



# ENTREPÔTS DE DONNÉES ET BIG-DATA

HMIN 122M

---

## Entrepôts de Données appliquées à BlaBlaCar

---

*Élèves :*

Adel TERKI  
Yasmine KHODJA  
Emile YOUSSEF  
Ines BENGHEZAL

*Enseignant :*

Federico ULLIANA  
Anne-muriel CHIFOLLEAU

8 novembre 2018



## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Analyse des besoins</b>	<b>3</b>
<b>3</b>	<b>Liste des actions</b>	<b>4</b>
<b>4</b>	<b>Traitements possibles pour chaque fait</b>	<b>4</b>
4.1	Réservation d'un trajet . . . . .	4
4.2	Proposition et recherche d'un trajet . . . . .	4
4.3	Assurer un véhicule . . . . .	4
<b>5</b>	<b>Deux actions/opérations les plus importantes à analyser</b>	<b>4</b>
<b>6</b>	<b>Modélisation</b>	<b>5</b>
6.1	Réservation d'un trajet . . . . .	5
6.2	Proposition et recherche d'un trajet . . . . .	6
6.3	Liste des mesures à enregistrer dans l'entrepôt . . . . .	6
<b>7</b>	<b>Possibilités de répondre aux traitements ?</b>	<b>6</b>
<b>8</b>	<b>Instance de l'entrepôt de données</b>	<b>7</b>
<b>9</b>	<b>Estimation de la taille des tables de l'entrepôt sur 12 mois</b>	<b>8</b>
<b>10</b>	<b>Requêtes analytiques</b>	<b>9</b>
<b>11</b>	<b>Exemples d'exécution des requêtes</b>	<b>11</b>
<b>12</b>	<b>Ensemble de vues matérialisées permettant de répondre aux requêtes</b>	<b>13</b>
<b>13</b>	<b>Conclusion</b>	<b>16</b>



## Table des figures

1	Data-mart du fait <b>Réservation</b> . . . . .	5
2	Data-mart des deux faits avec des dimensions communes <b>Proposition</b> et <b>Recherche</b> d'un trajet . . . . .	6
3	Capture de la table après l'exécution de la requête <b>1</b> . . . . .	11
4	Capture de la table après l'exécution de la requête <b>2</b> . . . . .	11
5	Capture de la table après l'exécution de la requête <b>3</b> . . . . .	12
6	Capture de la table après l'exécution de la requête <b>4</b> . . . . .	12
7	Capture de la table après l'exécution de la requête <b>5</b> . . . . .	12
8	Capture de la table après l'exécution de la requête <b>6</b> . . . . .	12
9	Capture de la table après l'exécution de la requête <b>7</b> . . . . .	12
10	Capture de la table après l'exécution de la requête <b>8</b> . . . . .	13
11	Capture de la table après l'exécution de la requête <b>9</b> . . . . .	13
12	Capture de la table après l'exécution de la requête <b>10</b> . . . . .	13
13	Capture de l'exécution de la requête <b>3</b> en utilisant la vue matérialisées <b>resa</b>	15



# 1 Introduction

De nos jours, le trafic routier devient de plus en plus dense. La quantité d'émission de gaz dégagée par les voitures entraîne une pollution sévère de l'environnement. Une des solutions proposées étant le covoiturage, BlaBlaCar offre ce service à travers une plateforme en ligne, donnant la possibilité à des conducteurs de proposer des trajets qu'ils effectuent fréquemment aux utilisateurs qui désirent se rendre à la même destination.

Cependant, le site rencontre certaines problématiques concernant l'offre et la demande des utilisateurs auxquels il faut remédier car ces derniers sont parfois contraints de prendre un autre service de transport avant de pouvoir atteindre leur destination ou avant d'utiliser les services de BlaBlacar.

Comment inciter les conducteurs à proposer plus de trajet et les orienter vers des trajets plus demandés que proposés ? Comment diminuer la densité de circulation sur les routes les plus encombrées ? Comment attirer plus de personne à assurer leur véhicule à travers la plateforme ? Comment évaluer les coûts de partenariat avec d'autres services de transport ?

# 2 Analyse des besoins

Pour répondre aux questions précédemment posées, il nous faut récolter des données nous permettant d'effectuer des traitements sur ces dernières afin d'arriver à une solution.

1. Connaître le nombre de trajets selon la ville de destination et la ville de départ, pour éventuellement créer une extension avec des voitures propres à la société qui effectueront dans un premier temps les trajets les plus utilisés à partir de chaque ville.
2. Connaître les conducteurs les plus fiables afin de les récompenser pour les inciter à proposer plus de trajets via BlaBlaCar.
3. Comparer les trajets les plus recherchés par les utilisateurs avec ceux proposés pour éventuellement suggérer d'autres itinéraires peu proposés mais souvent demandés.
4. Connaître les trajets les plus populaires ayant le prix le plus bas, afin de les afficher en première page.
5. Effectuer une analyse sur la rentabilité de la vente des tickets de bus (Ouibus).
6. Connaître le nombre d'utilisateurs qui assurent leur véhicule sur la plateforme selon la ville.
7. Connaître le nombre de cartes carburant et cartes lavage prises par les conducteurs ayant effectué un premier trajet afin d'adapter l'offre selon leurs préférences.

### 3 Liste des actions

Après avoir analysé chaque besoin, la liste des actions extraite est la suivante :

- Réservation d'un trajet : [Besoins 1-4-5]
- Proposition, Recherche : [Besoins 3-7 ]
- Assurer un véhicule : [Besoin 6]

### 4 Traitements possibles pour chaque fait

#### 4.1 Réservation d'un trajet

1. Afficher le nombre de trajets réservés en effectuant un group by sur la ville de départ et la ville de destination.
2. Afficher le nombre de trajets réservés ayant le prix minimum en effectuant un group by sur la ville de départ et la ville de destination.
3. Effectuer un snapshot sur la vente des tickets de bus par jour.

#### 4.2 Proposition et recherche d'un trajet

1. Afficher le nombre de trajets proposés en effectuant un group by sur la ville de départ et la ville de destination, et le comparer au nombre de trajets rechercher selon la même ville de départ et de destination.
2. Afficher les trajets recherchés en effectuant un group by sur la ville de départ et la ville de destination qui n'ont jamais été proposés (Trajets recherchés - trajets proposés).
3. Afficher le nombre de cartes carburant et le nombre de cartes lavage prises par les utilisateurs ayant proposés un premier trajet et comparer les deux nombres.

#### 4.3 Assurer un véhicule

1. Effectuer un snapshot sur le nombre de voiture assurées par mois.
2. Afficher le nombre d'assurance en effectuant un group by sur le type d'assurance.
3. Effectuer un snapshot annuel sur la durée moyenne des contrats signés par les utilisateurs.

### 5 Deux actions/opérations les plus importantes à analyser

1. Réservation d'un trajet :  
Pourquoi ? Parce que les informations relatives aux réservations de trajet nous permettront de connaître les préférences des utilisateurs, et les prises de décision de l'entreprise dépendent de leur demande.
2. Proposition/recherche d'un trajet :  
Pourquoi ? Parce qu'une étude temporelle comparative aux propositions et aux recherches nous permet de voir l'état de l'offre et la demande des utilisateurs du site.

## 6 Modélisation

Pour répondre aux questions 5 et 6, on propose la représentation ci-dessous :

### 6.1 Réservation d'un trajet

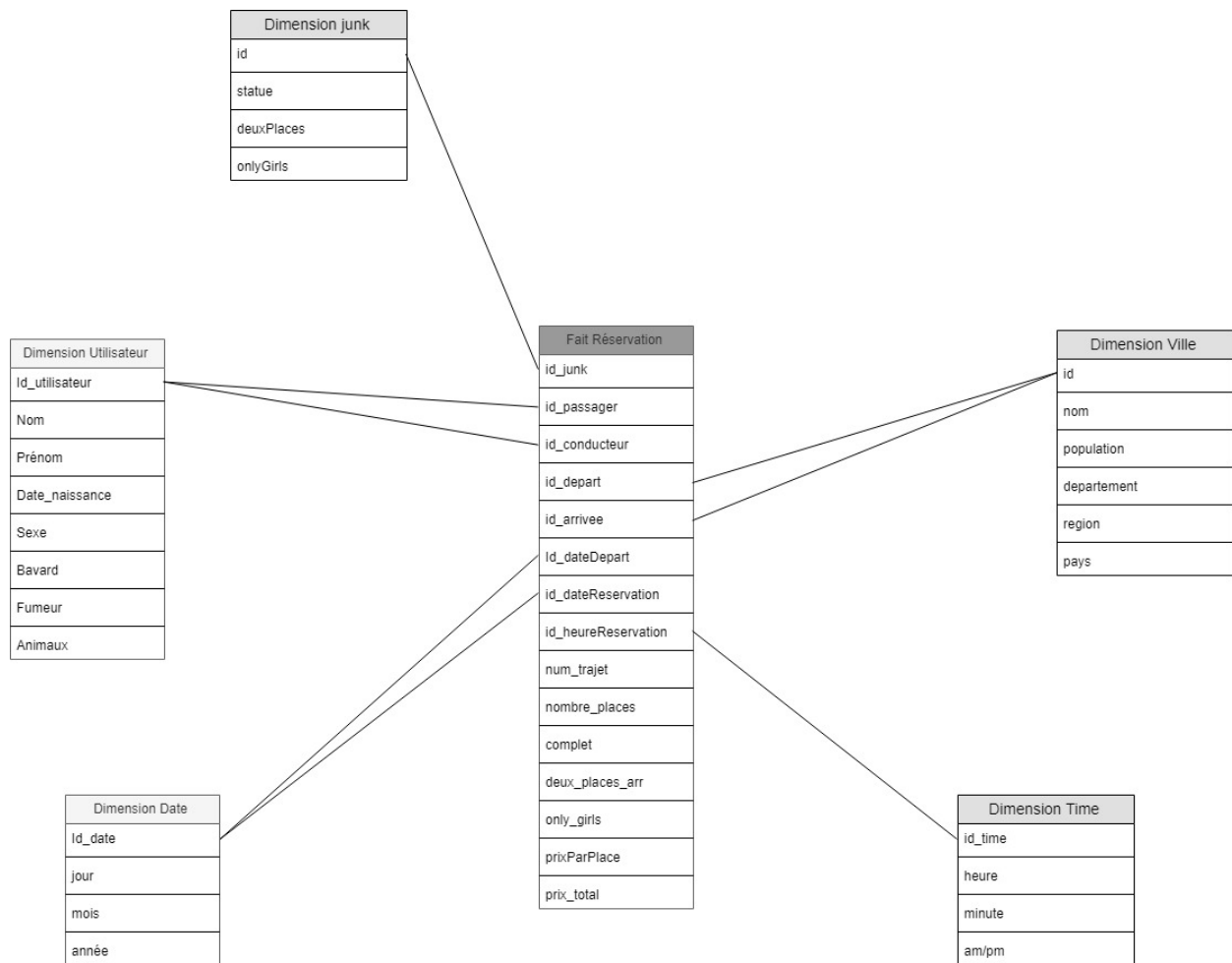


FIGURE 1 – Data-mart du fait **Réservation**

## 6.2 Proposition et recherche d'un trajet

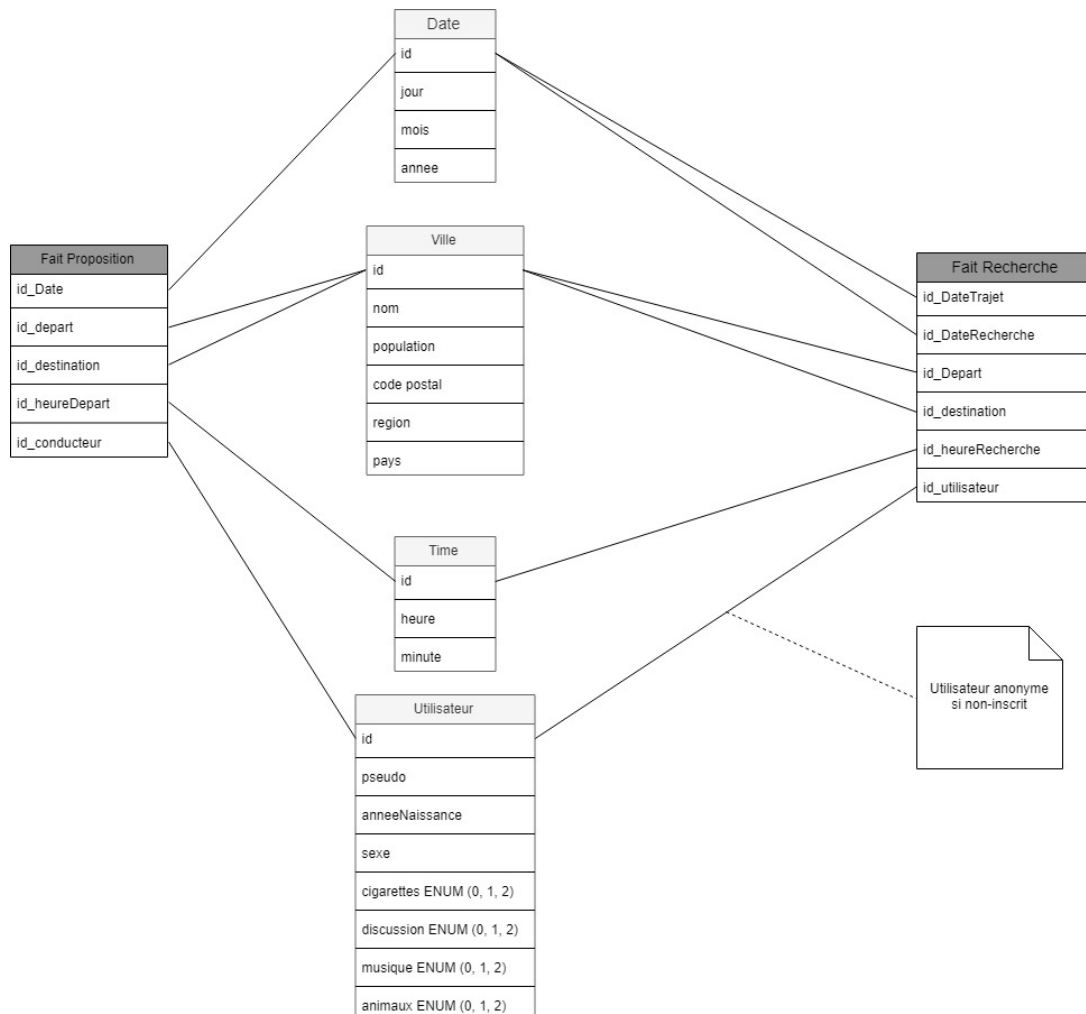


FIGURE 2 – Data-mart des deux faits avec des dimensions communes **Proposition** et **Recherche** d'un trajet

## 6.3 Liste des mesures à enregistrer dans l'entrepôt

- Réservation d'un trajet (additive)
- Proposition d'un trajet (non-additive)
- Recherche d'un trajet (non-additive)

## 7 Possibilités de répondre aux traitements ?

Cette modélisation nous permet de répondre aux traitements précédemment cités car toutes les informations nécessaires à la résolution de nos problèmes sont contenues dans les dimensions liées par les tables de faits.

— Réservation :

- Ce fait nous permet de visualiser les trajets effectués par ses utilisateurs et nous donne un aperçu global de leurs préférences.

- Une projection du plus grand nombre des réservations en fonction de la ville de départ et celle d'arrivée nous permettra de connaître les trajets les plus empruntés. En plus de cela, une recherche sur le prix le plus bas constaté dans les jours précédents permettra d'attirer plus de clientèles.
- Une requête filtrée par le nom d'utilisateur 'Ouibus' nous permet de connaître le nombre d'achat de tickets de bus par jour un jour ou un temps donné. En effet, les tickets proposés par Ouibus sur le site Blablacar sont effectués à travers un utilisateur du site au nom de la société.

— Recherche :

- Calculer le pourcentage de recherches et le nombre de propositions selon la ville de départ et la ville d'arrivée nous dévoile en quelques sortes les lieux potentiels où la société pourrait se faire plus de profits. Comme par exemple, en y implantant des voitures autonomes qui permettrait de faire des trajets entre 2 villes où les propositions sont très faibles comparées à la demande.

## 8 Instance de l'entrepôt de données

- Réservation :

id	id	id	id	id	id	id	num	id	nombre	prix	prix
date-Re-ser-va-tion	heu-reRe-ser-va-tion	dé-part	ar-ri-vée	da-te-Dep	pas-sa-ger	conduc-teur	Tra-jet	junk	place	par place	total
1	5	2	3	5	1	2	1	1	2	12	24
1	8	1	2	2	5	3	2	2	1	8	8
4	3	2	3	5	6	2	1	3	1	12	12

- Utilisateur :

id	nom	prénom	date naissance	sexe	bavard	fumeur	animaux
1	Jean	Michel	22/08/91	M	1	0	0
2	Ritchell	Kate	04/11/68	F	2	1	1
3	Dupont	Kean	06/06/87	M	0	2	0
5	Zelhof	Joe	12/01/88	M	2	0	1
6	Laforge	Marie	25/01/83	F	2	1	0

- Ville :

id	nom	population	departement	region
1	Montpellier	275 318	34	Occitanie
2	Marseille	861 635	13	Provence-Alpes-Côte d'Azur
5	Paris	2 206 488	75	Ile-de-France
8	Bordeaux	249 712	33	Nouvelle-Aquitaine
3	Lille	232 741	59	Hauts-de-France





- Junk\_dim :

id	statut	deuxPlaces	onlyGirls
1	reserve	0	0
2	reserve	0	1
3	reserve	1	0
4	reserve	1	1
5	confirme	0	1
6	confirme	0	0
7	confirme	1	0
8	confirme	1	1
9	annule	0	1
10	annule	0	0
11	annule	1	0
12	annule	1	1

- Date :

id	fullDate	day	month	monthNum	year
1	01/01/2018	monday	january	01	2018
4	05/09/2018	wednesday	september	09	2018

- Time :

id	timeFull	timeFull24	heure	heure24	minute	am_pm
5	03 :20 :00 pm	15 :20 :00	03	15	20	pm
8	04 :12 :00 pm	16 :12 :00	04	16	12	pm
3	08 :00 :00 am	08 :00 :00	08	08	00	am

## 9 Estimation de la taille des tables de l'entrepôt sur 12 mois

D'après les études de BlaBlaCar<sup>1</sup>, le nombre d'utilisateurs voyageant via la plateforme s'élève à plus de 20 millions de membres. Ce qui nous donne une estimation de la taille des tables de l'entrepôt de données suivante :

- **Utilisateur\_dim** : 20 Millions de lignes.
- **Time\_dim** : 1440 lignes (nombre de minutes par an).
- **Date\_dim** : 365 lignes (365 jours).
- **Ville\_dim** : 100 lignes (en prenant en compte que des villes françaises).
- **Junk\_dim** : 12 lignes.
- **Recherche** : 100 Millions de lignes (nombre de recherches faites par an approximativement).
- **Proposition** : 15 Millions de lignes.
- **Reservation** : 25 Millions de lignes (2 Millions d'utilisateurs voyagent chaque mois via la plateforme).

---

1. Article BlaBlaCar, <https://blog.blablacar.fr/blablalife/nouveautes/blablacar-dans-le-monde/10-millions-membres-blablacar>

## 10 Requêtes analytiques

1. Afficher le nombre de trajets réservés en effectuant un group by sur la ville de départ et la ville de destination :

```
select id_villeDep , id_villeArr , count(num_trajet)
from reservation
group by id_villeDep , id_villeArr ;
```

2. Afficher le nombre de trajets réservés ayant le prix minimum en effectuant un group by sur la ville de départ et la ville de destination.

```
select id_villeDep , id_villeArr , MIN(prixPlace)
from reservation
group by id_villeDep , id_villeArr ;
```

3. Nombre de places vendues et prix total des réservations avec Ouibus par jour.

```
select d.fullDate , sum(nombrePlace) as placeVendue ,
sum(prixTotal) as prixTotal
from reservation r , date_dim d
where r.id_conducteur = 17
      and r.id_dateResa = d.id
group by d.fullDate ;
```

4. Afficher le nombre de trajets proposés, le nombre de trajets recherchés et le rapport entre les deux en pourcentages à la même date selon la ville de départ et la ville de destination.

```
select p.id_dateDep , p.id_villeDep , p.id_villeArr ,
(select count(*) from proposition p2
where p.id_dateDep = p2.id_dateDep
      and p.id_villeDep = p2.id_villeDep
      and p.id_villeArr = p2.id_villeArr ) as nbProp ,
(select count(*) from recherche r2
where p.id_dateDep = r2.id_dateDep
      and p.id_villeDep = r2.id_villeDep
      and p.id_villeArr = r2.id_villeArr ) as nbRech ,
(select ((nbProp / nbRech) * 100)) as satisfaction
from proposition p , recherche r
where p.id_dateDep = r.id_dateDep
      and p.id_villeDep = r.id_villeDep
      and p.id_villeArr = r.id_villeArr
group by p.id_dateDep , p.id_villeDep , p.id_villeArr ;
```

5. Afficher les trajets recherchés en effectuant un group by sur la ville de départ et la ville de destination qui n'ont jamais été proposés (Trajets recherchés - trajets proposés).

```
select id_villeDep , id_villeArr
from recherche
MINUS (select id_villeDep , id_villeArr from proposition);
```

6. Afficher le rapport en pourcentages entre le nombre d'utilisateurs ayant pris une carte cadeau après leur premier trajet et le nombre d'utilisateurs ayant proposé un trajet au minimum et comparer les deux nombres.

```
select *
from(
select
  ((select count(DISTINCT id_conducteur) AS nbr_lavage
    from utilisateur_dim , proposition
    where id_conducteur = utilisateur_dim.id
        and utilisateur_dim.cadeau = 'LAVAGE')
  /(select count(distinct id_conducteur)
    from proposition) * 100) as pourcentage_lavage)
as LAVAGE,
(select ((select count(distinct id_conducteur)
  as nbr_carburant
  from utilisateur_dim , proposition
  where id_conducteur = utilisateur_dim.id
  and utilisateur_dim.cadeau = 'CARBURANT')
  /(select count(distinct id_conducteur)
  from proposition)
  * 100) as pourcentage_carburant) as CARBURANT,
(select (
  (select count(distinct id_conducteur) AS nbr_aucun
  from utilisateur_dim , proposition
  where id_conducteur = utilisateur_dim.id
  and utilisateur_dim.cadeau IS NULL )
  /(select count(distinct id_conducteur)
  from proposition)
  * 100) as pourcentage_aucun) as AUCUN;
```

7. Afficher le nombre de trajets réservés annulés selon la ville de départ et ville d'arrivée.

```
select id_villeDep , id_villeArr , count(num_trajet)
from reservation ,junk_dim
where junk_dim.id = reservation.id_junk
    and junk_dim.statue = 'annule'
group by id_villeDep , id_villeArr;
```

8. Les utilisateurs réserve-t-il souvent des trajets fumeur?

```
select id_villeDep , id_villeArr , count(num_trajet)
from reservation , utilisateur_dim
where utilisateur_dim.niv_fumeur = '1'
    or utilisateur_dim.niv_fumeur = '2'
group by id_villeDep , id_villeArr;
```

9. Estimer le nombre de réservations faites par jour.

```
select date_dim.fullDate , reservation.id_villeDep ,
    reservation.id_villeArr ,count(num_trajet)
```

```

from reservation , date_dim
where reservation.id_dateResa=date_dim.id
group by date_dim.fullDate , id_villeDep , id_villeArr ;

```

10. Comparer le nombre de trajets fait que par des filles et les autres.

```

select * from
(
select
((select count(*) from reservation , junk_dim
where reservation.id_junk=junk_dim.id
and junk_dim.onlyGirls=1)
/(select count(*) from reservation)*100)
as pourcentage_onlyGirls) as filles ,
(
select
((select count(*) from reservation , junk_dim
where reservation.id_junk=junk_dim.id
and junk_dim.onlyGirls<>1)
/(select count(*) from reservation)*100)
as pourcentage_notOnlyGirls) as not_only_girls

```

## 11 Exemples d'exécution des requêtes

- Requête 1 :

	id_villeDepart	id_villeArr	nbr_trajets
1	10	28	2
2	12	13	1
3	19	25	3
4	45	78	2
5	54	65	2
6	65	45	1

FIGURE 3 – Capture de la table après l'exécution de la requête 1

- Requête 2 :

	id_villeDepart	id_villeArr	min_prixPlace
1	10	28	12
2	12	13	13
3	19	25	17
4	45	78	15
5	54	65	45
6	65	45	10

FIGURE 4 – Capture de la table après l'exécution de la requête 2

- Requête 3 :

	date	placesVendues	prixTotal
1	2016-01-24	1	15
2	2016-01-25	2	30

FIGURE 5 – Capture de la table après l'exécution de la requête 3

- Requête 4 :

	date_depart	ville_depart	ville_arrivee	nbProposition	nbRecherche	satisfaction
1	18	10	28	1	2	50
2	26	65	45	1	1	100

FIGURE 6 – Capture de la table après l'exécution de la requête 4

- Requête 5 :

	id_villeDepart	id_villeArrivee
1	78	96
2	73	15
3	45	97
4	82	85
5	1	10
6	36	54
7	24	45

FIGURE 7 – Capture de la table après l'exécution de la requête 5

- Requête 6 :

	pourcentage_lavage	pourcentage_carburant	pourcentage_aucun
1	41.6667	33.3333	25

FIGURE 8 – Capture de la table après l'exécution de la requête 6

- Requête 7 :

	id_villeDep	id_villeArr	nombre_trajets
1	54	65	2

FIGURE 9 – Capture de la table après l'exécution de la requête 7

- Requête 8 :

	id_villeDepart	id_villeArrivee	nombre_trajets
1	10	28	18
2	12	13	9
3	19	25	27
4	45	78	18
5	54	65	18
6	65	45	9

FIGURE 10 – Capture de la table après l'exécution de la requête 8

- Requête 9 :

	date	id_villeDepart	id_villeArrivee	nombre_trajets
1	2016-01-17	10	28	2
2	2016-01-24	45	78	1
3	2016-01-25	45	78	1
4	2016-01-25	65	45	1
5	2016-01-30	54	65	2
6	2016-02-14	19	25	2
7	2016-02-15	12	13	1
8	2016-02-15	19	25	1

FIGURE 11 – Capture de la table après l'exécution de la requête 9

- Requête 10 :

	pourcentage_onlyGirls	pourcentage_notOnlyGirls
1	45.4545	54.5455

FIGURE 12 – Capture de la table après l'exécution de la requête 10

## 12 Ensemble de vues matérialisées permettant de répondre aux requêtes

- Requête 1/2/5/7/8 :

```
create materialized view
mv1 (ville_dep, ville_arr, nombre_trajet, prix_min)
as select id_villeDep,
          id_villeArr,
          count(num_trajet),
          min(prix_place)
from reservation
group by id_villeDep, id_villeArr;
```



- Requête 9

```
create materialized view
mv2 (date, ville_dep, ville_arr, nombre_trajet)
as select date_dim.fullDate,
          reservation.id_villeDep,
          reservation.id_villeArr,
          sum(nombrePlace),
          count(num_trajet)
from reservation
group by date_dim.fullDate, id_villeDep, id_villeArr;
```

- Requête 3

```
create materialized view
resa (dateR, conducteur, placeTotal, prixTotal)
as select id_dateResa,
          id_conducteur,
          sum(nombrePlace),
          sum(prixTotal)
from reservation
group by id_dateResa, id_conducteur;
```

- Requête 4/6

```
create materialized view
mv3 (dateDep, ville_dep, ville_arr, nombre_conducteurs)
as select proposition.id_dateDep,
          proposition.id_villeDep,
          proposition.id_villeArr,
          sum(id_conducteur)
from proposition
group by id_dateDep, id_villeDep, id_villeArr;

select d.fullDate as Day,
       r.conducteur,
       sum(placeTotal) as PlaceTotal,
       sum(prixTotal) as PrixTotal
from resa r, date_dim d
where r.conducteur = 17
and r.dateR = id
group by d.fullDate, r.conducteur;
```

[illegible]

FIGURE 13 – Capture de l'exécution de la requête **3** en utilisant la vue matérialisées **resa**





## 13 Conclusion

Dans le cadre de ce travail, nous avons cherché à proposer un entrepôt de données aidant la plateforme BlaBlaCar à traiter tous les besoins choisis comme étant des besoins primaires. Les traitements effectués nous ont permis d'avoir des résultats en des tables regroupant des analyses statistiques appliquées à la base de connaissances du site.