1. Exécuter le script SQL InitialCreate.sql sur master
2. Exécuter le script SQL V1\_0\_\_CreationTables.sql sur DronesDatabase
3. Exécuter le script SQL V1\_3\_\_ChiffrementDeNomDeFamille.sql sur DronesDatabase
4. Exécuter le script SQL V1\_1\_\_Insertion des données.sql sur DronesDatabase
5. Exécuter le script SQL V1\_2\_\_CreationVue.sql sur DronesDatabase
6. Exécuter le script SQL V1\_4\_\_RevenusPourUnModele.sql sur DronesDatabase
7. Exécuter le script SQL V1\_6\_\_Performance.sql sur DronesDatabase



Voici le header du projet.

Voici comment accéder aux différentes étapes du projet.

* **V1\_2\_\_CreationVue** : Création d’une vue SQL complexe. Modifications dans le projet pour afficher le résultat de la vue.
  + On peut y accéder à partir du bouton **V1\_2\_\_VueDétailléeDrones** du header.
  + Dans la vue on y voit le DroneId, NumSerie, NomModele, Vitesse, Prix, DateSortie de chaque drone en se basant sur la vue SQL Magasins.vw\_Drones de la base de données.
  + Action en question dans DronesController
  + A screen shot of a computer code

    Description automatically generated
* **V1\_3\_\_ChiffrementDe… :** Création d’une procédure stockée et autre, pour le chiffrement/déchiffrement d’une donnée sensible. Modifications dans le projet pour afficher cette donnée sensible.
  + On peut y accéder à partir du bouton **V1\_3\_\_ChiffrementDeNomDeFamille** du header.
  + Dans la vue on y voit la liste des clients avec leur nom de famille chiffré. On y retrouve aussi les boutons **Chiffrer les noms de famille des clients** et **Déchiffrer les noms de famille des clients** pour chiffrer/déchiffrer les noms de famille de tous les clients.
  + Cliquer sur **Details** d’un client amène à la vue Details pour ce client. On y retrouve les boutons **Chiffrer le nom de famille** et **Déchiffrer le nom de famille** pour chiffrer/déchiffrer le nom de famille du client courant.
  + Actions en question dans ClientsController

A screen shot of a computer program

Description automatically generated

* **V1\_4\_\_ProcedureStockeeMasterDetail :** Création d’une procédure stockée retournant des valeurs, avec paramètres. Modifications dans le projet pour avoir une vue EF dans laquelle on pourra entrer des données (Master) qui seront ensuite utilisées pour appeler la procédure stockée et afficher le résultat (Détail).
  + On peut y accéder à partir de du bouton **V1\_4\_\_RevenusModeles** du header.
  + Dans la vue on y voit la liste des modèles de drones. Pour chaque modèle on y retrouve le bouton suivant : **EXEC V\_1.4** qui va exécuter la procédure Magasins.usp\_RevenusPourUnModele de la base de données. Cette procédure va exécuter la fonction Magasins.ufn\_RevenusPourUnModele en lui envoyant l’id du modèle sélectionné et retournera les revenus générés.
  + Finalement, la vue qui va être affichée va présenter ce que la procédure a retourné soit les revenus générés par les ventes de ce modèle.
  + Actions en question dans ModelesController

A screen shot of a computer program

Description automatically generated

* **V1\_5\_\_TraitementDesImages :** Il y aura une migration qui fera le traitement des images, tel que vu dans le cours. Modifications dans le projet pour avoir l’upload de l’image et l’affichage de l’image quelque part.
  + On peut y accéder à partir de du bouton **V1\_5\_\_Images** du header.
  + Dans la vue on y voit la liste des modèles de drones. Pour chaque modèle on y retrouve les boutons suivants :
    - **Details Modele Image** : va rediriger vers la vue DetailModeleImage. Cette dernière affiche l’image du modèle sélectionné ainsi que les détails de l’image du modèle en question soit son nom et son id.
    - **Ajouter une image** : va rediriger vers la vue AjoutImagePourUnModele. Cette dernière va permettre d’uploader une image dans la base de données en lui spécifiant un nom et en l’assignant au modèle de drone qui a été sélectionné.
  + Actions en question dans ModelesController

A computer screen shot of a program code

Description automatically generated

* **V1\_6\_\_Performance :** Il y aura une migration qui créera des index, au moins 2, pour améliorer la performance de votre projet, index qui seront utiles compte tenu des recherches que vous allez faire dans les tables.
  + On peut y accéder à partir de du bouton V1\_6\_\_Performance du header.
  + Dans la vue on y voit la liste des adresses de manière paginée, de façon à rendre cela plus efficace lors des requêtes au serveur. On y retrouve aussi les filtres *Filtrer par ville* et *Filtrer par province* qui permettront de sélectionner une ville et/ou une province. Une fois nos filtres sélectionnés, on peut cliquer sur le bouton **Filtrer** qui permettra d’afficher uniquement les adresses qui correspondront aux filtres.
  + Action en question dans AdressesController

A screenshot of a computer program

Description automatically generated