UNIVERSITE DE BOURGOGNE
UFR SCIENCES ET TECHNIQUES
MASTER2 BDIA

RAPPORT TD/TP

INFORMATIQUE DECISIONNELLE

Étudiant :
Moussa TRAORE

Enseignants :
Eric LECLERCQ
Annabelle GILLET

Table des matières

I Introduction

1 Contexte du projet

Ce rapport vise à étudier divers aspects des données de l'application Yelp, dans le but de répondre à des questions pertinentes pour le domaine de l'informatique décisionnelle ou de la Business Intelligence (BI). Yelp est une plateforme en ligne où les utilisateurs peuvent noter et commenter des commerces tels que des restaurants, des cafés, des boutiques, etc. Les données recueillies sur cette plateforme offrent une mine d'informations précieuses sur les préférences des consommateurs et les tendances du marché dans divers secteurs d'activité.

2 Objectifs de l'analyse et de la mise en œuvre du data warehouse

En exploitant les données Yelp, nous visons à identifier les secteurs d'activité les plus populaires, à analyser la demande géographique, à évaluer la satisfaction client, à étudier la fidélité client, à détecter les tendances saisonnières ou cycliques, et à comparer la performance des entreprises concurrentes. L'objectif ultime est de fournir des insights précieux aux équipes de marketing et de vente pour les aider à prendre des décisions stratégiques et à améliorer leurs performances commerciales.

II Analyse du sujet

1 Description détaillée des données disponibles

Dans cette sous-section, nous explorons en détail les données brutes fournies par Yelp ainsi que les adaptations qui leur ont été apportées pour répondre aux besoins du projet.

Les données brutes de Yelp se présentent principalement sous forme de fichiers JSON, offrant une mine d'informations sur les entreprises, les avis des utilisateurs, les profils d'utilisateurs, les check-ins et les photos. Voici un aperçu détaillé des principaux fichiers de données :

- **business.json** : Ce fichier contient des données sur les entreprises répertoriées sur Yelp, y compris des informations telles que l'identifiant unique de l'entreprise, son nom, son adresse complète, sa ville, son État, son code postal, sa latitude et sa longitude géographique, sa notation en étoiles, le nombre d'avis, son statut d'ouverture, ses attributs (comme les options de stationnement), ses catégories et ses heures d'ouverture.
- **review.json** : Ce fichier regroupe les avis complets et détaillés rédigés par les utilisateurs de Yelp sur les entreprises. Chaque avis comprend une notation en étoiles, un texte descriptif détaillé sur l'expérience de l'utilisateur, ainsi que des informations sur l'utilité, la drôlerie et la pertinence de l'avis selon les autres utilisateurs.
- **user.json** : Cette ressource fournit des informations exhaustives sur les utilisateurs de Yelp, notamment leur nom, le nombre de critiques qu'ils ont rédigées, la date

à laquelle ils ont rejoint la plateforme Yelp, ainsi que des détails sur leurs amis et leurs activités sur le site.
— **checkin.json** : Enregistrant les check-ins des utilisateurs dans les entreprises, ce fichier offre un aperçu des tendances de fréquentation et de l'engagement des utilisateurs pour différentes entreprises répertoriées sur Yelp. Il fournit des données temporelles sur quand et avec quelle fréquence les utilisateurs visitent ces lieux.
— **tip.json** : Contenant des conseils rapides et succincts laissés par les utilisateurs sur les entreprises, ce fichier offre une perspective concise mais utile sur les points forts ou les particularités des différents établissements répertoriés sur Yelp.
— **photo.json** : Regroupant les photos téléchargées par les utilisateurs sur Yelp, ce fichier offre un aperçu visuel des entreprises répertoriées, avec des légendes et des classifications permettant de distinguer les types de photos, comme la nourriture, les boissons, les menus, les intérieurs ou les extérieurs.

De plus, afin de tirer pleinement parti de ces données dans le cadre du projet, certaines transformations ont été apportées pour les adapter à différents formats de stockage et faciliter leur manipulation. Celles-ci incluent la conversion du fichier `tip.json` en format CSV pour simplifier l'analyse des données, le stockage des informations sur les utilisateurs `users.json` et les critiques `reviews.json` dans une base de données PostgreSQL avec une organisation en plusieurs tables,rajout des tables `friend,elite` pour faciliter la gestion et l'exploitation . Les autres fichiers JSON sont restés intacts dans leur format d'origine pour assurer une continuité avec les données brutes fournies par Yelp.

2 Définition du cadre des analyses

Dans le cadre de ce projet d'analyse des données Yelp, le cadre des analyses est défini par les objectifs spécifiques suivants :
— Identifier les secteurs d'activité les plus populaires dans un pays donné afin d'améliorer le ciblage des efforts de marketing et de vente.
— Localiser les régions présentant une forte demande pour certains types d'entreprises, facilitant ainsi l'élaboration de stratégies de développement commercial efficaces.
— Analyser la satisfaction client à l'égard des entreprises répertoriées sur Yelp en scrutant les avis et les commentaires des utilisateurs.
— Évaluer la fidélité client en examinant la fréquence d'utilisation de la plateforme par les utilisateurs enregistrés.
— Étudier les tendances saisonnières ou cycliques dans les activités des utilisateurs et des entreprises pour optimiser les stratégies commerciales.
— Comparer la performance des entreprises concurrentes dans la même catégorie afin d'identifier les domaines d'amélioration et de compétitivité.

Ces objectifs serviront de base pour structurer notre analyse des données Yelp et guideront la mise en place du data warehouse pour répondre aux besoins spécifiques des équipes de marketing et de vente.

3 Objectifs de l'étude

Les objectifs de cette étude sont en corrélation directe avec les objectifs de l'analyse des données Yelp et sont définis comme suit :

- Identifier les tendances et les préférences des consommateurs dans différents secteurs d'activité.
- Analyser la géographie des interactions des utilisateurs avec les entreprises pour identifier les zones à fort potentiel commercial.
- Évaluer la satisfaction client à travers l'analyse des avis et des commentaires des utilisateurs.
- Mesurer la fidélité client en examinant les comportements d'utilisation récurrents de la plateforme Yelp.
- Détecter les variations saisonnières ou cycliques dans les activités des utilisateurs et des entreprises pour ajuster les stratégies commerciales en conséquence.
- Comparer la performance des entreprises concurrentes pour fournir des recommandations visant à améliorer leur compétitivité sur le marché.

Ces objectifs déterminent le périmètre de l'étude et guideront l'analyse approfondie des données Yelp ainsi que la mise en place des solutions décisionnelles pour répondre aux besoins des parties prenantes.

III Conception du data warehouse et des data marts

L'élaboration d'un data warehouse et de data marts constitue une étape cruciale dans la mise en place d'une infrastructure robuste pour l'analyse des données Yelp. Cette section se penche sur la spécification du schéma du data warehouse, déterminant la structure fondamentale pour l'intégration des données, ainsi que des data marts, permettant une organisation spécifique et efficace des données pour des analyses ciblées.

0.1 Datamart 1 : Données pour les analyses axées entreprises

- **Table LOCATION :**
  - Attributs : location_id (identifiant de localisation), business_id (identifiant de l'entreprise), location (nom de l'emplacement), city (ville), state (état), postal_code (code postal), latitude (latitude de l'emplacement), longitude (longitude de l'emplacement).
- **Table CATEGORIES :**
  - Attributs : category_id (identifiant de catégorie), category (catégorie de l'entreprise).
- **Table CHECKIN :**
  - Attributs : checkin_id (identifiant de check-in), date (date du check-in).
- **Table HOURS :**
  - Attributs : hours_id (identifiant d'horaires), Monday (horaires du lundi), Tuesday (horaires du mardi), Wednesday (horaires du mercredi), Thursday (horaires du jeudi), Friday (horaires du vendredi), Saturday (horaires du samedi), Sunday (horaires du dimanche).
- **Table REVIEWS :**

- Attributs : review_id (identifiant de l'avis), text (contenu de l'avis), date (date de l'avis).
- **Table TIP :**
  - Attributs : tip_id (identifiant du tip), text (contenu du tip), date (date du tip).
- **Table FACT_BUSINESS :**
  - Attributs : business_id (identifiant de l'entreprise), category_id (identifiant de catégorie), checkin_id (identifiant de check-in), hours_id (identifiant d'horaires), tip_compliments_count (nombre de compliments sur les tips), tip_id (identifiant du tip), business_name (nom de l'entreprise), location_id (identifiant de localisation), review_id (identifiant de l'avis), reviews_useful_count (nombre d'utilisateurs ayant trouvé l'avis utile), reviews_cool_count (nombre d'utilisateurs ayant trouvé l'avis cool), reviews_funny_count (nombre d'utilisateurs ayant trouvé l'avis drôle), reviews_stars_count (note moyenne de l'entreprise), total_users_reviews_count (nombre total d'avis par utilisateur), total_usersfans_count (nombre total de fans par utilisateur).

0.2 Datamart 2 : Données pour les analyses axées utilisateurs

- **Table REVIEWS :**
  - Attributs : review_id (identifiant de l'avis), text (contenu de l'avis), date (date de l'avis).
- **Table ELITE :**
  - Attributs : user_id (identifiant de l'utilisateur), year (année d'élite).

1 Description des data marts envisagés

Après avoir spécifié le schéma du data warehouse, il est essentiel d'envisager la mise en place de data marts, éléments clés dans l'organisation et l'exploitation des données pour des analyses plus ciblées et spécifiques. Dans cette section, nous examinerons en détail les différents data marts envisagés, mettant en lumière leur rôle dans la fourniture d'informations précieuses pour les différentes unités fonctionnelles de l'entreprise.

1.1 Analyse des Entreprises et Avis Yelp

Le Data Mart "Analyse des Entreprises et Avis Yelp" vise à fournir des insights approfondis sur les entreprises répertoriées sur la plateforme Yelp ainsi que sur les avis qui leur sont associés. Ce Data Mart exploite les données des tables `CATEGORIES`, `HOURS`, `LOCATION`, `TIP`, `CHECKIN`, `FACT_BUSINESS`, et `REVIEW` pour permettre une analyse complète et pertinente.

Description des Données

Ce Data Mart inclut une variété d'informations sur les entreprises et les avis Yelp :

- **Catégories d'entreprises :** Catégories dans lesquelles les entreprises sont classées.
- **Horaires d'ouverture des entreprises :** Horaires d'ouverture pour chaque jour de la semaine.
- **Informations sur l'emplacement :** Adresse, ville, état, code postal, coordonnées géographiques.

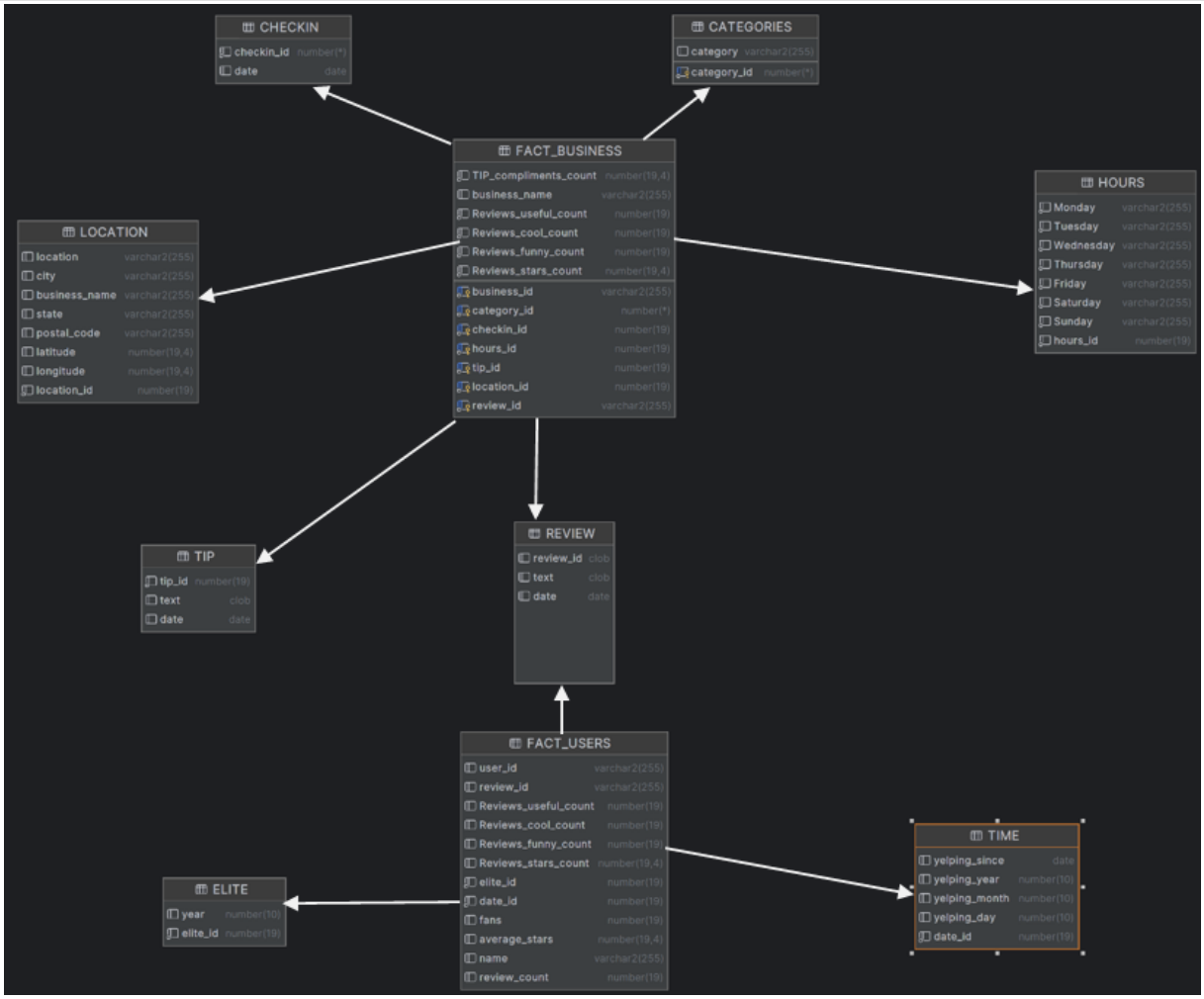

FIGURE III.1 – schema en étoile

— **Conseils (tips) :** Conseils laissés par les utilisateurs, avec la date.
— **Check-ins :** Informations sur les check-ins des utilisateurs dans les entreprises.
— **Données factuelles sur les entreprises :** Données agrégées sur les entreprises, y compris les évaluations, les critiques utiles, amusantes, et intéressantes.
— **Avis :** Informations sur les avis laissés par les utilisateurs, y compris les évaluations et les compteurs de réactions.

Objectifs

Les principaux objectifs de ce Data Mart sont les suivants :

— **Analyse de la performance des entreprises :** Évaluer la performance des entreprises en fonction des avis des utilisateurs, des horaires d'ouverture, des catégories, et de l'emplacement.
— **Identification des tendances du marché :** Détecter les tendances émergentes dans les catégories d'entreprises, les heures d'ouverture populaires, les conseils laissés par les utilisateurs, etc.
— **Personnalisation de l'expérience utilisateur :** Utiliser les données pour recommander des entreprises pertinentes aux utilisateurs en fonction de leurs préférences et de leur emplacement.

Utilisation Potentielle

Ce Data Mart peut être utilisé par Yelp pour :

— **Faciliter la prise de décision stratégique :** En fournissant aux décideurs des insights sur la performance des entreprises et les tendances du marché pour guider leurs décisions.

1.2 Analyse des Utilisateurs Yelp

Le data mart "Analyse des Utilisateurs Yelp" est conçu pour fournir des analyses approfondies sur les utilisateurs de la plateforme Yelp. Ce data mart exploite les données des tables `REVIEW`, `ELITE`, `TIME`, et `FACT_USERS` pour offrir des insights pertinents sur le comportement des utilisateurs et leurs interactions avec les évaluations et les commerces répertoriés sur Yelp.

Description des Données

Le data mart "Analyse des Utilisateurs Yelp" comprend les informations suivantes :

— **Données de base des utilisateurs :** Nom, identifiant, nombre de critiques effectuées, date depuis laquelle l'utilisateur est actif sur Yelp.

— **Activité des utilisateurs :** Nombre de critiques utiles, amusantes et intéressantes effectuées par l'utilisateur.

— **Élite :** Indicateur de statut "élite" pour l'utilisateur pour une année donnée.

— **Évaluations :** Identifiant de l'évaluation, note attribuée par l'utilisateur à un commerce.

— **Données temporelles :** Année, mois et jour depuis le début de l'activité de l'utilisateur sur Yelp.

Objectifs

Les objectifs principaux de ce data mart sont les suivants :

— **Analyse du comportement des utilisateurs :** Comprendre les préférences des utilisateurs en analysant les types de commerces évalués, les notes attribuées, et les activités d'interaction avec les critiques.

— **Identification des utilisateurs influents :** Identifier les utilisateurs ayant un impact significatif sur la plateforme Yelp, basé sur leur activité de critique, leur statut "élite", et leur engagement avec d'autres utilisateurs.

— **Analyse temporelle :** Étudier les tendances d'activité des utilisateurs au fil du temps, y compris les changements de comportement et les saisons d'activité accrue.

Utilisation Potentielle :

Ce data mart peut être utilisé pour :

— **Personnaliser l'expérience utilisateur :** En utilisant les informations sur les préférences des utilisateurs pour recommander des commerces et des activités pertinents.

— **Identifier les tendances émergentes :** Détecter les tendances émergentes dans les avis des utilisateurs et les comportements d'interaction, aidant ainsi Yelp à rester à l'avant-garde des préférences du marché.

IV Justification des choix

1 Explication des technologies choisies

1.1 Introduction

Un Data Warehouse est une base de données relationnelle hébergée sur un serveur dans un Data Center ou dans le Cloud. Il permet de stocker des données volumineuses provenant de sources variées et permet d'effectuer des requêtes rapides et complexes sur ces données-là. Il permet donc d'aider à l'analyse des données et à la prise de décision. D'un point de vue plus technique, le Data Warehouse contient un ensemble de données orientées sujet, intégrées, variables dans le temps et non volatiles.

1.2 Data Warehouse vs. Datamart

Un datamart représente un sous-ensemble du Data Warehouse. Tandis que le Data Warehouse couvre plusieurs sujets, un datamart est spécialisé sur un seul thème. Il est conçu pour accéder plus facilement à des données spécifiques. Les informations d'un datamart ciblent donc un métier précis ; il existe par exemple, des datamarts commerciaux constitués de données ciblées, organisées et regroupées.

1.3 Kimball vs. Inmon

Il existe plusieurs approches de modélisation d'un Data Warehouse mais deux approches sont les plus communes : l'approche « Kimball » qui est une approche ascendante et l'approche « Inmon » qui est une approche descendante. Ces deux approches considèrent le Data Warehouse en tant que référentiel central pour les données. De plus, les deux types d'approches utilisent les concepts ETL pour le chargement des données. Cependant, la principale différence réside dans la modélisation des données et leur chargement dans l'entrepôt de données. L'approche Kimball repose sur l'importance des datamarts. Dans cette approche, le Data Warehouse est simplement une combinaison de différents datamarts qui facilite le reporting et l'analyse. Cette conception de Kimball correspond à l'approche ascendante (« bottom-up »). L'approche Inmon représente un Data Warehouse comme un dépôt centralisé de toutes les données de l'entreprise. Dans cette approche, une organisation crée d'abord un modèle de Data Warehouse normalisé. Les datamarts des différents secteurs sont ensuite créés sur la base du modèle de l'entrepôt. C'est ce qu'on appelle une approche descendante (ou « top-down ») de l'entreposage des données.

2 Justification de notre choix de l'approche Kimball

Nous avons adopté l'approche de Kimball qui suit une approche ascendante pour le Data Warehouse dans laquelle les datamarts sont d'abord formés en fonction des besoins de l'utilisateur ; au lieu du modèle d'Inmon qui utilise une forme normalisée pour construire le Data Warehouse et ensuite créer les datamarts pour accéder aux données demandées.

En effet, dans ce TP, l'adoption des deux approches Kimball ou Inmon ne présente pas de différences assez importantes. Cependant, dans d'autres cas, il faut penser à bien choisir

quelle approche est la plus adéquate, car l'approche Inmon peut se présenter comme étant beaucoup trop complexe dans certaines situations.

Dans notre cas, le choix du modèle Kimball s'est justifié par plusieurs raisons essentielles :

- La modélisation dimensionnelle de Kimball est rapide à mettre en place car elle évite la normalisation, accélérant ainsi la phase initiale de conception de l'entrepôt de données.
- L'utilisation de schémas en étoile ou en flocon est plus accessible pour la plupart des utilisateurs, simplifiant ainsi les requêtes et l'analyse grâce à leur structure dénormalisée.
- Le Data Warehouse selon Kimball se concentre sur des domaines d'activité spécifiques, réduisant ainsi son empreinte dans la base de données et simplifiant la gestion du système.
- La séparation des données en tables de faits et dimensions permet une récupération rapide des données.
- La gestion de l'entrepôt de données selon Kimball nécessite une équipe de conception plus réduite, car les systèmes sources sont stables et orientés processus, facilitant également l'optimisation des requêtes.
- Comparativement à l'approche Inmon, la mise en œuvre et la maintenance selon Kimball sont moins coûteuses en termes de temps et d'investissement, les schémas normalisés étant plus complexes à concevoir et à maintenir.
- L'approche Inmon exige des compétences techniques plus avancées.
- Avec l'approche Inmon, davantage de processus ETL sont nécessaires pour la construction des datamarts, comparativement à l'approche Kimball.

3 Justification des décisions prises à chaque étape du processus

La modélisation dimensionnelle est un processus qui se déroule en plusieurs étapes, notamment le choix du processus métier et la détermination du grain. Dans les lignes qui suivent, nous allons expliquer le but de chaque étape, justifiant ainsi les choix que nous avons adoptés tout au long de ce travail pratique.

Choix du processus d'entreprise pour modéliser : En choisissant de modéliser les données à partir du processus d'entreprise, nous nous concentrons sur les activités commerciales essentielles qui sont au cœur de l'organisation. Cette approche est cruciale car elle permet de répondre directement aux besoins des utilisateurs finaux qui souhaitent comprendre et analyser les processus métier. Par exemple, les utilisateurs peuvent vouloir savoir quels types d'entreprises sont les plus populaires dans une région donnée ou quelles sont les heures d'ouverture les plus fréquentées. En commençant par les processus d'entreprise, nous nous assurons que notre entrepôt de données est aligné sur les besoins opérationnels et stratégiques de l'organisation.

Choix du grain : Le grain représente le niveau de détail auquel les données sont stockées dans le tableau de faits principal. Dans notre modèle, nous avons choisi un grain fin pour capturer les interactions spécifiques avec les entreprises, telles que les avis, les conseils, etc. Cette décision est motivée par la nécessité de fournir des données granulaires pour répondre aux questions commerciales détaillées. Par exemple, en capturant les avis individuels plutôt que des agrégats, nous permettons aux utilisateurs d'analyser les

tendances et les préférences des clients de manière plus précise.

Choix des dimensions : Les dimensions fournissent le contexte autour des mesures stockées dans le tableau de faits principal. Chaque dimension représente une caractéristique ou un aspect spécifique des interactions commerciales. Par exemple, la dimension LOCATION fournit des informations sur l'emplacement géographique des entreprises, tandis que la dimension ELITE capture des détails sur les années d'élite des utilisateurs qui laissent des avis ou des conseils. En choisissant les bonnes dimensions, nous permettons aux utilisateurs d'analyser les données sous différents angles et de tirer des insights significatifs pour prendre des décisions éclairées.

Identification des faits numériques : Les faits numériques sont les mesures quantifiables qui alimentent le tableau de faits principal. Dans notre modèle, nous avons identifié les mesures telles que le nombre de votes utiles, drôles et cool, ainsi que la notation en étoiles associée à chaque avis pour la table de fait **Business**. Ces mesures sont essentielles pour évaluer la performance des entreprises, la satisfaction client et d'autres aspects clés du processus métier. En identifiant ces faits numériques, nous assurons que notre entrepôt de données est capable de répondre aux questions commerciales les plus pertinentes et d'offrir des insights précieux pour améliorer les performances commerciales.

En résumé, en approfondissant chaque choix, nous nous assurons que notre modèle en étoile est construit de manière réfléchie pour répondre aux besoins spécifiques des utilisateurs finaux et pour fournir des analyses approfondies et exploitables pour soutenir la prise de décision stratégique.

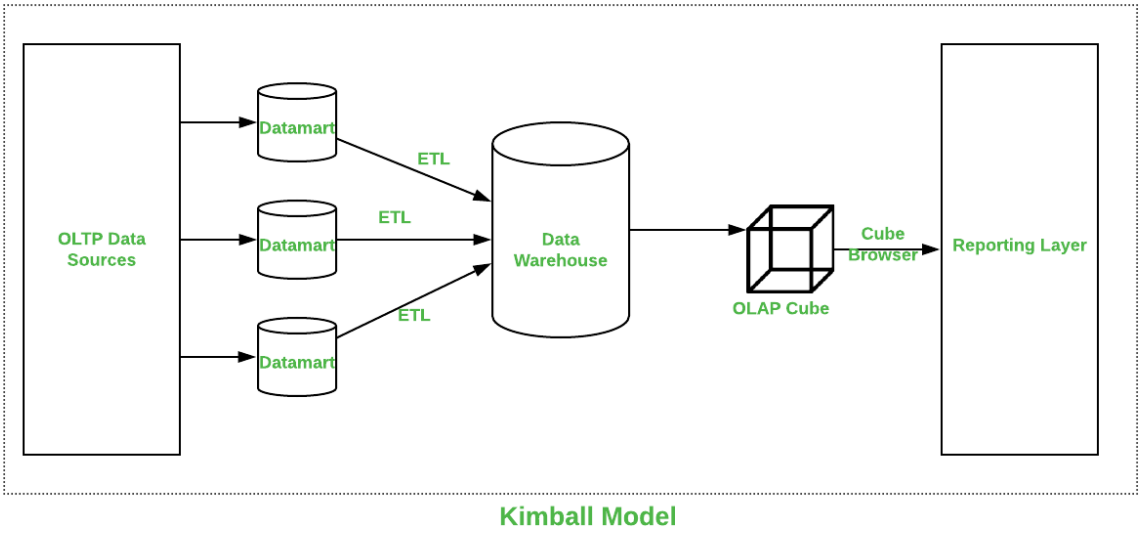

FIGURE IV.1 – https ://www.geeksforgeeks.org/difference-between-kimball-and-inmon/

V Requêtes et Visualisation

1 Description des différentes requêtes d'analyse

Nous avons crés environs 2 dizaines de requêtes d'analyse sur les 2 datamarts de notre base de données Oracle. Elles sont inspirées par les questions de départ que nous avons choisies :

- Quelles sont les entreprises les mieux notées ?
- Quels sont les types de catégories ?
- Quelles sont les entreprises avec le plus de critiques utiles par ville, pays, catégorie ?
- Quelles sont les entreprises les plus visitées ?
- Quelles sont les entreprises qui ont reçu le plus de conseils par ville, pays ?
- Quel jour de la semaine reçoit le plus de visites ?
- Quel utilisateur est le plus influent en termes de nombre de fans et d'années d'élite ?
- quel utilisateur à le plus grands nombre de reviews count ?
- quelle est la date de yelping la plus frequente ?

Requetes d'Analyse des Entreprises et Avis Yelp

```
--- Quels sont les entreprises les mieux notées ? ---
-- Les 100 entreprises avec le plus grand nombre de critiques --
SELECT DISTINCT "business_name", "Reviews_stars_count"
FROM FACT_BUSINESS
ORDER BY "Reviews_stars_count" DESC
FETCH FIRST 100 ROWS ONLY;

-- Les 100 entreprises les mieux notées par ville --
SELECT DISTINCT l."city" , b."business_name", b."Reviews_stars_count"
FROM FACT_BUSINESS b JOIN
"LOCATION" l ON b."location_id"=l."location_id"
GROUP BY l."city", b."business_name", b."Reviews_stars_count"
ORDER BY "Reviews_stars_count" DESC
FETCH FIRST 100 ROWS ONLY;

-- Les 100 entreprises les mieux notées par etat --
SELECT DISTINCT l."state",b."business_name", b."Reviews_stars_count"
FROM FACT_BUSINESS b JOIN
"LOCATION" l ON b."location_id"=l."location_id"
GROUP BY l."state", b."business_name", b."Reviews_stars_count",
b."business_id"
ORDER BY "Reviews_stars_count" DESC
FETCH FIRST 100 ROWS ONLY;
```

FIGURE V.1 – Requêtes 1

```
--- Quels sont les types de categories ? ---
----- Type categories

select distinct "category"
from CATEGORIES
order by "category"

-- Les categories les mieux notées -- (4 secondes)
SELECT c."category",
SUM(b."Reviews_stars_count") as total_reviews
FROM FACT_BUSINESS b
JOIN "CATEGORIES" c ON b."category_id"=c."category_id"
GROUP BY CUBE (c."category")
ORDER BY total_reviews DESC
FETCH FIRST 100 ROW ONLY;
-- Les categories ayant reçu le plus de conseils --
SELECT c."category",
SUM("TIP_compliments_count") AS total_tip_compliments
FROM FACT_BUSINESS b
JOIN "CATEGORIES" c ON b."category_id"=c."category_id"
GROUP BY CUBE (c."category")
ORDER BY total_tip_compliments DESC
FETCH FIRST 100 ROW ONLY;
```

FIGURE V.2 – Requêtes 2

```
   --- Quels sont les entreprises avec les plus de critiques utiles  par
ville,pays et Categorie ? ---
-- Entreprise les plus critiques en generals --
SELECT DISTINCT "business_name", "Reviews_useful_count"
FROM FACT_BUSINESS
ORDER BY "Reviews_useful_count" DESC
FETCH FIRST 100 ROWS ONLY;

-- Les 100 entreprises les plus de critiques utiles par ville --
SELECT DISTINCT l."city" , b."business_name", b. "Reviews_useful_count"
FROM FACT_BUSINESS b JOIN
"LOCATION" l ON b."location_id"=l."location_id"
GROUP BY l."city", b."business_name", b."Reviews_useful_count"
ORDER BY "Reviews_useful_count" DESC
FETCH FIRST 100 ROWS ONLY;

-- Les 100 entreprises les plus de critiques utiles par etat --
SELECT DISTINCT l."state",b."business_name", b."Reviews_useful_count"
FROM FACT_BUSINESS b JOIN
"LOCATION" l ON b."location_id"=l."location_id"
GROUP BY l."state", b."business_name", b."Reviews_useful_count",
b."business_id"
ORDER BY "Reviews_useful_count" DESC
FETCH FIRST 100 ROWS ONLY;

-- Les categories les plus de critiques utiles --
SELECT c."category",
SUM(b."Reviews_useful_count") as total_reviews
FROM FACT_BUSINESS b
JOIN "CATEGORIES" c ON b."category_id"=c."category_id"
GROUP BY CUBE (c."category")
ORDER BY total_reviews DESC
FETCH FIRST 100 ROW ONLY;
```

FIGURE V.3 – Requêtes 3

```
     --- Quels sont les entreprises les plus visités ? ---

-- Les 10 commerces les plus visités -- (21 secondes)
SELECT b."business_name", COUNT(*) AS num_visits
FROM FACT_BUSINESS b
JOIN CHECKIN c ON b."checkin_id" = c."checkin_id"
WHERE b."business_name" IS NOT NULL
GROUP BY ROLLUP( b."business_name")
OFFSET 0 ROWS FETCH FIRST 10 ROWS ONLY;


-- Les 10 commerces les plus visités par pays -- (26 secondes)
SELECT l."state", b."business_name", COUNT(*) AS num_visits
FROM FACT_BUSINESS b
JOIN CHECKIN c ON b."checkin_id" = c."checkin_id"
JOIN LOCATION l ON b."location_id" = l."location_id"
WHERE b."business_name" IS NOT NULL
GROUP BY ROLLUP(l."state", b."business_name")
ORDER BY l."state", num_visits DESC
OFFSET 0 ROWS FETCH FIRST 10 ROWS ONLY;


-- Les 10 commerces les plus visités par ville -- (29secondes)
SELECT l."city", b."business_name", COUNT(*) AS num_visits
FROM FACT_BUSINESS b
JOIN CHECKIN c ON b."checkin_id" = c."checkin_id"
JOIN LOCATION l ON b."location_id" = l."location_id"
GROUP BY ROLLUP(l."city", b."business_name")
ORDER BY l."city", num_visits DESC
FETCH FIRST 10 ROWS ONLY
```

FIGURE V.4 – Requêtes 4

```
--- Quel jour de la semaine reçoit le plus de visites ? ---
-- Pour afficher le jour avec le plus de check-ins --
   (4Secondes)
SELECT TO_CHAR(c."date", 'Day') AS day_of_week,
COUNT(DISTINCT b."checkin_id") AS total_checkins
FROM CHECKIN c
JOIN FACT_BUSINESS b ON c."checkin_id" = b."checkin_id"
GROUP BY TO_CHAR(c."date", 'Day')
ORDER BY COUNT(DISTINCT b."checkin_id") DESC;

-- Pour afficher le jour avec le plus de check-ins par ville --
SELECT l."city", TO_CHAR(c."date", 'Day') AS day_of_week,
COUNT(DISTINCT b."checkin_id") AS total_checkins
FROM CHECKIN c
JOIN FACT_BUSINESS b ON c."checkin_id" = b."checkin_id"
JOIN LOCATION l ON l."location_id"=b."location_id"
GROUP BY TO_CHAR(c."date", 'Day'),l."city"
ORDER BY COUNT(DISTINCT b."checkin_id") DESC;
```

FIGURE V.5 – Requêtes 5

```
--- Quels sont les entreprises qui ont reçu le plus de conseils?dans
quelle ville ? pays ? ---


-- Les entreprises ayant reçu le plus conseil en generale -- (21secondes)
SELECT "business_id",
"business_name",
SUM("TIP_compliments_count") AS total_tip_compliments
FROM FACT_BUSINESS
GROUP BY CUBE ("business_id", "business_name")
ORDER BY total_tip_compliments DESC
FETCH FIRST 100 ROW ONLY;
```

FIGURE V.6 – Requêtes 6

Requetes d'Analyse des Utilisateurs Yelp

Les requêtes de la 1ère datamarts qui analyse les activités des entreprise sont présentées ci-dessous :

```
-- Quel utilisateur est le plus influent en termes de nombre de fans et
d'années d'élite --
 (0,8 seconde)

SELECT fu."user_id",
fu."name",
fu."fans",
COUNT(e."year") AS elite_years
FROM FACT_USERS fu
JOIN ELITE e ON fu."elite_id" = e."elite_id"
GROUP BY fu."user_id", fu."name", fu."fans"
ORDER BY fu."fans" DESC, elite_years DESC;


-- utilisateurs avec le plus grands nombre de reviews count --
  (3S)

SELECT
"name",
SUM("review_count") AS total_reviews
FROM
FACT_USERS
GROUP BY
CUBE ("name");
```

FIGURE V.7 – Requêtes 1

```
-- Date yelping la plus frequente --
 (0,6 secondes)


SELECT
t."yelping_since",
COUNT(*) AS frequency
FROM
FACT_USERS u
JOIN
TIME t ON u."date_id" = t."date_id"
GROUP BY
t."yelping_since"
ORDER BY
frequency DESC;
```

FIGURE V.8 – Requêtes 2

```
-- Jour de debut yelping le plus frequent --

SELECT
TO_CHAR(t."yelping_since", 'DAY') AS day_of_week,
COUNT(*) AS frequency
FROM
FACT_USERS u
JOIN
TIME t ON u."date_id" = t."date_id"
GROUP BY
t."yelping_since", TO_CHAR(t."yelping_since", 'DAY')
ORDER BY
frequency DESC


-- utilisateur ayant le plus de fans et de critiques -

SELECT
fu."user_id",
fu."name",
fu."fans",
fu."review_count"
FROM
FACT_USERS fu
JOIN
ELITE e ON fu."elite_id" = e."elite_id"
GROUP BY ("name","user_id","fans","review_count")
ORDER BY
fu."fans" DESC,
fu."review_count" DESC
```

FIGURE V.9 – Requêtes 3

```
-- Utilisateurs avec la date de la dernière critique et la plus grande
moyenne d'étoiles -

SELECT
fu."user_id",
MAX(r."date") AS latest_review_date,
fu."average_stars",
TO_CHAR(r."review_id") AS review_id
FROM
FACT_USERS fu
JOIN
REVIEW r ON fu."review_id" = TO_CHAR(r."review_id")
GROUP BY
fu."user_id", fu."average_stars",TO_CHAR(r."review_id")
ORDER BY
fu."average_stars" DESC;
```

FIGURE V.10 – Requêtes 4

2 Visualisation sur Metabase

2.1 Introduction

Metabase est un outil open source utilisé en informatique décisionnelle (Business Intelligence) et pour l'analyse de données, permettant aux utilisateurs de créer des tableaux de bord, des rapports et des visualisations personnalisés à partir de diverses sources de données.

2.2 Avantages et Inconvénients

Quelques avantages et inconvénients de l'utilisation de Metabase dans un contexte professionnel sont les suivants :

- **Avantages :**
  - Interface intuitive (User-friendly)
  - Langage de requête accessible
  - Solution open source
  - Intégration fluide
  - Options de visualisation étendues
- **Inconvénients :**
  - Limitations de performances

2.3 Application au TP

Nous utilisons Metabase dans ce projet afin d'offrir aux utilisateurs une visualisation graphique, dans des tableaux de bord optimisés, des résultats des requêtes créées dans la partie précédente. Nous transformons nos données en mesures concrètes qui amélioreront le rendement et faciliteront la compréhension des utilisateurs qui pourront extraire des informations et des connaissances utiles à partir des données brutes.

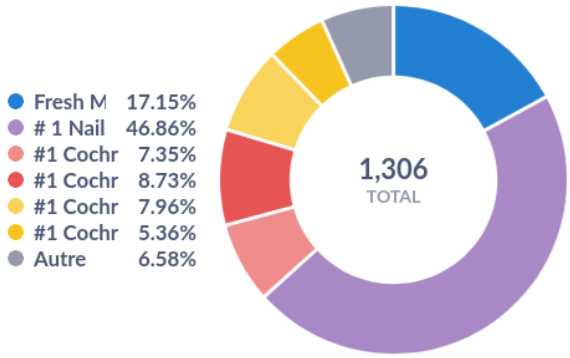

FIGURE V.11 – resultat 1

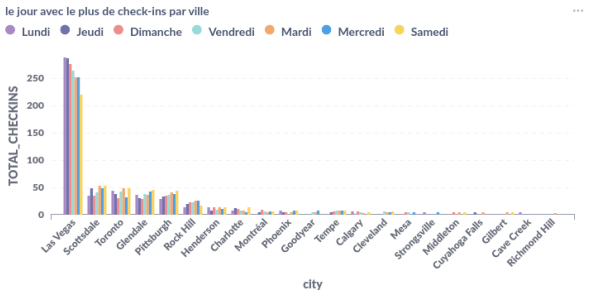

FIGURE V.12 – resultat 2

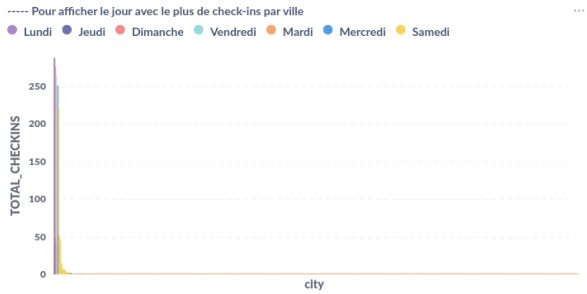

FIGURE V.13 – resultat 3

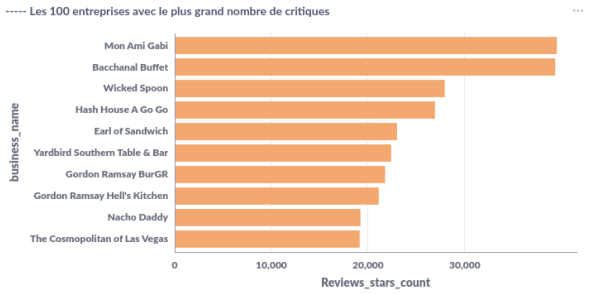

FIGURE V.14 – resultat 4

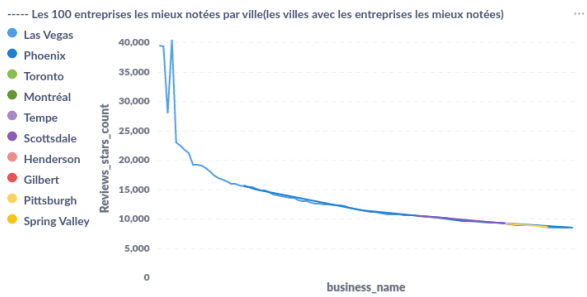

FIGURE V.15 – resultat 5

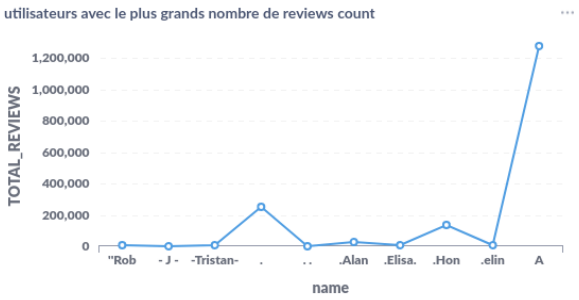

FIGURE V.16 – resultat 6

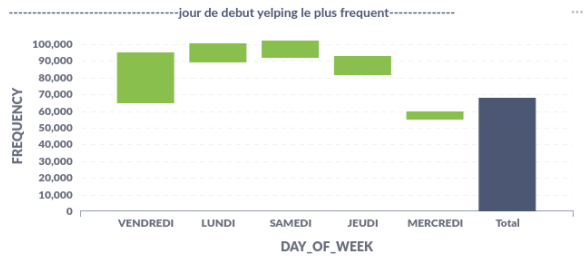

FIGURE V.17 – resultat 7

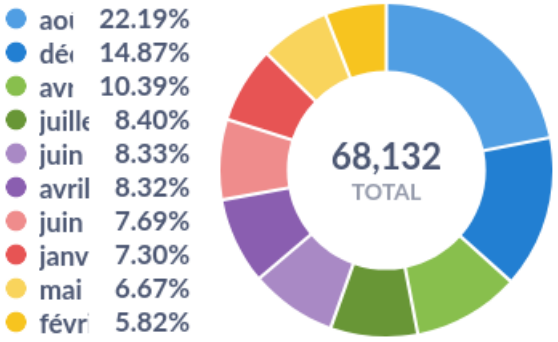

FIGURE V.18 – resultat 8

VI Évaluation des performances

:

L'opération de chargement du data warehouse, effectuée dans un délai inférieur à une heure, témoigne de la rapidité et de l'efficacité du processus d'intégration des données. Cette opération, qui consiste à extraire, transformer et charger (ETL) de grandes quantités de données dans le data warehouse, est cruciale pour garantir la disponibilité des informations pour l'analyse.

De plus, en examinant les performances des requêtes exécutées sur le data warehouse, nous observons que la requête la plus longue prend seulement 30 secondes pour s'exécuter, tandis que la requête la plus rapide ne nécessite que 0,6 seconde. Ces temps de réponse courts indiquent une optimisation réussie de la structure du data warehouse et des index, ainsi qu'une bonne répartition de charge des requêtes.

En conclusion, la rapidité du chargement des données et les temps de réponse rapides des requêtes démontrent que le système est bien conçu et performant, fournissant ainsi une base solide pour l'analyse et l'exploration des données.

VII Évaluation des performances

1 Évaluation de l'occupation mémoire

USER :
- **Nombre de lignes :** 1,968,703
- **Taille des colonnes :**
  - "user_id" : 43,311,466 octets
  - "name_user" : 10,503,243 octets
  - "yelping_since" : 13,780,921 octets
- **Calculs :**
  - Taille totale pour "user_id" : 452,514,117,998,598 octets
  - Taille totale pour "name_user" : 109,667,649,612,129 octets
  - Taille totale pour "yelping_since" : 271,158,970,804,263 octets
- **Taille totale de la table USER :** 832,340,738,415,990 octets

TIP :
- **Nombre de lignes :** 1,293,604
- **Taille des colonnes :**
  - "tip_id" : 15,928,523 octets
  - "text" : 80,064,647 octets
  - "date" : 24,567,760 octets
- **Calculs :**
  - Taille totale pour "tip_id" : 203,194,484,429,692 octets
  - Taille totale pour "text" : 103,552,949,733,188 octets
  - Taille totale pour "date" : 317,809,431,763,840 octets
- **Taille totale de la table TIP :** 624,556,865,926,720 octets

REVIEWS :
- **Nombre de lignes :** 8,021,122
- **Taille des colonnes :**
  - "review_id" : 176,464,684 octets
  - "text" : 4,746,216,980 octets
  - "date" : 56,147,854 octets
- **Calculs :**
  - Taille totale pour "review_id" : 1,414,631,052,832,648 octets
  - Taille totale pour "text" : 38,072,694,780,562,560 octets
  - Taille totale pour "date" : 449,947,474,868,988 octets
- **Taille totale de la table REVIEWS :** 39,937,273,308,264,196 octets

CHECKINS :
- **Nombre de lignes :** 3,620
- **Taille des colonnes :**
  - "checkin_id" : 26,710 octets
  - "date" : 25,340 octets
- **Calculs :**
  - Taille totale pour "checkin_id" : 95,146,200 octets
  - Taille totale pour "date" : 91,352,800 octets
- **Taille totale de la table CHECKINS :** 186,499,000 octets

FACT_BUSINESS :
- **Nombre de lignes :** 40,741,919

- **Taille des colonnes :**
  - "user_id" : 4,331,146 octets
  - "business_id" : 10,503,243 octets
  - "category_id" : 13,780,921 octets
  - "checkin_id" : 4,280,936 octets
  - "hours_id" : 3,651,142 octets
  - "TIP_compliments_count" : 389,180 octets
  - "tip_id" : 1,070,775 octets
  - "business_name" : 776,248 octets
  - "location_id" : 1,069,579 octets
  - "review_id" : 1,435,940 octets
  - "Reviews_useful_count" : 267,698,822 octets
  - "Reviews_cool_count" : 105,032,430 octets
  - "Reviews_funny_count" : 137,809,210 octets
  - "Reviews_stars_count" : 30,176,461 octets
  - "Total_UsersReviews_count" : 3,584,961 octets
  - "Total_Usersfans_Count" : 21,890,850 octets
- **Calculs :**
  - Taille totale pour "user_id" : 176,850,152,842,774 octets
  - Taille totale pour "business_id" : 425,284,124,425,517 octets
  - Taille totale pour "category_id" : 561,369,822,138,999 octets
  - Taille totale pour "checkin_id" : 174,135,627,759,384 octets
  - Taille totale pour "hours_id" : 148,750,336,748,658 octets
  - Taille totale pour "TIP_compliments_count" : 15,985,360,165,820 octets
  - Taille totale pour "tip_id" : 43,595,370,475,025 octets
  - Taille totale pour "business_name" : 31,824,297,594,312 octets
  - Taille totale pour "location_id" : 43,573,977,401,801 octets
  - Taille totale pour "review_id" : 58,675,173,925,560 octets
  - Taille totale pour "Reviews_useful_count" : 10,917,264,977,800,418 octets
  - Taille totale pour "Reviews_cool_count" : 4,277,517,119,320,470 octets
  - Taille totale pour "Reviews_funny_count" : 5,618,472,663,686,290 octets
  - Taille totale pour "Reviews_stars_count" : 1,230,316,286,681,759 octets
  - Taille totale pour "Total_UsersReviews_count" : 146,142,161,533,959 octets
  - Taille totale pour "Total_Usersfans_Count" : 892,040,528,228,750 octets
- **Taille totale de la table FACT_BUSINESS :** 1,049,724,405,568,612 octets

CATEGORIES :

- **Nombre de lignes :** 1,336
- **Taille des colonnes :**
  - "category" : 4,280,936 octets
  - "categoryid" : 4,064,702 octets
- **Calculs :**
  - Taille totale pour "category" : 5,716,178,496 octets
  - Taille totale pour "categoryid" : 5,435,972,672 octets
- **Taille totale de la table CATEGORIES :** 11,152,151,168 octets

HOURS :

- **Nombre de lignes :** 209,393

- **Taille des colonnes :**
- "Monday" : 3,651,142 octets
- "Tuesday" : 389,180 octets
- "Wednesday" : 1,070,775 octets
- "Thursday" : 776,248 octets
- "Friday" : 1,069,579 octets
- "Saturday" : 1,435,940 octets
- "Sunday" : 4,331,146 octets
- "hours_id" : 3,033,527 octets

Calculs :

- Taille totale pour "Monday" : 763,902,552,606 octets
- Taille totale pour "Tuesday" : 81,474,143,140 octets
- Taille totale pour "Wednesday" : 224,240,375,175 octets
- Taille totale pour "Thursday" : 162,561,960,664 octets
- Taille totale pour "Friday" : 223,920,842,447 octets
- Taille totale pour "Saturday" : 300,642,330,020 octets
- Taille totale pour "Sunday" : 907,619,648,318 octets
- Taille totale pour "hours_id" : 635,831,591,111 octets
- **Taille totale de la table HOURS :** 4,339,031,997,731 octets

LOCATION :

- **Nombre de lignes :** 194,588
- **Taille des colonnes :**
- "city" : 1,626,050 octets
- "state" : 389,180 octets
- "postal_code" : 1,070,775 octets
- "latitude" : 776,248 octets
- "longitude" : 1,069,579 octets
- "location_id" : 1,435,940 octets
- **Calculs :**
  - Taille totale pour "location" : 787,026,117,976 octets
  - Taille totale pour "city" : 315,831,223,800 octets
  - Taille totale pour "state" : 75,763,084,840 octets
  - Taille totale pour "postal_code" : 208,236,546,100 octets
  - Taille totale pour "latitude" : 151,111,352,224 octets
  - Taille totale pour "longitude" : 208,125,443,652 octets
  - Taille totale pour "location_id" : 279,106,585,520 octets

Taille totale de la table LOCATION : 3,562,374,261,476 octets

VIII Documentation technique

1 Architecture logicielle

Dans cette partie, plusieurs classes sont définies pour gérer les opérations d'Extraction, Transformation et Chargement (ETL) sur différentes données pertinentes du système. Voici une description détaillée de chaque classe :

1.1 ETL_LOCATIONS :

Cette classe gère les opérations ETL sur les données de localisation des entreprises. Méthode extract : sélectionne les colonnes pertinentes du DataFrame businessDF. Méthode transform : élimine les doublons et ajoute une colonne "location_id". Méthode load : charge les données transformées dans la table Oracle "LOCATION". Méthode joinWithBusiness : renvoie un DataFrame avec les colonnes "business_id", "business_name" et "location_id".

1.2 ETL_TIP :

Cette classe gère les opérations ETL pour les conseils (tips). Méthode transform : élimine les doublons et ajoute une colonne "tip_id". Méthode load : charge les données transformées dans la table Oracle "TIP". Méthode joinWithBusiness : agrège les données de conseils par identifiant d'entreprise.

1.3 ETL_HOURS :

Cette classe gère les opérations ETL pour les heures d'ouverture des entreprises. Méthode extract : crée une colonne structurée pour chaque jour de la semaine. Méthode transform : pivote les données et ajoute une colonne "hours_id". Méthode load : charge les données transformées dans la table Oracle "HOURS". Méthode joinWithBusiness : renvoie un DataFrame avec les colonnes "business_id" et "hours_id".

1.4 ETL_TIME :

Cette classe gère les opérations ETL pour les informations temporelles des utilisateurs. Méthode extract : sélectionne les colonnes "user_id" et "yelping_since". Méthode transform : ajoute des colonnes supplémentaires et un identifiant unique "date_id". Méthode load : charge les données transformées dans la table Oracle "TIME". Méthode joined : renvoie un DataFrame avec les colonnes "user_id" et "date_id".

1.5 ETL_ELITE :

Cette classe gère les opérations ETL pour les utilisateurs élites. Méthode extract : supprime les doublons et ajoute une colonne "elite_id". Méthode load : charge les données transformées dans la table Oracle "elite". Méthode joined : renvoie un DataFrame avec les colonnes "user_id" et "elite_id".

Nous avons deux classes ETL_REVIEWS pour assurer la cohérence entre les deux datamarts. Cette cohérence est établie grâce à la jointure de la table DW_BUS au niveau

du DW qui retourne à travers des méthodes les colonnes nécessaires pour chaque analyse (datamart), entre les avis des utilisateurs et ceux des entreprises, qui ont été conservés pour l'analyse des données dans les deux datamarts

1.6 ETL_REVIEWS :

Cette classe gère les opérations ETL pour les avis (reviews). Méthode transform : élimine les doublons basés sur les colonnes "review_id" et "business_id". Méthode load : charge les données transformées dans la table Oracle "Reviews". Méthode joinWithBusiness : agrège les données d'avis par identifiant d'entreprise et sélectionne les colonnes pertinentes.

1.7 ET_REVIEWS2 :

Seconde implémentation pour gérer les opérations ETL pour les avis (reviews). Méthode extract : renvoie les données complètes du DataFrame initial. Méthode joined : agrège les données d'avis par identifiant utilisateur et sélectionne les colonnes pertinentes. Méthode joinedDWBUS : sélectionne uniquement les colonnes "review_id" et "user_id". Méthode load : sauvegarde les données transformées dans la base de données Oracle.

1.8 DW_BUS :

Cette classe gère la création d'un Data Warehouse (DW) pour les données d'avis. Méthode joinDWBUS : joint les DataFrames reviewDF1 et reviewDF2 sur la colonne "review_id" en utilisant une jointure interne. Méthode joinedUserFacts : renvoie un DataFrame avec les faits concernant les utilisateurs en supprimant la colonne "business_id".

1.9 Main :

La classe Main est le cœur de notre processus ETL (Extract, Transform, Load) pour l'analyse des données. Elle coordonne les différentes étapes de traitement des données et assure leur chargement dans la base de données cible. Voici un aperçu des principales responsabilités de cette classe :

Initialisation de la session Spark : La méthode principale main de la classe Main commence par initialiser une session Spark, permettant ainsi l'interaction avec les données de manière distribuée.

Chargement des données : Les données sont extraites à partir de diverses sources telles que des fichiers CSV, des fichiers JSON et des bases de données PostgreSQL. Elles sont ensuite chargées dans des DataFrames Spark pour traitement.

Coordination du processus ETL : La classe Main utilise différentes classes ETL pour effectuer les opérations d'extraction, de transformation et de chargement des données. Ces opérations incluent le nettoyage des données, la suppression des doublons et la préparation des données pour le chargement dans la base de données cible.

Gestion des erreurs : Un mécanisme de gestion des erreurs est mis en place pour capturer et gérer toute exception pouvant survenir pendant le processus ETL, garantissant ainsi la robustesse de l'application.

Chargement des données dans la base de données cible : Une fois les données transformées, elles sont chargées dans une base de données Oracle en utilisant des requêtes JDBC (Java Database Connectivity).

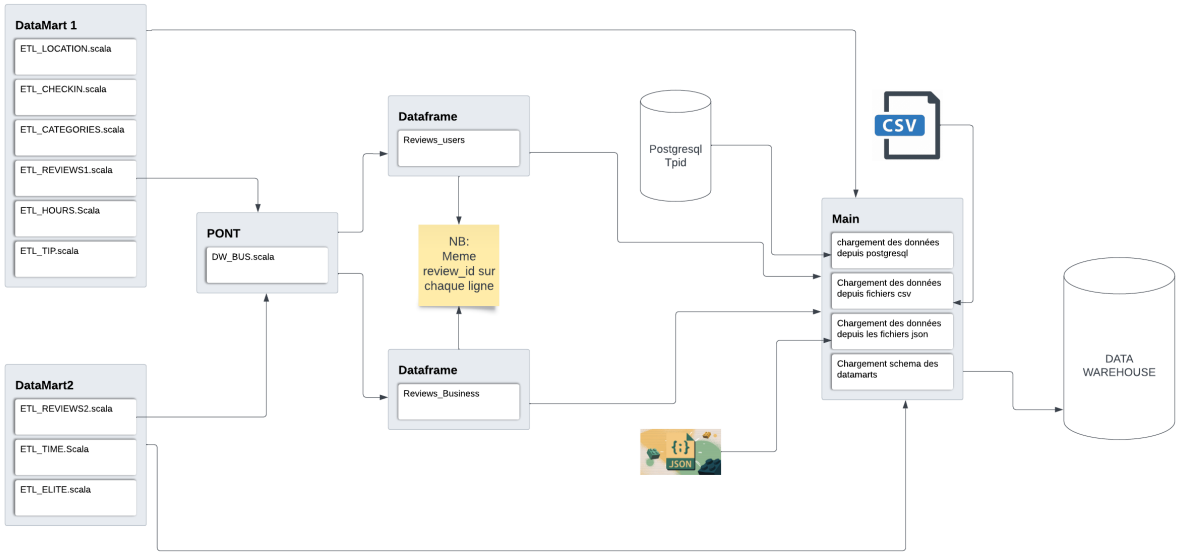

FIGURE VIII.1 – Architecture

2 Instructions pour compiler et exécuter le code

Pour compiler et exécuter le projet , suivez ces étapes :

1. Assurez-vous que vous avez déjà configuré votre projet Scala avec Spark et PostgreSQL et que votre code est prêt à être compilé et exécuté.
2. Ouvrez un terminal et accédez au répertoire racine du code source.
3. Exécutez la commande suivante pour compiler votre code :

   ```
   sbt clean compile
   ```

   Cela téléchargera les dépendances requises et compilera votre code Scala.
4. Une fois la compilation réussie, exécutez en utilisant la commande suivante :

   ```
   sbt run
   ```

   Pour ceux qui n'auront pas accès aux bases de données spécifiées dans le code source,il est possible de modifier les informations de connexions avec celles de vos bases de données personnelles

IX Conclusion

1 Récapitulation des principaux résultats

Résultat 1 : Les 10 commerces les plus visités : Cette analyse permet de comprendre quels sont les commerces les plus populaires, ce qui peut être utile pour les propriétaires d'entreprise afin d'ajuster leur offre ou leur marketing en conséquence, et pour les utilisateurs cherchant des recommandations populaires.

Résultat 2 : Le jour avec le plus de check-ins par ville : Cette information est précieuse pour comprendre les tendances de fréquentation des commerces par jour et par ville, aidant ainsi les entreprises à mieux planifier leurs horaires et leurs promotions.

Résultat 3 : Afficher le jour avec le plus de check-ins par ville : Similaire au résultat précédent, cette analyse offre des perspectives sur les jours les plus actifs pour chaque ville, ce qui peut aider les entreprises à mieux gérer leurs ressources et à anticiper les fluctuations de la demande.

Résultat 4 : Les 100 entreprises avec le plus grand nombre de critiques : Cette liste fournit un aperçu des entreprises qui suscitent le plus d'intérêt et d'engagement de la part des utilisateurs, ce qui peut être utilisé pour évaluer la réputation et l'attrait d'une entreprise.

Résultat 5 : Les 100 entreprises les mieux notées par ville : Cela permet de mettre en lumière les entreprises qui ont la meilleure réputation dans chaque ville, ce qui peut être utile pour les consommateurs à la recherche des meilleures options, et pour les entreprises à la recherche de feedback positif.

Résultat 6 : Utilisateurs avec le plus grands nombre de reviews count : Cette analyse permet d'identifier les utilisateurs les plus actifs et influents sur la plateforme, ce qui peut être utile pour les entreprises cherchant à engager des partenariats avec des utilisateurs influents ou à cibler des segments de marché spécifiques.

Résultat 7 : Jour de début Yelping le plus fréquent : Connaître le jour le plus fréquent pour le début de l'utilisation de Yelp permet de comprendre les habitudes des nouveaux utilisateurs, ce qui peut être exploité pour des campagnes de marketing ou pour améliorer l'expérience utilisateur.

Résultat 8 : Date Yelping la plus fréquente : Similaire au résultat précédent, cette analyse offre des informations sur les périodes où de nouveaux utilisateurs commencent à utiliser Yelp, ce qui peut être utilisé pour mieux comprendre les tendances saisonnières ou les événements influençant l'adoption de la plateforme.

2 Perspectives futures pour le projet

Pour poursuivre ce projet et en maximiser les bénéfices, plusieurs perspectives d'évolution peuvent être envisagées. Tout d'abord, il serait intéressant d'explorer davantage les données en intégrant des sources externes pour obtenir une vue plus holistique du marché. De plus, l'utilisation de techniques d'analyse avancées telles que l'apprentissage automatique pourrait permettre de dégager des insights plus précis et prédictifs. Par ailleurs,il serait pertinent d'explorer des possibilités de collaboration avec d'autres plateformes ou entreprises pour enrichir davantage les données et les analyses réalisées.

3

En conclusion, notre expérience de collaboration sur le projet d'analyse des données Yelp a été extrêmement enrichissante à la fois sur le plan professionnel et personnel. Travailler en binôme nous a permis de combiner nos compétences, nos idées et nos perspectives pour mener à bien ce projet de manière efficace et créative.

Nous avons pu constater l'importance et les avantages de travailler en équipe dans le domaine de l'analyse des données. La collaboration nous a non seulement permis de répartir les tâches de manière efficiente, mais elle nous a également offert un environnement propice à l'apprentissage mutuel et à l'échange d'idées.

Ensemble, nous avons pu surmonter les défis rencontrés et atteindre nos objectifs avec succès. Cette expérience de travail collaboratif renforce notre conviction dans les bénéfices de la collaboration et nous encourage à continuer à travailler en équipe dans nos projets futurs.

Enfin, ce projet nous a permis de développer nos compétences en analyse de données, de mieux comprendre l'importance des données dans les décisions commerciales et d'apprécier l'impact positif que notre travail peut avoir sur les entreprises et les utilisateurs.

X Annexes

0.1 Lien vers code source :

https ://filesender.renater.fr/ ?s=downloadtoken=5725622f-2a36-414d-a4c1-ff256c77c6be

1 Bibliographies

Astera : "Star Schema"

Lien : https ://www.astera.com/fr/knowledge-center/star-schema/ : :text=IntroduitCet article explore en profondeur le schéma en étoile, une technique de modélisation de base de données populaire utilisée dans les entrepôts de données. Il offre des informations précieuses sur la conception, les avantages et les meilleures pratiques liées au schéma en étoile.

GeeksforGeeks : "Data Cube or OLAP approach in Data Mining"

Lien : https ://www.geeksforgeeks.org/data-cube-or-olap-approach-in-data-mining/ Cette ressource offre une explication détaillée de l'approche du cube de données (OLAP) en exploration de données, fournissant une compréhension approfondie des concepts clés liés à l'agrégation multidimensionnelle.

LearnDataModeling.com : "Time Dimension"

Lien : https ://learndatamodeling.com/blog/time-dimension/google$_v ignetteCetteressourcefournit$