



# 交通监控数据库系统

## 数据库设计报告

文件状态：	文件标识：	GUCAS-TRAFFIC_CONTROL-SD-DATABASE
<input checked="" type="checkbox"/> 草稿	当前版本：	1.0
<input type="checkbox"/> 正式发布	作 者：	ZENG DING HENG
<input type="checkbox"/> 正在修改	完成日期：	2010-10-31

中国科学院研究生院      计算与通信学院

版 本 历 史

版本/状态	作者	参与者	起止日期	备注
VER1.0	曾定衡		2010-10-20 至 2010-10-31	Initial Version

目 录

1. 介绍 ..... 4

2. 数据库环境说明 ..... 5

3. 数据库的命名规则 ..... 5

4. 逻辑设计 ..... 6

5. 物理设计 ..... 7

    5.0 表和视图汇总 ..... 7

    5.1 表CAR\_INFOR ..... 10

    5.2 表DRIVER\_INFOR ..... 10

    5.3 表EVENT\_RECORD ..... 11

    5.4 表EAST\_RT ..... 11

6. 安全性设计 ..... 11

    6.1 防止用户直接操作数据库的方法 ..... 11

    6.2 用户帐号密码的加密方法 ..... 12

    6.3 角色与权限 ..... 12

7. 优化 ..... 13

8. 客户端设计说明 ..... 13

9. 结语 ..... 16

## 1. 介绍

随着生活水平的提高，城市汽车数量剧增，交通压力不断增大，市区内大范围的交通阻塞不仅浪费了市民大量的时间，而且对城市环境也造成了巨大危害。在一个倡导绿色环保的时代，必须实现城市交通提速。

本文的数据库系统用于监控十字路口的交通流量，根据即时流量信息，合理调节各向车辆的通行时间，并能够对过往十字路口的车辆进行称重；结合车牌号自动识别系统，从后台数据库可获知该车辆信息及车主信息，若有交通违规，如闯红灯，转向违例，超载，则即时记录，通过后台信息中心向车主发出通知。

## 2. 数据库环境说明

2.1. 数据库开发环境： Microsoft SQL Server 2005 Standard Edition

2.2. 应用软件开发环境： Borland C++ Builder 6

应用程序通过 ADO 组件与数据库进行连接和数据访问，直接生成可执行文件，供客户端使用；若用 BDE 则需要讲应用程序和 BDE 打包发布。

## 3. 数据库的命名规则

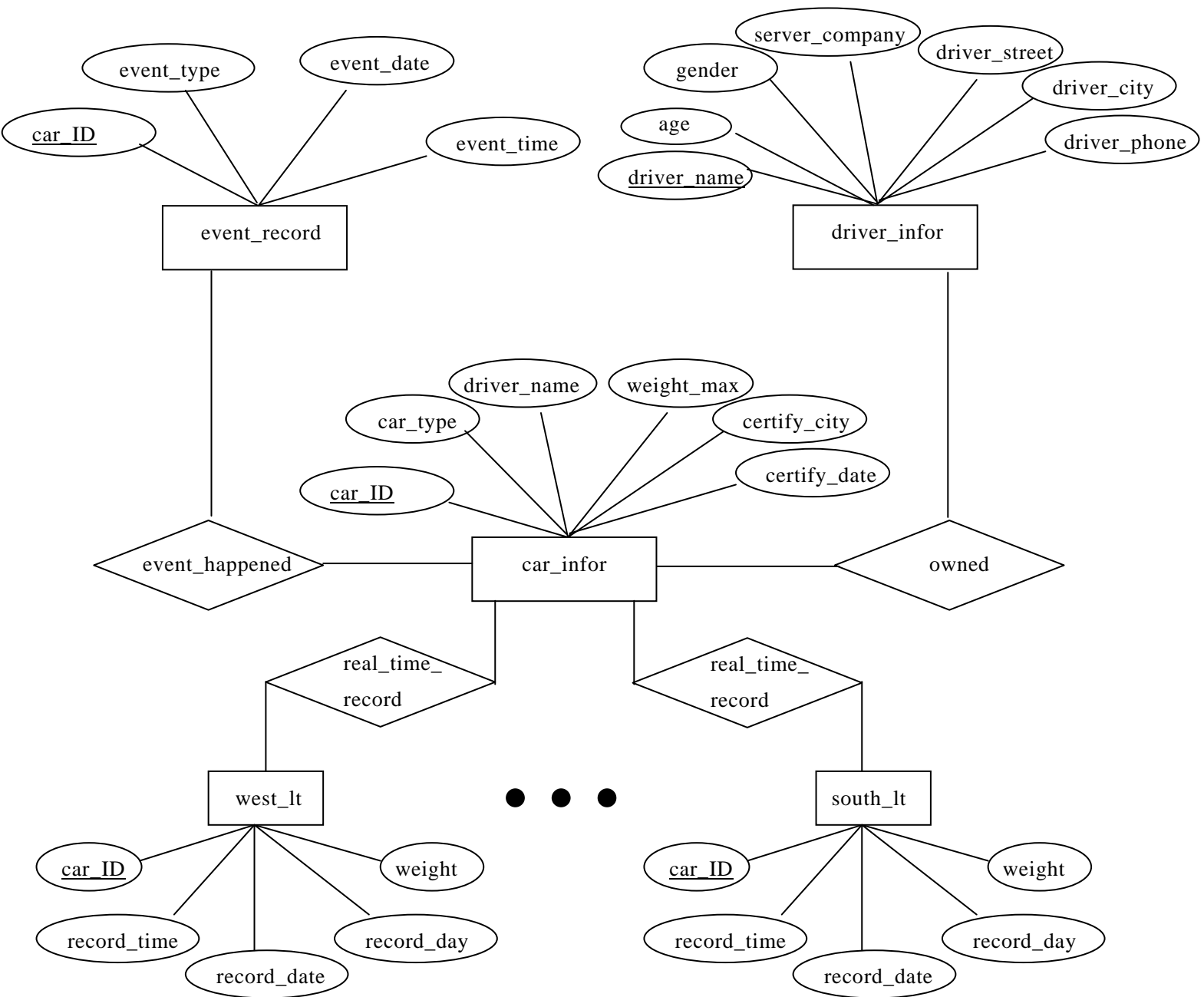
3.1. 数据库名称：首字母大写，若有下划线连接，则下划线后首字母也需大写，  
如：Traffic\_Control

3.2. 表名称：所有字母小写，可用下划线连接，如：driver\_infor, car\_infor

## 4. 逻辑设计

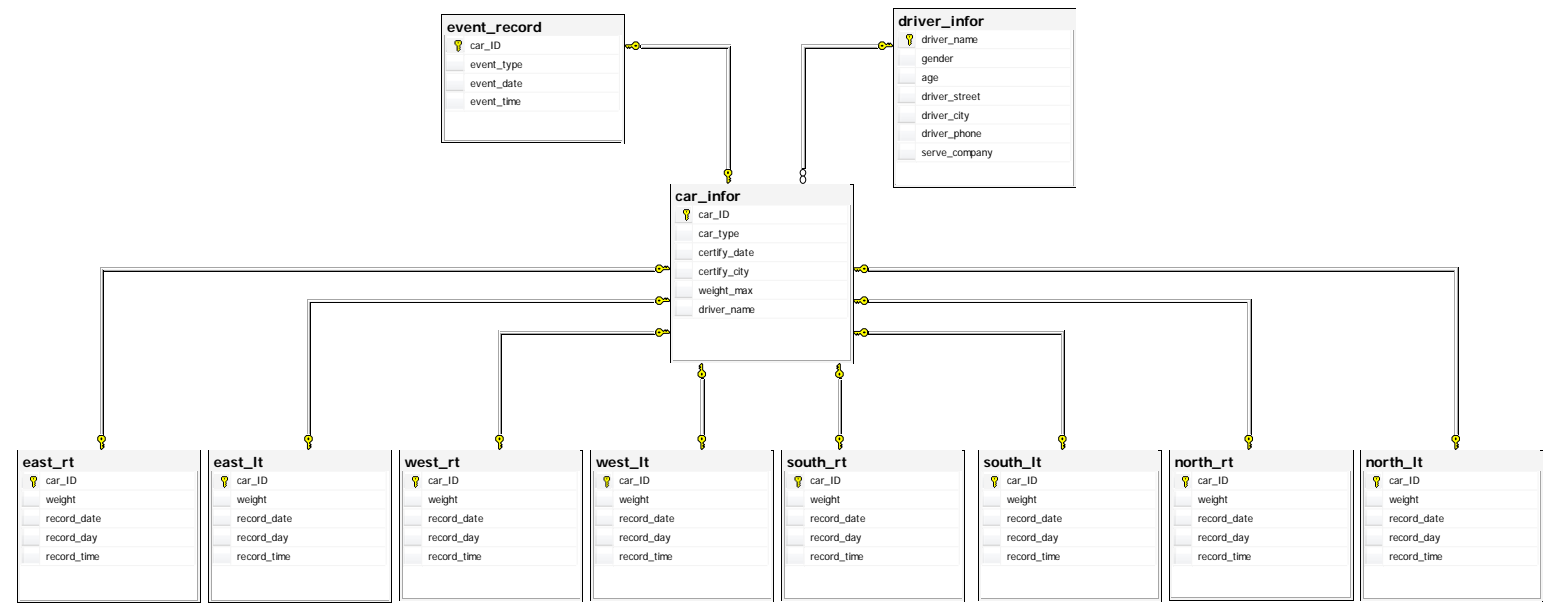
交通监控系统数据库 E-R 图如下：

注：该系统建立了八个用于记录四个方向进入和驶出十字路口的车辆通行表，分别为 west\_rt, west\_lt, east\_rt, east\_lt, south\_rt, south\_lt, north\_rt, north\_lt，各表的属性完全相同，图中只画出了 west\_lt 和 south\_lt 的实体-属性集。



## 5. 物理设计

参照 E-R 图，设计的表和数据库模式如下：



车辆通行表（west\_rt, west\_lt, east\_rt, east\_lt, south\_rt, south\_lt, north\_rt, north\_lt）和交通违例记录表（event\_record）中的属性 car\_ID 设计为外键，car\_infor 中的 car\_ID 设计为主键；表 car\_infor 中的属性 driver\_name 为表 driver\_infor 的外键。

### 5.0 表和视图汇总

表名	功能说明
car_infor	所有车辆的备案信息
driver_infor	所有车主的备案信息
event_record	交通违章车辆的记录
east_rt	记录东边进入十字路口的车辆
east_lt	记录东边驶出十字路口的车辆
south_rt	记录南边进入十字路口的车辆
south_lt	记录南边驶出十字路口的车辆
west_rt	记录西边进入十字路口的车辆
west_lt	记录西边驶出十字路口的车辆
north_rt	记录北边进入十字路口的车辆
north_lt	记录北边驶出十字路口的车辆

视图名	功能说明
ecdemic_car	所有外地车辆的统计信息
over_weight	所有超载车辆的统计信息
east_GO	由东边驶入十字路口，且向西、南，或者调头的车辆 ID 号
east_GO_CNT	east_GO 的计数，用于控制通行时间
north_GO	由北边驶入十字路口，且向东、南，或者调头的车辆 ID 号
north_GO_CNT	north_GO 的计数，用于控制通行时间
west_GO	由西边驶入十字路口，且向东、北，或者调头的车辆 ID 号
west_GO_CNT	west_GO 的计数，用于控制通行时间
south_GO	由南边驶入十字路口，且向西、北，或者调头的车辆 ID 号
south_GO_CNT	south_GO 的计数，用于控制通行时间

其中 ecdemic\_car 视图的创建 SQL 语句如下：

```
create view ecdemic_car as(
((SELECT      car_ID, weight, record_date, record_day, record_time
FROM          dbo.west_lt)
EXCEPT
(SELECT      car_ID, weight, record_date,record_day,record_time
FROM          dbo.west_lt
WHERE        (car_ID LIKE '京%'))
UNION
((SELECT      car_ID, weight, record_date, record_day, record_time
FROM          dbo.east_lt)
EXCEPT
(SELECT      car_ID, weight, record_date,record_day,record_time
FROM          dbo.east_lt
WHERE        (car_ID LIKE '京%'))
UNION
((SELECT      car_ID, weight, record_date, record_day, record_time
FROM          dbo.south_lt)
EXCEPT
(SELECT      car_ID, weight, record_date,record_day,record_time
FROM          dbo.south_lt
WHERE        (car_ID LIKE '京%'))
UNION
((SELECT      car_ID, weight, record_date, record_day, record_time
FROM          dbo.north_lt)
EXCEPT
(SELECT      car_ID, weight, record_date,record_day,record_time
FROM          dbo.north_lt
```



```
WHERE      (car_ID LIKE '京%'))))
```

其中 over\_weight 视图的创建 SQL 语句如下：

```
create view over_weight as(
SELECT      dbo.west_lt.car_ID, dbo.west_lt.weight,
dbo.west_lt.record_date, dbo.west_lt.record_time,
dbo.car_infor.car_type, dbo.car_infor.weight_max,
dbo.car_infor.driver_name
FROM        dbo.west_lt INNER JOIN dbo.car_infor ON
dbo.west_lt.car_ID = dbo.car_infor.car_ID AND dbo.west_lt.weight >
dbo.car_infor.weight_max
UNION
SELECT      dbo.east_lt.car_ID, dbo.east_lt.weight,
dbo.east_lt.record_date, dbo.east_lt.record_time,
car_infor_1.car_type, car_infor_1.weight_max,
car_infor_1.driver_name
FROM        dbo.east_lt INNER JOIN dbo.car_infor AS car_infor_1 ON
dbo.east_lt.car_ID = car_infor_1.car_ID AND dbo.east_lt.weight >
car_infor_1.weight_max
UNION
SELECT      dbo.south_lt.car_ID, dbo.south_lt.weight,
dbo.south_lt.record_date, dbo.south_lt.record_time,
car_infor_2.car_type, car_infor_2.weight_max,
car_infor_2.driver_name
FROM        dbo.south_lt INNER JOIN dbo.car_infor AS car_infor_2 ON
dbo.south_lt.car_ID = car_infor_2.car_ID AND dbo.south_lt.weight >
car_infor_2.weight_max
UNION
SELECT      dbo.north_lt.car_ID, dbo.north_lt.weight,
dbo.north_lt.record_date, dbo.north_lt.record_time,
car_infor_3.car_type, car_infor_3.weight_max,
car_infor_3.driver_name
FROM        dbo.north_lt INNER JOIN dbo.car_infor AS car_infor_3 ON
dbo.north_lt.car_ID = car_infor_3.car_ID AND dbo.north_lt.weight >
car_infor_3.weight_max)
```

其中 east\_GO 视图的创建 SQL 语句如下：

```
create view east_GO as(
SELECT      dbo.east_lt.car_ID AS east_GO
FROM        dbo.east_lt INNER JOIN dbo.south_rt ON
```

```
dbo.east_lt.car_ID = dbo.south_rt.car_ID
UNION
SELECT      east_lt_1.car_ID
FROM        dbo.east_lt AS east_lt_1 INNER JOIN dbo.east_rt ON
east_lt_1.car_ID = dbo.east_rt.car_ID
UNION
SELECT      east_lt_2.car_ID
FROM        dbo.east_lt AS east_lt_2 INNER JOIN dbo.west_rt ON
east_lt_2.car_ID = dbo.west_rt.car_ID)
```

车辆行驶方向的判断方法：  
若某车出现在表“西左行”（west\_lt）和表北右行（north\_rt）中，则该车的行驶方向为：  
由西左转至北，其他依次类推。  
该系统在计算交通流量时，以十字路口不限制车辆右转弯为前提，故不计右转弯的车辆。

其他方向通行车辆的视图依此法得到。

5.1 表car\_infor

表名	car_infor		
列名	数据类型（精度范围）	空/非空	约束条件
car_ID	char(10)	not null	
car_type	char(20)	null	
certify_date	float	null	
certify_city	char(20)	null	
weight_max	int	null	weight_max>=0
driver_name	char(20)	null	
补充说明			

5.2 表driver\_infor

表名	driver_infor		
列名	数据类型（精度范围）	空/非空	约束条件
driver_name	char(20)	not null	
gender	char(10)	null	
age	int	null	age>=18
driver_street	varchar(20)	null	
driver_city	char(20)	null	
driver_phone	char(20)	null	

serve_company	varchar(20)	null	
补充说明			

### 5.3 表event\_record

表名	event_record		
列名	数据类型（精度范围）	空/非空	约束条件
car_ID	char(10)	not null	
event_type	char(20)	null	
event_date	char(20)	null	
event_time	char(20)	null	
补充说明			

### 5.4 表east\_rt

表名	east_rt		
列名	数据类型（精度范围）	空/非空	约束条件
car_ID	char(10)	not null	
weight	int	null	weight>=0
record_date	float	null	
record_day	char(10)	null	
record_time	char(10)	null	
补充说明			

注：表 west\_rt, west\_lt, east\_lt, south\_rt, south\_lt, north\_rt, north\_lt 的属性同上表相同

## 6. 安全性设计

### 6.1 防止用户直接操作数据库的方法

本系统可由客户端直接对数据库进行操作，但是有权限设置，防止普通人员更改数据库，在登陆客户端时决定用户的权限，如下所示：



## 6.2 用户帐号密码的加密方法

对输入的密码进行字符屏蔽，若需再加强密码保护，可设置软键盘输入密码。



## 6.3 角色与权限

角色	可以访问的表与列	操作权限
普通用户	car_infor driver_infor east_rt, ..., north_lt	检索
		检索
		检索
管理员	所有表，视图	检索，更新

创建和删除表只能在服务器端完成

## 7. 优化

优先级	优化对象（目标）	措施
1	车辆通行表（east_rt,...,south_lt） 由于这 8 个表的数据量最大，且实时更新，所以对存储空间的要求较高，所以必须为其各属性值安排合理的数据类型及精度	record_date 的数据类型由 char 更改为 float 数据的字节数缩短为 4 字节，若为 char 至少需要 8 字节（sizeof"20100101"=8）
2	为了达到合理控制各向通行的时间，需要即时的交通流量统计信息，如果采用客户端从服务器读取数据再做统计则费时较大，需要一种快速得到交通流量信息的方法	在数据服务器端创建视图（east_GO，east_GO_CNT 等），得到各向交通流量数据，客户端通过访问视图，直接得到流量信息，做出交通调度。
3	对表 driver_infor 的查询：查找某几个城市（如北京，上海，广州）所有的车主信息，当数据量很大时，查询时间会很长	建立索引 driver_city_IDX，利用该索引进行查询，则效率会明显增加

## 8. 客户端设计说明

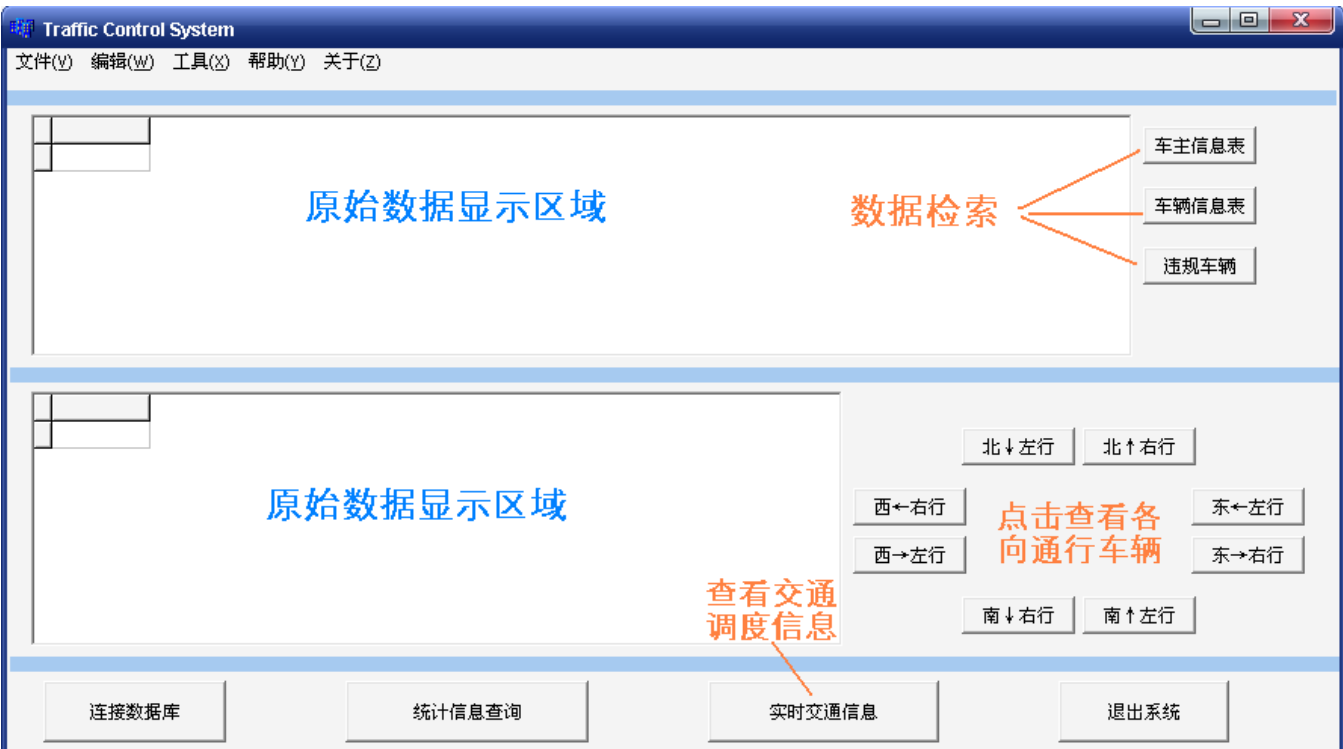
### 8.1 登陆界面



选择：普通用户，直接登陆，只有检索数据的权限

管理员，输入密码：dh 登陆，可以检索，修改数据

8.2 客户端主界面



上图中“左行”为驶入十字路口的车辆

“右行”为驶出十字路口的车辆

连接数据可后，点击相应按钮可查看原始数据信息：

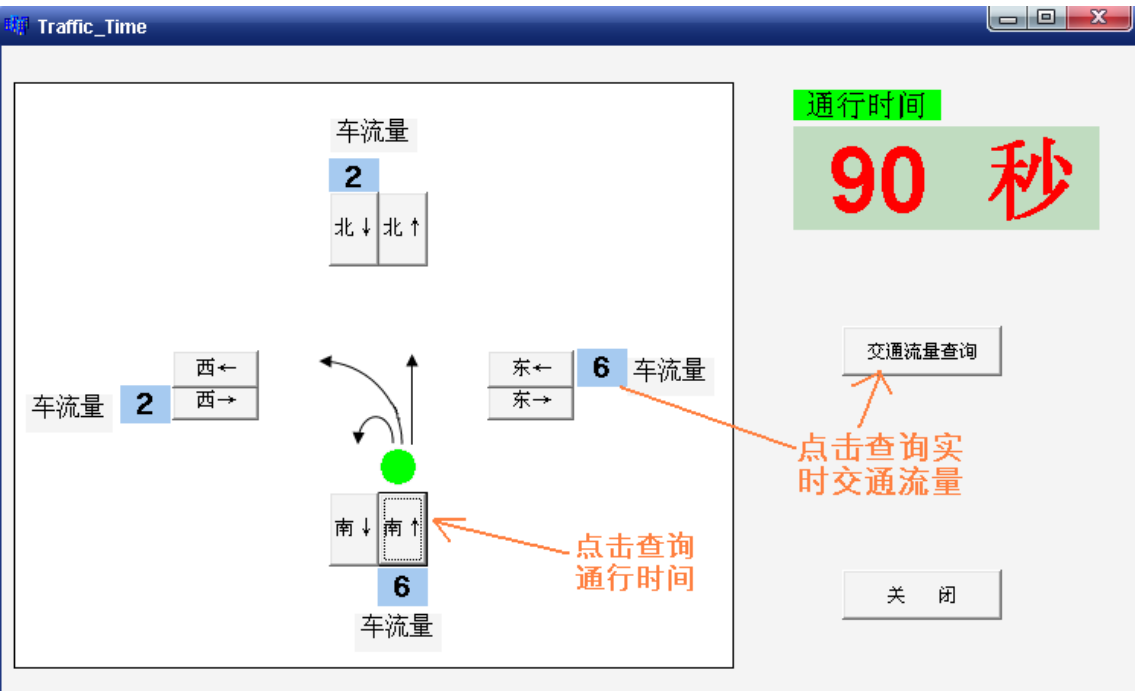


8.3 统计信息查询界面



在该界面中，可检索所有超载车辆记录和外地车辆记录；  
通过设定车辆年检日期，可查询到所有通过十字路口中未及时年检的车辆。

8.4 统计信息查询界面



各向车流量由访问视图(east\_GO\_CNT, west\_GO\_CNT, north\_GO\_CNT, south\_GO\_CNT)直接得到。

## 9. 结语

该交通监控系统只是对实际的模拟，实验所用数据为某个时间段之内（如 9: 00—9: 10）采集的，数据量较小，且客户端功能较为简单。若需实用，还要进行大量的工作，包括数据库的优化，服务器的架构，交通监控系统组网，实际环境的限制等等，其中车牌号识别系统是一大难题。倘若每辆车都装有 **RFID**（电子标签），在十字路口安装电子标签读写器，由于 **RFID** 可存储大量信息，包括车辆基本信息，车主信息等，故可考虑用 **RFID** 感知代替车牌号识别系统加以应用。

目前该系统的客户端软件可在局域网内通过 **IP/TCP** 方式，实现对数据库的连接访问。