

实验一 数据库平台与MySQL语言基础（1）

实验一 数据库平台与MySQL语言基础（1）

一、实验目的

二、实验内容

三、实验仪器、设备及材料

四、实验内容简介

五、实验步骤

5.1 DataGrip的启动与退出

5.2 DataGrip的项目创建与打开

5.3 建立MySQL数据库连接

5.4 创建与删除一个数据库（模式）

5.5 表的建立、修改与删除

5.6 用图形界面插入、修改、删除数据

5.7 单表查询

一、实验目的

（1）本章实验的内容主要涵盖教材第三章的部分内容，通过上机实验以加深对语言的理解；

（2）建立MySQL的工作空间和应用，初步了解SQL语言。

二、实验内容

本实验主要包括如下部分的内容。

- DataGrip的启动与退出
- DataGrip的项目的创建与打开
- 建立MySQL数据库连接
- 创建与删除数据库
- 表的创建、修改与删除
- 用图形界面插入修改与删除数据
- 单表查询

三、实验仪器、设备及材料

硬件：计算机；《C#语言与数据库技术基础教程》

软件：Windows 操作系统，[MySQL](#)，[DataGrip](#)

四、实验内容简介

结构化查询语言(Structured Query Language, SQL)，是关系数据库的标准语言，MySQL也是最常用的数据库管理软件之一。本次上机所需的环境是MySQL Community 版本。虽然MySQL软件自带Workbench这个简单的图形化界面工具，然而并不便于使用。我们使用的图形化工具为Jetbrains开发的DataGrip软件。学生在校期间可申请免费的学生版。

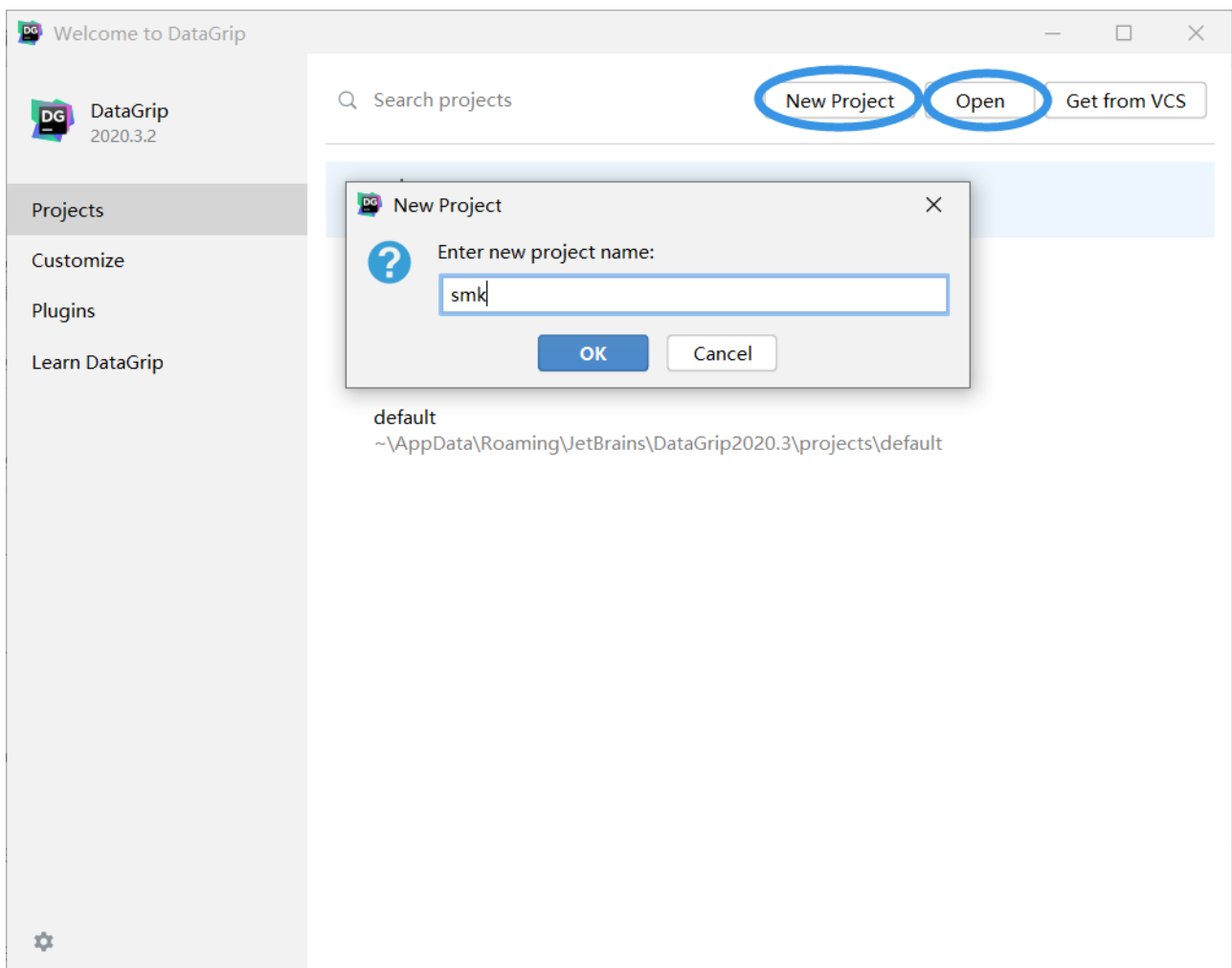
五、实验步骤

5.1 DataGrip的启动与退出

与其他常用软件类似，如Microsoft Word/PPT/Excel类似，此处不复赘述。

5.2 DataGrip的项目创建与打开

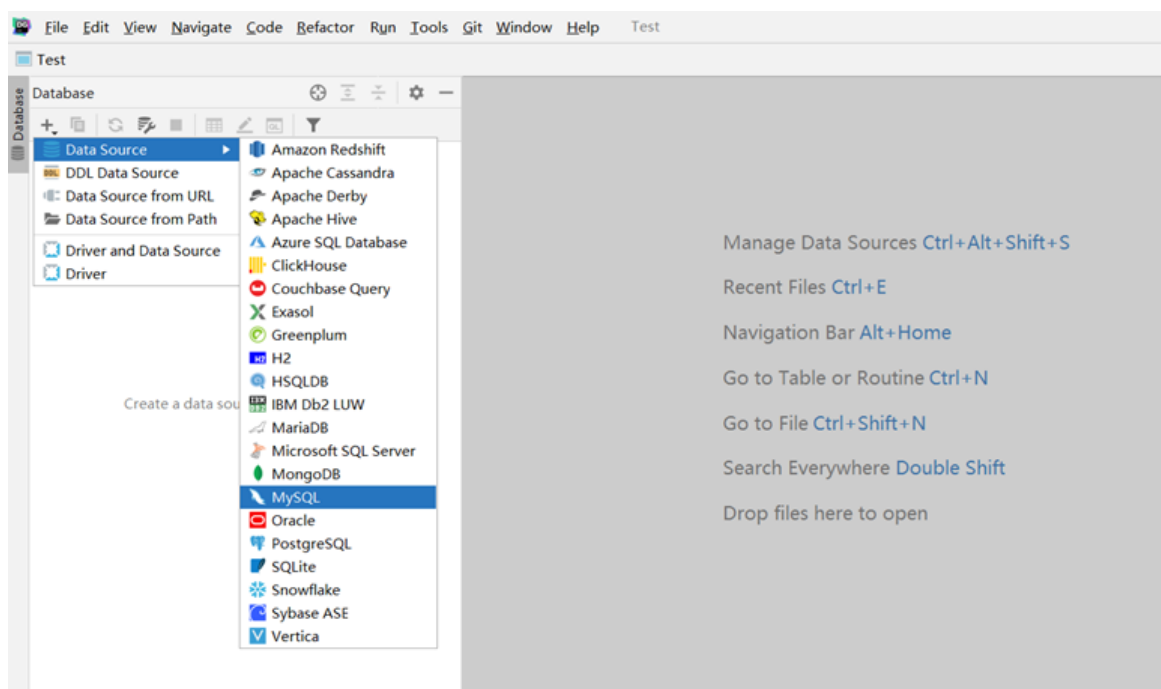
【任务一】 创建一个DataGrip项目。DataGrip需要项目才可以管理数据库。当电脑此前没有项目时，我们可创建一个；如果已经有某个未完成的项目，需要继续对它进行某些操作，则可以打开项目。



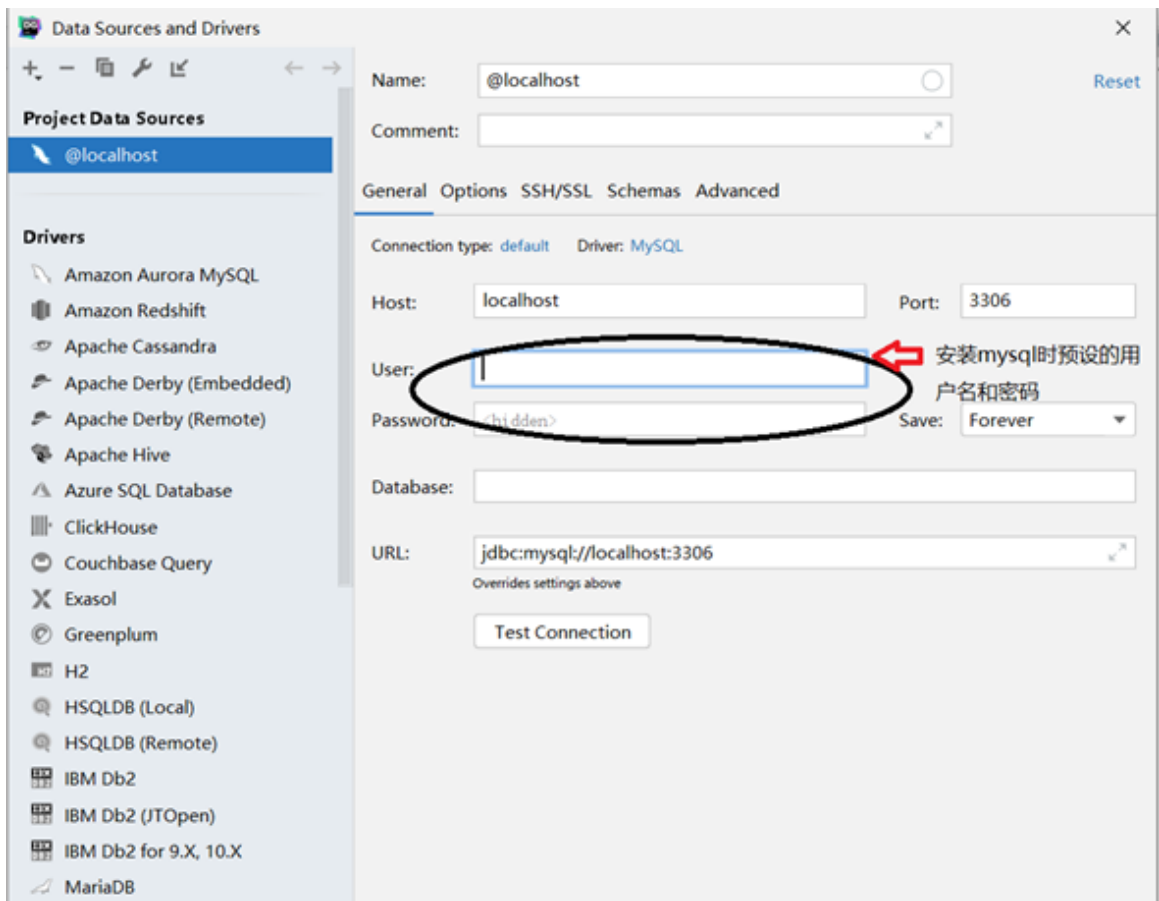
5.3 建立MySQL数据库连接

【任务二】 建立DataGrip项目与MySQL中localhost数据库服务器的连接。

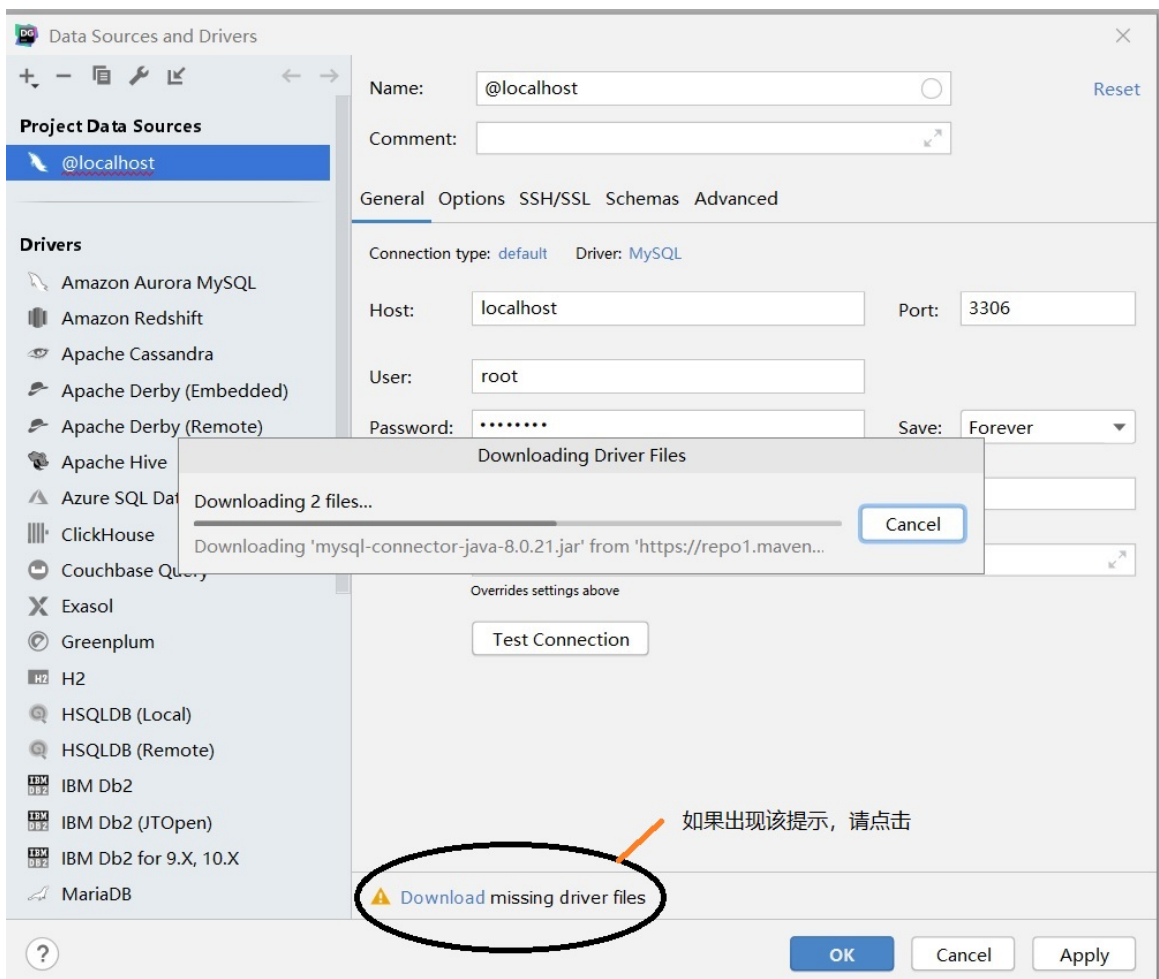
1. 右键新建data source的连接



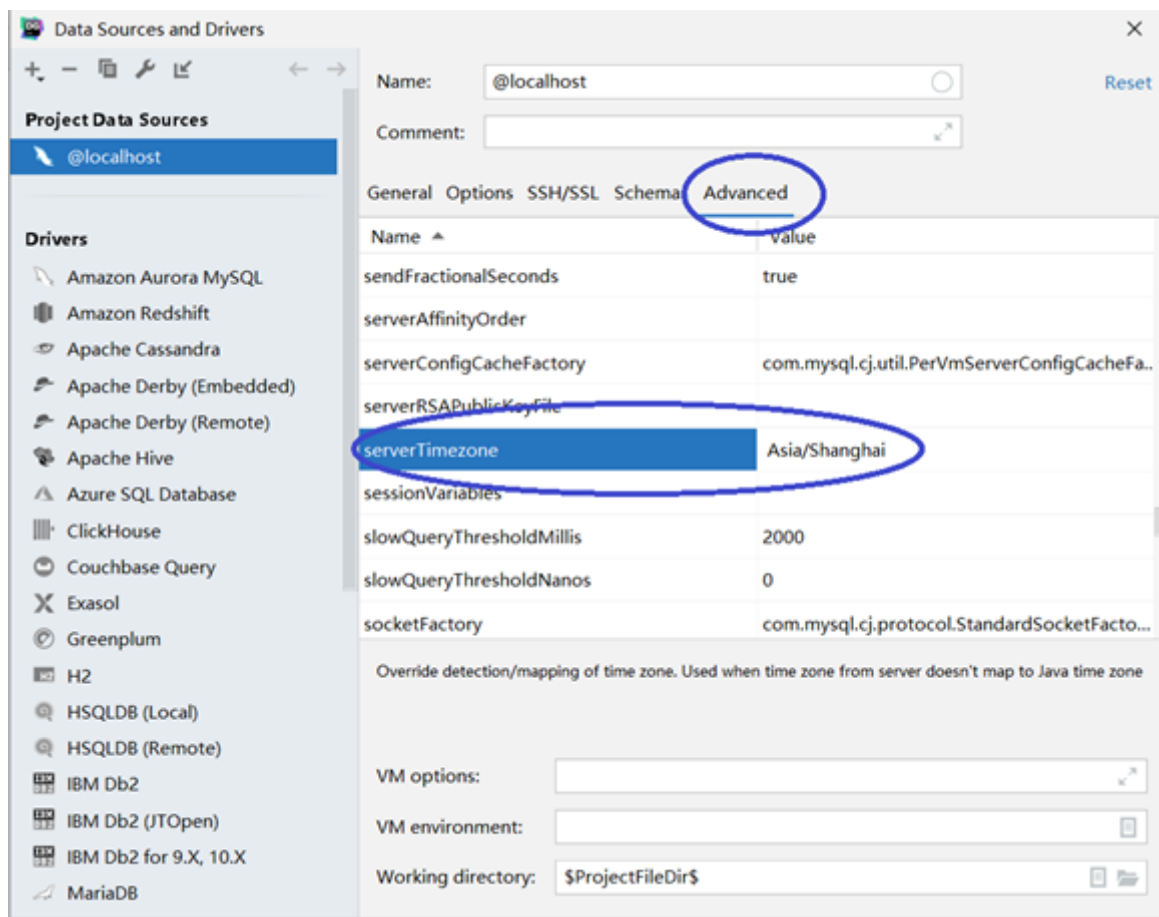
2. 填写安装MySQL时预设的用户名和密码。



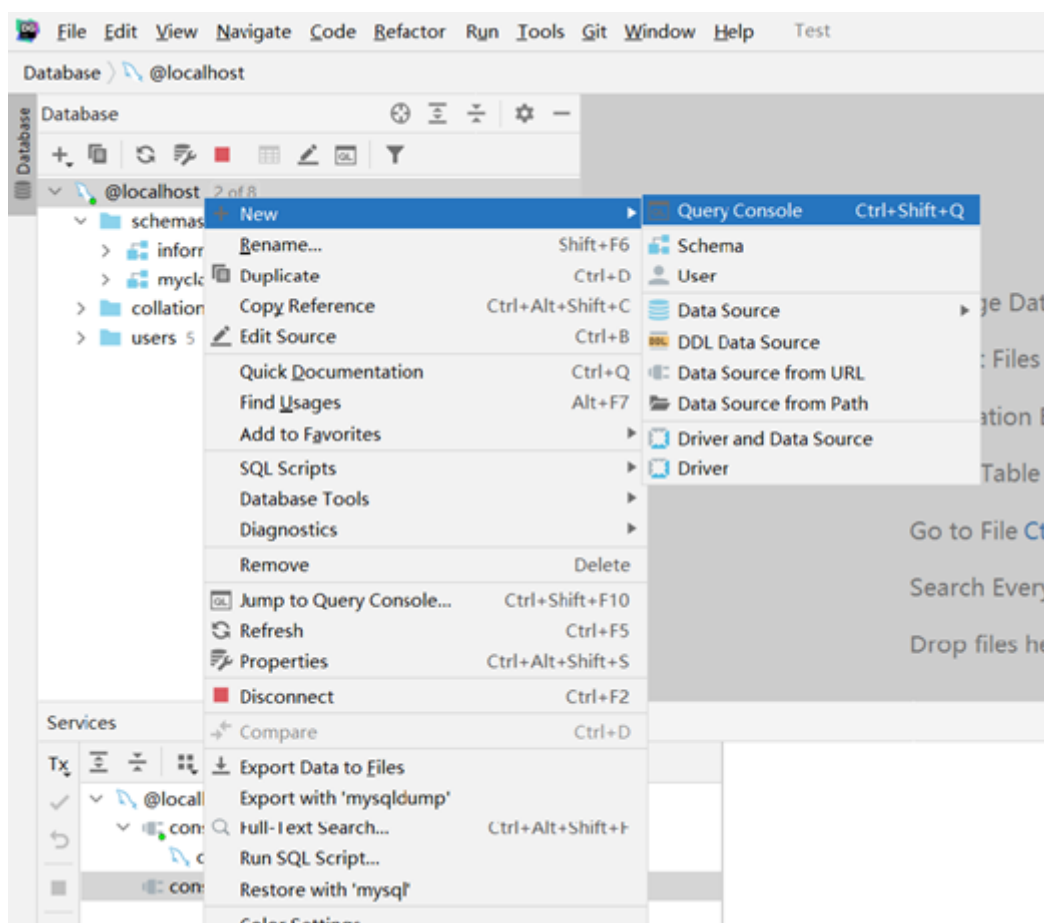
3. 安装所需驱动程序



4. 设置时区为Asia/Shanghai

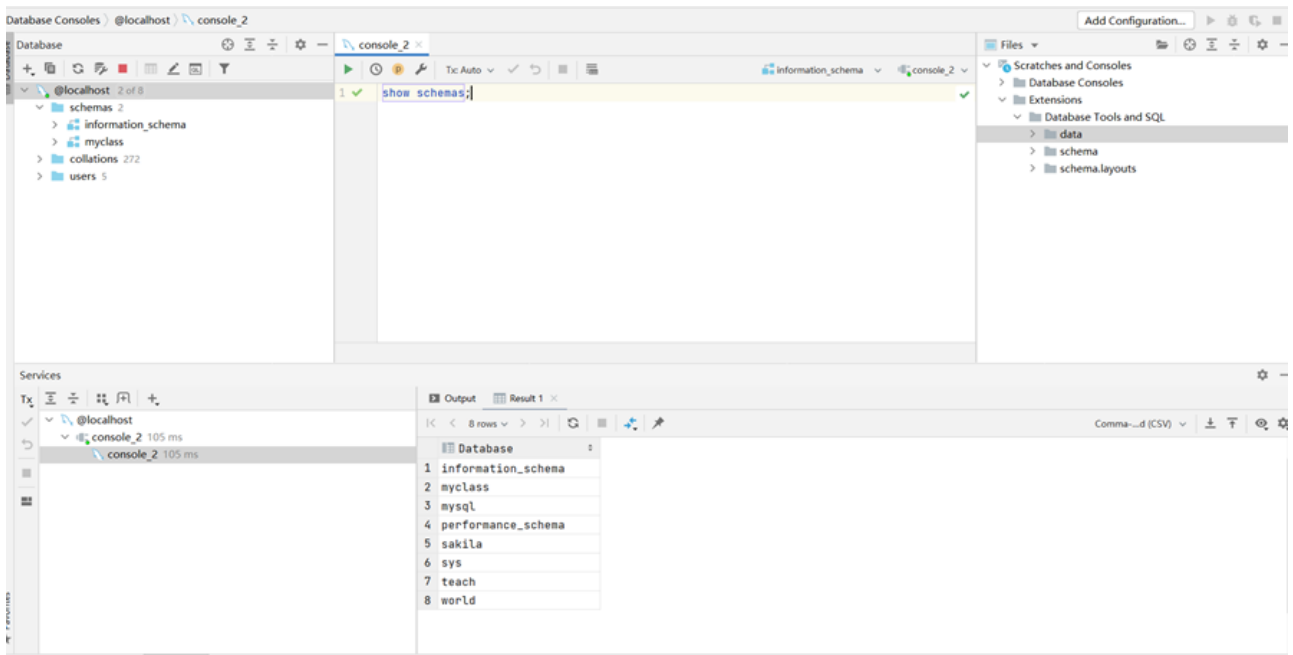


5. 连接成功。建立查询窗口。连接成功后，会打开一个默认查询窗口。如果没有可按照此步骤创建。



6. 打开查询窗口，查看已有的数据库。

```
show schemas;
```

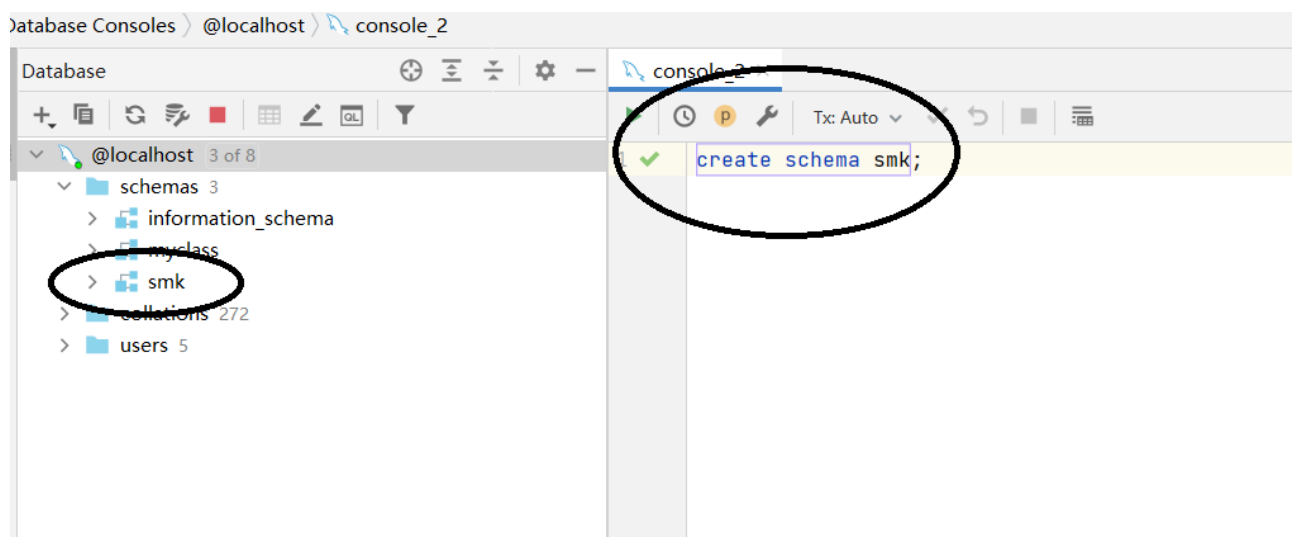


5.4 创建与删除一个数据库（模式）

【任务三】创建与删除一个数据库

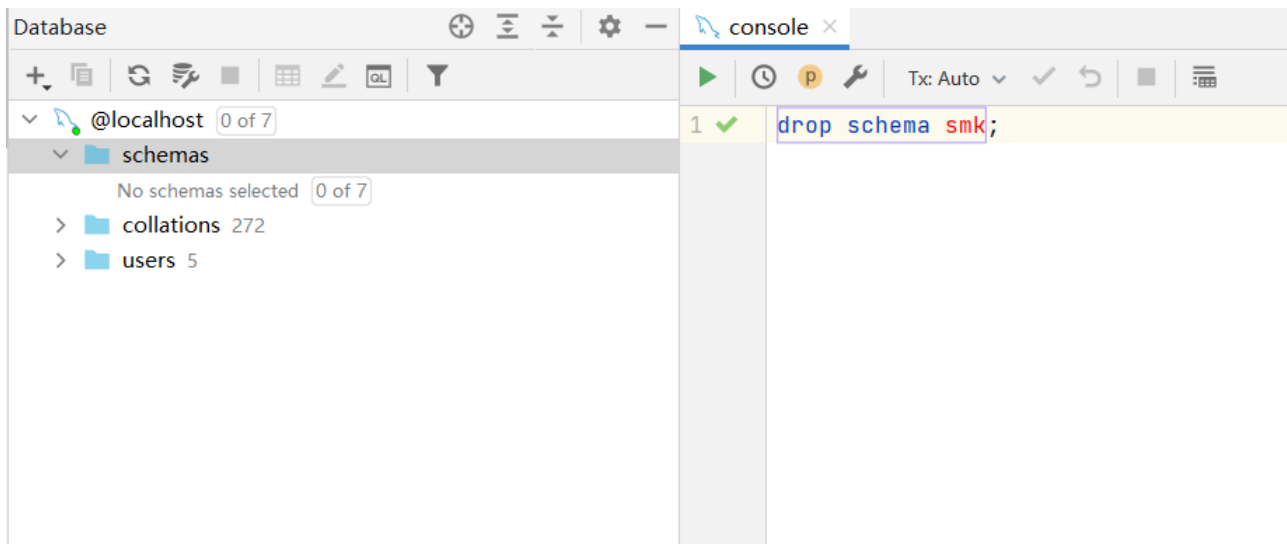
1. 创建数据库

```
create schema smk;
```



2. 删除数据库

```
drop schema smk;
```



5.5 表的建立、修改与删除

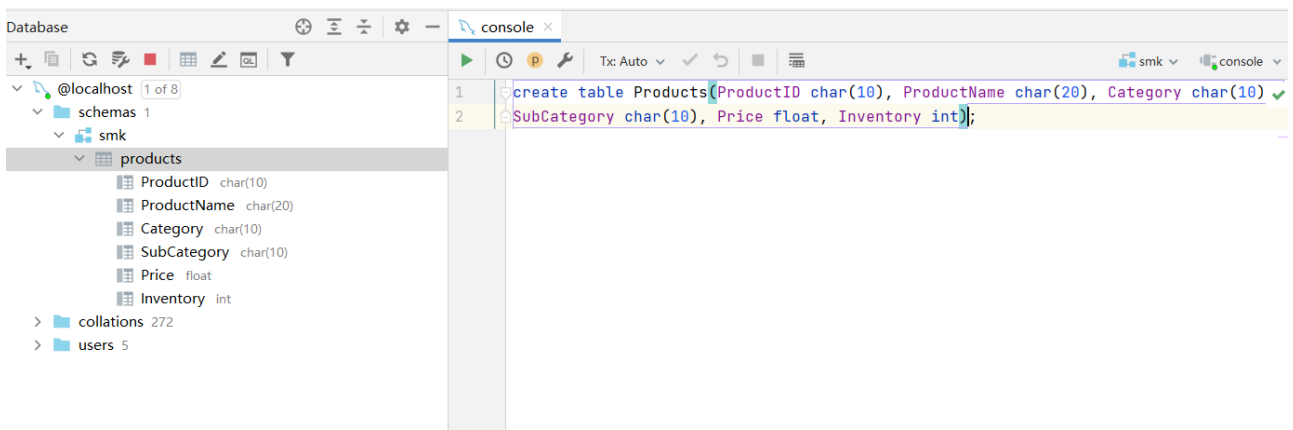
【任务三】表的创建。使用smk，并在数据库中添加 商品（**Product**），顾客（**Customer**）和销售（**Sale**）三个关系。商品表包括商品编号、商品名、分类、子类、价格和库存数量等属性；顾客表包括顾客编号、顾客名、性别、省、市、县等属性；销售表包括销售编号、日期、顾客编号、商品编号、销量和付款等属性。

1. 使用数据库smk

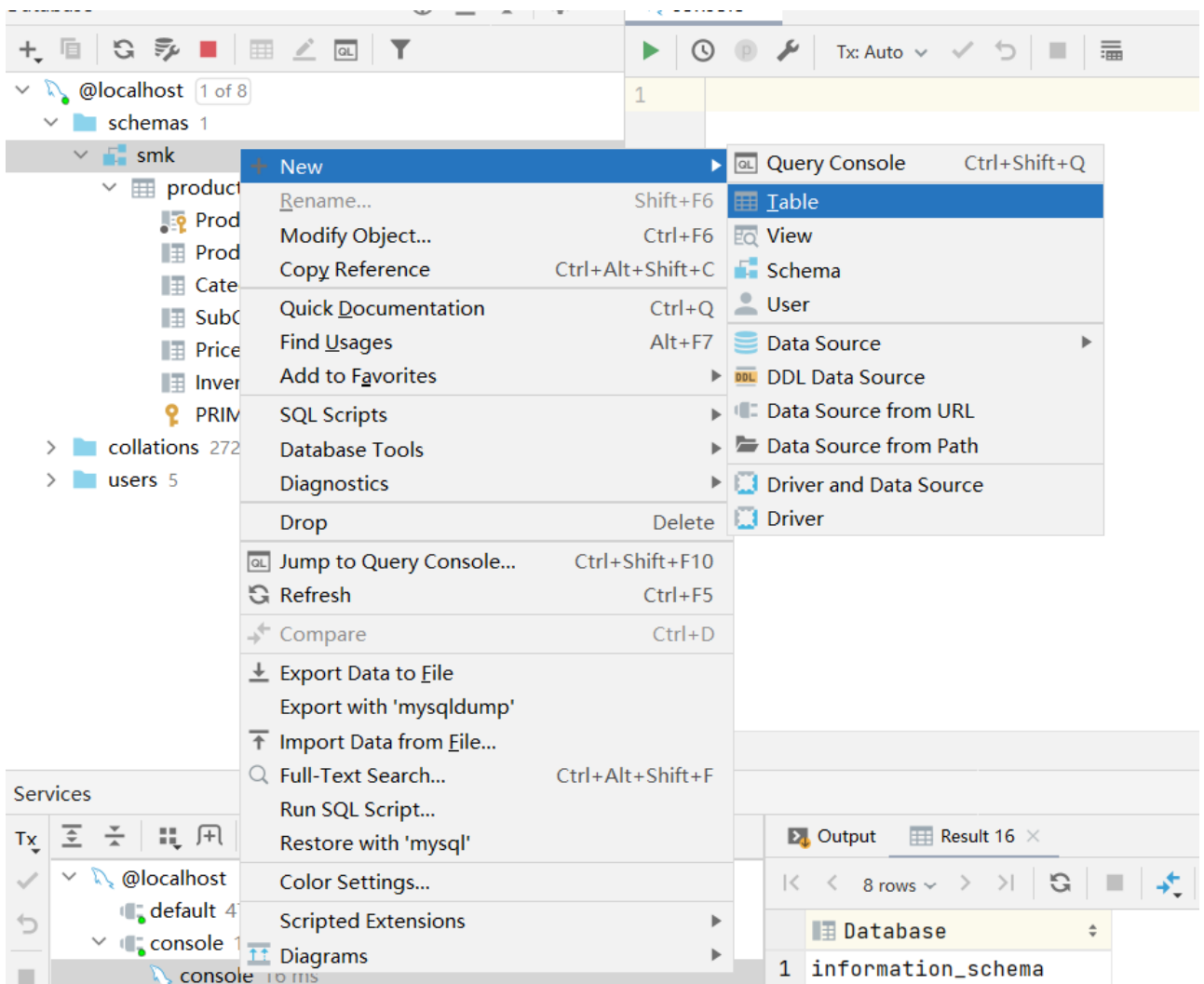
```
use smk;
```

2. 使用SQL语句创建Product表

```
create table product(ProductID char(10) primary key,  
                    ProductName char(20),  
                    Category char(10),  
                    SubCategory char(10),  
                    Price float,  
                    Inventory int);
```



3. 使用图形界面创建Customer表——打开命令窗口



4. 录入Customer表信息，点击Execute

CustomerID char(10)类型 not null 主键,
 CustomerName char(10)类型 允许null,
 Gender char(2)类型 允许null,
 Province char(10)类型 允许null,
 City char(10)类型 允许null,
 County char(10)类型 允许

Modify Table

Table: Comment:

Columns (6) | Keys (1) | Indexes | Foreign Keys

CustomerID	char(10)	-- part of primary key
CustomerName	char(10)	
Gender	char(2)	
Province	char(10)	
City	char(10)	
County	char(10)	

SQL Script

Action:

5. 动手做：使用以上任一方式创建Sale表。表格格式如下：

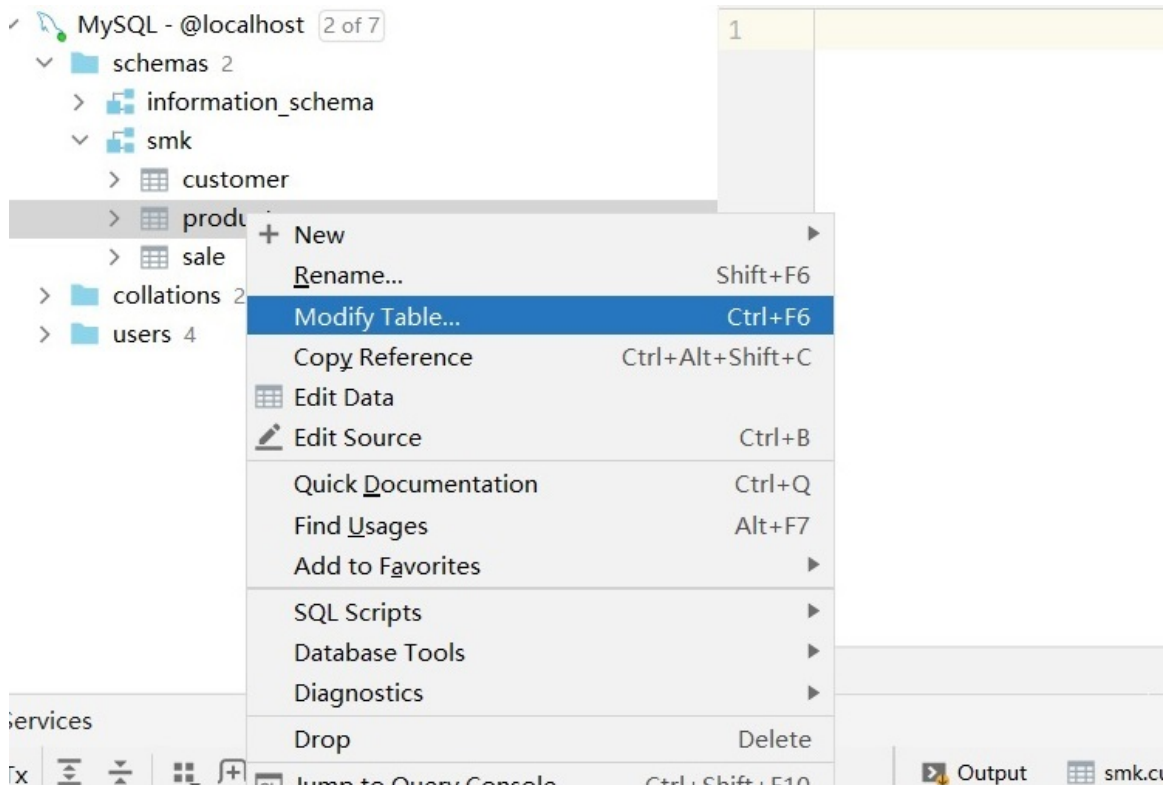
```

SaleID    int类型  not null 主键,
SaleDate  date类型  允许null,
CustomerID      char(10)类型  允许null,
ProductID      char(10)类型  允许null,
Amount         int类型  允许null,
Payment        float类型  允许

```

【任务四】表的修改。使用smk，并在product表中添加 Manufacturer 一列，将 Manufacturer 改名为 Producer，之后再删去该列。创建货车表（Truck），然后再删除货车表。本实验要求用SQL语句和图形界面两种方法实现以上各功能。注意修改表格之后，需要刷新数据库（Ctrl+F5）才能看到修改。

1. 添加或修改属性，右键 - Modify table...



2. 重命名表格，右键 - Rename...

3. 删除表格，右键 - Drop...

4. 实现以下各操作：

- 在product表中添加 生产商 这一属性

Manufacturer char(10)类型 允许null

SQL语句为

```
alter table product add Manufacturer char(10);
```

- 将product表中 Manufacturer这一属性改名为 Producer，数据类型改为char(20)。SQL语句为

```
alter table product change Manufacturer Producer char(20) null;
```

- 删去 Producer 这一属性。SQL语句为

```
alter table product drop column Producer;
```

- 创建 货车(Truck) 表格，包含以下属性

```
TruckID int类型 not null 主键,  
Driver char(10)类型 允许null,  
Area char(10)类型 允许null,  
Capacity int类型 允许null,  
Time2retire date类型 允许null,
```

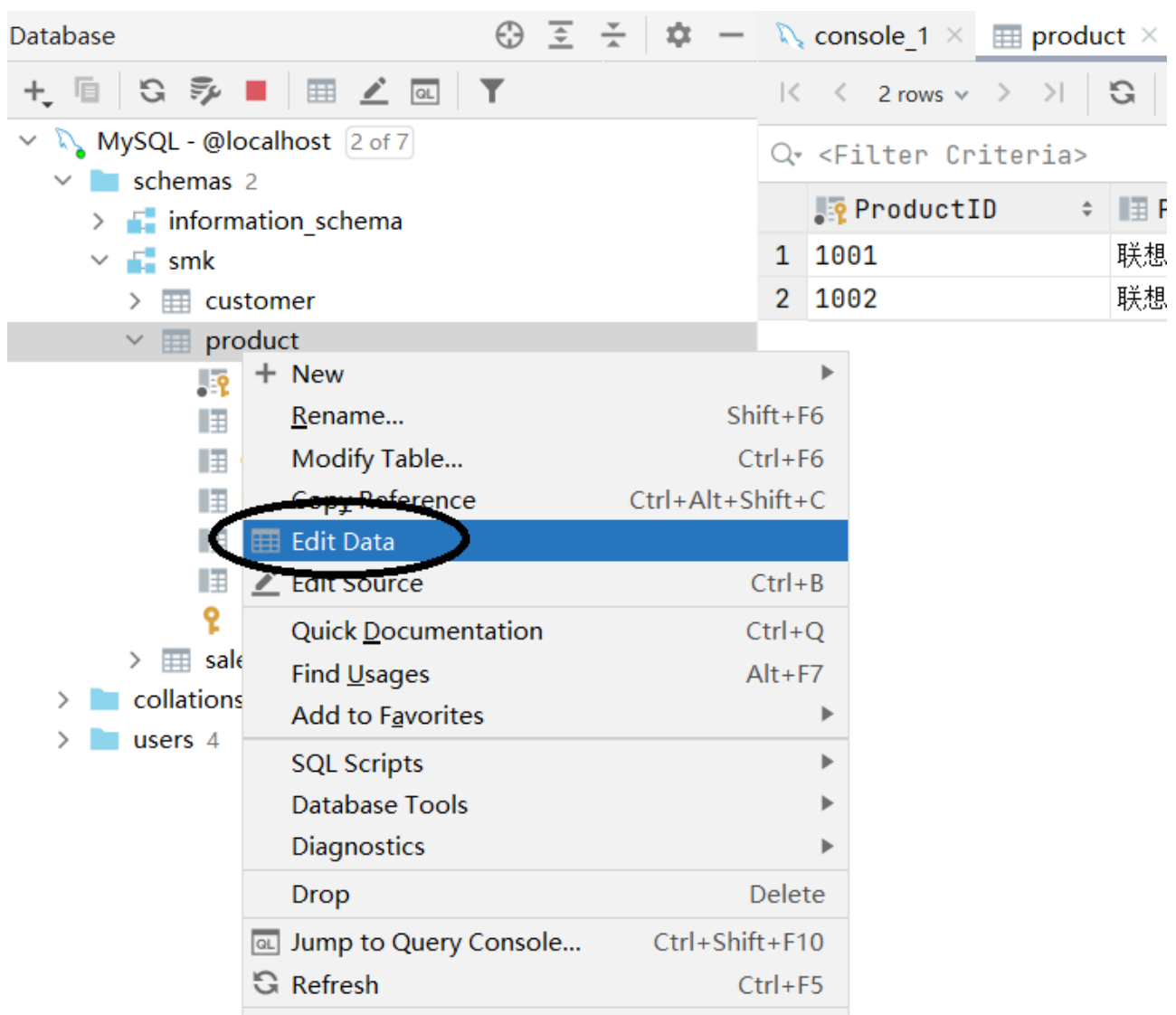
- 删除Truck表格

```
drop table truck;
```

5.6 用图形界面插入、修改、删除数据

【任务五】数据的插入。在product, customer, sale各表分别插入所需数据。

1. 在表处右键，Edit Data



2. 添加数据示例

- 在customer表中添加以下数据

LCB-PC.SMK - dbo.Customer

	顾客编号	姓名	性别	省份	城市	县
▶	2001	王华	女	北京	北京市	崇文区
	2002	李明	男	北京	北京市	海淀区
	2003	陈军	男	上海	上海市	黄浦区
	2004	许冰	男	上海	上海市	浦东区
	2005	李丽	女	湖北省	武汉市	洪山区
	2006	刘英	女	湖北省	武汉市	武昌区
*	NULL	NULL	NULL	NULL	NULL	NULL

1 / 6

- 在sale表中添加以下数据

LCB-PC.SMK - dbo.Sale

	销售编号	日期	顾客编号	商品编号	数量	金额
	1	2015-03-05	2001	1001	1	5600
	2	2015-04-17	2001	1003	1	2800
	3	2015-06-20	2002	1004	1	3800
	4	2015-07-01	2002	1005	1	4200
	5	2015-08-05	2003	1007	1	3500
	6	2015-08-10	2004	1002	1	4300
	7	2015-08-25	2005	1008	1	2600
	8	2015-09-01	2003	1007	1	3500
	9	2015-09-21	2003	1007	1	3500
	10	2015-10-01	2004	1006	1	4700
	11	2015-11-18	2005	1001	1	5600
▶	13	2015-11-20	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

12 / 12 单元格是只读的。

- 在图形界面删除和修改数据与插入数据类似

5.7 单表查询

【任务六】 单表查询。在product表中实现以下操作：

1. 查询所有商品的编号与名称。

```
select ProductID, ProductName from product;
```

2. 查询所有商品的所有属性。

```
select ProductID, ProductName, Category, SubCategory, Price, Inventory from product;
```

或者

```
select * from product;
```

3. 假设每种商品的销售量均为50件，查询每种商品还需入库多少才能满足需求。

```
select ProductID, 50-Inventory from product;
```

4. 查询所有商品类比（不合并）

```
select Category from product;
```

5. 查询所有商品类比（合并同类商品）

```
select distinct Category from product;
```

6. 查询家电类商品编号

```
select ProductID from product where Category='家电';
```

7. 试着自己思考：

- 查询所有售价在 4000 元以下的商品编号与名称；
- 查询所有库存量在38到48之间的商品编号与名称；（between...and...）
- 查询计算机和通信类的所有商品信息；（in关键词）
- 查询子类以“电”开头的所有商品信息；

```
select * from product where SubCategory like '电%';
```

- 查询子类第二个字为“机”的所有商品信息

```
select * from product where SubCategory like '_机%';
```

- 查询售价在4500以内的所有计算机类商品的编号与名称

```
select ProductID, ProductName from product where  
Category='计算机' and Price<4500;
```

- 查询所有计算机类商品的信息，并按照售价降序排序
- 查询所有家电类商品的信息，并按照库存降序排序，相同库存的按照售价升序排序

- 查询家电类商品种类数：（Count聚集函数）
- 查询所有商品总库存数：（Sum聚集函数）
- 查询各子类商品的均价与对应名称；

```
select SubCategory, avg(Price) from product group by  
SubCategory;
```

- 查询均价在3000元以上的子类商品类别，以及各子类商品的总库存量

```
select SubCategory, sum(Inventory) from product group by  
SubCategory having avg(Price)>3000;
```

- 查询总库存量在80以上的分类商品类别，以及各类商品分别的总售价