**6.6数据库模型层·SQLAlchemy详解**

# 配置项

|  |  |
| --- | --- |
| SQLALCHEMY\_DATABASE\_URI | 用于连接的数据库URI。对于mysql： mysql://username:password@hostname/database |
| SQLALCHEMY\_BINDS | 一个映射binds到连接URI的字典。更多内容可看后续文章 |
| SQLALCHEMY\_ECHO | 如果设置为True.SQLALCHEMY会记录所有发给stderr的语句，这对调试有用。 |
| SQLALCHEMY\_RECORD\_QUERIES | 可以用于显式地禁用或启用查询记录。查询记录在调试或测试模式自动启用。 |
| SQLALCHEMY\_NATIVE\_UNICODE | 可以用于显式禁用原生unicode支持。当使用不合适的指定无编码的数据库默认值时，这对于一些数据库适配器是必须的。 |
| SQLALCHEMY\_POOL\_SIZE | 数据库连接池的大小。默认是引擎默认值（通常是5） |
| SQLALCHEMY\_POOL\_TIMEOUT | 设定连接池的连接超时时间。默认是10 |
| SQLALCHEMY\_POOL\_RECYCLE | 多少秒后自动回收连接。这对MYSQL是必要的，它默认移除闲置多余8小时的连接。注意，如果使用了MYSQL， SQLALCHEMY自动设置这个值为2小时。 |

# 模型属性类型

|  |  |  |
| --- | --- | --- |
| 类型名 | Python类型 | 说明 |
| Integer | int | 普通整数，一般是 32 位 |
| SmallInteger | int | 取值范围小的整数，一般是 16 位 |
| Big Integer | int 或 long | 不限制精度的整数 |
| Float | float | 浮点数 |
| Numeric | decimal.Decimal | 定点数 |
| String | str | 变长字符串 |
| Text | str | 变长字符串，对较长或不限长度的字符串做了优化 |
| Unicode | unicode | 变长 Unicode 字符串 |
| Unicode Text | unicode | 变长 Unicode 字符串，对较长或不限长度的字符串做了优化 |
| Boolean | bool | 布尔值 |
| Date | datetime.date | 日期 |
| Time | datetime.time | 时间 |
| DateTime | datetime.datetime | 日期和时间 |
| Interval | datetime.timedelta | 时间间隔 |
| Enum | str | 一组字符串 |
| PickleType | 任何 Python 对象 | 自动使用 Pickle 序列化 |
| LargeBinary | str | 二进制文件 |

# 常用 SQLAlchemy 列选项

|  |  |
| --- | --- |
| 选项名 | 说明 |
| primary\_key | 如果设为 True，这列就是表的主键 |
| unique | 如果设为 True，这列不允许出现重复的值 |
| index | 如果设为 True，为这列创建索引，提升查询效率 |
| nullable | 如果设为 True，这列允许使用空值；如果设为 False，这列不允许使用空值 |
| default | 为这列定义默认值 |

# 插入记录

# 构造对象  
u = User()  
# 设置属性  
u.user\_name="admin"  
u.real\_name="张三"  
# 插入数据库  
db.session.add(u)  
# 提交事务  
db.session.commit()

# 修改记录

# 修改id=1的user\_name和real\_name  
User.query.filter\_by(id=1).update({  
 "user\_name": "admin1",  
 "real\_name": "admin2"  
})  
# 提交事务  
db.session.commit()

# 查询记录

## 根据主键查询

u = User.query.get(1)

## 根据其他条件查询-仅查一条

User.query.filter\_by(user\_name="admin").first()

## 根据多个条件查询

User.query.filter\_by(user\_name="admin",real\_name="张三").all()

## 等值查询

User.query.filter(User.user\_name="admin").all()

## 非等值查询

User.query.filter(User.user\_name!="admin").all()

## 大于查询

User.query.filter(User.id > 5).all()

## 大于等于查询

User.query.filter(User.id >= 5).all()

## 小于查询

User.query.filter(User.id < 5).all()

## 小于等于查询

User.query.filter(User.id >= 5).all()

## 区间查询

User.query.filter(User.id.between(1,3)).all()

## 全模糊查询

keywords = "admin"  
User.query.filter(User.user\_name.like("%"+keywords+"%")).all()

## 非全模糊查询

keywords = "admin"  
User.query.filter(User.user\_name.notlike("%"+keywords+"%")).all()

## 左模糊查询

keywords = "admin"  
User.query.filter(User.user\_name.like("%"+keywords)).all()

## 右模糊查询

keywords = "admin"  
User.query.filter(User.user\_name.like(keywords+"%")).all()

## IN查询

User.query.filter(User.id.in([1,2,3).all()

## 非IN查询

User.query.filter(User.id.notin([1,2,3).all()

## 组合查询

User.query.filter(User.id.in([1,2,3).filter(User.user\_name.like("%admin%")).all()

## 分页查询

page\_num = 1 # 当前页，默认1  
page\_size = 10 # 每页大小  
user\_obj=User.query.filter().paginate(page\_num, page\_size,False)  
# print(user\_obj.page) # 当前页码-从1开始  
# print(user\_obj.per\_page) # 每页大小  
# print(user\_obj.total) # 总记录数  
# print(user\_obj.items) # 数据集

# 删除记录

## 根据id删除

# 删除id=1的用户  
User.query.filter\_by(id=1).delete()  
# 提交事务  
db.session.commit()

## 删除多条记录

# where in (1,2,3)  
User.query.filter(User.id.in\_([1,2,3]).delete()  
db.session.commit()