**วิธีการก่อนหน้า**

วิธีการเดิมใช้การอัปโหลดไฟล์ CSV ผ่านหน้า Web UI ของ **Apache Superset** โดยตรง ผู้ใช้จำเป็นต้องเลือกไฟล์ กำหนดการแมปคอลัมน์ และตั้งค่าต่างๆ ผ่าน graphical interface ซึ่งเหมาะสำหรับข้อมูลขนาดเล็กและผู้ใช้ที่ไม่มีความรู้ทางเทคนิค

**ขั้นตอนการทำงานของวิธีการเดิม:**

1. อัปโหลดไฟล์ CSV ผ่านปุ่ม **"CSV Upload"** ใน Superset
2. กำหนดการแมปคอลัมน์ด้วยตนเองผ่าน UI
3. ตั้งค่าตารางและข้อมูลผ่านฟอร์มกราฟิก
4. บันทึก dataset และสร้าง visualizations ต่อ

**วิธีการใหม่ที่เสนอ**

วิธีการใหม่ใช้ **Python script ร่วมกับ SQLAlchemy และ Pandas** สำหรับการนำเข้าข้อมูลแบบอัตโนมัติ มีการทำความสะอาดข้อมูล (data cleaning) การตรวจสอบความถูกต้อง (validation) และการจัดการ errors อย่างครอบคลุม เหมาะสำหรับข้อมูลขนาดใหญ่และกระบวนการที่ต้องทำซ้ำ

**ขั้นตอนการทำงานของวิธีการใหม่:**

1. อ่านข้อมูลจากไฟล์ CSV โดยใช้ **Pandas**
2. ทำความสะอาดและแปลงรูปแบบข้อมูล (data cleaning & transformation)
3. เชื่อมต่อกับ **PostgreSQL** ผ่าน SQLAlchemy engine
4. นำเข้าข้อมูลแบบ **batch processing** ด้วย chunksize
5. ตรวจสอบและ validate ผลลัพธ์หลังการนำเข้า

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**My charts**

**วิธีการใหม่ที่ทำ**

A screenshot of a login

AI-generated content may be incorrect.

**Connect database**

**Refresh Interval ปรับไว้เพื่อลดโหลดบนระบบและแสดงข้อมูลแบบเกือบ real-time โดยไม่กินทรัพยากรเกินจำเป็น (setไว้5นาที)**

A screenshot of a computer

AI-generated content may be incorrect.

**My dashboard**

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**Charts ที่สร้าง**

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**import\_data.py**

A screen shot of a computer program

AI-generated content may be incorrect.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ตารางเปรียบเทียบ** | | **วิธีการเก่า** | **วิธีการใหม่ที่เสนอ** | |
| **เทคนิคการนำเข้า** | | อัปโหลดผ่าน **Web UI** โดยตรง | ใช้ **Python script ร่วมกับ SQLAlchemy** | |
| **ความเร็ว** | | **ปานกลาง** (ประมาณ 10,000 records/นาที) | **สูง** (ประมาณ 100,000 records/นาที) | |
| **ความยืดหยุ่น** | | จำกัด อยู่ในขอบเขตของ UI | สูงมาก สามารถปรับแต่งผ่านโค้ดได้ | |
| **การทำซ้ำ** | | ต้องทำแบบ Manual ทุกครั้ง | อัตโนมัติสมบูรณ์ สามารถใช้ script เดิมซ้ำได้ | |
| **การจัดการข้อผิดพลาด** | | จำกัด (แสดงข้อความ error ผ่าน UI) | ครบถ้วน มี **error handling** เต็มรูปแบบ | |
| **ความเหมาะสมกับข้อมูลขนาดใหญ่** | | ไม่เหมาะสม สำหรับไฟล์ ≥1GB | เหมาะสมมาก รองรับไฟล์ ≥10GB | |
| **การตรวจสอบคุณภาพข้อมูล** | | พื้นฐาน (basic validation) | ครอบคลุม ตรวจสอบความถูกต้องข้อมูลเต็มรูปแบบ | |
| **การบำรุงรักษา** | | ยาก ต้องจำขั้นตอน Manual | ง่าย สามารถใช้ **version control** ผ่าน Git | |
| **การขยาย Scale** | | จำกัด ขึ้นกับข้อจำกัดของ UI | สูง รองรับ **distributed processing** | |
| **การเรียนรู้** | | เรียนรู้ง่าย ไม่ต้องเขียนโค้ด | ต้องมีความรู้ด้าน **programming** | |
| **การติดตามการทำงาน** | | จำกัด ต้องตรวจสอบด้วยตนเอง | ครบถ้วน รองรับ **automated logging & monitoring** | |
| **ข้อจำกัดของแต่ละวิธีการ** | | |
| |  |  | | --- | --- | | วิธีการ | ข้อจำกัด | | วิธีการเดิม | ❌ จำกัดสำหรับข้อมูลขนาดใหญ่ ❌ ไม่มีระบบ recovery สำหรับ errors ❌ ทำซ้ำผลลัพธ์ได้ยาก ❌ ไม่มี version control สำหรับกระบวนการ | | วิธีการใหม่ | ❌ ต้องการความรู้ทางเทคนิคมากขึ้น ❌ ใช้เวลาในการพัฒนา script เบื้องต้น ❌ ต้องการ infrastructure สำหรับ automation | | | |

**ข้อได้เปรียบหลักของวิธีการใหม่**

**1. ด้านประสิทธิภาพ**

* ความเร็วเพิ่มขึ้น **10 เท่า** จากการประมวลผลแบบ batch
* รองรับข้อมูลขนาดใหญ่ เช่น **58 ล้าน records** ได้อย่างมีประสิทธิภาพ
* ลดการใช้งาน memory ผ่านการประมวลผลแบบ **chunks**

**2. ด้านความน่าเชื่อถือ**

* การจัดการ **errors** ครอบคลุมทั้งกระบวนการ
* **Data validation** ก่อนนำเข้าฐานข้อมูล
* **Atomic transactions** เพื่อความสมบูรณ์ของข้อมูล

**3. ด้านการดำเนินงาน**

* **Reproducibility** สูง ด้วย code-based approach
* **Version control** ผ่าน Git สำหรับการติดตามการเปลี่ยนแปลง
* **Automated testing** เพื่อรับประกันคุณภาพของข้อมูล
* ลดเวลาในการประมวลผลจาก **ชั่วโมงเหลือนาที**