

---

# 分布式电影订票系统 系统设计文档

分布式电影订票系统设计小组

成员：徐兰天

王文琪

李佳钰

于明飞

---

1 引言.....	3
1.1 编写目的.....	3
1.2 预期读者和阅读建议.....	3
2 数据库设计命名规范.....	3
2.1 数据库涉及字符规范.....	3
2.2 数据库对象命名规范.....	3
3 数据库实体关系设计.....	4
4 数据库逻辑设计.....	4
4.1 水平分片和读写分离.....	4
4.2 主从复制.....	6
5 数据库物理设计.....	6
5.1 初始数据大小.....	6
5.2 数据库增长速度.....	7
5.3 记录查找的类型和平均数量.....	7
5.4 网络和共享访问需求.....	7
5.5 性能 .....	7
5.6 安全性.....	7
5.7 数据备份和恢复.....	7
6 数据库基本表设计.....	8
6.1 用户表.....	8
6.2 电影表.....	8
6.3 销售表.....	8
7.授权设计.....	9
7.1 电影院售票系统用户组表.....	9
7.2 对象操作权限说明.....	9

---

# 1 引言

引言是数据库设计报告的概览，是为了帮助读者了解这份文档是如何编写的，并且如何阅读、理解和解释这份文档。

## 1.1 编写目的

编写数据库设计报告的目的包括：

- (1) 为数据库实施工程师提供设计文档；
- (2) 为系统架构师提供数据设计依据；
- (3) 为软件详细设计提供数据结构设计依据；
- (4) 测试工程师测试依据。

## 1.2 预期读者和阅读建议

本设计文档预期读者包括：项目经理，关注数据库设计，确定其复杂度，估算工作量和编写人力资源计划。详细设计人员关注数据结构与事务关系，制订详细设计计划，确定设计实现技术路线。测试经理，分析数据库设计可测试性，制订测试计划。质量保证相关人员，编写质量保证计划，分析并列出质量评审要点。需求经理，对比需求分析报告，编制需求跟踪矩阵表。

# 2 数据库设计命名规范

## 2.1 数据库涉及字符规范

采用 26 个英文字母和 0-9 这十个自然数，加上下划线-组成，共 36 个字符。不出现其他字符。鉴于大小写规则难记忆，而且有些情况下，数据库不区分大小写，同一采用小写字母。

数据表名称遵循数据字典中定义的数据集定义英文标识。

数据字段名称遵循数据字典数据项英文标识。

## 2.2 数据库对象命名规范

数据库对象包括表、视图（查询）、存储过程（参数查询）、函数、约束。对象名字由前缀和实际名字组成。

---

前缀：使用大写字母

表：T\_表名称

视图：V\_视图名称

存储过程：S\_存储过程名称

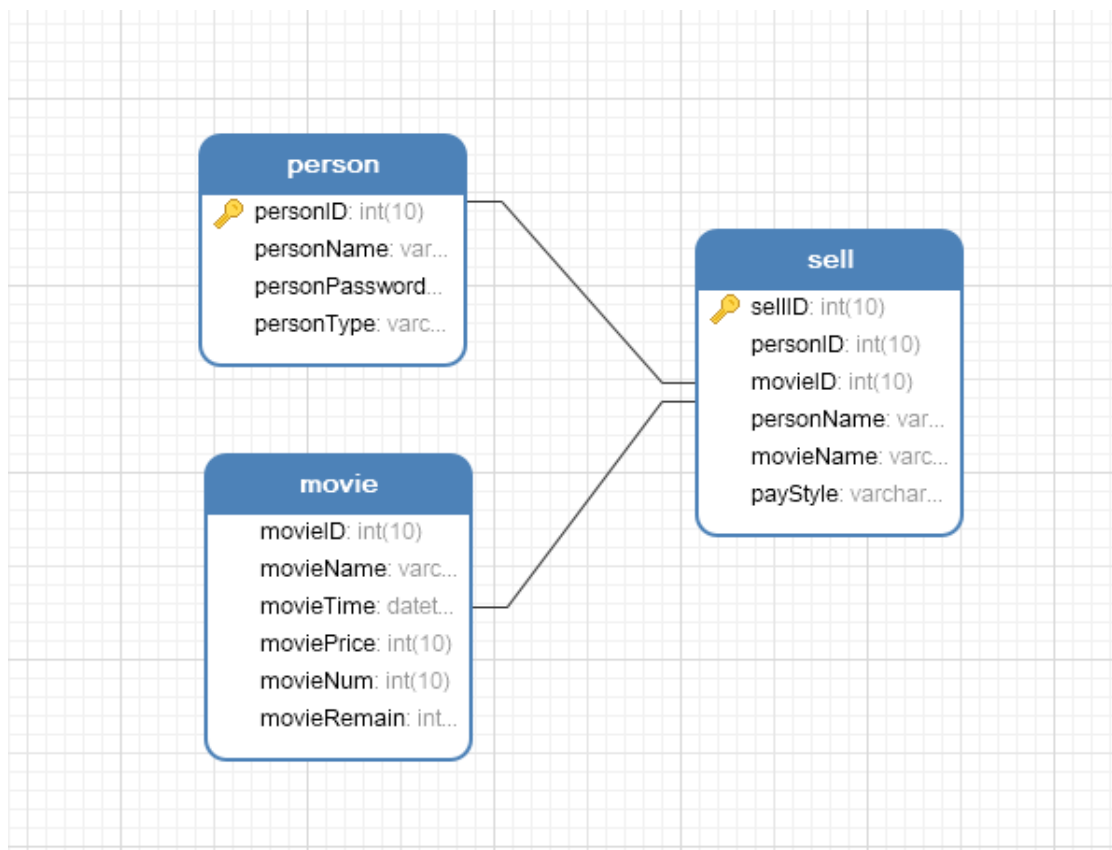
函数：FN\_函数名称

索引     IDX\_<表名>\_<索引标识>

主键     PK\_<表名>\_<主键标识>

外键     FK\_<表名>\_<主表名>\_<外键标识>

### 3 数据库实体关系设计



### 4 数据库逻辑设计

#### 4.1 水平分片和读写分离

MyCat 是一个开源的分布式数据库系统，前端用户可以把它看作是一个数据库代理，用 MySQL 客户端工具和命令行访问，而其后端可以用 MySQL 原生协议与多个 MySQL 服务器

通信，核心功能是分表分库。下图是 MyCat 的配置文件，定义了用户名，密码和端口号。

```
schema.xml  server.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE mycat:server SYSTEM "server.dtd">
3  <mycat:server xmlns:mycat="http://io.mycat/">
4    <system>
5      <property name="useSqlStat">0</property>
6      <property name="useGlobleTableCheck">0</property>
7      <property name="sequenceHandlerType">2</property>
8      <property name="processorBufferPoolType">0</property>
9      <property name="handleDistributedTransactions">0</property>
10     <property name="serverPort">8066</property>
11     <property name="useOffHeapForMerge">1</property>
12     <property name="memoryPageSize">1m</property>
13     <property name="spillsFileBufferSize">1k</property>
14     <property name="useStreamOutput">0</property>
15     <property name="systemReserveMemorySize">384m</property>
16     <property name="useZKSwitch">true</property>
17   </system>
18   <user name="root">
19     <property name="password">mysql</property>
20     <property name="schemas">mymovie</property>
21   </user>
22   <user name="user">
23     <property name="password">user</property>
24     <property name="schemas">mymovie</property>
25     <property name="readOnly">true</property>
26   </user>
27 </mycat:server>
```

如下图所示，本系统用 MyCat 将 sell 表以支付方式 0 和 1 分为两片，0 代表微信，1 代表支付宝，分别对应逻辑分片 dn1 和 dn2，分片规则为 sharding-by-intfile。dn1 对应数据库 mymovie，dn2 对应数据库 mymovie2。并设定 writeHost 为主机 A(localhost:3306)，设定 readHost 为从机 B(10.208.7.163)。主机 A 和从机 B 中各有两个数据库 mymovie 和 mymovie2。系统的所有读操作从从机 B 读，所有写操作向主机 A 写。写 sell 表的记录的 payStyle 属性为 0 时，写入主机 A 的 mymovie 库，payStyle 属性为 1 时，写入主机 A 的 mymovie2 库。

```
schema.xml
2  <!DOCTYPE mycat:schema SYSTEM "schema.dtd">
3  <mycat:schema xmlns:mycat="http://io.mycat/">
4    <schema name="mymovie" checkSQLSchema="true" sqlMaxLimit="100">
5      <table name="sell" primaryKey="payStyle" autoIncrement="false" dataNode="dn1,dn2"
6        rule="sharding-by-intfile" />
7      <table name="person" primaryKey="personID" autoIncrement="false" dataNode="fn1"/>
8      <table name="movie" primaryKey="movieID" autoIncrement="false" dataNode="fn2"/>
9    </schema>
10    <dataNode name="dn1" dataHost="localhost1" database="mymovie" />
11    <dataNode name="dn2" dataHost="localhost1" database="mymovie2" />
12    <dataNode name="fn1" dataHost="localhost2" database="mymovie" />
13    <dataNode name="fn2" dataHost="localhost2" database="mymovie" />
14    <dataHost name="localhost1" maxCon="1000" minCon="10" balance="1"
15      writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">
16      <heartbeat>select user()</heartbeat>
17      <!-- can have multi write hosts -->
18      <writeHost host="hostM1" url="localhost:3306" user="root"
19        password="mysql">
20      <!-- can have multi read hosts -->
21      <readHost host="hostS2" url="10.108.7.163:3306" user="root" password="20154418" />
22    </dataHost>
23    <dataHost name="localhost2" maxCon="1000" minCon="10" balance="1"
24      writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThreshold="100">
25      <heartbeat>select user()</heartbeat>
26      <writeHost host="hostM1" url="localhost:3306" user="root"
27        password="mysql">
28      <readHost host="hostM1" url="localhost:3306" user="root"
29        password="mysql" />
30    </dataHost>
31  </mycat:schema>
```

## 4.2 主从复制

仅仅完成以上操作是不够的。如果没有主从复制，主机 A 和从机 B 就会发生数据不一致。本系统用 MySQL 实现主从复制。其主要原理是从机利用主机的日志信息，执行与主机同样的操作。下图为查看主机 A 状态 File 和 Position，File 是同步给从库的数据库文件，到时候授权给从库时要用。

```
D:\MySQL\mysql-8.0.18-winx64\bin>mysql -uroot -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show master status;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000015 |      155 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

启动从服务的 slave 复制功能，查从服务器的 slave 状态，Slave\_IO\_Running 和 Slave\_SQL\_Running 的值都为 Yes，则说明主从复制的所有配置已成功，即从服务器已经可以自动与主服务器的数据库数据实现同步了。文件夹中录屏文件有测试效果演示。

```
mysql> change master to master_host='10.108.6.180',master_user='root',master_password='mysql',master_port=3306,master_log_file='mysql-bin.000015',master_log_pos=155;
Query OK, 0 rows affected, 1 warning (0.04 sec)

mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 10.108.6.180
Master_User: root
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000015
Read_Master_Log_Pos: 155
Relay_Log_File: DESKTOP-2TCP00A-relay-bin.000002
Relay_Log_Pos: 322
Relay_Master_Log_File: mysql-bin.000015
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: mymovie,mymovie2
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 155
Relay_Log_Space: 540
Until_Condition: None
```

## 5 数据库物理设计

### 5.1 初始数据大小

大约有几十条数据导入到新数据库中

---

大约有几十名员工对该数据库进行操作。最高峰时间为早上九点到十二点左右。

大约有几十名左右员工注册为系统用户。

## 5.2 数据库增长速度

因为电影院管理系统本质是录入和计算数据，所以数据库数据变动较大。

员工离开公司职务，那么就将该员工信息从数据库中删除，但做为备份数据存在，以便日后追查。

## 5.3 记录查找的类型和平均数量

产品信息查询大约 1 天 100 次。

数据录入操作大约 1 天 5 次。

数据计算操作大约 1 天 5 次。

## 5.4 网络和共享访问需求

所有员工必须安全地连接到数据库。系统必须支持每个员工的安全流程访问，以便快速完成业务。

## 5.5 性能

本数据库对于数据的改动较大。为此一般要求查询响应时间不能超过 5 秒钟。数据保存时间不能超过 5 秒钟。

## 5.6 安全性

数据库访问必须要有口令保护。

将系统用户按照不同的组别进行分类，分别赋予不同的数据库访问权限。

为了保证内部数据字段的安全性，多表查询尽量采取视图的方式进行检索。

将用户按照不同类别进行组织，按照用户视图格式对每个用户进行归类，然后按照用户视图列定的权限赋值相应权限。

## 5.7 数据备份和恢复

数据库备份在每天晚上 12 点。

## 6 数据库基本表设计

### 6.1 用户表

名称	用户表			代码		person		
数据表字段属性								
名称	字段	数据类型	是否为空	主键	外键	长度	值范围	域名
账号	personID	int	否	1		10		
用户名	personName	varchar				20		
密码	personPassword	varchar				20		
用户类型	personType	varchar				20		

### 6.2 电影表

名称	电影表				代码	Movie		
数据表字段属性								
名称	字段	数据类型	是否 为 空	主 键	外 键	长度	值范 围	域名
电影编号	movieID	Int	否	1		10		
电影名称	movieName	Varchar				20		
时间	movieTime	Datetime				0		
价格	moviePrice	Int				10		
数量	movieNum	Int				10		
余票	movieRemain	int				10		

### 6.3 销售表

名称	销售表				代码	Sell		
数据表字段属性								
名称	字段	数据类型	是否为空	主键	外键	长度	值范围	域名
销售编号	sellID	Int	否	1		10		



用户编号	personID	Int			1	10		
电影编号	movieID	Int			1	10		
用户名	personName	Varchar				20		
电影名称	movieName	Varchar				20		
支付类型	payStyle	varchar				20		

## 7.授权设计

### 7.1 电影院售票系统用户组表

用户组名	代码	用户级别	说明
数据库管理员	DatabaseAdmin	超级用户	数据库操作的所有权限
用户管理员	UserAdmin	数据库用户管理	对数据库用户授权等
系统管理员	SystemAdmin	系统管理最高级	业务系统的完全操作 权，包括对系统用户的 系统权限分配等
电影院管理员	Manager	影院管理	电影表、销售表的增删 改查
购票用户	Consumer	电影票购买	销售表查看

### 7.2 对象操作权限说明

权限名称	缩写	描述
DELETE	D	允许用户从对象删除记录
INSERT	I	允许用户借助 INSERT 或 IMPORT 命令将记录插入对象
SELECT	S	提供使用 select 语句查看对象内容的能力
UPDATE	U	允许用户使用 update 语句修改对象内的记录
ALTER	A	允许用户使用 alter 语句修改对象定义
INDEX	IN	允许用户使用 create index 语句在对象上创建索引
ALL	AL	全公司业务数据权限
Department	DE	部门范围数据权限