

Conception d'un système d'intégration de données et de développement de tableaux de bord pour un réseau d'agences bancaires.

Les collaborateurs constatent que :

1. La qualité des dossiers de prêts immobiliers accordés diverge au sein du réseau.
2. Les délais de prise de décision sur l'accord ou le rejet des prêts sont beaucoup trop longs.

En bref, l'objectif est double :

1. Réduire les délais de prise de décision sur l'accord des prêts.
2. Améliorer la qualité des dossiers de prêt validés.

Elle décrit plusieurs acteurs sur la chaîne de décision des prêts :

1. Les conseillers bancaires en agence qui n'ont pas d'outils pour prendre des décisions à l'heure actuelle, et se basent sur leur expérience ;
2. Le directeur d'agence qui peut confirmer ou infirmer une décision de l'un de ses conseillers ;
3. La cellule centrale d'étude des prêts immobiliers, qui a le dernier mot mais croule sous les demandes.

La cause racine du délai de prise de décision est la multiplicité des données à étudier pour accorder ou refuser un prêt immobilier :

1. La situation professionnelle des emprunteurs (catégorie socioprofessionnelle, statut d'emploi, régularité des revenus, et montants) ;
2. La situation familiale des emprunteurs (leur âge, le nombre de personnes à charge) ;
3. Le montant de l'apport personnel ;
4. La capacité d'épargne des emprunteurs (le reste à vivre).

Toutes ces données sont dans des bases de données « qui ne se parlent pas », et tout le travail est fait dans la tête des conseillers bancaires, et à la main par la cellule centrale.

Réconcilier des données : Pour résoudre l'enjeu, il faut réconcilier toutes les informations du processus de décision évoquées (situation professionnelle, situation familiale, montant de l'apport...) centraliser dans un Data Warehouse et créer un tableau de bord pour calculer un score de prêt immobilier ;

Le Crédit Breton gère plusieurs clients. Il faudra travailler avec un volume de données conséquent pour fournir un score de prêt immobilier à chacun.

Glossaire de données (source de données) :

Demandes de prêts :

- **Montant opération** : le montant des opérations immobilières associées aux demandes de prêt. C'est le coût total de l'opération, indépendamment de l'apport de l'emprunteur.
- **Durée** : la durée du prêt demandé en nombre de mois.
- **Date de demande** : la date de la demande de prêt.
- **Date de clôture** : la date de clôture de la demande de prêt est la date d'acceptation ou du refus du prêt.
- **Numéro client** : le numéro client du demandeur de prêt.
- **Numéro demande de prêt** : le numéro de la demande de prêt.
- **Accord** : décision finale sur l'accord ou le refus du prêt (Oui ou Non).
- **Numéro d'agence** : numéro de l'agence qui a reçu la demande de prêt.

Agences :

- **Numéro d'agence** : numéro de l'agence.
- **Localisation** : coordonnées GPS de l'agence.
- **Ville** : ville de l'agence.

Situation professionnelle

- **Numéro client** : numéro client du demandeur de prêt.
- **Catégorie socioprofessionnelle** : catégorie professionnelle de l'emprunteur (entrepreneur, cadre...).
- **Statut d'emploi** : type de contrat (CDI, CDD...).
- **Régularité des revenus** : 3 catégories de régularité (stables, irréguliers ou très irréguliers)
- **Revenu mensuel moyen** : revenu mensuel moyen sur les 12 derniers mois.

Apport :

- **Numéro de demande de prêt** : numéro de la demande de prêt.
- **Apport** : apport en € de l'emprunteur pour l'opération immobilière.

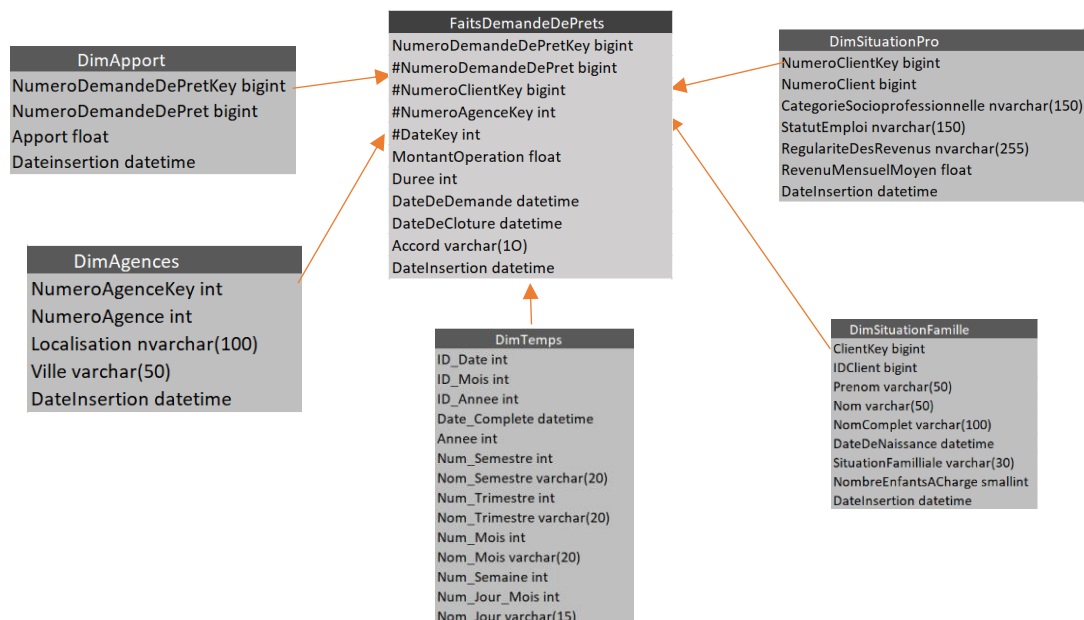
Situations de famille :

- **ID Client** : numéro client de l'emprunteur.
- **Date de naissance** : date de naissance de l'emprunteur.
- **Situation de famille** : situation familiale de l'emprunteur (Divorcé, Marié, etc.).
- **Nombre d'enfants à charge** : nombre d'enfants à charge de l'emprunteur.

Ces tables sont stockées dans la base de données SQL Server, hormis la table Apport stockée dans le fichier Microsoft Excel nommé Apport.xls.

Schéma de l'entrepôt

Voici le schéma en étoile de l'entrepôt de données :



Une fois le schéma en étoile validé, il faut créer l'entrepôt sous SQL Server.

Création du schéma :

```
CREATE SCHEMA PRET;
```

Création des tables des dimensions :

Table de dimension : DimApport

```
CREATE TABLE [PRET].[DimApport](
    [NumeroDemandeDePretKey] [bigint] IDENTITY(1,1) NOT NULL,
    [NumeroDemandeDePret] [bigint] NULL,
    [Apport] [float] NULL,
    [Dateinsertion] [datetime] NULL,
    CONSTRAINT [PK_DimApport] PRIMARY KEY (NumeroDemandeDePretKey)
);
```

Table de dimension : DimAgences

```
CREATE TABLE [PRET].[DimAgences](
    [NumeroAgenceKey] [int] IDENTITY(1,1) primary key,
    [NumeroAgence] [int] NULL,
    [Localisation] [nvarchar](100) NULL,
    [Ville] [varchar](50) NULL,
    [DateInsertion] [datetime] NULL,
);
```

Table de dimension : DimSituationFamille

```
CREATE TABLE [PRET].[DimSituationFamille](
    [ClientKey] [bigint] IDENTITY(1,1) PRIMARY KEY,
    [IDClient] [bigint] NULL,
    [Prenom] [varchar](50) NULL,
    [Nom] [varchar](50) NULL,
    [NomComplet] [varchar](100) NULL,
    [DateDeNaissance] [datetime] NULL,
    [SituationFamilliale] [varchar](30) NULL,
    [NombreEnfantsACharge] [smallint] NULL,
    [DateInsertion] [datetime] NULL,
);
```

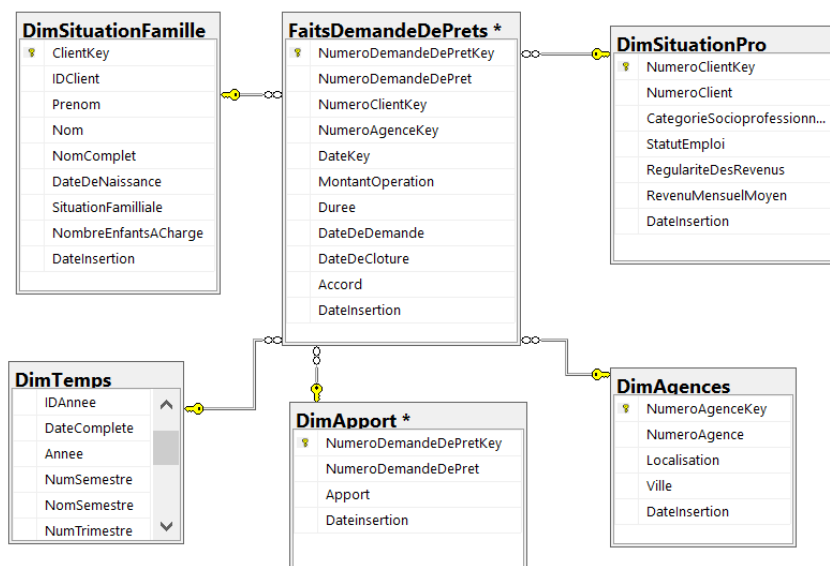
Table de dimension : DimSituationPro

```
CREATE TABLE [PRET].[DimSituationPro](
    [NumeroClientKey] [bigint] IDENTITY(1,1) PRIMARY KEY,
    [NumeroClient] [bigint],
    [CategorieSocioprofessionnelle] [nvarchar](150),
    [StatutEmploi] [nvarchar](150),
    [RegulariteDesRevenus] [nvarchar](255),
    [RevenuMensuelMoyen] [float],
    [DateInsertion] [datetime],
);
```

Table de dimension : FitsDemandeDePrets

```
CREATE TABLE [PRET].[FitsDemandeDePrets](
    [NumeroDemandeDePretKey] [bigint] IDENTITY(1,1) PRIMARY KEY,
    [NumeroDemandeDePret] [bigint],
    [NumeroClientKey] [bigint],
    [NumeroAgenceKey] [int],
    [DateKey] [int],
    [MontantOperation] [float],
    [Duree] [int],
    [DateDeDemande] [datetime],
    [DateDeCloture] [datetime],
    [Accord] [varchar](10),
    [DateInsertion] [datetime]
);
```

Voici le modèle en étoile final l'entrepôt de données sous SQL Server :

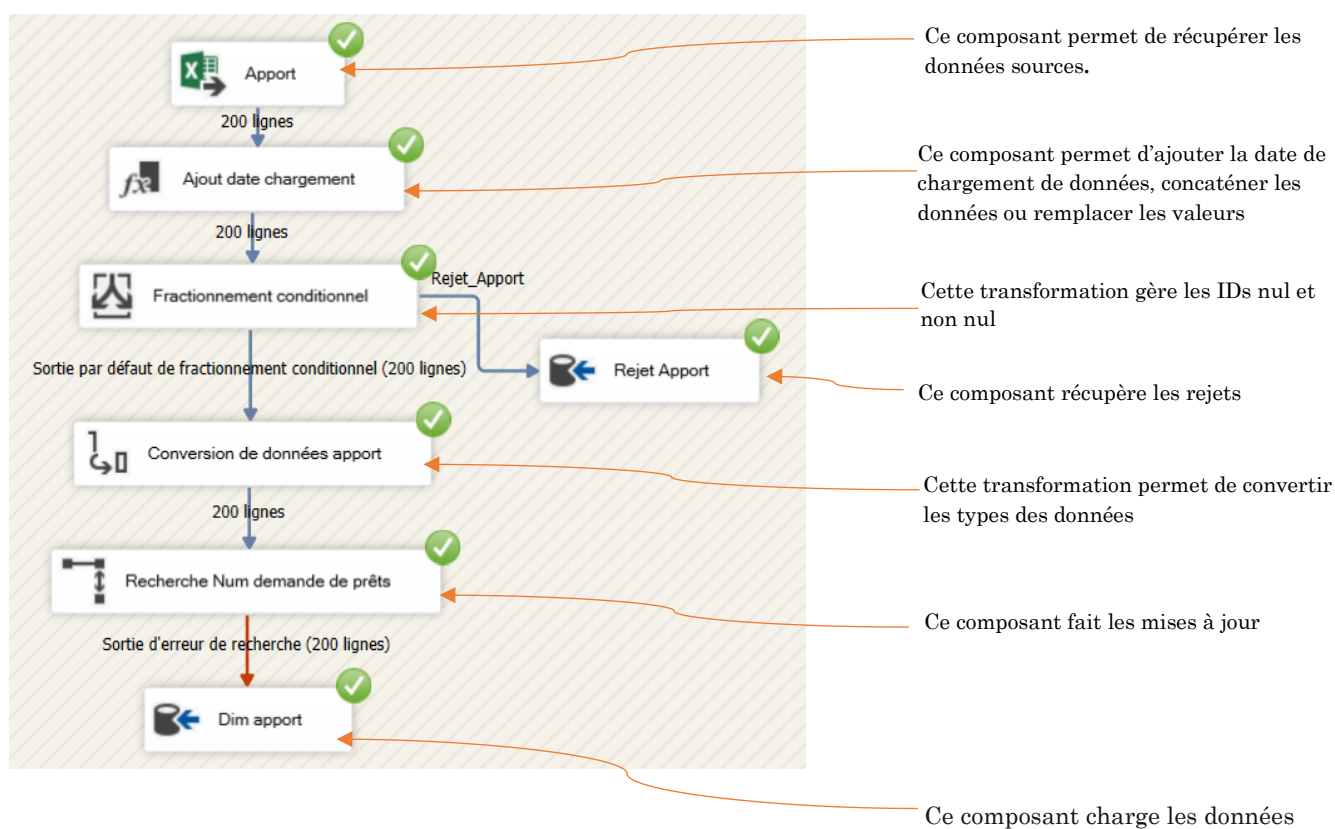


Maintenant que les tables de l'entrepôt sont créées, il faut réaliser les processus qui vont remplir ces tables à partir des données sources.

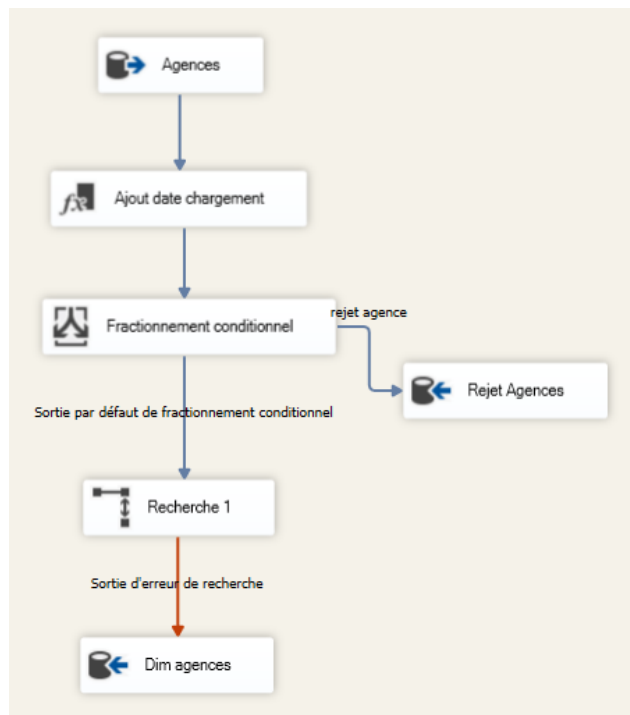
Spécification des données sources et cibles

Établir une connexion aux sources et l'entrepôt de données. Les connexions sources et destination sont maintenant disponibles dans le zone Gestionnaires de connexion. Il faut alors construire les différents flux pour remplir les tables de l'entrepôt.

Remplissage de la table DimApport :



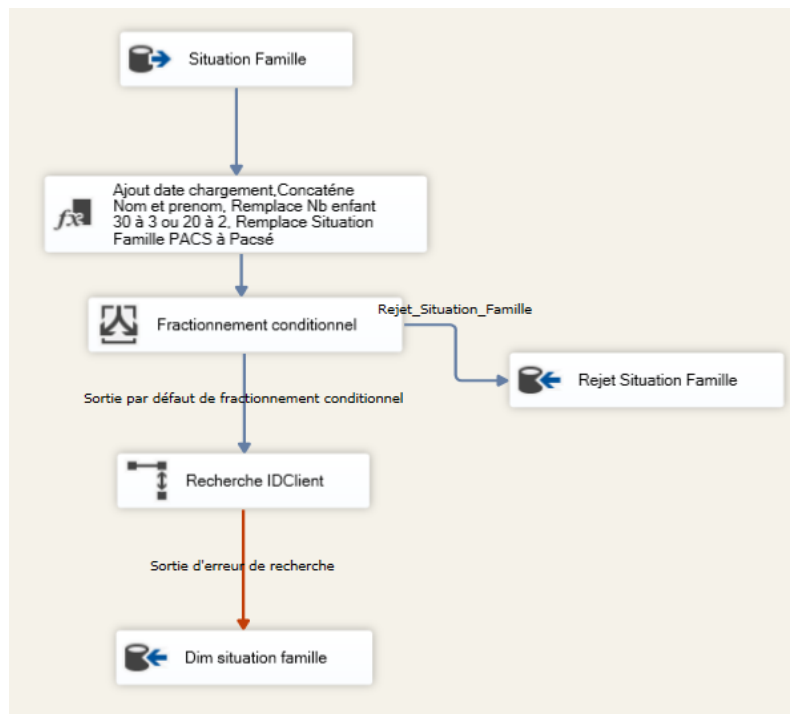
Remplissage de la table DimAgences :



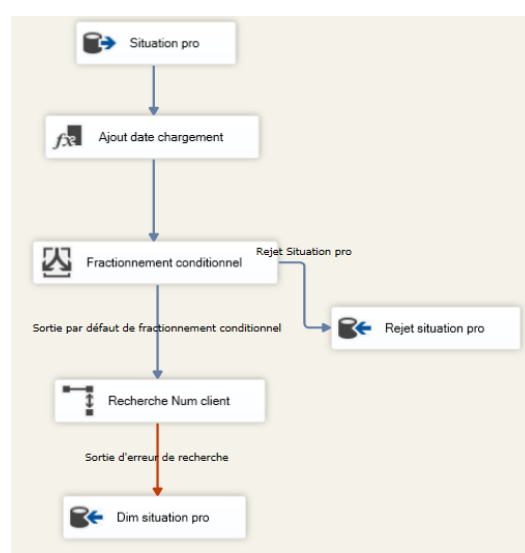
Remplissage de la table DimSituationFamille :

Cependant, cette table contient des anomalies du fait de mauvaises saisies manuelles. En visualisant la table source on peut s’apercevoir certaines personnes ont respectivement 30 et 20 enfants à charge, ce qui est fortement improbable.

On peut s’apercevoir également que les catégories des situations familiales ont des redondances, car nous avons deux valeurs possibles avec “PACS” et “Pacsé” pour une même situation.



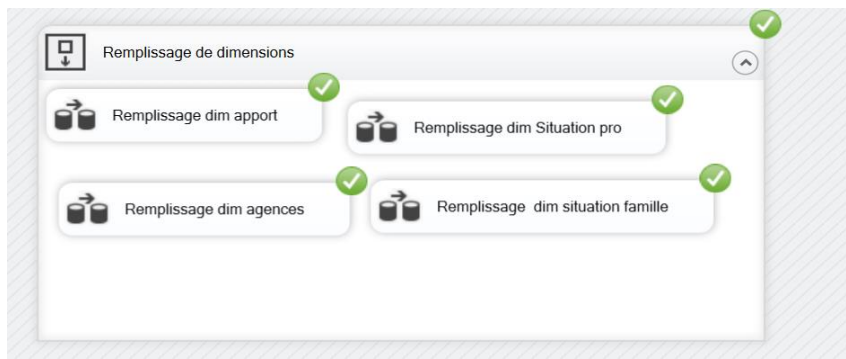
Remplissage de la table DimSituationPro :



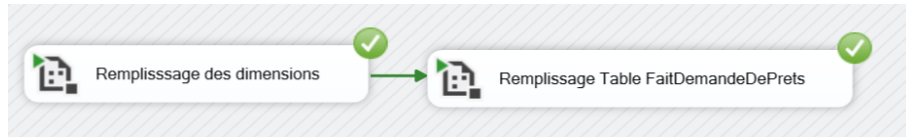
Remplissage de la table FaitsDemandeDePrets :



Regroupement des dimensions dans un conteneur de séquence :



Ordonnement des packages : Un dernier package est nécessaire pour faire appel à tous les packages créés auparavant, il sera nommé. À son exécution, le package doit créer et charger les dimensions ainsi que la table de faits. Ci-dessous un aperçu de l'ordonnement des packages :



Planification du package : Chargement Initial

Comme les données sources évoluent en continu, il est important de planifier régulièrement le package de chargement initial. Idéalement, ce chargement peut être lancé chaque nuit pour intégrer les modifications effectuées dans la journée. Pour automatiser cela, on peut s'appuyer sur SQL Server Agent, qui permet de planifier facilement l'exécution des packages :

Conclusion :

Ces flux servent à réaliser un premier chargement complet des données. Ensuite, seuls les nouveaux enregistrements doivent être pris en compte lors des chargements suivants, sans impacter les données déjà présentes dans le Data Warehouse.

Merci pour votre attention.