

课程介绍

学习目标



数据库事务机制

undo和redo日志、开启事务、提交事务、回滚事务



数据导出与导入

SQL文件的导出与导入，TXT文档的导入与导出



综合案例：设计数据表

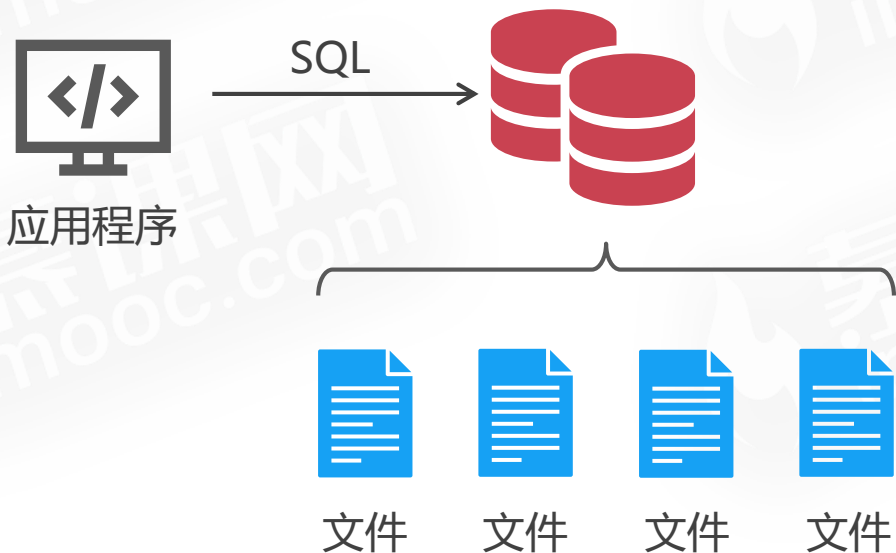
设计新闻管理系统的数据表



事务机制（一）

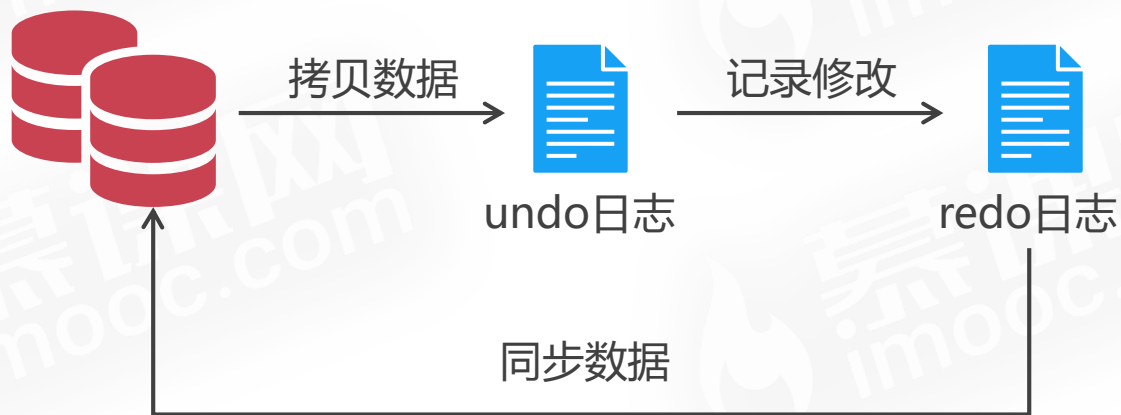
避免写入直接操作数据文件

- ◆ 如果数据的写入直接操作数据文件是非常危险的事情



利用日志来实现间接写入

◆ MySQL总共有5种日志，其中只有redo日志和undo日志与事务有关

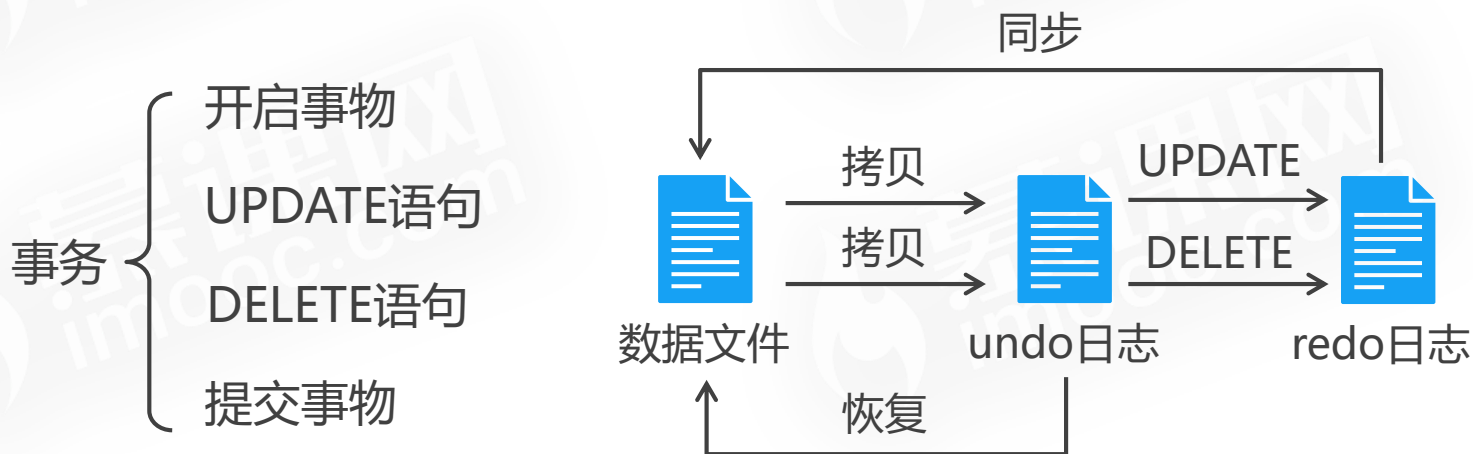


事务机制 (Transaction)

- ◆ RDBMS = SQL语句 + 事务 (ACID)
- ◆ 事务是一个或者多个SQL语句组成的整体，要么全部执行成功，要么全都执行失败

事务案例

- ◆ 把10部门中MANGER员工调往20部门，其他岗位的员工调往30部门，然后删除10部门



管理事务

- ◆ 默认情况下，MySQL执行每条SQL语句都会自动开启和提交事务
- ◆ 为了让多条SQL语句纳入到一个事务之下，可以手动管理事务

```
START TRANSACTION ;
```

```
SQL语句
```

```
[ COMMIT | ROLLBACK ] ;
```


事务的ACID属性



事务的原子性

- ◆ 一个事务中的所有操作要么全部完成，要么全部失败。事务执行后，不允许停留在中间某个状态



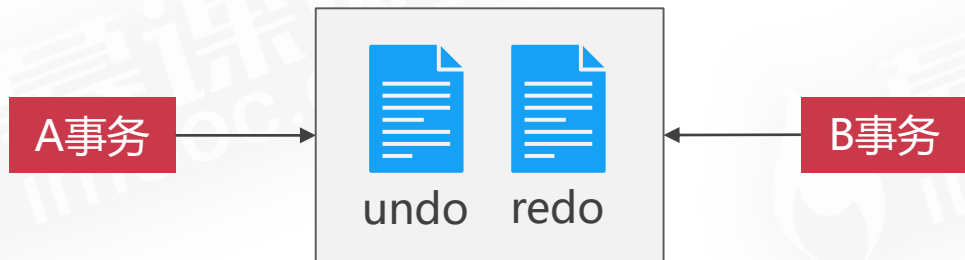
事务的一致性

- ◆ 不管在任何给定的时间、并发事务有多少，事务必须保证运行结果的一致性
- ◆ 事务可以并发执行，但是最终MySQL却串行执行



隔离性

- ◆ 隔离性要求事务不受其他并发事务的影响，如同在给定的时间内，该事务是数据库唯一运行的事物
- ◆ 默认情况下A事务，只能看到日志中该事务的相关数据



持久性

- ◆ 事务一旦提交，结果便是永久性的。即便发生宕机，仍然可以依靠事务日志完成数据的持久化



事务机制（二）

事务的四个隔离级别

序号	隔离级别	功能
1	read uncommitted	读取未提交数据
2	read committed	读取已提交数据
3	repeatable read	重复读取
4	serializable	序列化

业务案例1

A事务	车次	车厢	坐席	状态
	G8047	1	1A	未售出
B事务	G8047	1	1B	未售出

A事务	车次	车厢	坐席	状态
	G8047	1	1A	已售出
	G8047	1	1B	未售出

修改事务隔离级别

- ◆ READ UNCOMMITTED 代表可以读取其他事务未提交的数据

```
SET SESSION TRANSACTION ISOLATION LEVEL  
READ UNCOMMITTED;
```



业务案例2



修改事务隔离级别

- ◆ READ COMMITTED 代表只能读取其他事务提交的数据

```
SET SESSION TRANSACTION ISOLATION LEVEL  
READ COMMITTED;
```



业务案例3



修改事务隔离级别

- ◆ REPEATABLE READ 代表事务在执行中反复读取数据，得到的结果是一致的，不会受其他事务影响

```
SET SESSION TRANSACTION ISOLATION LEVEL  
REPEATABLE READ;
```

事务的序列化

- ◆ 由于事务并发执行所带来的各种问题，前三种隔离级别只适用在某些业务场景中，但是序列化的隔离性，让事务逐一执行，就不会产生上述问题了。

```
SET SESSION TRANSACTION ISOLATION LEVEL  
SERIALIZABLE;
```

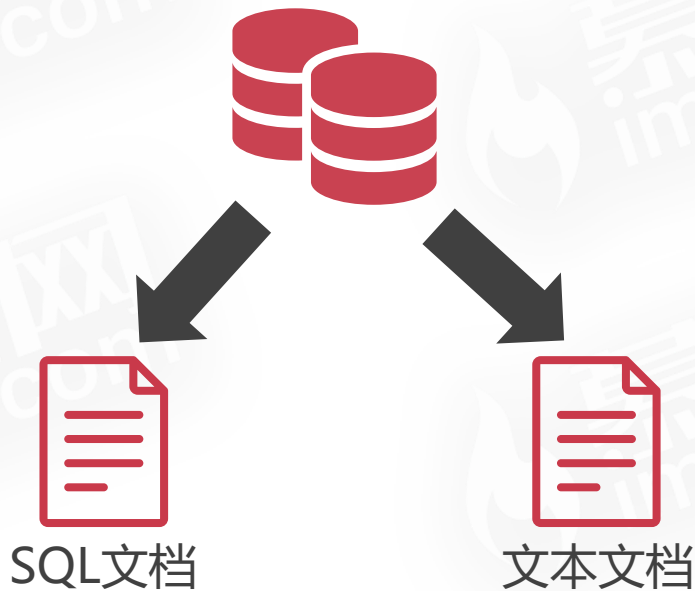
数据导入与导出

数据导出与备份的区别

- ◆ 数据导出，导出的纯粹是业务数据
- ◆ 数据备份，备份的是数据文件、日志文件、索引文件等等



数据导出的分类



导出SQL文件

- ◆ mysqldump用来把业务数据导出成SQL文件，其中也包括了表结构

```
mysqldump -uroot -p [ no-data ] 逻辑库 > 路径
```

- ◆ 如果业务数据非常多，建议只导出表结构，然后用SELECT INTO OUTFILE把数据导出成文本文档

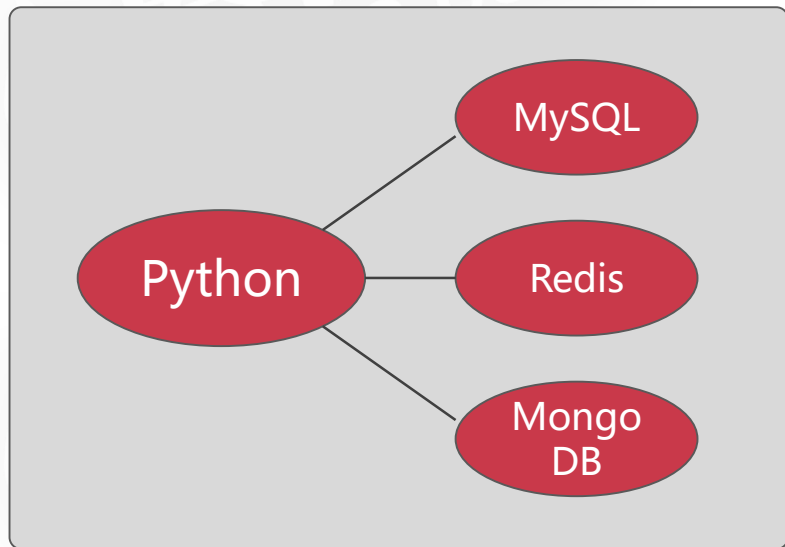
导入SQL文件

- ◆ source命令用于导入SQL文件，包括创建数据表，写入记录等

```
sql > USE demo ;  
sql > SOURCE backup.sql ;
```

《新闻管理系统》数据库设计

本门课程的项目案例



项目案例

新闻管理系统

《新闻管理系统》控制台程序

输入用户名：admin

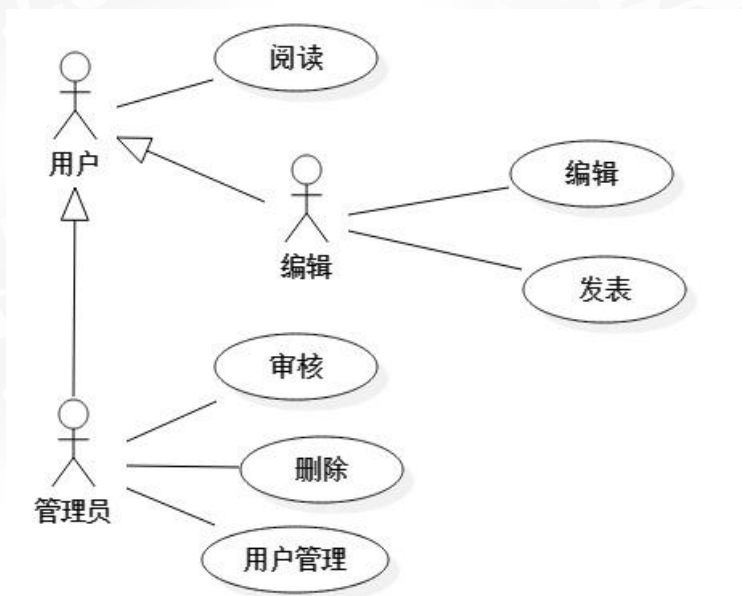
密码：*****

新闻管理系统

选择你要执行的操作：

1. 新闻列表
2. 新建新闻
3. 编辑新闻
4. 退出

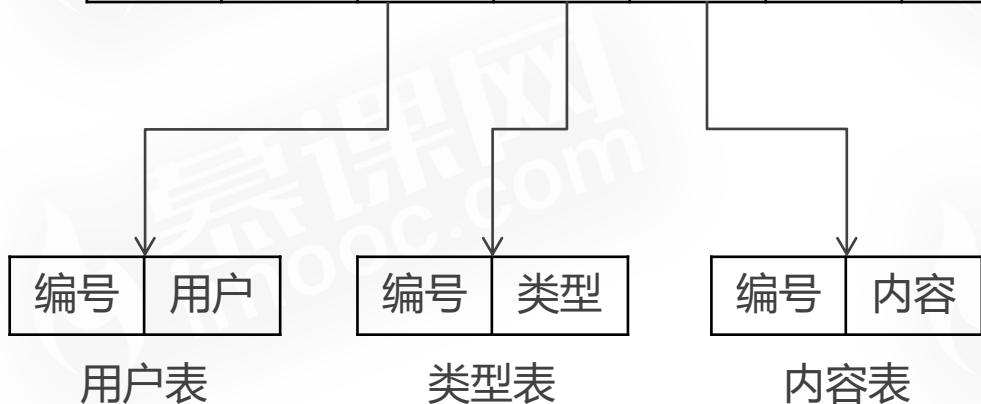
《新闻管理系统》用例图



新闻有哪些属性?

新闻表

编号	标题	作者	类型	内容	置顶	创建时间	修改时间	状态
----	----	----	----	----	----	------	------	----



创建逻辑库和数据表（一）

数据库ER图

t_role

- id: int UNSIGNED
- role: varchar(20)

t_type

- id: int UNSIGNED
- type: varchar(20)

t_user

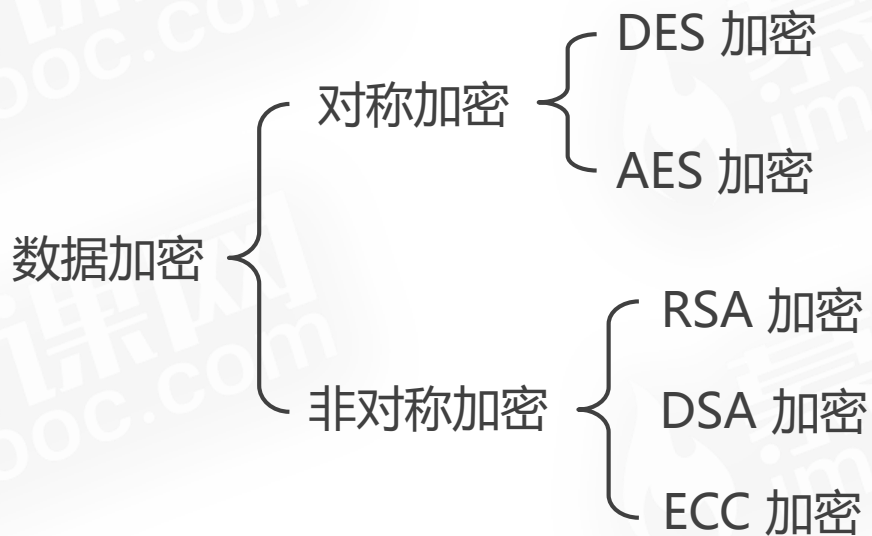
- id: int UNSIGNED
- username: varchar(20)
- password: varchar(500)
- email: varchar(100)
- role_id: int UNSIGNED

t_news

- id: int UNSIGNED
- title: varchar(40)
- editor_id: int UNSIGNED
- type_id: int UNSIGNED
- content_id: char(12)
- is_top: tinyint UNSIGNED
- create_time: timestamp(0)
- update_time: timestamp(0)
- state: enum

创建逻辑库和数据表（二）

数据加密



AES加密函数

- ◆ MySQL数据库提供了AES加密和解密的函数，所以数据的加密解密非常容易实现

AES_ENCRYPT(原始数据 , 密钥字符串)

```
SELECT AES_ENCRYPT("你好世界", "ABC123456");  
SELECT HEX(AES_ENCRYPT("你好世界", "ABC123456"));
```

AES解密函数

- ◆ AES解密要使用与加密相同的密钥，才能解密出原始数据

AES_DECRYPT(加密结果 , 密钥字符串)

```
SELECT  
  AES_DECRYPT (  
    UNHEX ( "E85A104B6142A7375E53C0545CAD48EE" ) ,  
    "ABC123456"  
  ) ;
```

创建逻辑库和数据表（三）

课程总结

技能清单

技能知识1

☐

技能知识2

☐

技能知识3

☐

技能知识4

☐

学习了事务机制的原理和特性



掌握了事务的管理



掌握了数据导出与导入的操作



完成新闻管理系统数据表的设计