Hibernate之SchemaExport+配置文件生成表结构

今天说点基础的东西，说说如何通过SchemaExport跟Hibernate的配置文件生成表结构。其实方法非常简单，只需要两个配置文件，两个Java类就可以完成。

首先要生成表，得先有实体类，以Person.java为例：

```
/**
 *
 * @author Administrator
 * @hibernate.class table="T_Person"
 */
public class Person {

    /**
     * @hibernate.id
     * generator-class="native"
     */
    private int id;

    /**
     * @hibernate.property
     */
    private String name;

    /**
     * @hibernate.property
     */
    private String sex;

    /**
     * @hibernate.property
     */
    private String address;

    /**
     * @hibernate.property
     */
    private String duty;

    /**
```

```
     * @hibernate.property
     */
    private String phone;

    /**
     * @hibernate.property
     */
    private String description;

    /**
     * @hibernate.many-to-one
     */
    private Orgnization org;

    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
    public String getDuty() {
        return duty;
    }
    public void setDuty(String duty) {
        this.duty = duty;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
```

```
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
    public String getSex() {
        return sex;
    }
    public void setSex(String sex) {
        this.sex = sex;
    }
    public Orgnization getOrg() {
        return org;
    }
    public void setOrg(Orgnization org) {
        this.org = org;
    }
}
```

接下来就是Person类对应的配置文件Person.hbm.xml，配置如下：

```
<hibernate-mapping>
  <class table="T_Person" name="com.tgb.model.Person">
    <id name="id">
      <generator class="native"/>
    </id>
    <property name="name"/>
    <property name="sex"/>
    <property name="address"/>
    <property name="duty"/>
    <property name="phone"/>
    <property name="description"/>
    <many-to-one name="org"></many-to-one>
```

```
  </class>
</hibernate-mapping>
```

还有包含Person.hbm.xml相关信息的Hibernate默认配置文件，hibernate.cfg.xml：

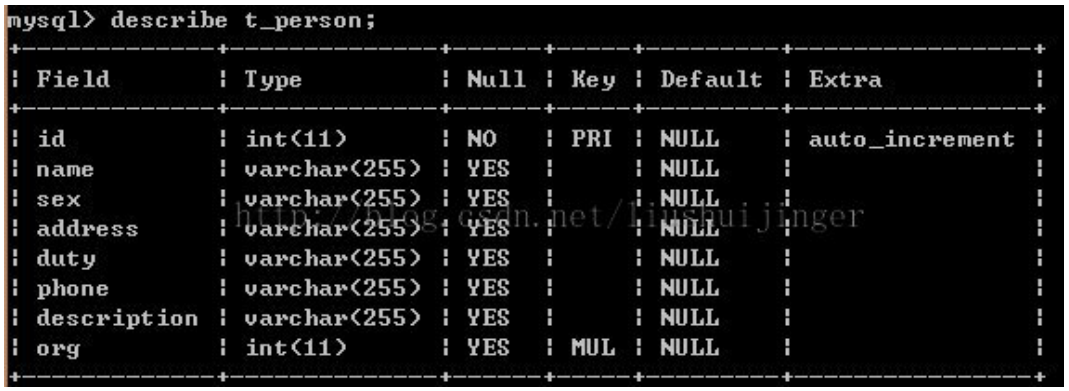
```
<hibernate-configuration>
  <session-factory>
    <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property
name="hibernate.connection.url">jdbc:mysql://127.0.0.1/test</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">123456</property>
    <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.show_sql">true</property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <property
name="hibernate.current_session_context_class">thread</property>
    <mapping resource="com/tgb/model/Person.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

万事俱备只欠东风，最后我们还需要一个根据上述内容生成数据表的小工具，即ExportDB.Java：

```
import org.hibernate.cfg.Configuration;
import org.hibernate.tool.hbm2ddl.SchemaExport;

public class ExportDB {

    /**
     * @param args
     */
    public static void main(String[] args) {

        // 默认读取hibernate.cfg.xml文件
        Configuration cfg = new Configuration().configure();
```

```
        // 生成并输出sql到文件（当前目录）和数据库
        SchemaExport export = new SchemaExport();

        // 创建表结构，第一个true 表示在控制台打印sql语句，第二个true 表示导入sql语句到数据库
        export.create(true, true);
    }
}
```

完成以上步骤以后，只需要执行ExportDB类即可，当然前提是已经在mysql中创建了对应的数据库，我们这里创建了一个名为test的测试数据库。执行成功之后我们就可以看到数据库里已经有了我们的t_person表了，如下图所示：



```
mysql> describe t_person;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int(11)      | NO   | PRI | NULL    | auto_increment |
| name        | varchar(255) | YES  |     | NULL    |                |
| sex         | varchar(255) | YES  |     | NULL    |                |
| address     | varchar(255) | YES  |     | NULL    |                |
| duty        | varchar(255) | YES  |     | NULL    |                |
| phone       | varchar(255) | YES  |     | NULL    |                |
| description | varchar(255) | YES  |     | NULL    |                |
| org         | int(11)      | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
```


OK，你会了吗，就是这么简单，如果之前没弄过，就来试试吧！