《数据库系统》课程设计报告

题目	宾馆客房管理系统		
小组成员信息			
姓名	**学号**	**班级**	**分工**
郝裕玮	18329015	计科 1 班	部分存储过程+部分 C++嵌入编程+共同完成实验报告+排版
张闯	18325071	计科 1 班	数据库构建+部分存储过程+部分 C++嵌入编程+共同完成实验报告
马淙升	19335153	计科 1 班	命令行窗口 UI 优化+部分 C++嵌入编程+共同完成实验报告

提交时间： 2022 年 1 月 9 日

一、开发环境与开发工具

MySQL Workbench 8.0 CE + Visual Studio 2019

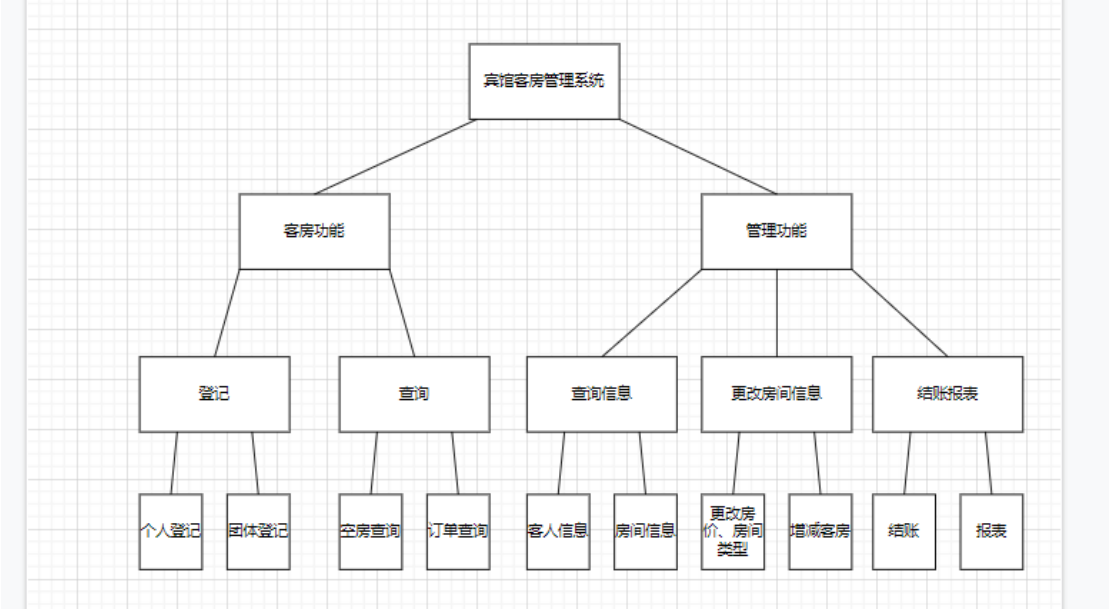
二、系统需求分析(2 分)

系统数据字典如下：

编号	数据结构	属性
1	房间 room	房间号，房间类型，房间价格
2	客人 guest	客人 id，客人名字
3	预定信息 reserve	日期，房间号，客人 id，折扣
4	当前订单 rorder	订单 id，客人 id，订单价格，入住日期，离店日期，是否团队预定
5	历史订单 oorder	订单 id，客人 id，订单价格，入住日期，离店日期，是否团队预定
6	汇总 summary	日期，总收入

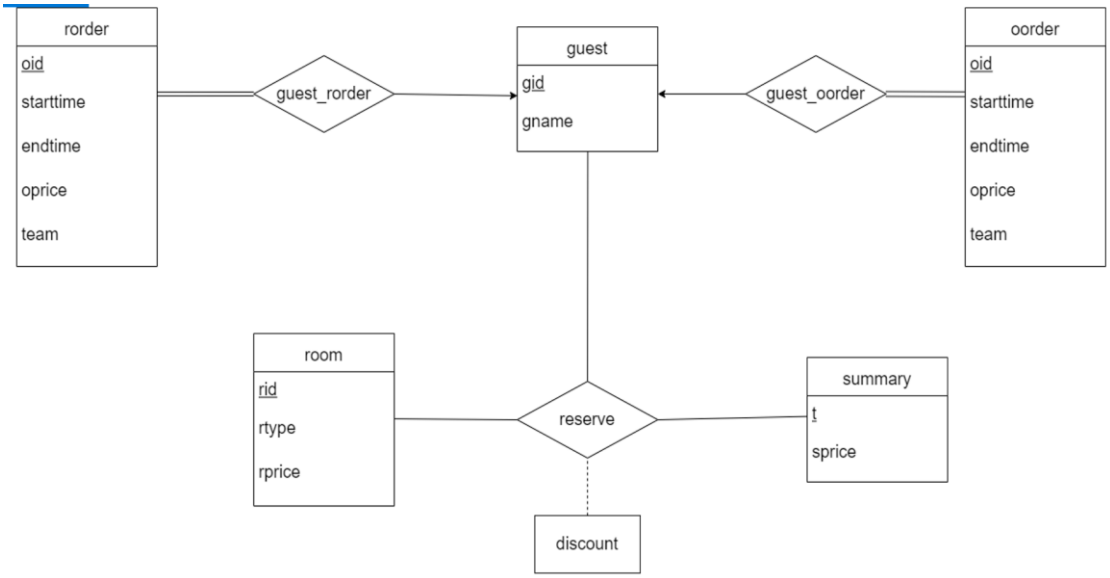
三、功能需求分析（3 分）

系统功能模块图：



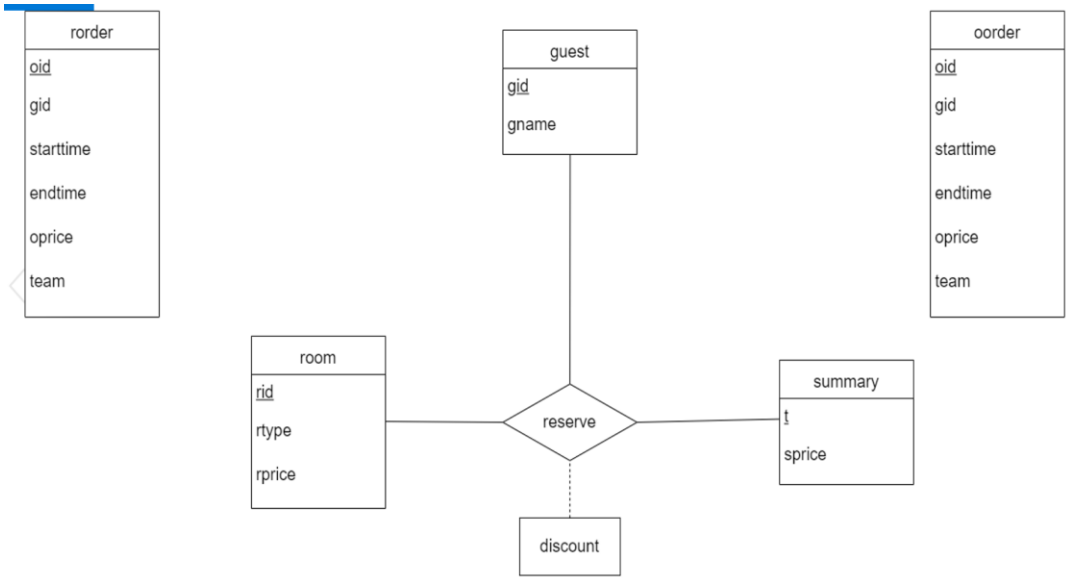
四、系统设计（10 分）

数据概念结构设计（系统 ER 图）（5 分）



数据库关系模式设计（3 分）

在 ER 图中，实体集 rorder 在联系集 guest_rorder 中是全参与的；实体集 oorder 在联系集 guest_oorder 中是全参与的。所以可将这两个联系集与对应的实体集进行合并，得到如下 ER 图：



我们根据 ER 图得到我们数据库的关系模式。包括以下关系：

guest：用来存储客户信息

room：用来存储房间信息

oorder：用来存储历史订单

rorder：用来存储当前订单

reserve：用来存储被预定的

summary：用来存储每天的营收

其中关系模式的外码约束如下：

oorder：gid 参照 guest 关系

rorder：gid 参照 guest 关系

reserve：gid 参照 guest 关系　rid 参照 room

我们用下列表格来表示我们设计的关系模式，其中，加粗部分为关系名称，下划线部分表示主码。

可以注意到，历史订单和当前订单具有相同的属性，但我们把他们分成两个

关系，这样可以在进行相关操作时只对其中一个关系进行，减少对关系操作带来的开销，同时，这也更加符合我们的逻辑。

summary 表主要用于统计营收，它的创建是非必要的，考虑到我们历史存储的订单数目可能会比较大（比如要保存一年内的订单，可能有十万条记录），而查询营收状况是一种频繁的需求，此时直接到历史订单表中查询某个时间段的营收将耗费较长时间，因此我们创建 summary 这一原本非必要的关系，以较小的空间开销，提高我们营收查询的效率。这个过程也可以通过建立索引来实现，但经过分析，我们得出建立 summary 综合开销和效率要优于建立索引的方式。

考虑到一些特殊情况下酒店可能会针对某间房打折扣（比如某间房出现问题被投诉），我们在预定的房间中增加了折扣这一属性。

最终，我们在 3NF 下建立如下的关系模式。

guest

gid，客人 id
gname，客人姓名

room

rid，房间 id
rtype，房间类型
rprice，房间价格

reserve

rid，房间号
t，日期
gid,客人 id
discount，折扣

Summary

t，日期
sprice，营收

oorder

oid，订单 id
starttime，入住时间
endtime，离开时间
gid，客人 id
oprice，订单价格
（房间价格*折扣）
team，是否为团体预定

rorder

oid，订单 id
starttime，入住时间
endtime，离开时间
gid，客人 id
oprice，订单价格
（房间价格*折扣）
team，是否为团体预定

数据库物理结构设计（2 分）

我们使用的的存储引擎为 InnoDB，主要因为 InnoDB 有如下优点：

（1）支持事务

（2）支持外键

（3）支持崩溃后的安全恢复

而我们酒店恰巧需要上述三个特性的支持，InnoDB 虽然不支持全文索引，但我们也根据这一特点设计了一些临时表和索引，减少了这一特性对数据库操作开销的影响。

具体关系模式中属性类型的物理设计如下：

guest	varchar(18) [gid]　varchar(10) [gname]

room	int [rid]　int [rtype]　float [rprice]

guest	varchar(18) [gid]　varchar(10) [gname]

reserve	int [rid]　varchar(18) [gid]　Date[t]　float [discount]

oorder	int [oid]　date [starttime]　Date [endtime]　varchar(18) [gid] float [oprice]　int [team]

rorder	int [oid]　Date [starttime]　Date [endtime]　varchar(18) [gid] float [oprice]　int [team]

summary	Date [t]　float [sprice]

五、系统功能的实现 (5 分)

主要功能模块的实现过程（简述）、运行界面

首先我们在 MySQL workbench 中创建了相关属性集：

```
create database hotel default charset=UTF8;

use hotel;
create table room(
    rid int not null,
    rtype int,
    rprice float,
```

```
    primary key(rid))default charset=UTF8;

create table guest(
    gid varchar(18) not null,
    gname varchar(10),
    primary key(gid))default charset=UTF8;

create table rorder(
    oid int unique AUTO_INCREMENT,
    starttime date,
    endtime date,
    gid varchar(18),
    oprice float,
    team int,
    primary key(oid),
    foreign key (gid)references guest(gid))auto_increment=1 default
charset=UTF8;

create table reserve(
    rid int,
    gid varchar(18)  default '00000000000000000',
    t date,
    discount float,
    primary key (rid,t),
    foreign key (rid) references room(rid),
    foreign key (gid) references guest(gid))default charset=UTF8;

create table summary(
    t date,
    sprice float,
    primary key(t)
)default charset=UTF8;

create table oorder(
    oid int,
    starttime date,
    endtime date,
    gid varchar(18),
    oprice float,
    team int,
    primary key(oid),
    foreign key (gid)references guest(gid))default charset=UTF8;
```

之后我们在 MySQL workbench 中实现了需要在嵌入式编程中调用的各种存储过程：

（1）search

```
use hotel;
delimiter //
create procedure search(rt int,st date,et date)
begin
    declare n int;
    declare i int default 0;
    set n=datediff(et,st);
    select num,rtype
    from(
    select count(distinct T.rid)as num,rtype
    from(select rid,count(distinct reserve.t) as d from reserve
where st<=reserve.t and et>reserve.t and
reserve.gid='00000000000000000' group by(rid))as T,room
    where d=n and room.rid=T.rid and room.rtype=rt)as S
    where S.num>0;
end//
```

搜索功能主要是利用存储过程 search(rt int,st date,et date)，查询入住时间为 st，离店时间为 et 的剩余 rt 类型房间还有多少，大致思路为从 reserve 表里选出在 st 与 et 之间每天的空房，并按照房号分组，统计每天都空闲的房号的数量。

（2）orderr

```
use hotel;
delimiter //
create procedure orderr(rt int,st date,et date,num int,id
varchar(18),te int)
begin
    declare n int;
    declare i int default 0;
    declare nprice float default 0;
    declare price float;
    declare d date;
    set n=datediff(et,st);
    while i<n
    do
        set d = date_add(st,interval i day);
```

```
        create table r select reserve.rid,t,rprice from reserve,room
where reserve.rid=room.rid and rt=room.rtype and d=reserve.t and
gid='00000000000000000' limit num;
        update reserve set gid=id where rid in(select rid from r) and
t=d;
        select sum(p)into price from (select
r.rprice*reserve.discount as p from r,reserve where
r.rid=reserve.rid and reserve.t=d)as P;
        set nprice=nprice+price;
        set i=i+1;
        drop table r;
    end while;
    insert into rorder(starttime,endtime,gid,oprice,team)
values(st,et,id,nprice,te);
    select oid from rorder where starttime = st and endtime = et and
gid=id and oprice=nprice and team=te;
end//
```

预定功能主要是利用存储过程 orderr(rt int,st date,et date,num int,id varchar(18),te int)，为客户 id 为 id 的用户预定从 st 到 et 的 num 间 rt 类型的房，以及添加是否团体预定的信息，大致思路为对每一天选出满足的 num 个房号进行修改，并添加订单信息到 rorder 表

（3）pay

```
use hotel;
delimiter //
create procedure pay(id int)
begin
    declare st date;
    declare price float;
    set st=(select endtime from rorder where oid=id);
    select oprice into price from rorder where oid=id;
    if st not in (select t from summary)then
    insert into summary values(st,0);
    end if;
    insert into oorder (select * from rorder where id=oid);
    delete from rorder where id=oid;
    update summary set sprice=sprice+price where st=t;
end//
```

结账功能主要是利用存储过程 pay(id int)来对 oid 为 id 的订单结账，将其移除 rorder 表并添加进历史订单 oorder 表，并将相关数据添加到 summary 表便于

实现报表功能。

（4）addr

```
use hotel;
delimiter //
create procedure addr(in rtype1 int,in num1 int,in rprice float)
begin
    declare Maxim int;
    declare cnt int default 1;
    declare num int;
    set num=num1;
    select max(rid) into Maxim from room where rtype=rtype1;
    while cnt<=num
        do
        insert into room values(Maxim+cnt,rtype1,rprice);
        set cnt=cnt+1;
    end while;
end//
```

该存储过程作用为：根据提供的房间类型 rtype1，选出该类型最大的房间号，并向后增加 num1 个同类型房间（房号依次递增）。

（5）minus

```
use hotel;
delimiter //
create procedure minus(in rtype1 int,in num1 int)
begin
    declare Minim int;
    declare cnt int default 0;
    declare num int;
    set num=num1;
    select min(rid) into Minim from room where rtype=rtype1;
    while cnt<=num-1
        do
        delete from room where rid=Minim+cnt;
        set cnt=cnt+1;
        end while;
end//
```

该存储过程作用为：根据提供的房间类型 rtype1，选出该类型最小的房间号，并删除 num1 个同类型房间（房号依次递增）。

最后是嵌入式程序中各种调用存储过程和 SQL 语句的函数以及 UI 的实现，完整代码由于过长，不再放在报告中，可见压缩包中的 mysql.cpp 即可（包含注释）。

具体的界面设计逻辑如下所示：

运行我们服务程序，会显示提示信息“Welcome to Our Hotel”

输出提示信息，需要登录或者注册：

（1）选择注册：

输入姓名、证件、密码，完成注册，若已经注册等原因，会显示注册失败，若注册成功，返回上一个登录过程。

（2）选择登录：

提示输入账号密码，若账号密码错误，则无法登录，登录成功将进入到酒店操作过程。

登录成功后会根据用户为管理员或者客户输出提示信息：

客户：权限包括查询空房，预定房间，查询个人订单

（1）查询空房：

输入入住时间和离开时间查询

（2）预定房间：

需要输入入住时间、离开时间、证件号、房型，若为团体预定，可以选择团体预定选项并一次性订多间房，若无该时段的房间类型，输出预定失败信息。

（3）查询个人订单

输入证件号，返回个人订单

管理员：权限包括客户权限+查询房间信息，查询客户信息，增加房间，删减房间，修改房间价格，修改房间类型，订单结账和查询报表。

（1）查询房间信息：

输入房号，输出房间信息

（2）查询客户信息：

有三种查询方式：

1.查询所有客户信息：

无需额外信息，输出所有客户信息

2.根据证件查询客户信息：

输入证件信息，返回相应客户信息

3.根据订单号查询客人信息：

输入订单号，查询客人信息。

（3）增加房间：

输入需要增加的房号、房型和价格，若该房间已存在，输出无法加入该房间。

（4）删减房间：

输入删除的房号，若不存在该房号，则提示无法删除。

（5）修改房间价格：

输入房号和价格，若房间存在，修改房间价格，否则修改失败。

（6）修改房间类型：

输入房号和类型，若房间存在，修改房间价格，否则修改失败。

（7）订单结账：

输入订单号，若订单号存在且未结账，结账成功，否则结账失败。

（8）查询报表：

输入查询时间范围，返回这段时间的营收

同时我们上述过程中都会有提示信息提示进行何种操作和输入何种信息，并会对输入进行检查，防止注入式攻击。

六、总结

涉及到的理论课知识有：

（1）SQL 语言的使用

（2）嵌入式编程 SQL

（3）ER 图的设计

通过本次实验，我们三人都对于嵌入式 SQL 有了更深一步的理解，并能够熟练建立关系模型绘制其 ER 图。虽然在编程过程当中遇到了很多困难，但都一一解决。并通过不断尝试各种输入来完善我们的程序模型，收获颇多。