[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON2/DAY04/CASE/01/index.html" \l "page_top_case)

# NSD Python2 DAY04

1. [案例1：向表中添加数据](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON2/DAY04/CASE/01/index.html" \l "case1)
2. [案例2：创建表](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON2/DAY04/CASE/01/index.html" \l "case2)
3. [案例3：添加数据](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON2/DAY04/CASE/01/index.html" \l "case3)
4. [案例4：操作数据](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/PYTHON2/DAY04/CASE/01/index.html" \l "case4)

## 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

1. [root@localhost ~]# yum install -y gcc
2. 已加载插件：fastestmirror, langpacks
3. dvd | 3.6 kB 00:00
4. Loading mirror speeds from cached hostfile
5. 匹配 gcc-4.8.5-16.el7.x86\_64 的软件包已经安装。正在检查更新。
6. 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

1. [root@localhost ~]# mkdir ~/.pip
2. [root@localhost ~]# vim ~/.pip/pip.conf
3. [global]
4. index-url = http://pypi.douban.com/simple/
5. [install]
6. trusted-host=pypi.douban.com

3)安装pymysql

1. [root@localhost ~]# pip3 install pymysql

步骤二：安装mariadb-server

1. [root@localhost ~]# yum install –y mariadb-server
2. ....
3. 已安装：
4. mariadb-server.x86\_64 1:5.5.56-2.el7
5. 作为依赖被安装：
6. mariadb.x86\_64 1:5.5.56-2.el7
7. perl-DBD-MySQL. x86\_64 0:4.023-5.el7
8. 完毕！
9. [root@localhost ~]# systemctl start mariadb
10. [root@localhost ~]# systemctl enable mariadb
11. [root@localhost ~]# mysqladmin password tedu.cn

步骤三：创建数据库

1)创建数据库

1. [root@localhost ~]# mysql -uroot -ptedu.cn
2. MariaDB [(none)]> CREATE DATABASE tedu DEFAULT CHARSET 'utf8';
3. 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

1. MariaDB [(none)]> USE tedu;
2. Database changed
3. MariaDB [tedu]> CREATE TABLE departments(dep\_id INT PRIMARY KEY, dep\_name VARCHAR(20));
4. 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

1. MariaDB [tedu]> CREATE TABLE employees (emp\_id INT PRIMARY KEY, emp\_name VARCHAR(20) NOT NULL, birth\_date DATE, phone CHAR(11), email VARCHAR(50), dep\_id INT, FOREIGN KEY(dep\_id) REFERENCES departments(dep\_id));
2. 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

1. MariaDB [tedu]> CREATE TABLE salary(auto\_id INT AUTO\_INCREMENT PRIMARY KEY, date DATE, emp\_id INT, basic INT, awards INT, FOREIGN KEY(emp\_id) REFERENCES employees(emp\_id));
2. 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：

1. [root@localhost day10]# vim insert\_data.py
2. import pymysql
3. 1)连接数据库
4. conn = pymysql.connect(
5. host='127.0.0.1',        #连接ip
6. port=3306,            #端口号
7. user='root',            #数据库用户名
8. passwd='tedu.cn',        #数据库密码
9. db='tedu',            #数据库名
10. charset='utf8'        #设置了数据库的字符集
11. )
12. 2)创建游标
13. cursor = conn.cursor()
14. 3)向部门表departments中插入数据
15. insert1 = "INSERT INTO departments(dep\_id, dep\_name) VALUES(%s, %s)"
16. result = cursor.execute(insert1, (1, '人事部'))        # execute执行insert语句
17. 4)将更新提交到数据库
18. conn.commit()
19. 5)关闭游标
20. cursor.close()
21. 6)关闭数据库连接
22. conn.close()

2)执行insert\_data.py文件：

1. [root@localhost day10]# python3 insert\_data.py

3)登录mariadb查看结果：

1. MariaDB [tedu]>> select \* from departments;
2. +--------+-----------+
3. | dep\_id | dep\_name |
4. +--------+-----------+
5. | 1 | 人事部 |
6. +--------+-----------+
7. 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：

1. #以上insert\_data.py文件第3步可用如下代码替换：
2. insert1 = "INSERT INTO departments(dep\_id, dep\_name) VALUES(%s, %s)"
3. data = [(2, '运维部'), (3, '开发部')]
4. cursor.executemany(insert1, data)

mariadb查看结果如下：

1. MariaDB [tedu]>> select \* from departments;
2. +--------+-----------+
3. | dep\_id | dep\_name |
4. +--------+-----------+
5. | 1 | 人事部 |
6. | 2 | 运维部 |
7. | 3 | 开发部 |
8. +--------+-----------+
9. 3 rows in set (0.01 sec)

步骤五：向employees表插入数据

1)新建insert\_emp.py文件，编写代码如下：

1. [root@localhost day10]# vim insert\_emp.py
2. import pymysql
3. 1)连接数据库
4. conn = pymysql.connect(
5. host='127.0.0.1',        #连接ip
6. port=3306,            #端口号
7. user='root',            #数据库用户名
8. passwd='tedu.cn',        #数据库密码
9. db='tedu',            #数据库名
10. charset='utf8'        #设置了数据库的字符集
11. )
12. 2)创建游标
13. cursor = conn.cursor()
14. 3)向部门表employees中插入数据
15. insert1 = "INSERT INTO employees(emp\_id, emp\_name, birth\_date,phone, email, dep\_id) VALUES(%s, %s, %s, %s, %s, %s)"
16. result = cursor.execute(insert1, (1, '王君', '2018-9-30',\
17. '15678789090', 'wj@163.com', 3))        # 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()

2)执行insert\_emp.py文件：

1. [root@localhost day10]# python3 insert\_emp.py

3)登录mariadb查看结果：

1. MariaDB [tedu]>> select \* from employees;
2. +--------+----------+------------+-------------+------------+--------+
3. | emp\_id | emp\_name | birth\_date | phone | email | dep\_id |
4. +--------+----------+------------+-------------+------------+--------+
5. | 1 | 王君 | 2018-09-30 | 15678789090 | wj@163.com | 3 |
6. +--------+----------+------------+-------------+------------+--------+
7. 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：

1. #以上insert\_emp.py文件第3步可用如下代码替换：
2. insert1 = "INSERT INTO employees (dep\_id, dep\_name) VALUES(%s, %s)"
3. data = [(2, '运维部'), (3, '开发部')]
4. cursor.executemany(insert1, data)

mariadb查看结果如下：

1. MariaDB [tedu]>> select \* from departments;
2. +--------+----------+------------+-------------+------------+--------+
3. | emp\_id | emp\_name | birth\_date | phone | email | dep\_id |
4. +--------+----------+------------+-------------+------------+--------+
5. | 1 | 王君 | 2018-09-30 | 15678789090 | wj@163.com | 3 |
6. | 2 | 李雷 | 2018-09-30 | 15678789090 | wj@163.com | 2 |
7. | 3 | 张美 | 2018-09-30 | 15678789090 | zm@163.com | 1 |
8. +--------+----------+------------+-------------+------------+--------+
9. 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：

1. [root@localhost day10]# vim insert\_sal.py
2. import pymysql
3. 1)连接数据库
4. conn = pymysql.connect(
5. host='127.0.0.1',        #连接ip
6. port=3306,            #端口号
7. user='root',            #数据库用户名
8. passwd='tedu.cn',        #数据库密码
9. db='tedu',            #数据库名
10. charset='utf8'        #设置了数据库的字符集
11. )
12. 2)创建游标
13. cursor = conn.cursor()
14. 3)向部门表salary中插入数据
15. insert2 = "INSERT INTO salary(date, emp\_id,basic, awards) VALUES(%s, %s, %s, %s)"
16. data = [('2018-9-30', 2, 1000, 2000), ('2018-9-30', 3, 3000, 6000),('2018-9-30', 1, 8000, 9000)]
17. cursor.executemany(insert2, data)
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()

2)执行insert\_sal.py文件：

1. [root@localhost day10]# python3 insert\_sal.py

3)登录mariadb查看结果：

1. MariaDB [tedu]>> select \* from salary;
2. +---------+------------+--------+-------+--------+
3. | auto\_id | date | emp\_id | basic | awards |
4. +---------+------------+--------+-------+--------+
5. | 1 | 2018-09-30 | 2 | 1000 | 2000 |
6. | 2 | 2018-09-30 | 3 | 3000 | 6000 |
7. | 3 | 2018-09-30 | 1 | 8000 | 9000 |
8. +---------+------------+--------+-------+--------+
9. 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

### 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

1. [root@serwang ~]# pip3 install sqlalchemy
2. Collecting sqlalchemy
3. Downloading http://pypi.doubanio.com/packages/aa/cc/48eec885d81f7260b07d
4. 961b3ececfc0aa82f7d4a8f45ff997e0d3f44ba/SQLAlchemy-1.2.11.tar.gz (5.6MB)
5. ...
6. ...
7. Installing collected packages: sqlalchemy
8. Running setup.py install for sqlalchemy ... done
9. 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

1. 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

1. [root@localhost day10]# vim dbconn.py
2. #!/usr/bin/env python3
3. from sqlalchemy import create\_engine
4. # 创建连接到数据库的引擎
5. engine = create\_engine(
6. #指定数据库、用户名、密码、连接到哪台服务器、库名等信息
7. 'mysql+pymysql://root:tedu.cn@localhost/tarena?charset=utf8',
8. encoding='utf8',
9. echo=True    #终端输出
10. )

2)创建ORM映射，生成ORM映射所需的基类

1. from sqlalchemy.ext.declarative import declarative\_base
2. Base = declarative\_base()

3)自定义映射类，创建部门表

1. from sqlalchemy import Column, String, Integer
2. class Departments(Base): # 必须继承于Base
3. \_\_tablename\_\_ = 'departments' # 库中的表名
4. # 每个属性都是表中的一个字段，是类属性
5. dep\_id = Column(Integer, primary\_key=True)    #Integer整数类型,primary\_key主键
6. # String字符串类型，nullable非空约束，unique唯一性约束
7. dep\_name = Column(String(20), nullable=False, unique=True)
8. def \_\_str\_\_(self):
9. return '[部门ID：%s, 部门名称：%s]' % (self.dep\_id, self.dep\_name)
10. if \_\_name\_\_ == '\_\_main\_\_':
11. # 在数据库中创建表，如果库中已有同名的表，将不会创建
12. Base.metadata.create\_all(engine)

4)测试脚本执行，生成部门表

1. [root@localhost day10]# python3 dbconn.py    #成功生成部门表

5)进入数据库查看结果

1. #登录数据库
2. [root@localhost day10]# mysql -uroot -ptedu.cn
3. Welcome to the MariaDB monitor. Commands end with ; or \g.
4. Your MariaDB connection id is 5
5. Server version: 5.5.56-MariaDB MariaDB Server
6. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
7. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
8. #查看数据库表
9. MariaDB [(none)]> use tarena;
10. Reading table information for completion of table and column names
11. You can turn off this feature to get a quicker startup with -A
12. Database changed
13. MariaDB [tarena]> show tables;
14. +------------------+
15. | Tables\_in\_tarena |
16. +------------------+
17. | departments |
18. +------------------+
19. 1 row in set (0.00 sec)
20. MariaDB [tarena]> show create table departments;
21. +-------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
22. | Table | Create Table |
23. +-------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
24. | departments | CREATE TABLE `departments` (
25. `dep\_id` int(11) NOT NULL AUTO\_INCREMENT,
26. `dep\_name` varchar(20) NOT NULL,
27. PRIMARY KEY (`dep\_id`),
28. UNIQUE KEY `dep\_name` (`dep\_name`)
29. ) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
30. +-------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
31. 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

1. from sqlalchemy import ForeignKey    导入外键
2. class Employees(Base): # 必须继承于Base
3. \_\_tablename\_\_ = 'employees' # 库中的表名
4. # 每个属性都是表中的一个字段，是类属性
5. emp\_id = Column(Integer, primary\_key=True)     #Integer整数类型,primary\_key主键
6. name = Column(String(20), nullable=False) # String字符串类型，nullable非空约束
7. gender = Column(String(6))
8. phone = Column(String(11))
9. email = Column(String(50))
10. dep\_id = Column(Integer, ForeignKey('departments.dep\_id'))    #与departments中dep\_id做外键关联
11. def \_\_str\_\_(self):
12. return '员工：%s' % self.name

4)测试脚本执行，生成员工表

1. [root@localhost day10]# python3 dbconn.py    #成功生成员工表

5)进入数据库查看结果

1. #登录数据库
2. [root@localhost day10]# mysql -uroot -ptedu.cn
3. Welcome to the MariaDB monitor. Commands end with ; or \g.
4. Your MariaDB connection id is 5
5. Server version: 5.5.56-MariaDB MariaDB Server
6. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
7. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
8. #查看数据库表
9. MariaDB [(none)]> use tarena;
10. Reading table information for completion of table and column names
11. You can turn off this feature to get a quicker startup with -A
12. Database changed
13. MariaDB [tarena]> show tables;
14. +------------------+
15. | Tables\_in\_tarena |
16. +------------------+
17. | departments |
18. | employees |
19. +------------------+
20. 2 rows in set (0.00 sec)

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

1. from sqlalchemy import Date    导入外键
2. class Employees(Base): # 必须继承于Base
3. \_\_tablename\_\_ = 'employees' # 库中的表名
4. # 每个属性都是表中的一个字段，是类属性
5. emp\_id = Column(Integer, primary\_key=True)     #Integer整数类型,primary\_key主键
6. name = Column(String(20), nullable=False) # String字符串类型，nullable非空约束
7. gender = Column(String(6))
8. phone = Column(String(11))
9. email = Column(String(50))
10. dep\_id = Column(Integer, ForeignKey('departments.dep\_id'))    #与departments中dep\_id做外键关联
11. def \_\_str\_\_(self):
12. return '员工：%s' % self.name
13. class Salary(Base): # 必须继承于Base
14. \_\_tablename\_\_ = 'salary' # 库中的表名
15. auto\_id = Column(Integer, primary\_key=True) #Integer整数类型,primary\_key主键
16. date = Column(Date)    #导入日期
17. emp\_id = Column(Integer, ForeignKey('employees.emp\_id')) #与employees中emp\_id做外键关联
18. basic = Column(Integer)        #基本工资
19. awards = Column(Integer)    #奖金

4)测试脚本执行，生成员工表

1. [root@localhost day10]# python3 dbconn.py    #成功生成工资表

5)进入数据库查看结果

1. #登录数据库
2. [root@localhost day10]# mysql -uroot -ptedu.cn
3. Welcome to the MariaDB monitor. Commands end with ; or \g.
4. Your MariaDB connection id is 5
5. Server version: 5.5.56-MariaDB MariaDB Server
6. Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
7. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
8. #查看数据库表
9. MariaDB [(none)]> use tarena;
10. Reading table information for completion of table and column names
11. You can turn off this feature to get a quicker startup with -A
12. Database changed
13. MariaDB [tarena]> show tables;
14. +------------------+
15. | Tables\_in\_tarena |
16. +------------------+
17. | departments |
18. | employees |
19. | salary |
20. +------------------+
21. 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

1. [root@localhost day10]# vim add\_department.py
2. from dbconn import Departments
3. hr = Departments(dep\_id=1, dep\_name='hr')
4. print(hr.dep\_name)
5. print(hr.dep\_id)

测试执行结果：

1. [root@localhost day10]# python3 add\_department.py
2. hr
3. 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

1. MariaDB [tarena]> select \* from departments;
2. 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

1. from sqlalchemy.orm import sessionmaker
2. Session = sessionmaker(bind=engine)

3)添加新对象

在add\_department.py文件中添加如下代码：

1. from dbconn import Departments, Session
2. session = Session()    #创建会话类实例
3. session.add(hr)    #向实例绑定的数据库添加数据
4. session.commit()    #将数据提交到实例对应数据库
5. session.close()    #关闭session类

测试执行结果：

1. [root@localhost day10]# python3 add\_department.py

登录数据库查看部门表中数据

1. MariaDB [tarena]> select \* from departments;
2. +--------+----------+
3. | dep\_id | dep\_name |
4. +--------+----------+
5. | 1 | hr |
6. +--------+----------+
7. 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：

1. ops = Departments(dep\_id=2, dep\_name='operations')
2. dev = Departments(dep\_id=3, dep\_name='development')
3. finance = Departments(dep\_id=4, dep\_name='财务部')
4. deps = [ops, dev]
5. session = Session()
6. session.add\_all(deps)
7. session.add(finance)
8. session.commit()
9. 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：

1. engine=create\_engine(
2. 'mysql+pymysql://root:tedu.cn@localhost/tarena?charset=uU8',
3. encoding='uU8',
4. echo=True
5. )

测试执行结果：

1. [root@localhost day10]# python3 add\_department.py

登录数据库查看部门表中数据

1. MariaDB [tarena]> select \* from departments;
2. +--------+-------------+
3. | dep\_id | dep\_name |
4. +--------+-------------+
5. | 3 | development |
6. | 1 | hr |
7. | 2 | operations |
8. | 4 | 财务部 |
9. +--------+-------------+
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：

1. from dbconn import Employees, Session
2. wj = Employees(
3. emp\_id=1,name='王俊',gender='男',phone='15678789090',email='wj@163.com', dep\_id=3
4. )
5. wwc = Employees(
6. emp\_id=2,name='吴伟超',gender='男',phone='13499887755',email='wwc@qq.com', dep\_id=3
7. )
8. dzj = Employees(
9. emp\_id=3, name='董枝俊', gender='男', phone='18900998877', email='dzj@163.com', dep\_id=3
10. )
11. ltd = Employees(
12. emp\_id=4, name='李通达', gender='男', phone='13378904567', email='ltd@163.com', dep\_id=2)
13. wxy = Employees(
14. emp\_id=5, name='王秀燕', gender='女', phone='15098765432', email='wxy@tedu.cn', dep\_id=2)
15. gq = Employees(
16. emp\_id=6, name='高琦', gender='女', phone='15876543212', email='gq@tarena.com', dep\_id=1)
17. wzf = Employees(
18. emp\_id=7, name='王召飞', gender='男', phone='15609871234', email='wzf@sohu.com', dep\_id=1)
19. sy = Employees(
20. emp\_id=8, name='孙燕', gender='女', phone='18567895435', email='sy@163.com', dep\_id=4)
21. gpf = Employees(
22. emp\_id=9, name='高鹏飞', gender='男', phone='13566889900', email='gpf@163.com', dep\_id=2)
23. emps = [wj, wwc, dzj, ltd, wxy, gq, wzf, sy, gpf]
24. session = Session()
25. session.add\_all(emps)
26. session.commit()
27. session.close()

测试执行结果：

1. [root@localhost day10]# python3 add\_employees.py

登录数据库查看部门表中数据

1. MariaDB [tarena]> select \* from employees;
2. +--------+-----------+--------+-------------+---------------+--------+
3. | emp\_id | name | gender | phone | email | dep\_id |
4. +--------+-----------+--------+-------------+---------------+--------+
5. | 1 | 王俊 | 男 | 15678789090 | wj@163.com | 3 |
6. | 2 | 吴伟超 | 男 | 13499887755 | wwc@qq.com | 3 |
7. | 3 | 董枝俊 | 男 | 18900998877 | dzj@163.com | 3 |
8. | 4 | 李通达 | 男 | 13378904567 | ltd@163.com | 2 |
9. | 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：

1. from dbconn import Salary, Session
2. jan2018\_1 = Salary(date='2018-01-10', emp\_id=1, basic=10000, awards=2000)
3. jan2018\_2 = Salary(date='2018-01-10', emp\_id=2, basic=11000, awards=1500)
4. jan2018\_3 = Salary(date='2018-01-10', emp\_id=3, basic=11000, awards=2200)
5. jan2018\_4 = Salary(date='2018-01-10', emp\_id=4, basic=11000, awards=3000)
6. jan2018\_5 = Salary(date='2018-01-10', emp\_id=1, basic=13000, awards=2000)
7. jan2018\_6 = Salary(date='2018-01-10', emp\_id=6, basic=15000, awards=3000)
8. jan2018\_7 = Salary(date='2018-01-10', emp\_id=7, basic=9000, awards=3000)
9. jan2018\_8 = Salary(date='2018-01-10', emp\_id=8, basic=13000, awards=2000)
10. jan2018\_9 = Salary(date='2018-01-10', emp\_id=9, basic=13000, awards=1500)
11. session = Session()
12. sals = [jan2018\_1, jan2018\_2, jan2018\_3,jan2018\_4, jan2018\_5, jan2018\_6, jan2018\_7, jan2018\_8, jan2018\_9]
13. session.add\_all(sals)
14. session.commit()
15. session.close()

测试执行结果：

1. [root@localhost day10]# python3 add\_ salary.py

登录数据库查看部门表中数据

1. MariaDB [tarena]> select \* from salary;
2. +---------+------------+--------+-------+--------+
3. | auto\_id | date | emp\_id | basic | awards |
4. +---------+------------+--------+-------+--------+
5. | 1 | 2018-01-10 | 1 | 10000 | 2000 |
6. | 2 | 2018-01-10 | 2 | 11000 | 1500 |
7. | 3 | 2018-01-10 | 3 | 11000 | 2200 |
8. | 4 | 2018-01-10 | 4 | 11000 | 3000 |
9. | 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

## 4 案例4：操作数据

### 4.1 问题

1. 修改部门表，将人事部改为人力资源部
2. 如果存在设计部，将设计部删除
3. 查询所有每个员工及其所在部门

### 4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：修改表中的记录，首先需要把记录找到

1. qset12 = session.query(Department).filter(Department.dep\_name=='人事部')
2. hr = qset12.one()

步骤二：只要把映射的实体类对象重新赋值保存即可

1. hr.dep\_name='人力资源部'
2. session.commit()

步骤三：删除记录与修改类似，先找到记录再执行删除

1. qset13 = session.query(Employee).filter(Department.dep\_name=='设计部')
2. dep = qset13.one()
3. session.delete(dep)
4. session.commit()

步骤四：查询每个员工所在的部门涉及的是多表查询，因为员工表中只有部门ID，部门名称在部门表中存储

1. qset10 = session.query(Employee.emp\_name, Department.dep\_name).join(Department)
2. for row in qset10:
3. print(row)