



# **Projet d'entrepôt de données : Zoo**

Groupe 4 :

ETTAHIRI Sofian : 21901227  
HADDAD Gatien : 21901879  
POINTEAU Gabrielle : 21917975

2022 - 2023

# Table des matières

<b>1</b>	<b>Analyse</b>	<b>3</b>
1.1	Objectifs généraux de notre Zoo . . . . .	3
1.2	Actions/opérations nécessaire . . . . .	3
1.3	3 traitements possibles pour chaque action . . . . .	4
1.4	Ordonner les actions par leurs rentabilité potentielle . . . . .	5
<b>2</b>	<b>Conception</b>	<b>5</b>
2.1	Identifier les deux actions importantes . . . . .	5
2.2	Conception des data-mart pour chaque actions . . . . .	5
2.2.1	Action importante . . . . .	5
2.2.2	Action moins importante . . . . .	6
2.2.3	Liste des faits pour chaque actions . . . . .	7
2.2.4	Mesure additive, semi-additive, non-additive . . . . .	7
2.3	Dimensions nécessaires aux deux modèles (au moins 5) . . . . .	7
2.4	Est ce qu'on peut répondre aux traitements avec notre modèle? . . . . .	10
2.5	Exemple d'instances . . . . .	11
2.6	Estimation de la taille sur 12 mois et est-ce que c'est raisonnable? . . . . .	11
<b>3</b>	<b>Implémentation et le requêtage</b>	<b>12</b>
3.1	Implementation . . . . .	12
3.2	Requetes analytiques . . . . .	12
3.3	Vues matérialisées . . . . .	12

---

## PROJET de Entrepôt de données : Zoo

---

# 1 Analyse

## 1.1 Objectifs généraux de notre Zoo

Nous avons décidé de nous intéresser à la gestion d'un Zoo qui aura pour but de garantir au maximum le confort de nos animaux pour qu'ils se sentent le plus à l'aise possible. C'est donc dans cette optique qu'il nous faut améliorer le chiffre d'affaires du zoo, pour utiliser cet argent pour améliorer les infrastructures ou encore la qualité de la nourriture des animaux etc...

Pour cela on va donc essayer de fluidifier l'entrée au parc en ouvrant plus de caisse par exemple. Le zoo proposera également plusieurs stands de nourritures ou de souvenirs disponible tout au long du parcours du visiteur ou encore la présence de boîtes de donations au niveau des enclos des animaux qui seront entièrement dédiés à l'amélioration du confort de ces derniers.

C'est pour cela qu'avoir accès aux informations comme les lieux les plus fréquentés, l'heure de passage des visiteurs dans certains endroits nous permettra d'établir un plan de stratégie pour faire évoluer le zoo.

## 1.2 Actions/opérations nécessaire

a) **Parcours visiteurs** : Les visiteurs ont un badge et un système de détection au sol pour qu'on repère que le visiteur vient de passer devant l'enclos d'un certain animal. De cette manière nous serons informés du parcours d'un même visiteur et des enclos par lesquels il est passé. Cela nous permet également de savoir combien de personnes sont venues voir l'enclos au cours de la journée ou au cours d'une plus grande période. On peut en outre mettre en place une entrée enfant pour éventuellement mettre plus d'activités sur des stands que les enfants aiment (événement enfantin, glace, jouet, peluche, autre...).

b) **Dons des visiteurs** : Les visiteurs ont la possibilité de faire un don aux différents enclos du zoo (en plus de payer leur entrée). On peut considérer que si jamais à la fin de la journée la boîte de don est pleine, alors le lendemain on en rajoutera une. On peut également analyser quand les visiteurs donnent de l'argent, si jamais la boîte est vide, est-ce qu'ils ont toujours envie de donner ou pas : être le premier à donner. Et à l'inverse si la boîte est pleine : Est-ce qu'ils pensent qu'il y a déjà trop d'argent pour cet enclos.

Le but étant qu'on veut savoir à la fin d'une certaine période quel enclos a été le plus visité et donc par extension quel animal a été le plus attractif pour les visiteurs. Cela sous-entend qu'on considère que l'enclos qui a le plus d'argent est l'enclos où les animaux sont les plus attrayants.

c) **Entrées visiteurs** : Chaque visiteur rentrant dans le zoo a préalablement acheté son ticket, on peut donc tracer les entrées de visiteurs en fonction de l'heure, du jour, vacances scolaires, si elles viennent en groupes, etc... On peut également compter le nombre de personnes, et on pourra ajuster le nombre de caisses selon l'affluence.

d) **Stands nourriture, boissons, souvenirs...** : Les visiteurs ont la possibilité de pouvoir manger et boire dans le zoo et donc ils ont accès à des stands de nourriture et de boisson tout au long de leur visite. L'intérêt serait donc d'analyser les achats des visiteurs dans les différents stands, pour regarder ceux qui ont une grande demande ou pas. Cela permettrait de limiter la queue dans certains stands, ou de rajouter des stands dans certaines zones à grande affluence. De plus, cela donne la possibilité d'augmenter un article dans un magasin pour faire plus de profit pour l'amélioration du zoo en matière de soins des animaux, infrastructures etc...

### 1.3 3 traitements possibles pour chaque action

a) **Parcours visiteurs** :

- Pourcentage de visiteurs allant voir chaque enclos
- Classement des dernières heures de visite de chaque enclos
- Par profil de visiteur quels enclos il regarde en dernier.

b) **Dons** :

- Donner le montant total des dons par enclos et par heure
- Temps moyen de remplissage de chaque boîte (en heure)
- Classements des enclos par nombre de dons (nombre de fois que les visiteurs donnent de l'argent)
- Classements des enclos par nombre d'argents donnés (somme total de l'argent donnée)

c) **Entrées** : .

- Jours les plus visités
- Classement des heures les plus visitées en fonction des jours
- Taille moyenne des groupes personnes selon les heures
- Voir les régions dont viennent le moins de visiteurs et augmenter les publicités dans ces régions

d) **Nourritures, Boissons, Souvenirs** :

- A quelle heure les utilisateurs achètent ils le plus ?
- Donner le montant total des ventes par produit, par jour et par heure
- Classement des boutiques les plus prisait par type de stand

## 1.4 Ordonner les actions par leurs rentabilité potentielle

- 1) *Nourritures, Boissons, Souvenirs* : Leur étude possède un potentiel de rentabilité assez haut. En effet, à l'aide des requêtes permettant d'identifier les "heures de pointe" de certains food truck, etc, on pourrait augmenter/réduire leur production et/ou modifier leur emplacement pour maximiser le rendement.
- 2) *Parcours visiteurs* : Les statistiques sur parcours des visiteurs sont également importantes pour optimiser l'agencement des enclos. On pourra identifier les enclos où les dons sont élevés (en matière de pourcentage de visiteurs) et placés ces enclos à des emplacements stratégiques.
- 3) *Entrées* : L'étude concernant les entrées de visiteurs permettrait de réguler le personnel selon l'affluence, mais le gain potentiel est inférieur à celui des actions citées ci-dessus.
- 4) *Dons* : Les requêtes sur les dons sont intéressantes, mais les profits générés par l'exploitation de ces données seraient négligeables.

## 2 Conception

La conception de l'entrepôt dépendra exclusivement des traitements que vous avez indiqués.

### 2.1 Identifier les deux actions importantes

Nous choisissons donc d'analyser les actions sur les *stands de nourriture* et le *parcours des visiteurs* pour améliorer

### 2.2 Conception des data-mart pour chaque actions

#### 2.2.1 Action importante

Nous pouvons voir sur la figure 1 à la page 6, le modèle en étoile pour l'action sur le parcours des visiteurs. Nous avons choisi de considérer le modèle comme un modèle record-update, en stockant le profil, l'enclos, la mesure du nombre de visites par enclos et par visiteur.

Ainsi, si un visiteur revient devant un enclos, on aura une mise à jour sur l'attribut *nombre\_visite* en l'incrémentant de 1. Ainsi que sur l'heure de la visite. On aura donc 1 ligne pour chaque enclos et chaque visiteur.

Nous avons été contraints d'enlever la table animale du départ car comme on ne pouvait pas choisir un modèle transactionnel nous ne pouvions pas garder les animaux. Car lorsque ont mettrait à jour une ligne pour un animal de l'enclos il faudrait le faire sur toute.

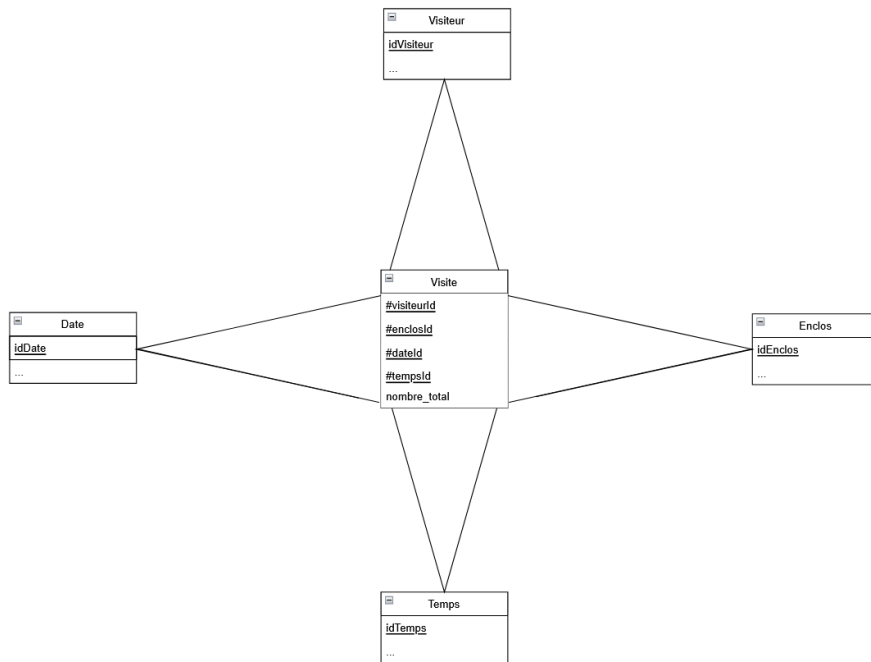


FIGURE 1 – DataMart de l'action sur le parcours des visiteurs

### 2.2.2 Action moins importante

On voit sur la figure 2 à la page 7 le modèle en étoile de l'action concernant les stands de nourriture, de boisson et de souvenirs.

Ainsi on a la description détaillée de la table de fait : Ventes. On a les différents identifiants des 6 dimensions, ainsi que la quantité d'achat effectué du produit et pour finir le prix total de l'achat. Nous avons considéré que ce modèle serait transactionnel. Ainsi à chaque nouvelle vente on affiche une nouvelle ligne.

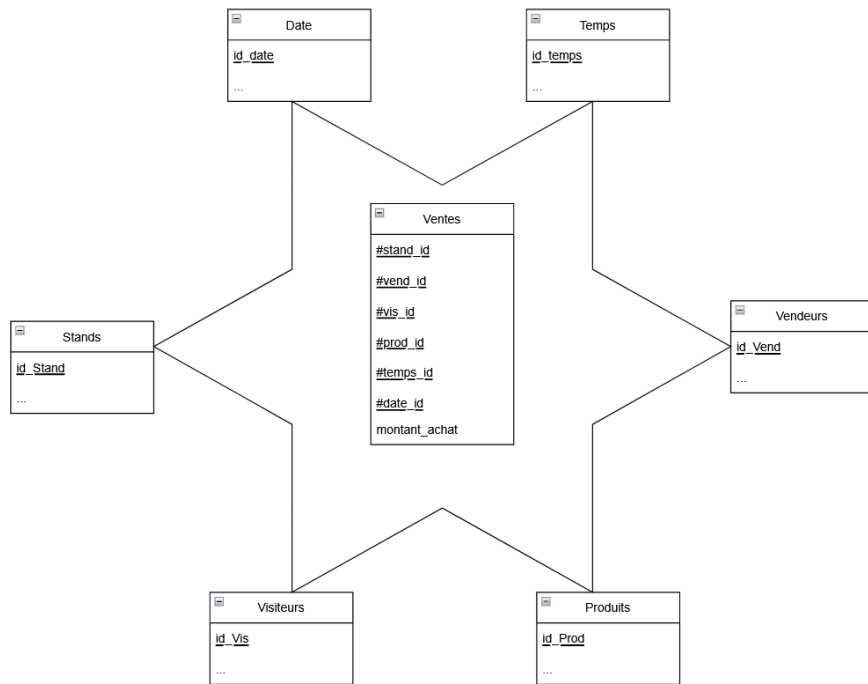


FIGURE 2 – DataMart de l'action sur les stands de nourriture, boissons et souvenirs

### 2.2.3 Liste des faits pour chaque actions

- Pour les stands de nourriture, la mesure est chiffre d'affaires du stand, elle est additive, on peut sommer le montant des achats par date, par visiteur, ou par heure.
- Pour le parcours de chaque visiteur, la mesure est le nombre de fois que chaque profil de visiteur sont venu voir un enclos.

### 2.2.4 Mesure additive, semi-additive, non-additive

- Pour les stands de nourriture, la mesure est additive car on peut sommer le nombre de ventes par heure, par visiteur, etc...
- Pour le parcours de chaque visiteur, la mesure est également additive, car on peut compter le nombre de visites de chaque enclos par profil.

## 2.3 Dimensions nécessaires aux deux modèles (au moins 5)

- *Stands* : Dans la figure 3 à la page 9 nous pouvons voir les dimensions nécessaires au modèle. Nous devons définir certains attributs qui ne sont pas évidents à comprendre.

\* *Vendeurs* : Définissons l'attribut *type\_contrat* comme une énumération correspondant pour quelles tâches le vendeur a été engagée. On aura donc la liste suivante : {alimentation, sucrerie, souvenir}.

Certains vendeurs pourraient avoir plusieurs types de contrats, cependant nous choisissons le fait qu'un vendeur ne peut avoir qu'un contrat pour pouvoir avoir la durée de son contrat. Si jamais ce vendeur, a un autre contrat on devra recréer une ligne avec un autre identifiant.

Par ailleurs l'attribut *numero\_tel\_zoo* est le numéro de talki walki du vendeur pour pouvoir le contacter dans le zoo.

Tous les autres attributs sont par leur nom, explicite à comprendre.

\* Stands : L'attribut *type\_Stand* est une énumération des trois types de stands possibles dans le zoo : {alimentation, sucrerie, souvenirs}.

De plus, l'attribut *zone* quant à lui, représente les différentes zones du zoo. C'est donc également une énumération qui contiendrait par exemple {Océanique, Afrique,...}.

On peut également savoir si le stand est fermé par l'attribut *is\_ferme*.

Par ailleurs, les autres attributs sont assez explicites par leur nom.

\* Date : Pour cette dimension, nous avons choisi de détailler un maximum les éléments de date avec l'attribut *nom\_semaine* qui est une énumération des jours dans la semaine telle que {lundi, mardi, mercredi,...}.

De plus, le *numero\_mois* et le *nom\_mois* sont respectivement les indicateurs du mois tel que {1,2,3,...} et {Janvier, Février, Mars, ...}.

Nous avons également mis des indicateurs de si l'établissement est fermé, en vacances, en weekend ou si c'est un jour férié pour pouvoir directement faire des statistiques sur ces données.

\* Produits : Pour la dimension produit, l'attribut *type\_prod* est comme l'attribut *type\_Stand* dans la dimension Stands et également fait de la même manière pour l'attribut *type\_stockage*.

Et l'attribut *auto\_retour* sert à autoriser le retour du produit, par exemple le produit peut être rendu pour mauvaise taille.

\* Visiteurs : Pour cette table les attributs sont plutôt explicites, on a pas besoin de les définir.

\* Temps : Nous avons choisi de mettre l'heure en heure anglaise de façon à ce que ce soit de 00 :00 à 12 :00. Ainsi l'attribut *AM\_PM* nous indique dans quelle tranche de la journée on est.

De plus, l'attribut *creneau* est une énumération de la période de la journée dans laquelle on est. Ainsi, on a {matin, midi, après-midi, soir, nuit}, qui correspondrait aux tranches horaires suivantes {8h-11h30, 11h30-14h, 14h-17h, 17h-20h, 20h-8h}.

Nous venons donc d'expliquer tous les attributs des dimensions pour l'action des stands présents dans le zoo.



Vendeurs	Stands	Date	Produits	Visiteurs	Temps
<u>id_Vend</u>	<u>id_Stand</u>	<u>id_date</u>	<u>id_Prod</u>	<u>id_Vis</u>	<u>id_temps</u>
nom_vend	nom_Stand	date_entiere	nom_prod	nom_vis	temps_entier
prenom_vend	type_Stand	numero_jour	marque	prenom_vis	heures
age	longitude	nom_semaine	prix	age	minutes
sexe	latitude	numero_mois	type_prod	sexe	secondes
salaire	zone	nom_mois	tranche_age	adresse_habit	AM_PM
type_contrat	is_ferme	annee	type_stockage	ville_habit	creneau
duree_contrat	nb_vente_annee_passe	saison	code_barre	departement_habit	is_derniere_heure
numero_tel_zoo	nb_article_différent	trimestre	auto_retour	pays_habit	horaire_ouverture
mobile	date_ouverture	is_weekend	taille	nationalite	horaire_fermeture
mail		is_vacances	poids	numero_tel	is_ferme
		is_jour_ferie	couleur	mail	
		is_ferme		nb_enfant	

FIGURE 3 – Dimensions de l'action sur les stands de nourriture, boissons et souvenirs

- *Parcours des visiteurs* : Dans la figure 4 à la page 10 nous pouvons voir les dimensions nécessaires au modèle. Nous devons définir certains attributs qui ne sont pas évidents à comprendre.

Cependant, les attributs des dimensions Temps, Date et Visiteurs sont les mêmes que pour l'action précédente donc nous n'avons pas besoin de les redéfinir.

\* Enclos : Pour cette dimension, nous pouvons expliciter l'attribut *typeEnclos* est une énumération des types d'enclos par exemple {neige, sec, rocher, arbre,...}. Ainsi, chaque enclos peut avoir plusieurs types d'enclos, par exemple les singes, on besoin de plusieurs arbres, tout en ayant un sol avec une petite pelouse, alors ils auront la liste {arbre, herbe courte}.

L'attribut *zoneEnclos* est le même que pour l'attribut *zone* dans l'action précédente dans la dimension Stand.

De plus, l'attribut *distanceEntree* permet de savoir à quelle distance l'enclos est de l'entrée.

On vient alors d'explicitier les attributs pour l'action sur le parcours des visiteurs.

Temps	Enclos	Visiteurs	Date
<u>idTemps</u>	<u>idEnclos</u>	<u>idVis</u>	<u>idDate</u>
tempsEntier	nomEnclos	ageVis	dateEntiere
heures	distanceEntree	sexeVis	numeroJour
minutes	capacité	adresseHabit	nomSemaine
secondes	zoneEnclos	villeHabit	numeroMois
AM_PM	typeEnclos	departementHabit	nomMois
creneau	longitudeEnclos	paysHabit	annee
isDerniereHeure	latitudeEnclos	nationalite	saison
horaireOuverture	nbAnimaux	nbEnfant	trimestre
horaireFermeture	nbEspeces		isWeekend
isFermeTemps			isVacances
			isJourFerie
			isFermeDate

FIGURE 4 – Dimensions de l'action sur le parcours des visiteurs

## 2.4 Est ce qu'on peut répondre aux traitements avec notre modèle ?

Reprenons les traitements pour les deux actions que nous avons choisis. Pour rappel nous avions les traitements suivants pour le parcours des visiteurs :

### Parcours des visiteurs :

*Pourcentage de visiteurs allant voir chaque enclos :* Pour chaque profil de visiteur, nous pouvons voir, pour chaque enclos, le nombre de fois qu'un enclos a été visité, alors en sélectionnant tous les profils et en additionnant le nombre de fois que les visiteurs sont venus on peut faire un pourcentage des visites.

*Classement des dernieres heures de visite de chaque enclos :* Pour le deuxième, on a le temps de la dernière visite qui est stocké lorsqu'on arrive devant un enclos alors on peut regarder le temps pour chaque enclos.

*Par profil de visiteur quels enclos il regarde en dernier. :* On stocke les profils des visiteurs ainsi que les enclos qu'ils vont regarder on peut donc sur certaines caractéristiques du profil, savoir quel type vient voir quel animal en dernier.

## Nourritures, Boissons, Souvenirs :

A quelle heure les utilisateurs achètent ils le plus ? : Nous pouvons le voir en faisant un classement par heure des achats dans les stands pour voir le pic d'affluence.

Donner le montant total des ventes par produit, par jour et par heure : On additionne tous les prix de chaque lignes par jour par heure et par produit.

Classement des boutiques les plus prisés par type de stand : On cherche a savoir parmi les 3 types de stand lesquels boutiques sont les plus prisés (ou il y a la plus de vente). Donc on peut répondre à ce traitement en récupérant toutes les lignes par boutiques et par type de stand et additionnels le prix total des ventes.

Nous avons donc repris les traitements et avons à regarder si pour chacun d'entre eux notre modèle pouvait répondre a chaque traitement. C'était bien le cas nous pouvons maintenant proposer des exemples d'instances.

## 2.5 Exemple d'instances

Comme nous avons modifié les éléments des tables on peut directement trouver des instances dans le fichier *tuples.sql*

## 2.6 Estimation de la taille sur 12 mois et est-ce que c'est raisonnable ?

On estime le nombre de visiteurs annuel à 500'000.

- Ventes : si chaque client en moyenne achète cinq produits différents, que ce soit nourriture, boisson ou sucrerie, on peut s'attendre à 500'000 lignes \* 5 pour les Ventes, à quoi on vient rajouter les 500'000 clients, les 200 employés sur 12 mois, On prend également en compte les tables Date et Temps, avec respectivement 365 et 86'400 (3600\*24) lignes, ce qui fait environ 3,1 millions de lignes sur 12 mois
- Visites : Sachant que le parc compte environ une cinquantaine d'enclos, on estime que chaque visiteur va visiter en moyenne 35 enclos, 500'000 \* 35, on se retrouve avec plus de 18 millions de lignes sur 12 mois.

Ces deux tailles sont raisonnables, car elles sont de l'ordre du million, le stockage nécessaire n'est alors pas démesuré

## 3 Implémentation et le requêtage

### 3.1 Implementation

Nous avons donc implémenté les deux modèles que nous avons expliqué plus tôt. Qui sont en fichier joint au rapport. Nous avons alors les tables dans le fichier *tables.sql*. Et les différents tuples dans le fichier *tuples.sql*.

### 3.2 Requetes analytiques

Nous avons alors écrit une 10ème de requête pour les deux modèles que l'on a dans le fichier *vente\_requete.sql* et *visite\_requete.sql*.

### 3.3 Vues matérialisées

Voici donc quelques vues matérialisées pour répondre à nos requêtes.

```
-- VUES MATERIALISEES
```

```
CREATE MATERIALIZED VIEW regionVisiteur (idVisiteur, region) AS
SELECT idVisiteur, region
FROM Visiteur
GROUP BY idVisiteur, region;
```

```
CREATE MATERIALIZED VIEW ageEnclos (age, idEnclos) AS
SELECT age, idEnclos
FROM Visiteurs
JOIN Visites ON Visiteurs.idVisiteur = Visites.visiteurId
JOIN Enclos ON Visites.enclosId = Enclos.idEnclos
GROUP BY age, idEnclos;
```

```
CREATE MATERIALIZED VIEW typeStandVentes(idStand,typeStand) AS
SELECT idStand, typeStand
FROM Stand, Ventes
WHERE Stand.idStand= Ventes.standId;
```

```
CREATE MATERIALIZED View prixProduits(idProduit,prixVentes) as
SELECT idProduit, prixVentes
FROM Produit, Ventes
WHERE Produit.idProduit = Ventes.produitId;
```