Sau khi thu thập dữ liệu:

Bước 1: tải mysql

Bước 2: cài đặt các bảng:

Bảng 1: nhóm các kĩ năng lại – gọi là category cũng được

|  |
| --- |
| CREATE TABLE skill\_groups (  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(100) NOT NULL UNIQUE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 2: Lưu từng kĩ năng cụ thể:

|  |
| --- |
| CREATE TABLE skills (  id INT AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(100) NOT NULL UNIQUE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 3: gán kĩ năng vào từng nhóm

|  |
| --- |
| CREATE TABLE skill\_details (  skill\_id INT NOT NULL,  group\_id INT NOT NULL,  PRIMARY KEY (skill\_id, group\_id),  FOREIGN KEY (skill\_id) REFERENCES skills(id) ON DELETE CASCADE,  FOREIGN KEY (group\_id) REFERENCES skill\_groups(id) ON DELETE CASCADE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 4: bảng công việc chính

|  |
| --- |
| CREATE TABLE jobs (  id INT AUTO\_INCREMENT PRIMARY KEY,  group\_id INT NOT NULL, -- category của job  title VARCHAR(255) NOT NULL,  link TEXT,  location VARCHAR(255),  experience VARCHAR(100),  work\_location\_detail TEXT,  working\_time VARCHAR(255),  deadline DATE,  salary\_raw VARCHAR(100),  salary\_normalized BIGINT,  currency\_unit VARCHAR(10),  FOREIGN KEY (group\_id) REFERENCES skill\_groups(id)  ON DELETE RESTRICT  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 5: Thông tin mô tả chi tiết: mô tả, yêu cầu, phúc lợi.

|  |
| --- |
| CREATE TABLE job\_details (  id INT AUTO\_INCREMENT PRIMARY KEY,  job\_id INT NOT NULL,  description TEXT,  requirements TEXT,  benefits TEXT,  FOREIGN KEY (job\_id) REFERENCES jobs(id) ON DELETE CASCADE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

Bảng 6: Gán kỹ năng vào công việc (Job Skills)

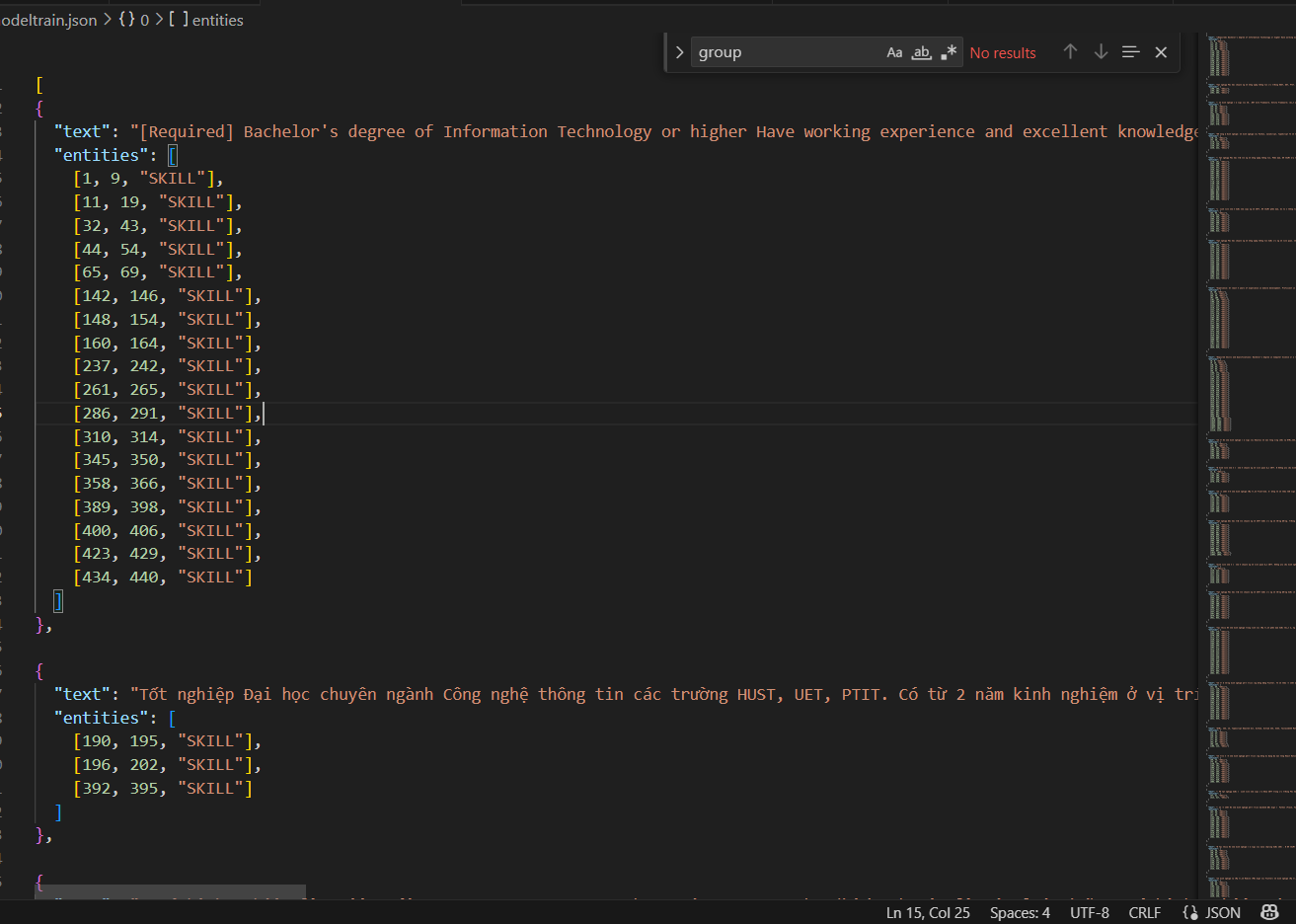
|  |
| --- |
| CREATE TABLE job\_skills (  job\_id INT NOT NULL,  skill\_id INT NOT NULL,  PRIMARY KEY (job\_id, skill\_id),  FOREIGN KEY (job\_id) REFERENCES jobs(id) ON DELETE CASCADE,  FOREIGN KEY (skill\_id) REFERENCES skills(id) ON DELETE CASCADE  ) ENGINE=InnoDB CHARSET=utf8mb4; |

**Bước 3: làm sạch dữ liệu:**

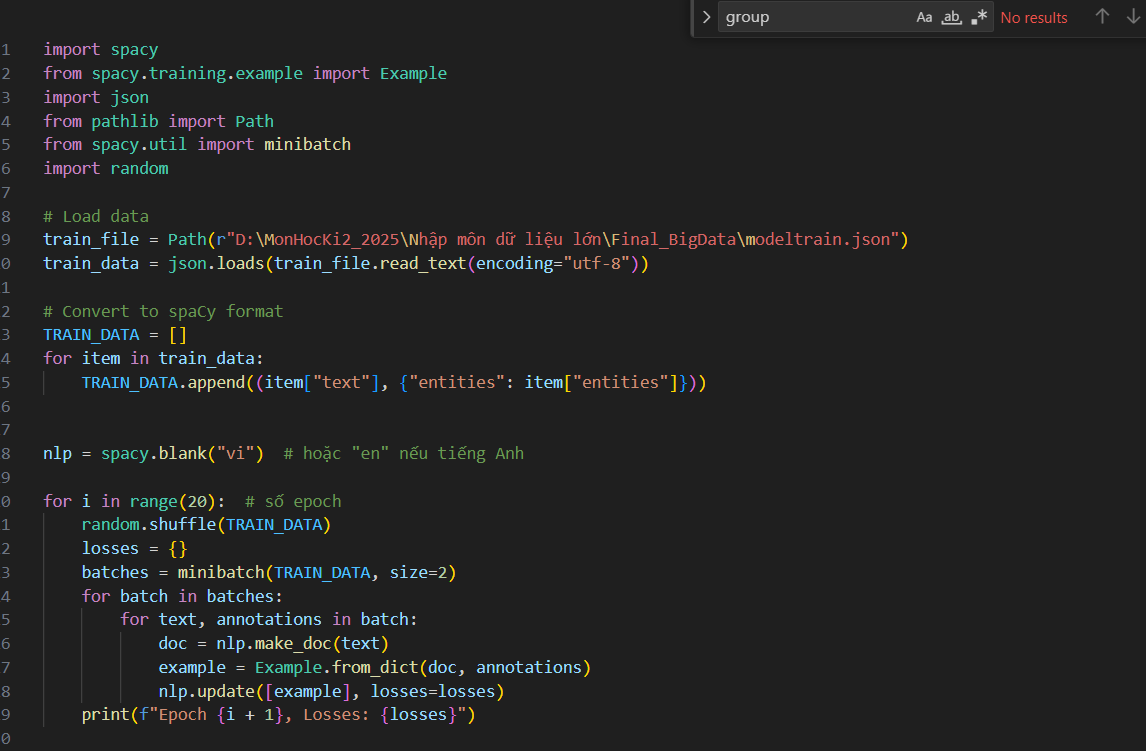
* train dữ liệu để lấy trường dữ liệu name cho bảng Skill. Bằng huấn luyện mô hình nhận diện thực thể (NER) tùy chỉnh với thư viện spaCy. Thì model NER là Nhận diện thực thể có tên là một kỹ thuật trong Xử lý ngôn ngữ tự nhiên (NLP), dùng để tìm và phân loại các cụm từ quan trọng trong văn bản.

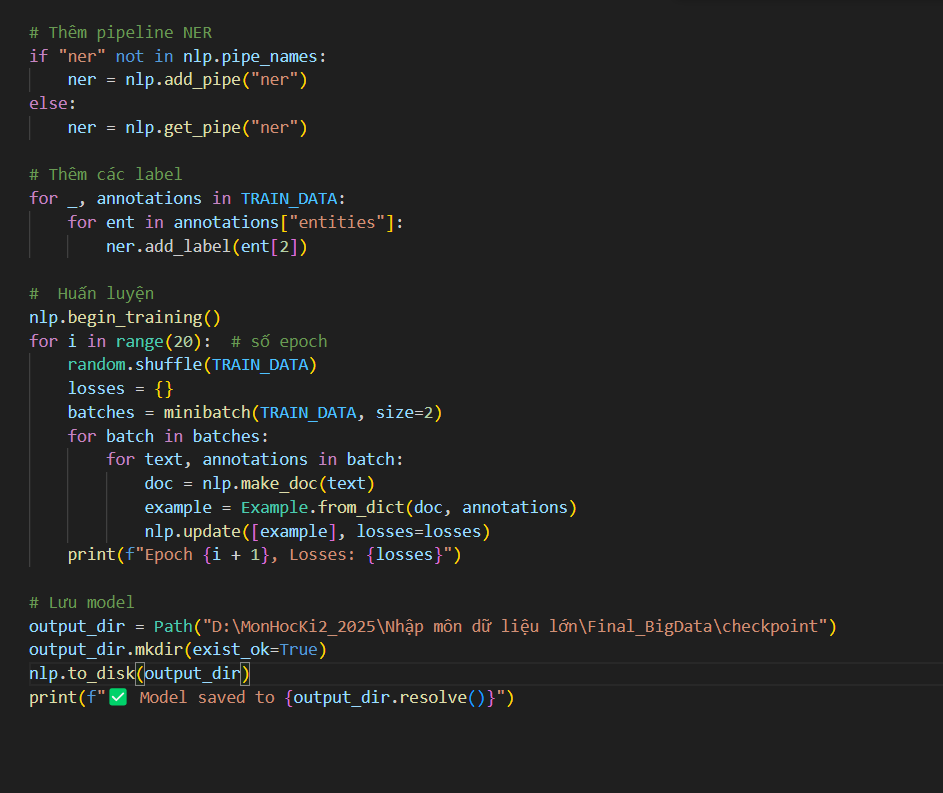
Đầu tiên: tạo file train gồm text, entities.

Có thể áp dụng trang web này để có thể chuyển label: <https://arunmozhi.in/ner-annotator/>



Bước 2: tạo thư mục python train\_model\_ner để có thể lấy dữ liệu cho bảng skill



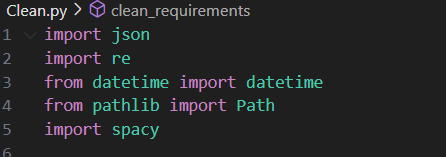


* cài đặt các thư viện cần thiết:

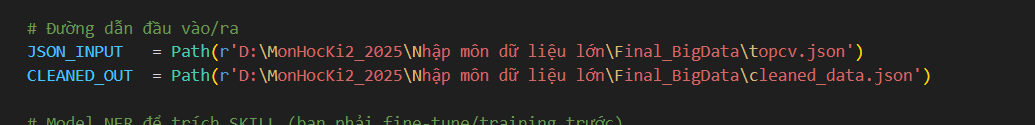
|  |
| --- |
| $ pip install spacy  $ pip install pyvi ==> giúp hỗ trợ tiếng việt |

Bước 3: clean dữ liệu:

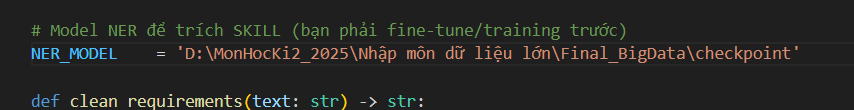
* import thư viện:



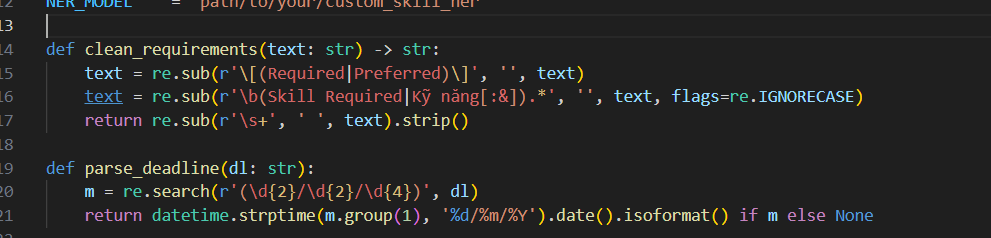
* đưa đường dẫn đầu vào:



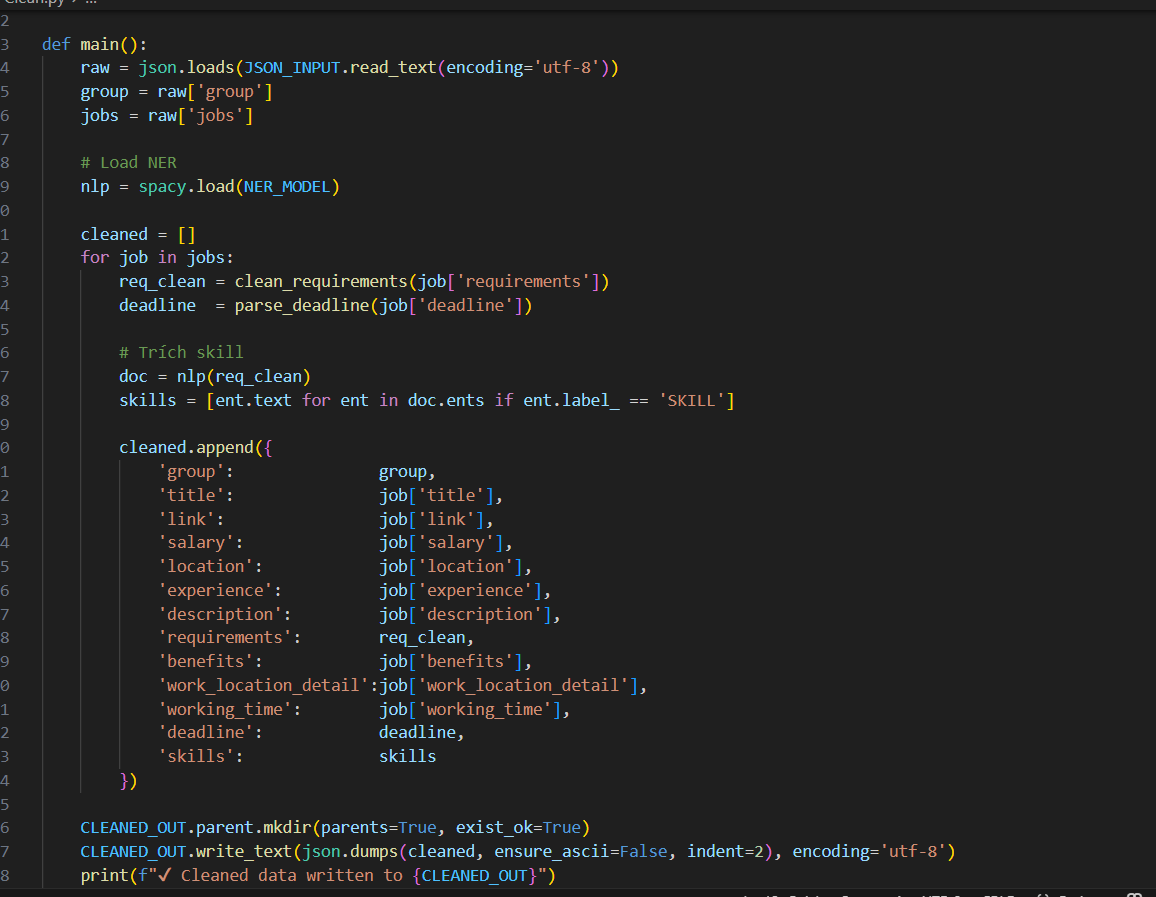
* đưa model đã huấn luyện vào mô hình:

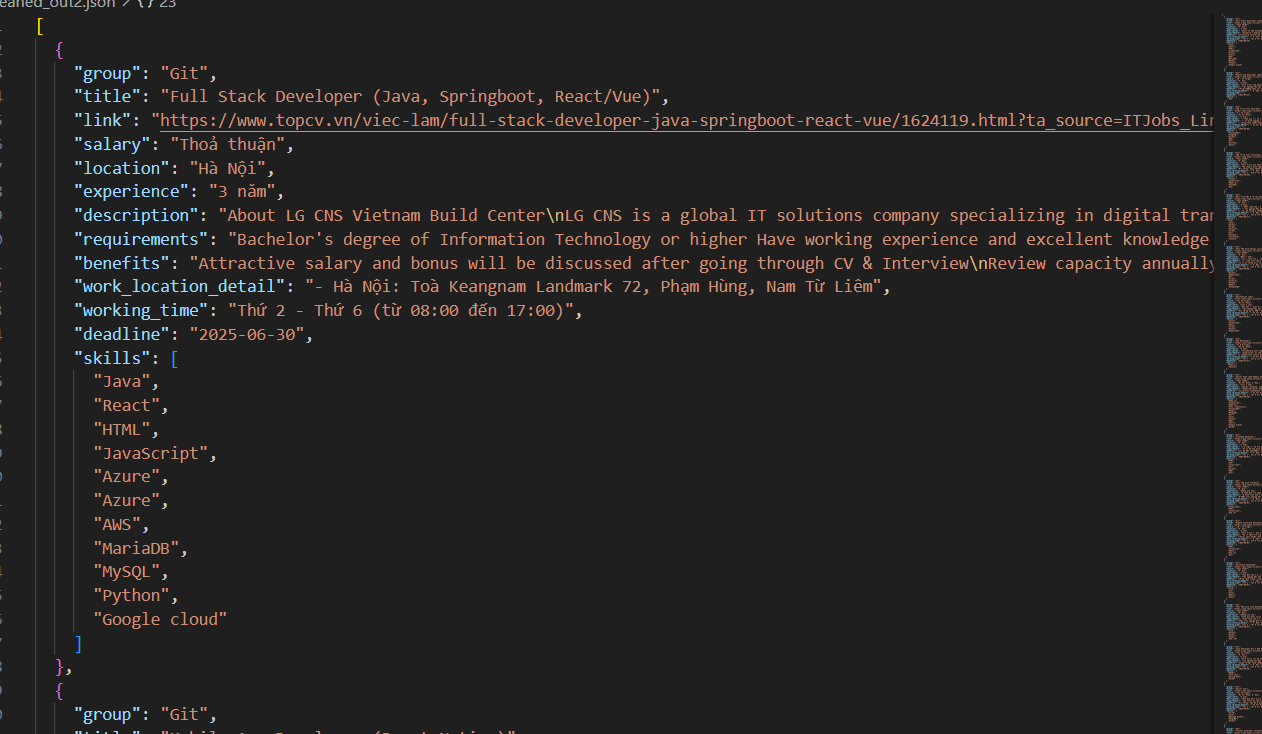


* làm sạch trường dữ liệu requirements, và ngày tháng:



* load model và đưa dữ liệu vào:

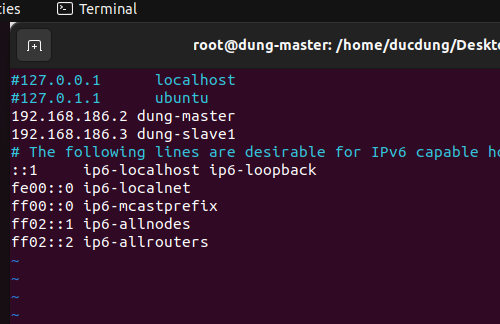


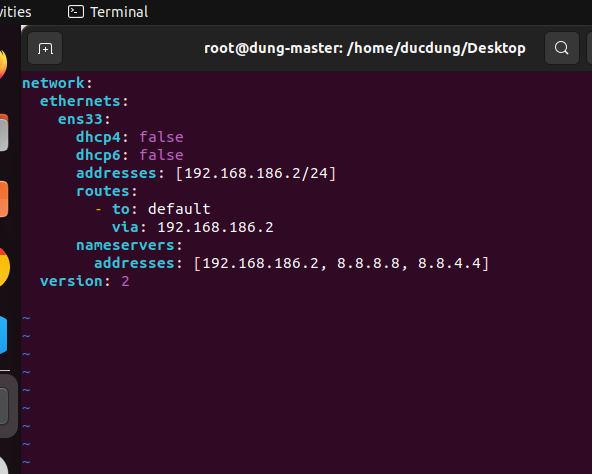
* dữ liệu mẫu làm sạch:  
  

Cấu hình kết nối Hadoop giữa máy Windows và Linux để trao đổi dữ liệu

Mục tiêu: Cần kết nối máy Windows với một cụm Hadoop trên Linux để trao đổi dữ liệu. Máy Windows dùng Java 1.8.0\_441 , máy Linux (Ubuntu) chạy Java 1.8.0\_442 và Hadoop 3.x. Mục tiêu là crawl dữ liệu trên Windows, lưu vào CSDL MySQL hoặc MongoDB, rồi đẩy lên sau đó từ hadoop ở ubuntu cũng có thể lấy dữ liệu từ hadoop của Window. Và ngược lại.

1. Cấu hình lại mạng thành :





1. kết nối mạng từ windows sang linux:

* nếu k ping được mạng!

 Vào **Control Panel** → **Windows Defender Firewall** → **Advanced settings**.

 Chọn **Inbound Rules** → tìm **File and Printer Sharing (Echo Request – ICMPv4-In)**.d

 Nháy đúp, trong tab **Advanced** chọn bật (Enable) cho profile **Private** (và Public nếu cần).

* sau bước đó mà vẫn k được !

tắt tưởng lửa tại win:

netsh advfirewall set allprofiles state off

* nếu ping thành công

b4: thiết lập file core-site.xml trong C:\HADOOP\hadoop-3.3.6\etc\hadoop

|  |
| --- |
| <configuration>    <property>      <name>fs.defaultFS</name>      <value>hdfs://dung-master:9000</value>    </property>  </configuration> |

Bước 5: Nếu muốn ghi lên file hadoop ở windows

1. **Trên Ubuntu**, mở file $HADOOP\_HOME/etc/hadoop/hdfs-site.xml, thêm vào trong <configuration>:

|  |
| --- |
| <property>  <name>dfs.permissions</name>  <value>false</value>  </property> |

1. **Restart HDFS**:

bash

Sao chépChỉnh sửa

$HADOOP\_HOME/sbin/stop-dfs.sh

$HADOOP\_HOME/sbin/start-dfs.sh

1. **Trên Windows**, bạn chỉ có thể trực tiếp đọc:

Nếu muốn ghi phải tạo file ở máy chính là master rồi cấp quyền :

hdfs dfs -chown Admin /win\_only ==> cấp quyền cho admin

hdfs dfs -chmod 700 /win\_only ==> cấp quyền cho user

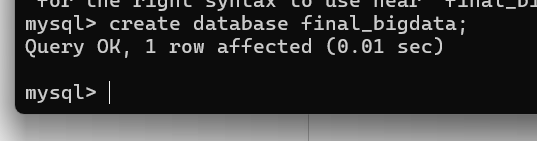
hdfs dfs -chmod 755 /win\_only ==> cho phép mọi người đọc trên trang UI

Bước 6: Cài đặt mySQL:

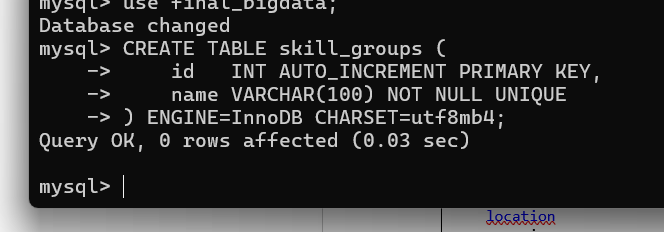
* 1. Setup mysql
  2. Tạo các bảng trong mysql

B1: tạo database:

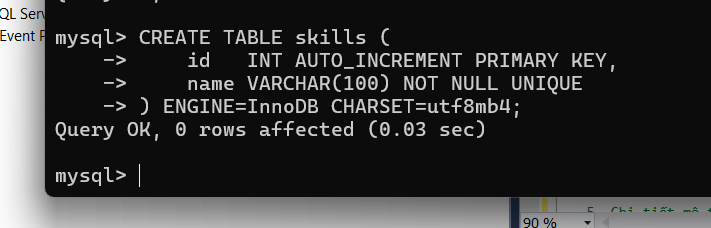
$ create database final\_bigdata;



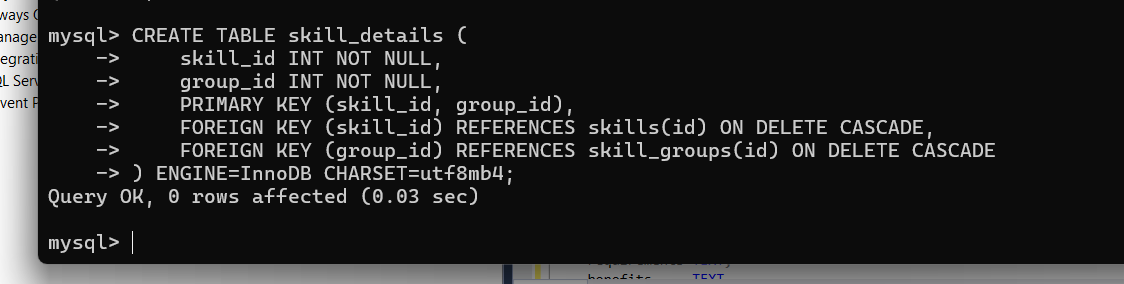
B2: Insert các bảng vào table:

--skill group 

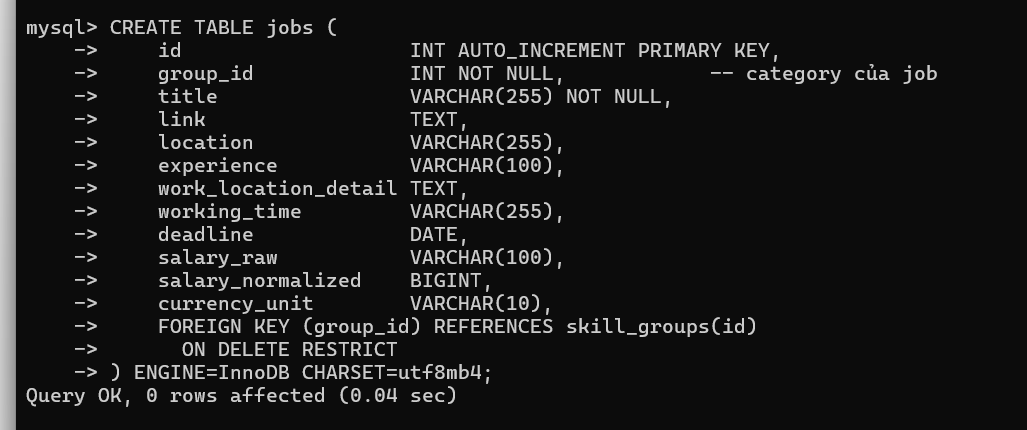
-- skill:



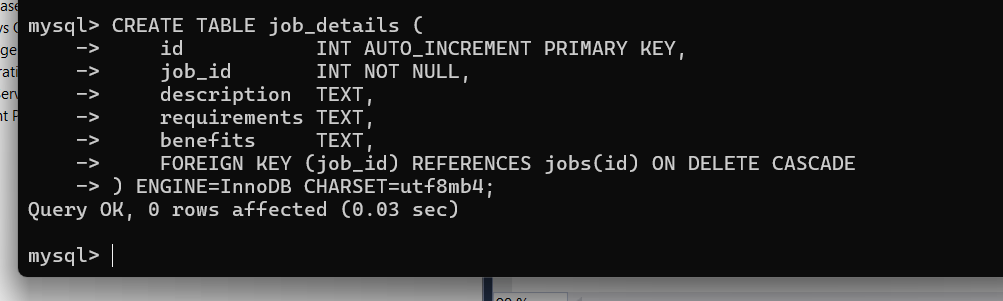
* Skill detail:



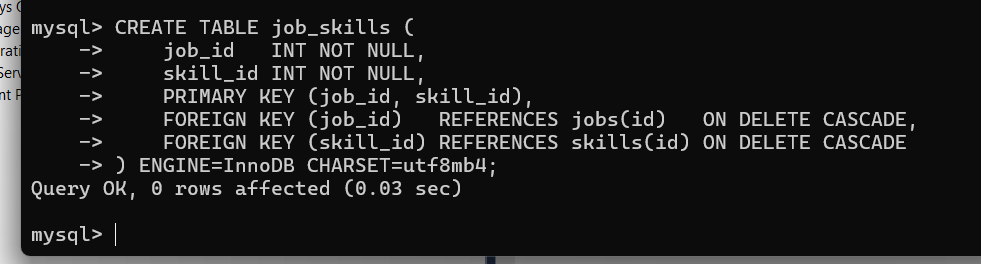
* Job:



* Job detail:



* Skill job:



Bước 7: Thêm dữ liệu vào database ở mysql để có thể lưu dữ liệu:

|  |
| --- |
| import json  import mysql.connector  from pathlib import Path  from datetime import datetime  import re  DB\_CONFIG = {      "host": "localhost",      "user": "root",      "password": "123456789",      "database": "final\_bigdata",      "charset": "utf8mb4"  }  # Load file cleaned\_data.json sau khi đã làm sạch dữ liệu  with open(r"D:\MonHocKi2\_2025\Nhập môn dữ liệu lớn\Final\_BigData\cleaned\_data.json", "r", encoding="utf-8") as f:      jobs = json.load(f)  conn = mysql.connector.connect(\*\*DB\_CONFIG)  cur = conn.cursor()  # 1. Thêm bảng group (skill\_groups)  groups = set(j['group'] for j in jobs)  category\_ids = {}  for g in groups:      cur.execute("INSERT IGNORE INTO skill\_groups(name) VALUES (%s)", (g,))  conn.commit()  cur.execute("SELECT id, name FROM skill\_groups")  for cid, cname in cur.fetchall():      category\_ids[cname] = cid  # 2. Insert skills  skill\_set = {skill for job in jobs for skill in job['skills']}  skill\_ids = {}  for s in skill\_set:      cur.execute("INSERT IGNORE INTO skills(name) VALUES (%s)", (s,))  conn.commit()  cur.execute("SELECT id, name FROM skills")  for sid, sname in cur.fetchall():      skill\_ids[sname] = sid  # 3. Thêm 2 trường dữ liệu group\_id và skill\_id vào bảng skill\_details  for job in jobs:      gid = category\_ids[job['group']]      for s in job['skills']:          sid = skill\_ids.get(s)          if sid:              cur.execute(                  "INSERT IGNORE INTO skill\_details(skill\_id, group\_id) VALUES (%s, %s)",                  (sid, gid)              )  conn.commit()  # 4. thêm các bảng jobs, job\_details, job\_skills  for job in jobs:      # parse deadline      dl = job.get('deadline')      deadline = datetime.strptime(dl, "%Y-%m-%d").date() if dl else None      # insert into jobs with group\_id      group\_id = category\_ids[job['group']]      cur.execute(          """          INSERT INTO jobs(              group\_id, title, link, location,              experience, work\_location\_detail, working\_time,              deadline, salary\_raw, salary\_normalized, currency\_unit          ) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)          """, (              group\_id,              job['title'], job['link'], job['location'],              job['experience'], job['work\_location\_detail'], job['working\_time'],              deadline,              job['salary\_raw'], job['salary\_normalized'], job['currency\_unit']          )      )      job\_id = cur.lastrowid      # thêm dữ liệu cho Job\_detals      cur.execute(          "INSERT INTO job\_details(job\_id, description, requirements, benefits) VALUES (%s,%s,%s,%s)",          (job\_id, job['description'], job['requirements'], job['benefits'])      )      # Thêm dữ liệu cho bảng job\_skills      for s in job['skills']:          sid = skill\_ids.get(s)          if sid:              cur.execute(                  "INSERT IGNORE INTO job\_skills(job\_id, skill\_id) VALUES (%s, %s)",                  (job\_id, sid)              )  conn.commit()  cur.close()  conn.close()  print("✔ Đã lưu vào MySQL thành công!") |