**78.6 Configure Hibernate Naming Strategy**

Hibernate uses [two different naming strategies](http://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html#naming) to map names from the object model to the corresponding database names. The fully qualified class name of the physical and the implicit strategy implementations can be configured by setting the spring.jpa.hibernate.naming.physical-strategy andspring.jpa.hibernate.naming.implicit-strategy properties, respectively. Alternatively, if ImplicitNamingStrategy or PhysicalNamingStrategy beans are available in the application context, Hibernate will be automatically configured to use them.

By default, Spring Boot configures the physical naming strategy with SpringPhysicalNamingStrategy. This implementation provides the same table structure as Hibernate 4: all dots are replaced by underscores and camel casing is replaced by underscores as well. By default, all table names are generated in lower case, but it is possible to override that flag if your schema requires it.

For example, a TelephoneNumber entity is mapped to the telephone\_number table.

If you prefer to use Hibernate 5’s default instead, set the following property:

spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

Alternatively, you can configure the following bean:

*@Bean*

**public** PhysicalNamingStrategy physicalNamingStrategy() {

**return** **new** PhysicalNamingStrategyStandardImpl();

}

See [HibernateJpaAutoConfiguration](https://github.com/spring-projects/spring-boot/tree/master/spring-boot-project/spring-boot-autoconfigure/src/main/java/org/springframework/boot/autoconfigure/orm/jpa/HibernateJpaAutoConfiguration.java) and [JpaBaseConfiguration](https://github.com/spring-projects/spring-boot/tree/master/spring-boot-project/spring-boot-autoconfigure/src/main/java/org/springframework/boot/autoconfigure/orm/jpa/JpaBaseConfiguration.java) for more details.

<https://docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/html/howto-data-access.html>

### **2.2. Naming strategies**

Part of the mapping of an object model to the relational database is mapping names from the object model to the corresponding database names. Hibernate looks at this as 2 stage process:

* The first stage is determining a proper logical name from the domain model mapping. A logical name can be either explicitly specified by the user (using @Column or @Table e.g.) or it can be implicitly determined by Hibernate through an [ImplicitNamingStrategy](https://docs.jboss.org/hibernate/orm/5.1/userguide/html_single/Hibernate_User_Guide.html#ImplicitNamingStrategy) contract.
* Second is the resolving of this logical name to a physical name which is defined by the [PhysicalNamingStrategy](https://docs.jboss.org/hibernate/orm/5.1/userguide/html_single/Hibernate_User_Guide.html#PhysicalNamingStrategy)contract.

#### **2.2.2. PhysicalNamingStrategy**

Many organizations define rules around the naming of database objects (tables, columns, foreign-keys, etc). The idea of a PhysicalNamingStrategy is to help implement such naming rules without having to hard-code them into the mapping via explicit names.

While the purpose of an ImplicitNamingStrategy is to determine that an **attribute named** accountNumber maps to a **logical column name** of accountNumber when not explicitly specified, the purpose of a PhysicalNamingStrategy would be, for example, to say that the physical column name should instead be abbreviated acct\_num.

|  |
| --- |
| It is true that the resolution to acct\_num could have been handled in an ImplicitNamingStrategy in this case. But the point is separation of concerns. The PhysicalNamingStrategy will be applied regardless of whether the attribute explicitly specified the column name or whether we determined that implicitly. The ImplicitNamingStrategy would only be applied if an explicit name was not given. So it depends on needs and intent. |

The **default** implementation is to **simply use the logical name as the physical name**. However applications and integrations can define custom implementations of this PhysicalNamingStrategy contract.

<https://docs.jboss.org/hibernate/orm/5.1/userguide/html_single/Hibernate_User_Guide.html>