



สถาบันไอจีเนียส
IT GENIUS INSTITUTE

COURSE ID
DBC-46

DATABASE / AI

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

PostgreSQL for AI Apps with pgvector

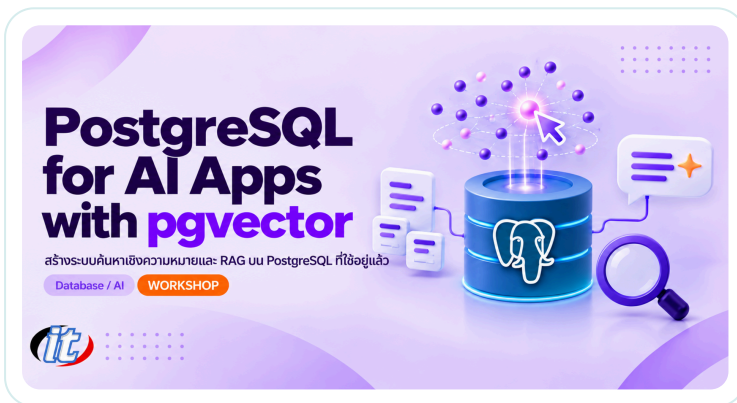
สร้างระบบค้นหาเชิงความหมายและ RAG บน PostgreSQL ที่ใช้อยู่แล้ว

● 6 ชั่วโมง

● 1 วัน

● พื้นฐาน SQL และ Python

● ระดับ Intermediate



ระยะเวลา: 1 วัน

6 ชั่วโมง (1 วัน วันละ 6 ชม.) · 09:00-16:00 น.



รูปแบบ: Classroom / Hybrid

Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house



รายละเอียดหลักสูตร

www.itgenius.co.th

ระบบค้นหาเชิงความหมายและ RAG กลายเป็นฟีเจอร์ที่หลายองค์กรต้องการเพิ่มเข้าไปในแอปพลิเคชัน แต่การนำ Vector Database แยกมาอีก ระบบหมายถึงข้อมูลที่ต้องซิงค์สองที่ สิทธิ์การเข้าถึงที่ต้องดูแลซ้ำ และงานปฏิบัติการที่เพิ่มขึ้น สำหรับทีมที่ใช้ PostgreSQL อยู่แล้ว ส่วนขยาย pgvector ช่วยให้เก็บเวกเตอร์ไว้ข้างข้อมูลธุรกิจพื้นฐานข้อมูลเดียวกัน ค้นหาด้วย SQL ที่คุ้นเคย และใช้ Transaction, Backup และระบบสิทธิ์เดิมได้ทันที

หลักสูตรนี้พาลงมือตั้งแต่ติดตั้ง PostgreSQL 18 กับ pgvector ด้วย Docker สร้าง Embedding จากเอกสารด้วย Python เลือกชนิดข้อมูลและ Index ที่เหมาะกับขนาดข้อมูล ปรับพารามิเตอร์ของ HNSW และ IVFFlat ไปจนถึง Hybrid Search ที่รวม Full-text Search กับเวกเตอร์ การแยกข้อมูลแต่ละลูกค้าด้วย Row-Level Security และการสร้าง RAG Pipeline ที่อ้างอิงแหล่งที่มาได้ ปิดท้ายด้วยการวัดคุณภาพการค้นหา การดูแลระบบจริง และ Workshop ระบบถามตอบจากเอกสารตัวอย่าง (อบรม 1 วัน 6 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

PG PostgreSQL 18

vec pgvector

RAG RAG

Python Python

AI Embeddings

ราคาค่าอบรม · PUBLIC / IN-HOUSE

3,900 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม ไฟล์ Docker Compose โค้ด Python ตัวอย่าง ชุดเอกสารสำหรับฝึก และ Certificate · ผู้เรียนเตรียม API Key ของผู้ให้บริการ Embedding/LLM หรือเลือกใช้โมเดลแบบ Local · รับจัด In-house ขอใบเสนอราคาได้



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- อธิบายหลักการ Embedding และการวัดความคล้ายด้วย Cosine, L2 และ Inner Product ได้
- ประเมินได้ว่าเมื่อใดควรเก็บเวกเตอร์ใน PostgreSQL และเมื่อใดควรใช้ Vector Database แยก
- ติดตั้ง PostgreSQL 18 กับ pgvector และเลือกชนิดข้อมูล vector, halfvec, sparsevec ได้เหมาะสม
- สร้าง Embedding จากเอกสารด้วย Python พร้อมแบ่ง Chunk และโหลดเข้าฐานข้อมูล
- สร้างและปรับแต่ง Index แบบ HNSW และ IVFFlat รวมถึงการค้นหาที่มีเงื่อนไขกรอง
- สร้าง Hybrid Search และแยกข้อมูลแต่ละ Tenant ด้วย Row-Level Security
- สร้าง RAG Pipeline ที่ดึงข้อมูลจาก PostgreSQL และตอบพร้อมอ้างอิงแหล่งที่มา
- วัดคุณภาพการค้นหาและประสิทธิภาพ แล้ววางแผนดูแลระบบในการใช้งานจริง

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Backend และ AI ที่ต้องสร้างฟีเจอร์ค้นหาเชิงความหมายหรือ RAG
- ทีมที่ใช้ PostgreSQL อยู่แล้วและไม่ต้องการเพิ่ม Vector Database อีกระบบ
- Data Engineer ที่ดูแล Pipeline ข้อมูลและต้องเตรียมข้อมูลให้ระบบ AI
- Tech Lead และสถาปนิกระบบที่ต้องเลือกแนวทางจัดเก็บเวกเตอร์ให้ทีม

03 จุดเด่นของหลักสูตร Highlights

- **ใช้ฐานข้อมูลที่มีอยู่** เวกเตอร์อยู่ข้างข้อมูลธุรกิจใน PostgreSQL
- **ลงมือทุกขั้นตอน** ติดตั้ง โหลดข้อมูล สร้าง Index และค้นหา
- **ค้นหาแม่นยำขึ้น** Hybrid Search รวมคำค้นกับความหมาย
- **ปลอดภัยแบบหลายลูกค้า** Row-Level Security และ Role อ่านอย่างเดียว
- **RAG ที่ตรวจสอบได้** คำตอบอ้างอิงเอกสารต้นทางทุกครั้ง

04 ความรู้พื้นฐาน Prerequisites

- เขียน SQL ได้ในระดับ SELECT, JOIN, CREATE TABLE และ CREATE INDEX
- เขียน Python พื้นฐานได้ เช่น ฟังก์ชัน การติดตั้งแพ็คเกจ และการเรียก API
- ใช้คำสั่ง Docker เบื้องต้นและ Command Line ได้
- มี API Key ของผู้ให้บริการ Embedding/LLM หรือพร้อมใช้โมเดลแบบ Local ที่วิทยากรเตรียมให้
- ไม่จำเป็นต้องมีประสบการณ์ด้าน Machine Learning มาก่อน

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook (Windows, macOS หรือ Linux) RAM 16 GB ขึ้นไป
ฐานข้อมูล	PostgreSQL 18 และ pgvector 0.8.x uu Docker Desktop
ภาษาและไลบรารี	Python 3.12 ขึ้นไป, psycopg 3 และ pgvector-python
Embedding/LLM	API ของผู้ให้บริการที่ผู้เรียนเลือก หรือโมเดลแบบ Local
เครื่องมือ	VS Code หรือ Editor ที่ถนัด และ psql หรือ DBeaver

หมายเหตุ: ผู้เรียนเตรียม API Key ของผู้ให้บริการ Embedding และ LLM ที่ตนเองใช้ หรือเลือกใช้โมเดล Embedding และ LLM แบบ Local ที่ทำงานบนเครื่องโดยไม่ต้องส่งข้อมูลออกภายนอก วิทยากรส่งไฟล์ Docker Compose และรายการติดตั้งให้ล่วงหน้า ควรติดตั้งและดึง Image ให้เรียบร้อยก่อนวันอบรม Workshop ใช้ชุดเอกสารตัวอย่างหรือเอกสารของผู้เรียนเองที่ไม่มีข้อมูลลับ

06 ภาพรวมการเรียนรู้ 1 วัน Big Picture

วัน	หัวข้อหลัก
ช่วงเช้า Section 1-4	แนวคิด Embedding และ Vector Search, ติดตั้ง PostgreSQL 18 กับ pgvector ด้วย Docker, ชนิดข้อมูล vector, halfvec, sparsevec, สร้าง Embedding ด้วย Python พร้อม Chunking เอกสาร และ Index แบบ HNSW กับ IVFFlat
ช่วงบ่าย Section 5-10	Hybrid Search ด้วย Full-text Search และ Rank Fusion, กรอง Metadata และแยกข้อมูลลูกค้าด้วย Row-Level Security, สร้าง RAG Pipeline, วัฏ Recall และประสิทธิภาพ, ดูและบบจริง และ Workshop ระบบถามตอบเอกสาร

DAY
1

สร้างระบบค้นหาเชิงความหมายและ RAG บน PostgreSQL

เนื้อหาการอบรม · Section 1-6

รวม
6 ชั่วโมง

1 Embedding, Vector Search และทำไมต้อง PostgreSQL

- Embedding คืออะไร และทำไมข้อความที่ความหมายใกล้เคียงกันจึงอยู่ใกล้กันในพื้นที่เวกเตอร์
- การวัดความคล้าย: Cosine Distance, L2 Distance และ Inner Product ควรเลือกแบบใด
- เก็บเวกเตอร์ใน PostgreSQL เทียบกับ Vector Database แยก: ความสอดคล้องของข้อมูล สิทธิ์ และงานดูแล
- ข้อจำกัดที่ต้องรู้: ขนาดข้อมูล หน่วยความจำของ Index และปริมาณการค้นหาต่อวินาที

2 Lab: ติดตั้ง PostgreSQL 18 และ pgvector

- รับ PostgreSQL 18 พร้อม pgvector ด้วย Docker Compose และเปิดใช้ CREATE EXTENSION vector
- ชนิดข้อมูล vector, halfvec และ sparsevec กับทางเลือกให้เหมาะกับขนาดและจำนวนมิติ
- Binary Quantization ด้วยชนิด bit และการค้นหาแบบสองขั้นเพื่อลดหน่วยความจำ
- Lab: สร้างตาราง documents และ chunks แล้วลองค้นหาด้วยตัวดำเนินการระยะทางทั้งสามแบบ

3 Lab: สร้าง Embedding ด้วย Python และโหลดเอกสาร

- เรียก Embedding API หรือโมเดล Open Source แบบ Local จาก Python
- Chunking: แบ่งตามขนาด ตามหัวข้อ และการกำหนดส่วนซ้อนทับระหว่าง Chunk
- เก็บ Metadata ที่จำเป็น เช่น แหล่งที่มา หน้า วันที่ และเวอร์ชันของโมเดล
- Lab: โหลดเอกสารแบบ Batch ด้วย psycopg 3 และ COPY เพื่อความเร็ว

4 Lab: Exact vs Approximate Search และ Index

- Exact Search แบบ Sequential Scan เทียบกับ Approximate Nearest Neighbor
- HNSW: พารามิเตอร์ m, ef_construction และ hnsw.ef_search กับผลต่อความแม่นยำและความเร็ว
- IVFFlat: การกำหนด lists, ivfflat.probes และเหตุผลที่ต้องมีข้อมูลก่อนสร้าง Index
- Iterative Index Scan ใน pgvector 0.8 สำหรับการค้นหาที่มีเงื่อนไข WHERE
- Lab: สร้าง Index ทั้งสองแบบ แล้วเทียบเวลาและผลลัพธ์กับ Exact Search

5 Lab: Hybrid Search ด้วย Full-text Search

- จุดอ่อนของ Vector Search กับข้อเฉพาะ รหัสสืบลำ และคำย่อ
- Full-text Search ด้วย tsvector, tsquery และ GIN Index
- รวมผลสองแบบด้วย Reciprocal Rank Fusion ใน SQL คำสั่งเดียว
- Lab: เทียบผลค้นหาแบบเวกเตอร์ แบบคำค้น และแบบ Hybrid กับคำถามชุดเดียวกัน

6 Lab: กรอง Metadata และ Multi-tenant ด้วย RLS

- ออกแบบคอลัมน์ Metadata และ JSONB สำหรับกรองตามประเภท วันที่ และสิทธิ์
- Partial Index และ Partitioning เมื่อข้อมูลแต่ละกลุ่มมีขนาดต่างกันมาก
- Row-Level Security แยกข้อมูลแต่ละ Tenant ให้การค้นหาเห็นเฉพาะข้อมูลของตนเอง
- Lab: ทดสอบว่าผู้ใช้คนละ Tenant ค้นหาแล้วไม่เห็นเอกสารของกันและกัน



DAY
1

สร้างระบบค้นหาเชิงความหมายและ RAG บน PostgreSQL

เนื้อหาการอบรม · Section 7-10 (ต่อ)

รวม
6 ชั่วโมง

7 Workshop: สร้าง RAG Pipeline ด้วย Python

- ขั้นตอน RAG: รับคำถาม สร้าง Embedding ถึง Chunk ที่เกี่ยวข้อง และประกอบ Context
- เขียน Prompt ให้ LLM ตอบจาก Context เท่านั้น และบอกเมื่อไม่พบคำตอบ
- แบบแหล่งที่มาของแต่ละคำตอบ เช่น ชื่อเอกสารและหน้า
- Workshop: สร้าง API ถามตอบขนาดเล็กที่ดึงข้อมูลจาก PostgreSQL

8 Lab: วัดคุณภาพการค้นหาและประสิทธิภาพ

- สร้างชุดคำถามทดสอบพร้อมคำตอบที่ถูกต้องจากเอกสารจริง
- วัด Recall@k และเทียบผลของการปรับ Chunking, Index และ Hybrid Search
- อ่าน EXPLAIN ANALYZE ว่า Query ใช้ Index หรือไม่ และจุดที่ช้า
- Lab: ปรับ ef_search และ probes แล้วบันทึกผลความแม่นยำเทียบกับเวลา

9 ดูและระบบในการใช้งานจริง

- เวลาและหน่วยความจำในการสร้าง Index, maintenance_work_mem และการสร้างแบบขนาน
- อัปเดตเอกสาร ลบข้อมูลเก่า และการสร้าง Embedding ใหม่เมื่อเปลี่ยนโมเดล
- Backup, Restore และ Replication ของตารางที่มีเวกเตอร์
- บริการ Managed ที่รองรับ pgvector เช่น Supabase, Neon และ Amazon RDS
- ทางเลือก: เปิดฐานข้อมูลให้ AI Agent ผ่าน MCP อย่างปลอดภัยด้วย Role อ่านอย่างเดียว

10 Workshop: ระบบถามตอบจากเอกสารของผู้เรียน

- Workshop: นำชุดเอกสารตัวอย่างหรือเอกสารของผู้เรียนเข้าระบบตั้งแต่ Chunking ถึง Index
- ทดสอบคำถามจริง ตรวจสอบแหล่งที่มา และปรับจุดที่ค้นหาไม่เจอ
- สรุปแนวทางนำไปใช้กับระบบของตนเองและ Checklist ก่อนขึ้น Production

07 เรียนจบแล้วได้อะไร Outcomes

- ระบบค้นหาที่ใช้ได้จริง Vector Search และ Hybrid Search บน PostgreSQL
- เลือก Index เป็น HNSW หรือ IVFFlat พร้อมค่าที่ปรับแล้ว
- Pipeline ข้อมูล Chunking, Embedding และการโหลดแบบ Batch
- แยกข้อมูลปลอดภัย Row-Level Security สำหรับหลาย Tenant
- RAG พร้อมอ้างอิง API ถามตอบที่บอกแหล่งที่มา
- วัดผลได้ ชุดทดสอบ Recall@k และผล EXPLAIN

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรม ไฟล์ Docker Compose และสคริปต์ SQL สำหรับตั้งค่าฐานข้อมูล
- โค้ด Python สำหรับ Chunking, Embedding, Hybrid Search และ RAG Pipeline ที่ทำระหว่างเรียน
- ชุดคำถามทดสอบพร้อมสคริปต์วัด Recall@k และ Checklist ก่อนขึ้น Production
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร ไกยม วิทยากรของสถาบันไอทิจีนียัสถ่ายทอดการสร้างระบบค้นหาและ RAG บน PostgreSQL แบบลงมือทำทุกขั้นตอน ตั้งแต่การเตรียมข้อมูล การเลือกชนิดข้อมูลและ Index ไปจนถึงการวัดผลและดูแลระบบ ผู้เรียนจึงเข้าใจถึงวิธีทำและเหตุผลเบื้องหลังแต่ละการตัดสินใจ และนำโค้ดกับแนวทางที่ได้กลับไปปรับใช้กับฐานข้อมูลของตนเองได้ทันที

ราคาค่าอบรม 3,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

3,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

LINE @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE