**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

──────── \* ───────

**BÁO CÁO**

**Lập trình hướng đối tượng**

Mã lớp: 111557

Đề tài 5: MySQL

Giảng viên hướng dẫn: TS. Trịnh Tuấn Đạt

Nhóm sinh viên thực hiện: Nhóm 14

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ Tên | MSSV | Đóng góp |
| 1 | Nguyễn Huy Khôi | 20170086 | 20% |
| 2 | Lê Đình Bình | 20172971 | 20% |
| 3 | Phạm Thị Hương Quỳnh | 20183818 | 15% |
| 4 | Đỗ Mạnh Đông | 20155370 | 15% |
| 5 | Phạm Minh Anh | 20172950 | 15% |
| 6 | Trần Thị Thúy | 20173394 | 15% |

HÀ NỘI 12-2019

Phân công nhiệm vụ trong nhóm:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thành viên | % đóng góp | Nhiệm vụ đảm nhiệm |
| 1 | Nguyễn Huy Khôi | 20% | Chuẩn bị chương trình dự phòng 1 (là bản chính)  Thuyết trình chương trình với thầy |
| 2 | Lê Đình Bình  (nhóm trưởng) | 20% | Phân công công việc  Chuẩn bị chương trình dự phòng 2,  Làm báo cáo:   * Tìm hiểu công nghệ(phần 1) * Mô hình dữ liệu và giải thuật (phần 2) |
| 3 | Phạm Thị Hương Quỳnh | 15% | Chuẩn bị tập dữ liệu mẫu,  Làm báo cáo:   * Cấu trúc chương trình (phần 3) * Các biểu đồ của chương trình (phần 4) |
| 4 | Đỗ Mạnh Đông | 15% | Đánh giá cấu trúc và kiểm thử 2 chương trình  Đưa ra cải tiến chương trình chính để tối ưu cấu trúc, tốc độ thực thi |
| 5 | Phạm Minh Anh | 15% | Làm báo cáo:   * Kiểm thử hiệu năng chương trình (phần 5) * Đánh giá ưu nhược điểm (phần 6) |
| 6 | Trần Thị Thúy | 15% | Chuẩn bị ý tưởng các câu truy vấn,  Tối ưu câu lệnh SQL để tăng tốc độ xử lý,  Chuẩn bị slide thuyết trình |

Mục lục

[1,Tìm hiểu công nghệ:](#_6k922u8ohphz)  **4**

[1.1,Giới thiệu MySQL:](#_6k922u8ohphz) 4

[1.2,Các thuật ngữ:](#_t54qylilsi2d)  4

[1.3. Hướng dẫn cài đặt MySQL](#_xsg3wy72vvbk) 4

[1.4,Các kỹ thuật sử dụng:](#_m3kfu274lo94)  10

[1.4.1, Đọc và lấy dữ liệu từ file:](#_c5oxouec320a)  10

[1.4.2,Kỹ thuật xử lý theo lô: (Batch processing)](#_9gm9dtnkuuis) 11

[1.5, Cách truy vấn và giao tiếp với cơ sở dữ liệu bằng API:](#_m0mk1j6559l0)  11

[1.6, Mô hình dữ liệu trong MySQL](#_gzbqil17hjkd) 11

[2,Mô hình dữ liệu và giải thuật sinh dữ liệu:](#_eccu3kltuozo)  **12**

[2.1 Mô hình dữ liệu đã thiết kế (mô tả bằng ER diagram ):](#_yqqnp1520l5m) 12

[2.2, Giải thuật sinh dữ liệu:](#_stcdwlg2d4l7)  12

[2.3,Ví dụ:](#_cr77m2diylj3)  14

[3, Cấu trúc chương trình:](#_rbf0oyux3wtr)  **15**

[3.1,Package Helper:](#_ef33ji9lreo1)  15

[Là package chứa các lớp thực hiện các nhiệm vụ chuyên biệt (thao tác với cơ sở dữ liệu, đọc file …), được import trực tiếp tự lớp Oop chứa hàm main .](#_4ceqd5ey9y46) 15

[3.1.1,Lớp DatabaseHelper:](#_cd5pqd5drk9a)  15

[3.1.2,Lớp EntityGeneratorHelper:](#_6t0ojf8710st)  16

[3.1.3,Lớp RelationGeneratorHelper,](#_1yl3e9rctykr)  17

[3.1.4,Lớp ReadSQLFilerHelper:](#_jhgxf6ogn4p0)  18

[3.2,Package Model](#_76dn9erl5pd1) 18

[Chứa lớp Fact, lớp Entity và các lớp con của Entity bao gồm : Person,Article,Time, Event,Country, Aggrement,Organization,Location (8 lớp con)](#_hpqwixqk36vm) 18

[3.2.2,Lớp Entity:](#_lmuxbmwlg9mg)  18

[3.2.3,Các lớp con của lớp Entity,](#_f00dwyikoryu) 18

[3.2.4,Lớp Fact](#_u9lzu2deap8) 19

[3.3,Package oop](#_pew5a4oklymk) 20

[3.4,thư mục data](#_m9ya45slogf4) 21

[4,Các biểu đồ của chương trình](#_wwopwspxz77v) **21**

[4.1,Biểu đồ package:](#_scry0fxa3acb)  21

[4.2,Biểu đồ lớp :](#_d3a6fxgl063)  21

[**5,Kiểm thử hiệu năng chương trình:**](#_54ukdpinn9xg)  **24**

[5.1,Nội dung các câu truy vấn cơ bản:](#_10vh83hy2k6a)  24

[5.2, Nội dung các câu truy vấn thống kê,nâng cao:](#_jeaqkry4lf48)  25

[5.3, Đo thời gian chuẩn bị dữ liệu và thời gian truy vấn với từng testcase:](#_yu6razsnxtyw)  27

[6,Đánh giá ưu nhược điểm:](#_gko7ea87uei3)  **28**

[6.1, Đánh giá ưu nhược điểm của chương trình:](#_t2ybkuea3ydw)  28

[6.1.1, Ưu điểm:](#_3j6tn3lljl65)  28

[6.1.2, Nhược điểm:](#_i29hbdb5lqvl)  28

[6.2,Đánh giá ưu nhược điểm của công nghệ sử dụng (Mysql);](#_i3cym8mw58xy) 29

[6.2.1 Ưu điểm](#_55aa6qowhrcu) 29

[6.2.2 Nhược điểm](#_ot713qoe40bj) 30

# 1. Tìm hiểu công nghệ:

## 1.1. Giới thiệu MySQL:

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng.

Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows,Linux,Mac OS,....

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL). MySQL được sử dụng cho việc bổ trợ PHP,Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,…

## 1.2. Các thuật ngữ:

- Database (cơ sở dữ liệu): Là một tập thông tin có tổ chức, sao cho việc truy cập, quản lý và kết xuất nó được thực hiện một cách dễ dàng.

- Relational database (cơ sở dữ liệu quan hệ): Là một tập các đối tượng dữ liệu được tổ chức theo dạng thức bảng (table),

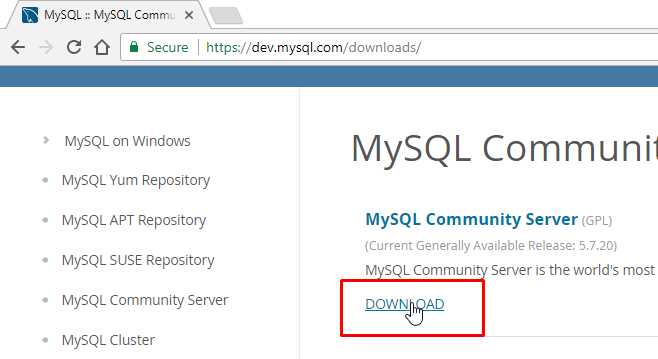
- Table (bảng): mỗi bảng lưu thông tin về các đối tượng(dưới dạng các bản ghi ) của cùng một loại thực thể, quan hệ nào đó, trong đó các đối tượng có cùng số lượng, tên thuộc tính, ở đây gọi là các trường .Các bản ghi tương ứng với các hàng trong khi các trường tương ứng với các cột trong bảng .

- Primary key (khóa chính): tập các cột mà mỗi tập các giá trị của chúng định danh một bản ghi duy nhất trong bảng.

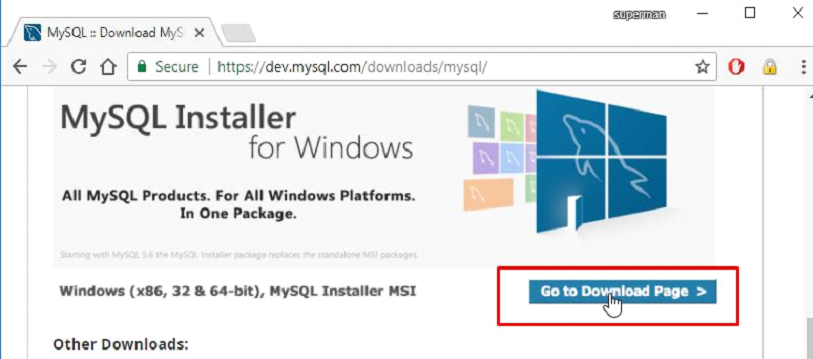
- Foreign key (khóa ngoài): dùng để ràng buộc tham chiếu dữ liệu giữa hai bảng với nhau.

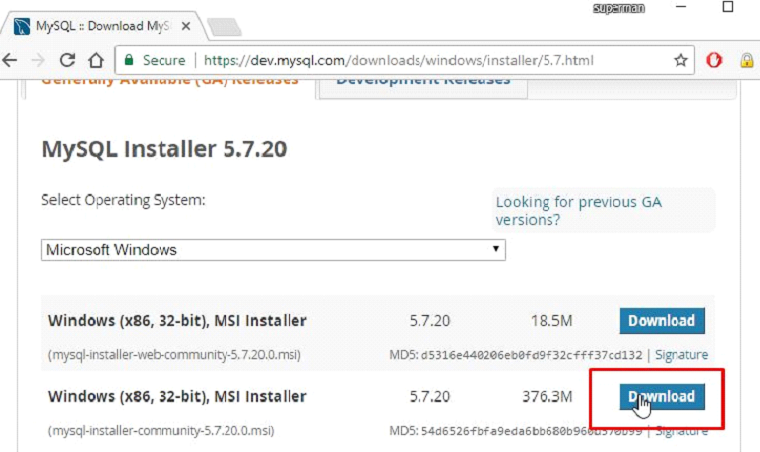
## 1.3. Hướng dẫn cài đặt MySQL

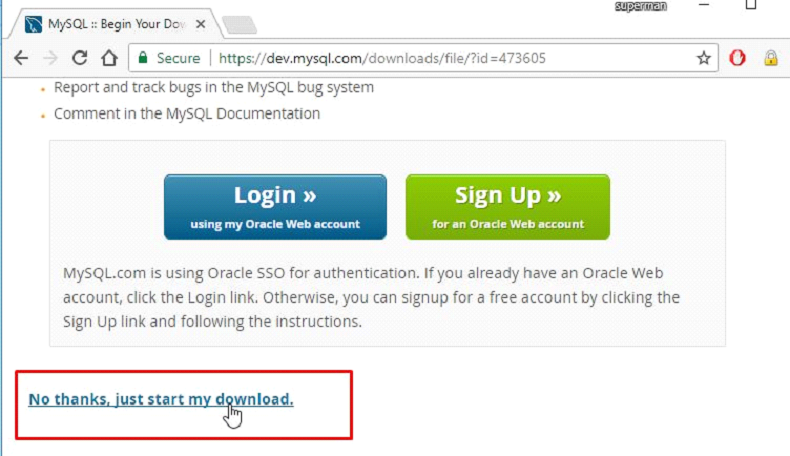
-Bước 1: Truy cập đường link để tải bản MySQL community <https://dev.mysql.com/downloads>



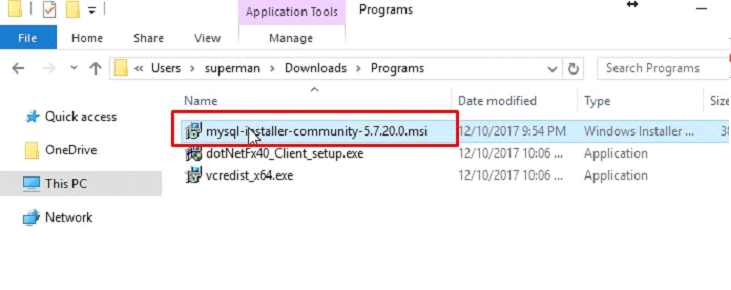
Bước 2: Lựa chọn file cài phù hợp với máy:



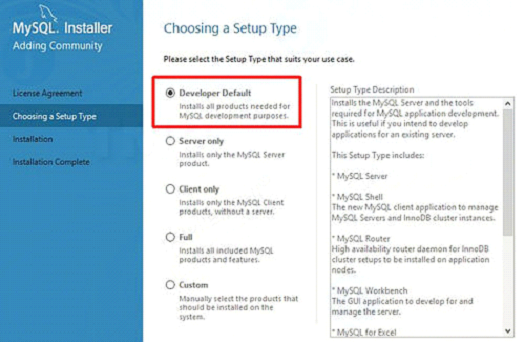




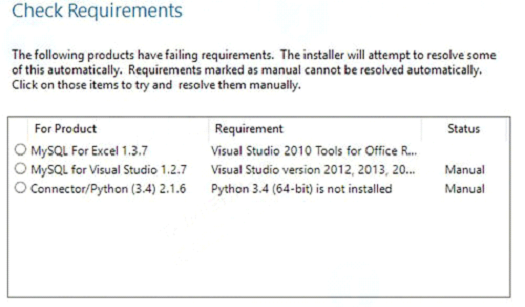
Bước 3: Kích đúp vào file vừa tải về:



Bước 4: Chọn chế độ cài đăt: "Developer Default ":

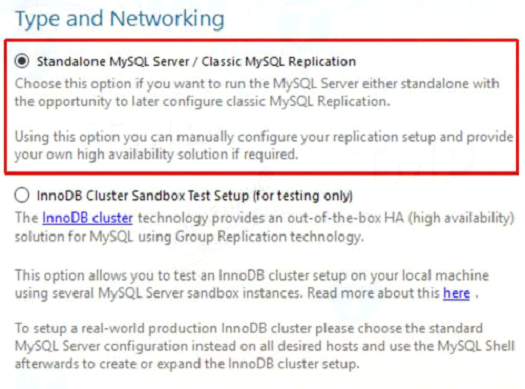


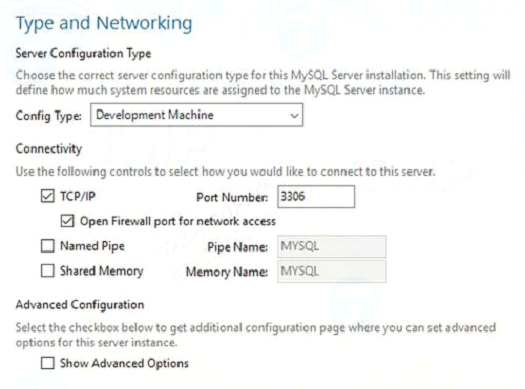
Bước 5: Cài đặt các sản phẩm bổ trợ:



Bước 6: Thiết Lập Máy Chủ MySQL

Trên cửa sổ cài đặt bao gồm các thiết lập cho máy chủ SQL. Chọn các thiết lập chính xác cho máy chủ thử nghiệm là bước quan trọng. Đối với loại máy chủ này, bạn chọn Standalone MySQL Server / Classic MySQL Replication.

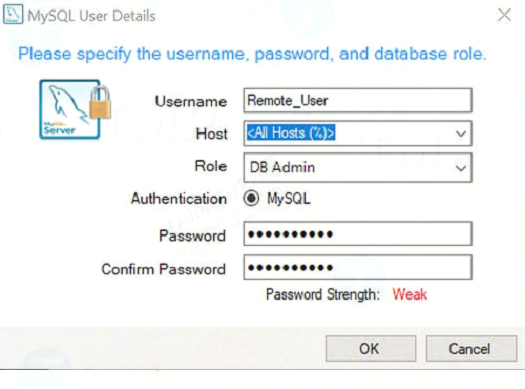




Tiếp theo với muc Server Configuration Type, bạn chọn Development Machine. Giữ nguyên các thiết lập mặc định của Connectivity là TCP/IP port 3306.

Bước 7: Thiết lập tài khoản, mật khẩu truy cập:

.



Bước 8: Tiếp theo là giữ nguyên cả Configure MySQL Server as a Windows Service và Start the MySQL Server at System Startup đã được kích hoạt. Chạy dịch vụ trong Standard System Account. Click chọn Next trên các cửa sổ tiếp theo, cuối cùng chọn Execute để hoàn tất quá trình thiết lập.

Chú ý: MySQL Workbench là phần mềm chúng ta có thể thao tác trực tiếp với cơ sở dữ liệu cũng như viết các truy vấn,cấu hình ....

## 1.4. Các kỹ thuật sử dụng:

### 1.4.1. Đọc và lấy dữ liệu từ file:

File JSON (từ JavaScript Object Notation) là một kiểu định dạng dữ liệu trong đó sử dụng văn bản thuần tuý, định dạng JSON sử dụng các cặp key - value để dữ liệu sử dụng. Ở đây, ta lưu các tập dữ liệu mẫu trong các file JSON. Trình tự lấy dữ liệu mẫu như sau (dùng thư viện simple-json-1.1.jar):

+ Tạo một đối tượng FileReader(path)- path là đường dẫn tương đối của file, để đọc nội dung file.

+ Sau đó chuyển đổi nội dung file thành một đối tượng JSON, ví dụ sampleData=(JSONObject) (new JSONParser()).parse( new FileReader("src/sampleEntityData.json"));

File SQL: lưu các câu truy vấn, ta sẽ đọc nội dung của nó thành các mảng String dùng Scanner trong java.

### 1.4.2. Kỹ thuật xử lý theo lô: (Batch processing)

Để giảm thời gian thực thi, ta sẽ thêm các câu truy vấn vào cùng một lô với phương thức addBatch(query) của lớp Statement . Sau đó thực hiện toàn bộ lô đó với phương thức executeBatch(), cuối cùng xóa hết câu truy vấn trong lô với clearBatch() .

## 1.5. Cách truy vấn và giao tiếp với cơ sở dữ liệu bằng API:

-Trình tự thực hiện tạo kết nối và truy vấn từ chương trình tới database (dùng Driver JDBC):

+Đăng ký driver của Mysql (Class.forName(driverName))

+Tạo một connection đến database thông qua hàm DriverManager.getConnection(url, username,password ), trong đó url là đường dẫn đến database còn username, password để xác thực đăng nhập .

+Tạo một đối tượng Statement (thông qua connection vừa tạo)

+Bắt đầu truy vấn bằng các phương thức của statement vừa tạo

## 

## 1.6.Mô hình dữ liệu trong MySQL

Mô hình dữ liệu ở đây là mô hình dữ liệu quan hệ (Relational Data Model ), dựa trên những khái niệm về toán học trên nền tảng lý thuyết và tập hợp.

Cấu trúc: Dữ liệu được tổ chức dưới dạng các quan hệ hay các bảng, trong đó:

+Các thuộc tính của quan hệ: các cột trong bảng .

+Các đối tượng của quan hệ: các hàng trong bảng

+Có thể bao gồm khóa chính để phân biệt các đối tượng trong bảng

+Có thể bao gồm khóa ngoại để ràng buộc dữ liệu với các bảng khác .

# 2.Mô hình dữ liệu và giải thuật sinh dữ liệu:

## 2.1 Mô hình dữ liệu đã thiết kế (mô tả bằng ER diagram):

## 

## 2.2. Giải thuật sinh dữ liệu:

Trên lý thuyết, mỗi bảng trên đều phải có khóa chính để định danh cho các bản ghi, đồng thời phải có khóa ngoài để tham chiếu đến bảng khác, thể hiện sự liên kết, thống nhất của cơ sở dữ liệu

Tuy nhiên, mô hình dữ liệu trên không dùng khóa trong,khóa ngoài vì:

+Thời gian thực thi rất lâu do mỗi khi thêm một bản ghi mới lại phải kiểm tra toàn bộ bản ghi trong cùng bảng (với khóa trong) hay các bản ghi trong bảng khác (với khóa ngoài)

+Cấu trúc dữ liệu và giải thuật sinh dữ liệu dưới đây đảm bảo được sự phân biệt của các bản ghi trong mỗi bảng bằng trường id cũng nhưng sự gắn kết giữa các bảng khi một số trường của bản này có liên hệ với trường id của bảng khác (ví dụ trường leader của bảng Organization có liên hệ với trường id của bảng Person)

**\*Giải thuật sinh dữ liệu:**  Với yêu cầu tạo N thực thể, M quan hệ: Vì có 8 loại thực thể (Person,Country, Event, .....) và 11 loại quan hệ (TakePlace, Meet,Contract, .....)nên để dễ kiểm soát, mỗi loại thực thể ta sẽ tạo N1=N/8 (đối tượng), mỗi một loại quan hệ tạo M1=M/11 (đối tượng ):

Với các lớp con của lớp Entity, dù chúng có bao nhiêu thuộc tính (hay trường trong bảng tương ứng ),ta đều chia làm 3 loại:

+Thuộc tính id:

* Định danh cho từng đối tượng của lớp đó, trường id có ý nghĩa như khóa chính của bảng vậy .
* Giá trị: <Tên lớp > +<Số thứ tự của đối tượng được tạo >
* Ví dụ : đối tượng Person đầu tiên có id là “Person0”.

+Các thuộc tính có kiểu là các lớp khác:

* Ví dụ: lớp Organization có thuộc tính là leader với kiểu là Person
* Thế hiện tính kết tập của chương trình, các trường như vậy có ý nghĩa như khóa ngoài tham chiếu đến các bảng khác vậy .
* Giá trị:

Trong chương trình: là một đối tượng của lớp được kết tập (trong ví dụ là một đối tượng Person) .

Còn trong cơ sở dữ liệu: là id của đối tượng trong bảng tương ứng (trong ví dụ là id của một bản ghi nào đó trong bảng Person ), được tạo ngẫu nhiên bằng cấu trúc: <Tên lớp kết tập > + <Random( M1)> , ví dụ “Person123” .

+Các thuộc tính còn lại: ví dụ như name, description, ....

* Ví dụ: Lớp Person có thuộc tính name là tên người, lớp Country có thuộc tính population là dân số của quốc gia đó..
* Là các thuộc tính bổ sung cho đối tượng
* Giá trị: mỗi thuộc tính loại này của mỗi lớp đều có một tập giá trị mẫu, ta sẽ lấy ngẫu nhiên một phần tử trong tập này .

Với các đối tượng của lớp Fact: ta cũng chia làm 3 loại thuộc tính

* relationId:
* Định danh cho từng đối tượng của lớp Fact
* Cấu trúc: <Tên quan hệ >+<Số thứ tự đối tượng được tạo ứng với quan hệ đó >
* Ví dụ: Đối tượng biểu diễn quan hệ gặp gỡ được tạo đầu tiên có relationId là: ”Meet0”

+articleId,time:

* Cung cấp thông tin về bài báo và thời gian trích xuất ra fact này .
* Có ý nghĩa như khóa ngoài tham chiếu 2 bảng Article và Time
* Cấu trúc:

articleId: ”Article”+<Random(M1)>, ví dụ “Article12”

time: “Time”+<Random(M1)> , ví dụ “Time10”

+subjectId,objectId:

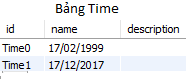
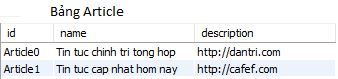
* Định danh của 2 đối tượng tham gia quan hệ
* Ý nghĩa như khóa ngoài tham chiếu đến 2 bảng subjectType, objectType
* Ví dụ: quan hệ “Organize10” có subjectId là “Person123”, có objectId là “Event345”

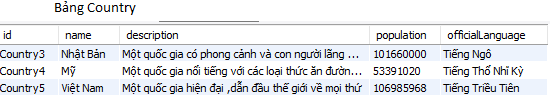
+subjectType,objectType:

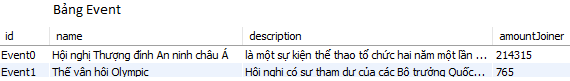
* Cung cấp tên bảng của 2 đối tượng tham gia quan hệ
* Ví dụ: quan hệ căng thẳng- Bargain có subjectType: Country, objectType: Country
* Dùng để xác định lớp của đối tượng nếu đóng gói dữ liệu từ cơ sở dữ liệu gửi về

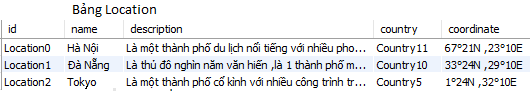
## 2.3. Ví dụ:

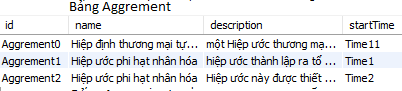
Một vài bản ghi từ các bảng như sau:

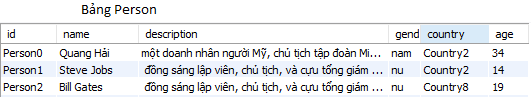


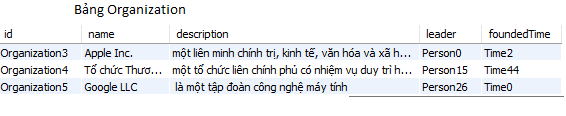


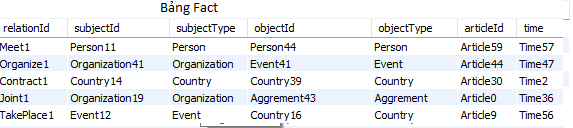












# 

# 3. Cấu trúc chương trình:

## 3.1. Package Helper:

### Là package chứa các lớp thực hiện các nhiệm vụ chuyên biệt (thao tác với cơ sở dữ liệu, đọc file …), được import trực tiếp tự lớp Oop chứa hàm main .

### 3.1.1. Lớp DatabaseHelper:

Là lớp làm việc trực tiếp với cơ sở dữ liệu MySQL

Thuộc tính quan trọng:

* **private Connection conn**: đối tượng để thực hiện kết nối tới database ;
* **private Statement statement**: đối tượng để thực hiện gửi các truy vấn lên database ;

Phương thức quan trọng:

* **public void createConnectionDB( url, username, password ):** thiết lập kết nối tới cơ sở dữ liệu với đường dẫn url, xác thực bằng username với password của người dùng .Sau phương thức này ta sẽ có một đối tượng Statement để thực thi các truy vấn lên cơ sở dữ liệu đã kết nối .
* **public void executeBatchQuery( queryList,showExecuteTime ):** tham số truyền vào là mảng các String, mỗi String là một truy vấn, biến showExecuteTime xác định có in ra màn hình thời gian thực thi cho cả lô truy vấn hay không, đây là hàm thực thi các truy vấn theo lô với các phương thức của Statement lần lượt như: addBatch(query), executeBatch(), clearBatch();
* **public void executeSeparateQuery (queryList, showExecuteTime, queryType):** tham số truyền vào là mảng các truy vấn đơn, biến logic showExecuteTime để xác định có in ra màn hình thời gian của từng truy vấn hay không, biến queryType cho biết loại truy vấn là truy vấn cập nhật hay truy vấn trả về dữ liệu (tương ứng executeUpdate(query) hay executeQuery(query)

### 3.1.2,Lớp EntityGeneratorHelper:

Là lớp thao tác với các lớp con của lớp Entity:

Các thuộc tính quan trọng: :

* **private String className**  : tên của lớp đang xét .(Person, Country,...)
* **private Field[] fieldList**  : danh sách các trường của lớp className
* **private JSONArray fieldValueSet** : danh sách các tập giá trị mẫu của các trường
* **private String subQuery**  : là truy vấn dùng để insert tất cả các bản ghi được sinh ra của lớp className vào bảng tương ứng
* **private ArrayList<String>queryList:** là mảng gồm các truy vấn subQuery ứng với từng lớp khác nhau .

Phương thức quan trọng:

* **public ArrayList<String> generateEntity(totalEntity):** tham số truyền vào là tổng số thực thể cần tạo N, hàm sẽ trả ra một mảng các truy vấn, từng truy vấn sẽ thực hiện thêm N1 bản ghi vào từng bảng tương ứng (ví dụ truy vấn 1 thêm N1 bản ghi Person vào bảng Person, truy vấn 2 thêm N1 bản ghi Country vào bảng Country, ....) (N1=N/(8 lớp ))

Phương thức generationEntity thực hiện như sau: tính toán số lượng đối tượng(bản ghi) của mỗi lớp (có 8 lớp: Person,Country, ....) - N1 . Sau đó xét từng lớp (tên lớp lưu vào thuôc tính className) thực hiện gọi:

* **void getFieldList():** lấy danh sách các trường của một lớp className, bao gồm cả các trường từ lớp cha Entity, cho vào thuộc tính fieldList
* **void getFieldValueSet():** Lấy danh sách các tập giá trị mẫu của từng trường trên (được quy định ở file sampleEntityData.json), lưu vào thuộc tính fieldValueSet, với fieldValueSet[i] là tập giá trị mẫu của trường fieldList[i] . Với những trường là id hoặc có kiểu là các lớp khác, fieldValueSet[i] =null .
* **void prepareSubQuery():** khởi tạo subQuery:

“Insert into “+<tên lớp”+(<fieldList[0]>,<fieldList[1]>,....) values “

Sau đó lặp lại N1 lần,mỗi lần gọi hàm dưới đây rồi ghép bản ghi được tạo vào subQuery

* **String createRecord(objectIndex) :** nhận về số thứ tự của đối tượng thuộc lớp className, trả về một xâu là nội dung bản ghi được tạo ứng với đối tượng đó, ví dụ (“Country1”,”Việt Nam”,”Một đất nước tươi đẹp”)

Sau khi xét xong một lớp, thêm subQuery ứng với lớp đó vào mảng queryList. Duyệt xong hết các lớp, trả ra queryList.

### 3.1.3,Lớp RelationGeneratorHelper,

Là lớp thao tác với lớp Fact ,

Các thuộc tính quan trọng:

* **private String relationName** : tên của loại quan hệ đang xét .
* **private Field[] fieldList :** danh sách các trường của lớp Fact
* **private JSONArray fieldValueSet** : danh sách các tập giá trị mẫu của các trường
* **private String subQuery** : lưu truy vấn dùng để insert tất cả các bản ghi của loại quan hệ relationName
* **private ArrayList<String>queryList** : là mảng gồm các truy vấn (ứng với tất cả các loại quan hệ )

Phương thực quan trọng:

* **public ArrayList<String> generateRelation(totalRelation):** tham số truyền vào là tổng số quan hệ cần tạo M, hàm sẽ trả ra một mảng các truy vấn, từng truy vấn sẽ thực hiện thêm M1 bản ghi ứng với từng loại quan hệ vào bảng fact (ví dụ truy vấn 1 thêm M1 bản ghi ứng với quan hệ “Meet” vào bảng Fact, truy vấn 2 thêm M1 bản ghi ứng với quan hệ “Organize” vào bảng Fact,....) (M1 = M/ (11 quan hệ ))

Phương thức generateRelation thực hiện như sau: tính toán số lượng đối tượng(bản ghi) của mỗi loại quan hệ (có 11 loại: Meet,Organize,....) .Sau đó xét từng loại quan hệ (quan hệ đang xét lưu vào thuôc tính relationName) thực hiện gọi:

* **private void getFieldList():** lấy danh sách các trường của lớp Fact, cho vào thuộc tính fieldList
* **private void getFieldValueSet():** Lấy danh sách các tập giá trị mẫu của 2 trường subjectType, objectType ứng với relationName (trong file sampleRelationData.json)

V í dụ: với relationName = Meet , 2 tập giá trị là [“Person”],[“Person”]

Với relationNam e= Support, 2 tập giá trị lại là: [“Person”,”Country”], [“Country”,”Aggrement”,”Event”]

* **private void prepareSubQuery():** khởi tạo subQuery:

“Insert into “+Fact+(<fieldList[0]>,<fieldList[1]>,....) values “

Sau đó, lặp lại M1 lần, mỗi lần gọi hàm dưới đây rồi ghép bản ghi được tạo vào subQuery:

* **private String createRecord(objectIndex) :** nhận vào số thứ tự bản ghi của relationName, trả về một xâu là nội dung bản ghi được sinh, ví dụ kết quả trả về với objectIndex =12, relationName= Support: (“Support12”,”Person4”,”Person”,”Event6”,”Event”,”Article20”,”Time2”) ;

Sau khi xét xong một loại quan hệ, thêm subQuery ứng với loại quan hệ này vào mảng queryList .Sau khi xét hết các loại quan hệ, trả ra queryList

### 3.1.4. Lớp ReadSQLFilerHelper:

Là lớp thao tác với các file SQL

Thuộc tính quan trọng: Không có thuộc tính nào

Phương thức quan trọng:lớp cung cấp một phương thức public duy nhất:

* **ArrayList<String> getQueryList(fileName):** tham số truyền vào là tên file sql trong folder data , hàm sẽ xử lý và trả ra một dãy các String, mỗi String là một truy vấn đơn lấy từ file sql đó .

## 3.2. Package Model

### Chứa lớp Fact, lớp Entity và các lớp con của Entity bao gồm : Person,Article,Time, Event,Country, Aggrement,Organization,Location (8 lớp con)

### 3.2.1. Lớp Entity:

Chỉ bao gồm 3 thuộc tính chính:

* private String id;
* private String name ;
* privateString description ;

Và các getter, setter mặc định (getId(),setName(),....)

### 3.2.2. Các lớp con của lớp Entity,

Là các lớp lưu dữ liệu ứng với từng loại thực thể, kế thừa từ lớp Entity với 3 thuộc tính mặc định, ngoài ra còn có thêm các thuộc tính tùy từng lớp cụ thể và các getter, setter mặc định tương ứng với các thuộc tính ấy:

+Time:

+Article:

+Country:

* private Integer population; //dân số của nước đó
* private String officialLanguage; //ngôn ngữ chính thức

+Event:

* private Integer amountJoiner; // số người tham dự sự kiện

+Location:

* private Country country; //quốc gia chứa địa điểm đó, trong database lưu id quốc gia
* private String coordinate; //tọa độ của địa điểm

+Aggrement:

* private Time startTime //thời điểm có hiệu lực, trong database lưu id thời gian (vd: Time1,..)

+Person:

* private String gender; //giới tính của người đó
* private Country country; //quốc tịch, trong database lưu id quốc gia
* private Integer age ; // tuổi của người đó

+Organization:

* private Person leader; //người lãnh đạo của tổ chức, trong database lưu id người
* private Time foundedTime; //thời điểm sáng lập, trong database lưu id thời gian

### 3.2.3. Lớp Fact

Là lớp lưu dữ liệu của các fact được trích xuất, bao gồm các thuộc tính:

* private String subjectId;
* private String objectId;
* private String subjectType;
* private String objectType;
* private String relationId;
* private String articleId;
* private String time;

Và các getter, setter tương ứng .Khi cần lấy thông tin cụ thể về subject hay object ta sẽ truy vấn lên các bảng subjectType và objectType theo định danh subjectId và objectId trong cơ sở dữ liệu,

## 3.3. Package oop

Mô tả: là nơi chứa lớp oop -lớp chứa hàm main:

3.3.1. Lớp oop:

Các thuộc tính quan trọng:

* **static final int[] totalEntity**={100,2000,20000,200000,1000000}: quy định số lượng thực thể (N) cho mỗi testcase
* **static final int[] totalRelation**={200,3000,40000,400000,2000000}:quy định số lượng quan hệ(M) cho mỗi testcase
* **static DatabaseHelper databaseHelper**  =new DatabaseHelper(): khởi tạo một đối tượng chuyên làm việc với database ;
* **static ReadSQLFilesHelper readSQLFileHelpe**r =new ReadSQLFilesHelper(): khởi tạo đối tượng chuyên làm việc với các file SQL
* **static EntityGeneratorHelper entityGeneratorHelper** =new EntityGeneratorHelper():khởi tạo đối tượng phụ trách việc sinh dữ liệu cho các thực thể và chuyển thành các câu lệnh SQL.
* **static RelationGeneratorHelper relationGeneratorHelper** =new RelationGeneratorHelper() khởi tạo đối tượng phụ trách việc sinh dữ liệu cho các quan hệ và chuyển thành các câu lệnh SQL.

Các phương thức quan trọng:

* **public void test (int testIndex):**thực hiện tất cả các công việc từ:
* Tạo database và tạo các bảng:
* Ban đầu đối tượng databaseHelper thực hiện phương thức createConnectionDB để tạo kết nối tới database .
* Sau đó, đối tượng readSQLFileHelper gọi phương thức getQueryList để đọc fle “initDatabase.sql”, trả về một mảng các truy vấn đơn
* Đối tượng databaseHelper gọi phương thức executeSeparateQuery với đầu vào là mảng đó, thực thi các câu lệnh để khởi tạo database;

+Sinh dữ liệu và thêm vào database:

* Ban đầu đối tượng entityGeneratorHelper gọi phương thức generateEntity với đầu vào là số lượng thực thể cần tạo N , thực thi để trả ra mảng các truy vấn đơn với nhiệm vụ là thêm N bản ghi thực thể vào database
* Sau đó, databaseHelper thực hiện phương thức executeBatchQuery nhận tham số vào là mảng các truy vấn trên để thêm các bản ghi vào
* Lặp lại thế với đối tượng relationGeneratorHelper với databaseHelper;

+Kiểm tra hiệu năng bằng các truy vấn:

* Ban đầu đối tượng readSQLFile thực hiện phương thức getQueryList để đọc file “primaryQueries.sql”, trả ra mảng các truy vấn đơn(10 truy vấn cơ bản )
* Sau đó, databaseHelper thực hiện phương thức executeSeparateQuery nhận tham số vào là mảng trên để thực thi lần lượt các truy vấn .
* Lặp lại như vậy với file “advancedQueries.sql”

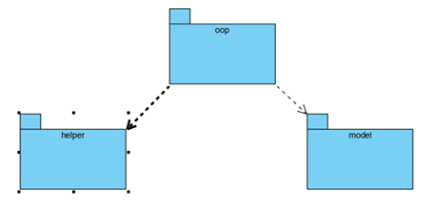
## 3.4. thư mục data

Chứa các file data như:

* initDatabase.sql: chứa các câu lệnh để tạo database và tạo các bảng cho database đó .
* primaryQueries.sql:chứa 10 truy vấn cơ bản
* advancedQueries.sql:chứa 10 truy vấn thống kê,nâng cao
* sampleEntityData.json: chứa tập giá trị mẫu của các trường thuộc các lớp Entity
* sampleRelationData.json: chứa tập các lớp của subject, ọbject ứng với các loại quan hệ

# 4. Các biểu đồ của chương trình

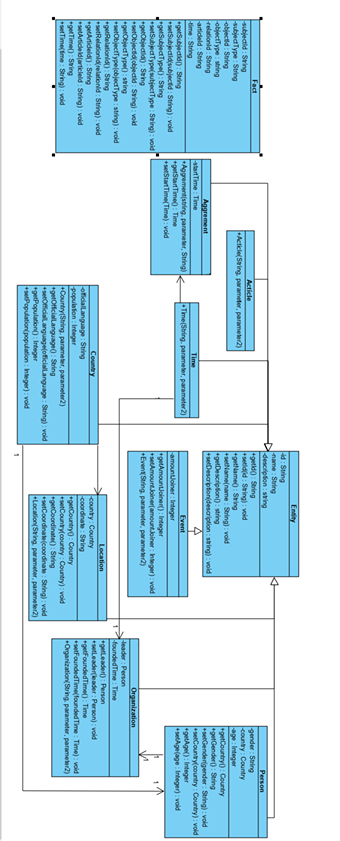
## 4.1. Biểu đồ package:



## 4.2. Biểu đồ lớp :

## 

## 



# 5. Kiểm thử hiệu năng chương trình:

Hàm main sẽ lần lượt thực hiện 5 test case với (N,M) tăng dần qua gọi phương thức test của lớp oop, kết quả sau đó được in ra màn hình .

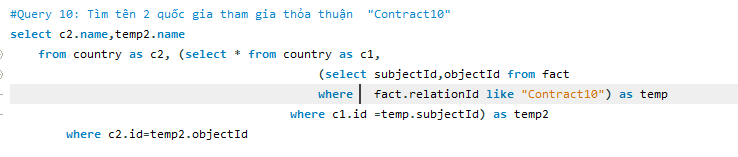
## 5.1. Nội dung các câu truy vấn cơ bản:

Mức 1: Loại truy vấn trong một bảng để lấy thông tin từ bản ghi có định danh cho trước:



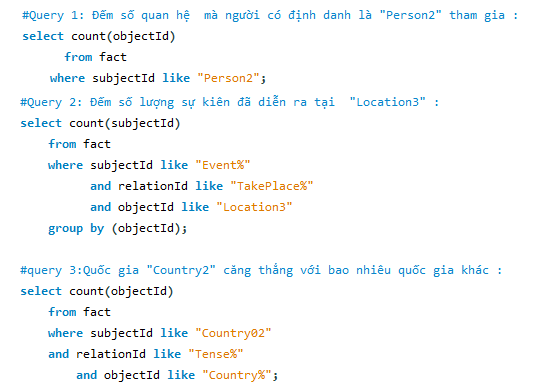
Mức 2: Truy vấn trong 2 bảng: Lấy thông tin từ bản ghi nào đó có một trường là “khóa ngoài” của một trường thuộc bảng khác:

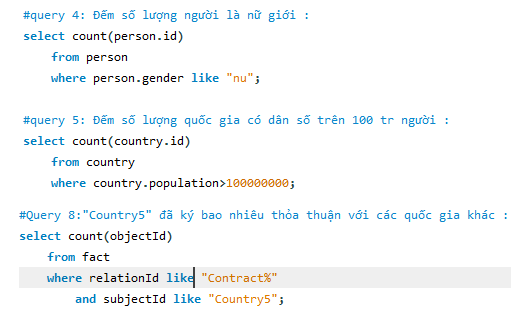


Mức 3: Truy vấn trong 3 bảng: Lấy các thông tin từ 2 bảng mà có trường là “khóa ngoài “ của các trường thuộc bảng thứ 3: 

## 5.2. Nội dung các câu truy vấn thống kê,nâng cao:

Mức 1: Truy vấn trong một bảng: Đếm số lượng bản ghi thỏa mãn một tính chất nào đó:





Mức độ 2: Truy vấn trong 2,3 bảng và hiển thị danh sách các thông tin bản ghi thỏa mãn:



Vì số lượng bản ghi tăng dần lên rất lớn nên các truy vấn trong 2,3 bảng cần tối ưu để có thể rút ngắn thời gian thực thi chương trình .

## 5.3, Đo thời gian chuẩn bị dữ liệu và thời gian truy vấn với từng testcase:

Bảng 1:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Truy vấn cơ bản(ms)  (N,M) | Sinh và thêm dữ liệu | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 100,200 | 12957 | 0 | 20 | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 9 |
| 2000,3000 | 28972 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 |
| 20000,40000 | 62970 | 65 | 13 | 0 | 8 | 8 | 13 | 0 | 156 | 12 | 21 |
| 200000,400000 | 172668 | 119 | 39 | 1 | 115 | 165 | 212 | 105 | 388 | 54 | 271 |
| 1000000,2000000 | 704344 | 1390 | 835 | 743 | 955 | 1611 | 1214 | 987 | 5140 | 1645 | 4193 |

Bảng 2:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Truy vấn nâng cao (ms)  (N,M) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 100,200 | 0 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2000,3000 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20000,40000 | 2 | 2 | 2 | 7 | 0 | 4 | 17 | 3 | 2 | 4 |
| 200000,400000 | 117 | 129 | 168 | 74 | 145 | 299 | 674 | 108 | 409 | 431 |
| 1000000,2000000 | 3092 | 6039 | 5326 | 318 | 391 | 3842 | 1803 | 1251 | 1062 | 2853 |

# 6. Đánh giá ưu nhược điểm:

## 6.1. Đánh giá ưu nhược điểm của chương trình:

## 6.1.1, Ưu điểm:

-Chương trình có hiệu năng cao, thời gian sinh dữ liệu giả lập dưới dạng các bản ghi và tạo truy vấn insert từ nhiều bản ghi một lúc có tốc độ thực thi rất nhanh (tạo 1000000 bản ghi trong vài giây ),tuy nhiên thời gian ở bước gửi truy vấn lên database và thực hiện chiếm thời gian rất lớn, có thể do tốc độ đọc ghi, truyền tải dữ liệu trong máy còn kém .

- Chương trình ngắn gọn, với các module được viết để thực hiện các nhiệm vụ chuyên biệt, không chồng lấn với nhau, đầu ra của module này là đầu ra của module kia, đảm bảo kiểm soát lỗi tốt khi có vấn đề xảy ra.

- Các tệp dữ liệu mẫu lưu dưới định dạng JSON giúp số lượng file dùng để lưu hết dữ liệu mẫu rất ít . -Dễ mở rộng do khi cần thêm một thuộc tính nào đó vào các lớp con của lớp Entity chỉ cần thêm khi báo thuộc tính vào lớp đó,thêm trường vào bảng tương ứng và khai báo tập giá trị mẫu của nó, không cần chỉnh sửa bất cứ hàm nào cả ..

### 6.1.2. Nhược điểm:

Do viết một hàm chung để tạo dữ liệu cho các lớp Entity nên ý tưởng hơi khó hiểu, dễ gây hiểu lầm, ngoài ra không tùy biến được số lượng đối tượng muốn tạo của từng lớp cụ thể

Vẫn còn chậm ở phần gửi truy vấn và thực hiện truy vấn trên cơ sở dữ liệu

## 6.2. Đánh giá ưu nhược điểm của công nghệ sử dụng (Mysql);

### 6.2.1 Ưu điểm

* Dễ sử dụng: MySQL là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh, sự linh hoạt về platform là 1 đặc tính nổi bật của MySQL với các phiên bản đang được hỗ trợ của Linux, Unix, Windows, MySQL cho phép tùy biến hoàn toàn theo ý muốn, thêm vào các yêu cầu thích hợp cho database server.
* Thực thi cao: Các chuyên gia cơ sở dữ liệu có thể cấu hình máy chủ cơ sở dữ liệu MySQL đặc trưng cho các ứng dụng đặc thù thông qua kiến trúc storage-engine. MySQL có thể đáp ứng khả năng xử lý những yêu cầu khắt khe nhất của từng hệ thống, MySQL còn đưa ra các “công cụ” cần thiết cho các hệ thống doanh nghiệp khó tính bằng tiện ích tải tốc độ cao, bộ nhớ cache và các cơ chế xử lý nâng cao khác.
* Sử dụng ngay: Các tiêu chuẩn đảm bảo của MySQL giúp cho người dùng vững tin và chọn sử dụng ngay, MySQL đưa ra nhiều tùy chọn và các giải pháp để người sử dụng dùng ngay cho server cơ sở dữ liệu MySQL
* Hỗ trợ giao dịch: MySQL hỗ trợ giao dịch mạnh 1 cách tự động, thống nhất, độc lập và bền vững, ngoài ra khả năng giao dịch cũng được phân loại và hỗ trợ giao dịch đa dạng mà người viết không gây trở ngại cho người đọc và ngược lại. Các dữ liệu được đảm bảo toàn vẹn trong suốt quá trình server có hiệu lực và các mức giao dịch độc lập được chuyên môn hóa cao.
* Nơi tin cậy để lưu trữ web và dữ liệu: Do MySQL có engine xử lý tốc độ cao và khả năng chèn dữ liệu nhanh, hỗ trợ tốt cho các chức năng chuyên dùng cho web,…nên MySQL là lựa chọn tốt nhất cho các ứng dụng web và các ứng dụng web doanh nghiệp.
* Bảo mật tốt: Doanh nghiệp nào cũng cần tính năng bảo mật dữ liệu tuyệt đối vì đó chính là lợi ích quan trọng hàng đầu, và đó cũng là lý do mà các chuyên gia về cơ sở dữ liệu chọn dùng MySQL. MySQL có các kỹ thuật mạnh trong việc xác nhận truy cập cơ sở dữ liệu và chỉ có người dùng đã được xác nhận mới có thể truy cập vào server cơ sở dữ liệu. Bên cạnh đó, tiện ích backup và recovery cho phép backup logic và recovery toàn bộ hoặc tại 1 thời điểm nào đó.
* Phát triển ứng dụng hỗn hợp: MySQL cung cấp hỗ trợ hỗn hợp cho bất kỳ sự phát triển ứng dụng nào nên MySQL được xem là cơ sở dữ liệu mã nguồn mở phổ biến nhất thế giới. Thư viện plugin có sẵn để nhúng vào cơ sở dữ liệu MySQL hỗ trợ trong bất kỳ ứng dụng nào. MySQL còn cung cấp các bộ kết nối cho phép tất cả các form của ứng dụng ưu tiên sử dụng MySQL như 1 server quản lý dữ liệu.
* Dễ quản lý: Quá trình cài đặt MySQL diễn ra khá nhanh chóng trên Microsoft Windows, Linux, Macintosh hoặc Unix. Sau khi cài đặt, các tính năng tự động mở rộng không gian, tự khởi động lại và cấu hình động được thiết lập sẵn sàng cho người quản trị cơ sở dữ liệu làm việc.
* Nhiều công cụ hữu ích: MySQL còn có các công cụ quản lý đồ họa mà 1 DBA có thể quản lý, sửa chữa và điều khiển hoạt động của nhiều server, điều khiển tác vụ thiết kế dữ liệu và ETL, quản trị cơ sở dữ liệu hoàn thiện cũng như quản lý công việc và thực hiện kiểm tra.
* Mã nguồn mở tự do và hỗ trợ xuyên suốt: Nhiều doanh nghiệp lo lắng việc sử dụng mã nguồn mở là không an toàn và không được hỗ trợ tốt vì đa số tin vào các phần mềm có bản quyền, nhưng đối với MySQL, các nhà doanh nghiệp hoàn toàn có thể yên tâm về điều này, MySQL có chính sách bồi thường hẳn hoi và luôn hỗ trợ tối đa cho quý doanh nghiệp.
* Chi phí thấp: Đối với các dự án phát triển mới, nếu các doanh nghiệp sử dụng MySQL thì đó là 1 chọn lựa đúng đắn vừa tiết kiệm chi phí vừa đáng tin cậy. Mức duy trì của MySQL không chiếm nhiều thời gian sửa chữa của người quản trị cơ sở dữ liệu và các doanh nghiệp thật sự hài lòng về khả năng xử lý thông qua việc sử dụng server cơ sở dữ liệu MySQL và kiến trúc scale-out.

### 6.2.2 Nhược điểm

* Giới hạn: Theo thiết kế, MySQL không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.
* Độ tin cậy: Cách các chức năng cụ thể được xử lý với MySQL (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán 9,…) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.
* Dung lượng hạn chế: Nếu số bản ghi lớn dần lên thì việc truy xuất dữ liệu là khá khó khăn, khi đó chúng ta sẽ phải áp dụng nhiều biện pháp để tăng tốc độ truy xuất dữ liệu như là chia tải database này ra nhiều server, hoặc tạo cache MySQL
* Không hỗ trợ các table format khác nhau: Một số table format sẽ không hỗ trợ transaction (row locking), không hỗ trợ hot backup, không hỗ trợ định nghĩa kiểu dữ liệu, không hỗ trợ XML, không hỗ trợ các tính năng OLAP,…