

Technical Manual – Nottinghamshire Traffic Infractions Database

Installation

The following assumes PHP version 5+ and a MySQL database are already configured.

MySQL

The script `sql_db_final.sql` can be used to build your own database. You can either run this in your database design tool or initiate it on the command line with:

```
mysql -h hostname -u user database < ~/pathgoeshere/sql_db_final.sql
```

Front-end Site Files

All site files (`.php` and `.css`) and the image directory under **InstallationFiles** need to be added together to your webserver root directory (p.ex in Apache this is under `~/var/www/html`).

The variable `$conn` in the file `"db.php"` must be modified to reflect your MySQL host, username, password, and database name e.g. –

```
$conn=mysqli_connect(yourhost, yourusername, yourpassword, yourdbname)
```

This only needs to be changed once in `"db.php"`. Note that the site will not work if you change the filename for any of the files.

The files included (to drop into the web root) are as follows:

InstallationFiles [DIR]:

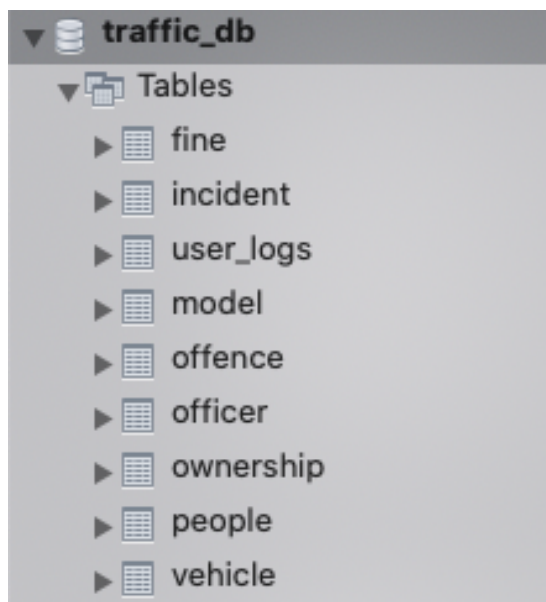
- `adminpanel.php`
- `db_style.css`
- `db.php`
- `favicon.ico`
- `index.php`
- `login.php`
- `logs.php`
- `lookup.php`
- `newentry.php`

newowner.php
reporting.php
image [DIR]:
 back.jpg
 header.png
 notts_crest.png

MySQL Database Overview

Database

The database contains the following tables:



Fine

This stores all of the fine information added by an administrator on the **adminpanel.php** page in columns:

fine_id(int, primary_key, not null, auto_increment=True),
fine_amount(int, not_null),
fine_points(int, not null),
incident_id(int, not null, foreign key references
incident(incident_id)on update CASCADE/on delete CASCADE)

Fine_id auto increments with each new row.

Fines are not created until an incident has already been created, so `incident_id` cannot be null as a fine must be associated with an existing incident.

Incident

This stores all of the incident information added on the **reporting.php** page in columns:

```
incident_id(int, primary key, not null, auto_increment=True),  
people_id(int, primary key, not null, foreign key references  
people(people_id) on update CASCADE/on delete CASCADE),  
vehicle_id(int, primary key, not null, foreign key references  
vehicle(vehicle_id) on update CASCADE/on delete CASCADE),  
incident_date(date, not_null), incident_report(varchar(500), not  
null),  
offence_id(int, not null, foreign key references offence(offence_id)  
on update RESTRICT/on delete RESTRICT)
```

{`incident_id`, `people_id`, `vehicle_id`} is a composite key.

`Incident_id` auto increments with each new row.

`People_id` and `vehicle_id` foreign keys cascade on deletion/update as both are required for an entry. `offence_id` is a foreign key but restricted as the entry must already be in the offence table.

User_logs

This stores all of the user logging information added whenever a query is made and viewed on the **logs.php** page in columns:

```
action_id(int, primary key, not null, auto_increment=True),  
username(varchar(50), not null),  
user_action(varchar(500), not null),  
action_time(timestamp(default=CURRENT_TIMESTAMP)), not null)
```

`user_logs` only stores queries which modify the database – not `SELECTs` – as this would likely slow the db if implemented at scale.

`action_id` provides a unique auto incremented identifier for each query made by a user.

`user_action` gives a full reproduction of the INSERT or UPDATE query made (whether successful or not) separated by `_` escape character(s).

`action_time` gives a MySQL timestamp of the time the query is made in the UTC time zone.

Model

This stores all of the visual model information for vehicles added on the **newentry.php** page in columns:

vehicle_model_id(*int, primary key, not null, auto_increment=True*),
vehicle_id(*int, primary key, not null, foreign key references vehicle(vehicle_id) on update CASCADE/on delete CASCADE*),
make(*varchar(50)*),
model(*varchar(50)*),
colour(*varchar(25)*)

{`vehicle_model_id`, `vehicle_id`} is a composite key.

`make`, `model`, and `colour` have been decomposed from the `vehicle_id` table to protect against update and delete anomalies.

Offence

This stores all of the offence information used to file reports and add fines on the **reporting.php** and **adminpanel.php** pages, respectively, in columns:

offence_id(*int, primary key, not null*),
offence_description(*varchar(128), not null*),
offence_maxfine(*int, not null*),
offence_maxpoints(*int, not null*)

The offence table provides a static table of offences to file incident reports with.

Officer

Unhashed in a normal table as cybersecurity not a requirement. This stores all of the login information for users in columns:

username(*varchar(50), primary key, not null*),

pass(*varchar(50), not null*)

Ownership

This associates vehicles (**vehicle_id**) with people (**people_id**) in columns:

vehicle(*int, primary key, not null, foreign key references vehicle(vehicle_id) on update CASCADE/on delete CASCADE*),
people_id(*int, foreign key references people(people_id) on update SET NULL/on delete SET NULL*)

One **vehicle** can have many **people_ids** associated with it, but the primary key ensures each vehicle has only one entry.

People

This stores all of the person information added on the **newentry.php** page in columns:

people_id(*int, primary key, not null, auto_increment=True*),
people_fname(*varchar(25), not null*),
people_lname(*varchar(25), not null*),
people_license(*varchar(50), unique=True, not null*),
people_address (*varchar(150), not null*)

Names have been segmented into **people_fname** and **people_lname** to allow for partial name query searches.

people_license must be unique (and is not null) to prevent duplicates from being entered into the system.

Vehicle

This stores all of the vehicle identifier information added on the **newentry.php** page in columns:

vehicle_id(*int, primary key, not null, auto_increment=True*),
people_plate(*varchar(7), primary key, not null*)

vehicle_plate must be unique (and is not null) to prevent duplicates from being entered into the system.