

**课 程 实 验 报 告**

**课程名称：**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期：**

**计算机科学与技术学院**

**目 录**

[1 课程实验概述 1](#_Toc676213769)

[2 实验一 基本SQL操作 2](#_Toc1508977521)

[2.1 实验目的 2](#_Toc1635603449)

[2.2 实验内容 2](#_Toc908810843)

[2.3 实验环境 3](#_Toc2071677197)

[2.4 实验步骤 3](#_Toc737059085)

[2.5 经验总结 3](#_Toc775927846)

# 1 课程实验概述

# 2 实验一 基本SQL操作

## 2.1 实验目的

1. 熟悉一种DBSM软件（sqlserver，mysql等）的使用，并且能够正确搭建其操作环境。
2. 掌握SQL语言的基本操作，包括表的创建、删除、修改以及查询等。

## 2.2 实验内容

1）数据定义

参照下面的内容建立自己实验所需的关系数据

创建三个关系:

商品表【商品名称、商品类型】

GOODS【GNAME char(20),GTYPE char(10)】

主关键字为(商品名称)。商品类型为(电器、文具、服装等)

商场【商场名称,所在地区】

PLAZA【PNAME char(20)，PAREAchar(20)】

主关键字为商场名称。所在地区为(洪山、汉口、汉阳、武昌等)

销售价格表【商品名称、商场名称、当前销售价格、目前举办活动类型】

SALE【GNAME char(20),PNAME char(20),PRICE FLOAT,ATYPEchar(10)】

主关键字为(商品名称、商场名称)。举办活动类型为(送券、打折),也可为空值,表示当前未举办任何活动。表中记录如(‘哈森皮靴’,‘亚贸广场’,200,‘打折’),同一商场针对不同的商品可能采取不同的促销活动。

2）数据更新

（1）向上述表格中用 sql 语句完成增、删、个、改的操作;

（2）编写一个触发器,并测试该触发器;

（3）将 SALE 表中的打折记录插入到新表 SALE\_CHEAP 中,并基于 SALE\_CHEAP 表创建一个统计每个商场各自打折商品平均价格的视图。

3）用 SQL 语句完成下述查询需求:

（1）查询所有以“打折”方式销售的商品的当前销售情况,并按照价格的降序排列;

（2）查询所有没有任何活动的商品及其所在的商场,结果按照商品排序;

（3）查询价格在 200~500 元之间的商品名称、所在的商场名称、价格,结果按照商场名称排序;

（4）查询每种商品的最低价格、商品名称;

（5）查询以“送券”方式销售的商品总数超过 2 种的商场名称;

（6）查询以“送券”方式销售的商品总数超过 2 种的商场所在地区;

（7）查询价格为下列取值之一的商品名称、所在商场名称、目前举办活动的类型(88、188、288);

（8）查询以“老”字开头的所有商品的名称;

（9）查询同时销售“剃须刀”和“电池”的商场名称;

（10）查询不举办任何活动的商场;

（11）查询所销售的商品包含了“校园超市”所销售的所有商品的商场名称。

4）SQL 思考题部分(选做)

（1）SQLSERVER 的时间数据类型 datatime,可以记录形如“2010-5-1 13:30:15”的时刻,表示2010 年 5 月 1 日 13 点 30 分 15 秒时刻,通过查询 SQLSERVER 的帮助文件,解决下述 SQL语句的需求:

（1.1）在查询结果中将 datetime 属性的值转换成字符串类型;

（1.2）查询当前时刻之前 2 年、或者 2 个月、或者 2 天、或者 2 小时、或者 2 分钟的记录;

（1.3）查询某个时刻所在的年、月、日、时、分、秒;

（1.4）查询某个月的最后一天,返回 datetime 类型;

（1.5）假设在 SALE 表中有交易时间(DEELTM)属性,类型为 datetime,查出所有在整点时刻的交易,查出所有在整 15 分钟时刻(每个小时有 4 个整 15 分钟,例如 8:00、8:15、8:30、8:45)的交易。

（2）如何将查询结果的某些列的某些值转换成特殊的形式(例如成绩记录存储的都是百分制的成绩,对查询结果中 60 分以上的记录直接显示成绩,而对 60 分以下的记录则在成绩属性中显示“不及格”)?

## 2.3 实验环境

操作系统：Ubuntu16.04

DBSM软件：mysql 5.7.18

## 2.4 实验过程

### 2.4.1 数据定义

1. 创建一个MyGoods的数据库，然后使用create table命令创建三张表（商品表、商场表和销售价格表）。此部分代码如下：

/\*建表\*/

Create table GOODS(

Gname char(20) primary key,/\*商品名称\*/

Gtype char(10)/\*商品类别\*/

)COMMENT='商品表';

create table PLAZA(

Pname char(20) primary key,/\*商场名称\*/

Parea char(20)/\*商场所在区域\*/

)COMMENT='商场表';

create table SALE(

Gname char(20),

Pname char(20),

Price float,/\*价格\*/

Atype char(10),

primary key (Gname,Pname),/\*设定主码\*/

foreign key (Gname) references GOODS(Gname),

foreign key (Pname) references PLAZA(Pname)

)COMMENT='销售价格表';

（2）使用“insert into 表名 values(...);”语句完成数据的导入。导入完成后，GOODS表中的数据如图2. 1所示;PLAZA表中数据如图2. 3所示。接下来，使用这些数据完成指定的需求。



图2. 1 GOODS表

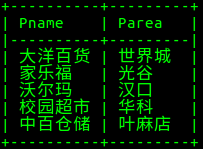


图2. 2 PLAZA表

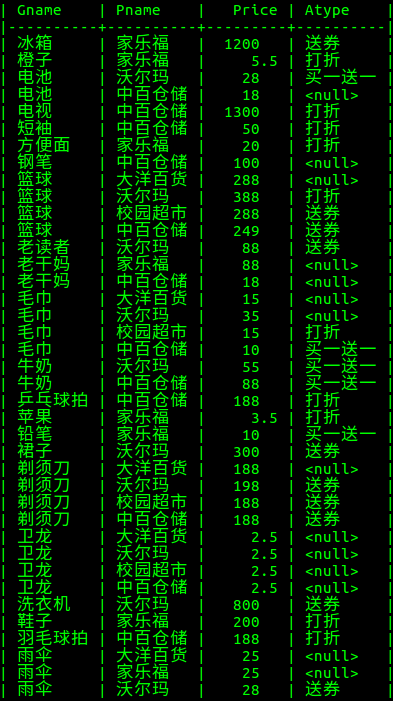


图2. 3 SALE表

### 2.4.2 查询功能

（1）实验内容2.3部分。

功能需求：将 SALE 表中的打折记录插入到新表 SALE\_CHEAP 中,并基于 SALE\_CHEAP表创建一个统计每个商场各自打折商品平均价格的视图。

解决需求：首先，用create table语句创建一个SALE\_CHEAP表，该表的数据结构与表SALE一致。然后使用select语句将SALE表中打折的商品插入SALE\_CHEAP表中。实验代码如下所示。

/\*往打折表中插入数据\*/

INSERT INTO SALE\_CHEAP(GNAME,PNAME,PRICE,ATYPE)

SELECT GNAME,PNAME,PRICE,ATYPE

FROM SALE

WHERE ATYPE='打折';

执行结果生成的SALE\_CHEAP表如图2. 4所示。



图2. 4 SALE\_CHEAP表

接着，基于SALE\_CHEAP表，使用create view语句创建一个平均价格的视图，此部分实验代码如下，生成的视图结果见图2. 5

create view AVE\_PRICE /\*创建视图AVE\_PRICE\*/

AS SELECT Pname,AVG(Price) /\*计算平均价格\*/

from SALE\_CHEAP

group by Pname /\*按商场名字分组\*/



图2. 5 视图AVE\_PRICE

1. 实验内容3.1部分。

功能需求：查询所有以“打折”方式销售的商品的当前销售情况,并按照价格的降序排列;

解决需求：先从SALE表中查找出Atype为打折的商品，然后利用“order by”子句对查询结果进行降价排序。实验代码如下所示，执行结果见图2. 6。

select Gname,Price

from SALE

where Atype = '打折'

order by Price desc;/\*按价格降价排序\*/



图2. 6 查询结果1

1. 实验内容3.2部分。

功能需求：查询所有没有任何活动的商品及所在的商场,结果按照商品排序;

解决需求：此查询与上一个基本一致，新增了对IS NULL子句的使用。实验代码如下所示，执行结果见图2. 7。

select Gname,Pname

from SALE

where Atype IS NULL /\*判断Atype是否为空\*/

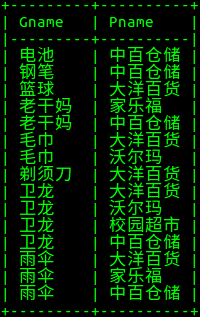


图2. 7 查询结果2

1. 实验内容3.3部分。

功能需求：查询价格在 200~500 元之间的商品名称、所在的商场名称、价格,结果按照商场名称排序;

解决需求：此查询新增了对比较运算符和逻辑与的使用。实验代码如下所示，执行结果见图2. 8。

select Gname,Pname,Price

from SALE where Price >= 200 and Price <= 500;



图2. 8 查询结果3

1. 实验内容3.4部分。

功能需求：查询每种商品的最低价格、商品名称;

解决需求：由于要求每种商品的最低价格，第一种思路是采用嵌套查询，第二种思路是采用group by子句先对表进行分组，然后调用MIN函数对每组进行求最小值。这里采用后一种思路，实验代码如下所示，执行结果见图2. 9。

select Gname,MIN(Price) /\*调用MIN子函数求最低价格\*/

from SALE

group by Gname;/\*按照名字分组\*/



图2. 9 查询结果4

1. 实验内容3.5部分。

功能需求：查询以“送券”方式销售的商品总数超过 2 种的商场名称;

解决需求：采用嵌套查询，先从PLAZA表中选出一个条目，然后在SALE表中找到对应商场的商品并且调用count子函数对活动类型为“送券”方式的商品进行计数，将计数结果大于2的输出。实验代码如下所示，执行结果见图2. 10。

select Pname

from PLAZA

where 2 < (

select count(\*) /\*调用count子函数进行计数\*/

from SALE

where SALE.Pname = PLAZA.Pname and Atype = '送券');



图2. 10 查询结果5

1. 实验内容3.6部分。

功能需求：查询以“送券”方式销售的商品总数超过 2 种的商场所在地区;

解决需求：基于上一个查询的实现，只需要将select所挑出的值改为Parea即可。实验代码基本一致，执行结果见图2. 11。



图2. 11 查询结果6



图2. 12 查询结果7

1. 实验内容3.7部分。

功能需求：查询价格为下列取值之一的商品名称、所在商场名称、目前举办活动的类型(88、188、288);

解决需求：此查询新增了对逻辑或（or）的使用。实验代码如下所示，执行结果见图2. 12。

select Gname,Pname,Atype

from SALE

where Price = '88' or Price = '188' or Price = '288';

1. 实验内容3.8部分。

功能需求：查询以“老”字开头的所有商品的名称;

解决需求：此查询新增了对正则规则的匹配，SQL语言中只支持任意字符“%”和任意单字符“\_”。实验代码如下所示，执行结果见图2. 13。

select Gname

from GOODS

where Gname like '老%';



图2. 13 查询结果8

1. 实验内容3.9部分。

功能需求：查询同时销售“剃须刀”和“电池”的商场名称;

解决需求：一开始，想到的是直接用INTERSECT子句对两个集合求交集找出同时满足两个条件的商品，却发现新版的MySql中不支持该子句，于是使用嵌套查询。实验代码如下所示，执行结果见图2. 14。

select Pname

from SALE

where Gname = '剃须刀' and

Pname IN(/\*在子查询中选出商品名称为电池\*/

select Pname

from SALE

where Gname = '电池'

);



图2. 14 查询结果9

1. 实验内容3.10部分。

功能需求：查询不举办任何活动的商场。

解决需求：先从SALE表中选出Pname和Atype两个属性列（用DISTINCT去重），并以此建立一个子表。接着利用嵌套查询找出有且仅有空的活动类型的数据项。实验代码如下所示，执行结果见图2. 15。

select Pname

from(/\*创建一个子表\*/

select DISTINCT Pname,Atype

from SALE

)as S1

where Atype NOT IN(

select Atype

from SALE

where SALE.Pname = S1.Pname and

SALE.Atype IS NOT NULL);

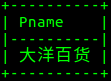


图2. 15 查询结果10

1. 实验内容3.11部分。

功能需求：查询所销售的商品包含了“校园超市”所销售的所有商品的商场名称。

解决需求：这是一个全称命题，而SQL语句中没有相应的语句，但是可以用存在量词对其进行表示。即改为“对于某商场，在校园超市里面，不存在某商品，该商场没有”。实验代码如下所示，执行结果见图2. 16。

select DISTINCT Pname

from PLAZA P1

where NOT EXISTS(

select \*

from SALE S1

where S1.Pname = '校园超市' and

NOT EXISTS(

select \*

from SALE S2

where P1.Pname = S2.Pname and

S2.Gname = S1.Gname)

);



图2. 16 查询结果11

经验证，上述查询结果全部与实验预期结果相符。

### 2.4.3 选做部分

## 2.5 经验总结

# 3 实验二 DBMS综合操作

## 3.1 实验目的

熟悉并掌握一款DBMS软件的备份机制、权限管理和控制以及内部存储等相关过程。

## 3.2 实验内容

（1）学习 sqlserver 的两种完全备份方式:数据和日志文件的脱机备份、系统的备份功能(选做)。

参照实验指导材料。

（2）学习系统的身份、权限配置操作

参照实验指导材料

（3）了解系统的查询性能分析功能(选做)

通过对第一部分任务中的 SQL 语句的查询性能分析,学习 SQL 查询的关系代数处理及其优化原理。

（4）了解 SQLSERVER 的数据导入导出功能(选做)

操作 SQLSERVER 的企业管理器集成界面,通过交互式方式实现数据的导入和导出。

（5）了解 SQLSERVER 的存储过程、触发器、函数实现过程

通过查看 SQLSERVER 的联机帮助文档和示例,体会存储过程、函数和触发器的原理,尝试编制具备基本功能的存储过程、触发器和函数的实际例子,查看其执行效果。

（6）了解 SQLSERVER 与常用应用程序开发环境集成的实现过程(选作)。

通过开发简单的 DBMS 应用程序,了解基本的数据库应用编程方法与技术,包括:数据库连接、嵌入式的数据读取操作、数据更新操作、事务提交技术等等。

## 3.3 实验环境

操作系统：Ubuntu16.04

DBSM软件：mysql 5.7.18

## 3.4 实验过程

1. 使用mysqldump -u root -p MyGoods> a.sql进行备份。
2. Mysql的身份、权限配置操作

mysql数据库中有4个权限表：user、db、tables\_priv和columns\_priv。

权限表的存取过程如下：

1. 先从user表中的host、user、password三个字段中判断连接的IP、用户名、密码是否存在表中，存在则通过身份验证。
2. 通过权限验证，进行权限分配时，按user>db>tables\_priv>columns\_priv的顺序进行分配。即先检查全局权限表的user，如果user中对应的权限为Y，则此用户对所有的数据库的权限都是Y，将不再检查其余表;否则，检查db表，依此类推。

基本命令：grant 权限 on databaseName.tableName to userName@localhost identified by “password”

Mysql共有27种权限，下面列举几种自己实践过的。

|  |  |
| --- | --- |
| Usage | 连接登录权限，默认授予 |
| Select | 查询权限 |
| Create | Create table权限 |
| Create view | Create view权限 |
| insert | Insert权限 |
| Alter | 修改表格权限 |
| Update | 更新表格权限 |
| delete | 删除表格权限 |
| drop | 删除表或数据库的权限 |
| Index | 创建索引的权限 |

1. mysql系统的查询分析功能

Explain select ...

1. mysql数据的导入与导出功能

·数据导出

Select ... Into outfile ‘文件名’

直接执行上面的语句会报错，需要指定输出文件的安全路径，可以先用语句‘show variables like “secure\_file\_priv”’找出安全路径，然后，在文件名中添加路径即可。当然，也可以通过修改配置文件更改路径。

·数据导入

Load data infile ‘文件名’into ...

也可以仿照前面的备份使用mysqldump命令进行数据的导入与导出。

（5）mysql的存储过程、触发器、函数实现过程

·存储过程

存储过程是预编译的，在首次运行一个存储过程时查询，优化器对其进行分析优化，并且给出最终被存储在系统表中的执行计划。而批处理的SQL语句在每次运行时都要进行编译和优化。所以存储过程的速度要快一些。

语法：create procedure 过程名（过程参数...）[特性...]过程体

·函数

存储函数需要返回一个类型，且函数体中必须要有一个有效的return语句，调用过程与调用内部函数MAX等类似。

语法：create function 函数名（参数...）

·触发器

触发器是一种特殊的存储过程，在特定表执行更新，插入，删除操作时触发，具有更好的数据控制能力。

1. 在C/C++中使用Mysql数据库

首先，需要在编译参数中指定mysqlclient库，常用编译命令是

g++ -o 可执行文件名 \*.cpp -L /usr/include/mysql -l mysqlclient

说明：-l后省略了lib\*\*\*.so。

然后，程序中包含头文件<mysql/mysql.h>，下面简记几种常用的mysql函数。

1. MYSQL \*mysql\_init(MYSQL \*mysql)

功能：获得或初始化一个MYSQL结构，返回mysql句柄。

1. void mysql\_close(MYSQL \*mysql)

功能：关闭一个服务器连接，并释放与连接相关的内存。

MYSQL \* mysql\_connect(MYSQL \*mysql,const char \*host,const char \*user,const char \*passwd);

功能：连接一个mysql服务器。

说明：该函数并不推荐使用，而是被下面这个函数替代了。

3）MYSQL \*mysql\_real\_connect(MYSQL \*mysql,const char \*host,const char \*user,const char \*passwd,const char \*db,unsigned int port,const char \*unix\_socket,unsigned int client\_flag);

函数传入值： mysql表示一个现存mysql结构的地址

   host表示MYSQL服务器的主机名或IP

   user表示登录的用户名

   passwd表示登录的密码

   db表示要连接的[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "/home/iyuge2/文档\\x/_blank)

   port表示MySQL服务器的TCP/IP端口

   unix\_socket表示连接类型

   client\_flag表示MySQL运行ODBC数据库的标记

函数返回值： 如果连接成功，一个MYSQL\*连接句柄：如果连接失败，NULL

1. int mysql\_query(MYSQL \*mysql,const char\*query);

功能：执行SQL语句char\*query。成功返回零。

1. int mysql\_select\_db(MYSQL\*mysql,const char \*db);

功能：选中一个数据库

1. int mysql\_create\_db(MYSQL\*mysql,const char \*db);

功能：创建一个数据库

7）unsigned int mysql\_num\_fields(MYSQL\_RES \*res);

功能：返回指定结果中列的数量。