

# Mastering Generative AI & Agents for Developers – Contest #1

 Get Featured on Our Official LinkedIn Handle!

 Hey AI Innovators!

It's time to put your **Generative AI** skills to the test with an exciting challenge! This is your opportunity to **build something innovative, showcase your expertise, and get featured on Codecademy's official LinkedIn handle!**

 Project Challenge: Build Your AI Agent Project

 SQL Query Buddy — Conversational AI for Smart Data Insights (RAG + LangChain + VectorDB)

**SQL Query Buddy** is a **conversational AI agent** that lets users explore databases using **natural language** — just like chatting with a data analyst.

It converts plain-English questions into **accurate, optimized SQL queries**, executes them, and goes a step further — providing **AI-driven insights** on top of the results. This helps users understand **patterns, trends, and anomalies** in their data instantly.

Built with **LangChain**, **Vector Databases**, and **Gradio**, the agent maintains **context across conversations**, supports **multi-table reasoning**, and delivers **actionable insights**, not just raw numbers.

## Core Features

### Conversational Querying

Ask complex, natural questions such as:

- “Show top 5 customers by total sales.”
- “Now filter them to California only.”
- “What’s the total revenue from them this year?”

The chatbot remembers context — it understands “them” and “this year” naturally.

### RAG-Powered SQL Generation

Uses **LangChain + VectorDB** to semantically search table schemas and metadata before generating SQL — ensuring **accuracy and context-awareness** across multiple tables.

### Query Optimization

Suggests **faster JOINs, indexing strategies, and optimized aggregations** to improve database performance.

### AI-Driven Insights (Beyond Raw Results)

After executing a query, SQL Query Buddy uses the **LLM** to interpret and summarize data with contextual insights, such as:

- “Sales in California grew 15% month-over-month.”
- “Customer Alice Chen contributed 22% of total Q1 revenue.”
- “Product category *Electronics* accounts for 40% of total sales.”

This makes the tool **insightful, interpretive, and decision-supportive** — not just a query translator.

## Explainable SQL

Each query includes a beginner-friendly explanation:

“This query selects the top 5 customers by summing their total order amounts and sorting them in descending order.”

## Context Retention

Maintains **conversation history** — remembers previous queries, filters, and results for **smooth, human-like follow-ups**.

## Chat Interface

A clean, interactive interface that displays:

-  User questions
-  Generated SQL
-  Raw query results
-  AI-generated insights & explanations

## Example Natural-Language Questions

Use these for demos or testing:

1. “Show me the top 5 customers by total purchase amount.”
2. “Which product category made the most revenue this quarter?”
3. “List customers who haven’t ordered anything in the last 3 months.”
4. “Show total sales per region for 2024.”
5. “Find the average order value for returning customers.”
6. “How many unique products were sold in January?”
7. “Which salesperson generated the highest sales last month?”
8. “From the previous result, filter customers from New York only.”
9. “Show the trend of monthly revenue over time.”
10. “How many orders contained more than 3 items?”

Each query automatically:

- Generates SQL
- Executes it
- Displays raw results
- Provides an **AI insight narrative** summarizing trends or anomalies

## Example Schema (Retail Commerce Domain)

Table	Description
<b>customers</b>	Customer information and region
<b>products</b>	Product catalog with prices and categories
<b>orders</b>	Tracks each purchase and total
<b>order_items</b>	Links orders to products with quantity and subtotal

```
-- ======  
-- 📦 SCHEMA: Sample Retail Commerce Database  
-- ======  
  
-- 1 Customers Table  
  
CREATE TABLE customers (  
    customer_id INTEGER PRIMARY KEY,  
    name TEXT NOT NULL,  
    email TEXT UNIQUE,  
    region TEXT,  
    signup_date DATE  
);
```

-- 2 Products Table

```
CREATE TABLE products (
    product_id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    category TEXT,
    price DECIMAL(10,2)
);
```

-- 3 Orders Table

```
CREATE TABLE orders (
    order_id INTEGER PRIMARY KEY,
    customer_id INTEGER,
    order_date DATE,
    total_amount DECIMAL(10,2),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

-- 4 Order Items Table

```
CREATE TABLE order_items (
    item_id INTEGER PRIMARY KEY,
    order_id INTEGER,
    product_id INTEGER,
    quantity INTEGER,
    subtotal DECIMAL(10,2),
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
```

);

-- =====

--  Sample Customers

-- =====

```
INSERT INTO customers (customer_id, name, email, region, signup_date) VALUES
(1, 'Alice Chen', 'alice.chen@example.com', 'California', '2023-02-01'),
(2, 'John Patel', 'john.patel@example.com', 'New York', '2023-05-15'),
(3, 'Maria Lopez', 'maria.lopez@example.com', 'Texas', '2022-11-30'),
(4, 'David Johnson', 'david.johnson@example.com', 'Florida', '2023-07-22'),
(5, 'Sofia Khan', 'sofia.khan@example.com', 'Illinois', '2023-04-10');
```

-- =====

--  Sample Products

-- =====

```
INSERT INTO products (product_id, name, category, price) VALUES
(1, 'Laptop Pro 15', 'Electronics', 1200.00),
(2, 'Wireless Mouse', 'Accessories', 40.00),
(3, 'Standing Desk', 'Furniture', 300.00),
(4, 'Noise Cancelling Headphones', 'Electronics', 150.00),
```

```
(5, 'Office Chair Deluxe', 'Furniture', 180.00);
```

```
-- =====
```

```
-- 📦 Sample Orders
```

```
-- =====
```

```
INSERT INTO orders (order_id, customer_id, order_date, total_amount) VALUES  
(101, 1, '2024-01-12', 1240.00),  
(102, 2, '2024-03-05', 340.00),  
(103, 3, '2024-02-20', 1600.00),  
(104, 1, '2024-04-02', 330.00),  
(105, 4, '2024-05-15', 480.00),  
(106, 5, '2024-06-10', 180.00);
```

```
-- =====
```

```
-- 📦 Sample Order Items
```

```
-- =====
```

```
INSERT INTO order_items (item_id, order_id, product_id, quantity, subtotal) VALUES  
(1, 101, 1, 1, 1200.00),  
(2, 101, 2, 1, 40.00),  
(3, 102, 2, 2, 80.00),  
(4, 102, 4, 1, 150.00),  
(5, 103, 3, 5, 1500.00),  
(6, 103, 2, 2, 80.00),  
(7, 104, 5, 1, 180.00),  
(8, 104, 2, 3, 120.00),  
(9, 105, 4, 3, 450.00),  
(10, 106, 5, 1, 180.00);
```

## How to Generate more Data

🔗 <https://www.mockaroo.com>

Mockaroo lets you:

- Upload your schema (columns and data types)
- Generate thousands of **realistic-looking rows** (names, emails, prices, dates)
- Export as **SQL, CSV, or JSON**

### Example setup

Field	Type	Notes
customer_id	Row Number	Auto increment
name	Full Name	Random
email	Email Address	Random
region	State / Country	Choose from list
signup_date	Date	Range (2022–2024)

For orders:

- Add fields like `order_id`, `customer_id` (foreign key style), `order_date`, `total_amount`.
- Use Mockaroo's "*Formula*" feature for totals (e.g., `rand(100,2000)`).

**Output as SQL** and directly import into your database.  
Perfect for demo datasets up to 100,000 rows.

## Tech Stack

Layer	Technology
<b>Frontend (Chat UI)</b>	Gradio / React
<b>AI Layer</b>	LangChain + GPT Models / LLM
<b>Vector Search</b>	Vector Database (FAISS / Pinecone / Milvus / Chroma)
<b>Backend</b>	Python (FastAPI or LangChain Agent)
<b>Database</b>	SQLite / PostgreSQL / MySQL
<b>Retrieval Framework</b>	RAG (Schema Embeddings + Contextual Retrieval)

## Project Guidelines

- The project must be **fully functional** — no incomplete features in demo or code.
- Add **extra capabilities** to make your project stand out.
- Ensure **clean UI/UX** — intuitive and user-friendly.
- **Experiment & innovate** — bring unique twists or features.
- **Originality matters** — no plagiarism or copied code.
- Post your project on **LinkedIn**, tag **Codecademy**, and use the hashtag **#CodecademyGenAIBootcamp**.
- Submit your LinkedIn post link in the **#project-showcase** channel on Discord.



## Important Dates

Milestone	Date
<b>Submission Deadline</b>	15th Feb 2026
<b>Winner Announcement</b>	22th Feb 2026

🏅 Winners will be featured on Codecademy's official LinkedIn page — and you can proudly add this to your **portfolio or resume** as a recognized AI project!

## 💡 Why Participate?

- Apply your **Generative AI & Agent-Building** skills in a real-world context.
- Add an **impressive AI project** to your portfolio.
- Boost your visibility and employability with a recognized achievement.
- Get featured and stand out as an **AI innovator**.
- The **best project** will be selected by your instructor — their decision is final.