

# 订餐系统 说明文档

雷怡然 2016013274

／\* 本说明文档分为三个部分，第一部分为订餐系统中使用的SQL语句的说明，按照语句功能的不同，分为了：查询、添加、删除、更新，一共四类，进行分类说明，在每类说明中依次列出了每条SQL语句的名称以及内容。第二部为创新功能说明，分为创新动机、原理、SQL语句说明和操作说明四部分。第三部分为新增表列的说明\*／

## 一、SQL语句及说明

### 1.查询功能

#### a) 查询空闲派送员：

```
SELECT *  
FROM 派送员  
WHERE 是否繁忙? =false  
ORDER BY 派送员编号;
```

功能：查询出空闲的派送员，以便实现订单分派。

#### b) 查询派送中派送员：

```
SELECT 派送员.派送员编号, 派送员.姓名, 订单.派送员反馈完成派送?  
FROM 派送员, 订单  
WHERE 派送员.是否繁忙? = True AND 派送员.派送员编号 = 订单.派送员编号;
```

功能：查询出繁忙的派送员，实现“完成分派”的操作。

#### c) 订单总览：

```
SELECT 订单.订单编号, 菜单.菜名 AS 订餐, 菜单.价格, 用户.用户姓名, 派送员.姓名 AS 派送员, 订单.  
预定时间 AS 下单时间, 订单.派送状态  
FROM 订单, 菜单, 用户, 派送员  
WHERE [订单].[预定菜品编号]=菜单.菜品编号 AND 订单.用户编号 = 用户.用户编号 AND 订单.派  
送员编号 = 派送员.派送员编号;
```

功能：查询出所需要的订单信息。

#### d) 未派送订单区域查询：

```
SELECT 用户.地址 AS 所属区域, COUNT(*) AS 未派送订单数量  
FROM 用户, 订单  
WHERE 用户.用户编号=订单.用户编号 AND 订单.派送状态='未派送'  
GROUP BY 地址;
```

功能：查询未派送订单的用户地址，以实现订单分派。

#### e) 用户订单查询精确查询：

```
SELECT *  
FROM 订单  
WHERE 用户编号 = forms!用户订单查询!c;
```

功能：在已知用户编号的前提下，实现精确的订单查询。

f) 用户订单查询模糊查询:

```
SELECT *  
FROM 订单  
WHERE 用户编号 LIKE '**& {forms!用户订单查询!w1 & '**';
```

功能: 在不完全清楚用户编号的情况下, 实现模糊的订单查询。

## 2.插入功能

a) 订餐:

```
INSERT INTO 订单 ( 用户编号, 预定菜品编号 )  
VALUES ([{forms!订餐!w1, {forms![订餐]!c);
```

功能: 在订单表中插入新的订单。

b) 添加菜品:

```
INSERT INTO 菜单  
VALUES ({forms![添加菜单]!b1, {forms![添加菜单]!b2, {forms![添加菜单]!b3);
```

功能: 把新的菜品信息插入到菜单表中。

c) 添加派送员:

```
INSERT INTO 派送员  
VALUES ({forms!管理派送员!b1, {forms!管理派送员!b2, NULL);
```

功能: 把新的派送员信息插入到派送员表中。

d) 添加用户:

```
INSERT INTO 用户  
VALUES ({forms![添加用户]!b5, {forms![添加用户]!b1, {forms![添加用户]!b2, {forms![添加用户]!b3,  
{forms![添加用户]!b4);
```

功能: 把新的用户信息插入到用户表中。

## 3.删除功能

a) 删除派送员:

```
DELETE *  
FROM 派送员  
WHERE 派送员编号={forms!管理派送员!c;
```

功能: 在派送员表中删除选定派送员。

## 4.更新功能

a) 订单由未派送到正在派送:

```
UPDATE 订单 SET 派送状态 = '正在派送', 派送员编号 = {orms!订单分配!c2}
WHERE 用户编号 IN (SELECT 用户编号
                    FROM 用户
                    WHERE 地址 = {orms!订单分配!c1}
                    );
```

功能: 订单分派时, 改变订单派送状态, 由“未派送”改为“正在派送”。

b) 订单由正在派送到已派送:

```
UPDATE 订单 SET 派送状态 = '已送达'
WHERE 派送员编号 = {orms!订单分配!c3};
```

功能: 订单分派完成后, 改变订单派送状态, 由“正在派送”改为“已派送”。

c) 分派员由繁忙到空闲:

```
UPDATE 派送员 SET 是否繁忙? = False
WHERE 派送员编号 = {orms!订单分配!c3};
```

功能: 完成订单派送后, 更新派送员的繁忙状态。

d) 分派员由空闲到繁忙:

```
UPDATE 派送员 SET 是否繁忙? = True
WHERE 派送员编号 = {orms!订单分配!c2};
```

功能: 订单开始分配时, 更新派送员的繁忙状态。

## 二、创新功能说明

### 1. 创新动机

在本次系统设计中, 我们设计了“订单分派”系统, 可以高效地让老板端掌握订单派送情况。但是, 在“订单分派”界面中, 老板端**只能查询到“派送中的派送员”** (也就是只知道哪些派送员状态为“繁忙”), **并不知道派送员是否已经完成了派送**。这给后续的“派送员由繁忙状态转为空闲状态”的操作, 造成了极大的困难。派送员的是否完成派送的信息难以传回到老板端。

另一方面, 本次系统设计中, **并没有针对派送员使用的界面**, 也就是说, 派送员**无法获取订单信息**, 但在现实生活中, 派送员是必须要知道订单派送地址以及预定者的电话、姓名等信息的。订单的信息无法有效地从老板端传递到派送员端。

从上述两方面考虑, 我们可以发现, 在**“派送员和老板的信息沟通”这一环节**, 现有的系统并不能做出有效的沟通, 因此, 本次创新就以解决上述两个问题为出发点, 以建立老板—派送员高效沟通系统为目的, 进行创新。

## 2.原理

a)

增加了子界面——“分派员”，该界面由一个文本框、一个表、一个组合框和两个按钮组成。在该界面中，分派员输入自己的“分派员编号”，便可在“表”中查询此时自己的分派任务，即获取订单地址、用户姓名、用户电话等信息。这样便实现了信息从老板端传递到分派员。

b)

在表“订单”中增添了列——“派送员反馈完成派送？”，其值为“真”或“假”，用来反映来自派送员的信息——“是否完成派送”（“真”即为完成派送，“假”表示未完成派送，默认值为假）。该值反映在老板端“订单分配”子界面“派送中的派送员”组合框中（原来是两列，分别是派送员编号、派送员姓名，现在改为三列，在原有的基础上增加了第三列“派送员反馈完成派送？”）。老板端便可清楚地知道派送是否完成。

当完成派送任务后，分派员在a)中所叙述的字界面中选中完成的派送，点击“完成派送”按钮，此时表“订单”中的列“派送员反馈完成派送？”的值便会从“False”变为“True”，这表示分派员已经完成了派送，相应地，在老板端的“订单分派”子界面中便可看到值的改变。

通过以上步骤，便实现了把“分派员是否完成派送”的信息从分派员传递到老板端的功能，这时老板就可以轻松完成之后的“分派员由繁忙状态变为空闲状态”的操作了。

c)

需要特别指出的是，a) 中子界面的“表”显示的是未完成的派送的订单信息，并且可以通过“刷新”按钮实现“表”和“组合框”的刷新。

## 3.SQL语句及说明

1.在分派员界面查询订单信息用到了一下sql语句。

派送员查看派送任务创新：

```
SELECT 订单.订单编号, 用户.用户姓名, 用户.电话, 用户.地址, 用户.楼号, 订单.预定时间 AS 下单时间  
FROM 订单, 用户
```

```
WHERE 订单.派送员编号 = forms!派送员创新!w AND 订单.派送状态 = '正在派送' AND 订单.用户编号  
= 用户.用户编号;
```

2.在更新表“订单”中列“派送员反馈完成派送？”时用到了如下语句。

更新派送员反馈创新：

```
UPDATE 订单 SET 派送员反馈完成派送? = true
```

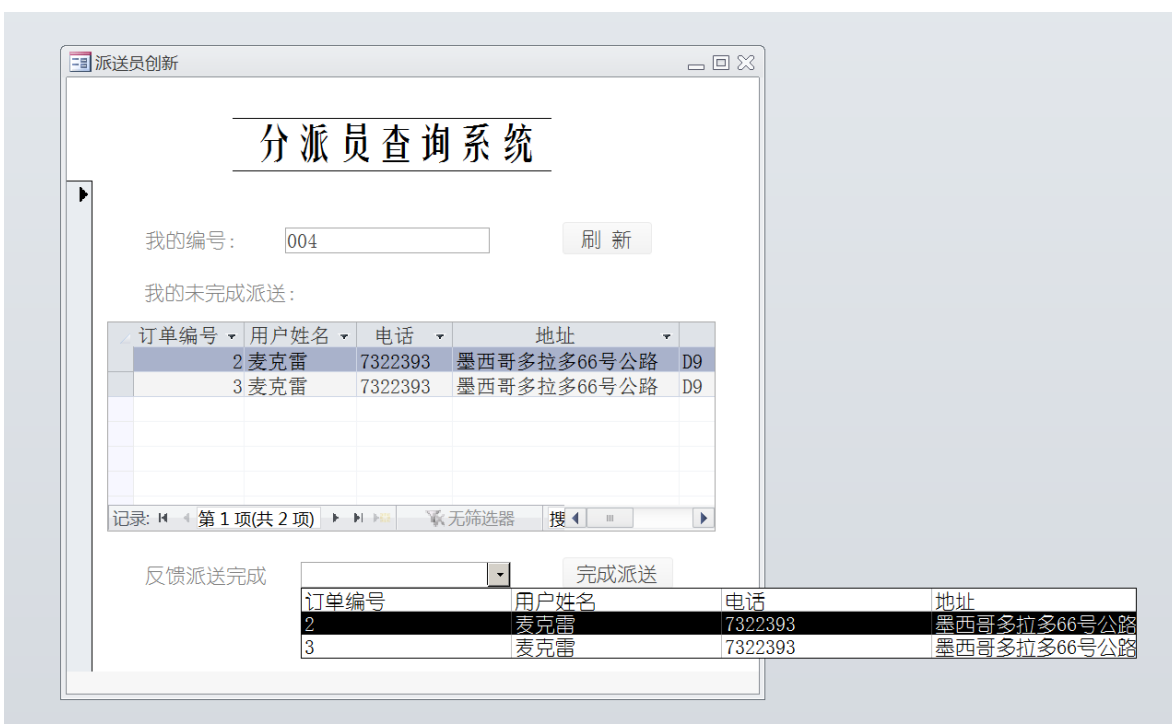
```
WHERE 派送员编号 = forms!派送员创新!w AND 订单编号 = forms!派送员创新!c;
```

## 4.操作说明

### 1.主窗体



### 2.”分派员“子界面



3.”订单分配“子界面有相应变化

订单分配

订单分配

所属区域

闲置派送员

派送中派送员

分派

派送已完成

老板通过该列可以知道派送是否完成

派送员编号	姓名	派送员反馈完成派送？
001	雷怡然	True
004	吕二哲	False
004	吕二哲	False
007	刘建国	True

三.增加表列说明

在表“订单”中增加了列“派送员反馈完成派送？”，用来表示从派送员端反馈的“派送是否完成”的信息。

新增列						
订单编号	用户编号	派送员编号	预定菜品编号	预定时间	派送状态	派送员反馈完成派送？
	8	001	003	2016/11/9 19:00:00	已送达	<input checked="" type="checkbox"/>
2	9	004	009	2016/12/21 8:00:00	正在派送	<input type="checkbox"/>
3	9	004	013	2016/12/21 8:00:00	正在派送	<input type="checkbox"/>
4	1	007	001	2016/11/23 21:00:00	已送达	<input checked="" type="checkbox"/>
8	2		003	2016/11/22 19:53:39	未派送	<input type="checkbox"/>
9	3		004	2016/11/22 20:27:33	未派送	<input type="checkbox"/>
*	(新建)			2016/11/26 20:07:27	未派送	<input type="checkbox"/>