Dynamic Webpage: Driver Database

Credit: PROJECT C WEB DESIGN AND DATABASE MANAGEMENT

## 

Congratulations! You have been hired by the Alberta government to develop a new Alberta driver and vehicle database. On the front end, the government wants a GUI website where all the database information can be displayed. Your assignment is to create an interactive website that allows a user to create, view, and remove data stored in a database. The website must satisfy the following functional requirements:

A user of this dynamic webpage must be able to:

## Basic Requirements

1. View a list of registered People and all of their details (please see SQL Table Definitions)
2. View a list of registered Vehicles and all of their details (please see SQL Table Definitions)

## Intermediate Requirements

1. Create new People through the use of <form> and <input> elements on the page
2. Create new Vehicles through the use of <form> and <input> elements on the page
3. View a list of Vehicles that belong to a given Person
4. Search feature to search for a person by their first name or last name
5. Remove People from the database (remember to check if they have any vehicles owned first.)
6. Remove Vehicles from the database
7. Search for a Person by a set of fields that are NOT their database id

There are many possible interpretations of these requirements, and many more enhancements are possible. For example, users may want to update existing Person and Vehicle records. Or they may want to perform more complex searches, i.e. who are the drivers licensed to vehicles from 1990 to 2000? Be creative!

Not a requirement, but if you do want to use additional tables, present them to your teachers to receive formative feedback.

## SQL Table Definition Examples

* Person
  + id (Ex. 1; SMALLINT; PRIMARY KEY)
  + firstname (VARCHAR 30)
  + lastname (VARCHAR 30)
  + dateOfIssue (Date)
  + gender (Ex. F/M; CHAR 1)

mysql> CREATE TABLE person (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

firstname VARCHAR(30) NOT NULL,

lastname VARCHAR(30) NOT NULL,

dateOfIssue DATE NOT NULL,

gender CHAR(1) NOT NULL

);

mysql> describe person;

+-------------+-----------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-----------------+------+-----+---------+----------------+

| id | int(6) unsigned | NO | PRI | NULL | auto\_increment |

| firstname | varchar(30) | NO | | NULL | |

| lastname | varchar(30) | NO | | NULL | |

| dateOfIssue | date | NO | | NULL | |

| gender | char(1) | NO | | NULL | |

+-------------+-----------------+------+-----+---------+----------------+

mysql> INSERT INTO person VALUES ( NULL, 'Peter', 'Elliot', NOW(),'m');

mysql> SELECT \* FROM person;

+----+-----------+----------+-------------+--------+

| id | firstname | lastname | dateOfIssue | gender |

+----+-----------+----------+-------------+--------+

| 1 | Peter | Elliot | 2015-12-03 | m |

+----+-----------+----------+-------------+--------+

* vehicle
  + vin (Vehicle Identification number; Ex. JF1GD616X; VARCHAR 9; PRIMARY KEY)
  + maker (Ex. Mazda; VARCHAR 20)
  + type (Ex. Protege; VARCHAR 20)
  + year (Ex. 2015; SMALLINT)
  + primaryDriver (Ex. 1; SMALLINT; FOREIGN KEY)

CREATE TABLE vehicle(

vin VARCHAR(9) PRIMARY KEY,

maker VARCHAR(20) NOT NULL,

type VARCHAR(20) NOT NULL,

year SMALLINT DEFAULT 2015,

primaryDriver SMALLINT NOT NULL

);

mysql> describe vehicle;

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| vin | varchar(9) | NO | PRI | NULL | |

| maker | varchar(20) | NO | | NULL | |

| type | varchar(20) | NO | | NULL | |

| year | smallint(6) | YES | | 2015 | |

| primaryDriver | smallint(6) | NO | | NULL | |

+---------------+-------------+------+-----+---------+-------+

INSERT INTO vehicle VALUES('JF1GD616X','Chevrolet','Corvette',DEFAULT,1);

select \* from vehicle;

+-----------+-----------+----------+------+---------------+

| vin | maker | type | year | primaryDriver |

+-----------+-----------+----------+------+---------------+

| JF1GD616X | Chevrolet | Corvette | 2015 | 1 |

+-----------+-----------+----------+------+---------------+

1 row in set (0.00 sec)

## Other Resources

Adobe Color Wheel:

<https://color.adobe.com/create/color-wheel/>

***Dynamic Webpage Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

## Assessment Criteria

### Basic Feature Requirements (60%)

|  |  |
| --- | --- |
| **Item** | **This/these items are present**  **(30 points for each)** |
| The website includes a feature to view all the people in the database. |  |
| The website includes a feature to view all vehicles in the database. |  |
|  | **Total**  **/60** |

### Intermediate Feature Requirements (10%)

Choose 2 Features to implement

|  |  |
| --- | --- |
| **Item** | **This/these items are present**  **(5 marks each)** |
| The website includes a feature to create a new person through the use of <form> and <input> elements on the page |  |
| The website includes a feature to create a new vehicle through the use of <form> and <input> elements on the page |  |
| The website includes a feature to remove a person from the database. |  |
| The website includes a feature to remove a vehicle from the database. |  |
| The website includes a search feature to search for a person by a set of fields that is NOT their database ID |  |
| The website includes a feature to view which vehicles are owned by which people in the database. |  |
| The website includes a search feature to search for a person by their first name or last name |  |
|  | **Total**  **/10** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ITEM** | **Outstanding**  **3** | **Proficient**  **2** | **Requires improvement**  **1** | **Insufficient evidence**  **0** |
| **Do the HTML pages follow proper style conventions?**  **(x5; 15)**   * HTML tags are all well-formed (all tags are properly closed and encapsulated) * HTML is easy to read (proper indentation) * Comments are included outlining elements and attributes * Document type is included at top of document * Tag elements are all lower case * Empty HTML elements are properly closed * Attributes are contained in quotations * Alt text is provided for all images * Consistency is maintained throughout document | Checklist has been  consistently followed | Checklist has been followed but not  consistently | Checklist has been  followed inconsistently  with several  omissions/  errors |  |
| **Is CSS implemented follows proper style conventions?**  **(x3; 15%)**   * Style is defined in separate, linked style.css document * Semicolons included after every declaration * Indentation of all block content * Consistency is maintained throughout document * ID and class names are short, but descriptive | Checklist has been  consistently followed | Checklist has been followed but not  consistently | Checklist has been  followed inconsistently  with several  omissions/  errors |  |
| **Comments:** | | | | **Overall**  **Total**  **/30** |

### Overall Grade

|  |  |
| --- | --- |
| **Basic Requirements (60%)** | **/60** |
| **Intermediate Requirements**  **(10%)** | **/10** |
| **HTML and CSS Conventions**  **(40%)** | **/30** |
| **Total** | **/100** |
| **Comments** |  |