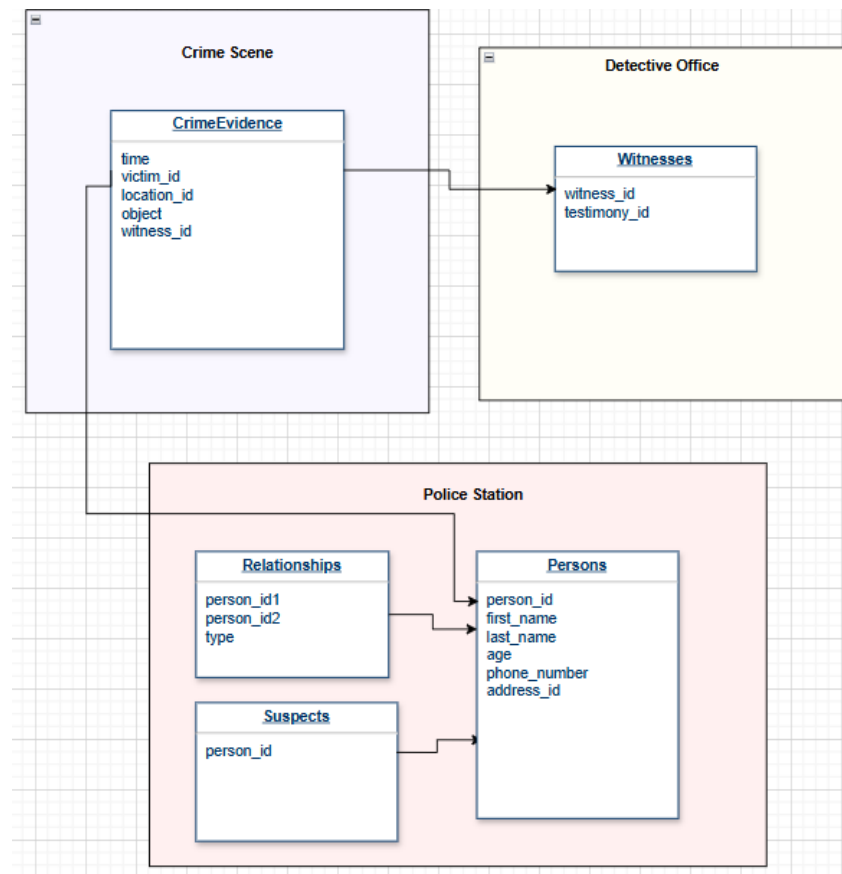
after correct answer, the player will move to detective's office.



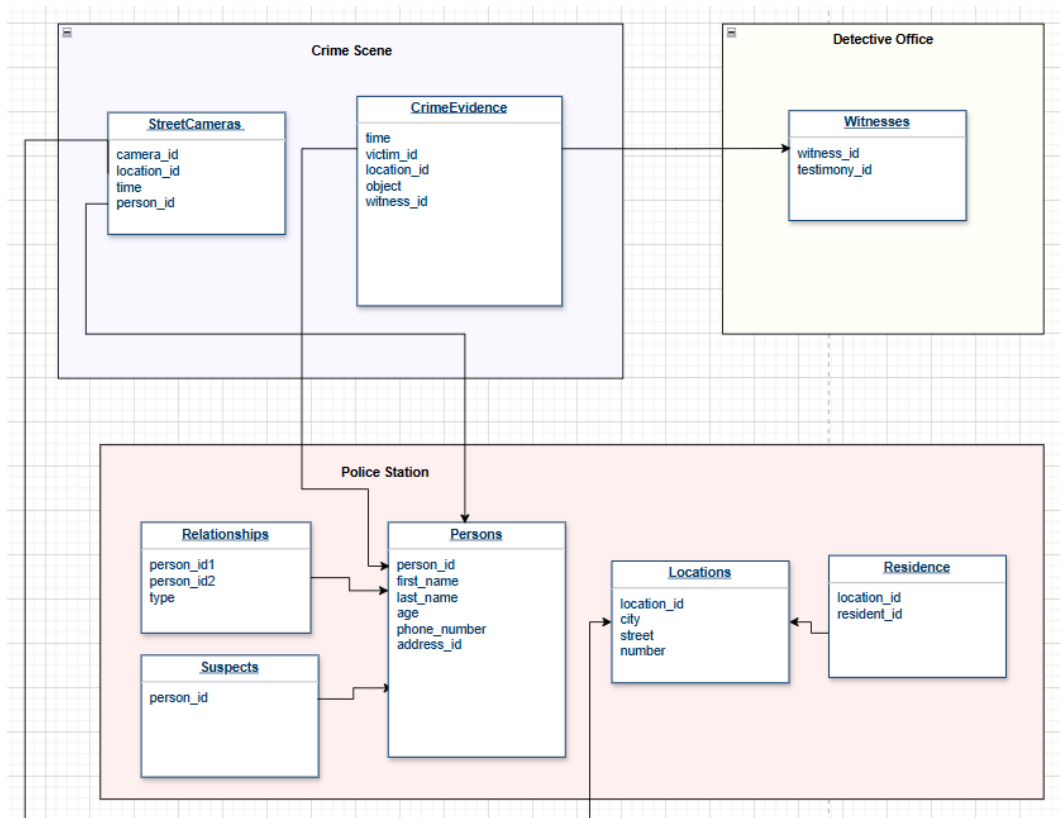
Level 2: After moving to the detective's office, the Witnesses table is introduced. The player learns to join tables to match evidence with witness testimonies.



Level 3: In the police station, additional tables such as Persons, Suspects and Relationships are unlocked. The player now faces multi-table queries to identify connections between suspects, witnesses, and victims, and filtering on wanted fields.



On next levels, more tables are unlocked such as Locations, Residence, and StreetCameras.



SQL mechanics

The game progresses through solving questions using SQL data retrieval, seamlessly blending education and gameplay. The SQL mechanics are designed to provide a hands-on, interactive learning experience while challenging the player's logical and problem-solving skills.

Platform Integration

The system operates on two interconnected platforms:

- 1. PC:** Displays the main game environment, the evolving schema, and query results. Players interact with the narrative and explore the crime scene or detective's office for clues.
- 2. mobile:** Serves as the SQL query-building platform, allowing players to interact directly with the database to construct queries and manage data.

Phone Interface Features

The phone acts as the core tool for database interaction. The interface includes the following components:

- 1. Query Panel:** An empty panel for constructing SQL queries, with an "EXECUTE" button to run the retrieval operation.
- 2. Reserved Words Panel:** Displays essential SQL keywords (SELECT, FROM, WHERE, JOIN, aggregate functions, etc.).
- 3. Options List:** Shows available tables and fields for the player to query.
Once a table is selected, its fields are displayed for players to drag into the query panel.

Query Construction

Players can construct their queries by dragging reserved words and tables to the query panel. After selecting SELECT and FROM, players can choose fields from the selected table to display, and add conditions like WHERE clauses or aggregate functions.

The Suspects Table

In addition to query-building, players are provided with a "suspects" table, which serves as a dynamic tool for tracking and managing their investigation. Players can:

- 1. Add Records:** Use INSERT INTO to add individuals they consider suspicious.
- 2. Delete Records:** Use DELETE to remove individuals they rule out as suspects.
- 3. Retrieve Records:** Query the suspects table to guess the final answer - who they believe committed the crime. Incorrect guesses result in losing a life and deducting points based on the scoring system, adding stakes to the player's choices.

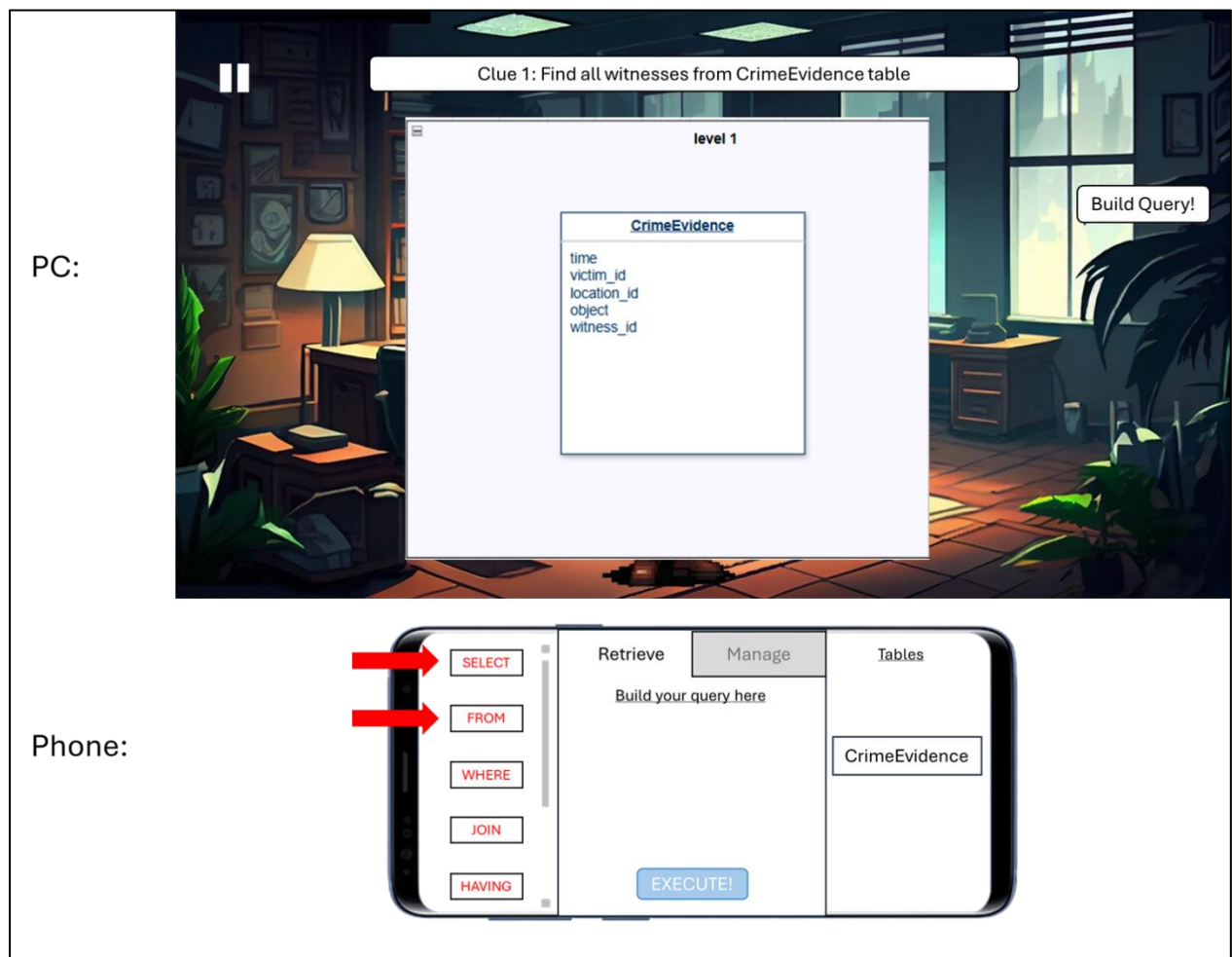
Schema Presentation on PC

The SQL mechanics use the PC for game display and schema visualization, while the phone handles query-building. Players start with simple tables like CrimeEvidence and expand the schema as new levels introduce advanced SQL concepts. Query results displayed on the PC drive the narrative, reinforcing learning through gradual progression and immediate feedback.

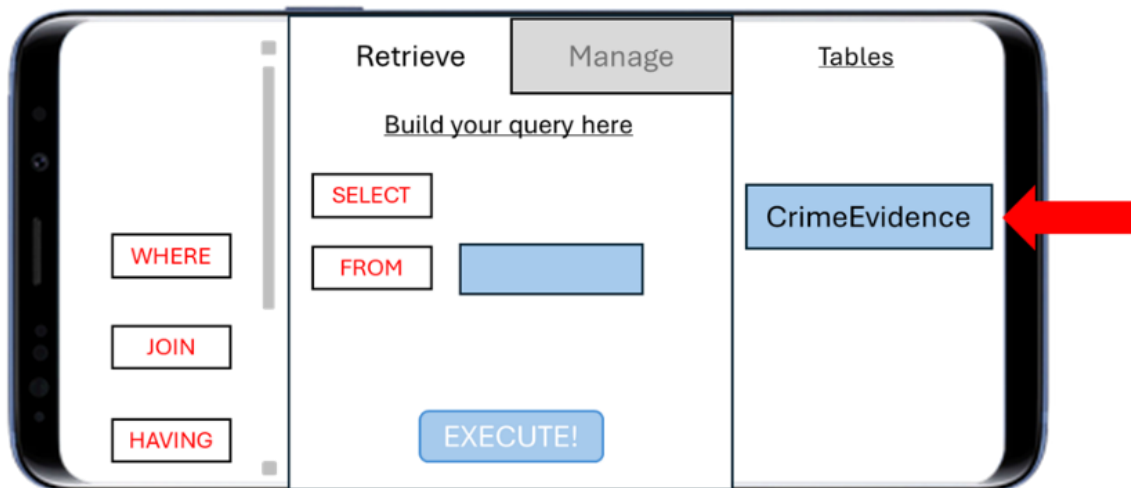
The player starts in his initial place, and has a task to do



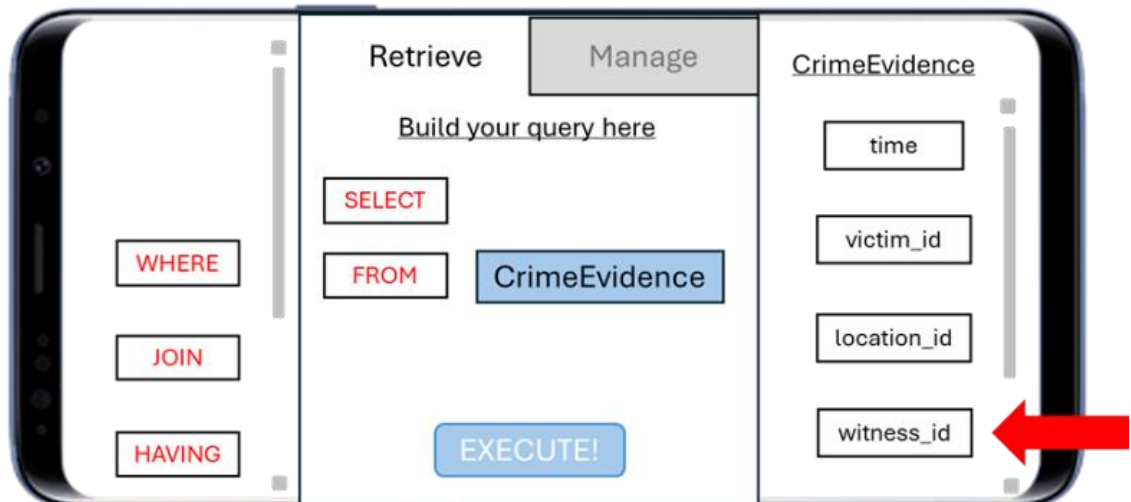
as clicking on SQL query mode on mobile, the screen on PC show the current scheme and table.



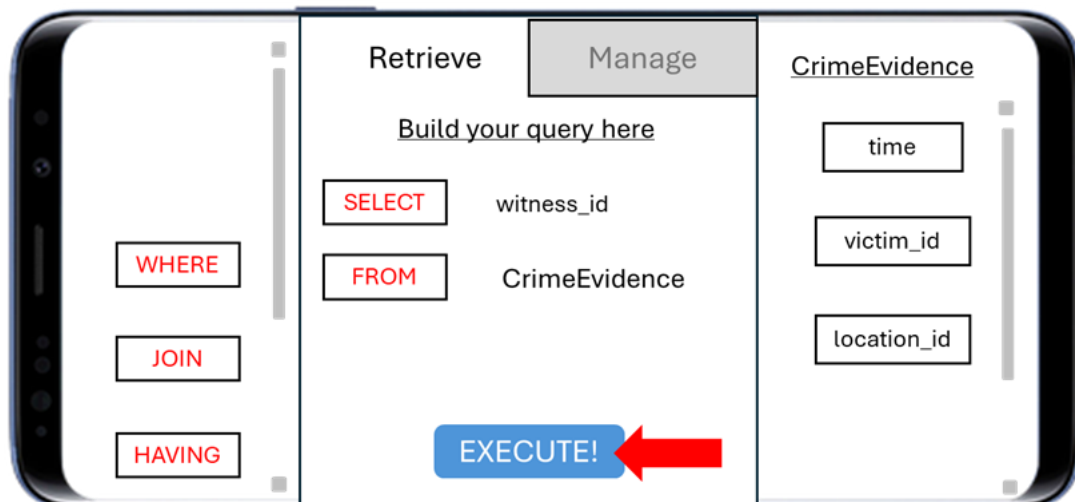
After dragging the From clause, a list of available tables is shown on right side



After choosing the table, its fields are shown on the right side.



After building a valid query, the player can click on Execute button



The output is shown on PC and an answer (correct or incorrect).



Running the Project

To set up the project from scratch:

- Clone the Repository (https://github.com/ofekcofman98/UnityFinalProject_Ofek_Ariel).
- Open the Project in Unity.
- Install SQLite4Unity3D for database integration.
- Install Unity WebGL Build Module.
- Set up Google Cloud services for saving player progress.
- Run the game in Unity by clicking "Play".
- Generate a WebGL build to host on a web server.

Code Maintenance

Project Structure

Organized assets folder: Stores game assets, models, and UI elements

Scripts: Organized into modules:

- 3D & Movement: handles player movement and interactions with the environment.
- GameLogic Core game functionality.
- Database SQL query handling and DBs.
- UI Manages menu screens and query interfaces.
- Scenes: Stores game environments and levels

Components septations

Keep 3D movement and environmental interactions separate from game logic and database operations to avoid dependencies between systems.

Design Patterns

Using reusable and modular design patterns learnt to improve maintainability and scalability.

Future Work

Advanced SQL Features

Add challenges involving GROUP BY, HAVING and more.

New Game Modes

Multiplayer Mode: Players could compete or collaborate to solve cases.

AI generator

AI that would build a case by itself