

UNIVERSITÉ DE MONTPELLIER

PROJET ENTREPÔT DE DONNÉES ET BIG DATA

UBER-EATS

Uber **Eats**

ASMA Redouane,
LEHOUAOUI Sara,
ZEDDAM Lylia



Table des matières

1	Introduction	2
1.1	Présentation	2
1.2	Objectifs	3
2	Conception	4
2.1	Data-Mart	4
2.2	Mesures	7
2.3	Traitements proposés	7
2.4	Estimation	7
3	Implémentation et requêtage	8
3.1	Vues virtuelles	8
3.2	Requêtes analytiques	8
3.2.1	Facture Client	9
3.2.2	Livreur	13
3.3	Vues matérielles	15
4	Conclusion	16

Chapitre 1

Introduction

1.1 Présentation

Le sujet de notre étude se porte sur Uber-Eats, c'est un service de livraison de plats cuisinés lancé par Uber en 2015 et basé en Californie. C'est une plateforme de commerce électronique qui met en relation les différents types de restauration, les clients et les livreurs.

Cette entreprise opère dans le monde entier et son application est l'une des applications les plus utilisées dans le domaine de la livraison

Les commandes sont prises via l'application ou leur site web auprès des restaurants partenaires et sont livrées par des coursiers indépendants (livreurs).

Le client choisit un restaurant parmi les différents restaurants proposés puis ajoute un plat ou plusieurs à son panier et passe sa commande.

Suite à la validation du panier, les frais de service (10 % du montant de la commande) et de livraison (en fonction de la localisation géographique) sont ajoutés au montant de la commande.

Si le client a souscrit au programme de fidélité (Pass Uber-Eats : livraison offerte sur toute commande d'au moins 12 euros passée auprès des restaurants admissibles avec une souscription de 5.90 par mois), il ne sera plus concerné par les frais de livraison.

Si le client possède un code promo reçu auparavant par Uber-Eats (Uber-Eats distribue des codes promo tout au long de l'année à ses clients selon son algorithme), le montant de la promotion s'appliquera au prix finale de la commande et sera déduite directement.

Lorsque que le paiement est effectué par le client, le livreur reçoit une notification de prise en charge de cette commande. S'il décide de l'accepter, il récupérera le plat du restaurant et l'apportera au client.

Les revenus de l'entreprise se constituent de frais de service (expliqués auparavant) et d'une commission de 30% en moyenne sur la facture générée par le restaurant partenaire.

1.2 Objectifs

L'objectif de cette étude de marché sera d'analyser le chiffre d'affaire en prenant en considération les commandes validées avec des codes promotionnels. Mais aussi en s'intéressant au nombre de livraisons effectuées par les livreurs dans le but d'améliorer le chiffre d'affaire.

Pour atteindre notre but, les informations nécessaires qui nous seront utiles seraient d'étudier la rentabilité des codes promotionnels et celui du pass Uber-Eats. Mais encore la gestion des livreurs actifs pour assurer les livraisons selon les différentes circonstances météorologiques et leur localisation.

Nous nous intéresserons également aux restaurants ayant effectué le plus de commandes afin de les mettre en avant sur la plateforme.

Chapitre 2

Conception

2.1 Data-Mart

Nous avons choisi deux tables de faits importantes à analyser.
La première table de fait nommée FactureClient qui est une modélisation détaillée de la commande. Elle regroupe les dimensions client, restaurant, promotion, catégorie d'âge, date et le temps.
Voir figure ci-dessous

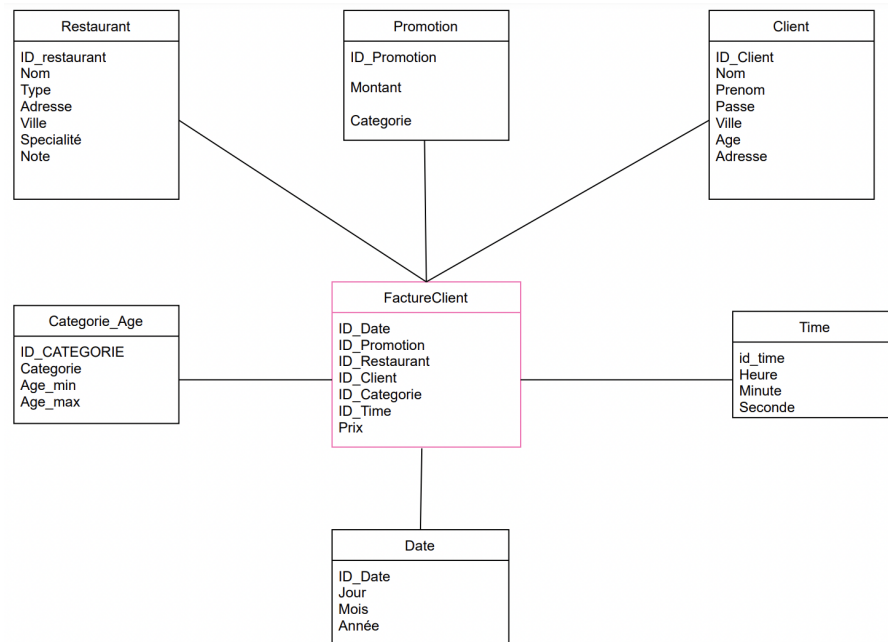


Figure 1.3 - DATA-MART FactureClient

La seconde nommée Livreur qui est une modélisation moins détaillée et instantanée. Elle regroupe les dimensions localisation, climat, catégorie, date et temps. Voir la figure ci-dessous

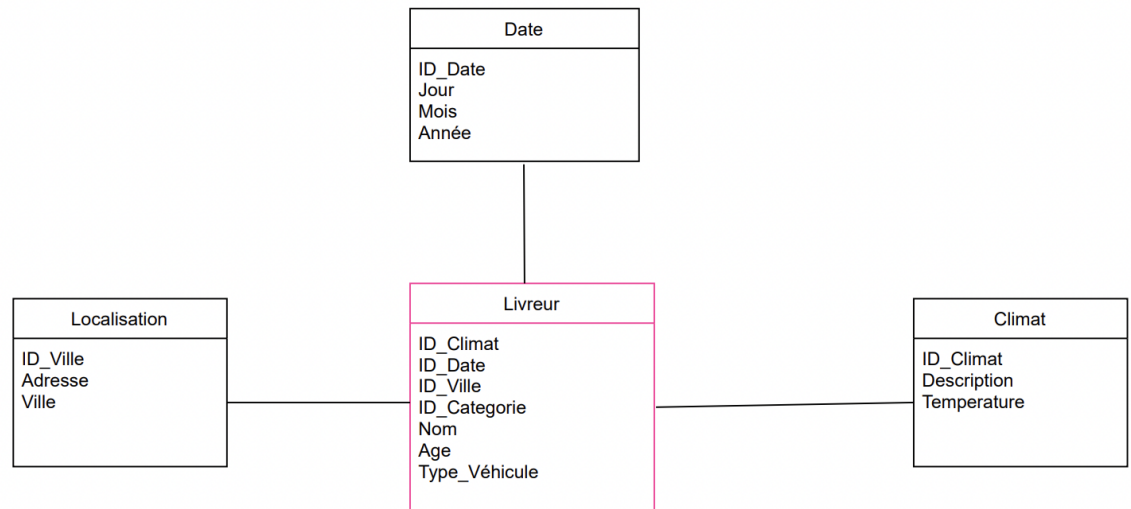


Figure 2.3 - DATA-MART Livreur

Les deux tables de faits ont des dimension en commun Date,Time et catégorieAge de ce fait notre Data-mart sera modéliser en constellation. Voir figure ci-dessous

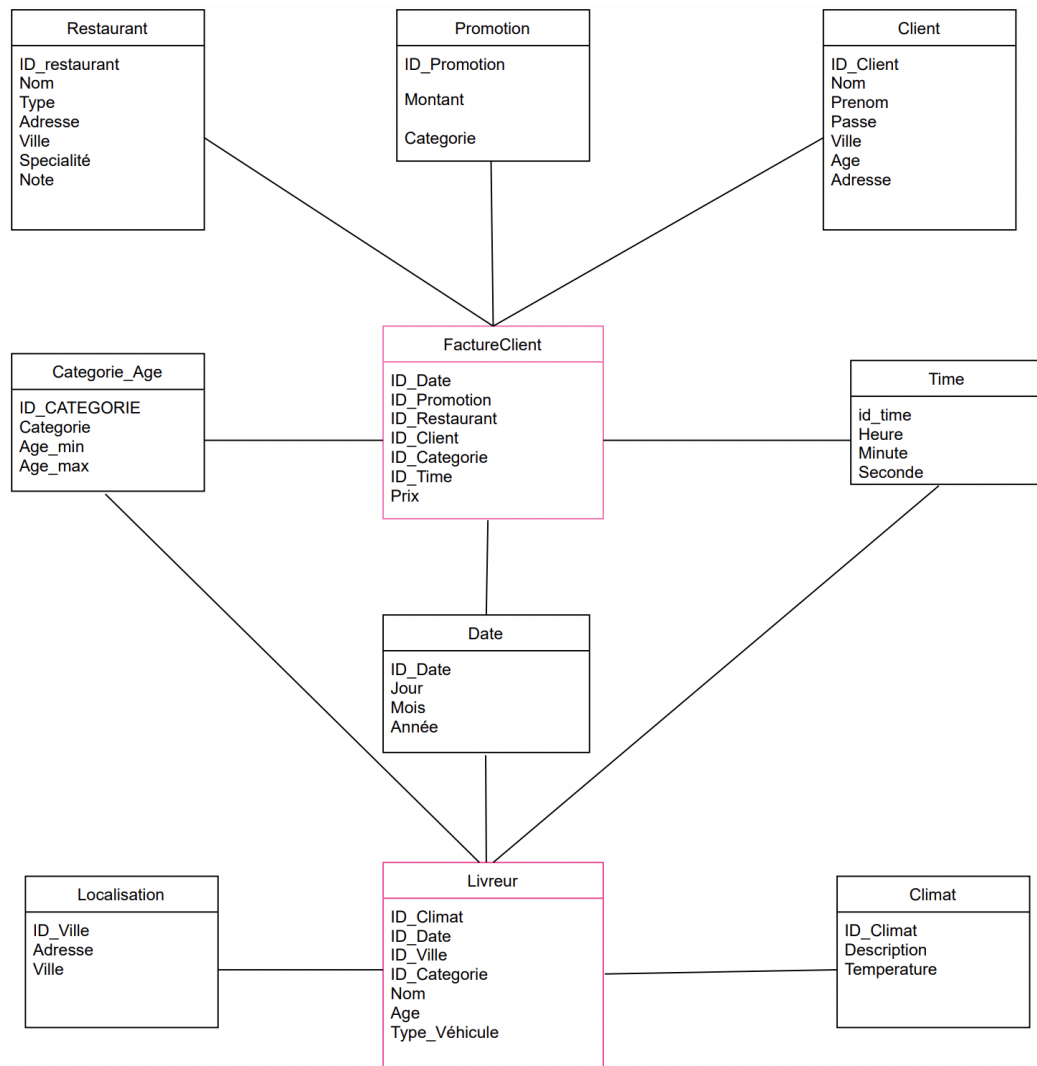


Figure 3.3 - DATA-MART

2.2 Mesures

La table de fait FactureClient contient une mesure prix qui est une mesure qu'on peut additionner et donc additive.

La table de fait Livreur contient des mesures qui donnent des détails sur le profil du livreur (Nom, Âge et typeVehicule), ce sont des descriptions, des chaînes de caractère et donc des mesures non additives.

2.3 Traitements proposés

Les différents traitements proposés :

1. Table Facture client :
 - Le nombre de commande par catégorie de promotion
 - Le nombre de commande sans promotion
 - Le nombre de commande par pass Uber-Eats
 - Le nombre de commande par catégorie d'âge
 - Le nombre de commande par restaurant par mois
 - Le nombre de commande par mois
2. Table Livreur :
 - Le nombre de livreurs par climat(pluvieux,ensoleillé..)
 - Le nombre de livreurs par catégorie d'âge.
 - Le nombre de livreurs par localisation

2.4 Estimation

La taille des tables de l'entrepôt (en terme du nombre de lignes) sur 12 mois
Au cours de l'année 2020, on a estimé la taille de la table Restaurant à près de 25000 lignes , la table Client à 15 millions de lignes en France. Environ 35.000 livreurs utilisent l'application Uber Eats. Ces personnes d'âge moyen de 25 ans y restent autour de dix mois. UberEats aide les restaurants à vendre 1.6 millions de repas supplémentaires par semaine ce qui fait 83.2 millions par an.

Chapitre 3

Implémentation et requêtage

3.1 Vues virtuelles

Notre Data-Mart contient des dimensions communes que nous allons représenter par des vues virtuelles pour regrouper les informations au sein d'une même entité.

Les vues sont généralement utilisés pour accorder des privilèges ou limiter l'accès à l'utilisateur ou bien pour joindre deux tables.

Dans notre cas, nous l'utiliserons plutôt pour la dernière raison.

Voici les vues des dimensions partagées date, catégorie et temps.

```
CREATE VIEW VIEW_DATE
AS SELECT A.id_date
FROM DATEE A, DATES B WHERE A.id_DATE= B.id_DATE;

CREATE VIEW VIEW_CATEGORIE
AS SELECT c.id_categorie
FROM CATEGORIE_AGE c, CATEGORIE d WHERE c.ID_categorie = d.ID_categorie;

CREATE VIEW VIEW_TIME
AS SELECT d.id_time
FROM TIMEE d, TIMES e WHERE d.id_time= e.id_time;
```

3.2 Requêtes analytiques

Nous avons implémenté les différentes propositions vues ci-dessus avec des requêtes SQL. Nous analyserons par la suite les résultats obtenus afin de proposer des solutions pour atteindre notre objectif.

3.2.1 Facture Client

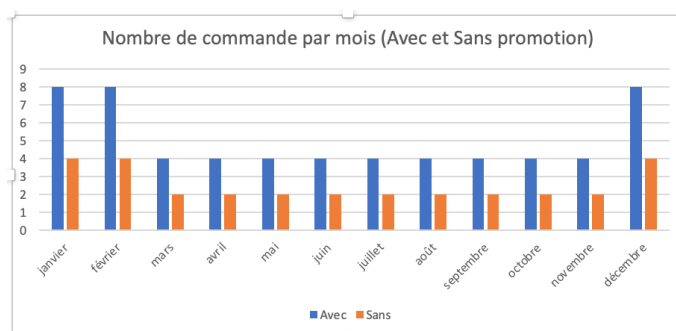
- Le nombre de commande par catégorie de promotion par mois

```
#NOMBRE DE COMMANDE par categorie de promotion par mois
SELECT COUNT(*),DATEE.mois,PROMOTION.categorie
FROM FACTURE_CLIENT, PROMOTION, DATEE
WHERE FACTURE_CLIENT.id_promotion= PROMOTION.id_promotion
AND FACTURE_CLIENT.id_DATE = DATEE.mois
GROUP BY PROMOTION.categorie, DATEE.mois;
```

- Le nombre de commande sans promotion par mois

```
#NOMBRE DE COMMANDE SANS PROMOTION par mois
SELECT COUNT(*),DATEE.mois
FROM FACTURE_CLIENT, DATEE
WHERE FACTURE_CLIENT.id_DATE = DATEE.mois
AND FACTURE_CLIENT.id_promotion is null
GROUP BY DATEE.mois;
```

On a regroupé le résultat de ces 2 requêtes sous forme d'élément graphique pour bien voir la différence.



Histogramme représentant le nombre de commande par mois avec et sans promotion

Nous observons que le nombre de commande effectué avec des codes promotionnels est plus important que celui des commandes sans utilisation de code promotionnel.

Suite à cette analyse nous proposons d'envoyer plus de code promotionnel au client. Afin de cibler la catégorie de client à qui nous enverrons ces codes promotionnels et sur quelle période de l'année, nous allons nous pencher vers des requêtes plus spécifiques.

— Le nombre de commande par catégorie d'âge

```
#nb de cmd par categorie d age
SELECT COUNT (*), CATEGORIE_AGE.categorie
FROM FACTURE_CLIENT, CATEGORIE_AGE
WHERE FACTURE_CLIENT.id_client = CATEGORIE_AGE.id_categorie
GROUP BY CATEGORIE_AGE.categorie;
```

Nous avons mis les résultats sous forme d'histogramme :

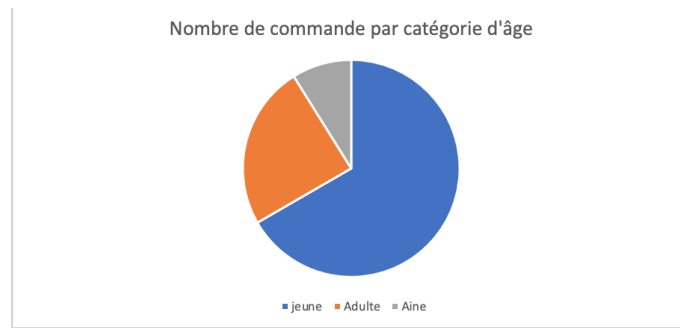


Diagramme représentant Nombre de commande par catégorie d'âge

Le nombre de commandes est plus important chez les jeunes d'après le diagramme. Pour les encourager à commander davantage, on pourra envoyer plus de code promotionnel pour cette tranche d'âge.

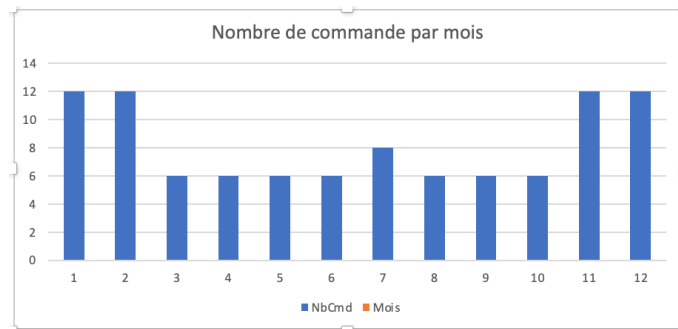
Pour la catégorie d'âge qui achète le moins, les aînées, on pourra leur soumettre des questionnaires de satisfaction pour savoir si améliorer certains services les aidera à utiliser plus l'application.

Cette catégorie d'âge pourrait ne pas utiliser aussi souvent l'application par manque d'expérience et de connaissance envers ces nouvelles technologies, on leur proposera dans ce cas des tutoriels pour mieux les guider sur l'utilisation de l'application.

— Le nombre de commande par mois

```
#NOMBRE DE COMMANDE PAR Mois
SELECT COUNT (*), DATEE.mois
FROM DATEE, FACTURE_CLIENT
WHERE FACTURE_CLIENT.id_DATE = DATEE.mois
GROUP BY DATEE.mois;
```

Résultats de la requête :



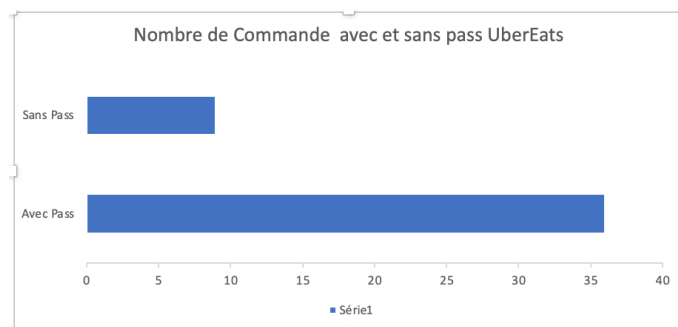
Histogramme montrant le nombre de commande par mois

Nous remarquons d'après la figure ci-dessus que le nombre de commandes est plus grand en hiver (novembre, décembre, janvier, février) et bas pendant le reste de l'année. Ce qui est logique vu que les personnes veulent plus profiter des terrasses et des restaurants sur place lorsqu'il fait beau et restent chez eux dans le cas contraire. Il serait donc intéressant de proposer des promotions dans cette période de l'année où les commandes sont basses (et surtout en été) en vue d'inviter les clients à commander chez Uber-Eats.

— Le nombre de commande par pass UberEats

```
#NOMBRE DE COMMANDE par PASS
SELECT COUNT (*), CLIENT.pass
FROM FACTURE_CLIENT, CLIENT
WHERE FACTURE_CLIENT.id_client = CLIENT.id_client
GROUP BY CLIENT.pass;
```

Résultats de la requête :



Barre groupées montrant le nombre de commande par pass

Les personnes ayant un pass Uber-Eats commandent plus de repas sur Uber-Eats que les personnes qui n'en possèdent pas. En effet la réduction

des coûts de livraison intéresse fortement notre clientèle tout en avançant l'entreprise.

Pour promouvoir cet abonnement nous allons proposer une période d'essai gratuite plus longue pour que le client puisse constater qu'il économise plus en faisant des commandes avec le pass.

On pourrait mettre en avant ce dernier lors du paiement pour que tous les clients prennent connaissance de cet abonnement. On fera aussi une simulation du prix final de la commande avec déduction des frais de livraison (application du pass) pour montrer la différence de prix et inciter la clientèle à se inscrire.

- Le nombre de commandes effectuées dans un restaurant par mois

```
#NOMBRE DE COMMANDE PAR Restaurant Par mois
SELECT COUNT(*),RESTAURANT.ID_RESTAURANT, DATEE.mois
FROM FACTURE_CLIENT, RESTAURANT, DATEE
WHERE FACTURE_CLIENT.id_restaurant= RESTAURANT.ID_RESTAURANT
AND FACTURE_CLIENT.id_DATE = DATEE.mois
GROUP BY RESTAURANT.ID_RESTAURANT, DATEE.mois;
```

Les résultats de la requête :

Mois	NbCmd	Restaurant		Nb Cmd	Restaurant
janvier	2	Leclerc		12	petit jardin
	6	Petit Jardin		5	le tram
	2	Puja		1	les viges
fevrier	8	Petit Jardin		1	le passé
	2	l'Escale		3	le puja
	2	Trame		4	pastis
mars	2	Diligence		2	baltazare
	2	Passé		3	hockeyeurs
	2	Angelus		3	le bistrok
Avril	2	Angelus		2	le lab
	2	Pastis		1	la diligence
	2	Trame		3	leclerc
Mai	4	Bistrok		3	l'escale
	2	Escale		2	l'angelus

Dans le tableau de gauche nous pouvons observer le nombre de commandes effectuées par restaurant et par mois.

Nous remarquons que pour chaque mois il existe un restaurant qui a effectué plus de ventes que les autres.

L'idée serait de le mettre en avant pendant tout le mois qui suit sur le site et l'application pour qu'il soit remarqué par la clientèle.

Par ailleurs, on sait que plus le restaurant réalise des ventes (à travers Uber-Eats), plus ça avantage Uber-Eats. On a donc pensé à récompenser les restaurateurs qui ont effectué le plus de commandes pendant l'année, on peut voir cela dans le tableau de droite, en leur dédiant un espace sur la plateforme visible tout au long de l'année suivante.

On espère donc un plus grand profit, l'an prochain, de ces restaurants

qui ont été favorisées.

3.2.2 Livreur

— Le nombre de livreurs par description du climat

```
#nombre de livreur par climat
SELECT COUNT(*), CLIMAT.description
FROM LIVREUR, CLIMAT
WHERE LIVREUR.ID_CLIMAT = CLIMAT.ID_CLIMAT
GROUP BY CLIMAT.description;
```

Résultats de la requête :

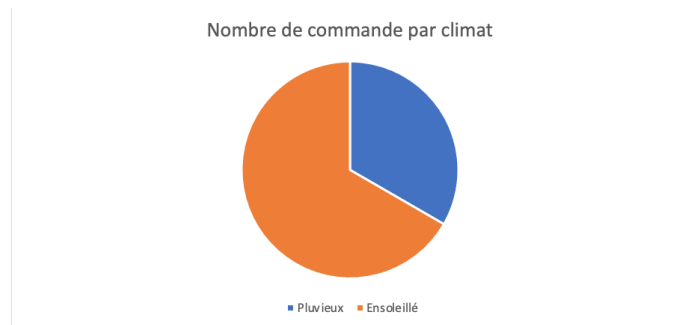


Diagramme montrant le nombre de commande par description du climat

On remarque qu'il y a plus de livreurs actifs sous le soleil que sous la pluie. Le but est d'étendre le temps de travail des livreurs pour tirer profit au maximum de chacun des employés (livreurs). On proposera alors d'augmenter leur commission dans les périodes pluvieuses pour susciter leur intérêt.

— Le nombre de livreurs par tranche d'âge

```
#nombre de livreur par categorie d'age
SELECT COUNT(*), CATEGORIE.categorie
FROM LIVREUR, CATEGORIE
WHERE LIVREUR.id_categorie = CATEGORIE.id_categorie
GROUP BY CATEGORIE.categorie;
```

Résultats de la requête :

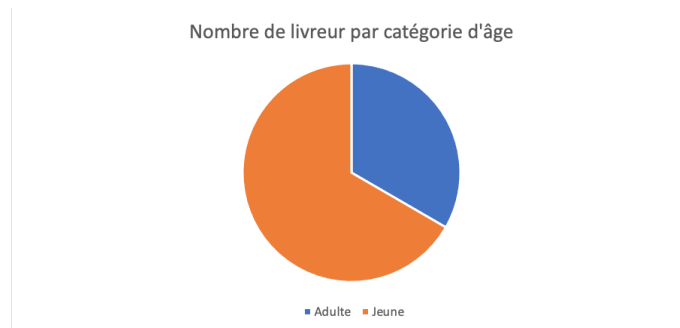


Diagramme montrant le nombre de livreurs par catégorie d'âge

On constate qu'il y a plus de jeunes livreurs qui travaillent sur la plateforme. On voudra alors agrandir ce nombre car plus il y a de livreurs plus les livraisons se feront rapidement. On va donc essayer de recruter plus de livreurs, on va cibler les endroits fréquentés par les jeunes pour essayer de les recruter. Par exemple on distribuera des flyers de l'entreprise dans des universités, des bars etc...

— Le nombre de livreurs par ville

```
#nombre de livreur par localistaion
SELECT COUNT(*), LOCALISATION.ville
FROM LIVREUR, LOCALISATION
WHERE LIVREUR.ID_Localisation = LOCALISATION.ID_Localisation
GROUP BY LOCALISATION.ville;
```

Résultats de la requête :

Ville	Nb Livreurs
Limoges	4
Aurillac	1
Montpellier	5
Parsi	7
Saint-Malo	3
Rouen	12
Beziers	3
Agde	1

Tableau modélisant le nombre de livreurs par ville

D'après le tableau ci-dessus on remarque que certaines villes comportent plus de livreurs que d'autres, on aimerait bien étendre la zone géogra-

phique d' Uber-Eats et donc augmenter le nombre de livreur dans les villes ou celui ci est bas.

Là encore, à plus grande échelle, on essaiera de promouvoir la plateforme sur les villes qui contiennent le moins de livreurs afin d'inciter les personnes à travailler chez Uber-Eats.

3.3 Vues matérielles

On utilisera des vues matérielles pour nos requêtes, les données sont pré-calculées et l'interrogation d'une vue matérialisée est plus rapide que l'exécution d'une requête sur la table de base de la vue.

Les vues matérialisées sont conçues pour améliorer les performances d'interrogation de charges de travail composées de modèles d'interrogation communs et répétés.

Leur but est d'accélérer les opérations d'agrégation, de projection et de sélection coûteuses.

On retrouvera les vues matérielles pour nos requêtes dans l'archive du code.

Chapitre 4

Conclusion

Le Data-Mart nous a permis d'organiser et de recueillir des informations simples et rapides sur les commandes, les restaurants, les clients et livreurs. L'analyse de ces informations nous a aidé à prendre des décisions commerciales efficaces et éclairées, des décisions grâce auxquelles nous avons pu améliorer les relations avec les restaurants, les clients et les livreurs, mais surtout nous avons pu améliorer le chiffre d'affaire.