

1.首页：nas_user_reservation按日期查询显示当日预约

2.开单页面：

2.1nas_order订单表

```
1 create table if not exists nas_order(  
2 id int primary key AUTO_INCREMENT,  
3 u_id int not null, -- 用户id  
4 order_num char(16) not null, -- 订单号  
5 order_price double not null, -- 价格  
6 order_paymethod char(16) not null, -- 支付方式  
7 order_state char(16) not null, -- 订单状态, 是否支付  
8 order_createdate datetime not null, -- 创建订单时间,  
9 remarks text, -- 订单备注  
10 location char(8) --添加所属地字段  
11 );
```

2.2nas_order_service订单服务表

```
1 create table if not exists nas_order_service(  
2 id int primary key AUTO_INCREMENT,  
3 order_id int not null,  
4 service_id int not null  
5 );  
6 -- 订单服务一对多
```

2.3 nas_order_goods订单商品表

```
1 create table if not exists nas_order_goods(  
2 id int primary key AUTO_INCREMENT,  
3 order_id int not null,  
4 goods_id int not null  
5 );  
6 -- 订单商品一对多
```

3.添加会员页面：nas_user 会员表

```
1 drop table nas_user;  
2 create table if not exists nas_user(  
3 id int primary key AUTO_INCREMENT,  
4 name char(16) not null,  
5 phone int(11) not null,  
6 location char(16) not null, -- 注册地  
7 remarks text -- 备注  
8 );  
9 desc nas_user; -- 查看表结构  
10 -- car_id int not null, -- 车辆id 在user_car中设置u_id外键, use->car 一对多  
11 -- car_plate char(16) not null, -- 车牌
```

```
12 -- car_mileage double not null, -- 行驶里程
13 -- car_insurance datetime not null, -- 保险到期日期
14 -- car_model char(16) not null, -- 车品牌型号
```

4.订单管理页面：nas_order订单表 同开单页面

5.会员管理页面：

5.1 本店会员->多表连接查询 user&储蓄卡&年卡&礼包

5.1.1储蓄卡信息表：nas_debit_card

```
1 create table if not exists nas_debit_card(
2     id int primary key AUTO_INCREMENT,
3     u_id int not null, -- 用户id
4     card_species char(16) not null, -- 卡片种类
5     card_amount decimal not null, -- 余额
6     card_deduction decimal(2,1) not null, -- 则扣
7     card_location char(16) not null -- 办理地点
8 );
```

5.1.2 年卡信息表

```
1 create table if not exists nas_year_card(
2     id int primary key AUTO_INCREMENT,
3     u_id int not null, -- 用户id
4     yc_amount decimal not null, -- 额度
5     yc_fromtime datetime not null, -- 办理日期
6     yc_totime datetime not null, -- 到期日期
7     yc_state bit not null,
8     yc_location char(16) not null -- 办理地点
9 );
```

5.1.3 大礼包信息表

```
1 create table if not exists nas_user_giftpack(
2     id int primary key AUTO_INCREMENT,
3     u_id int not null,
4     gift_price int not null, -- 价格
5     gift_fromtime datetime not null, -- 办理日期
6     gift_totime datetime not null, -- 过期日期
7     gift_location char(16) not null,
8     gift_details text not null
9 );
10 --礼包详情这里可另建表做连接查询
```

5.2 添加会员项：nas_user&nas_car做多表插入

5.3会员信息项:

nas_user&nas_car&nas_debit_card&nas_year_card&nas_user_payrecord&nas_user_pointsrecord

5.3.1nas_user会员基本信息表

```
1 drop table nas_user;
2 create table if not exists nas_user(
3 id int primary key AUTO_INCREMENT,
4 name char(16) not null,
5 phone int(11) not null,
6 location char(16) not null, -- 注册地
7 remarks text -- 备注
8 );
```

5.3.2 nas_user_car用户车辆

```
1 create table if not exists nas_user_car(
2 id int primary key AUTO_INCREMENT,
3 u_id int not null, -- 用户id
4 brand char(8) not null, -- 品牌
5 series char(8) not null, -- 系列
6 plate char(8) not null, -- 车牌号
7 model char(8) not null, -- 型号
8 engine_model char(8), -- 车架号
9 driving_license char(16), -- 行驶证
10 old int, -- 年份
11 guide_price decimal, -- 指导价
12 insurance_expiry_date datetime not null, -- 保险到期
13 mileage decimal(16,2) not null -- 里程
14 );
```

5.3.3 储蓄卡: nas_debit_card通过连接查询得到用户姓名

5.3.4 年卡: nas_year_card通过连接查询得到用户电话、用户姓名、车牌号

5.3.5 大礼包: nas_user_giftpack通过连接查询得到用户电话、用户姓名

5.3.6 消费记录: nas_order通过连接查询得到用户名、手机号

5.3.7 积分记录: nas_integral_record通过连接查询得到用户名、手机号

```
1 create table if not exists nas_integral(
2 id int primary key AUTO_INCREMENT,
3 u_id int not null, -- 用户id
4 amount decimal not null -- 总积分
5 );
6 drop table nas_integral_record;
7 create table if not exists nas_integral_record(
8 id int primary key AUTO_INCREMENT,
9 u_id int not null,
10 changes int not null,
11 detail char not null,
12 changedate datetime not null
13 );
14 -- 地10行change关键字冲突
```

```

15 -- 积分变动 状态通过正负来判断
16 -- 操作描述 也可以创建操作类型表
17 -- 操作日期

```

6.员工管理页面：nas_employee员工表

```

1 create table if not exists nas_employee(
2 id int primary key AUTO_INCREMENT,
3 name char(16) not null,
4 phone int(11) not null,
5 gender char(16) not null, -- 性别
6 post char(16) not null, -- 岗位
7 state int(11) not null -- 是否在职
8 );
9
10 show create table nas_employee; -- 查看创建语句

```

7.商品管理页面：

7.1nas_goods商品表

```

1 create table if not exists nas_goods(
2 id int primary key AUTO_INCREMENT,
3 goods_name varchar(255) not null,-- 商品名字
4 goods_price decimal(10) not null,-- 商品价格
5 goods_commission decimal(10) not null,-- 商品提成
6 goods_stock decimal(10) not null,-- 商品库存
7 goods_state varchar(255) not null,-- 商品状态
8 goods_classification varchar(255) not null, -- 商品分类
9 goods_description varchar(255),-- 商品描述
10 goods_addtime date-- 商品添加时间
11 );

```

7.2 nas_goods_class 商品分类表

```

1 create table if not exists nas_goods_class(
2 class_id int primary key auto_increment,-- 商品id
3 class_name varchar(32) not null );-- 商品类型名

```

8.本店预约页面：nas_user_reservation用户预约表

| id | 用户 | 用户电话 | 车辆 | 预约服务 | 预约时间 | 预约状态 | 预约申请时间 | 备注 |
|----|----|------|----|------|------|------|--------|----|
| | | | | | | | | |

```

1 drop table nas_Service;
2 create table if not exists nas_service(
3 id int primary key AUTO_INCREMENT,

```

```

4      detail char(32) not null, -- 服务描述
5      price decimal not null -- 服务价格
6  );
7  drop table nas_user_reservation;
8  create table if not exists nas_user_reservation(
9  id int primary key AUTO_INCREMENT,
10 u_id int not null,
11 service_id int not null, -- 预约服务类型
12 appoint_time datetime not null, -- 预约时间
13 state varchar(255) not null, -- 预约状态
14 application_time datetime not null, -- 预约申请时间
15 remarks varchar(255) not null -- 备注
16 );

```

9.预售卡页面:

9.1 nas_reservation_card 预售卡表

```

1  create table if not exists nas_reservation_card(
2  id int primary key AUTO_INCREMENT,
3  u_id int, -- 用户id
4  species varchar(255) not null, -- 预约卡种类
5  num int not null, -- 卡号
6  amount decimal not null -- 金额
7  );

```

9.2 nas_card_recharge 充值记录表

```

1  create table if not exists nas_card_recharge(
2  id int primary key AUTO_INCREMENT,
3  r_amount decimal not null,
4  r_state varchar(255) not null, -- 支付状态（是否支付）
5  card_id int not null,
6  employee_id int not null -- 销售人员id
7  );
8  --id int primary key AUTO_INCREMENT, -- 充值记录id
9  --r_amount decimal not null, -- 充值金额

```

10.消耗管理页面:

10.1 nas_consumables 消耗品表

```

1  CREATE TABLE if not exists nas_consumables(
2  `id` int(11) NOT NULL AUTO_INCREMENT,
3  `con_id` int(11) NOT NULL COMMENT '消耗品编号',
4  `con_name` varchar(255) DEFAULT NULL COMMENT '消耗品名称',
5  `con_number` int(255) DEFAULT NULL COMMENT '消耗品数量',
6  PRIMARY KEY (`id`),
7  UNIQUE KEY `con_id` (`con_id`)
8  );

```

10.2nas_consumables_records消耗品记录表

可以根据时间段查询出入库记录

```
1  -- 出入库表
2  CREATE TABLE if not exists nas_consumables_records(
3    `id` int(11) NOT NULL AUTO_INCREMENT,
4    c_id int not null, -- 消耗品id
5    `status` int(1) DEFAULT NULL COMMENT '出入库标识 1 出库 0入库',
6    `ware_num` int(11) DEFAULT NULL COMMENT '出入库编号',
7    `ware_date` datetime DEFAULT NULL COMMENT '出入库时间',
8    PRIMARY KEY (`id`),
9    UNIQUE KEY `ware_id` (`ware_id`)
10 );
```

11.本店评论页面：nas_station_comment车站评论表

```
1  create table if not exists nas_station_comment(
2    id int primary key AUTO_INCREMENT,
3    reservation_id int not null, -- 根据预约服务得到打分评价
4    submit_time datetime not null, -- 评论时间
5    detail char not null,
6    level int not null
7  );
8  -- level int not null --分数
9  -- detail char not null, --评论
10 -- 连接查询得到服务的用户名、服务时间、订单号、服务类型
```

12.业绩统计页面：根据前面的订单、服务做连接查询进行页面的数据统计

13.售后管理页面：根据reservation预约表和服务表做连接查询

复习以下：

1. InnoDB支持事务，MyISAM不支持，对于InnoDB每一条SQL语言都默认封装成事务，自动提交，这样会影响速度，所以最好把多条SQL语言放在begin和commit之间，组成一个事务；

2. InnoDB支持外键，而MyISAM不支持。对一个包含外键的InnoDB表转为MYISAM会失败；

3. InnoDB是聚集索引，使用B+Tree作为索引结构，数据文件是和（主键）索引绑在一起的（表数据文件本身就是按B+Tree组织的一个索引结构），必须要有主键，通过主键索引效率很高。但是辅助索引需要两次查询，先查询到主键，然后再通过主键查询到数据。因此，主键不应该过大，因为主键太大，其他索引也都会很大。

MyISAM是非聚集索引，也是使用B+Tree作为索引结构，索引和数据文件是分离的，索引保存的是数据文件的指针。主键索引和辅助索引是独立的。

4. InnoDB不保存表的具体行数，执行select count(*) from table时需要全表扫描。而MyISAM用一个变量保存了整个表的行数，执行上述语句时只需要读出该变量即可，速度很快（注意不能加有任何WHERE条件）；

5. InnoDB不支持全文索引，而MyISAM支持全文索引，在涉及全文索引领域的查询效率上MyISAM速度更快高；PS：5.7以后的InnoDB支持全文索引了

6. MyISAM表格可以被压缩后进行查询操作

7. InnoDB支持表、行(默认)级锁，而MyISAM支持表级锁

8、InnoDB表必须有主键（用户没有指定的话会自己找或生产一个主键），而Myisam可以没有

9、InnoDB存储文件有frm、ibd，而Myisam是frm、MYD、MYI

InnoDB: frm是表定义文件，ibd是数据文件

Myisam: frm是表定义文件，myd是数据文件，myi是索引文件

如何选择

1. 是否要支持事务，如果要请选择innodb，如果不需要可以考虑MyISAM；
2. 如果表中绝大多数都只是读查询，可以考虑MyISAM，如果既有读也有写，请使用InnoDB。
3. 系统崩溃后，MyISAM恢复起来更困难，能否接受；
4. MySQL5.5版本开始InnoDB已经成为Mysql的默认引擎(之前是MyISAM)，说明其优势是有目共睹的，如果你不知道用什么，那就用InnoDB，至少不会差。

InnoDB为什么推荐使用自增ID作为主键？

答：自增ID可以保证每次插入时B+索引是从右边扩展的，可以避免B+树和频繁合并和分裂（对比使用UUID）。如果使用字符串主键和随机主键，会使得数据随机插入，效率比较差。

innodb引擎的4大特性

插入缓冲 (insert buffer),二次写(double write),自适应哈希索引(ahi),预读(read ahead)