



AniPython

pandas写/读mysql

第一步

1. 安装 mysql 数据库，并打开服务

2. 安装依赖

```
$ pip install pymysql
```

```
$ pip install sqlalchemy
```


第二步

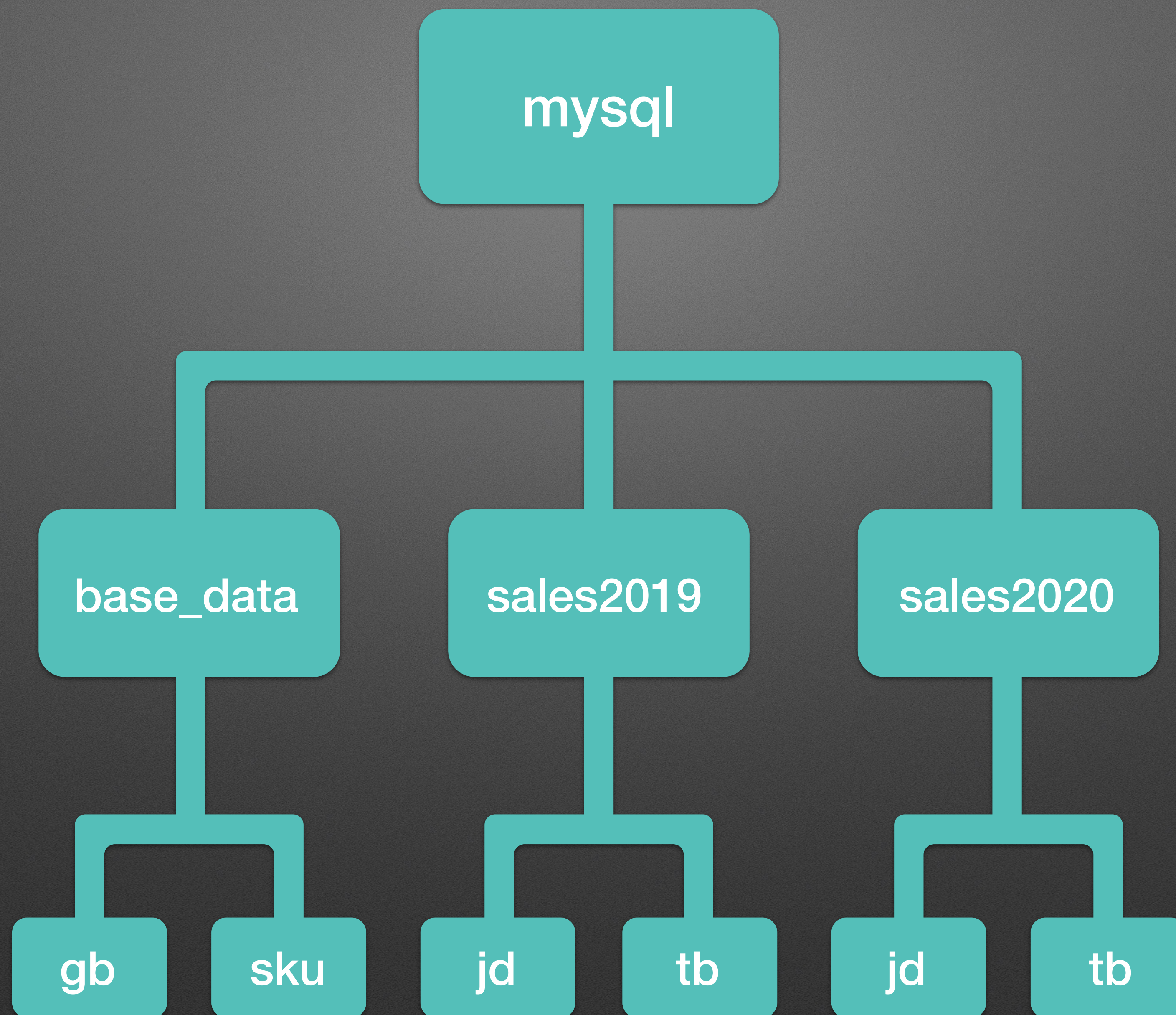
手动创建一个库和一个表, 如:

库名: sales2020

表名: jd

库(database)

表(table)



第三步

```
from sqlalchemy import create_engine  
con = create_engine(  
    "mysql+pymysql://root:password@localhost/dbname" )
```



用户名



密码



数据库地址



库名

第四步

写sql

```
df.to_sql('tb_name', con=con, index=False, if_exists='append')
```

读sql

```
df1 = pd.read_sql('tb_name', con=con, columns=['a', 'b'])
```

```
df2 = pd.read_sql("select a,b from tb_name", con=con)
```

执行sql: 增(insert)/删(delete)/改(update)操作

```
con.execute("delete from tb_name where id=1")
```


pandas写mysql

```
DataFrame.to_sql(self, name, con, if_exists,  
                  index, index_label, chunksize)
```

name: str

SQL表名.

con: sqlalchemy.engine.Engine

if_exists: {'fail', 'replace', 'append'}, default 'fail'

如果表已经存在的行为

fail: 引发ValueError异常

replace: 在插入新值之前删除表

append: 向现有表插入新值

index: bool, default True

index是否作为一列进行写入

index_label: str or sequence, default None

设置写入index时的列名, 如果为None, 则使用原索引名

chunksize: int

分批写入, 指定每批行数

pandas读mysql

```
DataFrame.read_sql(self, sql, con, columns)
```

sql: str

SQL语句 or 表名

con: sqlalchemy.engine.Engine

columns: list, default: None

选择的列名列表(仅在sql参数为表名时使用)

总结

1. 安装 mysql 数据库

安装依赖

```
$ pip install pymysql
```

```
$ pip install sqlalchemy
```

2. 手动创建一个库，一个表

3. 创建连接引擎

```
con = sqlalchemy.create_engine(  
    "mysql+pymysql://root:password@localhost/dbname")
```

4. "增删改查"操作

```
DataFrame.to_sql()    # 写  
pandas.read_sql()     # 读  
con.execute()         # 执行sql语句
```