

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

---oOo---



BÀI TẬP VỀ NHÀ
DATABASE TESTING

Mã nhóm: K1

Lớp: 19_3

Email: 19120383@student.hcmus.edu.vn

Môn học: Kiểm thử phần mềm

Thành viên: 19120383 – Huỳnh Tân Thọ

19120426 – Phan Đăng Diễm Uyên

19120469 – Sử Nhật Đăng

19120496 – Trần Thị Khánh Duyên

Giảng viên: Lâm Quang Vũ

Hồ Tuấn Thanh

Trương Phước Lộc

Thành phố Hồ Chí Minh, tháng 11 năm 2022

MỤC LỤC

1. Tạo dữ liệu cho ứng dụng OrangeHRM.....	1
1.1. Công cụ sử dụng	1
1.2. Cấu hình để sử dụng công cụ	1
1.3. Tạo dữ liệu cho bảng ohrm_employment_status	2
1.4. Tạo dữ liệu cho bảng ohrm_job_title.....	3
1.5. Tạo dữ liệu cho các phòng ban (ohrm_subunit)	4
1.6. Tạo dữ liệu cho bảng ohrm_emp_termination.....	5
1.7. Tạo dữ liệu của nhân viên cho mỗi phòng ban (hr_hs_employee)	5
1.8. Tạo profile picture cho mỗi nhân viên (hs_hr_emp_picture)	20
1.9. Tạo dữ liệu cho bảng Loại nghỉ phép (ohrm_leave_type)	22
1.10. Tạo dữ liệu cho bảng Yêu cầu nghỉ phép (ohrm_leave_request).....	22
1.11. Tạo dữ liệu cho bảng nghỉ phép (ohrm_leave)	24
1.12. Chạy project	27
2. Kiểm thử cơ sở dữ liệu cho ứng dụng OrangeHRM	29
2.1. Công cụ sử dụng	29
2.2. Tạo Maven Project bằng IntelliJ Community	29
2.3. Cài đặt các dependency trên pom.xml	30
2.4. Cấu hình cho DBUnit.....	32
2.5. Viết test case	35
2.6. Thực hiện các test case	38
3. Báo cáo tự đánh giá	40

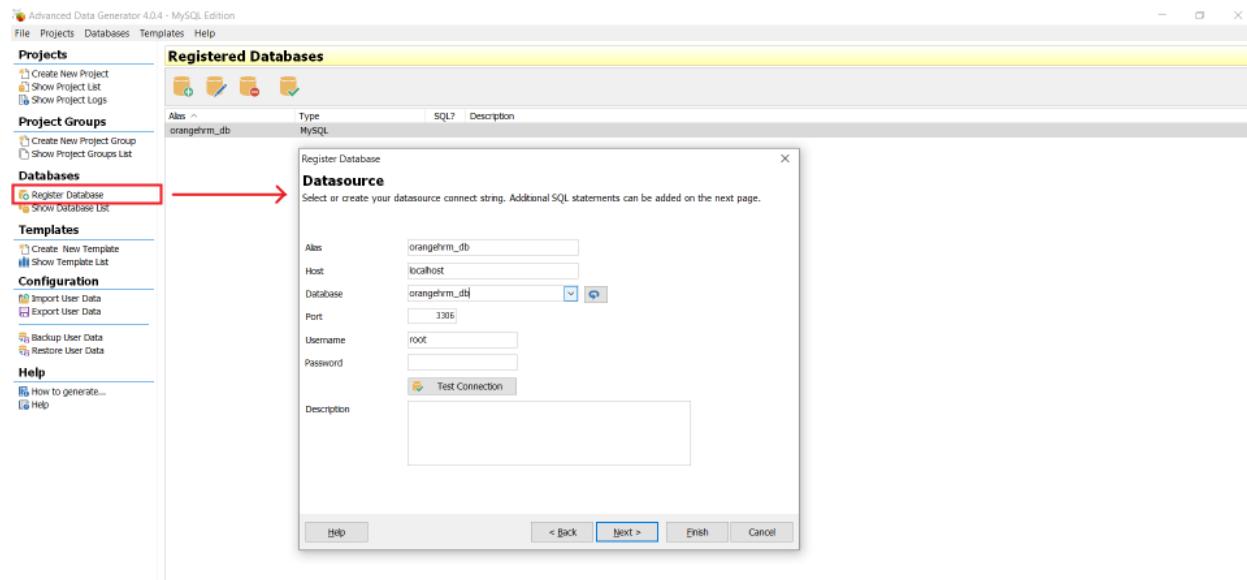
1. Tạo dữ liệu cho ứng dụng OrangeHRM

1.1. Công cụ sử dụng

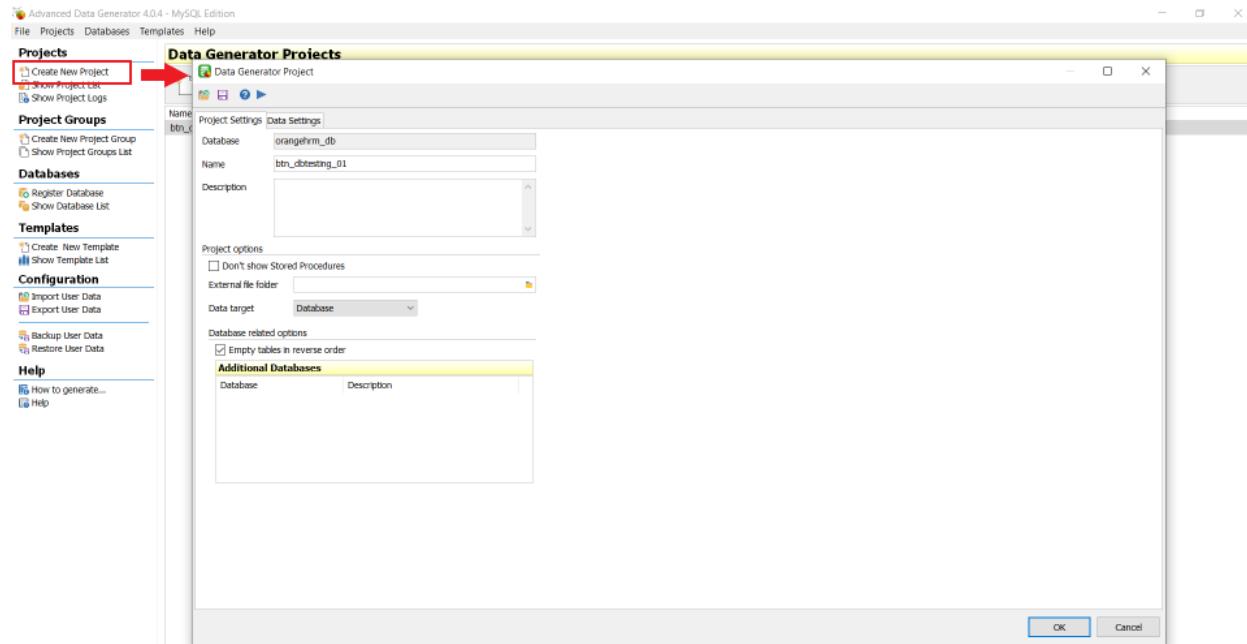
- Upscene Advanced Data Generator MySQL Edition
- Phiên bản 4.0.4

1.2. Cấu hình để sử dụng công cụ

- Vào công cụ, chọn Register Database, điền các thông tin cần thiết để kết nối với MySQL

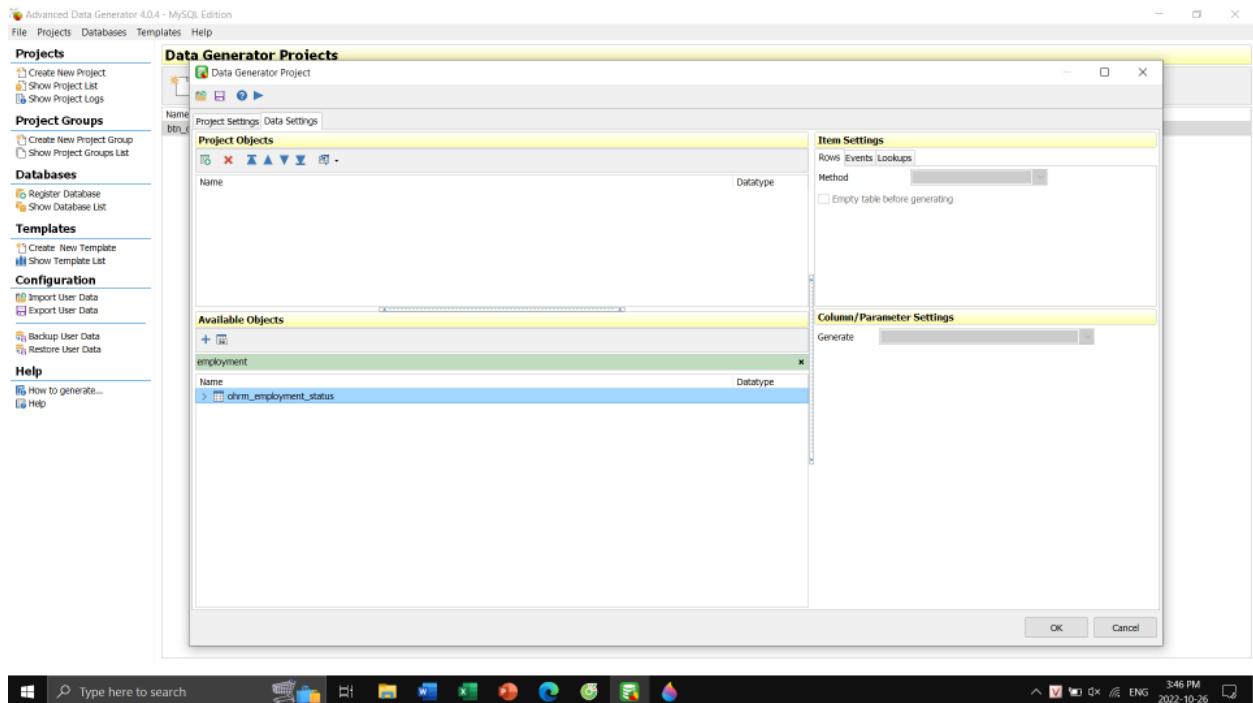


- Tạo project và đặt tên cho nó

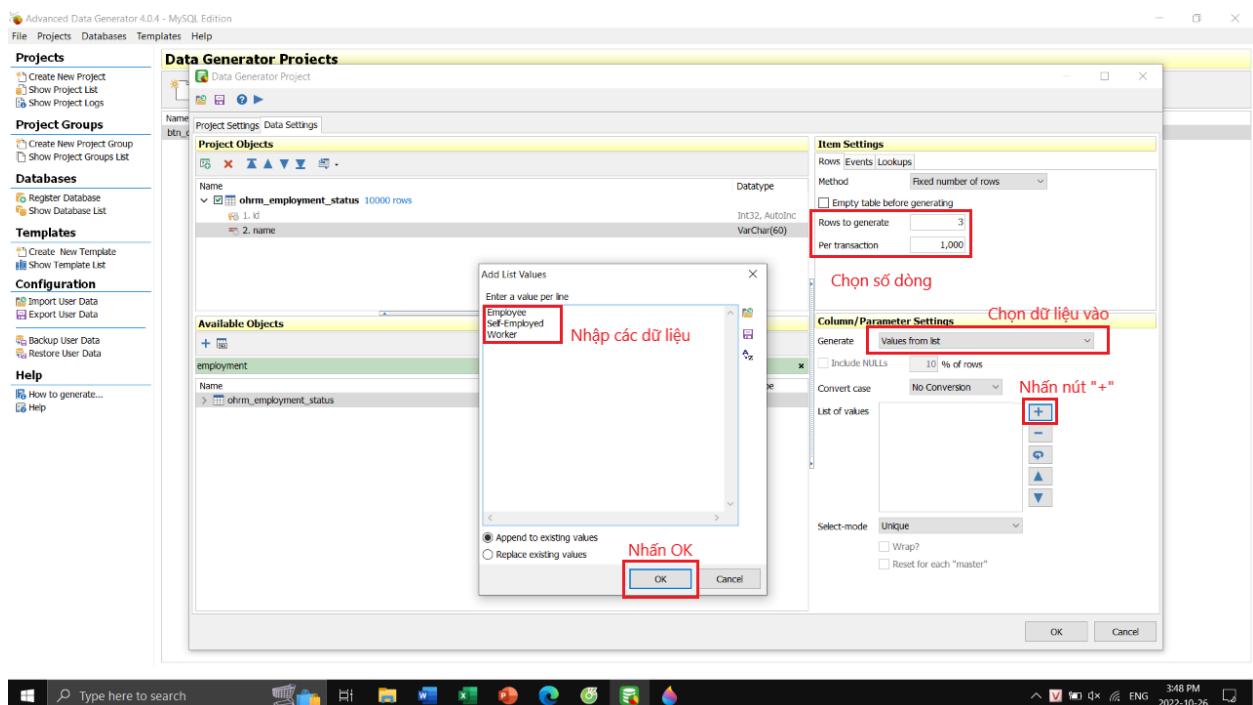


1.3. Tạo dữ liệu cho bảng ohrm_employment_status

- Tìm bảng có tên tương ứng và double click

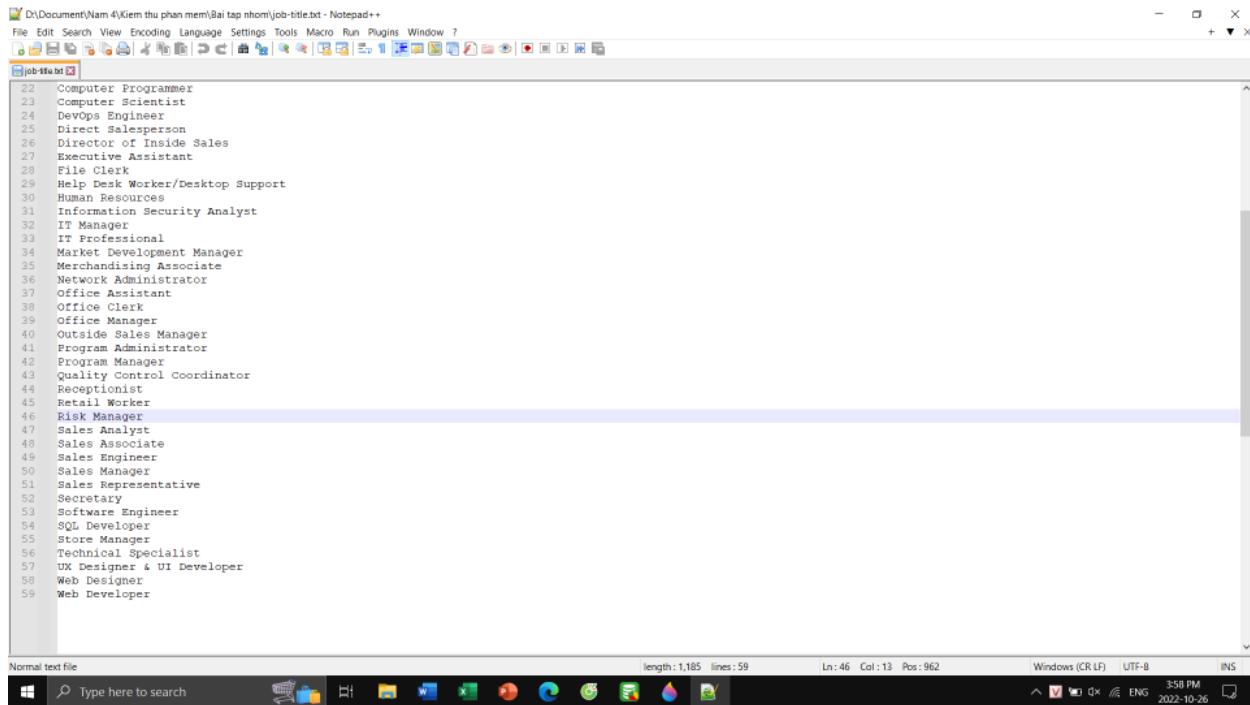


- Tạo dữ liệu cho trường name bằng phương thức Value from list



1.4. Tạo dữ liệu cho bảng ohrm_job_title

- Chuẩn bị dữ liệu là file txt gồm danh sách tên 59 job titles.



D:\Document\Nam 4\Kiem thu phan mem\Bai tap nhom\job-title.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

Job-title.txt

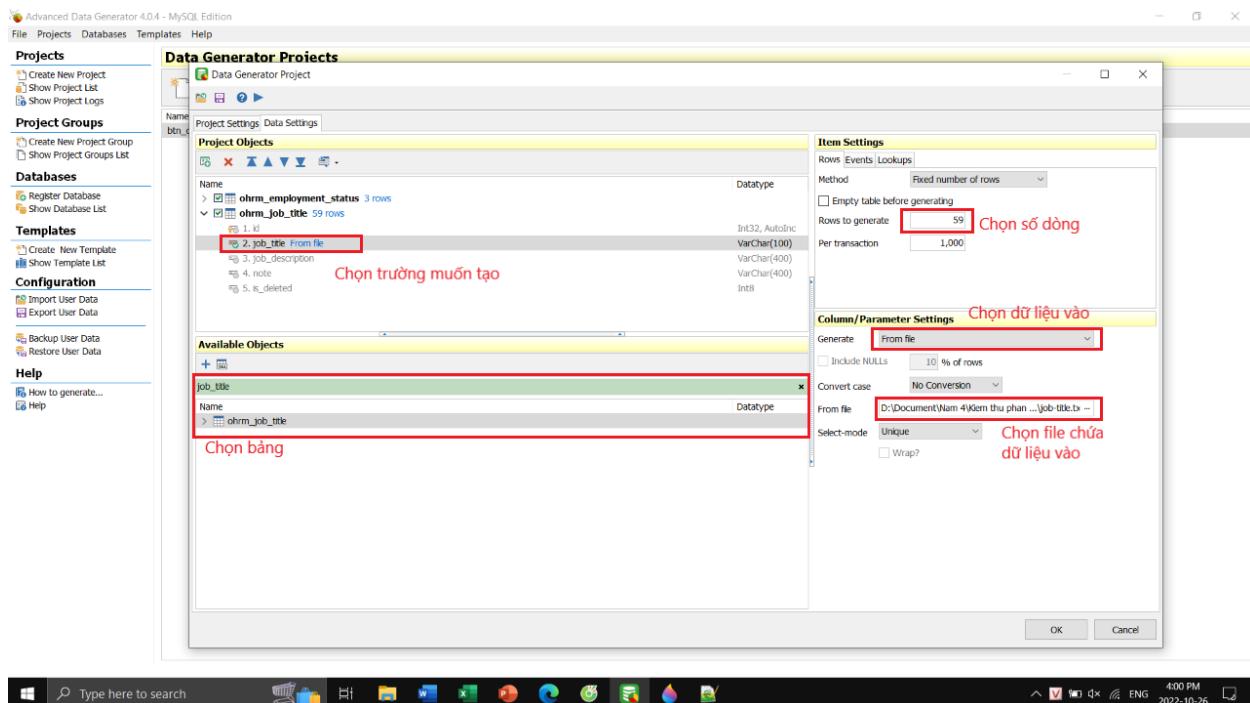
22 Computer Programmer
23 Computer Scientist
24 DevOps Engineer
25 Direct Salesperson
26 Director of Inside Sales
27 Executive Assistant
28 File Clerk
29 Help Desk Worker/Desktop Support
30 Human Resources
31 Information Security Analyst
32 IT Manager
33 IT Professional
34 Market Development Manager
35 Merchandising Associate
36 Network Administrator
37 Office Assistant
38 Office Clerk
39 Office Manager
40 Outside Sales Manager
41 Program Administrator
42 Program Manager
43 Quality Control Coordinator
44 Receptionist
45 Retail Worker
46 Risk Manager
47 Sales Analyst
48 Sales Associate
49 Sales Engineer
50 Sales Manager
51 Sales Representative
52 Secretary
53 Software Engineer
54 SQL Developer
55 Store Manager
56 Technical Specialist
57 UX Designer & UI Developer
58 Web Designer
59 Web Developer

Normal text file length : 1,185 lines : 59 Ln: 46 Col : 13 Pos: 962 Windows (CR LF) UTF-8 358 PM INS

Type here to search

Windows Taskbar: Type here to search, Start, File Explorer, Mail, Photos, OneDrive, Task View, Edge, File, Settings, Control Panel, Task Manager, Power, System, Date/Time

- Tìm và chọn bảng có tên ohrm_job_title để tạo dữ liệu



Advanced Data Generator 4.0.4 - MySQL Edition

File Projects Databases Templates Help

Projects

- Create New Project
- Show Project List
- Show Project Logs

Project Groups

- Create New Project Group
- Show Project Groups List

Databases

- Register Database
- Show Database List

Templates

- Create New Template
- Show Template List

Configuration

- Import User Data
- Export User Data

Backup User Data

Restore User Data

Help

- How to generate...
- Help

Data Generator Projects

Project Settings Data Settings

Project Objects

- ohrm_employment_status 3 rows
- ohrm_job_title 59 rows
 - 2. job_title From file
 - 3. job_description
 - 4. note
 - 5. is_deleted

Available Objects

- job_title
 - Name: job_title Datatype: Int32, AutoInc
 - Name: ohrm_job_title Datatype: VarChar(100)
 - Name: job_description Datatype: VarChar(400)
 - Name: note Datatype: VarChar(400)
 - Name: is_deleted Datatype: Int8

Item Settings

Method: Fixed number of rows

Rows to generate: 59 Chọn số dòng

Per transaction: 1,000

Column/Parameter Settings

Generate: From file

Include NULLS: 10 % of rows

Convert case: No Conversion

From file: D:\Document\Nam 4\Kiem thu phan ...job-title.txt Chọn file chứa dữ liệu vào

Select-mode: Unique

Wrap?: Chọn file chứa dữ liệu vào

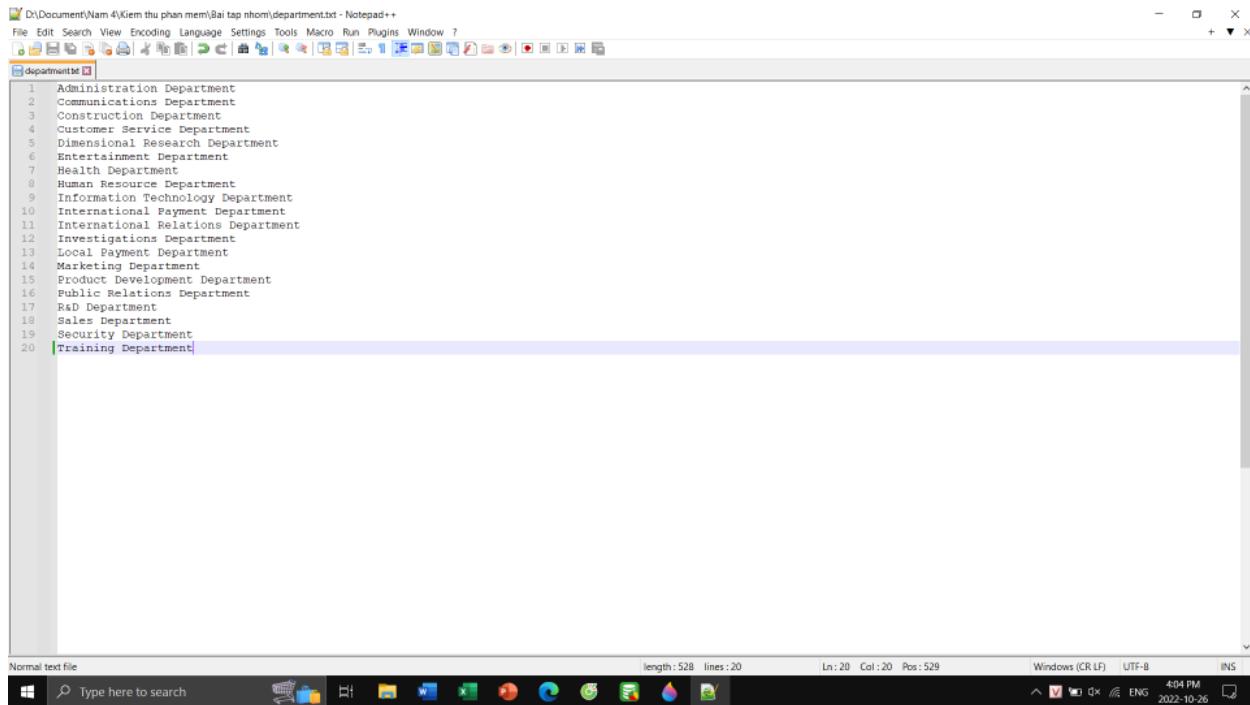
OK Cancel

Type here to search

Windows Taskbar: Type here to search, Start, File Explorer, Mail, Photos, OneDrive, Task View, Edge, File, Settings, Control Panel, Task Manager, Power, System, Date/Time

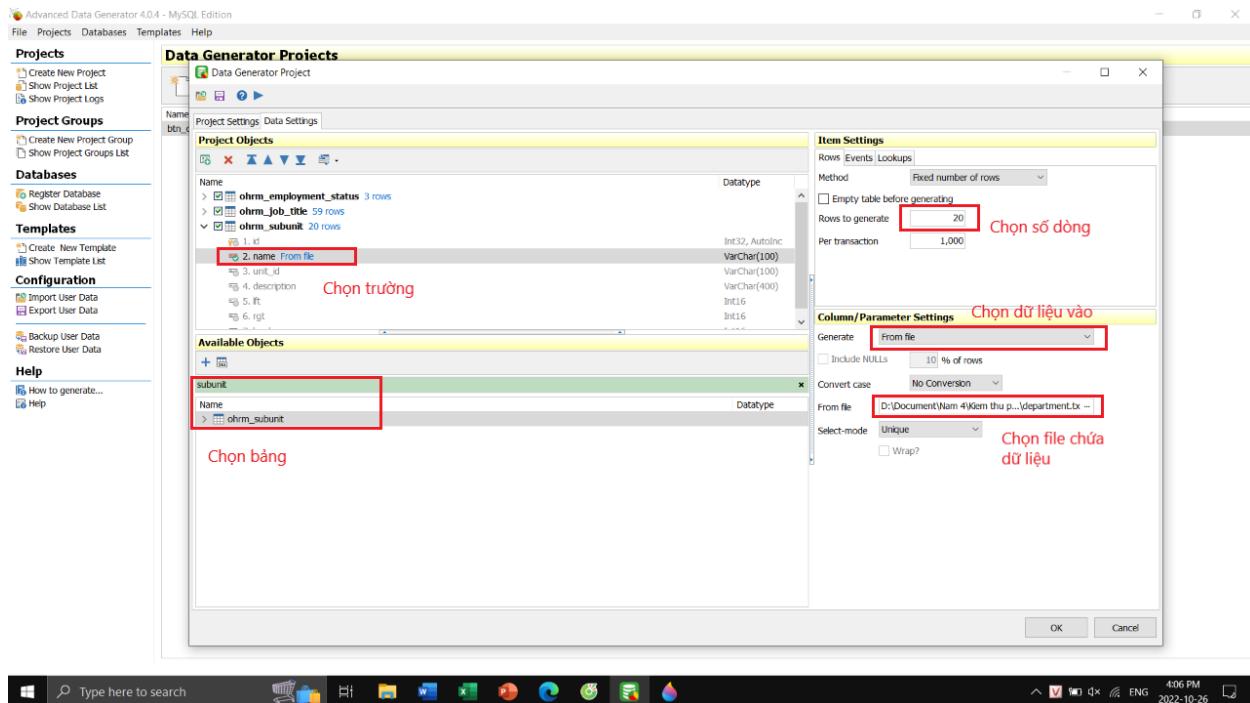
1.5. Tạo dữ liệu cho các phòng ban (ohrm_subunit)

- Chuẩn bị dữ liệu là file txt gồm danh sách tên của 20 phòng ban



D:\Document\Nam 4\Kiem thu phan mem\Bai tap nhom\department.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
department.txt
1 Administration Department
2 Communications Department
3 Construction Department
4 Customer Service Department
5 Dimensional Research Department
6 Entertainment Department
7 Health Department
8 Human Resource Department
9 Information Technology Department
10 International Payment Department
11 International Relations Department
12 Investigations Department
13 Local Payment Department
14 Marketing Department
15 Product Development Department
16 Public Relations Department
17 R&D Department
18 Sales Department
19 Security Department
20 Training Department

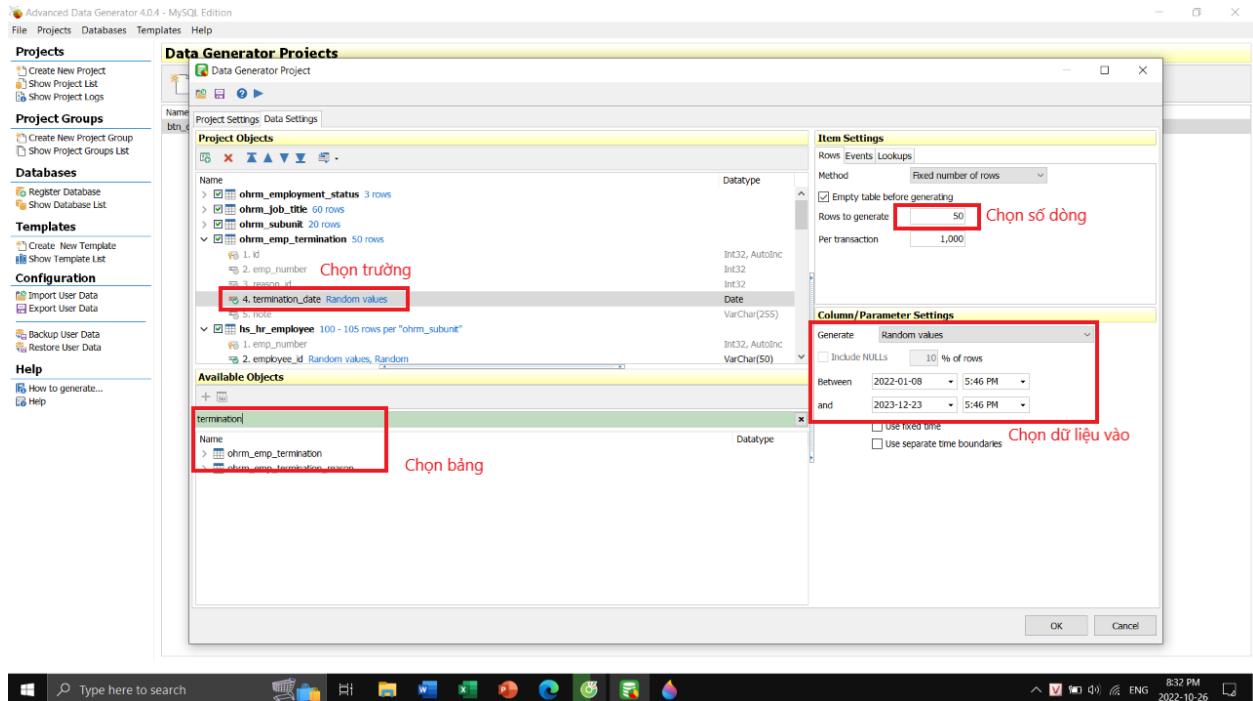
- Tìm và chọn bảng có tên ohrm_job_title để tạo dữ liệu



Advanced Data Generator 4.0.4 - MySQL Edition
File Projects Databases Templates Help
Data Generator Projects
Projects
Create New Project Show Project List Show Project Logs
Project Groups
Create New Project Group Show Project Groups List
Templates
Create New Template Show Template List
Configuration
Import User Data Export User Data
Backup User Data Restore User Data
Help
How to generate... Help
Item Settings
Name: ohrm_subunit
Datatype: Int32, Automic
Datatype: VarChar(100)
Method: Fixed number of rows
Rows to generate: 20 Chọn số dòng
Per transaction: 1,000
Column/Parameter Settings
Generate: From file Chọn dữ liệu vào
Include NULLS: 10 % of rows
Convert case: No Conversion
From file: D:\Document\Nam 4\Kiem thu p...department.txt Chọn file chứa dữ liệu
Select-mode: Unique
Wrap? Chọn file chứa dữ liệu
Available Objects
subunit
Name: ohrm_subunit
Datatype: Int32, Automic
Datatype: VarChar(100)
Datatype: VarChar(400)
Datatype: Int16
Datatype: Int16
Chọn trường
2. name from file
3. unit_id
4. description
5. ft
6. rgt
Chọn bảng

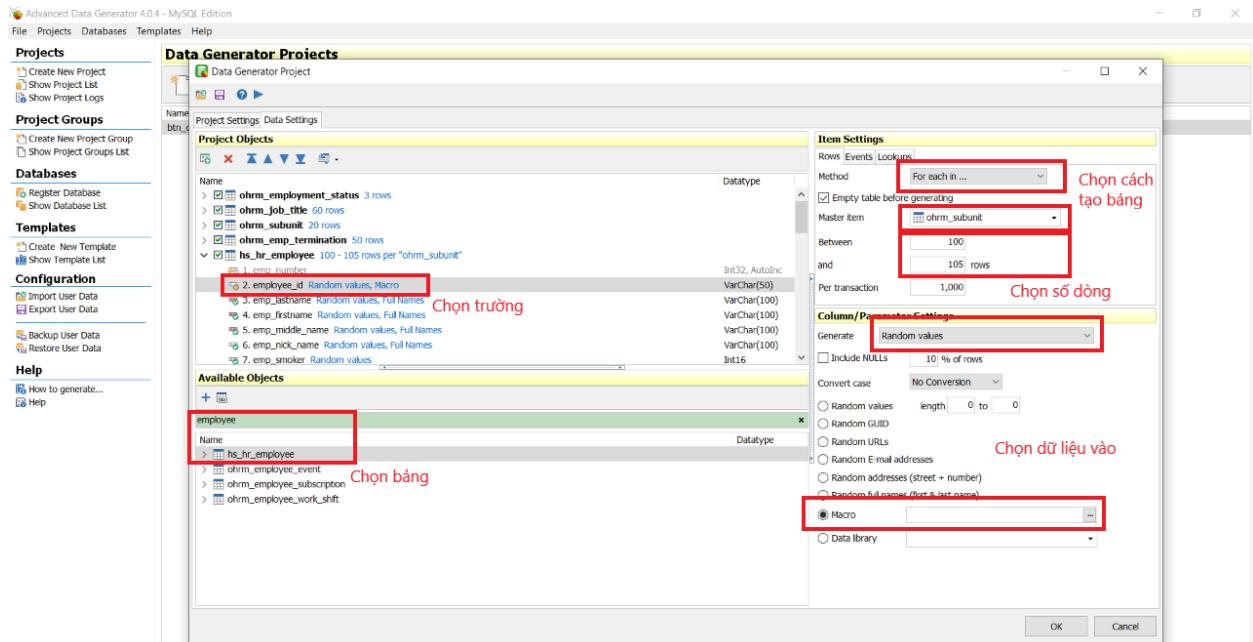
1.6. Tạo dữ liệu cho bảng ohrm_emp_termination

- Tìm và chọn bảng có tên ohrm_emp_termination để tạo dữ liệu.

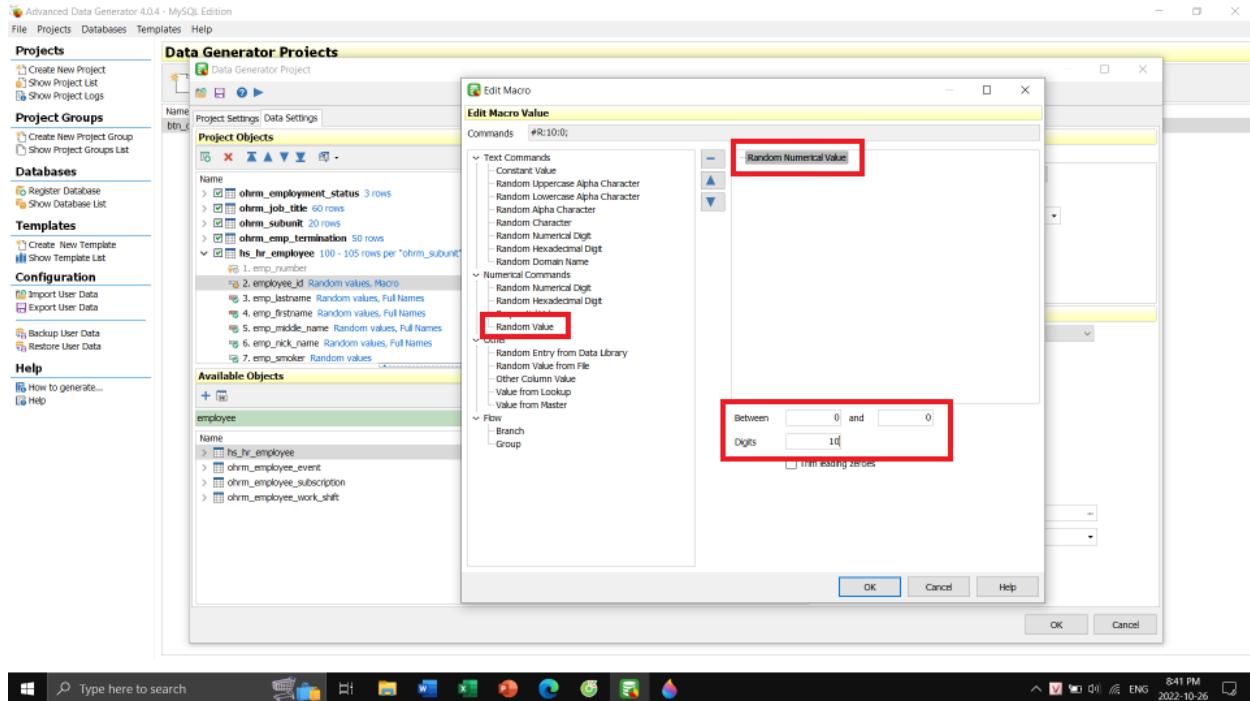


1.7. Tạo dữ liệu của nhân viên cho mỗi phòng ban (hr_hs_employee)

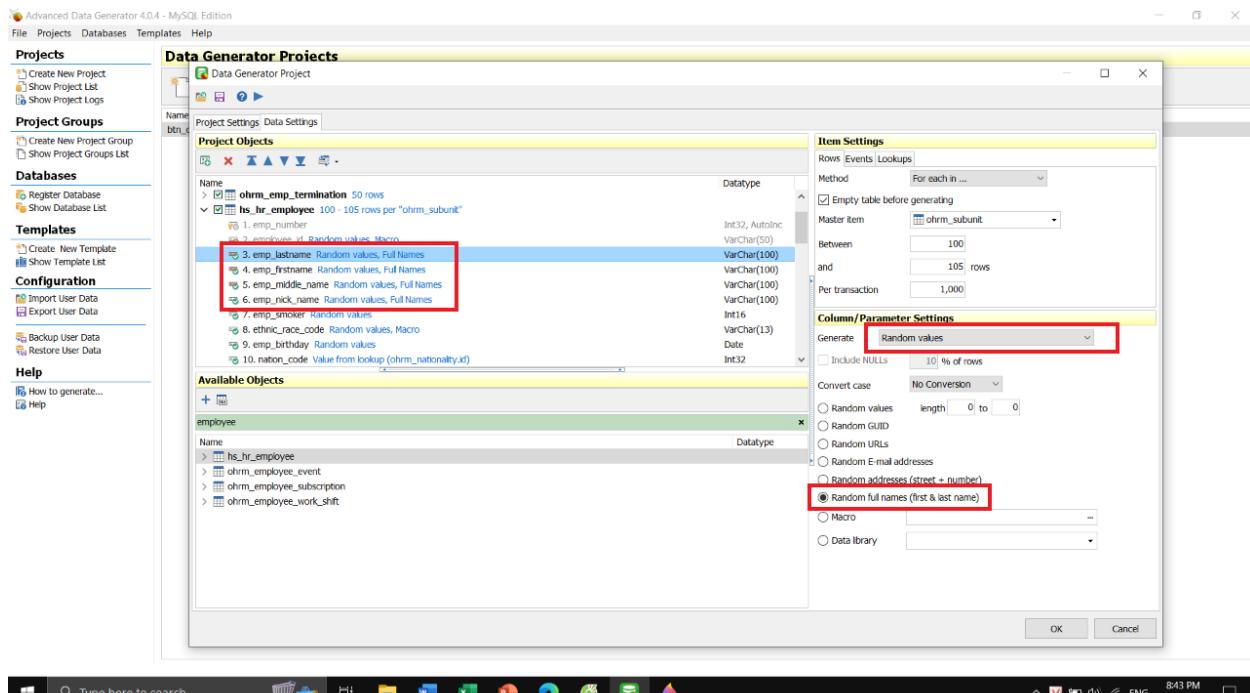
- Tìm và chọn bảng có tên hr_hs_employee để tạo dữ liệu
- Trong Items Settings → Methods, chọn For each in... để tạo dữ liệu với mỗi dòng trong bảng phòng ban (ohrm_subunit)



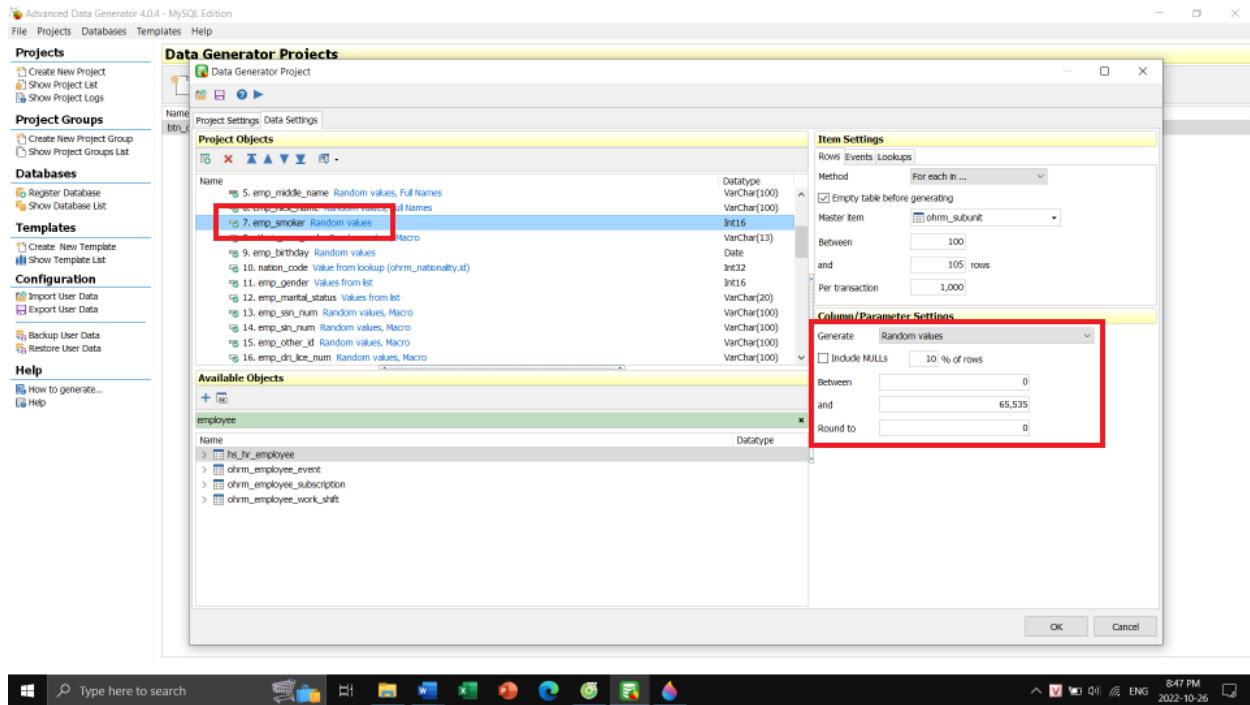
- Chọn kiểu dữ liệu muốn tạo random



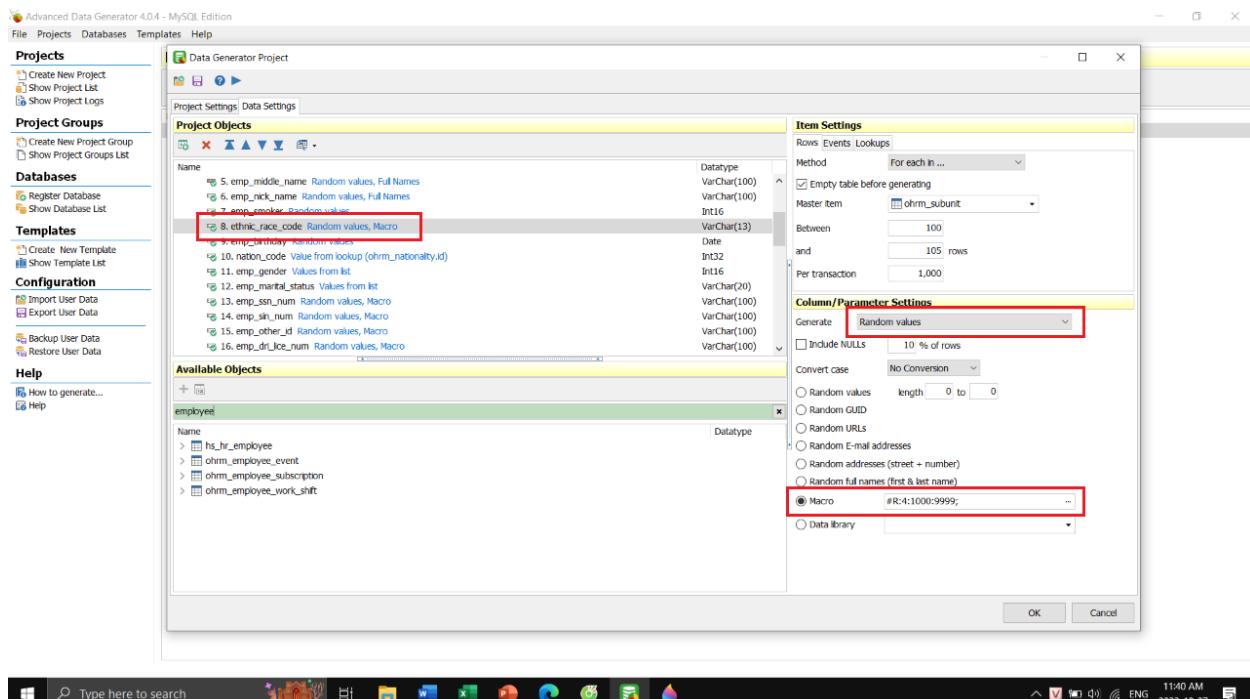
- Tạo dữ liệu cho các trường lastname, firstname,... bằng Random Values → Random full names



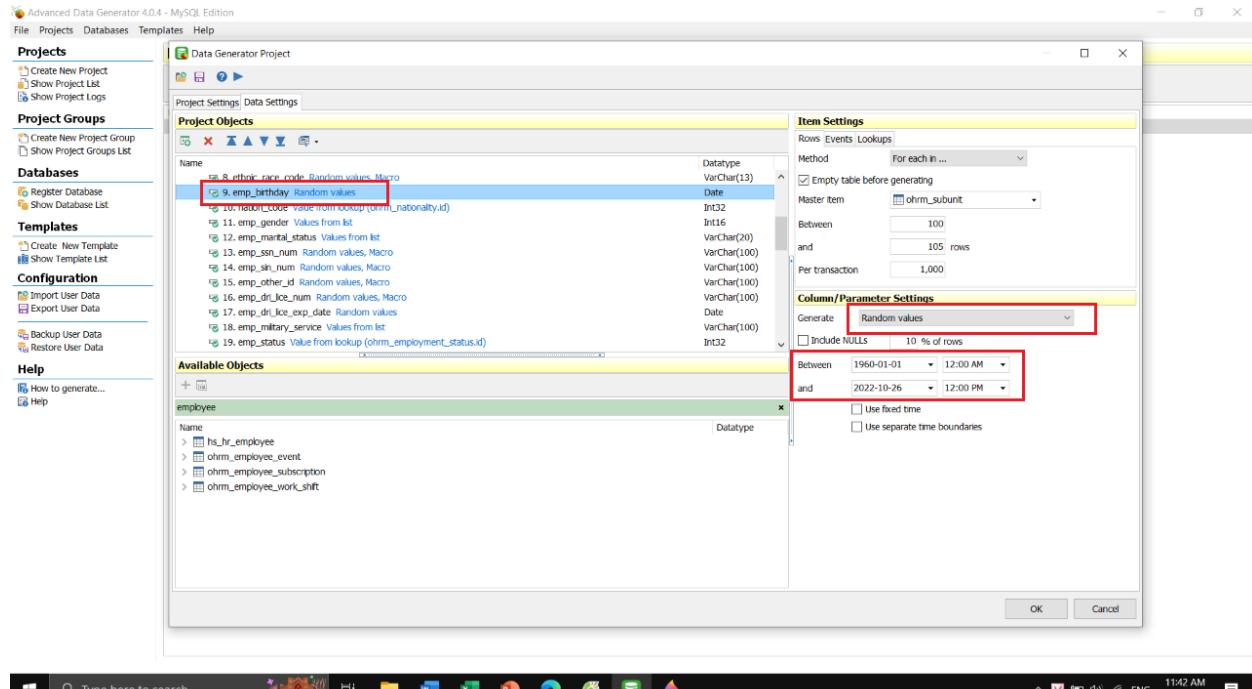
- Tạo dữ liệu cho trường smoker bằng cách tạo một số nguyên ngẫu nhiên



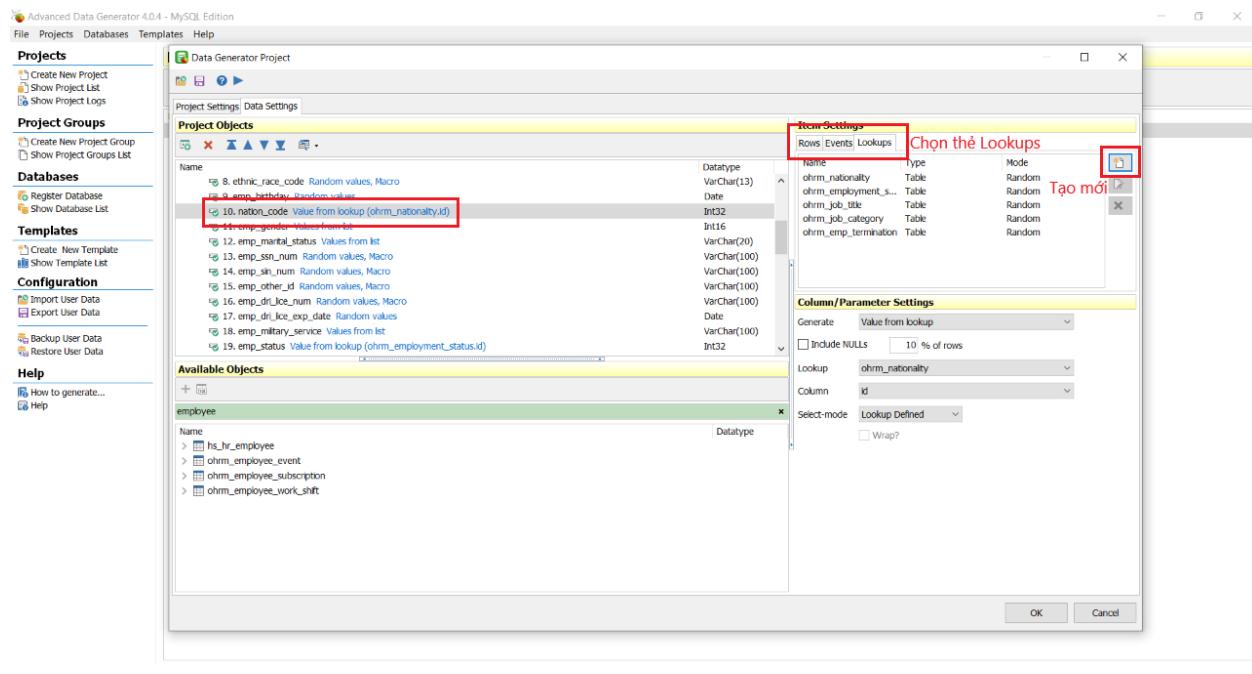
- Tạo dữ liệu cho trường ethnic_race_code bằng cách tương tự trường employee_id



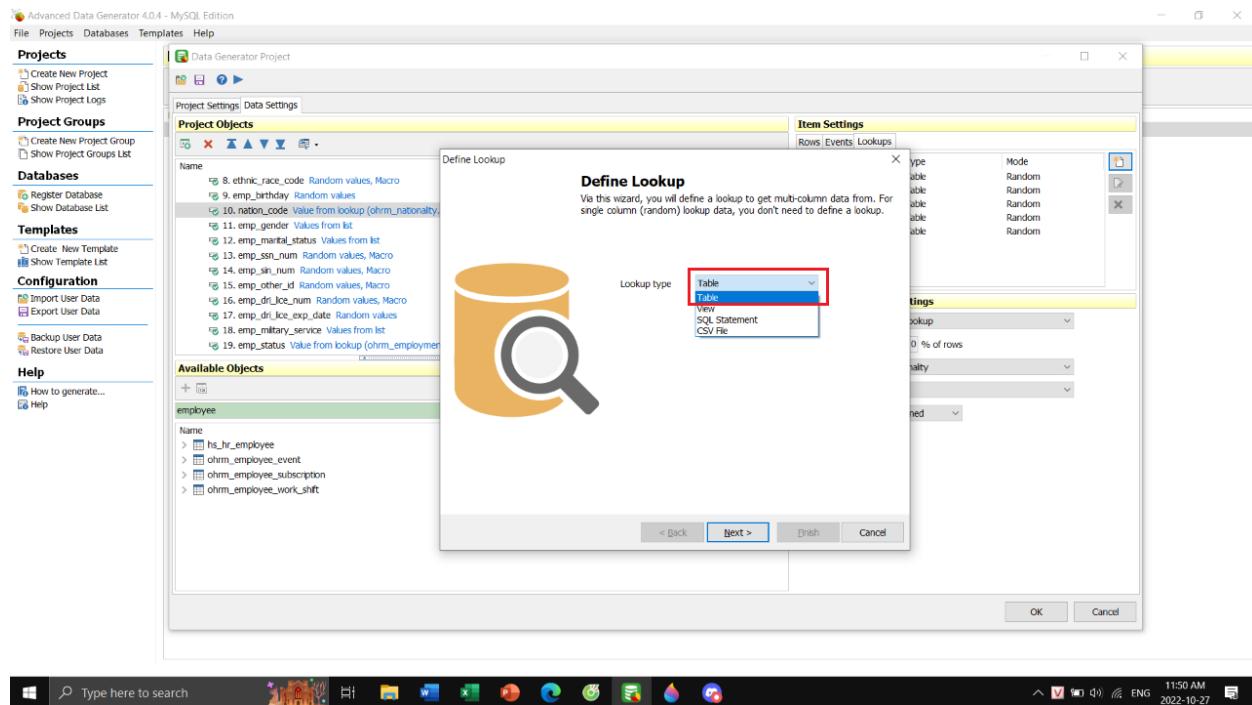
- Tạo dữ liệu cho trường emp_birthday bằng cách tạo ngẫu nhiên 1 ngày trong khoảng thời gian nào đó



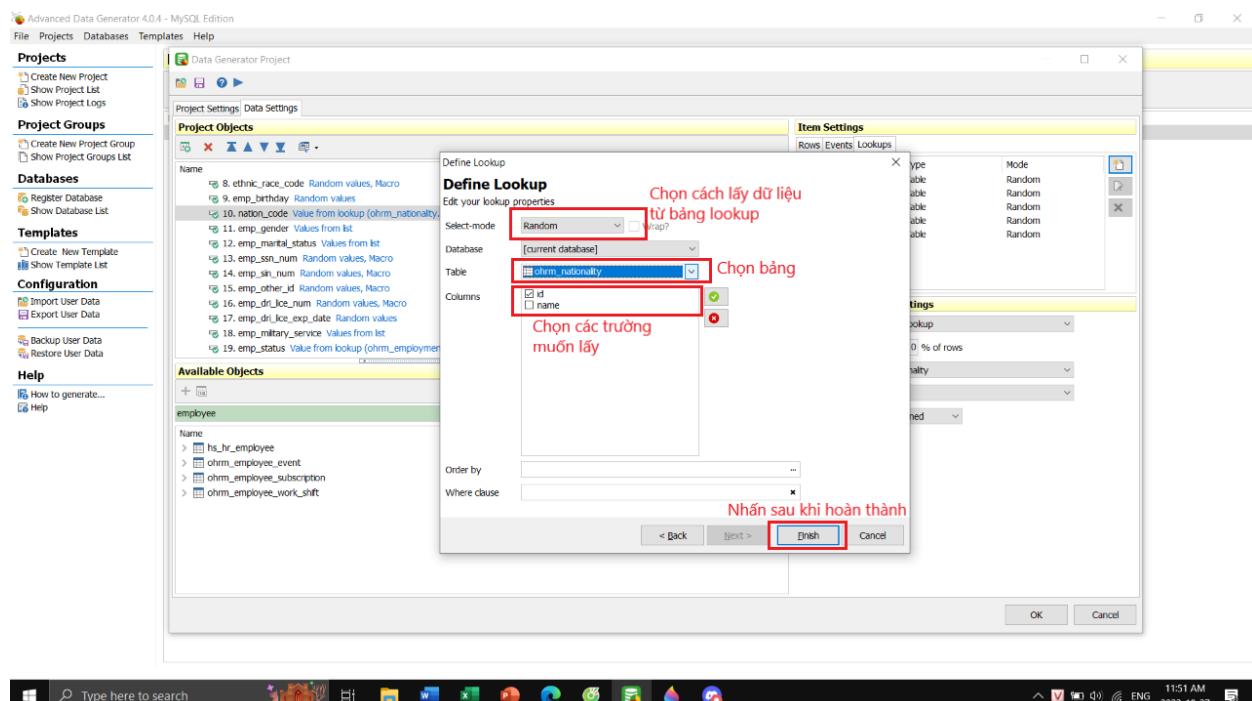
- Để tạo dữ liệu cho trường nation_code, trước tiên ta phải định nghĩa một lookup cho trường này, vì trường này có khóa ngoại đến bảng ohrm_nationality, nên nếu tạo ngẫu nhiên sẽ gây lỗi.



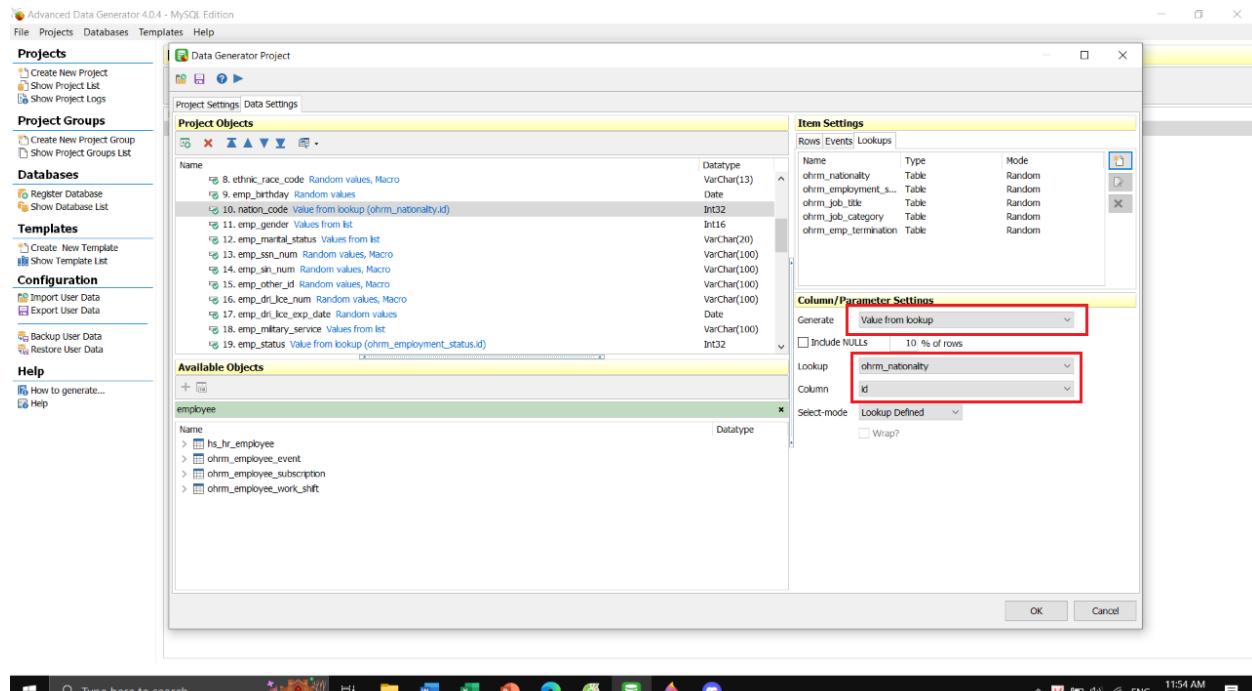
- Ta chọn loại lookup muốn tạo là Table



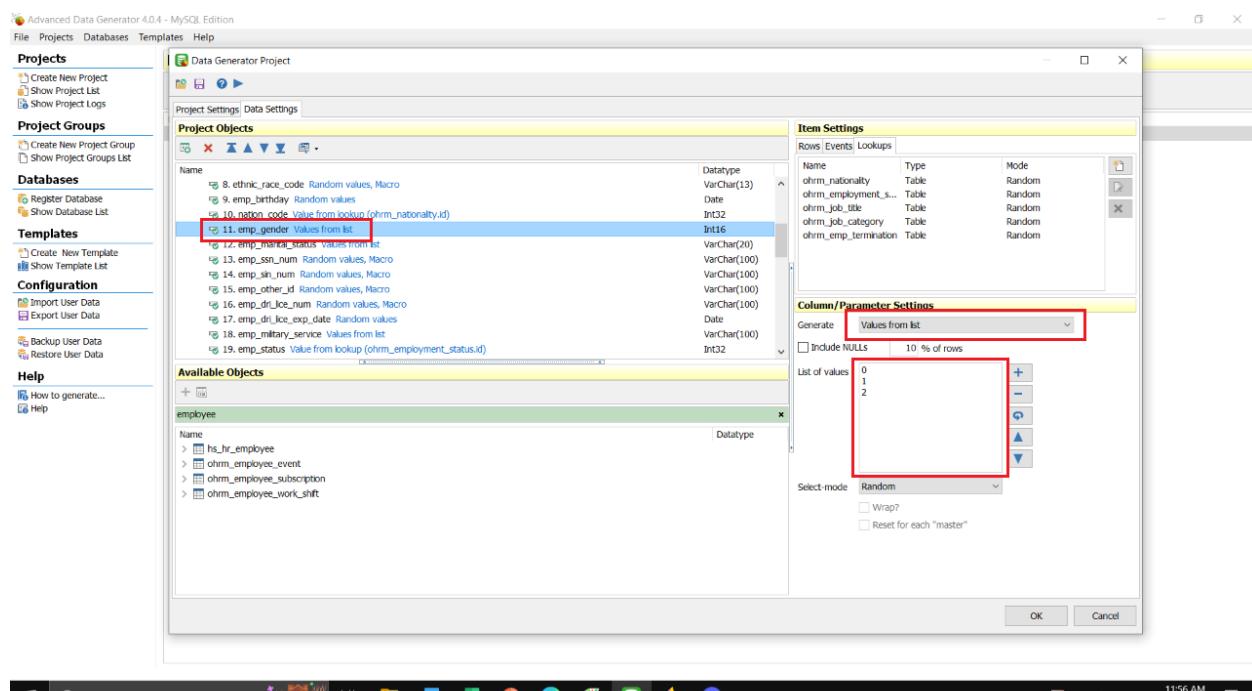
- Sau đó chọn bảng và nhấn Finish



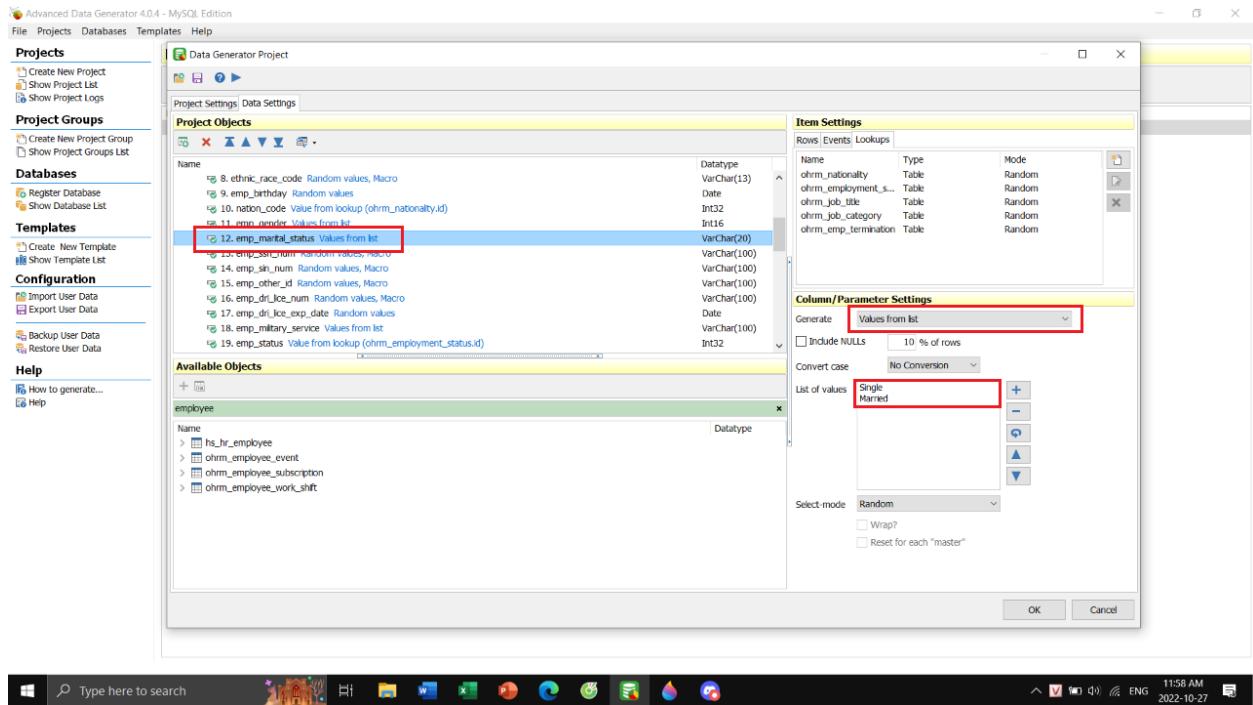
- Ở thuộc tính nation_code, ta chọn Value from lookup, chọn lookup đã tạo và trường muốn thực hiện lookup



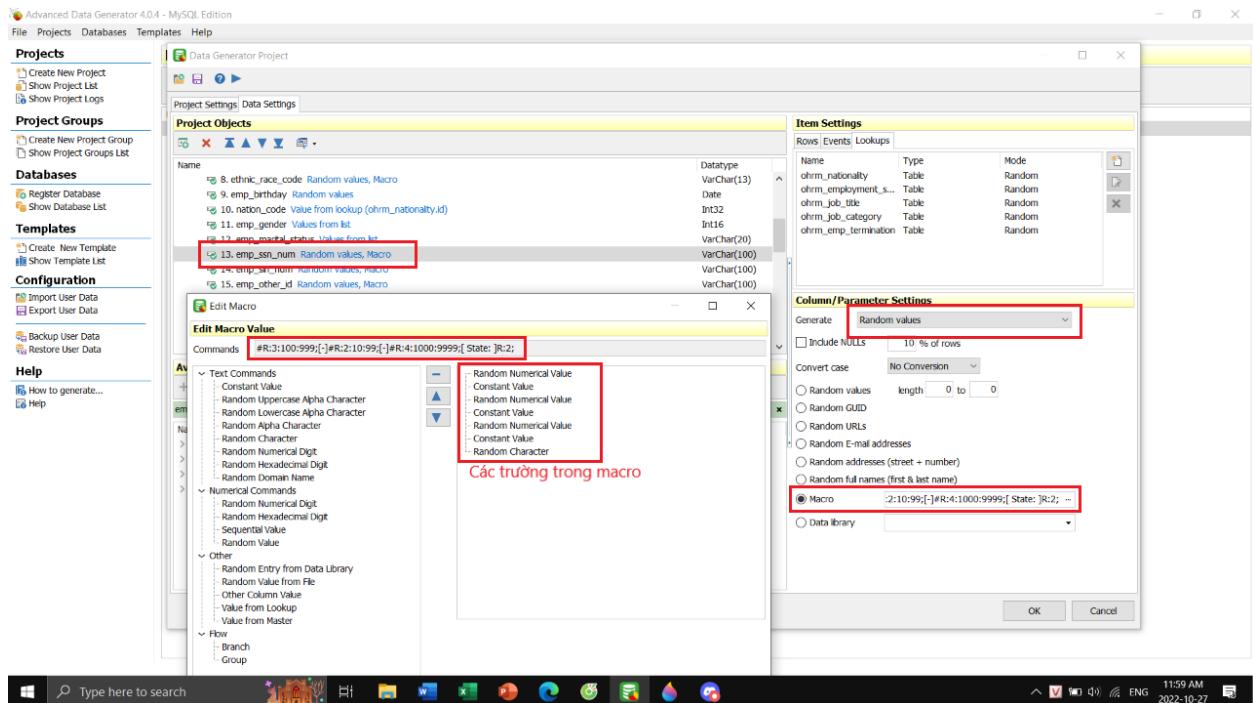
- Tạo dữ liệu cho trường emp_gender bằng phương thức Value from list



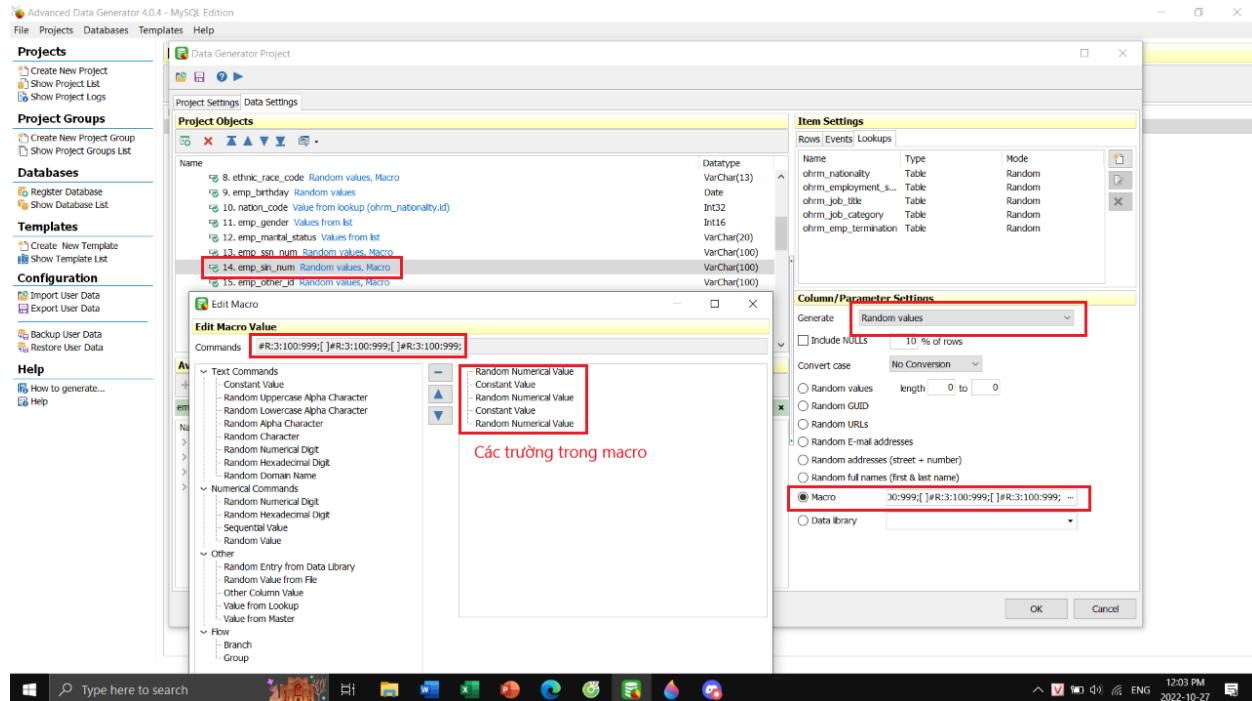
- Tạo dữ liệu cho trường emp_marital_status bằng phương thức Value from list



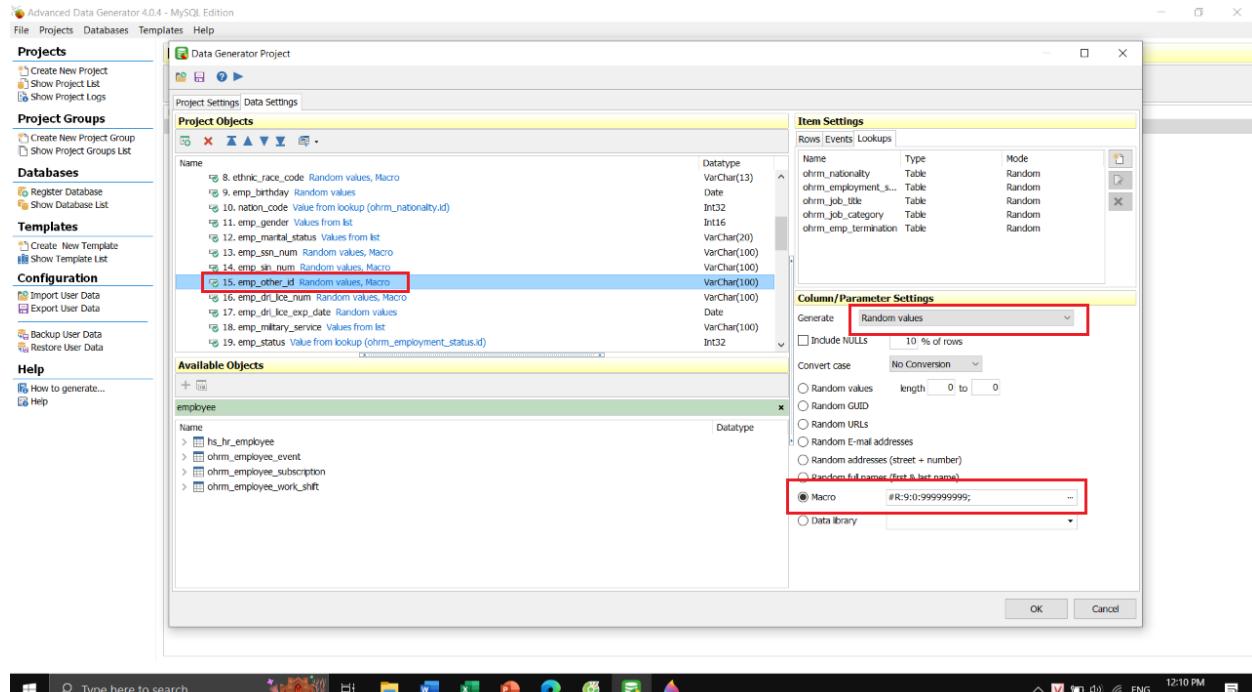
- Tạo dữ liệu cho trường emp_ssn_num bằng cách dùng macro để tạo ngẫu nhiên chuỗi các số và kí tự theo mẫu XXX-XX-XXXX State: YY. Trong đó X là các chữ số, Y là các chữ cái



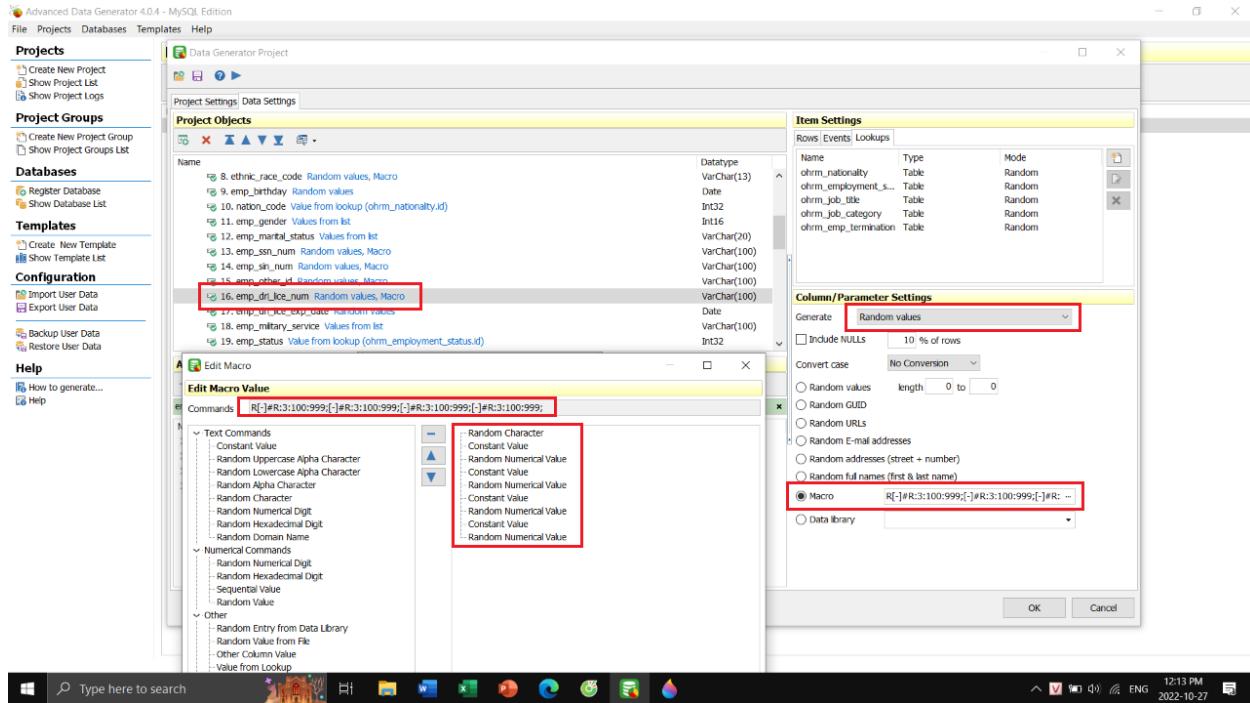
- Tạo dữ liệu cho trường emp_sin_num bằng cách dùng macro để tạo ngẫu nhiên chuỗi các số và kí tự theo mẫu XXX XXX XXX. Trong đó X là các chữ số



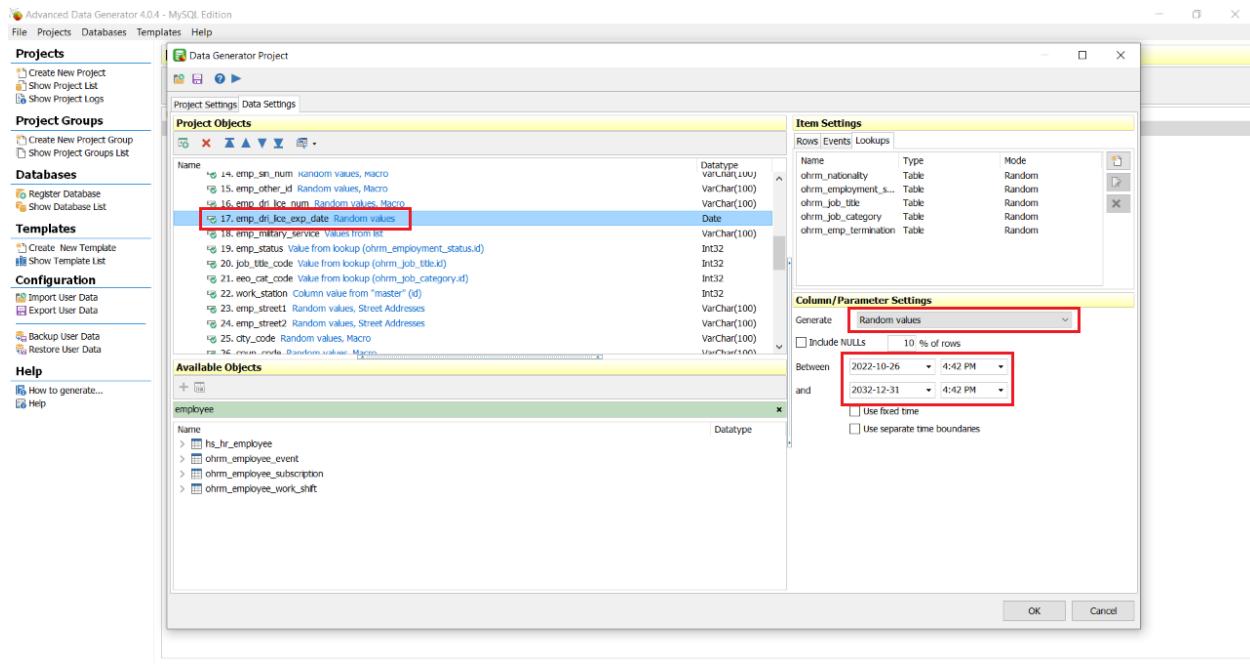
- Tạo dữ liệu cho trường emp_other_id là dãy gồm 9 chữ số



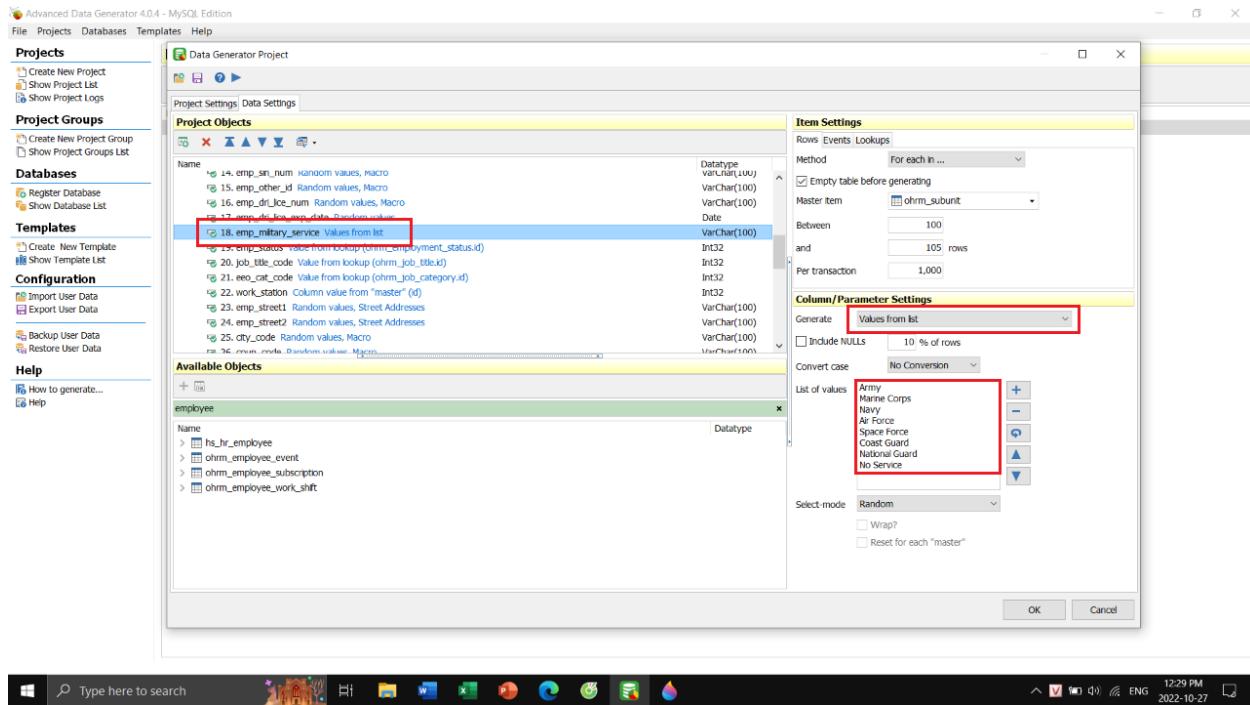
- Tạo dữ liệu cho trường emp_dri_lice_num bằng cách dùng macro để tạo ngẫu nhiên chuỗi các số và kí tự theo mẫu Y-XXX-XXX-XXX-XXX. Trong đó X là các chữ số, Y là các chữ cái.



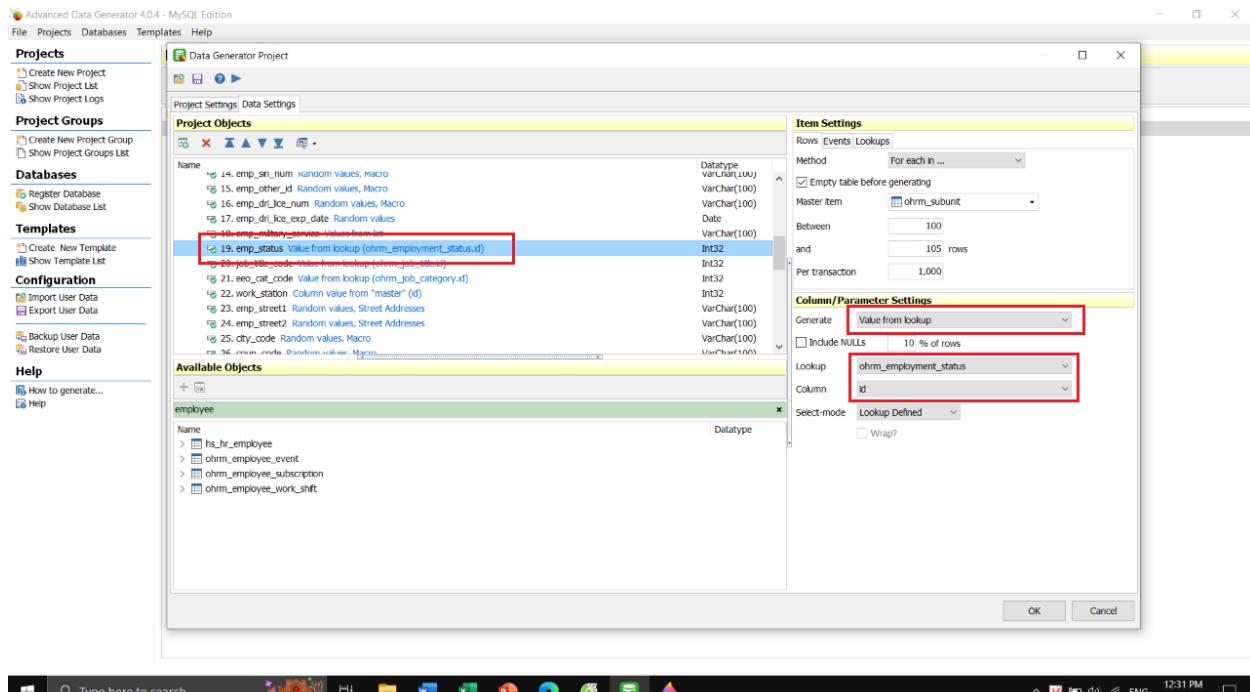
- Tạo dữ liệu cho trường emp_dri_lice_exp_date bằng cách tạo ngẫu nhiên 1 ngày trong khoảng thời gian nào đó



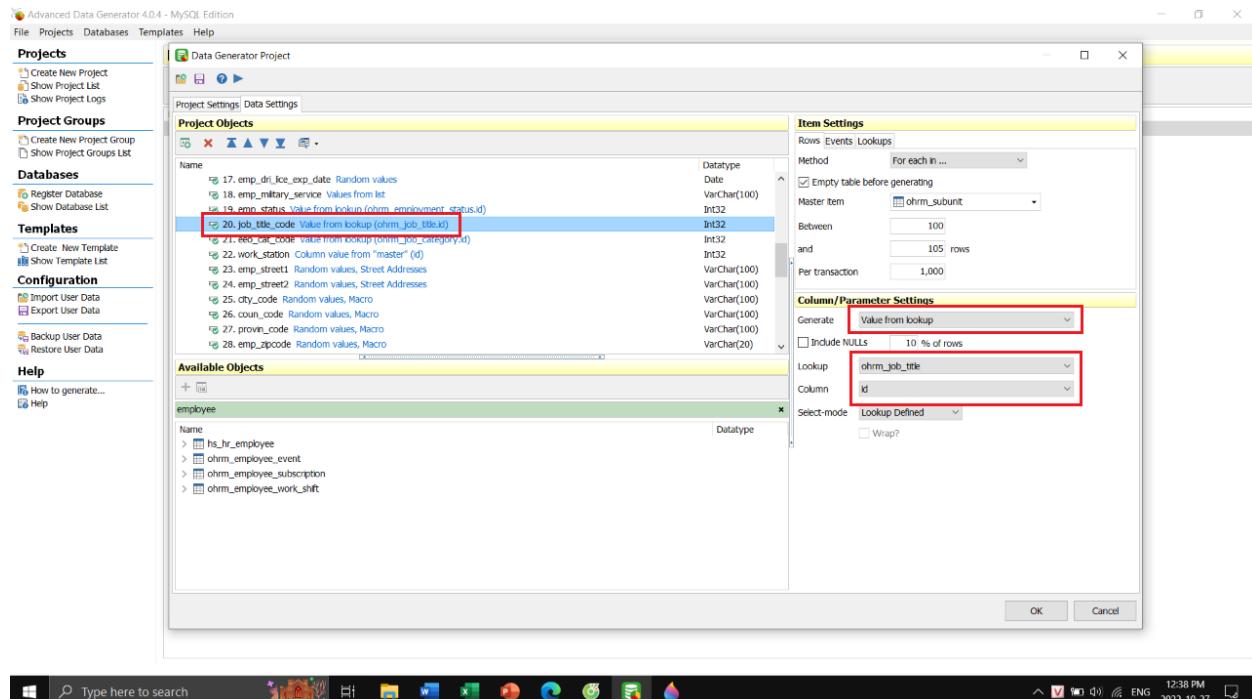
- Tạo dữ liệu cho trường emp_military_service bằng phương thức Value from list



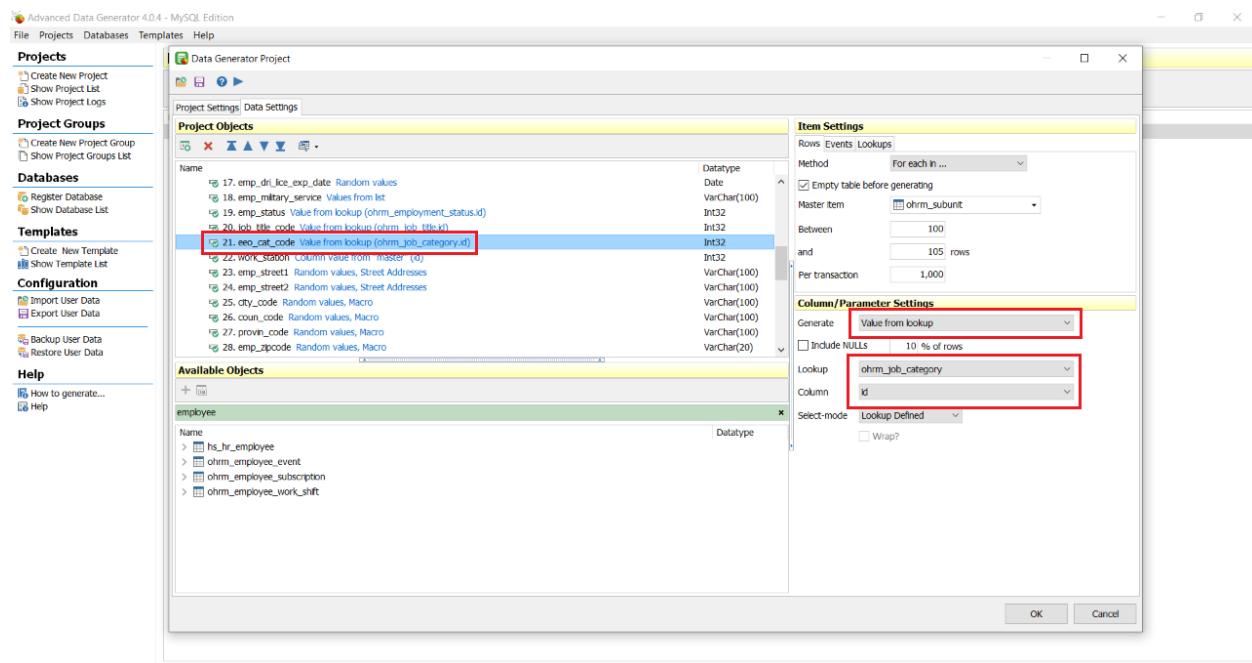
- Tạo dữ liệu cho trường emp_status bằng cách tạo lookup tới trường id trong bảng ohrm_employee_status ta đã tạo trước đó do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên



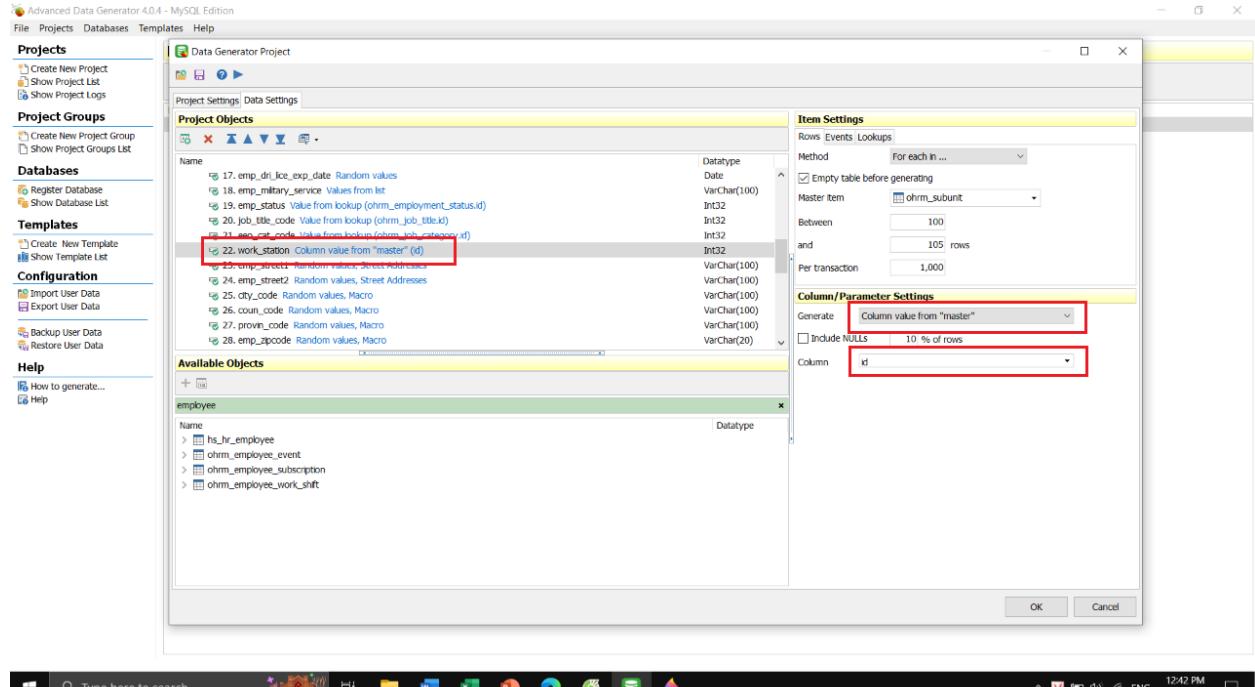
- Tạo dữ liệu cho trường job_title_code bằng cách tạo lookup tới trường id trong bảng ohrm_job_title ta đã tạo trước đó do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên



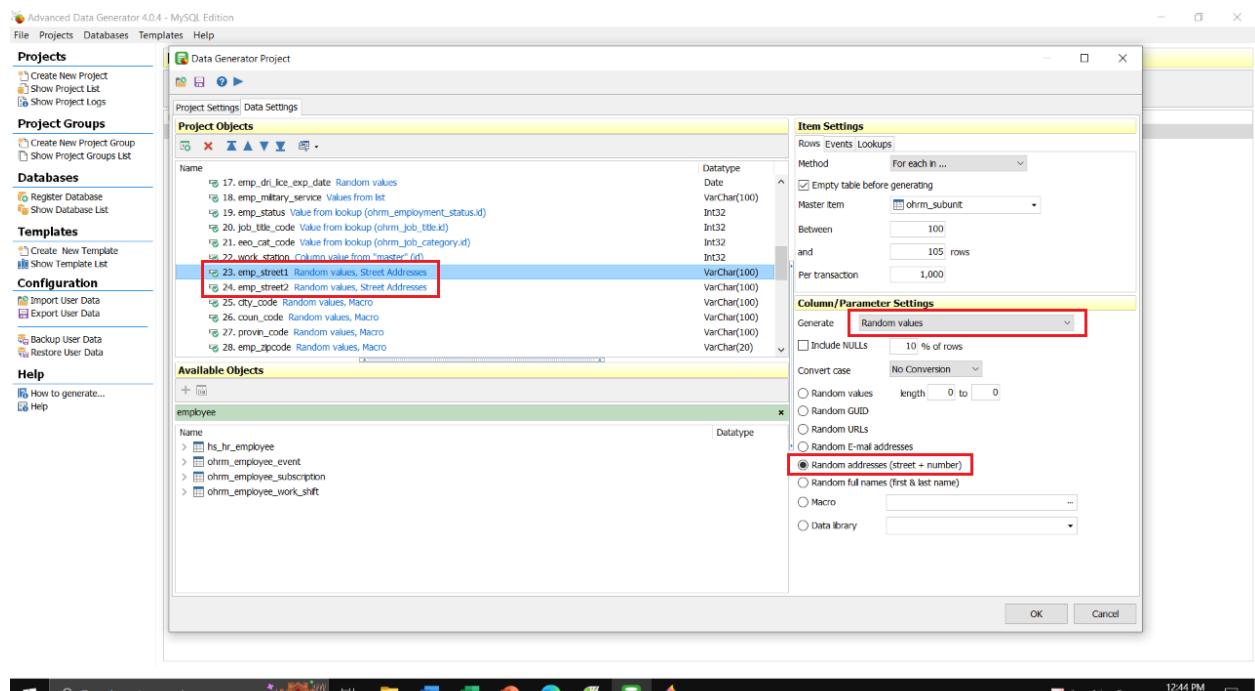
- Tạo dữ liệu cho trường eeo_cat_code bằng cách tạo lookup tới trường id trong bảng ohrm_job_category do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên



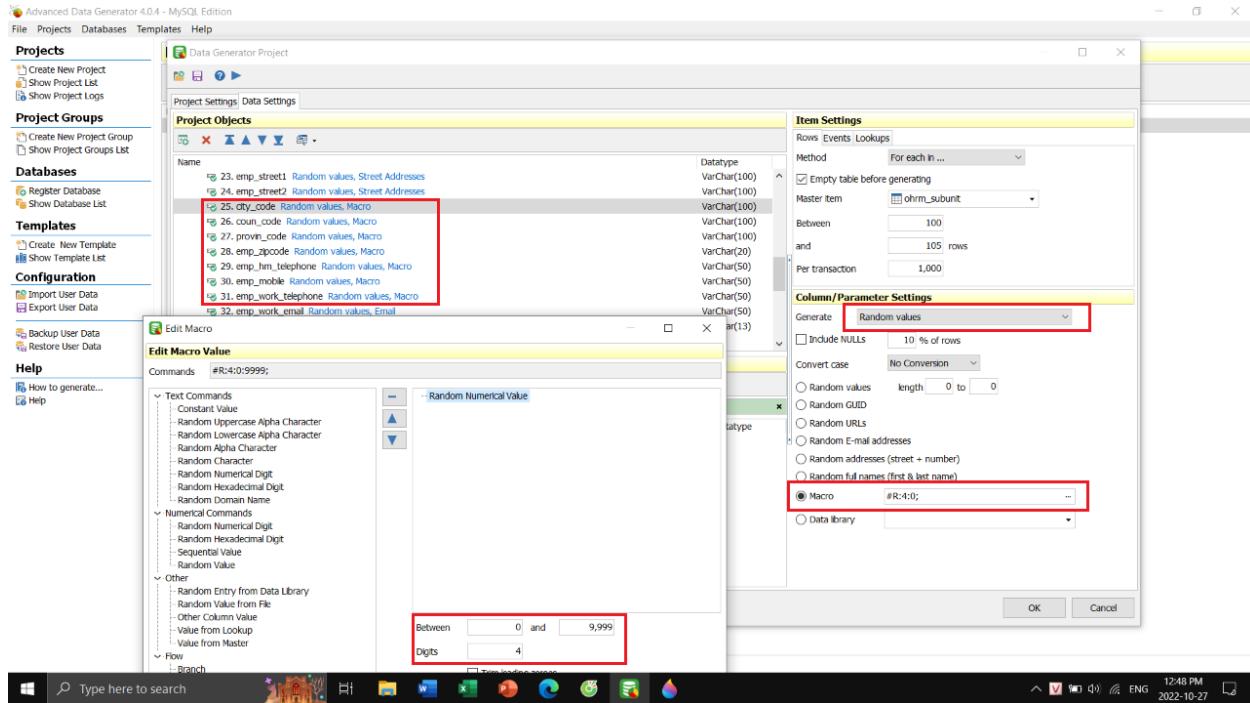
- Tạo dữ liệu cho trường work_station bằng phương thức Column value from “master” để lấy id của phòng ban tương ứng



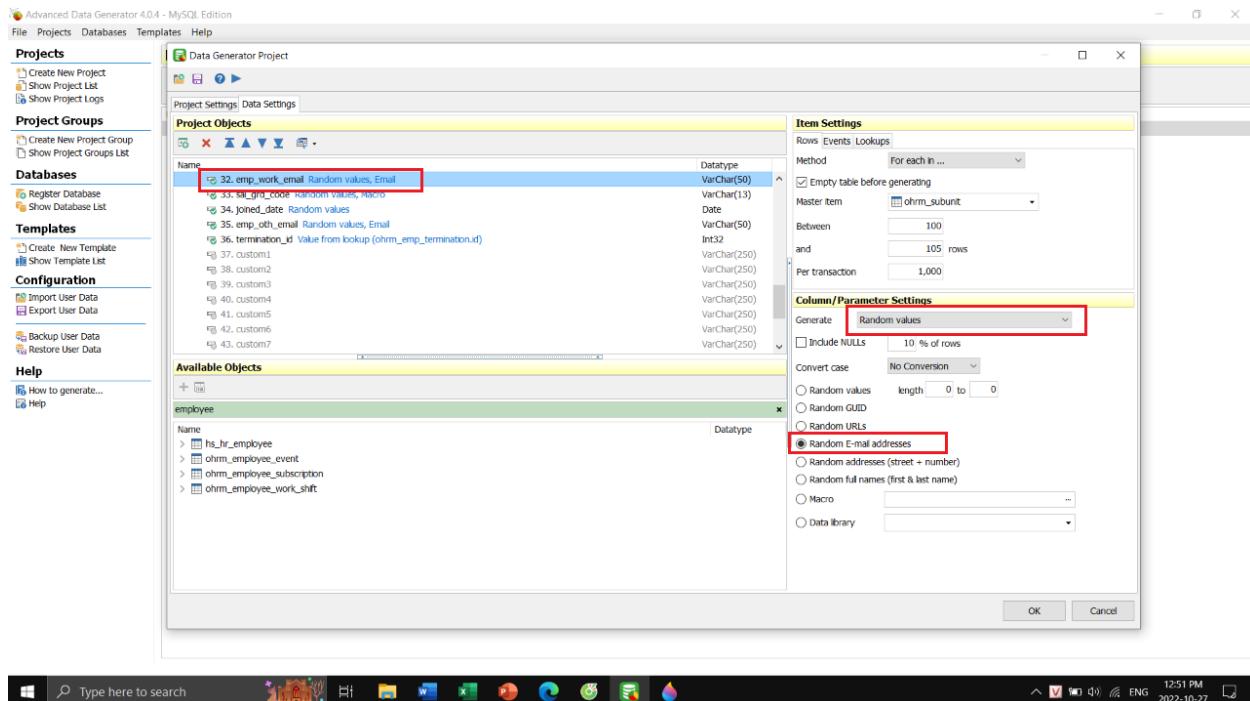
- Tạo dữ liệu cho các trường emp_street1 và emp_street2 bằng Random addresses



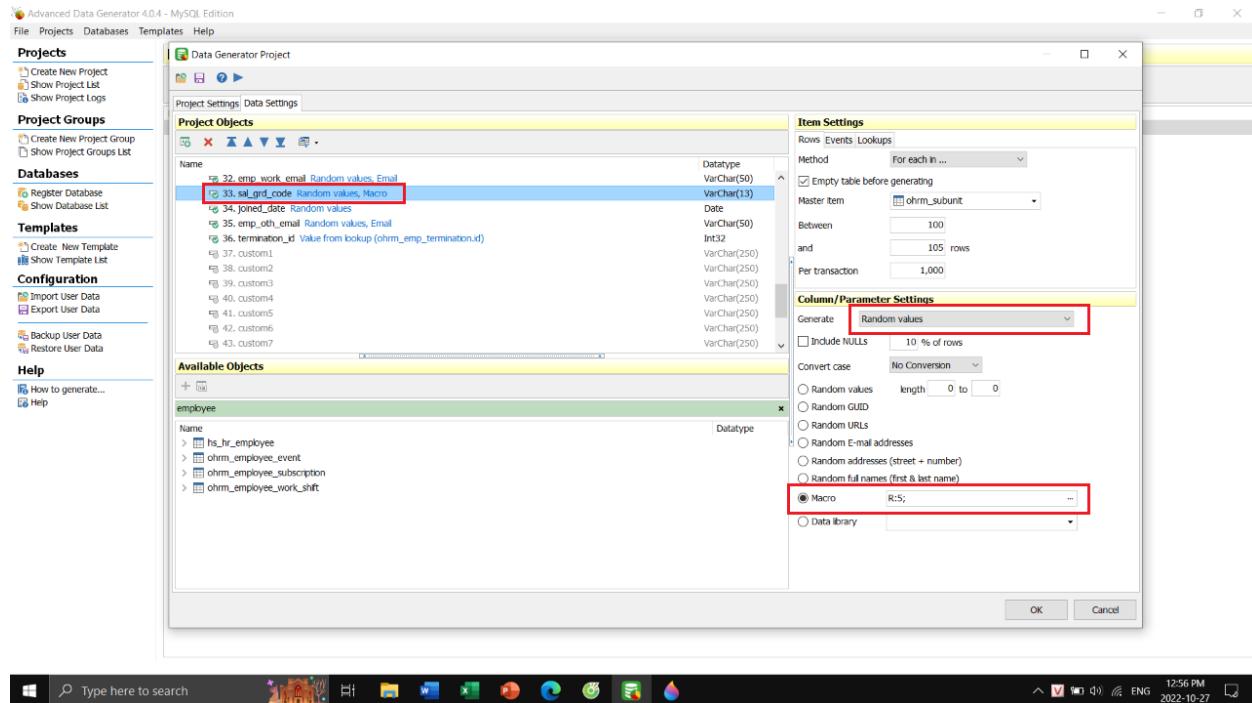
- Tạo dữ liệu cho các trường city_code, coun_code, provin_code là một dãy ngẫu nhiên 4 chữ số, trường emp_zipcode có 6 chữ số, emp_hm_telephone, emp_mobile, emp_work_telephone là dãy gồm 7 chữ số



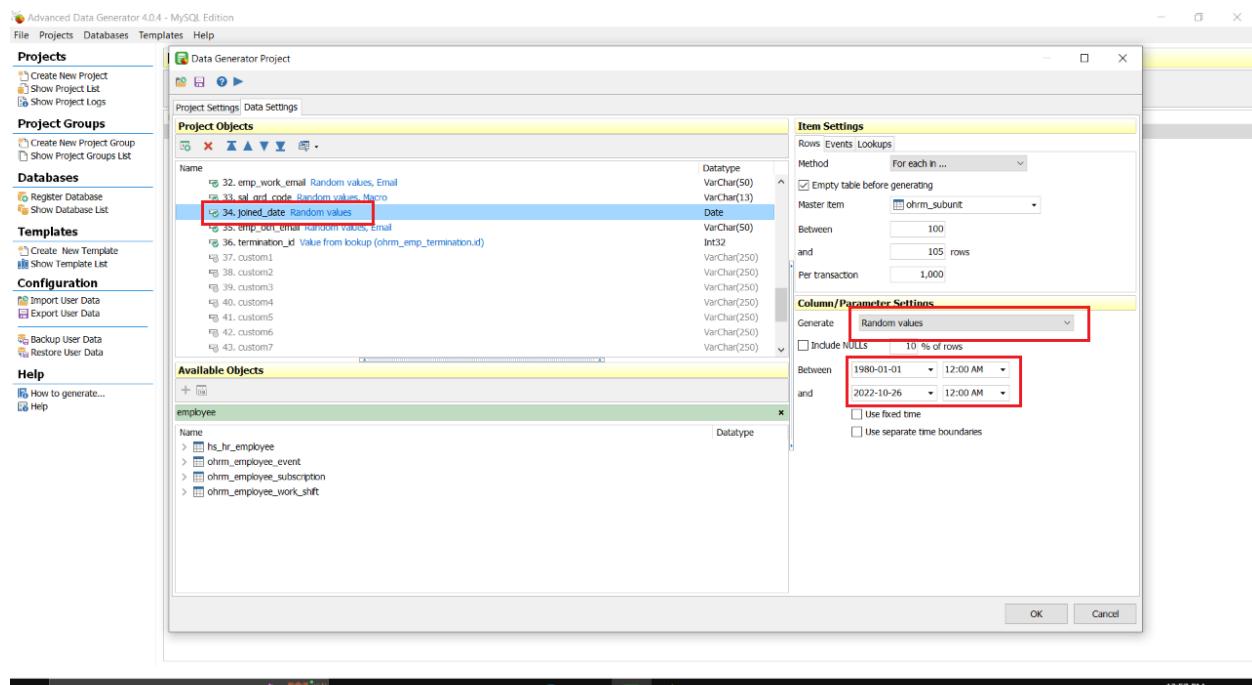
- Tạo dữ liệu cho trường emp_work_email bằng Random e-mail addresses



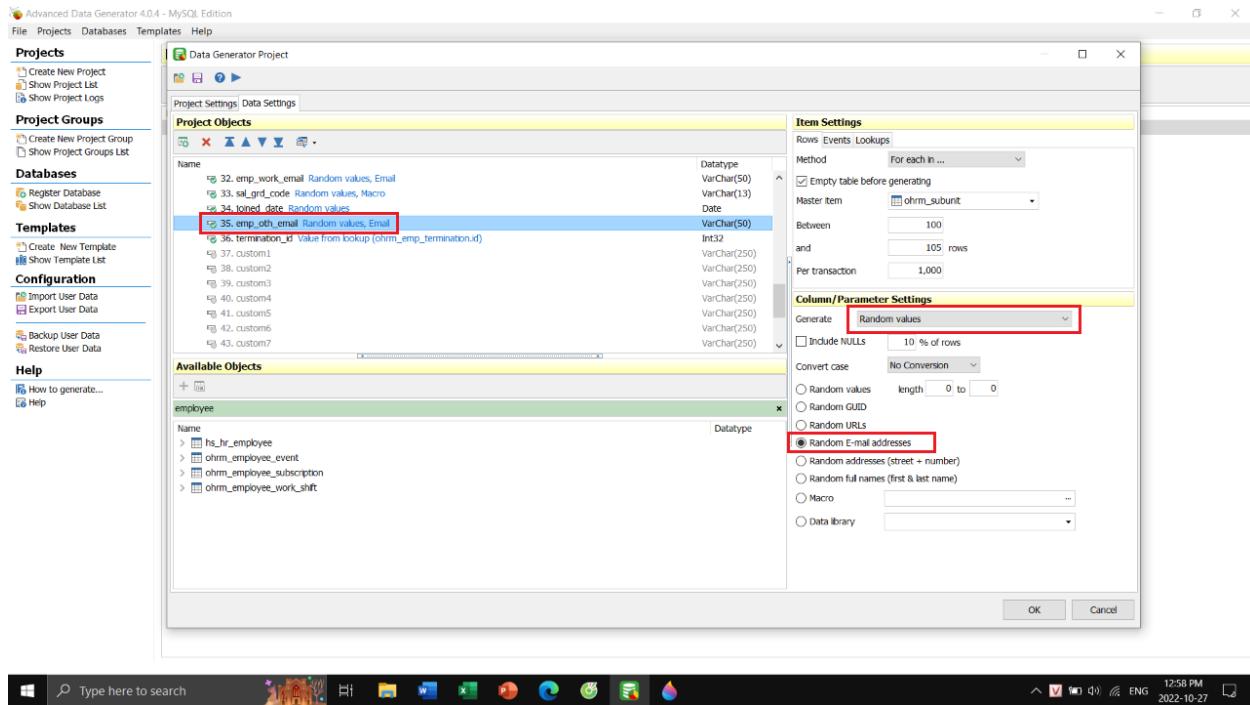
- Tạo dữ liệu cho trường sal_grd_code là một dãy ngẫu nhiên 5 chữ số



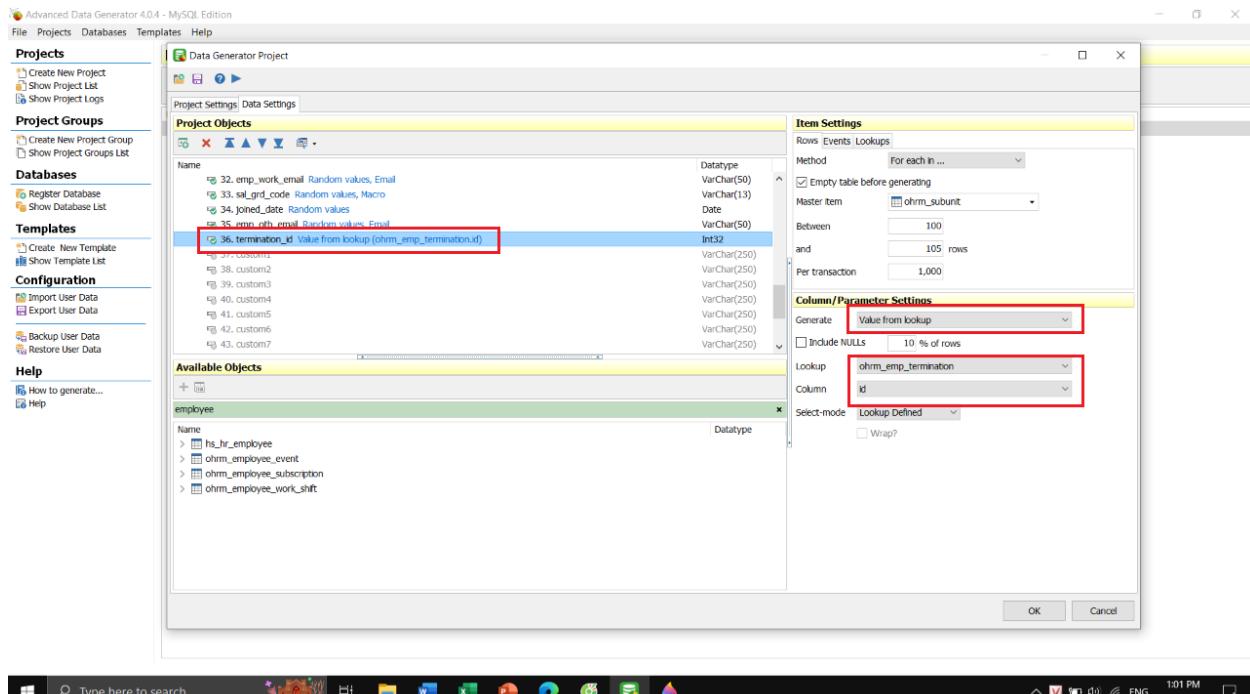
- Tạo dữ liệu cho trường joined_date bằng cách tạo ngẫu nhiên 1 ngày trong khoảng thời gian nào đó



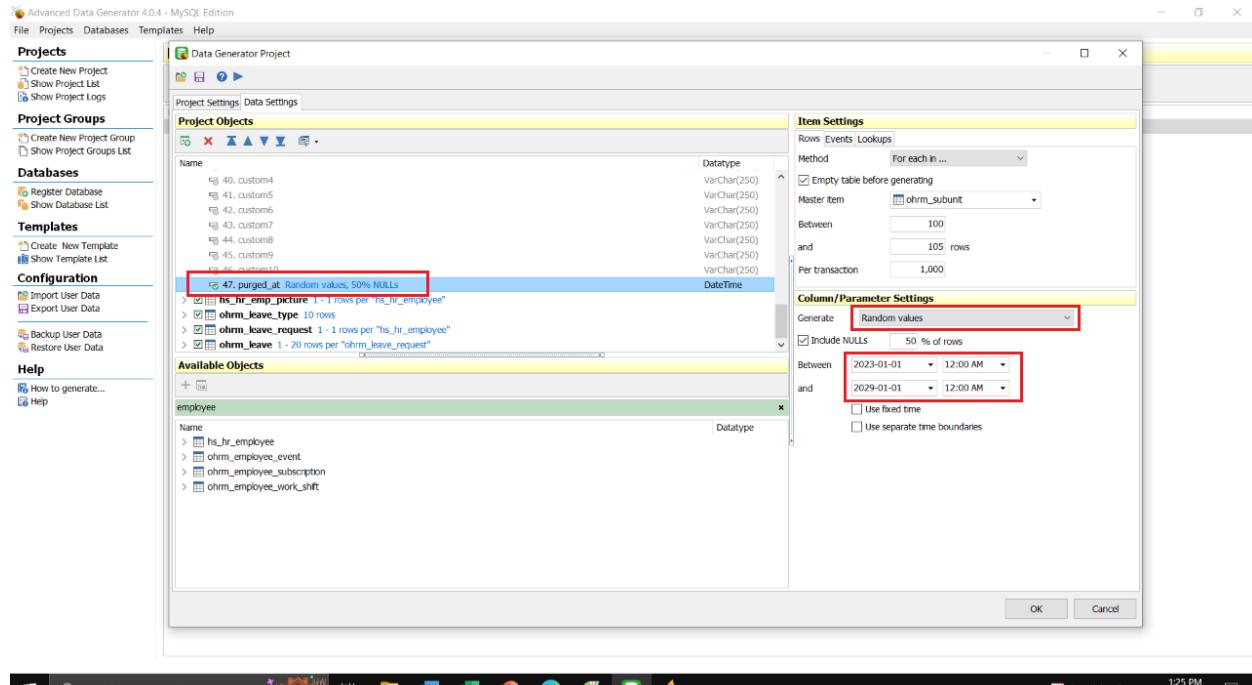
- Tạo dữ liệu cho trường emp_oth_email bằng Random e-mail addresses



- Tạo dữ liệu cho trường termination_id bằng cách tạo lookup tới trường id trong bảng ohrm_emp_termination ta đã tạo ở trên do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên

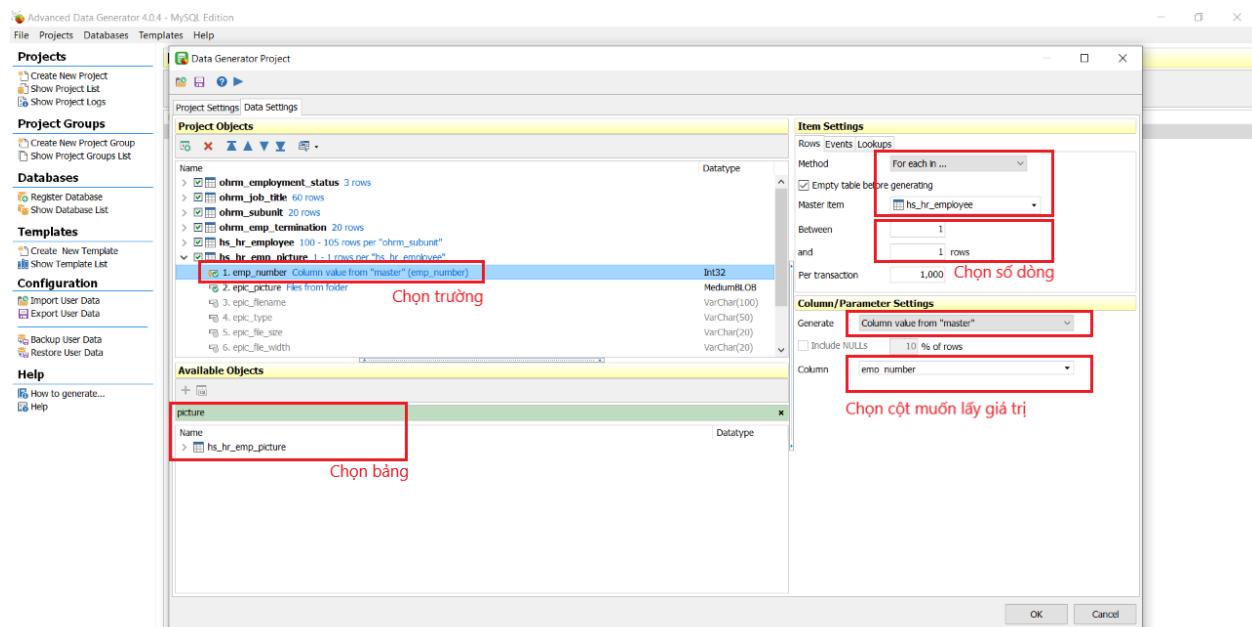


- Tạo dữ liệu cho trường purged_at bằng cách tạo ngẫu nhiên 1 ngày trong khoảng thời gian nào đó. Ngoài ra, ta cho 50% số lượng dòng là null

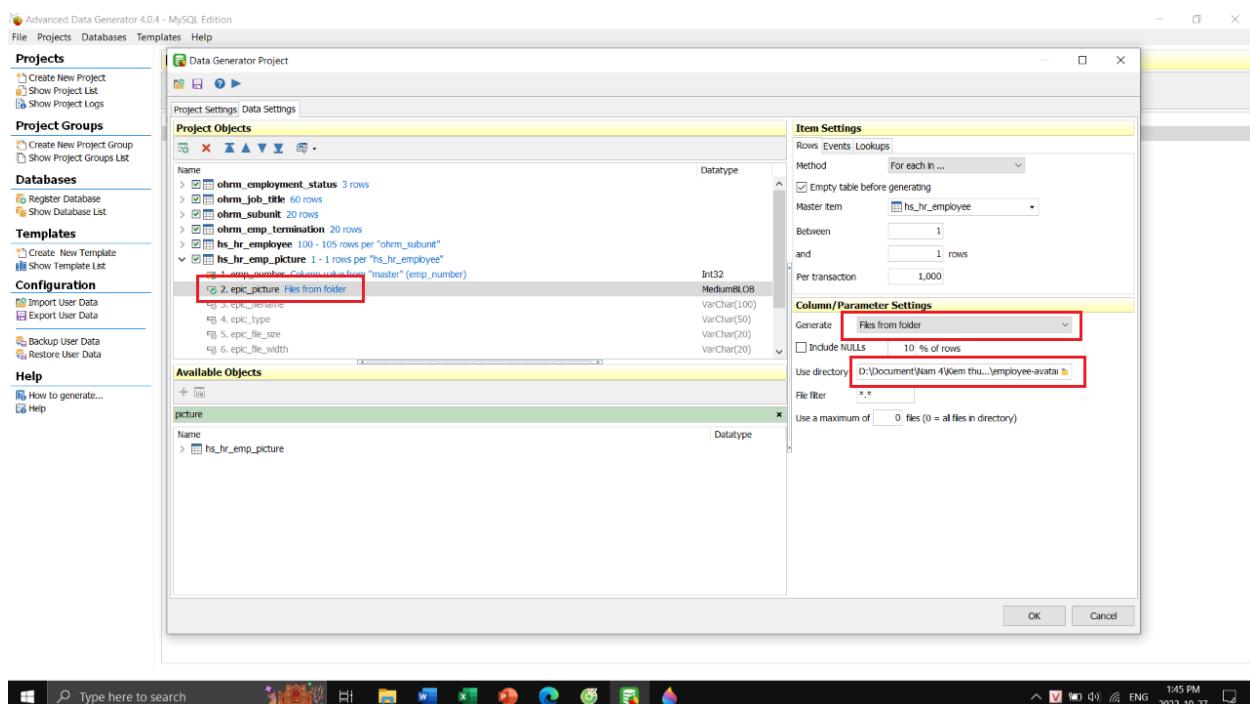
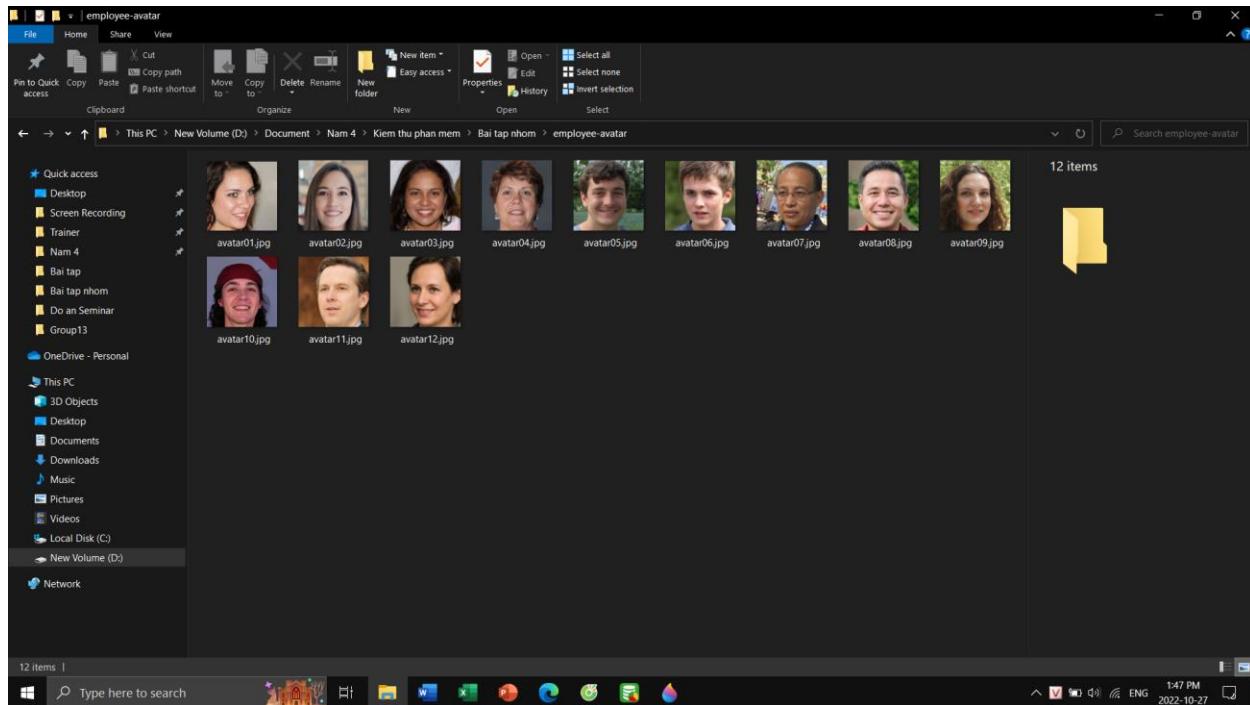


1.8. Tạo profile picture cho mỗi nhân viên (hs_hr_emp_picture)

- Tìm và chọn bảng có tên hs_hr_emp_picture để tạo dữ liệu
- Trong Items Settings → Methods, chọn For each in... để tạo dữ liệu với mỗi dòng trong bảng nhân viên (hs_hr_employee)

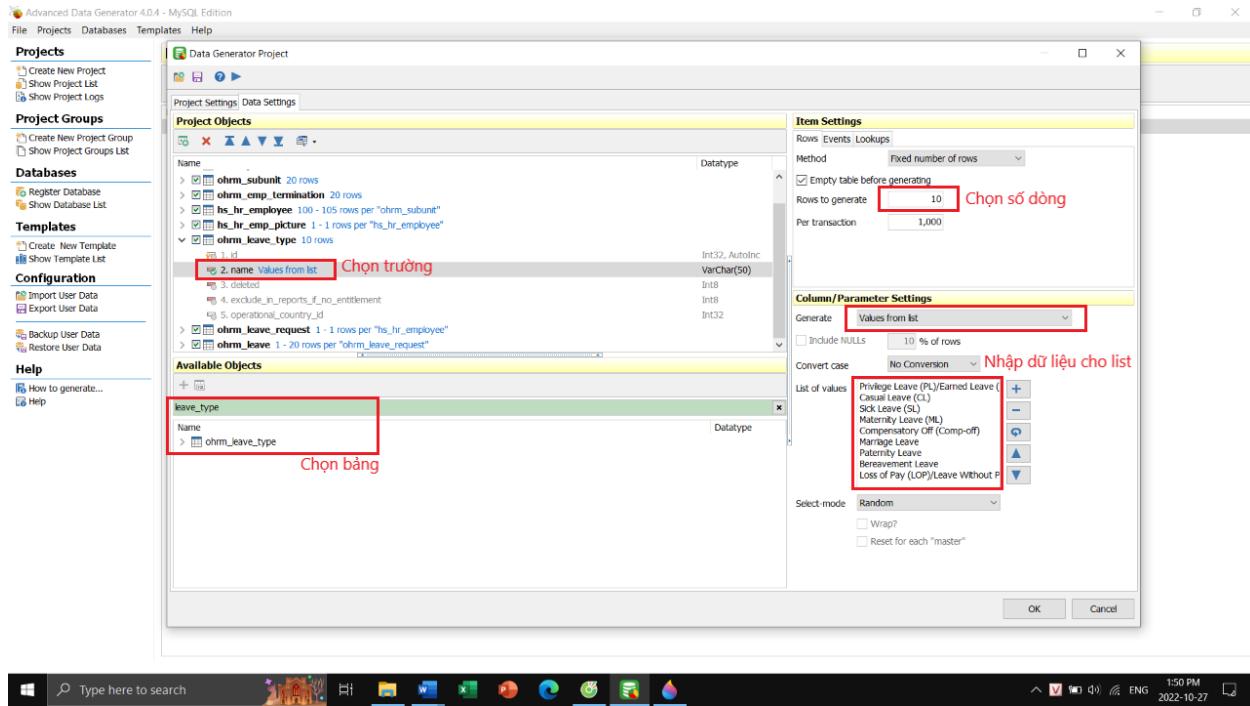


- Tạo dữ liệu cho trường epic_picture, ta chuẩn bị sẵn một folder chứa ảnh, sau đó chọn phương thức Files from folder, và chọn thư mục đã tạo.



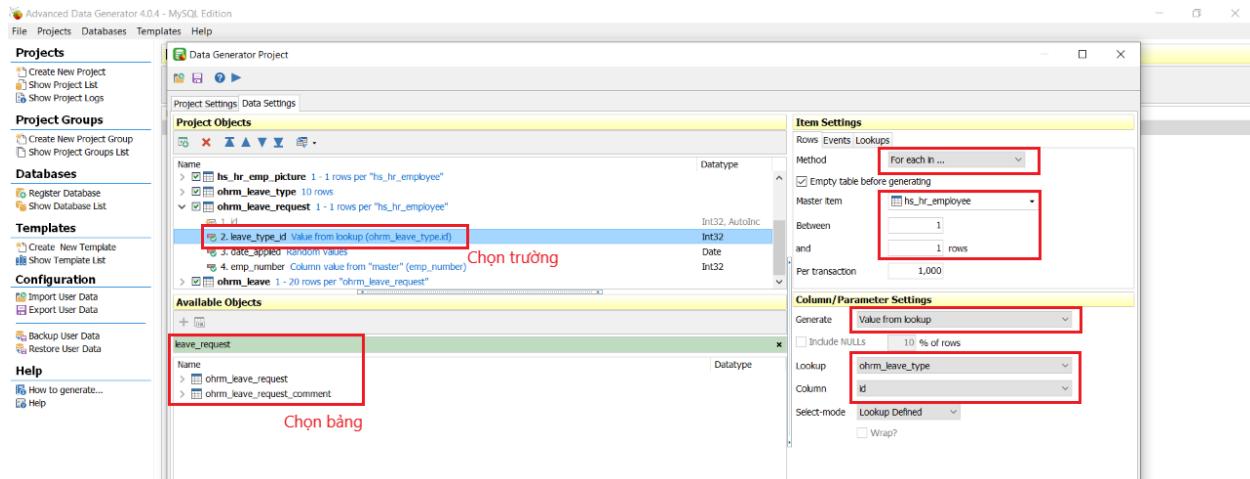
1.9. Tạo dữ liệu cho bảng Loại nghỉ phép (ohrm_leave_type)

- Tìm và chọn bảng có tên ohrm_leave_type.
- Tạo dữ liệu cho trường name bằng phương thức Value from list

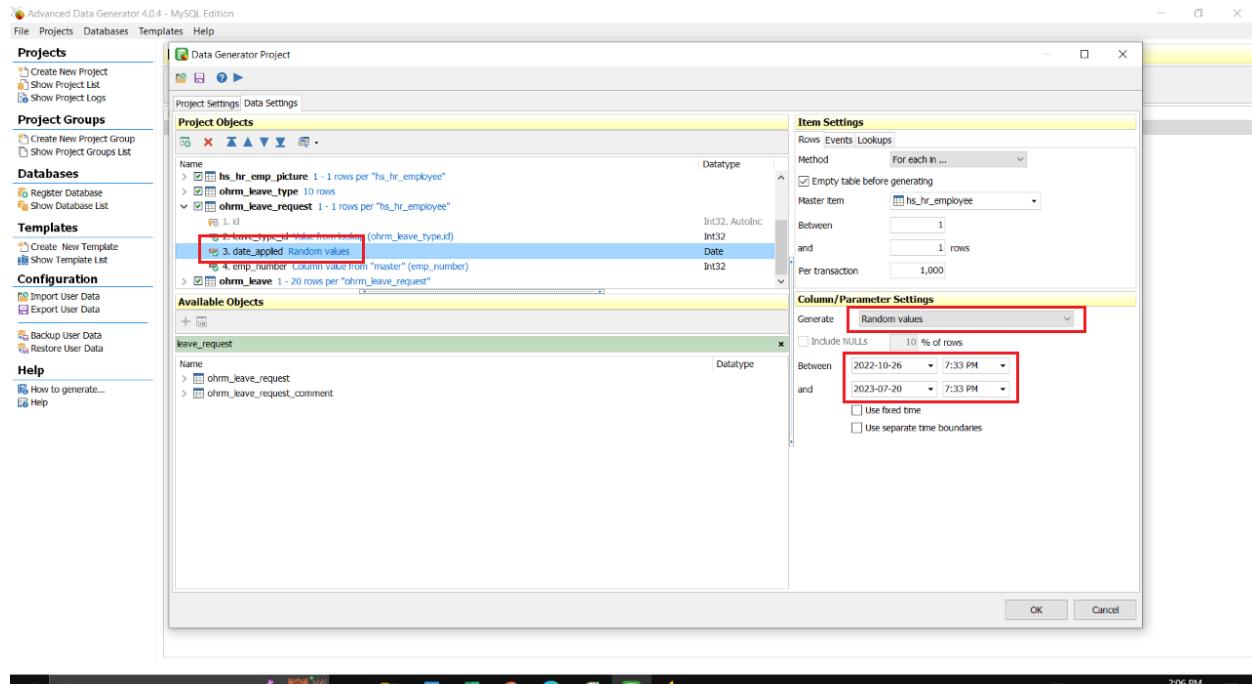


1.10. Tạo dữ liệu cho bảng Yêu cầu nghỉ phép (ohrm_leave_request)

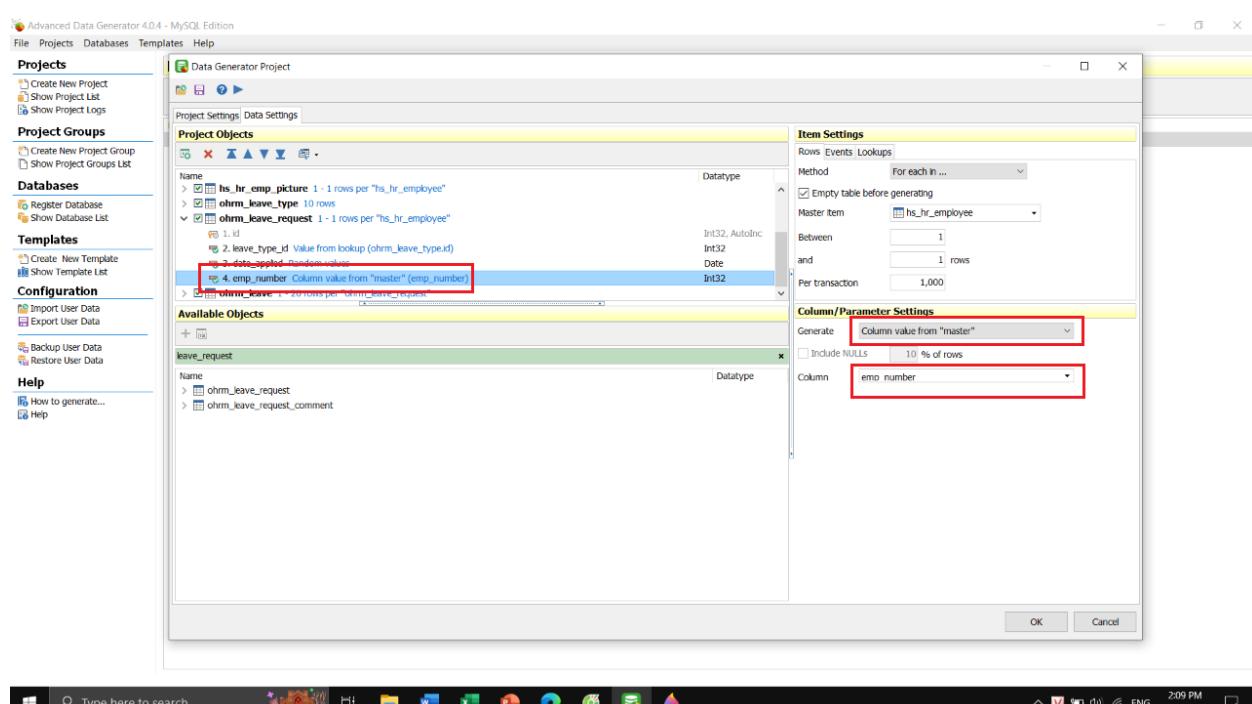
- Tìm và chọn bảng có tên ohrm_leave_request.
- Trong Items Settings → Methods, chọn For each in... để tạo dữ liệu với mỗi dòng trong bảng nhân viên (hs_hr_employee)
- Tạo dữ liệu cho trường leave_type_id bằng cách tạo lookup tới trường id trong bảng ohrm_leave_type ta đã tạo ở trên do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên



- Tạo dữ liệu cho trường date_applied bằng cách tạo ngẫu nhiên 1 ngày trong khoảng thời gian nào đó

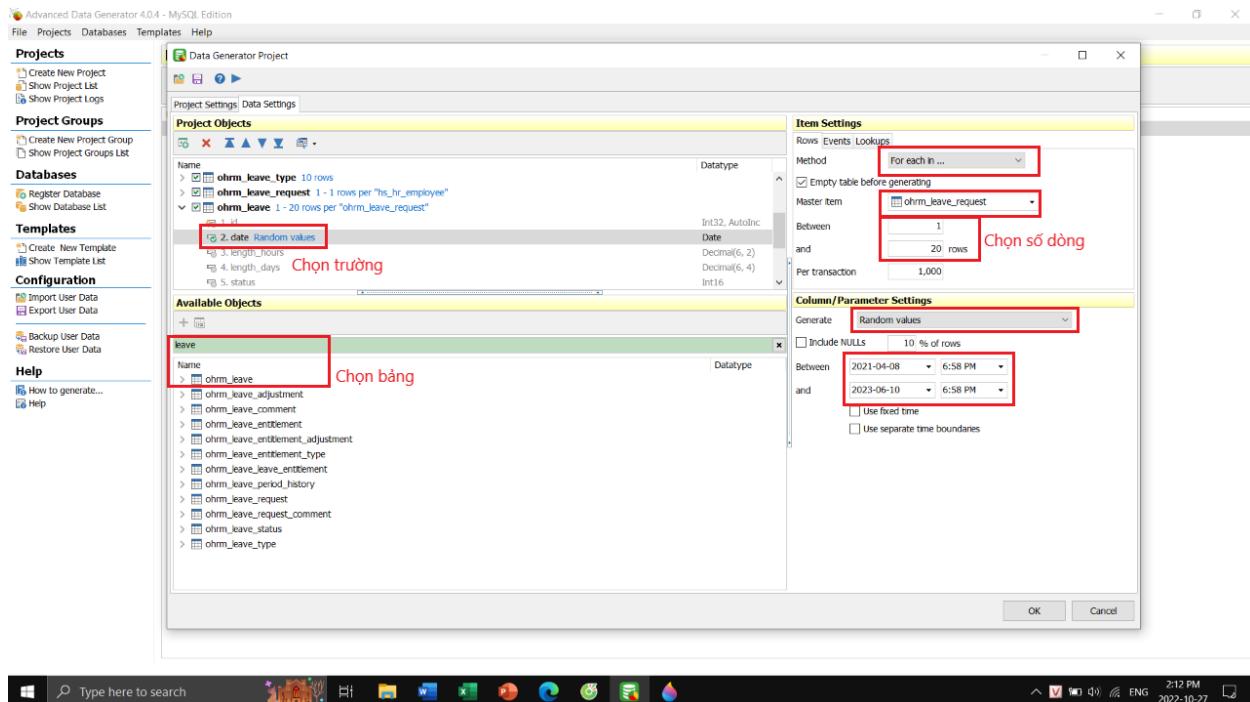


- Tạo dữ liệu cho trường emp_number bằng phương thức Column value from “master” để lấy mã của nhân viên tương ứng

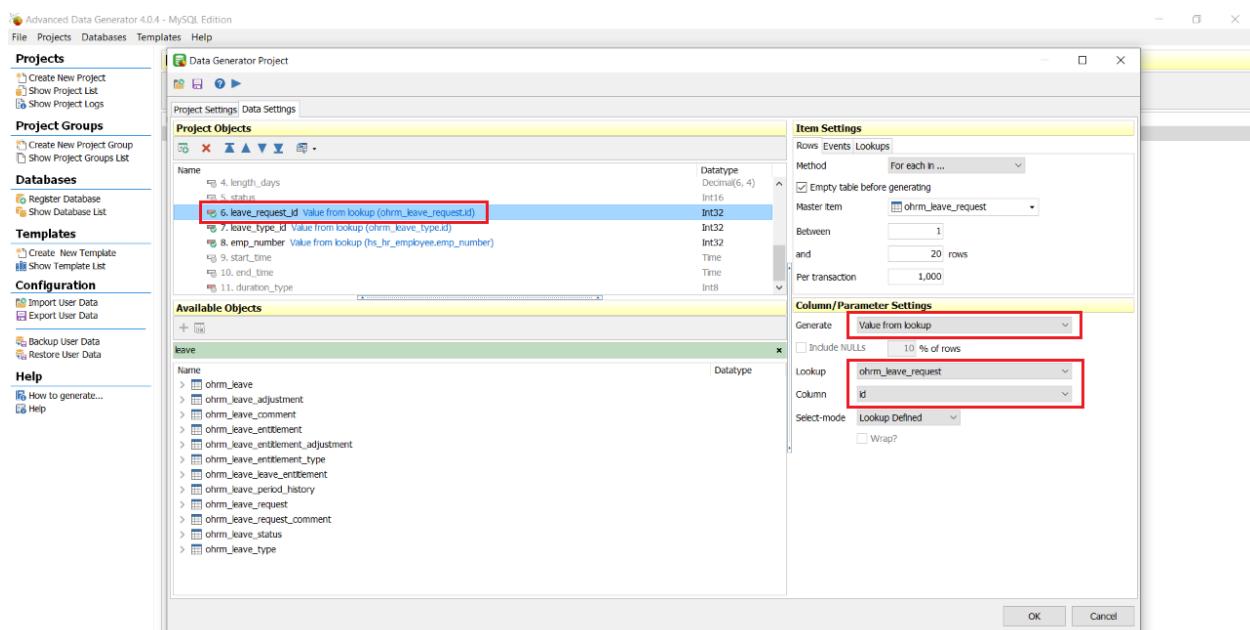


1.11. Tạo dữ liệu cho bảng nghỉ phép (ohrm_leave)

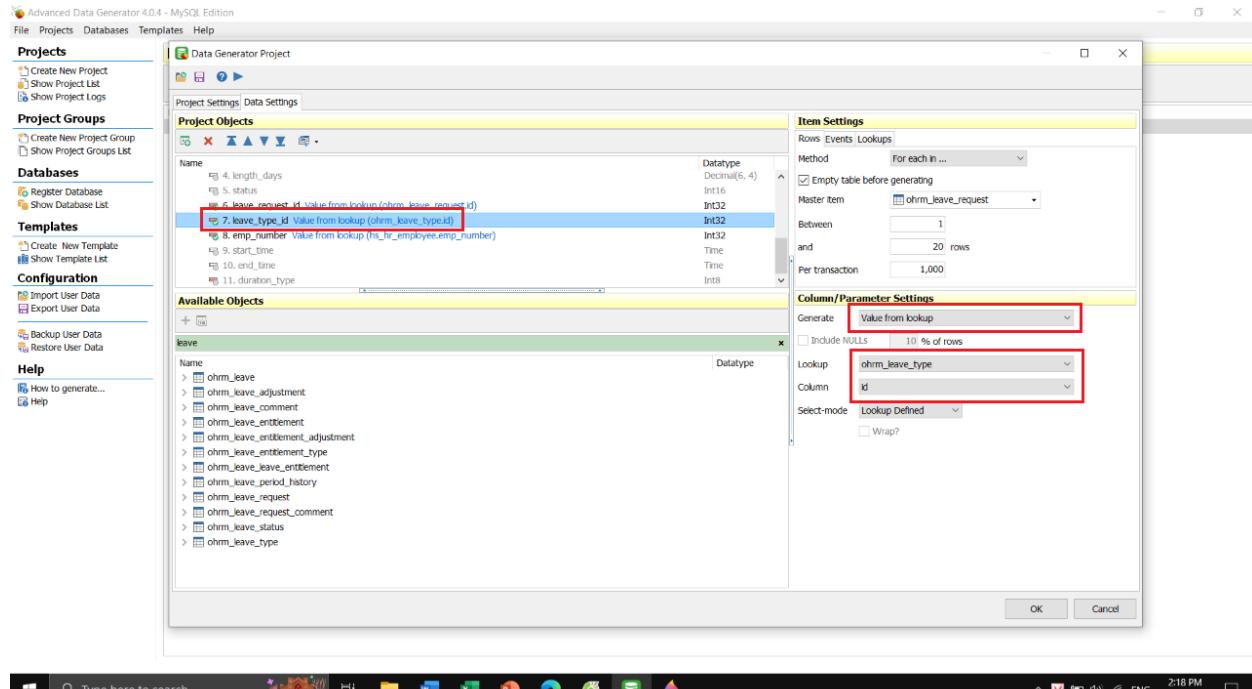
- Tìm và chọn bảng có tên ohrm_leave
- Trong Items Settings → Methods, chọn For each in... để tạo dữ liệu với mỗi dòng trong bảng Yêu cầu nghỉ phép (ohrm_leave_request)
- Tạo dữ liệu cho trường date bằng cách tạo ngẫu nhiên 1 ngày trong khoảng thời gian



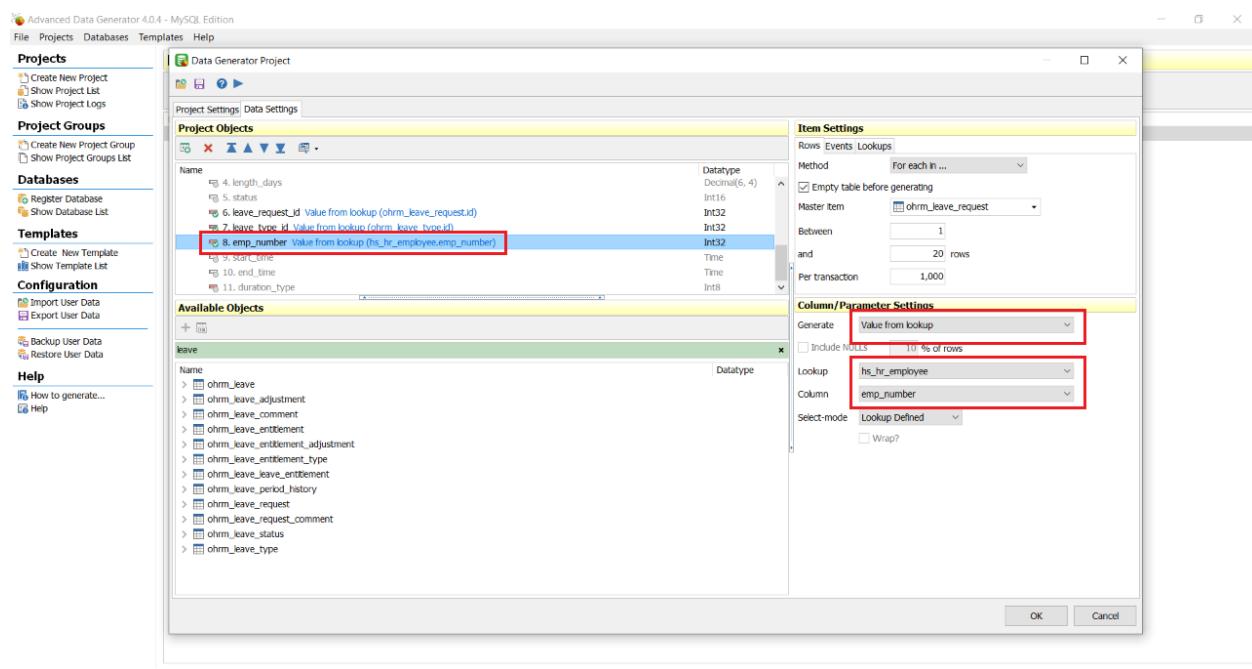
- Tạo dữ liệu cho trường leave_request_id bằng cách tạo lookup tới trường id trong bảng ohrm_leave_request ta đã tạo ở trên do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên



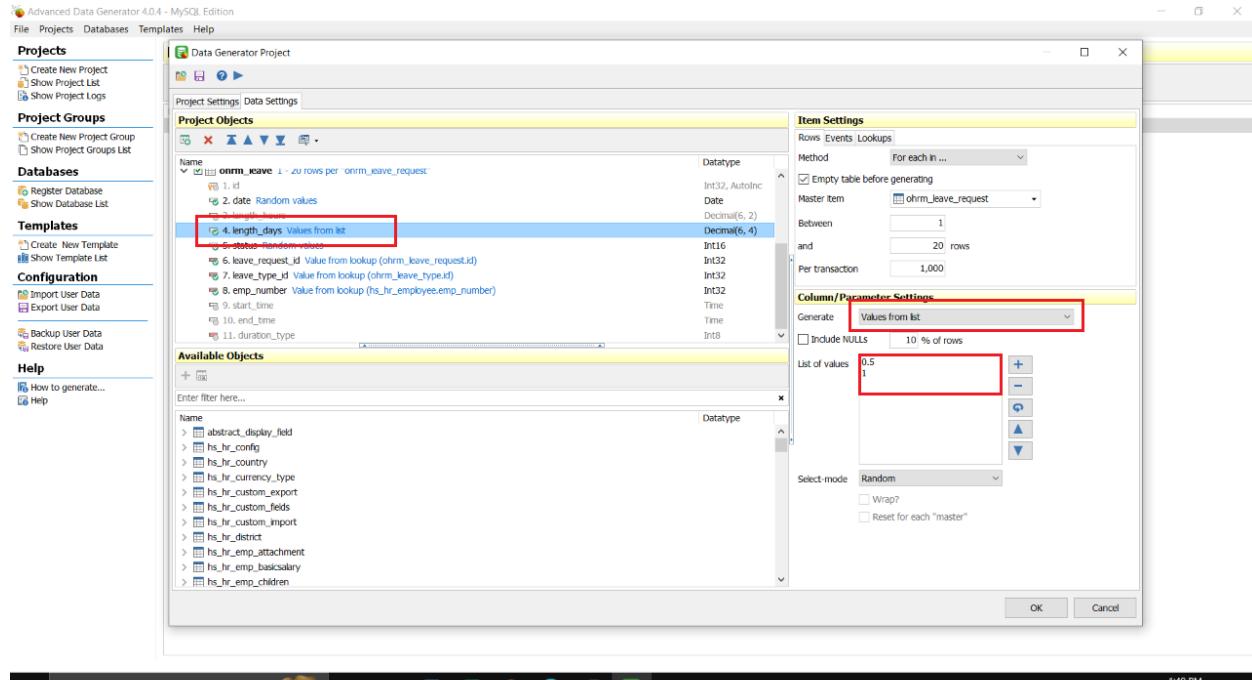
- Tạo dữ liệu cho trường leave_type_id bằng cách tạo lookup tới trường id trong bảng ohrm_leave_type ta đã tạo ở trên do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên



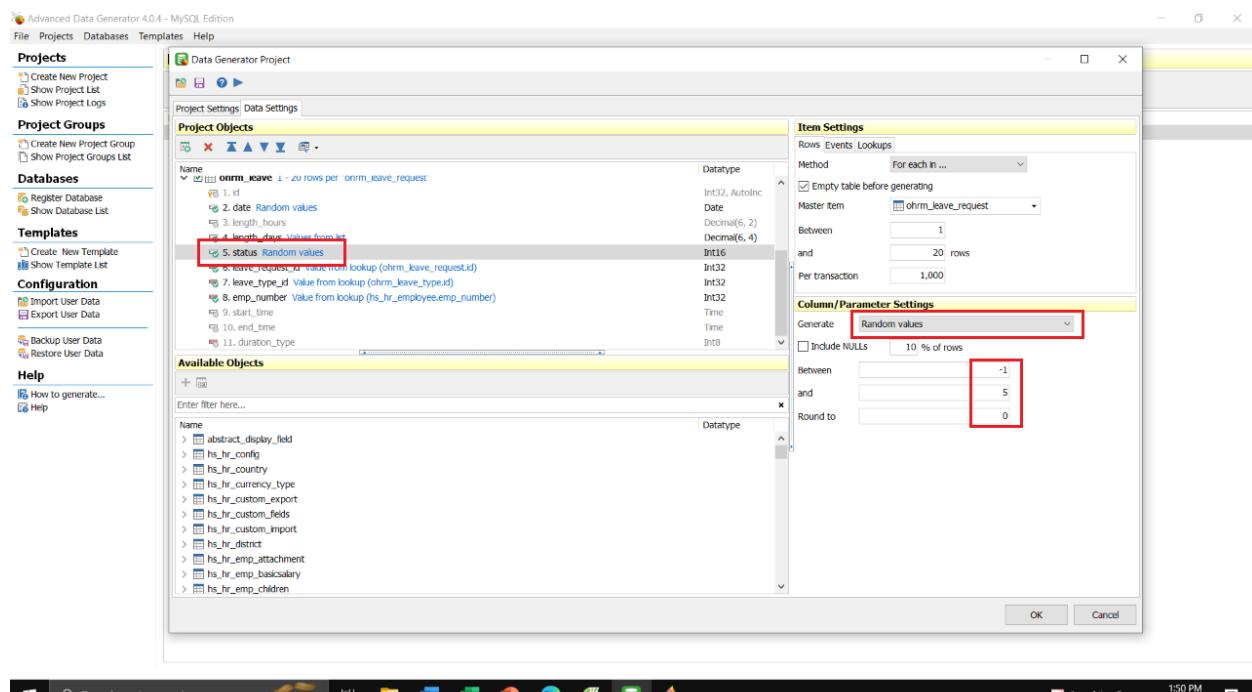
- Tạo dữ liệu cho trường emp_number bằng cách tạo lookup tới trường emp_number trong bảng hs_hr_employee ta đã tạo ở trên do trường này là khóa ngoại. Cách tạo 1 lookup tương tự như đã trình bày ở trên



- Tạo dữ liệu cho trường length_days bằng phương thức Value from list với các giá trị 0.5 và 1

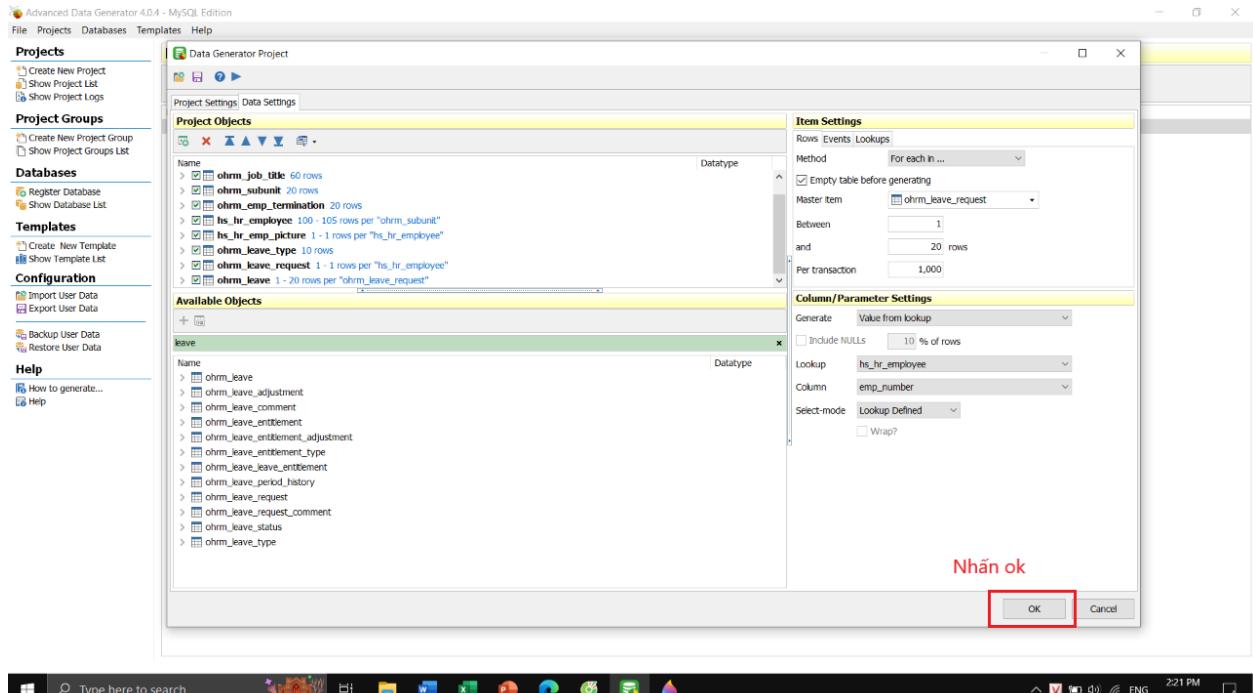


- Tạo dữ liệu cho trường status bằng cách tạo giá trị ngẫu nhiên từ -1 đến 5 ứng với các khóa chính trong bảng ohrm_leave_status

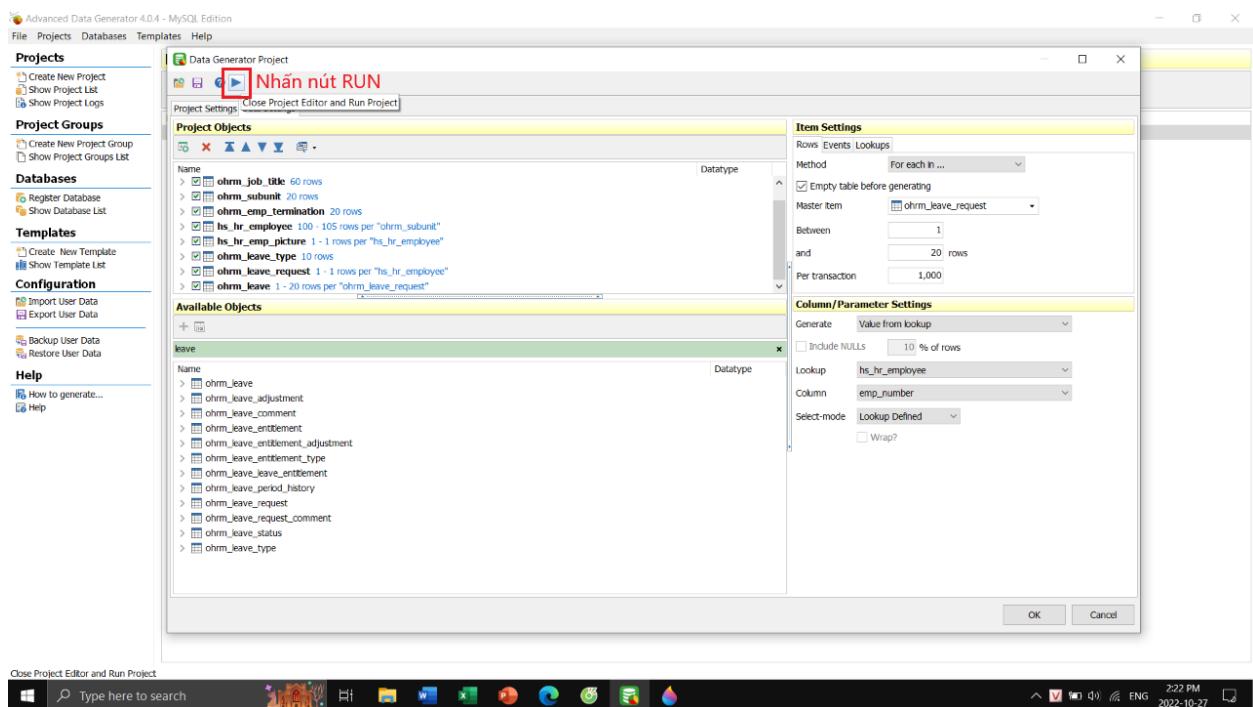


1.12. Chạy project

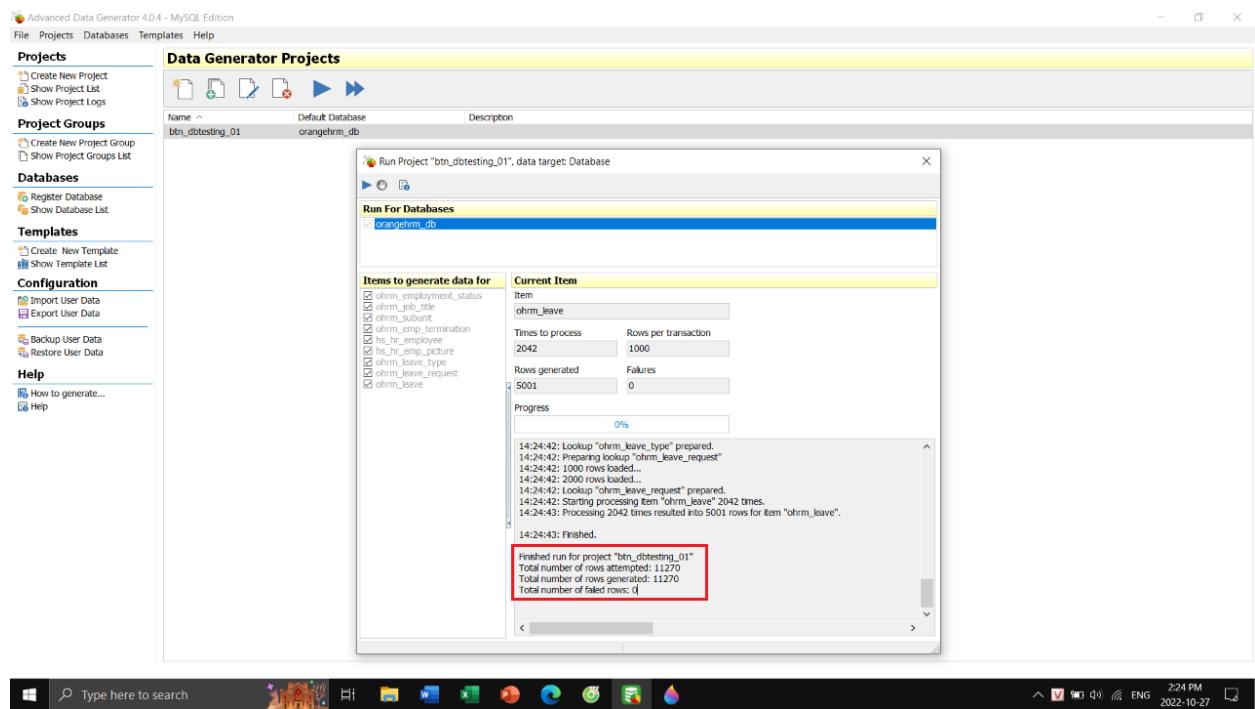
- Nhấn OK để lưu project



- Nhấn nút RUN



- Kết quả



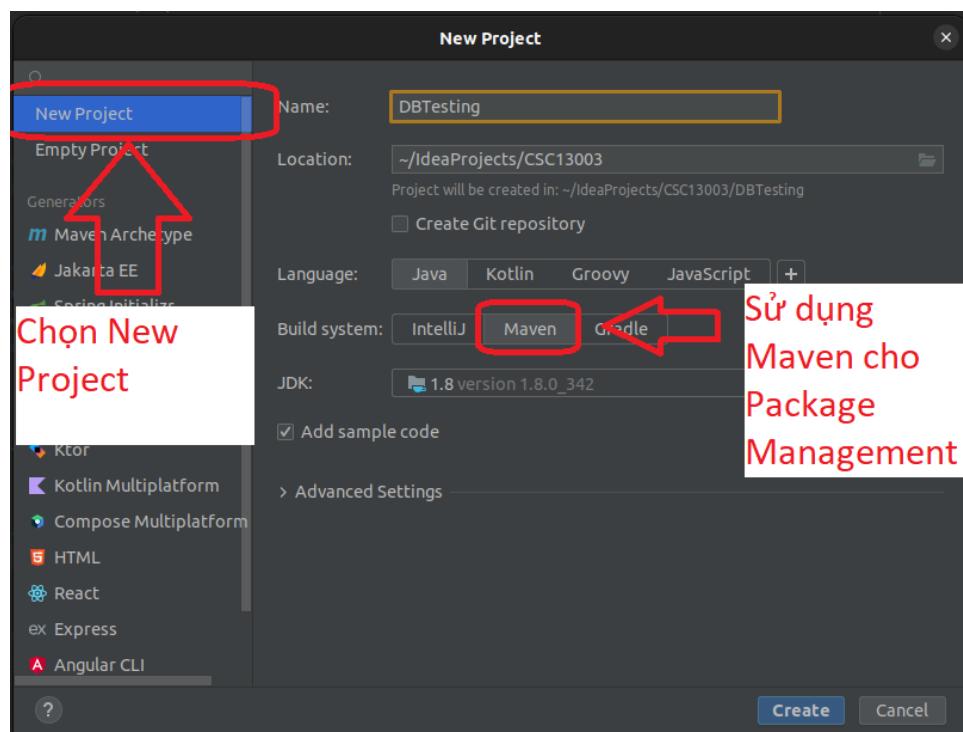
2. Kiểm thử cơ sở dữ liệu cho ứng dụng OrangeHRM

2.1. Công cụ sử dụng

- IntelliJ Community cho Java IDE
- Java SDK 8
- DBUnit 2.7.3
- JUnit 4.12

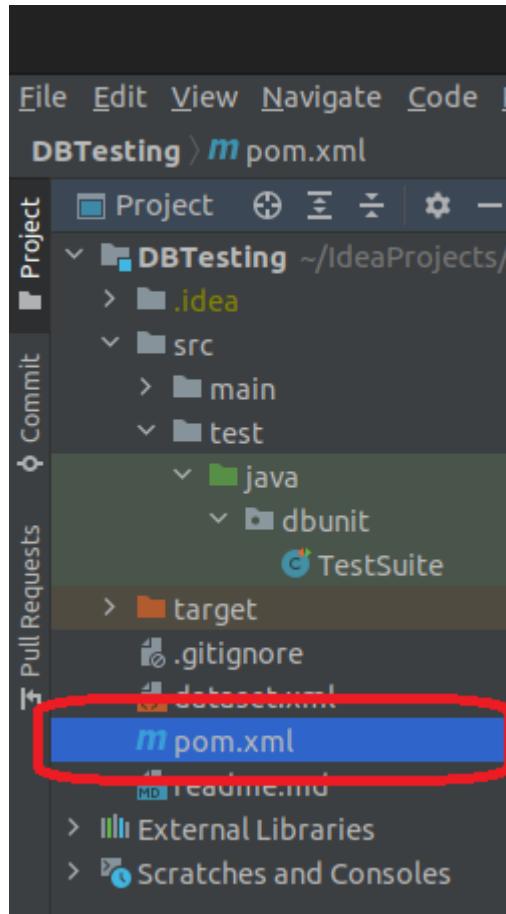
2.2. Tạo Maven Project bằng IntelliJ Community

- Sử dụng IntelliJ Community để tạo một Maven Project



2.3. Cài đặt các dependency trên pom.xml

- Sau khi đã tạo project, trên cây thư mục, ta sẽ nhìn thấy một tệp tên là pom.xml, đây là tệp được dùng để quản lý các dependency trong Maven.



- Tiến hành cài đặt các dependency cho DBUnit, vì DBUnit sử dụng JUnit, nên ta cần phải cài đặt cả JUnit theo, cụ thể ta cần:
 - o DBUnit 2.7.3
 - o JUnit 4.12
- Mặt khác, vì OrangeHRM sử dụng MySQL/MariaDB, ta cần phải có connector của các database này để có thể cho phép Java giao tiếp, connector này thực chất là dependency, cụ thể:
 - o MySQL Connector Java 8.011

- Như vậy, pom.xml sẽ có 2 mục quan trọng như sau:

- <properties> cho quản lý version

```

<properties>
    <maven.compiler.source>8</maven.compiler.source>
    <maven.compiler.target>8</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <dbunit-version>2.7.3</dbunit-version>
    <junit-version>4.12</junit-version>
    [REDACTED]
    <mysqlConnector-version>8.0.11</mysqlConnector-version>
    [REDACTED]

</properties>

```

- <dependencies> cho quản lý dependencies

```

22
23      <dependencies>
24          <dependency>
25              <groupId>org.dbunit</groupId>
26              <artifactId>dbunit</artifactId>
27              <version>${dbunit-version}</version>
28              <scope>test</scope>
29          </dependency>
30
31          <dependency>
32              <groupId>junit</groupId>
33              <artifactId>junit</artifactId>
34              <version>${junit-version}</version>
35              <scope>test</scope>
36          </dependency>
37
38
39
40
41
42
43
44          <dependency>
45              <groupId>mysql</groupId>
46              <artifactId>mysql-connector-java</artifactId>
47              <version>${mysqlConnector-version}</version>
48          </dependency>
49
50
51
52
53
54
55      </dependencies>

```

- Sau khi đã chuẩn bị xong pom.xml, nhấn Ctrl + Shift + O để IntelliJ download các dependencies này về.

2.4. Cấu hình cho DBUnit

- Trong thư mục src/test/java, ta tạo một class để extends DBTestCase, đây sẽ là nơi để ta setup cho các test case này

```
1 package dbunit;
2
3 import org.dbunit.DBTestCase;
4 import org.dbunit.PropertiesBasedJdbcDatabaseTester;
5 import org.dbunit.database.DatabaseConfig;
6 import org.dbunit.dataset.IDataSet;
7 import org.dbunit.dataset.xml.FlatXmlDataSetBuilder;
8 import org.dbunit.ext.mysql.MySqlMetadataHandler;
9 import org.dbunit.operation.DatabaseOperation;
10 import org.junit.Test;
11
12 import java.io.FileInputStream;
13 import java.sql.*;
14 import java.time.Year;
15
16 public class TestSuite extends DBTestCase {
```

- Do DBTestCase có một abstract method, tên là getDataSet(), ta buộc phải override cho method này. Đây là một method dùng để load một file dataset nằm bên ngoài (trong ví dụ này, chính là dataset.xml). Đây là những dataset mặc định ban đầu mà DBUnit sẽ insert vào trong cơ sở dữ liệu của chúng ta.

```
18     public static final String DATASET_PATH = "dataset.xml";
```



```
55
56     //Load the default dataset
57     @Override
58     protected IDataSet getDataSet() throws Exception {
59         return new FlatXmlDataSetBuilder().build(new FileInputStream(DATASET_PATH))
60     }
```

- Đây là bên trong file dataset.xml, sẽ insert một record vào bảng ohrm_login với các column tương ứng

```
1 <dataset>
2     <ohrm_login id="12" user_id="1234" user_name="abc" user_role_name="NhanVien" user_role_predefined="1"/>
3 </dataset>
```

- Tiếp theo, class của ta cần một constructor, constructor này sẽ nhận tham số là 1 string name, và ta cần setup connect string để có thể kết nối với cơ sở dữ liệu
 - Dòng 42 để kết nối với MySQL Connector, với giá trị ở dòng 27, dòng này ta không cần edit để phù hợp với configuration.
 - Dòng 43 để kết nối bằng connection string cho database của OrangeHRM, ở đây có 3 tham số ta cần chú ý để config cho phù hợp:
 - IP: dòng 28, đây chính là địa chỉ IP mà ta đã host cho database của OrangeHRM, nếu dùng localhost thì ta đặt 127.0.0.1
 - Port: dòng 29, port của IP phía trên, để kết nối với database, nếu cài đặt mặc định theo xampp sẽ là port 3306
 - Schema's name: dòng 30, đây là tên của database khi ta install OrangeHRM, ở đây khi install, em đặt tên là orangehrm_test
 - Sau khi cài đặt cả 3 tham số trên, ta móc chúng lại với nhau bằng connection string ở dòng 32, sau đây nạp vào dòng 43.
 - Dòng 45 để truyền username trong DBMS, dùng để thao tác trên database, username sẽ được config ở dòng 33, em sử dụng giá trị root
 - Dòng 46 để truyền password của username phía trên, được config ở dòng 34.
 - Sau khi setup xong, ở dòng 49, ta thực hiện kết nối với cơ sở dữ liệu qua biến connection

```

24     private static final int BATCH_SIZE = 50;
25
26     //Connection profile
27     private static final String DRIVER_CLASS = "com.mysql.cj.jdbc.Driver";
28     private static final String IP = "127.0.0.1";           //editable
29     private static final String PORT = "3306";            //editable
30     private static final String SCHEMA_NAME = "orangehrm_test"; //editable
31
32     private static final String CONNECTION_STR = "jdbc:mysql://" + IP + ":" + PORT + "/" + SCHEMA_NAME + "?useSSL=false";
33     private static final String USERNAME = "root"; //editable
34     private static final String PASSWORD = "";      //editable
35
36
37     //Initialize and connect to the database with the given profile
38     public TestSuite(String name) {
39         super(name);
40
41         //Setup connection profile
42         System.setProperty(PropertiesBasedJdbcDatabaseTester.DBUNIT_DRIVER_CLASS, DRIVER_CLASS);
43         System.setProperty(PropertiesBasedJdbcDatabaseTester.DBUNIT_CONNECTION_URL, CONNECTION_STR);
44         System.setProperty(PropertiesBasedJdbcDatabaseTester.DBUNIT_USERNAME, USERNAME);
45         System.setProperty(PropertiesBasedJdbcDatabaseTester.DBUNIT_PASSWORD, PASSWORD);
46         System.setProperty(PropertiesBasedJdbcDatabaseTester.DBUNIT_SCHEMA, SCHEMA_NAME);
47
48         try {
49             connection = DriverManager.getConnection(CONNECTION_STR, USERNAME, PASSWORD);
50         } catch (SQLException e) {
51             throw new RuntimeException(e);
52         }
53     }
54 }
```

- Tiếp theo, ta cần setup để sao cho sau mỗi test case, DBUnit sẽ reload lại dataset, bằng cách override lại 2 method này của DBUnit:

```

61
62     @Override
63     protected DatabaseOperation getSetUpOperation() throws Exception {
64         return DatabaseOperation.REFRESH;
65     }
66
67     @Override
68     protected DatabaseOperation getTearDownOperation() throws Exception {
69         return DatabaseOperation.DELETE_ALL;
70     }
71

```

- Ta cũng cần phải sử dụng batch statement, hay còn gọi là bundle statement, để có thể execution nhiều câu query 1 lúc mà không phải đóng/mở connection đến database liên tục. Ở đây, em đặt giá trị batch size là 97.

```

22
23     //Database configuration
24     private static final int BATCH_SIZE = 97;
25

```

```

71
72     @Override
73     protected void setUpDatabaseConfig(DatabaseConfig config) {
74         config.setProperty(DatabaseConfig.PROPERTY_BATCH_SIZE, BATCH_SIZE); //setup batch size for sql execution
75         config.setProperty(DatabaseConfig.PROPERTY_METADATA_HANDLER, new MySqlMetadataHandler());
76         config.setProperty(DatabaseConfig.FEATURE_BATCHED_STATEMENTS, true);
77     }
78

```

2.5. Viết test case

- Test case là 1 method có annotation @Test

```
78  
79     @Test  
80     public void testQuery() throws SQLException {  
81
```

- Đầu tiên, để code clean hơn, ta nên khai báo các biến ứng với các tham số cần sử dụng:

```
81  
82         //Variable to store the result after querying  
83         Integer result = 0; ➔ Biến lưu kết quả sau khi query  
84  
85         //Parameters for the query, because there is no sensitive information  
86         // => there is no need to use binding parameter technique  
87         String thisYear = Year.now().toString(); ➔ Lấy từ hệ thống năm 2022  
88         final int maxLeaveDayLength = 12; ➔ Số ngày nghỉ tối đa  
89
```

- Thực hiện mở connection để chuẩn bị query
 - Chú ý: Sẽ là best practice khi sử dụng PreparedStatement thay vì Statement trong trường hợp này, vì prepared statement cho phép ta binding parameter, tránh được SQL Injection. Tuy nhiên, do các thông tin trong câu SQL không mang sự nhạy cảm, nên em xin phép được concatenate string với Statement trong câu SQL cho đơn giản.

```
85  
90         //JDBC's statement to query  
91         Statement statement = connection.createStatement();
```

- Ta viết câu query và concatenate các parameter đã chuẩn bị sẵn
 - Giải thích câu query:
 - Về yêu cầu, ta cần **đếm**, do đó, ta sẽ có ý tưởng sẽ Count một cái bảng sao cho: bảng này trả về **danh sách** nhân viên nghỉ quá 12 ngày trong năm 2022, đây chính là bảng trong câu nested query ở phần *From*.
 - Về câu nested query:
 - *Select*: Do trả về danh sách nhân viên, ta sẽ select emp_number của các nhân viên này
 - *From*: ta thực hiện join các bảng:
 - hs_hr_employee: đây là bảng chứa thông tin các nhân viên
 - ohrm_leave_request: thông tin về request xin nghỉ phép của 1 nhân viên
 - ohrm_leave: thông tin về vấn đề nghỉ phép của 1 nhân viên
 - ohrm_leave_type: danh sách các loại hình nghỉ phép của công ty, thực chất không cần join bảng này vẫn cho ra đáp án đúng, nhưng em sẽ join vào cho nặng thêm feature.
 - ohrm_leave_status: danh sách các status của 1 leave, danh sách này cố định gồm 7 row kể từ khi cài đặt OrangeHRM
 - *Where*:
 - Em cho rằng 1 nhân viên nghỉ phép, thì phải là 1 leave sao cho có status là “TAKEN” hoặc “SCHEDULED”, đây là 2 trong 7 loại status nằm trong ohrm_leave_status.
 - Đồng thời, do ta đang cần đếm những ngày nghỉ phép của nhân viên nằm trong năm 2022, ta sẽ chỉ lấy những ngày nghỉ nằm trong năm này.
 - *Group by*:
 - Do ta sẽ cần *Having* bên dưới để tính tổng số ngày nghỉ của 1 nhân viên bất kì, nên em sẽ Group theo emp_number của các nhân viên
 - *Having*:
 - Ta sẽ chỉ lấy những nhân viên nào có tổng số ngày nghỉ vượt quá 12 ngày.

```

92
93     String sql =
94         "SELECT COUNT(*) " +
95         "FROM " +
96         "(" +
97             "SELECT DISTINCT e1.emp_number " +
98             "FROM hs_hr_employee e1 " +
99                 "JOIN ohrm_leave_request lr1 ON e1.emp_number = lr1.emp_number " +
100                "JOIN ohrm_leave ll ON ll.leave_request_id = lr1.id " +
101                  "JOIN ohrm_leave_type lt1 ON ll.leave_type_id = lt1.id " +
102                    "JOIN ohrm_leave_status ls1 ON ls1.status = ll.status " +
103                      "WHERE ls1.name IN ('TAKEN', 'SCHEDULED') AND YEAR(ll.date) = " + thisYear + " " +
104                        "GROUP BY e1.emp_number " +
105                          "HAVING SUM(ll.length_days) > " + maxLeaveDayLength + " " +
106                            ") as temp ";

```

- Ta execute câu query và đo thời gian chạy:

```

107
108     //Try to measure the execution time, in milisecond
109     long start = System.currentTimeMillis();
110     ResultSet rs = statement.executeQuery(sql);
111     long end = System.currentTimeMillis();
112

```

- Ta fetch đáp án từ kết quả của câu query ra và lưu vào biến result, sau đó đóng statement lại

```

112
113     //Parse the answer to the result variable
114     while (rs.next()) {
115         result = rs.getInt(1);
116     }
117
118     //clear the resource after using
119     rs.close();
120

```

- Ta assert để kiểm tra xem kết quả có nhỏ hơn 1s hay không, nếu không thì bị TLE (Time limit exceeded). Sau đó, print kết quả ra console

```

121
122     //Assert if the answer is TLE: the execution time is larger 1 second
123     Assert.assertTrue(result < 1000L);
124
125     //Output the result
126     System.out.println("Number of employees who leave more than " + maxLeaveDayLength + " days in this year: " + result);
127     System.out.println("Execution time: " + (end-start) + "ms");

```

2.6. Thực hiện các test case

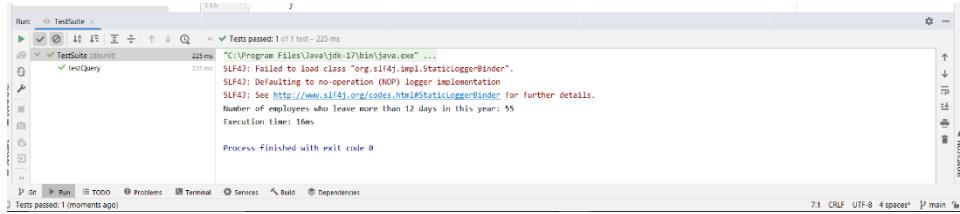
- Do ta cần tăng số lượng nhân viên lên để benchmark execution time của câu query:
 - Sẽ tốt hơn nếu ta sử dụng một java framework trong code để có thể generate tự động số lượng nhân viên trong database, với số lượng làm input, tuy nhiên nếu làm cách này, ta không thể tái sử dụng được các kiến thức từ việc database generation đã nói ở mục 1.

=> Do đó, ta sẽ tăng số lượng nhân viên manually và mỗi lần tăng, ta sẽ chạy lại test case
- Chạy lần 1:
 - Số lượng phòng ban: 20
 - Số lượng nhân viên mỗi phòng ban: 100
 - Số lượng leave request của mỗi nhân viên: 60 – 100
 - Số lượng leave của mỗi leave request: 1 – 20

- Chạy lần 2:
 - Số lượng phòng ban: 20
 - Số lượng nhân viên mỗi phòng ban: 500
 - Số lượng leave request của mỗi nhân viên: 60 – 100
 - Số lượng leave của mỗi leave request: 1 – 20

- Chạy lần 3:
 - Số lượng phòng ban: 20
 - Số lượng nhân viên mỗi phòng ban: 1000
 - Số lượng leave request của mỗi nhân viên: 60 – 100
 - Số lượng leave của mỗi leave request: 1 – 20

- Chạy lần 4:
 - Số lượng phòng ban: 20
 - Số lượng nhân viên mỗi phòng ban: 5000 (max của Upscene Advance Data Generator MySQL Edition)
 - Số lượng leave request của mỗi nhân viên: 60 – 100
 - Số lượng leave của mỗi leave request: 1 – 20



- Nhận xét:
 - Tất cả các câu query đều pass, không bị TLE.
 - Số lượng nhân viên nghỉ quá 12 ngày có tăng nhưng rất chậm, sau mỗi lần tăng số lượng nhân viên:
 - Lý do: Vì 1 leave có tổng cộng 7 status, nhưng ta chỉ lấy 2 trong 7 status này được xem như là xin nghỉ được chấp nhận (TAKEN, SCHEDULED)
➔ Tỉ lệ khi generate ra được 1 nhân viên có 12 ngày nghỉ, mà ngày nghỉ đó có trạng thái là TAKEN/SCHEDULED giảm đi đáng kể
 - Thời gian chạy câu query cũng tăng, nhưng rất chậm, ngay cả khi đã kết 5 bảng với nhau:
 - Lý do: Giới hạn của Upscene Advance Data Generator MySQL Edition chỉ cho phép tăng lên tối đa 5000 row → không thể generate ra đủ data để có thể làm database bị nghẽn cỗ chai
 - Chỉ có 1 connection duy nhất truy cập vào database.
 - Câu query chỉ chạy 1 lần duy nhất → không bị buffered thời gian lên do phải lập execute plan nhiều lần.
 - Câu query sử dụng Nested Query (truy vấn lồng), tuy nhiên đây chỉ là Independent Nested Query (truy vấn lồng phân cấp), tức là câu query cha và query con không tham chiếu đến nhau. Do đó câu query chạy nhanh hơn rất nhiều so với Co-related Nested Query (truy vấn lồng tương quan)

3. Báo cáo tự đánh giá

Yêu cầu	Mô tả kết quả đạt được	Thang điểm	Điểm tự đánh giá
Yêu cầu 1	Cài đặt và generate dữ liệu theo đúng yêu cầu bằng Upscene Advanced Data Generator MySQL Edition	5	5
Yêu cầu 2	Chạy câu query, pass testcase và đo được thời gian câu query được thực thi	5	5
	Tổng	10	10