Sara Ray

DMV Database Design Project

Section 1 – Discovery

1.1 Identified entities and attributes

Vehicle
- VehicleID_ID
- VIN_ID
- VehicleType_Code
- ModelYear_Number
- Make_Name
- Model_Name
- PlateNumber_Code

Owner
- OwnerID_ID
- OwnerFullName_Name

Operator (licensed driver)
- OperatorID_ID
- OperatorFullName_Name
- DateOfBirth_Date
- HairColor_Name
- EyeColor_Name
- MustWearGlasses_Flag

Address
- AddressID_ID
- Street_Name
- City_Name
- State_Code
- ZipCode_Code

Registration
- RegistrationID_ID
- VehicleID_ID (FK)
- RegistrationDate_Date
- SalesPrice_Amount
- AssessedValue_Amount
- RegistrationFee_Amount

SmogTest
- SmogTestID_ID
- VehicleID_ID (FK)
- TestDate_Date
- Passed_Flag

DMVCenter
- CenterID_ID
- CenterName_Name
- CenterStreet_Name
- CenterCity_Name
- CenterState_Code

- CenterZip_Code

PoliceOfficer
- OfficerNumber_ID
- DepartmentName_Name
- Rank_Name (nullable)

Exam
- ExamID_ID
- OperatorID_ID (FK)
- CenterID_ID (FK)
- OfficerNumber_ID (FK)
- ExamDate_Date

License
- LicenseNumber_ID
- OperatorID_ID (FK, unique)
- IssueDate_Date
- ExpiryDate_Date

Accident
- AccidentID_ID
- AccidentDateTime_DateTime
- LocationDesc_Desc

Citation
- CitationID_ID
- CitationDate_Date
- CitationLocation_Desc

TrafficCodeSection
- SectionCode_Code
- SectionTitle_Name
- SectionDesc_Desc

AuthorizationCode (for NC, M, CF, CP)
- AuthCode_Code
- AuthCodeDesc_Desc

1.2 Identified relationships

Key relationships:

- **Owner – Owns – Vehicle**
  Each owner can own many vehicles; each vehicle can have many owners over its lifetime. Implemented via OwnerVehicle.

- **Owner – ResidesAt – Address**
  Each owner has exactly one address; each address belongs to exactly one owner (DMV duplicates address rows if multiple people share a physical location).

- **Operator – ResidesAt – Address**
  Similarly, each operator has exactly one address, and each address row belongs to one operator.

- **Operator – Takes – Exam**
  Each exam is taken by exactly one operator; an operator may attempt multiple exams over time.

- **PoliceOfficer – Administers – Exam**
  Each exam is administered by exactly one officer; each officer may administer many exams.

- **DMVCenter – Hosts – Exam**
  Each exam occurs at exactly one DMV center; each center hosts many exams.

- **Operator – Has – License**
  Every operator has exactly one license; every license belongs to exactly one operator (one-to-one).

- **License – Has – AuthorizationCode**
  A license may have one or more authorisation codes (NC, M, CF, CP). Implemented through LicenseAuthorization.

- **Accident – Involves – Vehicle**
  An accident may involve zero or many vehicles (to allow for accidents with no identifiable registered vehicle).

- **Accident – Involves – License**
  An accident may involve zero or many licensed drivers. Some accidents may involve unlicensed drivers.

- **Citation – Cites – Vehicle**
  A citation may cite zero or one vehicle.

- **Citation – Cites – License**
  A citation may cite zero or one licensed driver.

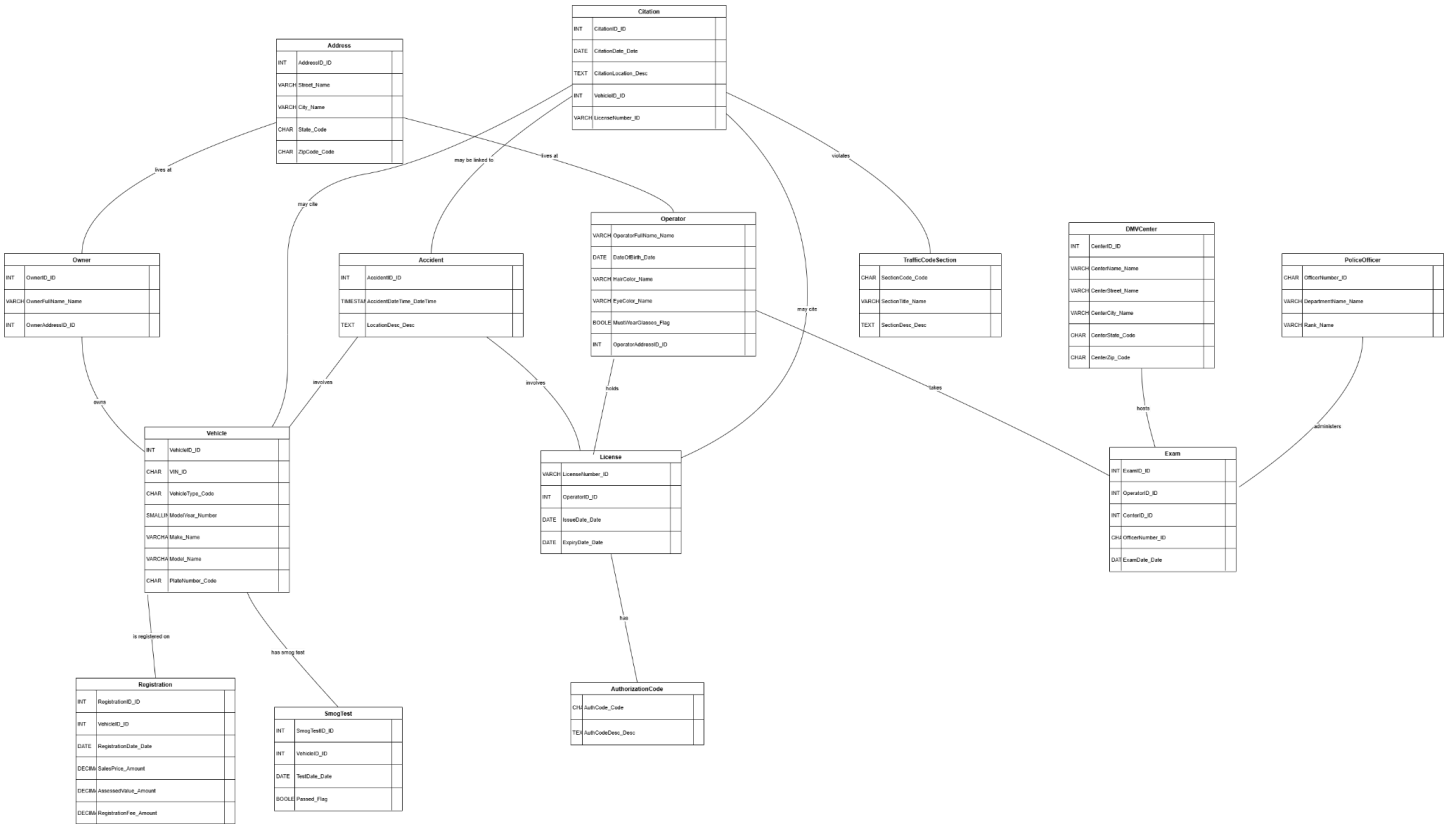
- **Citation – Causes – Accident**
  A citation may be associated with multiple accidents, but each accident is caused by at most one citation.

- **Citation – Violates – TrafficCodeSection**
  Each citation may violate one or more traffic code sections.

- **Vehicle – Has – Registration**
  Each vehicle has one registration per year; over time, a vehicle has many registrations.

- **Vehicle – Has – SmogTest**
  Vehicles may have smog tests according to the rules. Each test belongs to one vehicle.

1.3 Initial ER diagram



1.4 Explanation and assumptions

The entities I chose came directly from the project description. As I went through the requirements, it was clear that the DMV needed to track vehicles, owners, licensed drivers, addresses, exams, licenses, accidents, citations, and the traffic code sections behind those citations. I kept registrations and smog tests as their own tables because they happen multiple times over the life of a vehicle and have their own details.

I treated owners and operators as separate entities because someone can own a vehicle without being the one who drives it. I kept addresses in their own table, even though the DMV duplicates addresses when people share a location. This follows the “ResidesAt” structure described in the instructions.

I didn’t create extra tables for smog-test rules. Those rules can be enforced outside the schema. I also allowed some foreign keys to be optional, like citations that don’t have a matching vehicle or license, so the system can still record the event even without complete information.

Section 2 – Relationship and attribute cardinality

2.1 Relationship cardinalities

- **Owner – ResidesAt – Address**
    - Each Owner has exactly one Address (1 and only 1).
    - Each Address belongs to exactly one Owner (1 and only 1).
      When multiple people share a physical address, the data is duplicated instead of shared.
- **Owner – Owns – Vehicle (OwnerVehicle)**
    - One Owner can own many Vehicles (0..many).
    - One Vehicle can have many Owners (0..many).
      When a vehicle is junked, the Vehicle row is kept, but all OwnerVehicle rows may end, so the minimum on the Owner side is 0.
- **Accident – Involves – Vehicle (AccidentVehicle)**
    - One Accident can involve many Vehicles (0..many).
    - A Vehicle can appear in many Accidents (0..many).
      Minimum on the Vehicle side is 0 because an accident may involve a vehicle with no plate, or one that cannot be matched to the DMV record.
- **Accident – Involves – License (AccidentLicense)**
    - One Accident can involve many Licenses (0..many).
    - One License can be involved in many Accidents (0..many).
      Minimum on the License side is 0 because an accident may involve unlicensed drivers.
- **Citation – Cites – Vehicle**
    - One Citation may cite zero or one Vehicle (0..1).
    - A Vehicle may be cited by many Citations (0..many).
      Minimum on the Vehicle side is 0 to allow citations with no identifiable registered vehicle.
- **Citation – Cites – License**
    - One Citation may cite zero or one License (0..1).
    - A License may be cited by many Citations (0..many).
      Minimum on the Licence side is 0 to allow citations issued to unlicensed drivers.
- **Citation – Causes – Accident (CitationAccident)**
    - One Citation may be associated with many Accidents (0..many).

    - Each Accident is caused by at most one Citation (0..1).

- **Operator – Has – License**
    - One Operator has exactly one License (1..1).
    - One License belongs to exactly one Operator (1..1).
- **Operator – ResidesAt – Address**
  Same pattern as Owner–ResidesAt–Address: one-to-one in practice.
- **DMVCenter – Hosts – Exam**
    - One DMVCenter hosts many Exams (0..many).
    - Each Exam occurs at one DMVCenter (1).
- **PoliceOfficer – Administers – Exam**
    - One PoliceOfficer administers many Exams (0..many).
    - Each Exam is administered by one PoliceOfficer (1).
- **Operator – Takes – Exam**
    - One Operator may take many Exams (0..many attempts).
    - Each Exam belongs to exactly one Operator (1).
- **License – Has – AuthorizationCode (LicenseAuthorization)**
    - One License has one or more AuthorizationCodes (1..many).
    - Each AuthorizationCode may appear on many licenses (0..many).
- **Vehicle – Has – Registration**
    - One Vehicle has many Registrations over time (0..many).
    - Each Registration belongs to one Vehicle (1).
- **Vehicle – Has – SmogTest**
    - One Vehicle may have many SmogTests (0..many).
    - Each SmogTest belongs to one Vehicle (1).

2.2 Attribute cardinality

Required

- Vehicle.VIN_ID
- Vehicle.ModelYear_Number
- Owner.OwnerFullName_Name
- License.LicenseNumber_ID
- Exam.ExamDate_Date

Optional

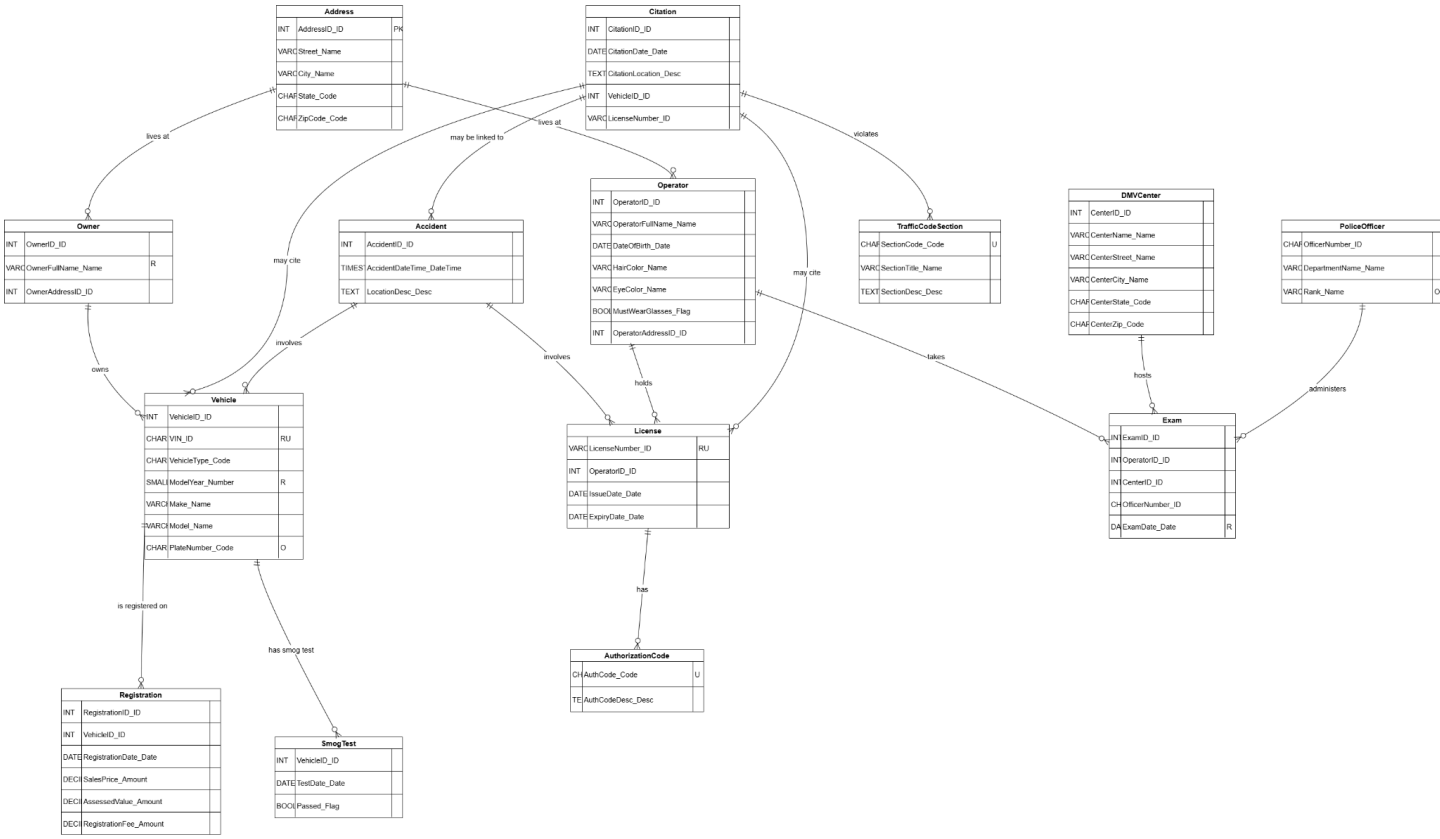
- PoliceOfficer.Rank_Name
- Vehicle.PlateNumber_Code

Unique

- Vehicle.VIN_ID
- License.LicenseNumber_ID
- TrafficCodeSection.SectionCode_Code
- AuthorizationCode.AuthCode_Code

Plural

- Exam part scores.
- Authorisation codes on a licence.
- Vehicle owners.
- Vehicles and driver licences involved in accidents.
- Accidents associated with a citation.
- Traffic code sections associated with a citation.



Section 3 – Implementing entities (primary keys)

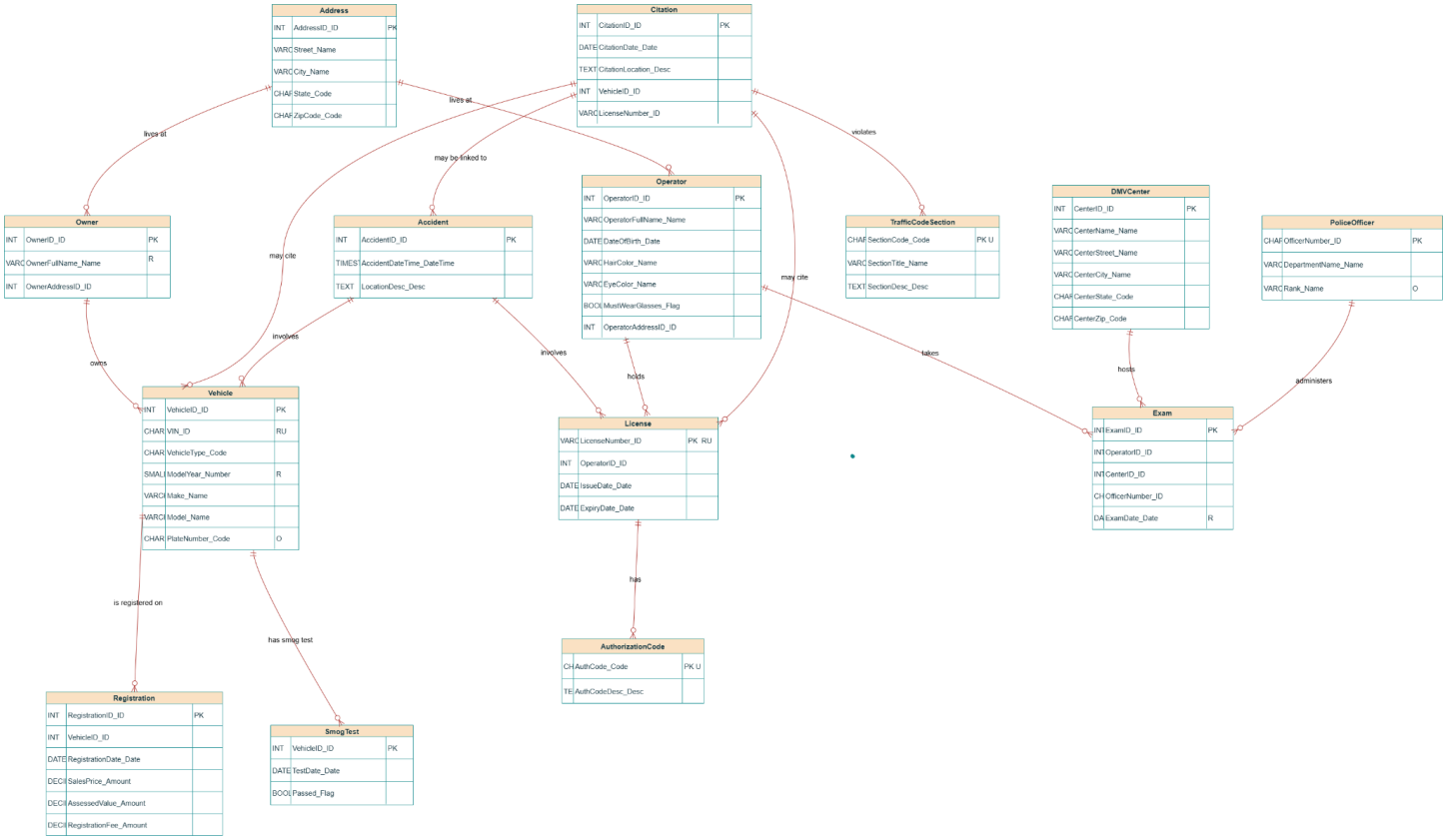
3.1 Primary key selection table

The table below summarises primary key choices. “Yes/No” indicates whether the property holds for the chosen key.

Table	Primary key	Singular	Required	Unique	Stable	Simple	Meaningless	Artificial
Vehicle	VehicleID_ID	yes	yes	yes	yes	yes	yes	yes
Owner	OwnerID_ID	yes	yes	yes	yes	yes	yes	yes
Operator	OperatorID_ID	yes	yes	yes	yes	yes	yes	yes
Address	AddressID_ID	yes	yes	yes	yes	yes	yes	yes
Registration	RegistrationID_ID	yes	yes	yes	yes	yes	yes	yes
SmogTest	SmogTestID_ID	yes	yes	yes	yes	yes	yes	yes

DMVCenter	CenterID_ID	yes	yes	yes	yes	yes	yes	yes
PoliceOfficer	OfficerNumber_ID	yes	yes	yes	yes	yes	no	no
Exam	ExamID_ID	yes	yes	yes	yes	yes	yes	yes
License	LicenseNumber_ID	yes	yes	yes	yes	yes	no	no
Accident	AccidentID_ID	yes	yes	yes	yes	yes	yes	yes
Citation	CitationID_ID	yes	yes	yes	yes	yes	yes	yes
TrafficCodeSection	SectionCode_Code	yes	yes	yes	yes	yes	no	no
AuthorizationCode	AuthCode_Code	yes	yes	yes	yes	yes	no	no
OwnerVehicle	(OwnerID_ID, VehicleID_ID, OwnershipStartDate_Date)	yes	yes	yes	yes	no	yes	no
ExamPartScore	(ExamID_ID, PartCode_Code)	yes	yes	yes	yes	no	yes	no
LicenseAuthorization	(LicenseNumber_ID, AuthCode_Code)	yes	yes	yes	yes	no	yes	no
AccidentVehicle	(AccidentID_ID, VehicleID_ID)	yes	yes	yes	yes	no	yes	no
AccidentLicense	(AccidentID_ID, LicenseNumber_ID)	yes	yes	yes	yes	no	yes	no
CitationAccident	(CitationID_ID, AccidentID_ID)	yes	yes	yes	yes	no	yes	no
CitationTrafficSection	(CitationID_ID, SectionCode_Code)	yes	yes	yes	yes	no	yes	no

I used simple numeric IDs for most of the main tables so that the foreign keys stay small, consistent, and easy to work with. These IDs don't carry any real-world meaning and are set to auto-increment in SQL. I only used natural IDs when they are already short and stable, like OfficerNumber_ID. For the linking tables, I used composite keys so each record in the relationship is unique without having to create an extra ID.



Section 4 – Implementing relationships (foreign keys)

4.1 Many-to-one relationships

- **Exam – PoliceOfficer**
    - Exam.OfficerNumber_ID → PoliceOfficer.OfficerNumber_ID
- **Exam – DMVCenter**
    - Exam.CenterID_ID → DMVCenter.CenterID_ID
- **Exam – Operator**
    - Exam.OperatorID_ID → Operator.OperatorID_ID
- **Registration – Vehicle**
    - Registration.VehicleID_ID → Vehicle.VehicleID_ID
- **SmogTest – Vehicle**
    - SmogTest.VehicleID_ID → Vehicle.VehicleID_ID
- **License – Operator**
    - License.OperatorID_ID → Operator.OperatorID_ID

4.2 One-to-one relationship

- License.OperatorID_ID is both a foreign key and UNIQUE.
- This ensures each operator has at most one licence, and each licence is linked to exactly one operator.

4.3 Many-to-many relationships

- **Owner – Owns – Vehicle**
    - OwnerVehicle(OwnerID_ID FK, VehicleID_ID FK, OwnershipStartDate_Date, OwnershipEndDate_Date)
    - PK = (OwnerID_ID, VehicleID_ID, OwnershipStartDate_Date).
- **Accident – Involves – Vehicle**

- AccidentVehicle(AccidentID_ID FK, VehicleID_ID FK)
- PK = (AccidentID_ID, VehicleID_ID). VehicleID_ID is nullable to allow accidents with no identifiable registered vehicle.

- **Accident – Involves – License**
  - AccidentLicense(AccidentID_ID FK, LicenseNumber_ID FK)
  - PK = (AccidentID_ID, LicenseNumber_ID). LicenseNumber_ID nullable to allow accidents with unlicensed drivers.

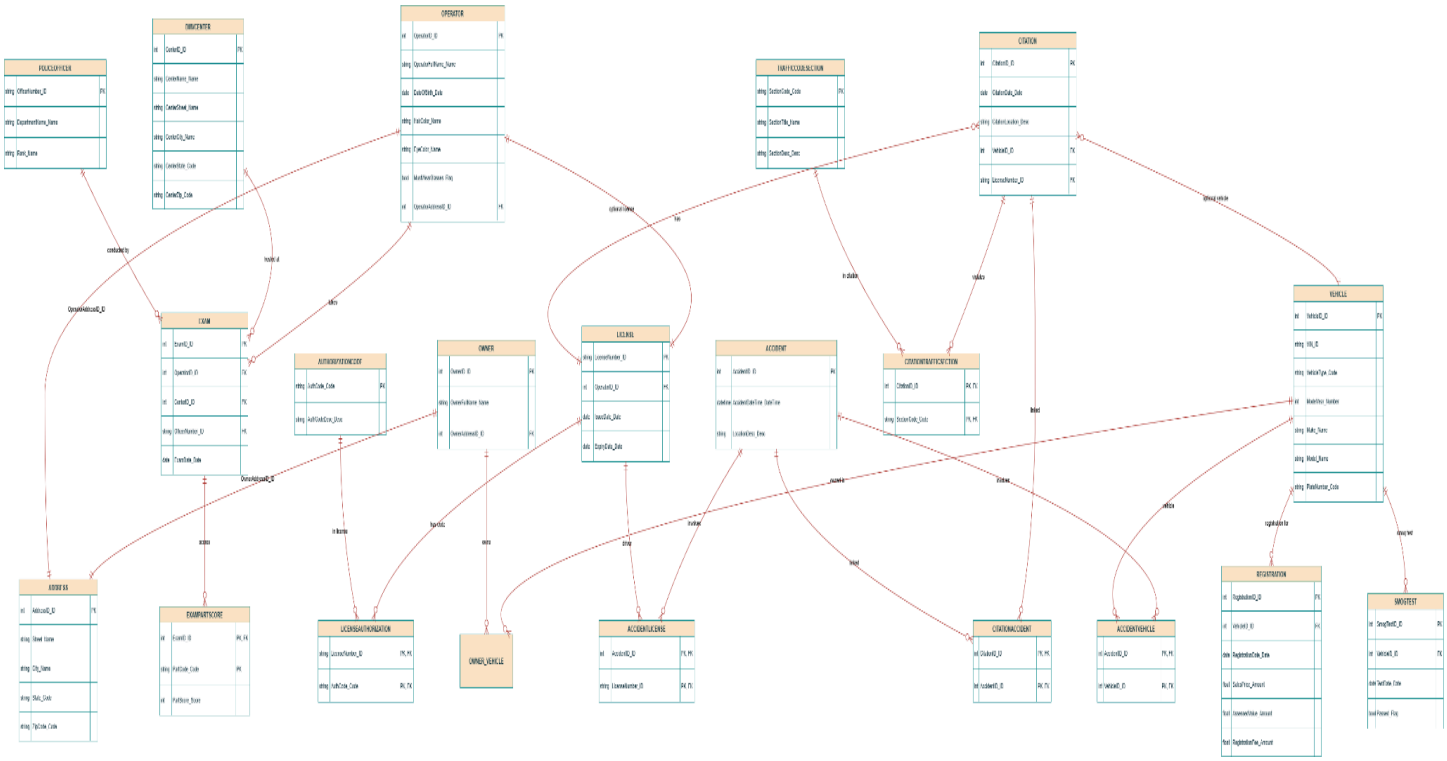
- **Citation – Causes – Accident**
  - CitationAccident(CitationID_ID FK, AccidentID_ID FK)
  - PK = (CitationID_ID, AccidentID_ID). A unique constraint on AccidentID_ID enforces the “at most one citation per accident” rule.

- **Citation – Violates – TrafficCodeSection**
  - CitationTrafficSection(CitationID_ID FK, SectionCode_Code FK)
  - PK = (CitationID_ID, SectionCode_Code).

- **License – Has – AuthorizationCode**
  - LicenseAuthorization(LicenseNumber_ID FK, AuthCode_Code FK)
  - PK = (LicenseNumber_ID, AuthCode_Code).

4.4 Foreign key actions

Required relationships use NOT NULL foreign keys, so the related row must exist (for example, each exam must have an operator, officer, and DMV center). Optional relationships allow NULL in the foreign key (for example, a citation may not have a vehicle or license). In the associative tables, foreign keys make sure each link points to valid parent rows, which keeps the data consistent.



Section 5 – Implementing plural attributes

5.1 List of plural attributes and weak tables

Some entities in the DMV system can have multiple related values. Each plural attribute is implemented using its own weak table:

- Exam part scores:
  Each exam has several individual part scores.
  ExamPartScore(ExamID_ID, PartCode_Code, PartScore_Score)

- License authorization codes:
  A license may include more than one authorization code.
  LicenseAuthorization(LicenseNumber_ID, AuthCode_Code)

- Multiple owners per vehicle:
  A vehicle can have different owners over time.
  OwnerVehicle(OwnerID_ID, VehicleID_ID, OwnershipStartDate_Date, OwnershipEndDate_Date)

- Vehicles and licenses involved in accidents:
  An accident may involve multiple vehicles and multiple drivers.
  AccidentVehicle(AccidentID_ID, VehicleID_ID)
  AccidentLicense(AccidentID_ID, LicenseNumber_ID)

- Citations linked to accidents:
  A single citation may be connected to more than one accident.
  CitationAccident(CitationID_ID, AccidentID_ID)

- Traffic code sections violated on a citation:
  A citation may list multiple sections of the traffic code.
  CitationTrafficSection(CitationID_ID, SectionCode_Code)

5.2 ERD updates and reasoning

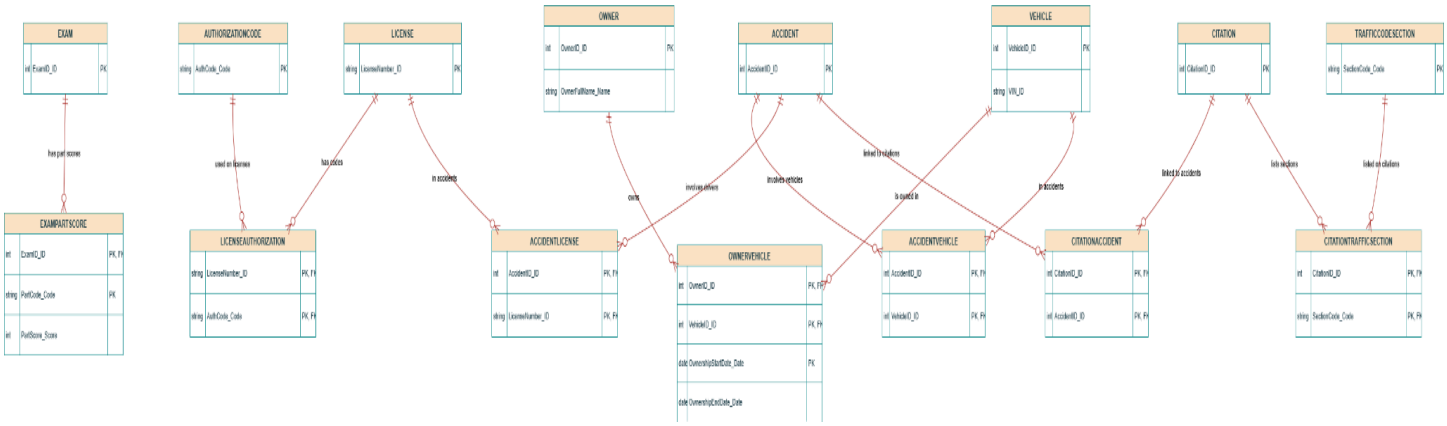
For each plural attribute, a separate weak table is used so that multiple values can be stored cleanly.
These tables include:

- A foreign key back to the main entity

- An attribute or code representing the repeated value

- A composite primary key to make each combination unique

This keeps the design organized and correctly handles all one-to-many and many-to-many situations.



Section 6 – SQL Script

```
CREATE TABLE Vehicle (
    VehicleID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    VIN_ID CHAR(17) NOT NULL UNIQUE,
    VehicleType_Code CHAR(10) NOT NULL,
    ModelYear_Number SMALLINT UNSIGNED NOT NULL,
    Make_Name VARCHAR(50) NOT NULL,
    Model_Name VARCHAR(50) NOT NULL,
    PlateNumber_Code CHAR(10) NULL
);

CREATE TABLE Owner (
    OwnerID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    OwnerFullName_Name VARCHAR(100) NOT NULL
);

CREATE TABLE Operator (
    OperatorID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    OperatorFullName_Name VARCHAR(100) NOT NULL,
    DateOfBirth_Date DATE NOT NULL,
    HairColor_Name VARCHAR(30) NOT NULL,
    EyeColor_Name VARCHAR(30) NOT NULL,
    MustWearGlasses_Flag BOOLEAN NOT NULL
);
```

```
CREATE TABLE Address (
   AddressID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   Street_Name VARCHAR(100) NOT NULL,
   City_Name VARCHAR(50) NOT NULL,
   State_Code CHAR(2) NOT NULL,
   ZipCode_Code CHAR(10) NOT NULL
);

ALTER TABLE Owner
   ADD COLUMN OwnerAddressID_ID INT UNSIGNED NOT NULL,
   ADD CONSTRAINT fk_Owner_Address
      FOREIGN KEY (OwnerAddressID_ID)
      REFERENCES Address (AddressID_ID);

ALTER TABLE Operator
   ADD COLUMN OperatorAddressID_ID INT UNSIGNED NOT NULL,
   ADD CONSTRAINT fk_Operator_Address
      FOREIGN KEY (OperatorAddressID_ID)
      REFERENCES Address (AddressID_ID);

CREATE TABLE Registration (
   RegistrationID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   VehicleID_ID INT UNSIGNED NOT NULL,
   RegistrationDate_Date DATE NOT NULL,
   SalesPrice_Amount DECIMAL(11,2) NOT NULL,
   AssessedValue_Amount DECIMAL(11,2) NOT NULL,
   RegistrationFee_Amount DECIMAL(11,2) NOT NULL,
   CONSTRAINT fk_Reg_Vehicle
      FOREIGN KEY (VehicleID_ID) REFERENCES Vehicle (VehicleID_ID)
);

CREATE TABLE SmogTest (
   SmogTestID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   VehicleID_ID INT UNSIGNED NOT NULL,
   TestDate_Date DATE NOT NULL,
   Passed_Flag BOOLEAN NOT NULL,
   CONSTRAINT fk_Smog_Vehicle
      FOREIGN KEY (VehicleID_ID) REFERENCES Vehicle (VehicleID_ID)
);

CREATE TABLE DMVCenter (
   CenterID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   CenterName_Name VARCHAR(100) NOT NULL,
   CenterStreet_Name VARCHAR(100) NOT NULL,
   CenterCity_Name VARCHAR(50) NOT NULL,
   CenterState_Code CHAR(2) NOT NULL,
   CenterZip_Code CHAR(10) NOT NULL);
```

```
CREATE TABLE PoliceOfficer (
    OfficerNumber_ID CHAR(10) PRIMARY KEY,
    DepartmentName_Name VARCHAR(100) NOT NULL,
    Rank_Name VARCHAR(50)
);

CREATE TABLE License (
    LicenseNumber_ID CHAR(12) PRIMARY KEY,
    OperatorID_ID INT UNSIGNED NOT NULL UNIQUE,
    IssueDate_Date DATE NOT NULL,
    ExpiryDate_Date DATE NOT NULL,
    CONSTRAINT fk_License_Operator
        FOREIGN KEY (OperatorID_ID) REFERENCES Operator (OperatorID_ID)
);

CREATE TABLE Exam (
    ExamID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    OperatorID_ID INT UNSIGNED NOT NULL,
    CenterID_ID INT UNSIGNED NOT NULL,
    OfficerNumber_ID CHAR(10) NOT NULL,
    ExamDate_Date DATE NOT NULL,
    CONSTRAINT fk_Exam_Operator
        FOREIGN KEY (OperatorID_ID) REFERENCES Operator (OperatorID_ID),
    CONSTRAINT fk_Exam_Center
        FOREIGN KEY (CenterID_ID) REFERENCES DMVCenter (CenterID_ID),
    CONSTRAINT fk_Exam_Officer
        FOREIGN KEY (OfficerNumber_ID) REFERENCES PoliceOfficer (OfficerNumber_ID)
);

CREATE TABLE ExamPartScore (
    ExamID_ID INT UNSIGNED NOT NULL,
    PartCode_Code CHAR(10) NOT NULL,
    PartScore_Score TINYINT UNSIGNED NOT NULL,
    PRIMARY KEY (ExamID_ID, PartCode_Code),
    CONSTRAINT fk_ExamPart_Exam
        FOREIGN KEY (ExamID_ID) REFERENCES Exam (ExamID_ID)
);

CREATE TABLE TrafficCodeSection (
    SectionCode_Code CHAR(10) PRIMARY KEY,
    SectionTitle_Name VARCHAR(100) NOT NULL,
    SectionDesc_Desc TEXT NOT NULL
);
```

```
CREATE TABLE AuthorizationCode (
   AuthCode_Code CHAR(2) PRIMARY KEY,
   AuthCodeDesc_Desc TEXT NOT NULL
);

CREATE TABLE LicenseAuthorization (
   LicenseNumber_ID CHAR(12) NOT NULL,
   AuthCode_Code CHAR(2) NOT NULL,
   PRIMARY KEY (LicenseNumber_ID, AuthCode_Code),
   CONSTRAINT fk_LAuth_License
      FOREIGN KEY (LicenseNumber_ID) REFERENCES License (LicenseNumber_ID),
   CONSTRAINT fk_LAuth_AuthCode
      FOREIGN KEY (AuthCode_Code) REFERENCES AuthorizationCode (AuthCode_Code)
);

CREATE TABLE Accident (
   AccidentID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   AccidentDateTime_DateTime DATETIME NOT NULL,
   LocationDesc_Desc TEXT NOT NULL
);

CREATE TABLE Citation (
   CitationID_ID INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   CitationDate_Date DATE NOT NULL,
   CitationLocation_Desc TEXT NOT NULL,
   VehicleID_ID INT UNSIGNED NULL,
   LicenseNumber_ID CHAR(12) NULL,
   CONSTRAINT fk_Citation_Vehicle
      FOREIGN KEY (VehicleID_ID) REFERENCES Vehicle (VehicleID_ID),
   CONSTRAINT fk_Citation_License
      FOREIGN KEY (LicenseNumber_ID) REFERENCES License (LicenseNumber_ID)
);

CREATE TABLE OwnerVehicle (
   OwnerID_ID INT UNSIGNED NOT NULL,
   VehicleID_ID INT UNSIGNED NOT NULL,
   OwnershipStartDate_Date DATE NOT NULL,
   OwnershipEndDate_Date DATE NULL,
   PRIMARY KEY (OwnerID_ID, VehicleID_ID, OwnershipStartDate_Date),
   CONSTRAINT fk_OV_Owner
      FOREIGN KEY (OwnerID_ID) REFERENCES Owner (OwnerID_ID),
   CONSTRAINT fk_OV_Vehicle
      FOREIGN KEY (VehicleID_ID) REFERENCES Vehicle (VehicleID_ID)
);
```

```
CREATE TABLE AccidentVehicle (
   AccidentID_ID INT UNSIGNED NOT NULL,
   VehicleID_ID INT UNSIGNED NULL,
   PRIMARY KEY (AccidentID_ID, VehicleID_ID),
   CONSTRAINT fk_AV_Accident
      FOREIGN KEY (AccidentID_ID) REFERENCES Accident (AccidentID_ID),
   CONSTRAINT fk_AV_Vehicle
      FOREIGN KEY (VehicleID_ID) REFERENCES Vehicle (VehicleID_ID)
);

CREATE TABLE AccidentLicense (
   AccidentID_ID INT UNSIGNED NOT NULL,
   LicenseNumber_ID CHAR(12) NULL,
   PRIMARY KEY (AccidentID_ID, LicenseNumber_ID),
   CONSTRAINT fk_AL_Accident
      FOREIGN KEY (AccidentID_ID) REFERENCES Accident (AccidentID_ID),
   CONSTRAINT fk_AL_License
      FOREIGN KEY (LicenseNumber_ID) REFERENCES License (LicenseNumber_ID)
);

CREATE TABLE CitationAccident (
   CitationID_ID INT UNSIGNED NOT NULL,
   AccidentID_ID INT UNSIGNED NOT NULL,
   PRIMARY KEY (CitationID_ID, AccidentID_ID),
   CONSTRAINT fk_CA_Citation
      FOREIGN KEY (CitationID_ID) REFERENCES Citation (CitationID_ID),
   CONSTRAINT fk_CA_Accident
      FOREIGN KEY (AccidentID_ID) REFERENCES Accident (AccidentID_ID)
);

CREATE TABLE CitationTrafficSection (
   CitationID_ID INT UNSIGNED NOT NULL,
   SectionCode_Code CHAR(10) NOT NULL,
   PRIMARY KEY (CitationID_ID, SectionCode_Code),
   CONSTRAINT fk_CTS_Citation
      FOREIGN KEY (CitationID_ID) REFERENCES Citation (CitationID_ID),
   CONSTRAINT fk_CTS_Section
      FOREIGN KEY (SectionCode_Code) REFERENCES TrafficCodeSection (SectionCode_Code)
);
```

Bonus

For the bonus part of the project, I changed the design so that Owner and Operator are both treated as subtypes of a new supertype called Person. Instead of storing someone's name and address in multiple tables, each person now has one PersonNumber, and all the shared information goes into the Person table.

Once I added the Person supertype, I updated the SQL so that Owner and Operator use PersonNumber as their primary key and each one points back to the Person table with a one-to-one relationship. I also updated the Address table so addresses are connected directly to Person, instead of being linked separately to owners and operators. Because of this, I didn't need the earlier ALTER TABLE statements anymore.

After that, I updated the tables that rely on owners or operators. For example, License and Exam now reference the Operator subtype using PersonNumber, and OwnerVehicle uses PersonNumber to connect each person to the vehicles they own.

SQL Script

```
CREATE TABLE Person (
   PersonNumber_ID INT AUTO_INCREMENT PRIMARY KEY,
   FullName_Name   VARCHAR(100) NOT NULL
);

CREATE TABLE Vehicle (
   VehicleID_ID     INT AUTO_INCREMENT PRIMARY KEY,
   VIN_ID           CHAR(17) NOT NULL,
   VehicleType_Code CHAR(10) NOT NULL,
   ModelYear_Number SMALLINT NOT NULL,
   Make_Name        VARCHAR(50) NOT NULL,
   Model_Name       VARCHAR(50) NOT NULL,
   PlateNumber_Code CHAR(10)
);

CREATE TABLE Owner (
   PersonNumber_ID INT PRIMARY KEY,
   FOREIGN KEY (PersonNumber_ID) REFERENCES Person(PersonNumber_ID)
);

CREATE TABLE Operator (
   PersonNumber_ID      INT PRIMARY KEY,
   DateOfBirth_Date     DATE NOT NULL,
   HairColor_Name       VARCHAR(30) NOT NULL,
   EyeColor_Name        VARCHAR(30) NOT NULL,
   MustWearGlasses_Flag BOOLEAN NOT NULL,
   FOREIGN KEY (PersonNumber_ID) REFERENCES Person(PersonNumber_ID)
);

CREATE TABLE Address (
   AddressID_ID     INT AUTO_INCREMENT PRIMARY KEY,
   PersonNumber_ID  INT NOT NULL,
   Street_Name      VARCHAR(100) NOT NULL,
```

```
    City_Name       VARCHAR(50) NOT NULL,
    State_Code      CHAR(2) NOT NULL,
    ZipCode_Code    CHAR(10) NOT NULL,
    FOREIGN KEY (PersonNumber_ID) REFERENCES Person(PersonNumber_ID)
);

CREATE TABLE License (
    LicenseNumber_ID CHAR(12) PRIMARY KEY,
    PersonNumber_ID  INT NOT NULL,
    IssueDate_Date   DATE NOT NULL,
    ExpiryDate_Date  DATE NOT NULL,
    FOREIGN KEY (PersonNumber_ID) REFERENCES Operator(PersonNumber_ID)
);
```