

Chloé LEYRIS

Elora GODARD

# Intégration de données dans un datawarehouse – SAE 3.02 –

## Table des matières

I.	Modèle final et plan de transformation .....	3
II.	Explication du flux Talend.....	4
III.	Schéma relationnel .....	5
IV.	Les requêtes SQL .....	5
V.	Bilan du projet.....	11
VI.	ANNEXE.....	14
	Parties personnelles.....	14
	SCRIPT SQL.....	15

## I. Modèle final et plan de transformation

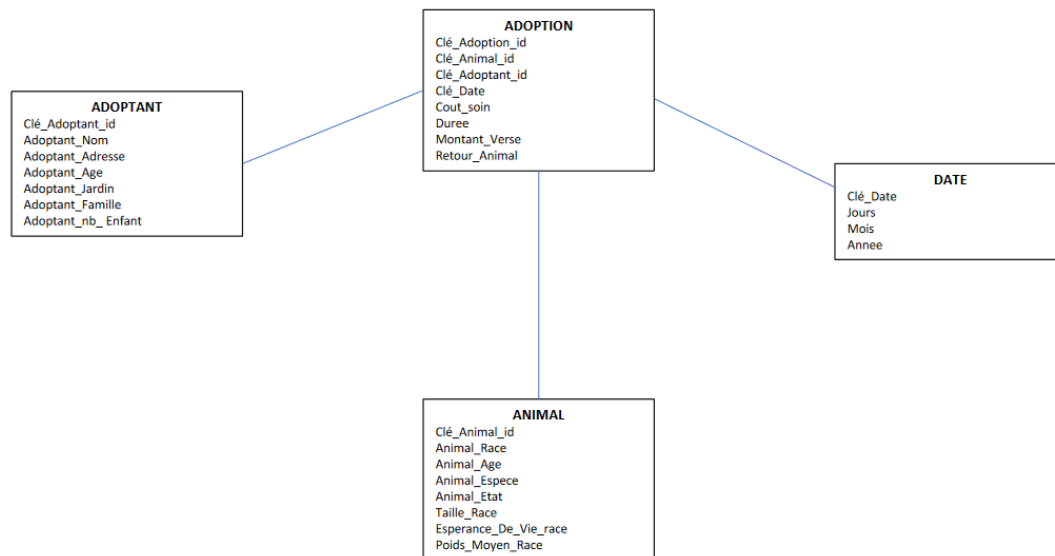


Figure 1 : Modèle final de l'entrepôt de données

Le modèle en étoile (figure1) présente une table de fait Adoption et 3 tables de dimensions : Adoptant, Animal et Date. Nous avons décidé de prendre Adoption comme table de fait car il s'agit du point clé qui relie toutes les informations nécessaires à l'analyse. On y retrouve le coût des soins (voir figure 2 pour le calcul), la durée de séjour, le montant versé par les adoptants et enfin si l'animal est revenu à la SPA. De plus, la dimension Date permettra d'analyser les adoptions par mois. La dimension Adoptant relie toutes les informations de l'adoptant. Enfin, nous avons regroupé les informations des animaux et des races dans une seule dimension, afin d'avoir toutes les informations de l'animal dans une seule table.

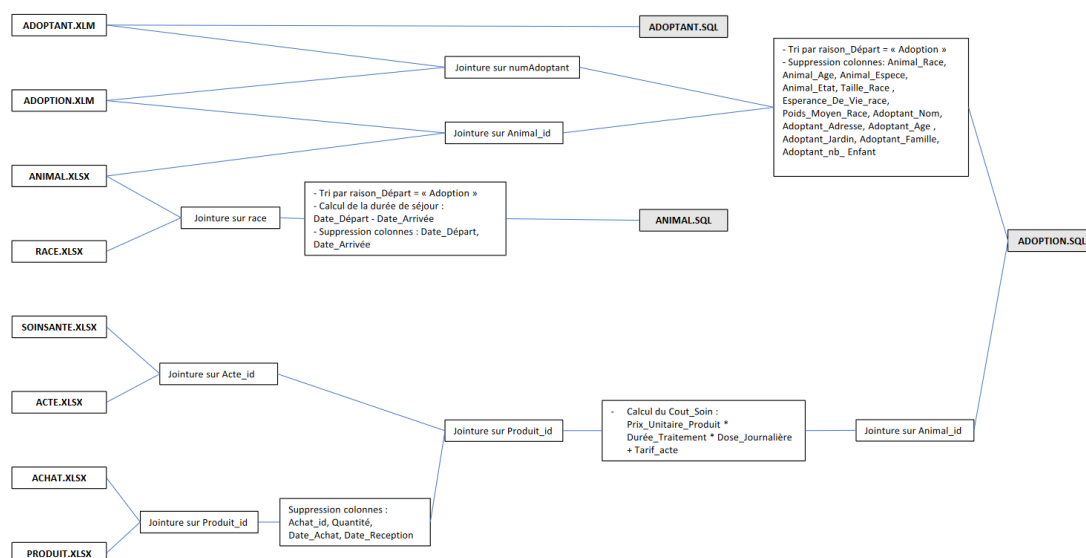


Figure 2 : Plan de transformation final

Le schéma de transformation (figure 2) décrit le processus de modification que nous allons implémenter sur Talend. Comme mentionné précédemment, nous fusionnons les données de la table Animal et de la table Race. Ensuite, nous procédons à un tri pour ne conserver que les enregistrements liés aux adoptions et nous calculons la durée du séjour. Nous éliminons également les colonnes superflues. Ainsi, nous obtenons notre première table Animal, prête à être exportée vers SQL.

En parallèle, nous importons la table Adoptant au format XML et l'exportons directement en SQL sans apporter de modifications. Cette approche vise à conserver toutes les informations jugées utiles pour notre analyse. De plus, nous réalisons une jointure sur NumAdoptant entre Adoptant et Adoption, ainsi qu'une jointure entre Animal et Adoption. Ces opérations nous permettent d'établir les liens entre les trois tables.

Ensuite, nous effectuons les jointures nécessaires entre les différentes tables qui nous seront utiles pour calculer le coût des soins. Enfin, nous créons la jointure avec la table Adoption. Ainsi, toutes les données requises pour notre analyse sont consolidées dans notre table de fait Adoption, que nous exportons ensuite vers SQL.

## II. Explication du flux Talend

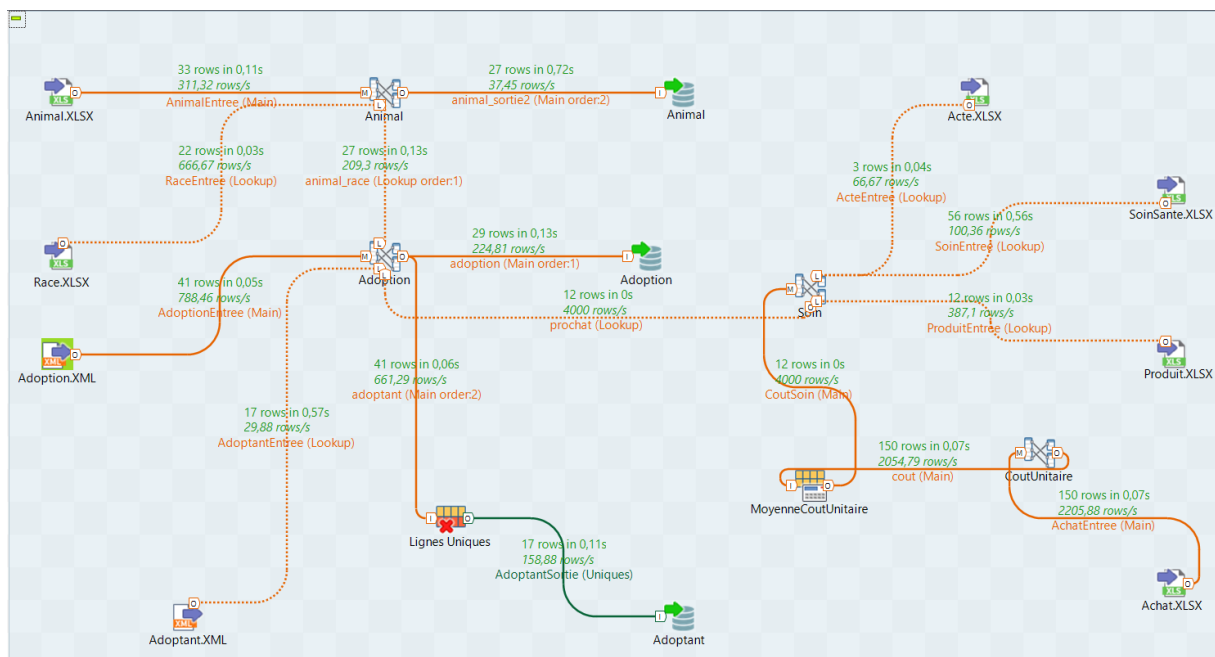


Figure 3 : Flux Talend

Nous avons initié le processus en effectuant une jointure entre les tables Race et Animal grâce au composant « tMap » appelé Animal sur la figure 3. Ce composant va nous permettre de réaliser la jointure sur « race » et les calculs de la durée du séjour. En effet, nous avons utilisé la fonction « TalendDate.diffDate » pour faire la différence entre la date de départ et la date d'arrivée de l'animal à la SPA. De plus, nous avons précisé que nous voulions uniquement les animaux adoptés : « "adoption".equals(AnimalEntree.raison\_depart) ». Ensuite, nous avons ajouté les éléments nécessaires pour extraire la table Animal vers notre base de données SQL, en utilisant le composant « tMysqlOutput ». Une seconde jointure a été réalisée entre le « tMap » Animal et les deux fichiers XML, Adoption et Adoptant, regroupés dans un autre « tMap ». Ce dernier nous a permis d'effectuer

les opérations nécessaires pour obtenir les informations pour notre analyse. Par exemple, nous avons utilisé une condition pour modifier les « malheureusement » de l'attribut retour, en « oui ». Nous avons ensuite mis les colonnes qui nous intéressaient (comme la durée de séjour, calculé dans le tMap Animal) puis nous avons exporté notre table dans QL.

De plus, nous avons utilisé le composant « tUniqRow » pour les adoptants, éliminant ainsi les doublons et exportant 17 lignes distinctes dans la table Adoptant de la base de données.

En parallèle, du côté droit, nous nous sommes concentrés sur le calcul du coût des soins. Pour cela, nous avons importé la table Achat, réalisant plusieurs opérations essentielles : dans un premier « tMap », nous avons converti les prix unitaires en type float et remplacé les virgules par des points pour pouvoir réaliser les calculs. Ensuite, un « tAggregateRow » a été utilisé pour obtenir la moyenne des prix pour chaque produit. Ces résultats ont été regroupé dans un second « tMap » avec les tables produit, acte, et soin, afin de calculer le coût total. Nous avons simplement ajouté le tarif de l'acte au prix unitaire calculé.

Enfin, nous avons relié ce résultat au « tMap » Adoption pour ajouter le coût des soins, avant d'exporter la table dans notre base de données.

### III. Schéma relationnel

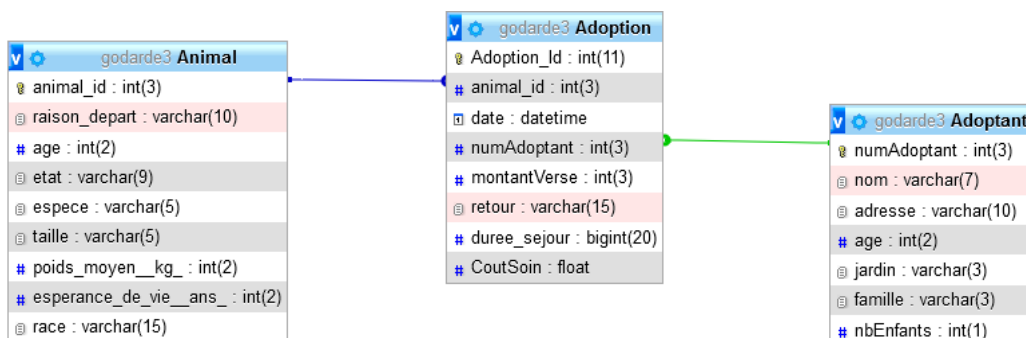


Figure 4 : Schéma relationnel de notre entrepôt de données

Notre schéma relationnel permet de voir les clés primaires et référentielles que nous avons défini. La table Animal a pour clé primaire « animal\_id ». Elle est liée ensuite à la clé référentielle « animal\_id » dans la table Adoption. De même pour les adoptants, dont la clé primaire est « numAdoptant » qui est lié à la clé référentielle « numAdoptant » de Adoption. Enfin, la clé primaire de Adoption est « Adoption\_id » que nous avons créée directement sur MySQL, en la générant comme clé primaire (paramètre A\_I coché). Elle est donc codée comme des nombres allant de 1 à 29 (notre nombre de lignes).

### IV. Les requêtes SQL

Dans cette partie nous allons présenter les requêtes SQL que nous avons effectué afin de répondre aux problématiques de la SPA.

Tout d'abord, la SPA souhaite savoir les mois où il y a en moyenne le plus d'adoptions d'animaux. Pour cela nous avons réalisé la requête suivante :

```
SELECT MONTH(Adoption.date) AS MOIS, COUNT(Adoption.Adoption_Id) As NombreAdoption
FROM Adoption
GROUP BY MONTH(Adoption.date)
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

Et ainsi on obtient le résultat suivant :

MOIS	NombreAdoption
12	6
9	5
6	3
4	3
1	3
11	3
8	2
7	2
3	1
10	1

Ainsi, on constate que les mois qui regroupent le plus d'adoptions sont les mois de décembre (6 adoptions) et de septembre (5 adoptions). En revanche, le mois d'octobre et le mois de mars sont les mois où il y a le moins d'adoptions.

Ensuite, la SPA se questionne sur la durée moyenne, minimale et maximale de séjour d'un animal adopté. Nous avons donc fait la requête suivante :

```
SELECT AVG(Adoption.duree_sejour) AS DureeMoyenne, MIN(Adoption.duree_sejour) AS DureeMinimale, MAX(Adoption.duree_sejour) AS DureeMaximale
FROM Adoption
```

On obtient le résultat suivant :

DureeMoyenne	DureeMinimale	DureeMaximale
127.2414	2	318

Nous constatons que la durée moyenne du séjour d'un animal est de 127 jours, soit environ 4 mois, avec une durée minimale de 2 jours et une durée maximale de 318 jours.

Puis, nous avons étudié quelles étaient les races d'animaux les plus adoptées :

```
SELECT Animal.espece AS Espece, Animal.race AS Race, COUNT(Adoption.Adoption_Id) AS NbAdoption
FROM Adoption, Animal
WHERE Animal.animal_id=Adoption.animal_Id
GROUP BY Animal.race
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

On obtient enfin le résultat suivant :

Espece	Race	NbAdoption
chien	Malinois	4
chien	Bichon	4
chien	Chihuahua	3
chien	Labrador	3
autre	Perroquet	3
chien	Dobermann	2
chien	Bulldog	2
chien	Fox Terrier	2
chien	Berger allemand	1
chien	Épagneul	1
chien	Caniche	1
chat	Persan	1
chat	Sphynx	1
chien	Griffon	1

On constate que les races les plus adoptées sont les Malinois et les Bichons, avec 4 adoptions chacun. Globalement, on constate qu'il y a beaucoup plus de chiens adoptés que les autres espèces. Enfin, les chats ne présentent que 2 adoptions au total et 3 pour les autres espèces.

Espece	NbAdoption
chien	24
autre	3
chat	2

```
SELECT Animal.espece AS Espece, COUNT(Adoption.Adoption_Id) AS NbAdoption
FROM Adoption, Animal WHERE Animal.animal_id=Adoption.animal_Id
GROUP BY Animal.espece
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

Puis, nous allons voir ce qui caractérise un animal qui a plus de chance d'être adopté. Pour cela nous allons regarder plusieurs éléments. Tout d'abord, nous avons vu précédemment qu'une caractéristique assez évidente est l'espèce de l'animal. En effet, il y a 24 chiens qui ont été adopté sur 29 adoptions. Cependant, cela peut être dû au fait qu'il y a plus de chiens au refuge que d'autres espèces.

Nous allons donc regarder les autres caractéristiques que l'on peut étudier.

### La taille :

```
SELECT Animal.taille AS Taille, COUNT(Adoption.Adoption_Id) AS NbAdoption FROM Adoption,
Animal
WHERE Animal.animal_id=Adoption.animal_Id
GROUP BY Animal.taille
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

Taille	NbAdoption
moyen	12
petit	11
grand	6

On constate donc que les animaux de taille moyenne ou petite sont deux fois plus adoptées que des animaux de grande taille.

#### Le poids :

```
SELECT Animal.poids_moyen__kg_ AS PoidsMoyen, COUNT(Adoption.Adoption_Id) AS NbAdoption
FROM Adoption, Animal
WHERE Animal.animal_id=Adoption.animal_Id
GROUP BY Animal.poids_moyen__kg_
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

PoidsMoyen	NbAdoption
1	6
6	5
5	5
3	4
15	3
2	2
12	2
10	1
7	1

Les animaux les plus adoptés ont un poids qu'on peut considérer comme moyen voir petit. En effet, les animaux de 1kg en moyenne sont les plus adoptés, suivi des animaux de 6, 5 et 3 kg en moyenne.

#### L'Age :

```
SELECT Animal.age AS AgeAnimal, COUNT(Adoption.Adoption_Id) AS NbAdoption
FROM Adoption, Animal
WHERE Animal.animal_id=Adoption.animal_Id
GROUP BY Animal.age
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

AgeAnimal	NbAdoption
6	5
0	5
1	4
5	3
7	3
3	3
12	2
4	2
2	1
14	1

Les animaux les plus adoptés sont plutôt jeunes (entre 0 et 6 ans).

#### L'espérance de vie :

```
SELECT Animal.esperance_de_vie__ans_ AS AgeAnimal, COUNT(Adoption.Adoption_Id) AS
NbAdoption
FROM Adoption, Animal
WHERE Animal.animal_id=Adoption.animal_Id
GROUP BY Animal.esperance_de_vie__ans_
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

EsperanceDeVie	NbAdoption
8	18
3	3
15	3
7	2
6	1
10	1
11	1

On constate que près de 60% des adoptions sont des animaux ayant une espérance de vie de 8 ans.

#### L'état de l'animal :

```
SELECT Animal.etat AS EtatAnimal, COUNT(Adoption.Adoption_Id) AS NbAdoption
FROM Adoption, Animal
WHERE Animal.animal_id=Adoption.animal_Id
GROUP BY Animal.etat
ORDER BY COUNT(Adoption.Adoption_Id) DESC
```

EtatAnimal	NbAdoption
Bon	12
ok	7
maltraité	6
Bien	3
mauvais	1

Plus de 60% des adoptions sont des animaux dans un état considéré « Bon » ou « Ok ».

En résumé, les traits distinctifs qui se dégagent sont les animaux de petite taille, jeunes, avec une espérance de vie de 8 ans et dans un état de santé qualifié de « moyen ». Les autres caractéristiques semblent moins significatives pour l'analyse, car elles ne génèrent pas un nombre d'adoptions nettement supérieur pour un attribut particulier.

Ensuite, nous nous questionnons à propos du profil des adoptants :

```
SELECT DISTINCT Adoptant.*
FROM Adoptant, Adoption
WHERE Adoptant.numAdoptant = Adoption.numAdoptant
ORDER BY Adoptant.age ASC
```

numAdoptant	nom	adresse	age	jardin	famille	nbEnfants
300	Tatoud	Eybens	20	non	non	0
294	Bernard	Valence	30	oui	oui	1
310	Derbey	Valence	30	non	oui	2
326	Derbey	Vinay	34	oui	oui	2
291	Derbey	Vinay	35	oui	oui	2
308	Derbey	Vinay	35	oui	oui	2
292	Derby	Grenoble	38	oui	oui	2
320	Durant	Grenoble	40	oui	oui	4
293	Derbey	Valence	40	non	oui	2
331	Mandier	Grenoble	40	non	non	0
309	Derbey	Vinay	42	oui	oui	2
328	Mandier	Valence	45	oui	oui	1
334	Simonet	Grenoble	50	oui	non	0
311	Mandier	Valence	50	oui	oui	1
325	Gacon	Vinay	54	oui	oui	2
324	Martin	Pontcharra	60	oui	oui	0

Nous avons décidé de trier les âges par ordre croissant afin de faciliter la lecture. On constate donc que l'âge des adoptants varient entre 20 et 60 ans avec une plus forte distribution entre 30 et 50 ans. De plus, presque 90% des adoptants ont un jardin et des enfants.

Pour finir, nous avons étudié la participation moyenne des adoptants en regardant la moyenne du montant versé pour chaque espèce d'animal.

```
SELECT Animal.espece AS Espece, AVG(Adoption.montantVerse) AS ParticipationMoyenne
FROM Animal, Adoptant, Adoption
WHERE Adoption.numAdoptant=Adoptant.numAdoptant AND Animal.animal_id=Adoption.animal_id
GROUP BY Animal.espece
ORDER BY ParticipationMoyenne DESC
```

Espece	ParticipationMoyenne ▾ 1
chat	110.00
chien	85.00
autre	83.33

Les chats bénéficient du montant le plus élevé parmi tous les animaux, suivi par les chiens avec tout de même un montant moindre.

Pour finir, nous n'avons pas pu faire la requête concernant le coût des soins car nous n'avons pas réussi à importer correctement le coût des soins sur SQL.

## V. Bilan du projet

### Etape 1 : Compréhension des données existantes, proposition d'un modèle en étoile de l'entrepôt de données et d'un plan de transformation des données

Dans cette première étape nous avons analysé les différents fichiers des données qui étaient disponibles et nous avons tout d'abord élaboré le dictionnaire des données. Cette création nous a permis de bien comprendre les liens entre les différents fichiers. Nous n'avons pas rencontré de difficultés majeures sur cette étape. En revanche, l'étape suivante fut un peu plus compliquée. En effet, nous avons dû établir le modèle en étoile de l'entrepôt de données ainsi que le plan de transformation des données. En effet, nous n'avons pas réussi à bien comprendre la problématique et nous avons eu du mal à savoir quelles étaient les tables qui allaient être les plus importantes dans notre analyse.

Pour le modèle en étoile nous nous sommes posé la question au sujet de la table adoption. Nous ne savions pas si elle devait être la table de fait ou une table de dimension. Nous avons finalement pensé qu'il serait mieux qu'elle soit en table de dimension car nous pensions que l'axe d'analyse était concentré sur le séjour de l'animal adopté uniquement. De plus, nous avons aussi eu la question de la table de dimension « Soins ». Nous avons décidé de l'ajouter, pensant qu'elle représenterait un axe d'analyse important, cependant elle était utile uniquement pour calculer le coût des soins.

Pour le plan de transformation, il a été difficile de visualiser les étapes car nous n'avions pas encore une connaissance suffisante sur Talend qui nous permettrait de faire un plan de transformation correct. En effet, nous ne savions pas si on devait regrouper toutes les tables en une seule, pour ne faire qu'une sortie avec toutes nos données à l'intérieur, représentant alors notre table de fait. Nous avons donc choisi cette approche.

Suite au retour de Mme. FRONT, nous avons pu avoir un meilleur axe d'analyse. Ainsi, nous avons mieux compris la problématique de la SPA. De plus, nous avons pu expérimenter Talend plus en profondeur ce qui nous a permis de comprendre comment faire un meilleur plan de transformation.

## **Etape 2 : ETL sous Talend et construction de l'entrepôt de données sous MySQL**

Cette étape consistait à mettre en place notre plan de transformation dans Talend ainsi que l'exporter sur SQL. Pour la première étape nous avons créé le schéma relationnel et la vue qui nous permettrait d'implémenter le modèle de l'entrepôt de données sous MySQL :

### **Schéma relationnel :**

**Animal** (Clé Animal ID, race, âge, espèce, état, taille, espérance\_de\_vie, poids\_moyen)

**Adoptant** (Cle Adoptant ID, nom, adresse, âge, jardin, famille, nbEnfants)

**Date** (Clé date, jour, mois, année)

**Adoption** (Clé Adoption ID, Clé Animal ID, Clé Adoptant ID, Clé date, cout\_soin, durée, montant\_versé)

### **Vues :**

```
CREATE VIEW CubeAdoption
```

```
AS SELECT Adoption. *, Animal.race, Animal.age, Animal.espece, Animal.état, Animal.taille,
Animal.esperance_de_vie, Animal.poids_moyen, Adoptant.nom, Adoptant.adresse, Adoptant.age,
Adoptant.jardin, Adoptant.famille, Adoptant.nbEnfants, Date.jour, Date.mois, Date.année
```

```
FROM Adoption, Adoptant, Animal, Date
```

```
WHERE Adoption.cle_Animal_ID = Animal.cle_Animal_ID AND Adoption.cle_Adoptant_ID =
Adoptant.cle_Adoptant_ID AND Adoption.cle_Date_ID = Date.cle_Date_ID
```

Nous n'étions pas sûres de l'utilité de cette étape mais nous l'avons quand même réalisé pour nous donner une idée de nos tables finales.

Ensuite, nous n'avons pas eu le temps d'implémenter un jeu de test fictif donc nous avons directement commencer notre flux sur Talend. Nous avons rencontré plusieurs difficultés dans la création de notre processus ETL.

Tout d'abord, nous avons importé les tables Animal et Race pour les regrouper, car il s'agissait de la jointure la plus simple. Nous ne savions pas si nous devions trier les animaux uniquement par adoption ou si on décidait de tous les garder. Nous avons jugé qu'il était préférable de trier directement, pour ne garder que les animaux adoptés, car notre axe d'analyse concerne les adoptions. Nous avons ensuite eu un souci car nous ne pouvions pas utiliser le même lien pour deux sorties différentes. En effet on souhaitait relier le tMap de cette jointure à notre sortie SQL mais aussi à un autre tMap concernant l'Adoption. Nous avons donc dû créer 2 liens pour pouvoir faire les relations que nous souhaitions. De plus, dans le tMap permettant de joindre Animal et Race nous avons calculé la durée du séjour. Nous avons essayé plusieurs choses avant de trouver la fonction TalendDate.diffDate. Avant ça nous essayions simplement de faire la différence de la date de départ et la date d'arrivée, mais Talend nous retournait une erreur.

Ensuite, nous avons fait la partie adoption. Nous avons commencé par faire la jointure entre Adoption et Adoptant. Nous nous sommes posés la question sur s'il était nécessaire de remettre la raison du départ = adoption. Dans le doute, nous l'avons fait tout de même. De plus, nous avons eu une difficulté sur les retours des animaux. En effet, nous avons modifié les « malheureusement » en « oui » mais nous n'avons pas pu ajouter le remplacement des « null » en « non ». En effet, nous avons un message d'erreur à l'exécution. Nous avons donc décidé de laisser les null. De plus, lors de l'implantation sur la base de données, les retours qui sont égales à « oui » ne s'exportent pas. Nous n'avons finalement pas trouvé la réponse à ce problème.

Pour la table Adoptant, nous n'étions pas sûres de devoir passer par la jointure des Adoptions. En effet, nous avions un doute sur le fait que `proprio_id` dans la table Adoption était la même chose que `numAdoptant` dans la table Adoptant. Finalement, nous avons su que c'était la même chose et il était nécessaire de les lier. Ensuite, nous avons le problème des doublons dans la table Adoptant. Nous avons donc cherché sur Internet comment résoudre ce problème. Nous avons donc utilisé le composant « `tUniqRow` » qui permet de supprimer les doublons.

Pour finir, nous avons essayé les calculs du coût des soins. Nous avons tout d'abord essayé de relier les tables Acte, Produit, Soin et Achat afin de faire le calcul :  $\text{prix\_unitaire} * \text{dose\_journalière} * \text{duree\_traitement} + \text{tarif\_acte}$ . Cependant la requête ne fonctionnait pas. Nous avons finalement été aiguillé par des camarades, qui nous ont expliqué qu'il fallait remplacer les virgules du fichier source en points. Nous avons donc pris la table Achat et nous avons converti le prix unitaire en type float et modifié les virgules. Nous avons ensuite fait la moyenne de chaque prix\_unitaire. Cependant, la formule que nous voulions faire de base de fonctionne pas. Nous avons uniquement pu ajouter le tarif de l'acte au prix unitaire. Nous avons encore un problème dans notre base de données. En effet, nous n'obtenons le coût des soins que pour 2 lignes et des « null » sur toutes les autres. Nous n'avons finalement pas pu trouver la raison de ce problème. De plus, par la suite, il aurait fallut faire la somme des soins pour chaque animal, mais nous n'avons pas pu arriver jusque-là.

### **Etape 3 : Exploitation de l'entrepôt de données**

Pour finir, dans cette étape nous avons réalisé nos requêtes. Avant tout, nous devons définir nos clés primaires et clés référentielles sur SQL. Cette étape fut plutôt simple, comme nous l'avons fait l'année dernière. Nous avons juste un doute sur l'ajout d'un attribut clé primaire dans la table Adoption, bien que nous jugions cela nécessaire pour pouvoir lier les tables. Nous avons donc finalement ajouté une colonne que nous avons défini comme clé primaire et qui a généré automatiquement des nombres différents allant de 1 à 29, correspondant au nombre de lignes.

Pour la résolution des requêtes, nous n'avons pas eu de difficultés très importantes et avons réussi à en faire la plupart. Nous avons tout de même un doute sur la participation. Nous étions en désaccord sur le fait que la participation était le pourcentage d'adoptant par espèce ou s'il s'agissait des montants versés en moyenne pour chaque animal. En discutant avec nos camarades, nous avons finalement compris qu'il s'agissait des montants versés.

### **Réflexion sur d'autres problématiques**

D'autres problématiques peuvent se poser ici. En effet, nous avons des données sur les bénévoles de la SPA, mais nous ne faisons aucune analyse dessus. Nous pourrions ainsi faire un axe sur le profil des bénévoles. De plus, nous avons des informations sur les dédommagements des bénévoles.

De plus, nous pourrions faire aussi une analyse sur les animaux qui ne sont pas adoptés, pour en faire une comparaison avec l'analyse que nous avons faite au cours de cette SAE.

Enfin, il aurait peut-être été intéressant d'étudier les retours des animaux. Nous aurions pu voir la raison, l'état de l'animal par la suite, etc.

## VI. ANNEXE

### Parties personnelles

#### Chloé :

Dans le cadre de cette SAE, Elora et moi avons opté pour une approche collaborative dans l'accomplissement de la plupart des tâches. Nous avons adopté la méthode de travailler en tandem, alternant le contrôle de la souris et du clavier. Cette méthodologie reflète notre familiarité avec ce mode de collaboration, qui se révèle être le plus adapté à mes préférences de travail.

Malgré notre choix de travailler principalement en binôme, nous avons également réalisé certaines tâches de manière individuelle. Par exemple, j'ai été responsable de la conception du plan de transformation et du modèle en étoile au propre, de manière autonome. Pour les requêtes SQL, nous avons décidé de les répartir entre nous, car toutes deux à l'aise avec cet aspect du projet.

La décision de collaborer étroitement nous a permis de résoudre rapidement d'éventuelles erreurs survenues lors de l'utilisation de Talend. Travailler à deux nous a donné la possibilité de combiner nos compétences et de surmonter plus aisément les défis qui ont pu se présenter.

Au début de cette SAE, j'ai abordé l'utilisation de Talend avec quelques appréhensions, étant donné que je n'étais pas familière avec ce logiciel. Cependant, à mesure que le projet avançait, j'ai acquis une compréhension approfondie de l'utilité de Talend pour l'intégration et le traitement efficace d'une base de données. En somme, cette expérience m'a permis de développer mes compétences et d'apprécier pleinement les avantages de ce logiciel dans le contexte de notre projet.

#### Elora :

Comme l'a dit Chloé dans sa partie, nous avons principalement travaillé ensemble. En effet, nous sommes habitués à fonctionner de cette manière, ce qui nous permet d'avancer plus efficacement. Lorsque l'une d'entre nous a un doute sur une étape, l'autre est capable de lui venir en aide ou d'apporter une nouvelle vision.

Cependant, notamment pour le premier livrable, nous nous sommes découpé le travail. J'ai fait le dictionnaire des données, qui a vraiment permis de trouver les liens entre les différents fichiers. Par la suite, sur Talend, je me suis plus occupée des formules et de la partie du calcul du coût de soins. Cette dernière tâche s'est avérée difficile et il était compliqué à mettre en place, donc j'ai finalement demandé des conseils à mon binôme, et nous avons pu arriver à un certain résultat, bien que ce dernier ne soit pas totalement satisfaisant.

De plus, nous nous sommes découpé le travail pour l'élaboration des requêtes sur SQL. Cela nous a permis d'aller plus vite et nous n'avons pas rencontré d'importantes difficultés car nous avons déjà fait cela l'année dernière.

Enfin, nous avons travaillé ensemble pour l'élaboration du rapport, mais je me suis chargé de la rédaction, car j'ai plus de facilité à taper à l'ordinateur.

Le logiciel Talend est un logiciel difficile à appréhender mais travailler en groupe dessus m'a permis de comprendre la plupart des composantes et comprendre mieux le principe même de la SAE.

## SCRIPT SQL

```
1  -- phpMyAdmin SQL Dump
2  -- version 5.0.4deb2+deb11u1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Hôte : localhost:3306
6  -- Génération le : mer. 13 déc. 2023 09:24
7  -- Version du serveur : 10.3.39-MariaDB-0+deb10u1-log
8  -- Version de PHP : 7.3.33-11+0-20230612.108+debian10-1.gbpif186d
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 START TRANSACTION;
12 SET time_zone = "+00:00";
13
14
15 /*140101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 /*140101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 /*140101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 /*140101 SET NAMES utf8mb4 */;
19
20 --
21 -- Base de données : 'godarde3'
22 --
23
24 -- -----
25
26 --
27 -- Structure de la table 'Adoptant'
28 --
29
30 CREATE TABLE `Adoptant` (
31   `numAdoptant` int(3) NOT NULL,
32   `nom` varchar(7) DEFAULT NULL,
33   `adresse` varchar(10) DEFAULT NULL,
34   `age` int(2) DEFAULT NULL,
35   `jardin` varchar(3) DEFAULT NULL,
36   `famille` varchar(3) DEFAULT NULL,
37   `nbEnfants` int(1) DEFAULT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
39
```

```
40 --
41 -- Chargement des données de la table 'Adoptant'
42 --
43
44 INSERT INTO `Adoptant` (`numAdoptant`, `nom`, `adresse`, `age`, `jardin`, `famille`, `nbEnfants`) VALUES
45 (291, 'Derbey', 'Vinay', 35, 'oui', 'oui', 2),
46 (292, 'Derby', 'Grenoble', 38, 'oui', 'oui', 2),
47 (293, 'Derbey', 'Valence', 40, 'non', 'oui', 2),
48 (294, 'Bernard', 'Valence', 30, 'oui', 'oui', 1),
49 (300, 'Tatoud', 'Eybens', 28, 'non', 'non', 0),
50 (308, 'Derbey', 'Vinay', 35, 'oui', 'oui', 2),
51 (309, 'Derbey', 'Vinay', 42, 'oui', 'oui', 2),
52 (310, 'Derbey', 'Valence', 30, 'non', 'oui', 2),
53 (311, 'Mandier', 'Valence', 50, 'oui', 'oui', 1),
54 (320, 'Durant', 'Grenoble', 40, 'oui', 'oui', 4),
55 (324, 'Martin', 'Pontcharra', 60, 'oui', 'oui', 0),
56 (325, 'Gacon', 'Vinay', 54, 'oui', 'oui', 2),
57 (326, 'Derbey', 'Vinay', 34, 'oui', 'oui', 2),
58 (327, 'Derbey', 'Valence', 35, 'non', 'oui', 2),
59 (328, 'Mandier', 'Valence', 45, 'oui', 'oui', 1),
60 (331, 'Mandier', 'Grenoble', 40, 'non', 'non', 0),
61 (334, 'Simonet', 'Grenoble', 50, 'oui', 'non', 0);
62
63 -- -----
64
65 --
66 -- Structure de la table 'Adoption'
67 --
68
69 CREATE TABLE `Adoption` (
70   `Adoption_Id` int(11) NOT NULL,
71   `animal_id` int(3) DEFAULT NULL,
72   `date` datetime DEFAULT NULL,
73   `numAdoptant` int(3) DEFAULT NULL,
74   `montantVerse` int(3) DEFAULT NULL,
75   `retour` varchar(15) DEFAULT NULL,
76   `duree_sejour` bigint(20) DEFAULT NULL,
77   `CoutSoin` float DEFAULT NULL
78 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
79
```

```
80 --
81 -- Chargement des données de la table 'Adoption'
82 --
83
84 INSERT INTO `Adoption` (`Adoption_Id`, `animal_id`, `date`, `numAdoptant`, `montantVerse`, `retour`, `duree_sejour`, `CoutSoin`) VALUES
85 (1, 356, '2021-12-25 00:00:00', 309, 120, NULL, 168, NULL),
86 (2, 352, '2022-06-01 00:00:00', 331, 130, 'non', 106, NULL),
87 (3, 345, '2022-12-16 00:00:00', 328, 130, 'non', 311, NULL),
88 (4, 343, '2022-04-04 00:00:00', 291, 70, NULL, 2, NULL),
89 (5, 341, '2023-01-17 00:00:00', 326, 60, 'non', 318, NULL),
90 (6, 336, '2022-09-18 00:00:00', 308, 50, 'non', 79, NULL),
91 (7, 334, '2021-09-16 00:00:00', 291, 70, NULL, 172, 42.4267),
92 (8, 330, '2021-12-19 00:00:00', 294, 120, 'non', 100, NULL),
93 (9, 327, '2022-12-25 00:00:00', 291, 80, 'non', 109, NULL),
94 (10, 324, '2022-11-24 00:00:00', 310, 120, NULL, 220, NULL),
95 (11, 323, '2021-04-29 00:00:00', 331, 120, 'non', 32, NULL),
96 (12, 318, '2021-06-19 00:00:00', 291, 50, 'non', 78, NULL),
97 (13, 316, '2021-03-14 00:00:00', 310, 80, NULL, 41, NULL),
98 (14, 315, '2021-08-12 00:00:00', 294, 130, 'non', 158, NULL),
99 (15, 314, '2022-01-28 00:00:00', 320, 50, 'non', 281, NULL),
100 (16, 313, '2022-09-06 00:00:00', 320, 50, NULL, 76, 27.84),
101 (17, 312, '2022-12-14 00:00:00', 292, 50, NULL, 28, NULL),
102 (18, 310, '2021-11-20 00:00:00', 325, 100, NULL, 102, NULL),
103 (19, 308, '2021-06-16 00:00:00', 325, 50, 'non', 35, NULL),
104 (20, 358, '2022-08-07 00:00:00', 292, 110, NULL, 182, NULL),
105 (21, 368, '2022-12-09 00:00:00', 291, 50, NULL, 45, NULL),
106 (22, 385, '2022-04-16 00:00:00', 320, 120, NULL, 70, NULL),
107 (23, 386, '2023-01-12 00:00:00', 334, 50, NULL, 216, NULL),
108 (24, 390, '2022-10-19 00:00:00', 311, 100, 'non', 107, NULL),
109 (25, 393, '2022-07-24 00:00:00', 324, 110, NULL, 67, NULL),
110 (26, 394, '2021-09-26 00:00:00', 320, 100, 'non', 24, NULL),
111 (27, 403, '2022-09-02 00:00:00', 291, 70, NULL, 36, NULL),
112 (28, 386, '2022-07-17 00:00:00', 300, 100, 'oui', 216, NULL),
113 (29, 345, '2022-11-25 00:00:00', 293, 70, 'oui', 311, NULL);
114
115 -----
116
```

```

164 --
165 -- Index pour les tables d charges
166 --
167 --
168 -- Index pour la table `Adoptant`
169 --
170 ALTER TABLE `Adoptant`
171 | ADD PRIMARY KEY (`numAdoptant`);
172
173 --
174 -- Index pour la table `Adoption`
175 --
176 ALTER TABLE `Adoption`
177 | ADD PRIMARY KEY (`Adoption_Id`),
178 | ADD KEY `animal_id` (`animal_id`),
179 | ADD KEY `numAdoptant` (`numAdoptant`);
180
181 --
182 -- Index pour la table `Animal`
183 --
184 ALTER TABLE `Animal`
185 | ADD PRIMARY KEY (`animal_id`);
186
187 --
188 -- AUTO_INCREMENT pour les tables d charges
189 --
190 --
191 --
192 -- AUTO_INCREMENT pour la table `Adoption`
193 --
194 ALTER TABLE `Adoption`
195 | MODIFY `Adoption_Id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=30;
196
197 -- Contraintes pour les tables d charges
198 --
199 --
200 -- Contraintes pour la table `Adoption`
201 --
202 ALTER TABLE `Adoption`
203 | ADD CONSTRAINT `Adoption_ibfk_1` FOREIGN KEY (`numAdoptant`) REFERENCES `Adoptant` (`numAdoptant`) ON DELETE CASCADE ON UPDATE CASCADE,
204 | ADD CONSTRAINT `Adoption_ibfk_2` FOREIGN KEY (`animal_id`) REFERENCES `Animal` (`animal_id`) ON DELETE CASCADE ON UPDATE CASCADE;
205 COMMIT;
206
207 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
208 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
209 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
210

```

```

117 --
118 -- Structure de la table `Animal`
119 --
120 CREATE TABLE `Animal` (
121 | `animal_id` int(3) NOT NULL,
122 | `raison_depart` varchar(10) DEFAULT NULL,
123 | `age` int(2) DEFAULT NULL,
124 | `etat` varchar(9) DEFAULT NULL,
125 | `espece` varchar(5) DEFAULT NULL,
126 | `taille` varchar(5) DEFAULT NULL,
127 | `poids_moyen_kg` int(2) DEFAULT NULL,
128 | `esperance_de_vie_ans` int(2) DEFAULT NULL,
129 | `race` varchar(15) DEFAULT NULL
130 |) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
131
132 -- D chargement des donn es de la table `Animal`
133 --
134
135 INSERT INTO `Animal` (`animal_id`, `raison_depart`, `age`, `etat`, `espece`, `taille`, `poids_moyen_kg`, `esperance_de_vie_ans`, `race`) VALUES
136 (308, 'adoption', 6, 'ok', 'chien', 'grand', 15, 8, 'Labrador'),
137 (310, 'adoption', 5, 'Bon', 'chat', 'moyen', 2, 11, 'Sphynx'),
138 (312, 'adoption', 2, 'bon', 'chien', 'grand', 12, 8, 'Dobermann'),
139 (313, 'adoption', 14, 'maltrait ', 'autre', 'petit', 1, 15, 'Perroquet'),
140 (314, 'adoption', 6, 'bon', 'chien', 'moyen', 6, 7, 'Fox Terrier'),
141 (315, 'adoption', 7, 'Bon', 'chien', 'grand', 10, 8, 'Berger allemand'),
142 (316, 'adoption', 1, 'Bon', 'chien', 'petit', 5, 6, 'Caniche'),
143 (318, 'adoption', 4, 'ok', 'chien', 'moyen', 5, 8, 'Malinois'),
144 (323, 'adoption', 3, 'ok', 'chien', 'grand', 15, 8, 'Labrador'),
145 (324, 'adoption', 5, 'bon', 'chien', 'grand', 12, 8, 'Dobermann'),
146 (327, 'adoption', 0, 'maltrait ', 'chien', 'petit', 1, 3, 'Chihuahua'),
147 (330, 'adoption', 3, 'ok', 'chat', 'moyen', 2, 10, 'Persan'),
148 (334, 'adoption', 1, 'Bon', 'chien', 'petit', 1, 3, 'Chihuahua'),
149 (336, 'adoption', 0, 'maltrait ', 'chien', 'moyen', 6, 7, 'Fox Terrier'),
150 (341, 'adoption', 6, 'Bon', 'chien', 'petit', 3, 8, 'Bichon'),
151 (343, 'adoption', 0, 'maltrait ', 'chien', 'petit', 3, 8, 'Bichon'),
152 (345, 'adoption', 12, 'Bon', 'autre', 'petit', 1, 15, 'Perroquet'),
153 (352, 'adoption', 0, 'Bon', 'chien', 'petit', 3, 8, 'Bichon'),
154 (350, 'adoption', 7, 'ok', 'chien', 'grand', 15, 8, 'Labrador'),
155 (358, 'adoption', 1, 'mauvais', 'chien', 'moyen', 6, 8, ' pagneul'),
156 (360, 'adoption', 5, 'bon', 'chien', 'moyen', 6, 8, 'Bulldog'),
157 (385, 'adoption', 3, 'ok', 'chien', 'moyen', 6, 8, 'Bulldog'),
158 (386, 'adoption', 6, 'Bien', 'chien', 'moyen', 5, 8, 'Malinois'),
159 (390, 'adoption', 0, 'maltrait ', 'chien', 'petit', 1, 3, 'Chihuahua'),
160 (393, 'adoption', 1, 'ok', 'chien', 'petit', 3, 8, 'Bichon'),
161 (394, 'adoption', 4, 'Bien', 'chien', 'moyen', 5, 8, 'Malinois'),
162 (403, 'adoption', 7, 'maltrait ', 'chien', 'moyen', 7, 8, 'Griffon');
163

```