

NSD Devops DAY02

1. [案例1：向表中添加数据](#)
2. [案例2：创建表](#)
3. [案例3：添加数据](#)

1 案例1：向表中添加数据

1.1 问题

1. 向employees表插入数据
2. 向salary表插入数据
3. 插入的数据需要commit到数据库中

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：PyMySQL安装

1) 安装gcc，有些软件包是C的源码

```
01. [ root@localhost ~] # yum install -y gcc
02. 已加载插件：fastestmirror, langpacks
03. dvd | 3.6 kB 00:00
04. Loading mirror speeds from cached hostfile
05. 匹配 gcc-4.8.5-16.el7.x86_64 的软件包已经安装。正在检查更新。
06. 无须任何处理
```

2)为了加速下载，可以使用国内开源镜像站点

```
01. [ root@localhost ~] # mkdir ~/.pip
02. [ root@localhost ~] # vim ~/.pip/pip.conf
03. [ global]
04. index-url = http://py pi.douban.com/simple/
05. [ install]
06. trusted-host=py pi.douban.com
```

3)安装pymysql

```
01. [ root@localhost ~] # pip3 install py my sql
```

[Top](#)

步骤二：安装mariadb-server

```

01. [root@localhost ~] # yum install -y mariadb-server
02.
03. ....
04.
05. 已安装：
06.      mariadb-server.x86_64 1:5.5.56-2.el7
07.
08. 作为依赖被安装：
09.      mariadb.x86_64 1:5.5.56-2.el7
10.      perl-DBD-MySQL.x86_64 0:4.0.23-5.el7
11.
12. 完毕！
13. [root@localhost ~] # systemctl start mariadb
14. [root@localhost ~] # systemctl enable mariadb
15. [root@localhost ~] # mysqladmin password tedu.cn

```

步骤三：创建数据库

1)创建数据库

```

01. [root@localhost ~] # mysql -uroot -ptedu.cn
02. MariaDB [(none)]> CREATE DATABASE tedu DEFAULT CHARSET 'utf8';
03. Query OK, 1 row affected (0.00 sec)

```

2)创建部门表

部门表字段：部门ID、部门名称

```

01. MariaDB [(none)]> USE tedu;
02. Database changed
03. MariaDB [tedu]> CREATE TABLE departments( dep_id INT PRIMARY KEY, dep_name VARCHAR
04. Query OK, 0 rows affected (0.04 sec)

```

3)创建员工表

员工表字段：员工编号、姓名、出生日期、部门ID、电话号码、email、引用外键id

[Top](#)

01. MariaDB [tedu] > CREATE TABLE employees (emp_id INT PRIMARY KEY, emp_name VARCHAR
02. Query OK, 0 rows affected (0.05 sec)

4)创建工资表

工资表字段：auto_id、员工编号、日期、基本工资、奖金、工资总和

01. MariaDB [tedu] > CREATE TABLE salary (auto_id INT AUTO_INCREMENT PRIMARY KEY, date
02. Query OK, 0 rows affected (0.05 sec)

步骤四：向departments表插入数据

1)新建insert_data.py文件，编写代码如下：

01. [root@localhost day 10] # vim insert_data.py
- 02.
03. import pymysql
04. 1) 连接数据库
05. conn = pymysql.connect(
06. host='127.0.0.1', #连接ip
07. port=3306, #端口号
08. user='root', #数据库用户名
09. passwd='tedu.cn', #数据库密码
10. db='tedu', #数据库名
11. charset='utf8' #设置了数据库的字符集
12.)
13. 2) 创建游标
14. cursor = conn.cursor()
15. 3) 向部门表departments中插入数据
16. insert1 = "INSERT INTO departments(dep_id, dep_name) VALUES(%s, %s) "
17. result = cursor.execute(insert1, (1, '人事部')) # execute执行insert语句
18. 4) 将更新提交到数据库
19. conn.commit()
20. 5) 关闭游标
21. cursor.close()
22. 6) 关闭数据库连接
23. conn.close()

[Top](#)

2) 执行insert_data.py文件：

```
01. [root@localhost day 10] # python3 insert_data.py
```

3) 登录mariadb查看结果：

```
01. MariaDB [tedu] >> select * from departments;
02. +-----+-----+
03. | dep_id | dep_name |
04. +-----+-----+
05. |      1 | 人事部   |
06. +-----+-----+
07. 1 row in set (0.00 sec)
```

4) 向部门表departments中插入数据还可以用如下方法：

```
01. #以上insert_data.py 文件第3步可用如下代码替换：
02.
03. insert1 = "INSERT INTO departments( dep_id, dep_name) VALUES( %s, %s) "
04. data = [(2, '运维部'), (3, '开发部')]
05. cursor.executemany( insert1, data)
```

mariadb查看结果如下：

```
01. MariaDB [tedu] >> select * from departments;
02. +-----+-----+
03. | dep_id | dep_name |
04. +-----+-----+
05. |      1 | 人事部   |
06. |      2 | 运维部   |
07. |      3 | 开发部   |
08. +-----+-----+
09. 3 rows in set (0.01 sec)
```

[Top](#)

步骤五：向employees表插入数据

1) 新建insert_emp.py文件，编写代码如下：

```

01. [ root@localhost day 10] # vim insert_emp.py
02.
03. import pymysql
04. 1) 连接数据库
05. conn = pymysql.connect(
06.     host='127.0.0.1',      #连接ip
07.     port=3306,            #端口号
08.     user='root',          #数据库用户名
09.     passwd='tedu.cn',     #数据库密码
10.     db='tedu',            #数据库名
11.     charset='utf8'        #设置了数据库的字符集
12. )
13. 2) 创建游标
14. cursor = conn.cursor()
15. 3) 向部门表employees中插入数据
16. insert1 = "INSERT INTO employees( emp_id, emp_name, birth_date,phone, email, dep_id) V.
17. result = cursor.execute( insert1, ( 1, '王君', '2018-9-30', \
18.     '15678789090', 'wj@163.com', 3) )      # execute执行insert语句
19. 4) 将更新提交到数据库
20. conn.commit()
21. 5) 关闭游标
22. cursor.close()
23. 6) 关闭数据库连接
24. conn.close()

```

2) 执行insert_emp.py文件：

```

01. [ root@localhost day 10] # python3 insert_emp.py

```

3) 登录mariadb查看结果：

```

01. MariaDB [tedu] >> select * from employees;
02. +-----+-----+-----+-----+-----+-----+
03. | emp_id | emp_name | birth_date | phone | email | dep_id |
04. +-----+-----+-----+-----+-----+-----+
05. |      1 | 王君 | 2018-09-30 | 15678789090 | wj@163.com | 3 |
06. +-----+-----+-----+-----+-----+-----+

```

[Top](#)

07. 1 row in set (0.00 sec)

4) 向部门表employees中插入数据还可以用如下方法：

```
01. #以上insert_emp.py 文件第3步可用如下代码替换：
02.
03. insert1 = "INSERT INTO employees ( dep_id, dep_name) VALUES( %s, %s) "
04. data = [( 2, '运维部'), ( 3, '开发部')]
05. cursor.executemany( insert1, data)
```

mariadb查看结果如下：

```
01. MariaDB [ tedu ] >> select * from departments;
02. +-----+-----+-----+-----+-----+-----+
03. | emp_id | emp_name | birth_date | phone      | email      | dep_id |
04. +-----+-----+-----+-----+-----+-----+
05. |      1 | 王君    | 2018-09-30 | 15678789090 | wj@163.com |      3 |
06. |      2 | 李雷    | 2018-09-30 | 15678789090 | wj@163.com |      2 |
07. |      3 | 张美    | 2018-09-30 | 15678789090 | zm@163.com |      1 |
08. +-----+-----+-----+-----+-----+-----+
09. 3 rows in set ( 0.00 sec)
```

步骤六：向salary表插入数据

1)新建insert_sal.py文件，编写代码如下：

```
01. [ root@localhost day10 ] # vim insert_sal.py
02.
03. import pymysql
04. 1) 连接数据库
05. conn = pymysql.connect(
06.     host='127.0.0.1',      #连接ip
07.     port=3306,            #端口号
08.     user='root',          #数据库用户名
09.     passwd='tedu.cn',     #数据库密码
10.     db='tedu',            #数据库名
11.     charset='utf8'        #设置了数据库的字符集
12. )
```

[Top](#)

```

13. 2) 创建游标
14. cursor = conn.cursor()
15. 3) 向部门表salary中插入数据
16. insert2 = "INSERT INTO salary( date, emp_id,basic, awards) VALUES( %s, %s, %s, %s) "
17. data = [ ( '2018-9-30', 2, 1000, 2000), ( '2018-9-30', 3, 3000, 6000), ( '2018-9-30', 1, 80
18. cursor.executemany( insert2, data)
19. 4) 将更新提交到数据库
20. conn.commit()
21. 5) 关闭游标
22. cursor.close()
23. 6) 关闭数据库连接
24. conn.close()

```

2)执行insert_sal.py文件：

```
01. [root@localhost day 10] # python3 insert_sal.py
```

3)登录mariadb查看结果：

```

01. MariaDB [tedu] >> select * from salary;
02. +-----+-----+-----+-----+-----+
03. | auto_id | date      | emp_id | basic | awards |
04. +-----+-----+-----+-----+-----+
05. |      1 | 2018-09-30 |      2 | 1000 | 2000 |
06. |      2 | 2018-09-30 |      3 | 3000 | 6000 |
07. |      3 | 2018-09-30 |      1 | 8000 | 9000 |
08. +-----+-----+-----+-----+-----+
09. 3 rows in set (0.01 sec)

```

2 案例2：创建表

2.1 问题

1. 创建employees表
2. 创建部门表
3. 创建salary表
4. 表间创建恰当的关系

[Top](#)

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：SQLAlchemy安装

注意：sqlalchemy可以连接各种数据库

```

01. [root@serwang ~] # pip3 install sqlalchemy
02. Collecting sqlalchemy
03.   Downloading http://py pi.doubanio.com/packages/aa/cc/48eec885d81f7260b07d
04.   961b3ececfc0aa82f7d4a8f45ff997e0d3f44ba/SQLAlchemy-1.2.11.tar.gz (5.6MB)
05.   ...
06.   ...
07.   Installing collected packages: sqlalchemy
08.     Running setup.py install for sqlalchemy ... done
09.   Successfully installed sqlalchemy-1.2.11
10.   You are using pip version 9.0.1, however version 18.0 is available.
11.   You should consider upgrading via the 'pip install --upgrade pip' command.

```

步骤二：为SQLAlchemy创建数据库

```

01. MariaDB [tedu] > CREATE DATABASE tarena DEFAULT CHARSET 'utf8';

```

步骤三：创建部门表，创建dbconn.py文件，编写如下代码：

1) 创建连接到数据库的引擎

```

01. [root@localhost day 10] # vim dbconn.py
02.
03.  #! /usr/bin/env python3
04.
05.  from sqlalchemy import create_engine
06.
07.  # 创建连接到数据库的引擎
08.  engine = create_engine(
09.      #指定数据库、用户名、密码、连接到哪台服务器、库名等信息
10.      'mysql+py mysql://root:tedu.cn@localhost/tarena?charset=utf8',
11.      encoding='utf8',
12.      echo=True  #终端输出
13.  )

```

[Top](#)

2)创建ORM映射，生成ORM映射所需的基类

```
01. from sqlalchemy.ext.declarative import declarative_base
02.
03. Base = declarative_base()
```

3)自定义映射类，创建部门表

```
01. from sqlalchemy import Column, String, Integer
02.
03. class Departments(Base): # 必须继承于Base
04.     __tablename__ = 'departments' # 库中的表名
05.     # 每个属性都是表中的一个字段，是类属性
06.     dep_id = Column(Integer, primary_key=True) # Integer整数类型, primary_key 主键
07.     # String字符串类型，nullable非空约束，unique唯一性约束
08.     dep_name = Column(String(20), nullable=False, unique=True)
09.
10.     def __str__(self):
11.         return '[ 部门ID: %s, 部门名称: %s]' % (self.dep_id, self.dep_name)
12. if __name__ == '__main__':
13.     # 在数据库中创建表，如果库中已有同名的表，将不会创建
14.     Base.metadata.create_all(engine)
```

4)测试脚本执行，生成部门表

```
01. [root@localhost day 10] # python3 dbconn.py #成功生成部门表
```

5)进入数据库查看结果

```
01. #登录数据库
02. [root@localhost day 10] # mysql -uroot -ptedu.cn
03. Welcome to the MariaDB monitor. Commands end with ; or \g.
04. Your MariaDB connection id is 5
05. Server version: 5.5.56 MariaDB MariaDB Server
06.
07. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
08.
```

[Top](#)

```

09. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
10. #查看数据库表
11. MariaDB [(none)]> use tarena;
12. Reading table information for completion of table and column names
13. You can turn off this feature to get a quicker startup with -A
14.
15. Database changed
16. MariaDB [tarena]> show tables;
17. +-----+
18. | Tables_in_tarena |
19. +-----+
20. | departments      |
21. +-----+
22. 1 row in set (0.00 sec)
23.
24.
25. MariaDB [tarena]> show create table departments;
26. +-----+-----+
27. | Table      | Create Table
28. +-----+-----+
29. | departments | CREATE TABLE `departments` (
30.   `dep_id` int(11) NOT NULL AUTO_INCREMENT,
31.   `dep_name` varchar(20) NOT NULL,
32.   PRIMARY KEY (`dep_id`),
33.   UNIQUE KEY `dep_name` (`dep_name`)
34. ) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
35. +-----+-----+
36. 1 row in set (0.00 sec)

```

步骤四：创建员工表，在dbconn.py文件中添加如下数据：

1)创建员工表

```

01. from sqlalchemy import ForeignKey  导入外键
02.
03. class Employees(Base): # 必须继承于Base
04.     __tablename__ = 'employees' # 库中的表名
05.     # 每个属性都是表中的一个字段，是类属性
06.     emp_id = Column(Integer, primary_key=True) # Integer整数类型, primary_key 主键
07.     name = Column(String(20), nullable=False) # String字符串类型，nullable非空约束

```

[Top](#)

```

08.     gender = Column( String( 6 ) )
09.     phone = Column( String( 11 ) )
10.     email = Column( String( 50 ) )
11.     dep_id = Column( Integer, ForeignKey( 'departments.dep_id' ) )    #与departments中dep_
12.
13.     def __str__( self ) :
14.         return '员工 : %s' % self.name

```

4)测试脚本执行，生成员工表

```

01. [ root@localhost day 10] # python3 dbconn.py    #成功生成员工表

```

5)进入数据库查看结果

```

01. #登录数据库
02. [ root@localhost day 10] # mysql -uroot -ptedu.cn
03. Welcome to the MariaDB monitor.  Commands end with ; or \g.
04. Your MariaDB connection id is 5
05. Server version: 5.5.56 MariaDB MariaDB Server
06.
07. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
08.
09. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
10. #查看数据库表
11. MariaDB [(none)]> use tarena;
12. Reading table information for completion of table and column names
13. You can turn off this feature to get a quicker startup with -A
14.
15. Database changed
16. MariaDB [tarena]> show tables;
17. +-----+
18. | Tables_in_tarena |
19. +-----+
20. | departments      |
21. | employees         |
22. +-----+
23. 2 rows in set (0.00 sec)

```

[Top](#)

步骤五：创建工资表，在dbconn.py文件中添加如下数据：

1)创建工资表

```

01  from sqlalchemy import Date    导入外键
02.
03.  class Employees( Base): # 必须继承于Base
04.      __tablename__ = 'employees' # 库中的表名
05.      # 每个属性都是表中的一个字段，是类属性
06.      emp_id = Column( Integer, primary_key=True)    #Integer整数类型,primary_key 主键
07.      name = Column( String( 20), nullable=False) # String字符串类型，nullable非空约束
08.      gender = Column( String( 6) )
09.      phone = Column( String( 11) )
10.      email = Column( String( 50) )
11.      dep_id = Column( Integer, ForeignKey( 'departments.dep_id'))    #与departments中dep_
12.
13.      def __str__( self):
14.          return '员工 : %s' % self.name
15.
16.  class Salary( Base): # 必须继承于Base
17.      __tablename__ = 'salary' # 库中的表名
18.      auto_id = Column( Integer, primary_key=True) #Integer整数类型,primary_key 主键
19.      date = Column( Date)    #导入日期
20.      emp_id = Column( Integer, ForeignKey( 'employees.emp_id')) #与employees中emp_id做
21.      basic = Column( Integer)    #基本工资
22.      awards = Column( Integer)    #奖金

```

4)测试脚本执行，生成员工表

```

01  [ root@localhost day 10] # python3 dbconn.py    #成功生成工资表

```

5)进入数据库查看结果

```

01  #登录数据库
02.  [ root@localhost day 10] # mysql -uroot -ptedu.cn
03.  Welcome to the MariaDB monitor.  Commands end with ; or \g.
04.  Your MariaDB connection id is 5
05.  Server version: 5.5.56 MariaDB MariaDB Server

```

[Top](#)

```

06.
07. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
08.
09. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
10. #查看数据库表
11. MariaDB [(none)]> use tarena;
12. Reading table information for completion of table and column names
13. You can turn off this feature to get a quicker startup with -A
14.
15. Database changed
16. MariaDB [tarena]> show tables;
17. +-----+
18. | Tables_in_tarena |
19. +-----+
20. | departments      |
21. | employees         |
22. | salary            |
23. +-----+
24. 3 rows in set (0.00 sec)

```

3 案例3：添加数据

3.1 问题

1. 分别在部门表、员工表和工资表中加入数据
2. 通过SQLAlchemy代码实现
3. 分别练习每次加入一行数据和每次可加入多行数据

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：向部门表添加数据，创建add_department.py文件，添加如下代码：

1)创建映射类的实例

```

01. [root@localhost day10]# vim add_department.py
02.
03. from dbconn import Departments
04.
05. hr = Departments(dep_id=1, dep_name='hr')
06. print(hr.dep_name)
07. print(hr.dep_id)

```

[Top](#)

测试执行结果：

```
01. [root@localhost day 10] # python3 add_department.py
02. hr
03. 1
```

登录数据库查看，部门表中数据为空，此时，并不会真正在数据库表中添加记录

```
01. MariaDB [tarena] > select * from departments;
02. Empty set (0.00 sec)
```

2) 想在数据库中添加数据需完成如下操作，创建会话类

在dbconn.py文件中，添加如下代码：

通过将sessionmaker与数据库引擎绑定，创建会话类Session

```
01. from sqlalchemy.orm import sessionmaker
02. Session = sessionmaker(bind=engine)
```

3)添加新对象

在add_department.py文件中添加如下代码：

```
01. from dbconn import Departments, Session
02.
03. session = Session() #创建会话类实例
04. session.add(hr) #向实例绑定的数据库添加数据
05. session.commit() #将数据提交到实例对应数据库
06. session.close() #关闭session类
```

测试执行结果：

```
01. [root@localhost day 10] # python3 add_department.py
```

登录数据库查看部门表中数据

[Top](#)

```

01. MariaDB [ tarena ] > select * from departments;
02. +-----+-----+
03. | dep_id | dep_name |
04. +-----+-----+
05. |      1 | hr      |
06. +-----+-----+
07. 1 row in set ( 0.01 sec)

```

数据成功添加

4)批量添加新数据

在add_department.py文件中添加如下代码：

```

01. ops = Departments( dep_id=2, dep_name='operations')
02. dev = Departments( dep_id=3, dep_name='development')
03. finance = Departments( dep_id=4, dep_name='财务部')
04. deps = [ ops, dev ]
05. session = Session( )
06. session.add_all( deps)
07. session.add( finance)
08. session.commit( )
09. session.close( )

```

此时注意：

添加过的数据不要再添加，即将session.add (hr) 注释掉。

如果文件中有中文，注意在dbconn.py文件中，将engine的参数修改为如下代码：

```

01. engine=create_engine(
02. 'mysql+py mysql://root:tedu.cn@localhost/tarena?charset=uU8',
03. encoding='uU8',
04. echo=True
05. )

```

测试执行结果：

```
01. [ root@localhost day 10 ] # python3 add_department.py
```

[Top](#)

登录数据库查看部门表中数据

```

01. MariaDB [tarena] > select * from departments;
02. +-----+-----+
03. | dep_id | dep_name |
04. +-----+-----+
05. |      3 | development |
06. |      1 | hr |
07. |      2 | operations |
08. |      4 | 财务部 |
09. +-----+-----+
10. 4 rows in set (0.00 sec)

```

数据成功添加

步骤二：向员工表批量添加数据，创建add_employees.py文件，添加如下代码：

```

01. from dbconn import Employees, Session
02.
03. wj = Employees(
04.     emp_id=1, name='王俊', gender='男', phone='15678789090', email='wj@163.com', dep_
05. )
06. wwc = Employees(
07.     emp_id=2, name='吴伟超', gender='男', phone='13499887755', email='wwc@qq.com', c
08. )
09. dzj = Employees(
10.     emp_id=3, name='董枝俊', gender='男', phone='18900998877', email='dzj@163.com', c
11. )
12. ltd = Employees(
13.     emp_id=4, name='李通达', gender='男', phone='13378904567', email='ltd@163.com'
14. wxy = Employees(
15.     emp_id=5, name='王秀燕', gender='女', phone='15098765432', email='wxy@tedu.cn
16. gq = Employees(
17.     emp_id=6, name='高琦', gender='女', phone='15876543212', email='gq@tarena.com
18. wzf = Employees(
19.     emp_id=7, name='王召飞', gender='男', phone='15609871234', email='wzf@sohu.c
20. sy = Employees(
21.     emp_id=8, name='孙燕', gender='女', phone='18567895435', email='sy@163.com', c
22. gpf = Employees(
23.     emp_id=9, name='高鹏飞', gender='男', phone='13566889900', email='gpf@163.com

```



```

24. emps = [ wj, wwc, dzj, ltd, wxy, gq, wzf, sy, gpf ]
25. session = Session()
26. session.add_all(emps)
27. session.commit()
28. session.close()

```

测试执行结果：

```
01. [ root@localhost day 10 ] # python3 add_employees.py
```

登录数据库查看部门表中数据

```

01. MariaDB [ tarena ] > select * from employees;
02. +-----+-----+-----+-----+-----+-----+
03. | emp_id | name   | gender | phone   | email        | dep_id |
04. +-----+-----+-----+-----+-----+-----+
05. |      1 | 王俊   | 男     | 15678789090 | wj@163.com   |      3 |
06. |      2 | 吴伟超 | 男     | 13499887755 | wwc@qq.com   |      3 |
07. |      3 | 董枝俊 | 男     | 18900998877 | dzj@163.com  |      3 |
08. |      4 | 李通达 | 男     | 13378904567 | ltd@163.com  |      2 |
09. |      5 | 王秀燕 | 女     | 15098765432 | wxy@tedu.cn  |      2 |
10. |      6 | 高琦   | 女     | 15876543212 | gq@tarena.com |      1 |
11. |      7 | 王召飞 | 男     | 15609871234 | wzf@sohu.com |      1 |
12. |      8 | 孙燕   | 女     | 18567895435 | sy@163.com   |      4 |
13. |      9 | 高鹏飞 | 男     | 13566889900 | gpf@163.com  |      2 |
14. +-----+-----+-----+-----+-----+-----+
15. 9 rows in set (0.00 sec)

```

数据成功添加

步骤三：向工资表添加数据，创建add_salary.py文件，添加如下代码：

```

01. from dbconn import Salary, Session
02.
03. jan2018_1 = Salary( date='2018-01-10', emp_id=1, basic=10000, awards=2000)
04. jan2018_2 = Salary( date='2018-01-10', emp_id=2, basic=11000, awards=1500) Top
05. jan2018_3 = Salary( date='2018-01-10', emp_id=3, basic=11000, awards=2200)
06. jan2018_4 = Salary( date='2018-01-10', emp_id=4, basic=11000, awards=3000)

```

```

07.  jan2018_5 = Salary( date='2018-01-10', emp_id=1, basic=13000, awards=2000)
08.  jan2018_6 = Salary( date='2018-01-10', emp_id=6, basic=15000, awards=3000)
09.  jan2018_7 = Salary( date='2018-01-10', emp_id=7, basic=9000, awards=3000)
10.  jan2018_8 = Salary( date='2018-01-10', emp_id=8, basic=13000, awards=2000)
11.  jan2018_9 = Salary( date='2018-01-10', emp_id=9, basic=13000, awards=1500)
12.  session = Session()
13.  sals = [jan2018_1, jan2018_2, jan2018_3, jan2018_4, jan2018_5, jan2018_6, jan2018_7, jan2
14.  session.add_all( sals)
15.  session.commit()
16.  session.close()

```

测试执行结果：

```
01.  [root@localhost day 10] # python3 add_salary.py
```

登录数据库查看部门表中数据

```

01.  MariaDB [tarena] > select * from salary;
02.  +-----+-----+-----+-----+-----+
03.  | auto_id | date      | emp_id | basic | awards |
04.  +-----+-----+-----+-----+-----+
05.  |      1 | 2018-01-10 |      1 | 10000 |    2000 |
06.  |      2 | 2018-01-10 |      2 | 11000 |    1500 |
07.  |      3 | 2018-01-10 |      3 | 11000 |    2200 |
08.  |      4 | 2018-01-10 |      4 | 11000 |    3000 |
09.  |      5 | 2018-01-10 |      1 | 13000 |    2000 |
10.  |      6 | 2018-01-10 |      6 | 15000 |    3000 |
11.  |      7 | 2018-01-10 |      7 | 9000  |    3000 |
12.  |      8 | 2018-01-10 |      8 | 13000 |    2000 |
13.  |      9 | 2018-01-10 |      9 | 13000 |    1500 |
14.  +-----+-----+-----+-----+-----+
15.  9 rows in set (0.00 sec)

```

数据成功添加

[Top](#)