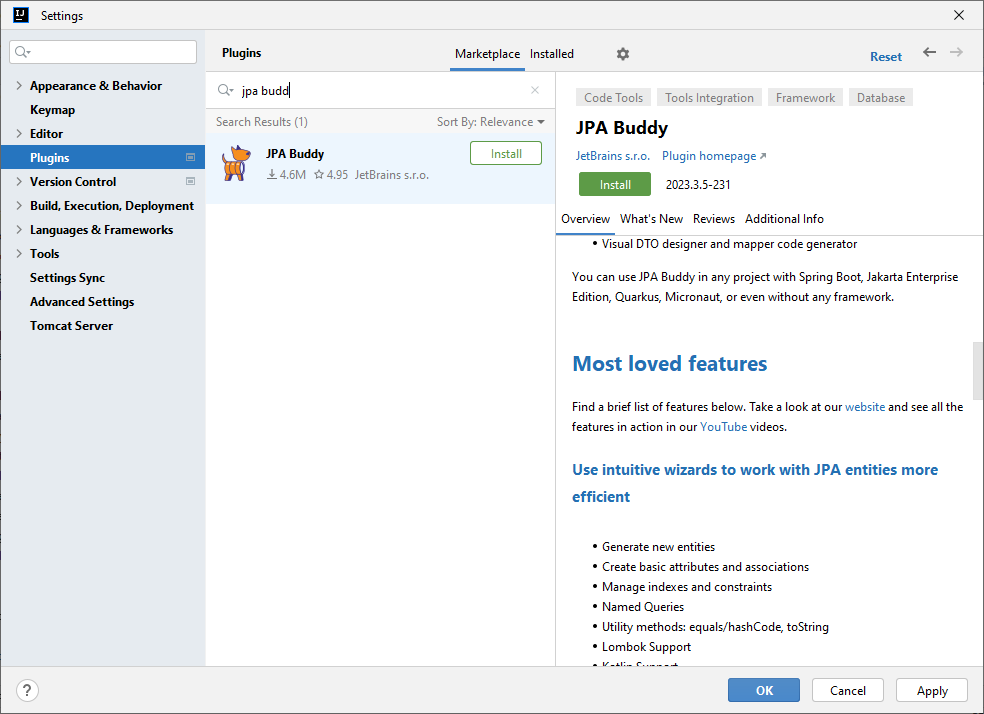
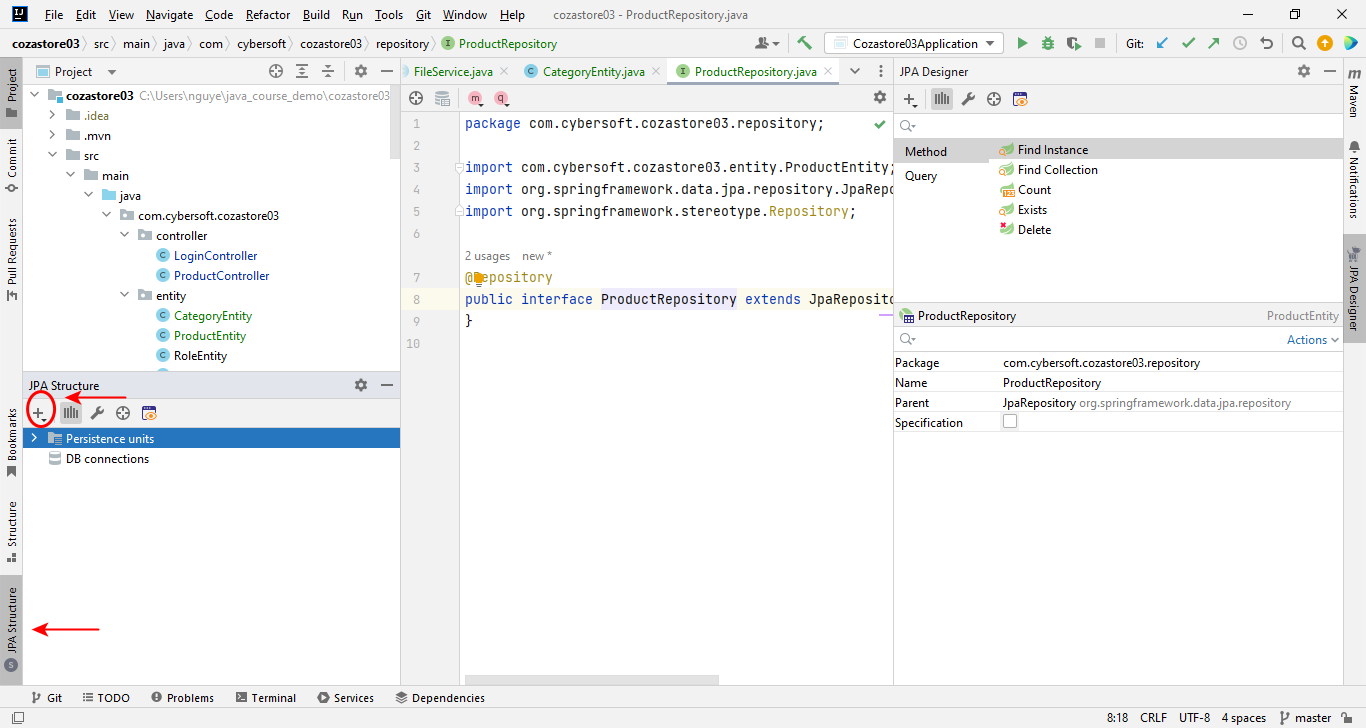
JPA Buddy (5 stars, cực kì nổi tiếng)

hoặc

JPA support

POJO generator





Không có thẻ Database sau khi JPA support (do plugin nó ko hỗ trợ Intellij IDEA bản Community)

<https://techmaster.vn/posts/37274/huong-dan-su-dung-jpa-buddy>

Hướng dẫn sử dụng JPA Buddy

25 tháng 09, 2022 - 2106 lượt xem

[**Spring Boot**](https://techmaster.vn/posts?keyword=&category=buvqnin0k7qv59jeq3sg)[**Lập trình web**](https://techmaster.vn/posts?keyword=&category=79)[**Cơ sở dữ liệu**](https://techmaster.vn/posts?keyword=&category=bkdh8n2tord8dudvd04g)[**Java**](https://techmaster.vn/posts?keyword=&category=bkhdci451cofuk9m7n20)

Written By: Hoàng Mạnh Cường (Java 10)  
Gmail: [manhcuong200997@gmail.com](mailto:manhcuong200997@gmail.com)  
Bài viết gốc: <https://www.baeldung.com/jpa-buddy>

JPA bao gồm gần như mọi khía cạnh giao tiếp giữa cơ sở dữ liệu và ứng dụng Java, được tích hợp vào tất cả framework chính

Nếu sử dụng Intellij, JPA Buddy sẽ rất hữu ích cho bạn. Plugin này sẽ hỗ trợ hướng dẫn các triển khai phổ biến nhất của JPA, gợi ý trực quan các triển khai của JPA, tạo ra code dựa trên phương pháp tốt nhất và tích hợp kiểm tra thông minh để cải thiện code của bạn.

**Lưu ý: Việc lạm dụng Buddy có thể khiến người sử dụng phụ thuộc vào Plugin, quên đi cấu trúc của phương thức, cách triển khai nên cần cân đối, linh hoạt sử dụng Plugin này khi code**

Giờ chúng ta cùng bắt đầu tìm hiểu về Buddy thôi!

**1. Tổng quan**

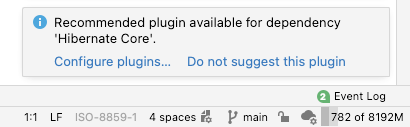
[JPA Buddy](https://www.baeldung.com/jpa-buddy) là một IntelliJ IDEA plugin được sử dụng rộng rãi bởi cả những lập trình viên mới và những người đã có lâu năm kinh nghiệm khi làm việc với mô hình dữ liệu JPA hoặc các công nghệ có liên quan tới Spring Data JPA, các công cụ quản lý phiên bản Database(Flyway, Liquibase), MapStruct,… Nó cung cấp các thiết kế trực quan, tạo code và các kiểm tra bổ sung nhằm đơn giản hóa việc phát triển và cải tiến code theo những cách triển khai tốt nhất cho JPA

Plugin này hoạt động ở cả 2 phiên bản IntelliJ IDEA là Community và Ultimate, và sử dụng mô hình **“freemium”**. Hầu hết các tính năng có sẵn đều miễn phí, và chúng ta cần đăng kí mua để truy cập những gói trả phí.

Trong bài hướng dẫn này, chúng ta sẽ tìm hiểu về những tính năng chính và xem chúng ta có thể sử dụng chúng như thế nào trong suốt chu kì phát triển của ứng dụng. Ví dụ, chúng ta sẽ sử dụng ứng dụng tham khảo phổ biến - **“Spring PetClinic”**.

**2. Mở đầu**

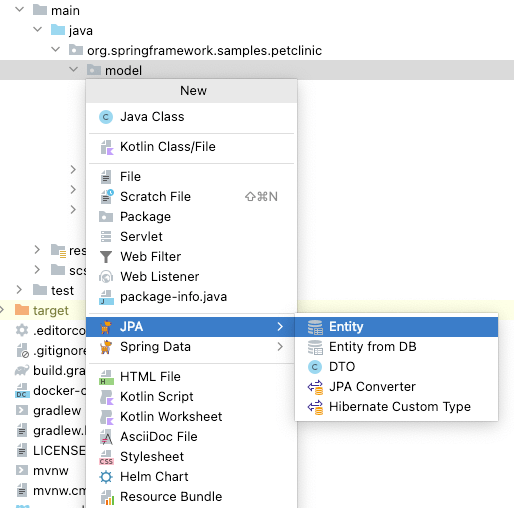
Chúng ta có thể kích hoạt tự động JPA Buddy khi thêm JPA Denpendency vào source code của ứng dụng. Plugin này nằm trong danh sách **“Recommended InteiilJ Plugins”** nên khi chúng ta chưa cài đặt nó, IDEA sẽ đề xuất chúng ta thực hiện. JPA Buddy có thể được cài đặt từ kho plugin Intellij:



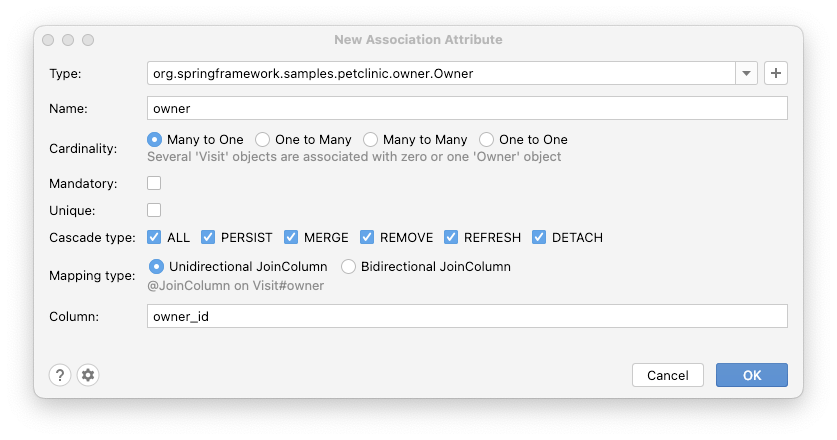
Chức năng của Buddy có thể thay đổi tùy thuộc vào thư viện được đính kèm cho ứng dụng.

**3. Làm việc với JPA Entities**

Thông thường, ứng dụng phát triển bắt đầu với một mô hình dữ liệu. JPA Buddy cung cấp trình thiết kế trực quan và bảng thành phần cho phép chúng ta tạo ra một entity và thêm các thuộc tính cơ bản và liên kết vào nó. Chúng ta có thể gọi ra hành động tương ứng từ menu ngữ cảnh hoặc cửa sổ công cụ JPA Structure:



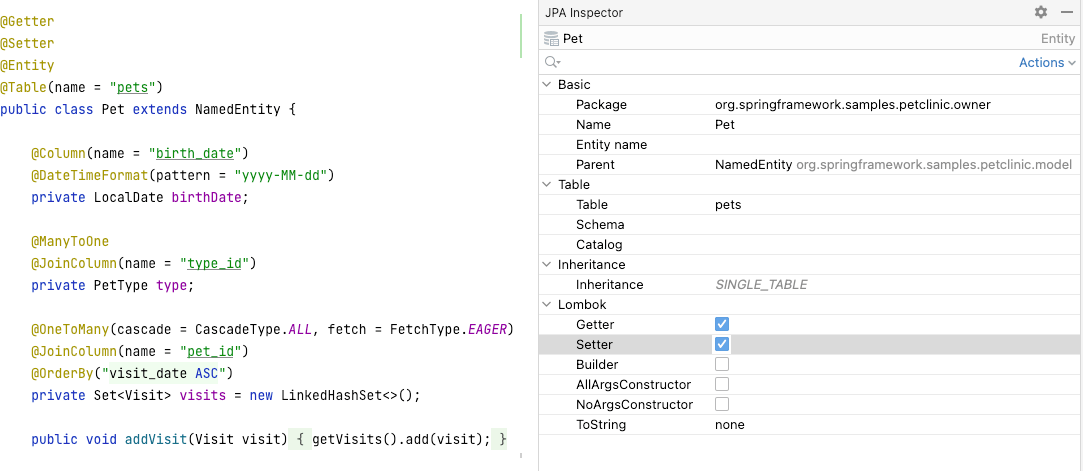
Nếu cần thêm hoặc chỉnh sửa thuộc tính của entity, chúng ta có thể sử dụng JPA palette và JPA inspector. Để thêm một thuộc tính, ta sẽ nhấp đúp chuột vào nó hoặc thực hiện kéo thả. Trình chỉnh sửa trực quan cho phép định nghĩa các tính chất của thuộc tính:



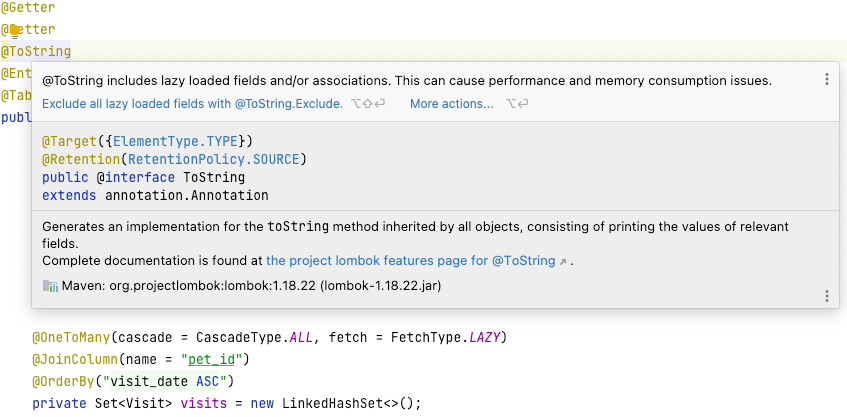
Để chỉnh sửa thuộc tính của entity, chúng ta có thể sử dụng JPA inspector. Theo thông số kĩ thuật của JPA, inspector cho phép người dùng xem và chỉnh sửa hầu hết các tùy chọn cho thuộc tính của một entity. Trình chỉnh sửa hoạt động theo 2 cách: khi cập nhật tính chất của thuộc tính, code sẽ được cập nhật. Và ngược lại, khi code của entity được cập nhật, mọi thay đổi sẽ được ánh xạ trong inspector.

**3.1 Hỗ trợ Lombok**

Theo JPA Specs, các entity nên có getter và setter cho các thuộc tính, và nó thường được coi là **“bản mẫu”** code. Lombok là thư viện phổ biến cho phép người dùng tránh phải viết các bản mẫu và thay thế bằng các annotation. Nhiều lập trình viên sử dụng Lombok trong định nghĩa các entity JPA, vì vậy JPA Buddy hỗ trợ đầy đủ. Người dùng có thể chỉnh sửa các thuộc tính liên quan tới Lombok trong JPA Inspector:

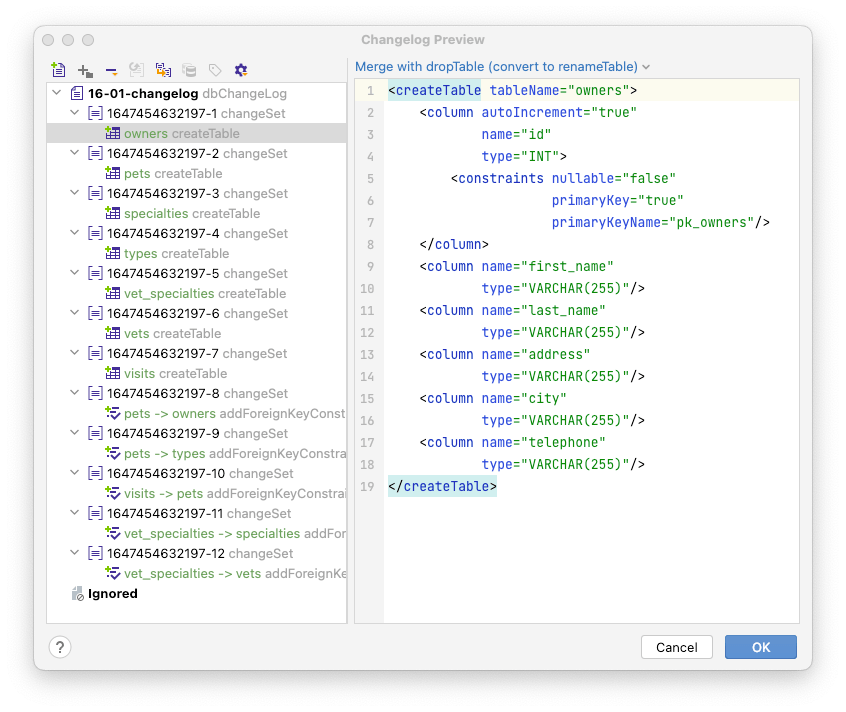


Chức năng của plugin không giới hạn trong việc kiểm tra và chỉnh sửa. Kiểm tra là phần quan trọng trong sự hỗ trợ này. Một số annotation của Lombok có thể gây ra những vẫn đề về hiệu suất hoặc sai sót không mong muốn khi sử dụng với JPA. Ví dụ, sử dụng annotation **@ToString** có thể gây ra **LazyInitException** vì nó sử dụng tất cả thuộc tính cho việc biểu diễn chuỗi của đối tượng, kể cả những đối tượng **“lazy”**. JPA Buddy đưa ra cảnh báo cho những trường hợp như vậy và đưa ra những cách xử lý nhanh chóng, và giúp người dùng tránh gặp phải lỗi trong JPA code:

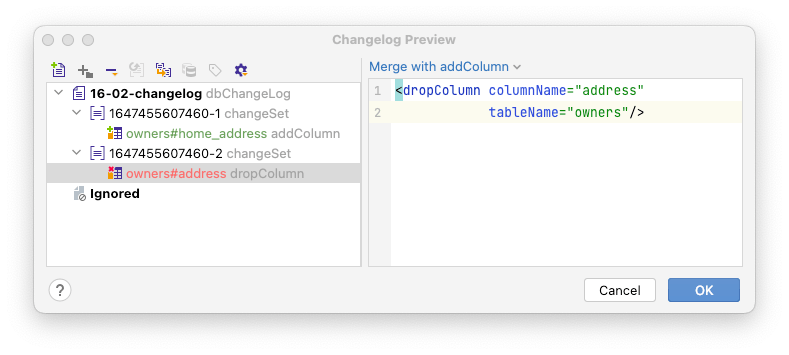


**4. DB Migration với Liquibase và Flyway**

Để tạo database từ mô hình dữ liệu JPA, chúng ta thường sử dụng công cụ quản lý phiên bản DB: Liquibase và Flyway là 2 ông lớn trong lĩnh vực này. Phần dễ xảy ra lỗi nhất trong quá trình chuyển đổi DB là viết các tập lệnh chuyển đổi dựa trên sự thay đổi của các JPA entity. JPA Buddy cung cấp cơ chế tạo ra tập lệnh khác nhau cho cả Liquibase và Flyway. Plugin này có thể so sánh biểu đồ DB hiện có với mô hình JPA hoặc thậm chí 2 biểu đồ:

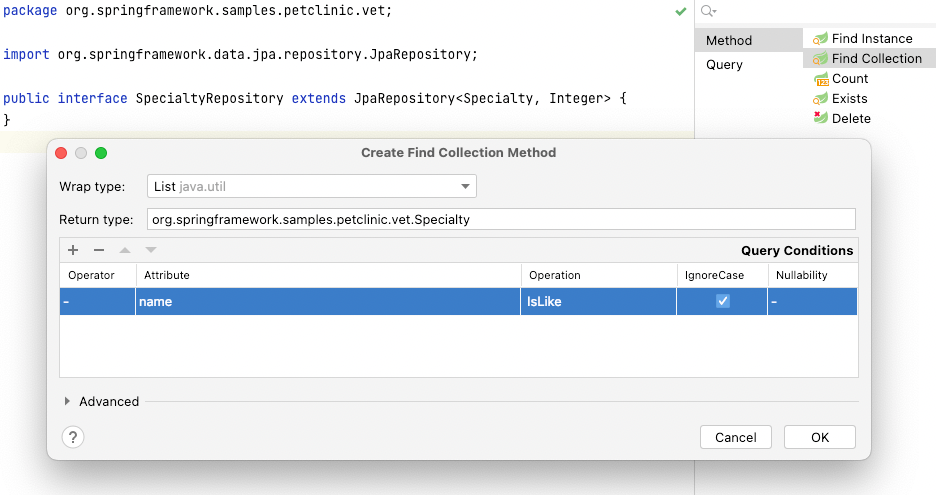


Ngoài ra, JPA Buddy cung cấp tính năng tự động hoàn thành tập lệnh và xem trước thông minh. Ở chế độ này, plugin phân tích những tập lệnh đã được tạo ra và đưa ra cảnh báo nếu thay đổi có thể gây ra lỗi cho nguồn dữ liệu được hướng tới khi cập nhật:

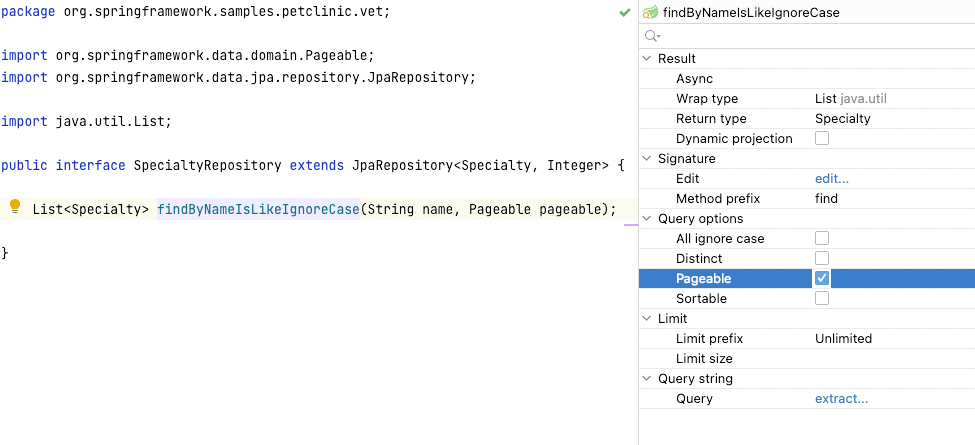


**5. Hỗ trợ Spring Data JPA**

Khi mô hình dữ liệu được tạo ra, bước tiếp theo là thực hiện triển khai các kho dữ liệu truy cập. Spring Data JPA có thể là framework phổ biến nhất cho việc này. JPA Buddy cho phép người dùng tạo ra các kho lưu trữ mới dựa trên định nghĩa entity. Trong repository code, chúng ta có thể sử dụng JPA palette để tạo ra nhiều phương thức và truy vấn khác nhau trong repository code bằng cách sử dụng các công cụ trực quan:



Đối với mỗi phương thức, người dùng có thể thay đổi các thuộc tính của nó bằng JPA inspector. Plugin cho phép người dùng thêm phân trang và sắp xếp cũng như tạo các phép tham chiếu cho kiểu dữ liệu trả về của các phương thức:

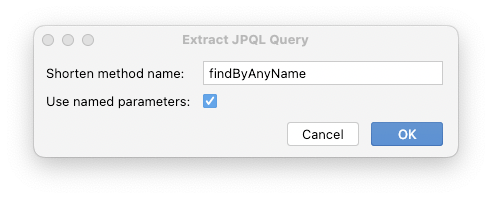


Một tính năng hữu ích nữa là trích xuất truy vấn. Đôi khi tên phương thức quá dài đối với các truy vấn phức tạp như thế này:

List<Owner> findDistinctByFirstNameIgnoreCaseOrLastNameIgnoreCaseOrPets\_NameIgnoreCaseAllIgnoreCaseOrderByFirstNameAsc(

String firstName, String lastName, String name);

JPA Buddy phân tích tên phương thức và cho phép người dùng thay đổi cấu trúc bằng cách đổi tên và thay thế JPQL query bằng annotation @Query:



@Query("select distinct o from Owner o left join o.pets pets " +

"where upper(o.firstName) = upper(:firstName) " +

"or upper(o.lastName) = upper(:lastName) " +

"or upper(pets.name) = upper(:name) " +

"order by o.firstName")

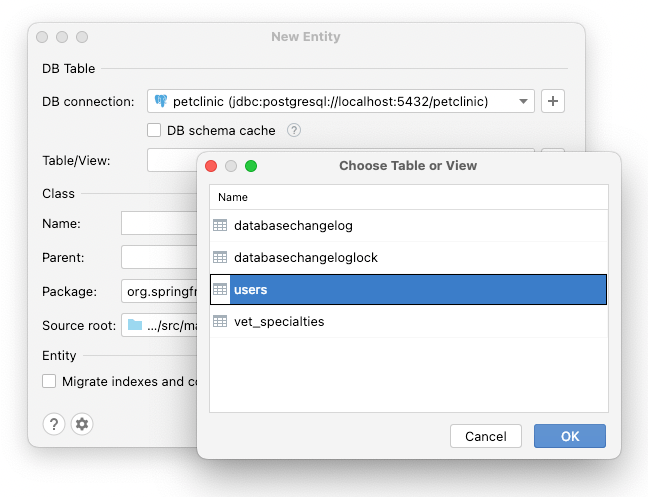
List<Owner> findByAnyName(@Param("firstName") String firstName,

@Param("lastName") String lastName,

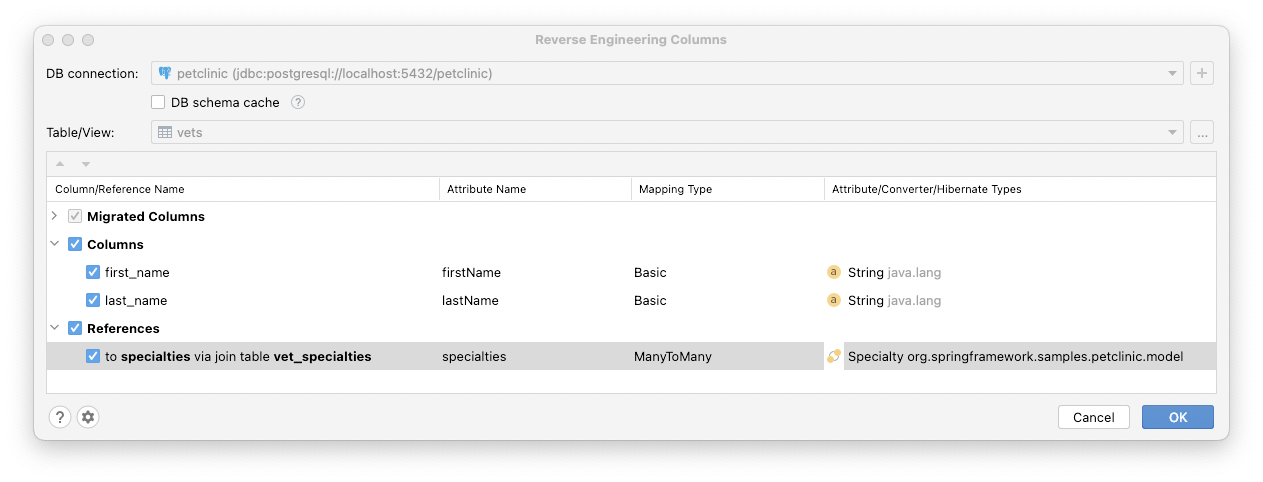
@Param("name") String name);

**6. Tạo entities dựa trên bảng Database**

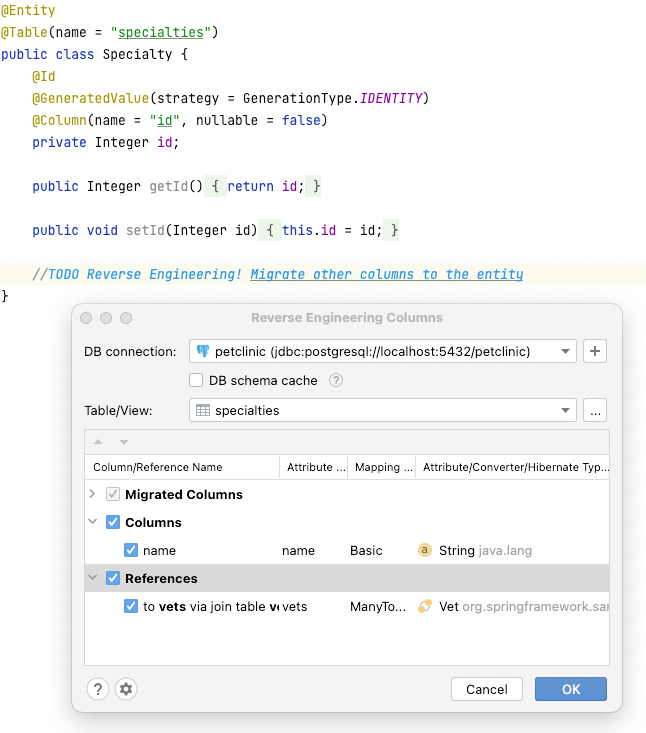
Dữ liệu thường tồn tại lâu hơn code, vì thế khi xây dựng JPA Data layer trên Database có sẵn, chúng ta cần phải tạo JPA entities dựa trên các bảng hiện có. JPA Buddy cung cấp chức năng này và cho phép lập trình viên tạo entites theo cách **“cherry - pick”**, lựa chọn từng bảng một:



Ngược lại với những giải pháp hiện có, plugin biểu diễn cách tạo thông minh và tự phát hiện mối liên hệ giữa các entity. Người dùng sẽ tạo ra những thuộc tính tương ứng cho cả **OneToMany** và **ManyToMany** entities không được hỗ trợ bởi các cột:

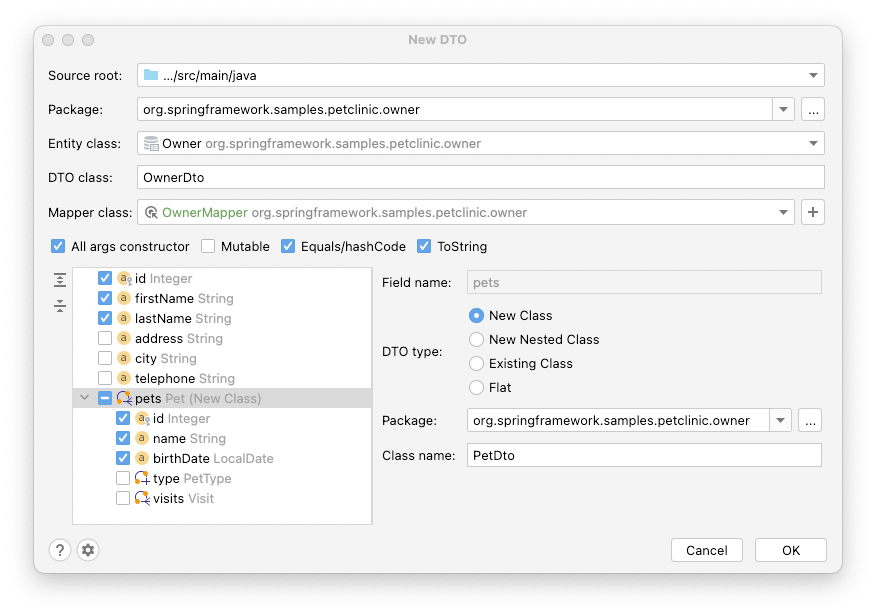


JPA Buddy tạo ra **“empty”** entities với chỉ mỗi cột ID và TODO comment trong nội dung của chúng cho các liên kết như vậy. Nhấp chuột vào TODO, người dùng có thể thực hiện quá trình nhập thêm cột vào trong entity hiện có:



**7. DTOs và tạo ra Mappers**

DTO là mẫu thiết kế có giá trị cho trường hợp truyền dữ liệu mà người dùng không thể ánh xạ trực tiếp tới JPA entities. Ví dụ, khi tạo REST API, chúng ta chỉ muốn hiển thị ra một số thuộc tính của entity. JPA Buddy không thể tạo ra DTOs và ánh xạ bằng thư viện MapStruct. Tất cả những gì chúng ta cần là lựa chọn các thuộc tính mong muốn cho DTOs:



Plugin tạo ra DTO và ánh xạ, bao gồm ánh xạ phù hợp cho liên kết. JPA Buddy hỗ trợ Lombok cho cả trường hợp này và tạo ra annotations thích hợp cho DTOs:

@Data

public class OwnerDto implements Serializable {

private final Integer id;

@NotEmpty

private final String firstName;

@NotEmpty

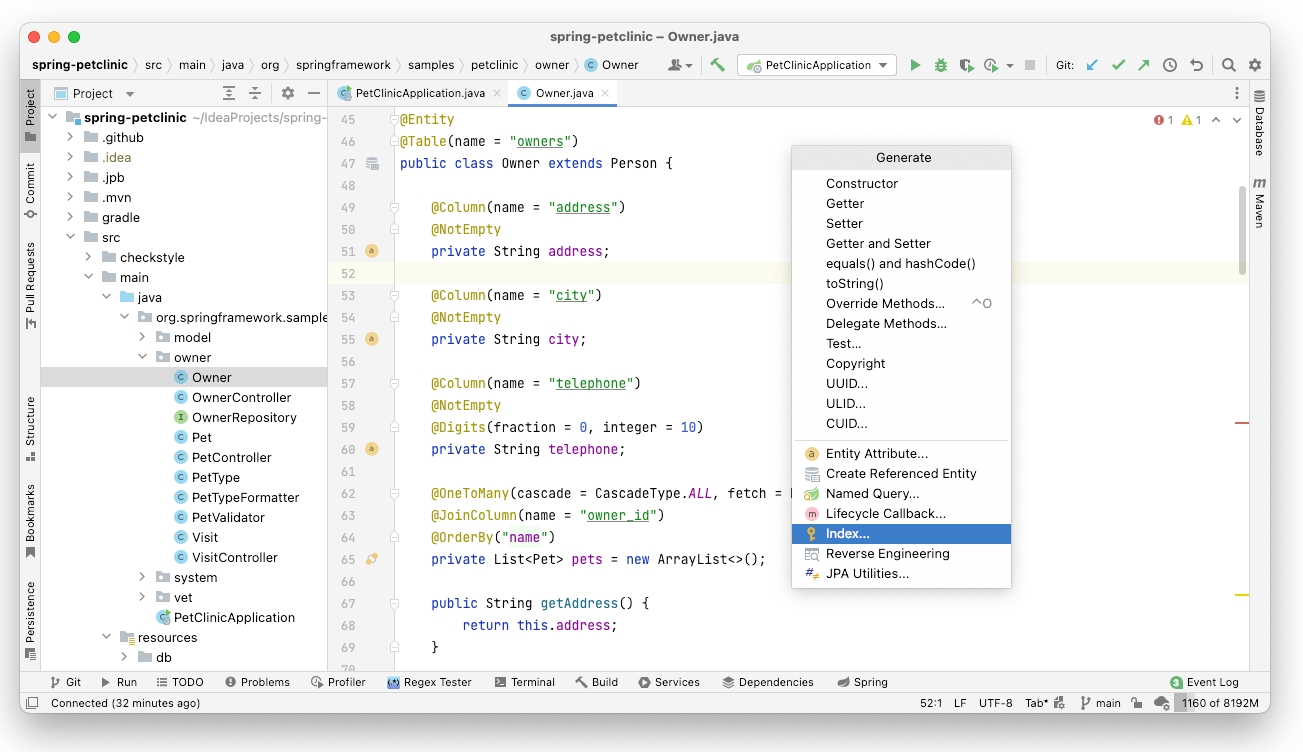
private final String lastName;

private final List<PetDto> pets;

}

**8. Chế độ tối giản (Minimalistic Mode)**

JPA Buddy có giao diện người dùng thân thiện, nhưng đối với những người thích chế độ xem “focused” ở Intellij IDEA, plugin cung cấp **“minimalistic mode”**  
(chế độ tối giản). Chúng ta có thể ẩn tất cả cửa sổ công cụ và chỉ sử dụng bàn phím để tạo entities, tạo và chỉnh sửa Spring Data JPA repositories, gọi trình hướng dẫn tạo DTOS,…:



**9. Kết luận**

JPA Buddy cung cấp một bộ công cụ mạnh mẽ giúp phát triển JPA dễ dàng hơn. Thật tốt là plugin này không chỉ hỗ trợ JPA mà còn cả những thư viện có liên quan được sử dụng trong việc phát triển lớp truy cập dữ liệu: Spring Data JPA, MapStruct, Lombok, và các giải pháp tạo lập phiên bản DB.

Nếu xem xét từ phía IntelliJ IDEA Community, plugin này có thể đơn giản hóa đáng kể công việc thực hiện với JPA. Những người sử dụng IDEA Ultimate có thể nhận được một số chức năng JPA Buddy từ các plugin đi kèm để hỗ trợ JPA và Spring Framework. Tuy nhiên, có vẻ như các tính năng tạo tập lệnh tạo dựng phiên bản DB và các tính năng tạo DTO vẫn là duy nhất trong tự động hóa phát triển DB.

Nếu bạn thích xem video, có một đoạn video 15 phút thể hiện sự hoạt động của JPA Buddy. Bạn có thể xem làm thế nào để tạo ứng dụng với JPA Buddy và Spring Boot trong Intellij IDEA:

**10. Lưu ý**

Đây là 1 số chia sẻ nhỏ cá nhân mình sau khi đã sử dụng JPA Buddy:

* Như đã chia sẻ ở đầu bài viết, việc sử dụng JPA Buddy tuy đem lại rất nhiều tiện lợi, tiết kiệm thời gian nhưng việc lạm dụng có thể khiến chúng ta phụ thuộc vào plugin này, quên đi cấu trúc code.
* Khi sử dụng JPA Buddy, nếu không quá bận thì sau mỗi lần tạo ra phương thức, class,… bạn nên xem lại cụ thể chi tiết bố cục, các thành phần của code, các annotation được sử dụng,…
* Nếu thời gian hoàn thiện project không quá gấp, có thể thực hiện xen kẽ giữa sử dụng JPA Buddy và code thủ công. Ví dụ, có 4 entity thì có thể tạo 2 entity bằng Buddy, 2 entity tự viết thủ công để luôn có thể ghi nhớ được những thành phần cần có.

**Bài viết của mình tới đây là kết thúc. Hi vọng những thông tin trong bài viết này có thể hỗ trợ tốt cho các bạn trong công việc hiện tại và trong tương lai.**

**Thanks for watching!**

