

实验二：数据定义

一、实验目的

- 理解用户、数据库、模式和基本表等数据库对象之间的关系
- 熟练掌握用户、数据库、模式、基本表和索引的创建、修改和删除方法
- 理解并熟练掌握数据类型的选择和使用方法
- 熟练掌握 sql 脚本文件的创建和使用
- 熟练掌握数据库的导入导出方法

二、实验要求

- 设计正确的 SQL 语句实现用户、数据库、模式、基本表和索引的创建、修改和删除
- 完成实验内容并撰写实验报告（按实验报告模板）
- 实验报告提交截止日期：2025 年 4 月 5 日星期六（4 号星期五为清明节，故推迟一天）

三、实验内容

1.业务背景： 现有一个销售数据库 Sales，该数据库包含 12 张表，各表的表名和信息描述见下表。

Table Names	Description	Records
CONTACTS	store contact person information of customers	319 records
COUNTRIES	store country information	25 records
CUSTOMERS	store customer master	319 records

Table Names	Description	Records
EMPLOYEES	store employee master	107 records
INVENTORIES	store inventory information of products	1112 records
LOCATIONS	store locations of warehouses	23 records
ORDERS	store order header information	105 records
ORDER_ITEMS	store order line items	665 records
PRODUCT_CATEGORIES	store product categories	5 records
PRODUCTS	store product information	288 records
REGIONS	store regions where the company operates	4 records
WAREHOUSES	store warehouse information	9 records

各表之间的关系见【[样例数据库模式.pdf 文件](#)】

图例特别说明：带星号 “*” 的字段为非空；employees 表字段 manager_id 值引用的是 employees 表字段 employee_id 的值（[引用自身表](#)）。

2.在[非 root 用户](#)下创建一个名为 sales 的数据库以及包含着该库下的 12 张表，表名见上述表格

3.在 customers 表上分别建立以下 [3 个索引](#)：name 字段上名为 idx_name 的索引、website 字段上名为 uni_idx_website 的唯一索引、name 和 address 字段上名为 comb_idx_name_addr 的组合索引

4.为 12 张表添加示例数据

5.将创建好的数据库整体导出到本地，即将结构和数据一起导出

四、实验步骤

说明：

(1)为提升编辑效率，减少重复输入，[建议先在文本编辑器](#)（如记事本）里[写好 SQL 语句](#)（编码设置为 UTF-8），然后复制+粘贴相应的 SQL 语句到命令行客户端或 IDE 如 navicate premium lite 中执行

(2)如果想批量执行一组 SQL 命令，可将这些命令编辑成一个 **SQL 脚本文件**（默认扩展名为.sql）

(3)**SQL 脚本文件**是一个文本文件，只包含注释语句和 sql 命令。mysql 的注释符包括单行注释和多行注释

- 单行注释：-- 空格 说明，例：-- select * from contacts; （若无空格则报错）
- 单行注释：#，例：#select count(*) from contacts; （#后的空格可有可无）
- 多行注释：/**/

示例：

```
-- select count(*) from contacts;
/*select count(*) from countries;
select count(*) from customers;*/
#select count(*) from employees;
```

(4)SQL 脚本的执行命令：source sql 文件的完整路径（注意：不需要分号结束）

```
mysql> source D:\sample.sql
Query OK, 0 rows affected (0.00 sec)

+-----+
| count(*) |
+-----+
|      107 |
+-----+
1 row in set (0.00 sec)
```

实验步骤如下：

1.以 root 用户身份创建一个新用户并授权 DBA 权限给该用户，用户名为自己中文拼音（具体名自定）

- 创建用户命令：create user user_name identified by 'new_password';
- 授权命令：grant all on *.* to user_name;
- 注：如已完成上述工作则跳过此步

2.以新用户登录 mysql 服务器

3. 创建一个名为 sales 的数据库以及包含着该库下的 12 张表 (contacts, ..., warehouses), 初次建表时不添加参照完整性约束 (为避免违反完整性约束带来的插入数据失败), 但主码和非空等约束不受影响, 可以先定义

- **注意:** number, number(d,p)和 varchar2 不是 mysql 支持的数据类型, 将其变更为 smallint, decimal(d,p)和 varchar (参见【refman-8.4-en.pdf】之 chapter 13 data types)

4. 使用 desc tble_name 命令分别显示 12 张表的结构 (可以先在文本文件里写然后复制粘贴到命令行客户端或 IDE 的 SQL 查询窗口, 也可以写成一个脚本文件然后执行它)

5. 在 customers 表上分别建立以下 3 个索引: name 字段上名为 idx_name 的索引、website 字段上名为 uni_idx_website 的唯一索引、name 和 address 字段上名为 comb_idx_name_addr 的组合索引。

6. 使用命令: show index from customers; 显示 customers 表上的所有索引, 注意观察 Index_type 字段的值, 该字段值为索引算法

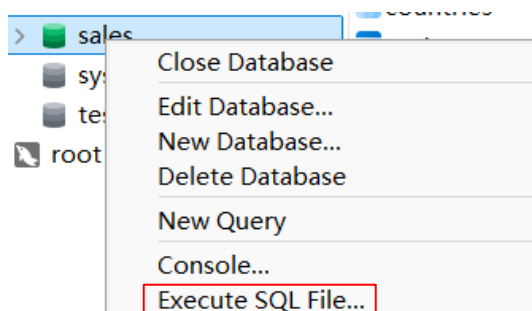
7. 为 12 张表插入示例数据, 三种常用的数据插入方式

- 方式一: insert into...values 命令 #mysql8.4 支持一条 insert 命令一次性插入多条数据
- 方式二: 写插入数据的脚本, 然后执行该脚本
- 方式三: 使用工具如 navicate premium lite (以下简称 navicate) 等进行数据导入

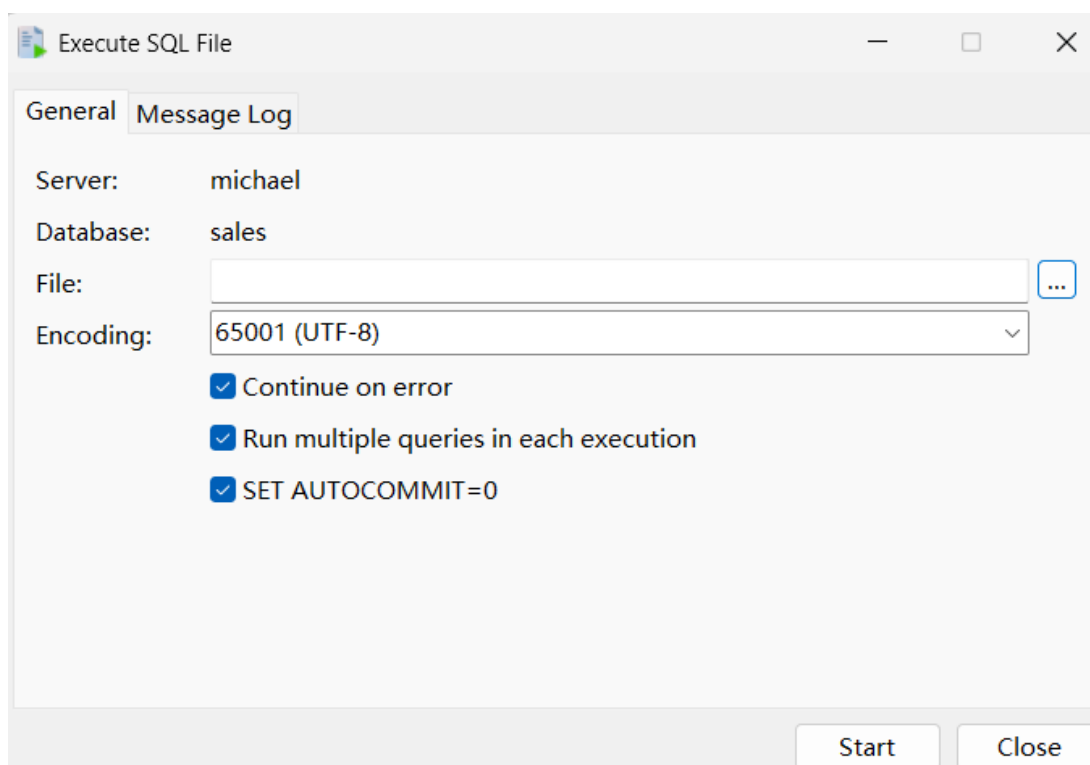
要求: 分别使用三种方式完成数据的插入操作, 顺序不限。当一种方式执行成功后切换到另外一种方式前需要先删除所有 12 张表中的数据

7.1 navcate 在数据库层面的导入导出:

- **导入:** 连接 sales 数据库>右键 sales>选中 Execute SQL file

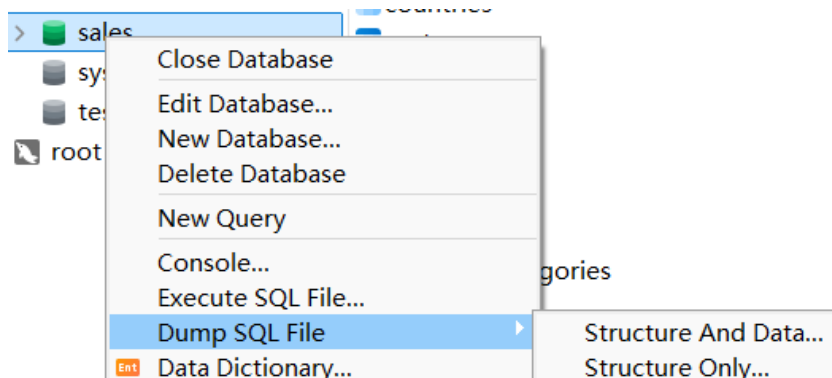


本质上是执行插入数据的脚本文件



找到并打开 SQL file>点击 start 开始执行

- 导出: 连接 sales 数据库>右键 sales>选中 Dump SQL file>选择 structure and data

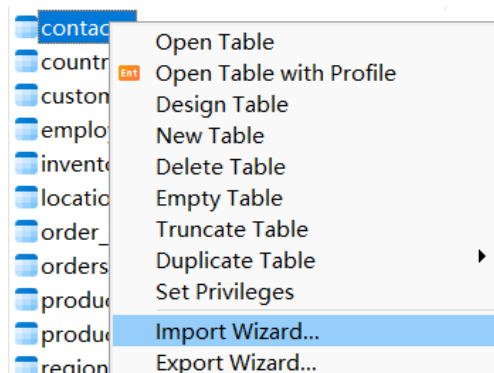


structure and data...表示导出整个数据库里的所有表结构及其数据

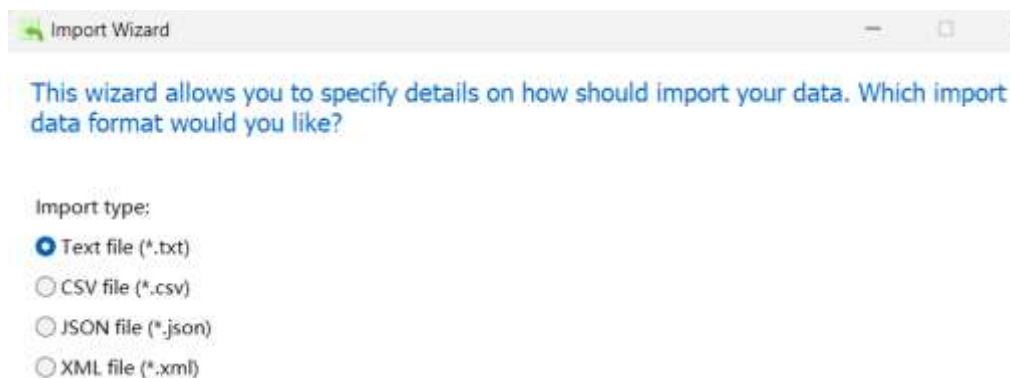
structure only...表示只导出整个数据库里的所有表到结构但不包含数据

7.2 navicate 在表级上的导入导出

- **导入：**选中并右键一张表进行 import wizard...或 export wizard...操作



navicate 在表上进行导入导出时支持 txt,csv,json,xml 等数据文件格式



问题：上述两种方式的差别是什么？

8.依次使用查询语句 “**SELECT * FROM 表名;**” 查询所有 12 张表的数据

9.依次使用查询语句 “**SELECT COUNT(*) FROM 表名**” 查询每张表的记录数

10.修改 12 张表的结构，**增加所有表的外码**（如果存在）

11.将 customers 表上的索引 idx_name 改名为 index_name

mysql8.4 索引改名命令：

```
ALTER TABLE tbl_name RENAME INDEX old_idx_name TO new_idx_name;
```

注意：mysql 的语法与教材不同

12.使用 show index 命令查看索引改名是否成功

13.备份整个数据库，包括库中所有表的结构及数据到本地，即导出整个数据库到本地，以备删除实验完成后重建 sales 数据库

14.验证删除操作效果。依次执行以下命令：

- drop table customers; -- 观察结果
- drop table customers cascade; #观察结果，若逐个删除 12 张表则需要注意删除顺序
- drop schema sales; -- 观察结果
- drop database sales; -- 观察结果
- drop user user_name; -- 观察结果

15.重建 sales 库及其中的 12 张表和示例数据（使用第 13 步保存的脚本）以备后续实验使用

五、实验思考

- 1.mysql8.4 中用户、数据库、模式、基本表和索引之间的关系是什么？
- 2.什么是 SQL 脚本文件？有什么作用？脚本文件包含哪些内容？如何创建、修改和删除脚本文件？
- 3.drop 命令中 cascade 的作用是什么？
- 4.实现数据库中数据导入导出的常用方法有哪些？

注意：这 4 个问题的解答放到实验报告的实验总结部分