**R16 partie 2 (suite labo R15)**

Révision de LINQ

Dans le cours Prog Web orientées services, il n’y aura pas vraiment de manipulations complexes pour « Retrieve » des données.

Ce cours est donc une bonne opportunité de faire des agrégations, des jointures, etc. avec les données. Et en même temps, on va voir comment faire des vues et les utiliser facilite tellement la vie!

Pour les exercices qui suivent, vous aurez à intégrer plusieurs **Views** fournies à votre projet et à ajouter des liens à la vue Index.cshtml, qui sont donnés dans le fichier **Liens Ds Index.txt**. Il ne vous restera qu’à compléter des actions dans le contrôleur artiste pour préparer les données à afficher dans les vues fournies. Vous pouvez retrouver le squelette des 6 actions à faire dans le fichier **Actions dans le Controlleur.txt**.

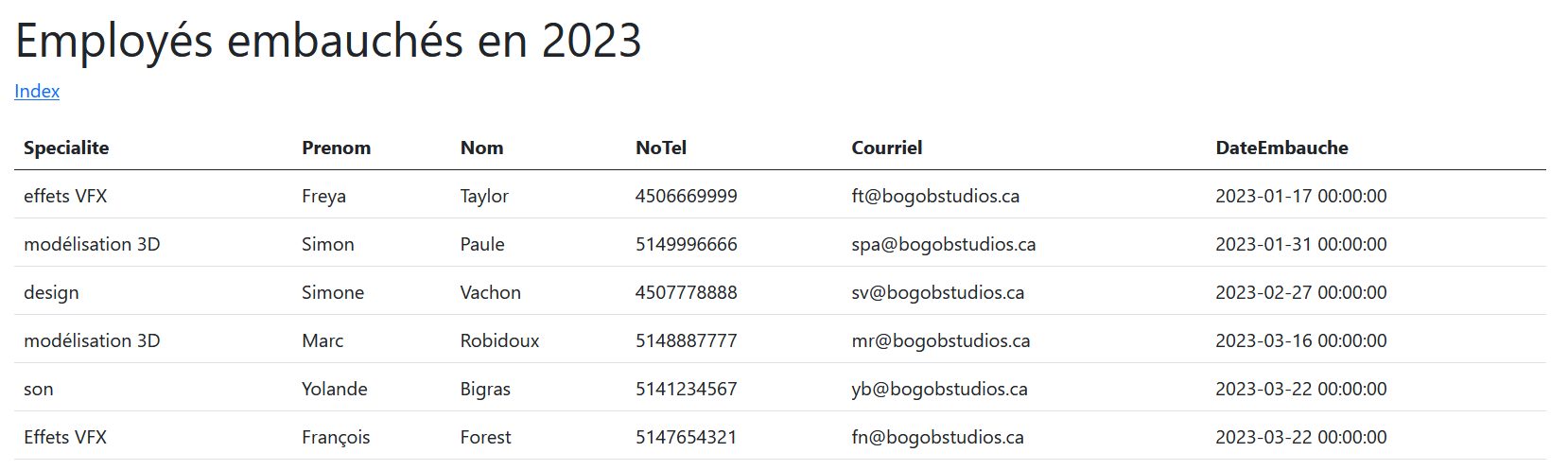
N’hésitez pas à installer le package **EntityFrameworkCore.Proxies** si vous voulez vous simplifier la vie pour certaines requêtes qui nécessitent les deux tables.

Une image contenant texte, capture d’écran, conception

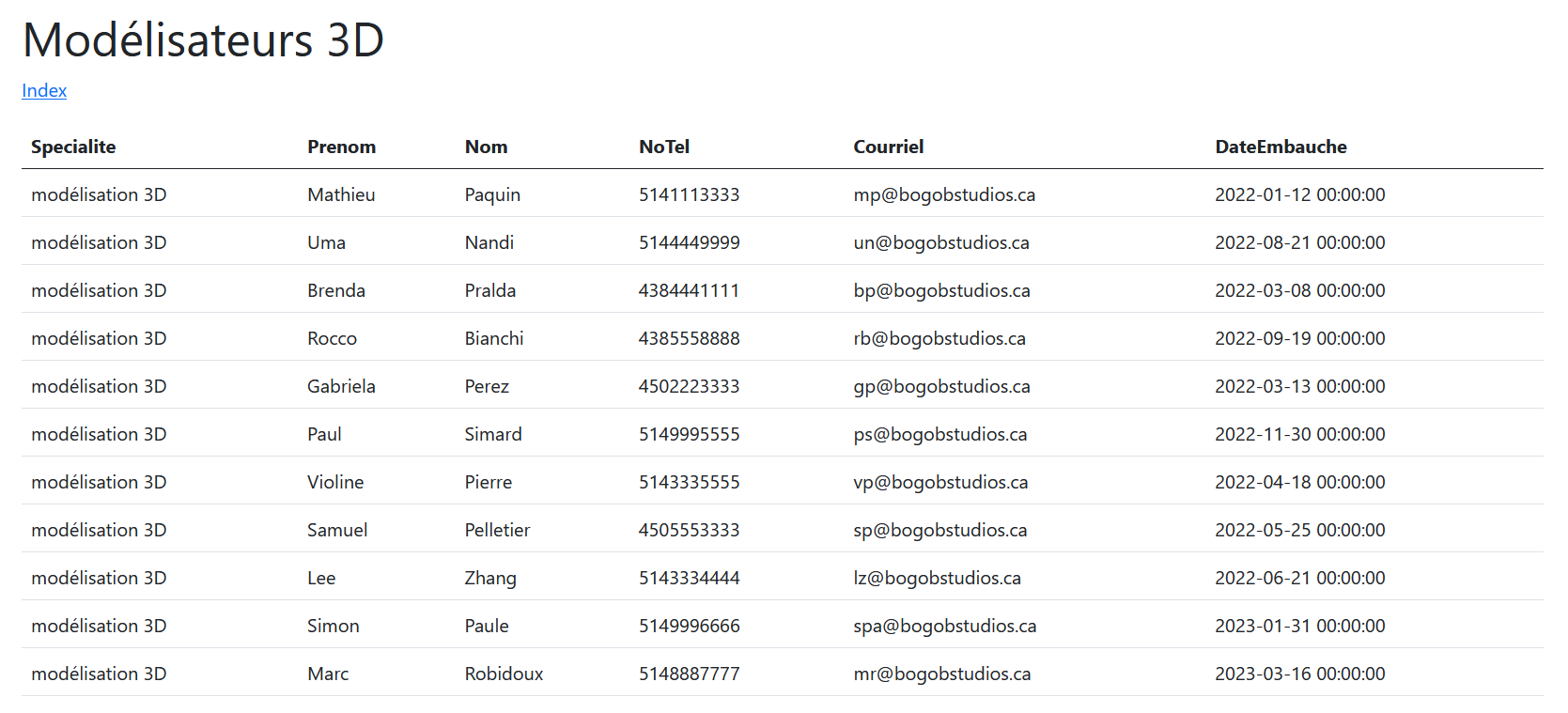
Description générée automatiquement

**Étape 3 - C'est plus rapide en SQL**

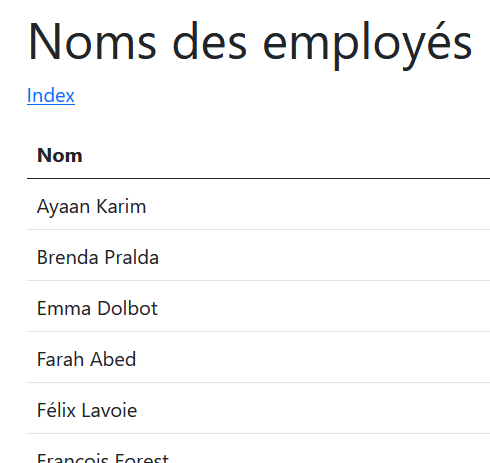
1. Quels employés ont été embauchés en 2023 ?
   * Utilisez **VwListeArtiste**.



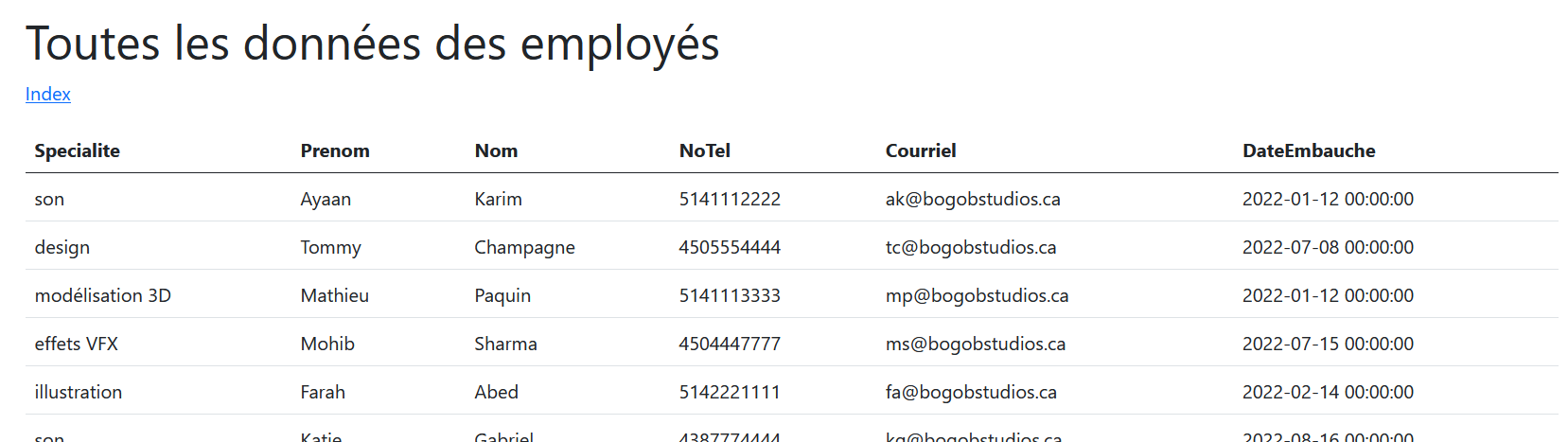
1. Quels employés ont la spécialité « modélisation 3D » ?
   * Utilisez **VwListeArtiste**.



1. Le prénom et le nom de tous les employés, classés par ordre alphabétique de prénom ?
   * Votre requête LINQ doit retourner une simple liste de string (**IEnumerable<string>**) comme ça pas besoin de créer de ViewModel.



1. Toutes les données des employés artistes avec jointure. (Pas le droit d’utiliser la vue **VwListeArtiste** !)
   * Vous aurez à créer un **ViewModel** nommé **ArtisteEmployeViewModel** avec les propriétés **Artiste** et **Employe**. Donc on envoie **IEnumerable<ArtisteEmployeViewModel>** à la vue Razor. Il est très possible que vous n’ayez pas à utiliser Join() pour y arriver !



1. **Combien d’artistes par spécialité ?**
   * Vous aurez à créer un ViewModel nommé **NbSpecialiteViewModel** avec les propriétés « **string Specialite** » et « **int Nb** ». Envoyez un **IEnumerable<NbSpecialiteViewModel>** à la vue Razor.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. **Combien d’artistes par spécialité ? En utilisant une vue**
   * Vous aurez à créer une vue SQL nommée **Employes.VW\_NbEmployesParSpecialite**
   * Puis dans le terminal, vous allez forcer les modifications avec la commande

**dotnet ef dbcontext scaffold Name=BDArtistes Microsoft.EntityFrameworkCore.SqlServer -o Models --context-dir Data --data-annotations --force**

(En assumant que vous avez appelé votre **chaine de connexion ‘BDArtistes’**)

Vous aurez alors le modèle **VwNbEmployesParSpecialite.cs** à votre disposition.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement