



M1 INFORMATIQUE AIGLE

HMIN122M

MINI-PROJET : ENTREPÔTS DE DONNÉES

Rendu sur la modélisation d'un entrepôts de données

Bachar RIMA

Joseph SABA

Tasnim SHAQURA MUHAMMAD

Jérémy BOURGIN

23 octobre 2018

Table des matières

Introduction	2
Questions	3
Question 1	3
Question 2	3
Question 3	4
Question 4	4
Questions 5 et 6	4
<i>Data Mart</i> de « voyages »	4
<i>Data Mart</i> de « maintenance »	5
<i>Data warehouse</i> de <i>tam-voyages</i>	6
Question 7	8
Question 8	9
Pour le <i>data mart</i> de « voyages »	9
Pour le <i>data mart</i> de « maintenance »	10

Introduction

Dans le cadre du mini-projet du module **HMIN122M**, nous avons décidé de modéliser un entrepôt de données pour le réseau de transport public de Montpellier, *tam-voyages*. Pour ce faire, nous avons proposé des *data marts* formant le *data warehouse* et permettant de réaliser des requêtes analytiques sur un ensemble important de données. Cette modélisation permettra ainsi de mettre en œuvre un outil d'analyse permettant de bien répondre aux problématiques suivantes :

1. Comment *tam-voyages* pourront-ils augmenter leur taux de ventes en se basant sur la circulation du réseau¹ ?
2. Comment *tam-voyages* pourront-ils suivre l'évolution et la maintenance de leurs matériaux de manière à réduire les dépenses qui y sont associées ?

Ces problématiques seront ainsi adressées en analysant les actions et opérations effectuées par *tam-voyages*, notamment en choisissant celles qui paraissent les plus pertinentes et les plus importantes en termes de données intégrées et flexibilité des critères d'analyse.

1. en particulier en examinant les lignes de tramway et les bus

Questions

Question 1

Les actions/opérations effectuées par *tam-voyages* considérées :

- Les voyages.
- La maintenance de véhicules.
- Les ventes de tickets et les abonnements.
- Les amendes.

Question 2

1. exemples de requêtes analytiques pour l'action « voyages » :
 - le nombre de voyageurs par bus, utilisant des tickets pour le mois de juillet.
 - le prix moyen par voyage pendant les vacances de Noël de 2018. :
 - le nombre de voyageurs abonnés par ligne pour chaque voyage pour les deux derniers mois.
 - l'arrêt le plus fréquenté par toutes les lignes de circulation.
2. exemples de requêtes analytiques pour l'action « maintenance » :
 - le nombre de bus maintenus pour le mois de septembre 2018.
 - les X mécaniciens les plus expérimentés convoqués pour la maintenance des bus l'année précédente.
 - les X premières véhicules nécessitant le plus de maintenance pour les 6 dernier mois.
3. exemples de requêtes analytiques pour l'action « ventes » :
 - le nombre d'abonnés ayant plus que 26 ans pour le mois d'août 2018.
 - le nombre d'abonnés par date de naissance pour l'année 2018.
 - les types d'abonnement les plus fréquents pour l'année 2018.
4. exemples de requêtes analytiques pour l'action « amendes » :

- les lignes qui ont générées le plus d’amendes pour les deux derniers mois.
- les lignes les plus contrôllées de la semaine dernière.
- le nombre des abonnés qui ont reçu des amendes par ligne, l’avant-midi.
- la somme total d’amendes rapportée par type de voyageur par ligne pour le dernier mois.

Question 3

Les actions considérées, par ordre d’importance :

1. « voyages ».
2. « ventes ».
3. « maintenance ».
4. « amendes ».

Question 4

Les actions les plus pertinentes à analyser vis-à-vis les problématiques avancées sont « voyages » et « maintenance » qu’on traitera de la manière suivante :

voyages : modèle en étoile détaillé.

maintenance : modèle en étoile *moins* détaillé, en particulier le modèle intitulé "*periodic snapshot*".

Questions 5 et 6

Data Mart de « voyages »

Les mesures de la table des voyages sont :

- `travel_cost` : additive

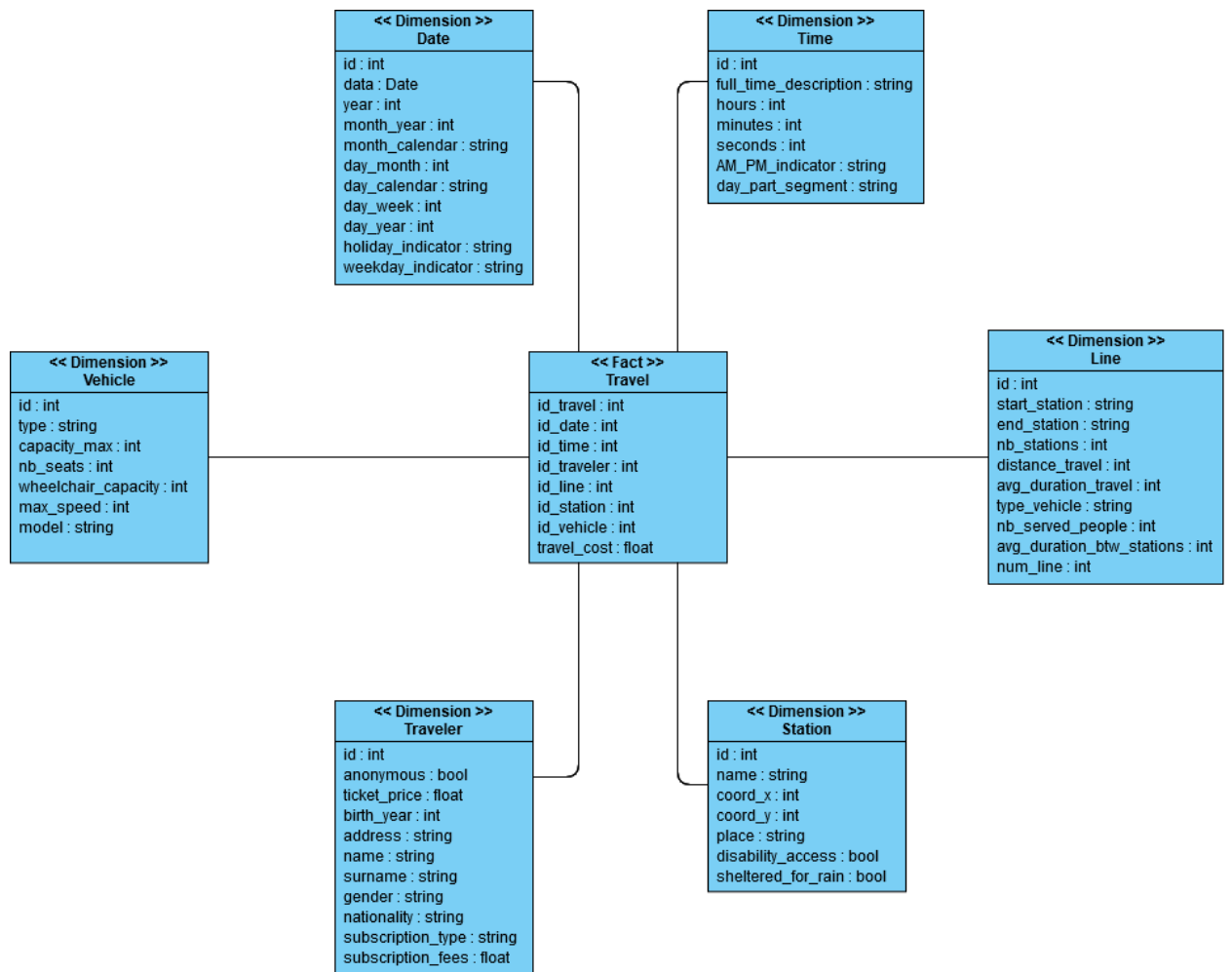


FIGURE 1 – modèle en étoile de l'action « voyages »

Data Mart de « maintenance »

Les mesures de la table des maintenances sont :

- `cost` : additive
- `estimated_time` : additive

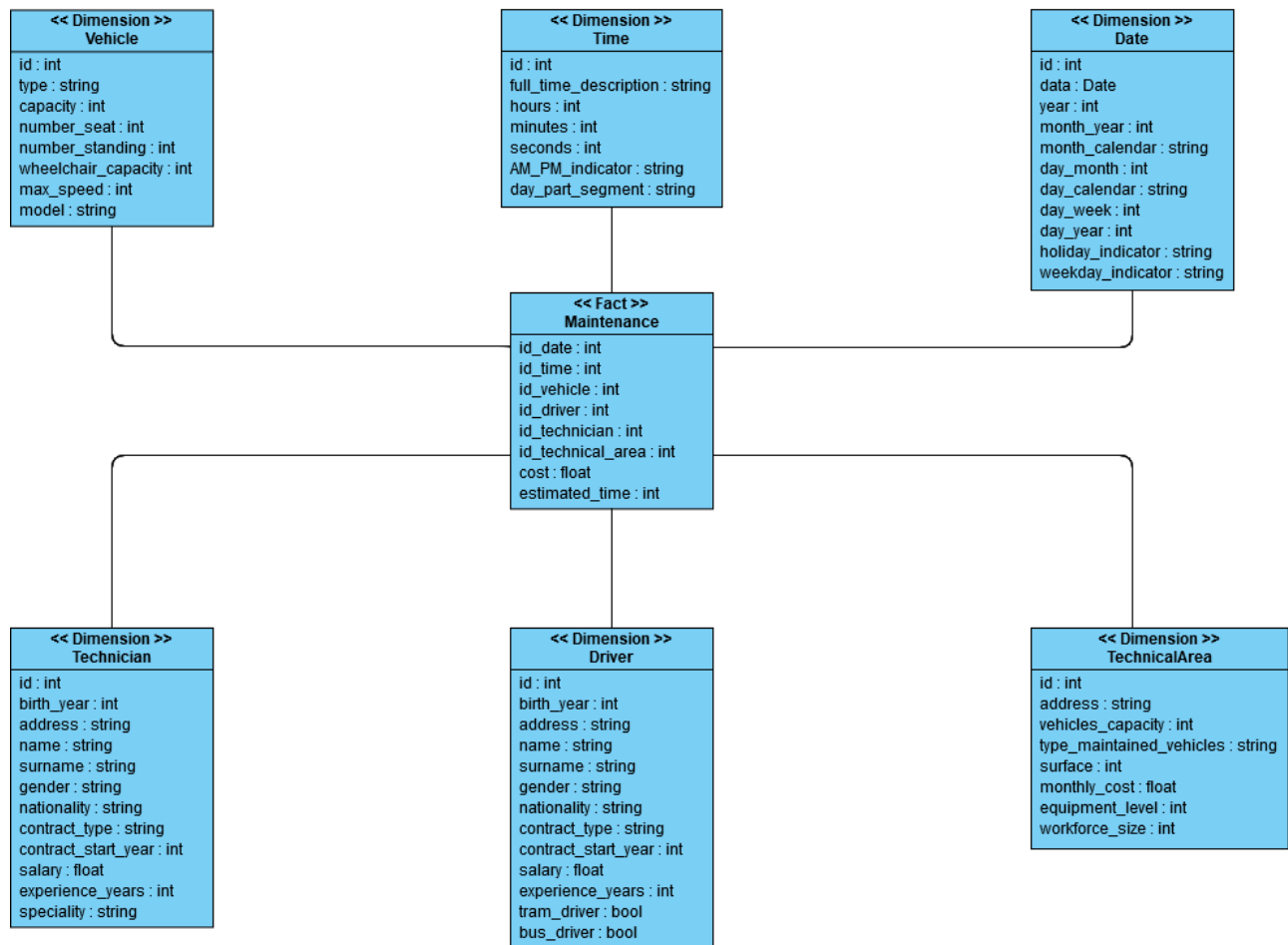


FIGURE 2 – modèle en étoile de l'action « maintenance »

Data warehouse de tam-voyages

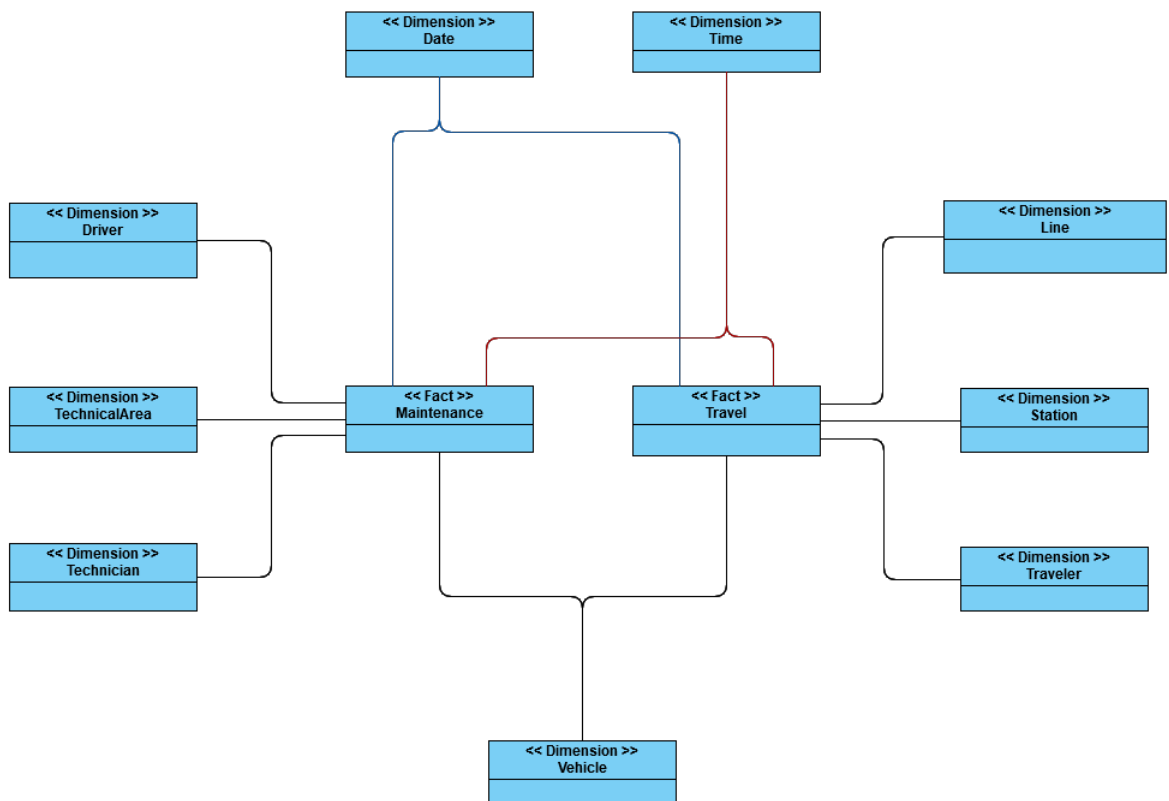


FIGURE 3 – le *data warehouse* résultant

Remarques

- l'attribut **id_travel** de la table **Travel** est la clé primaire utilisé pour identifier un voyage (*dimension dégénérée*).
- l'attribut **travel_cost** de la table **Travel** désigne le coût du voyage.
- l'attribut **avg_served_people** de la table **Line** désigne le nombre de passagers en moyenne desservis par la ligne.
- l'attribut **place** de la table **Station** désigne l'endroit où se trouve la station (avenue *X*, rue *Y*, ...).
- l'attribut **wheelchair_capacity** de la table **Vehicle** désigne la capacité théorique maximale de personnes handicapées et de leurs fauteuils roulants.
- la table **Traveler** est une dimension qui contient deux dimensions corrélées (les abonnés et les non abonnés) :
 1. si le tuple désigne un **voyageur abonné**, alors on traite le tuple en tant qu'un **voyageur concret** dont les informations sont à notre disposition.

2. si le tuple désigne un **voyageur non abonné**, alors on traite le tuple en tant qu'un **type de voyageur** défini par le ticket qu'il utilise pour faire le trajet.
 3. cette décision de corrélation est utilisée pour éviter la normalisation et l'introduction d'une superclasse abstraite étendue par les classes désignant les voyageurs abonnés et non abonnés.
 4. nous utilisons ainsi l'attribut **anonymous** afin de distinguer les deux types de voyageurs. En effet, **anonymous** valera *true* quand le voyageur est non abonné, sinon il valera *false*.
 5. les tuples de cette table contiendront ainsi des valeurs nulles pour certains attributs selon le type de voyageur.
- l'attribut **equipment_level** de la table **TechnicalArea** désigne le niveau de matériaux disponibles au local et pouvant prendre une valeur entre 1 (*pas assez équipé*) et 5 (*très bien équipé*).

Question 7

Avec la modélisation proposée de notre *data warehouse* il est bien possible de répondre aux traitements analytiques proposés :

Pour l'action « voyages »

1. le nombre de voyageurs par bus, utilisant des tickets pour le mois de juillet :
 - la table **Travel** contient l'identifiant d'une date, l'identifiant d'un véhicule et l'identifiant d'un voyageur utilisant un ticket (ou pas).
 - les attributs des tables respectives, ainsi que l'utilisation des fonctions d'agrégation, permettent bien de répondre à cette requête.
2. le prix moyen par voyage pendant les vacances de Noël de 2018 :
 - la table **Travel** contient l'identifiant d'une date et la mesure du prix d'un voyage **travel_cost**.
 - les attributs des tables respectives, ainsi que l'utilisation des fonctions d'agrégation, permettent bien de répondre à cette requête.
3. le nombre de voyageurs abonnés par ligne pour chaque voyage pour les deux derniers mois :
 - la table **Travel** contient l'identifiant d'une date, l'identifiant d'une ligne l'identifiant d'un voyageur abonné (ou pas).
 - les attributs des tables respectives, ainsi que l'utilisation des fonctions d'agrégation, permettent bien de répondre à cette requête.

4. l'arrêt le plus fréquenté par toutes les lignes de circulation :
 - la table `Travel` contient l'identifiant d'une ligne et l'identifiant d'un arrêt où un voyageur a monté à bord.
 - les attributs des tables respectives, ainsi que l'utilisation des fonctions d'agrégation, permettent bien de répondre à cette requête.

Pour l'action « maintenance »

1. le nombre de bus maintenus pour le mois de septembre 2018 :
 - la table `Maintenance` contient l'identifiant d'une date et l'identifiant d'un véhicule.
 - les attributs des tables respectives, ainsi que l'utilisation des fonctions d'agrégation, permettent bien de répondre à cette requête.
2. les X mécaniciens les plus expérimentés convoqués pour la maintenance des bus l'année précédente :
 - la table `Maintenance` contient l'identifiant d'une date, l'identifiant d'un véhicule et l'identifiant d'un technicien.
 - les attributs des tables respectives, ainsi que l'utilisation des fonctions d'agrégation, permettent bien de répondre à cette requête.
3. les X premières véhicules nécessitant le plus de maintenance pour les 6 dernier mois :
 - la table `Maintenance` contient l'identifiant d'une date et l'identifiant d'un véhicule.
 - les attributs des tables respectives, ainsi que l'utilisation des fonctions d'agrégation, permettent bien de répondre à cette requête.

Question 8

Pour le *data mart* de « voyages »

id_line	num_line	type_vehicle	start_station	end_station	nb_stations	distance_travel	avg_duration_travel	avg_duration_btw_stations	avg_served_people
101	7	bus	Hôtel du département	La Martelle	34	1510	45	5	40
102	7	bus	Hôtel du département	Les Bouisses	34	1515	45	4	40
201	1	tram	Odysseum	Mosson	30	15700	50	2	150

TABLE 1 – Line

id_station	name	coord_x	coord_y	place	disability_access	sheltered_for_rain
1	Hôtel du département	43.622108	3.835276	Avenue des moulins	false	true
2	Pergola	43.617558	3.839687	Rue Paul Rimbaud	false	true
3	Odysseum	43.603687	3.921722	Place de Lisbonne	true	true

TABLE 2 – Station

id_vehicle	type	model	max_speed	nb_seats	standing_capacity	wheelchair_capacity
1	bus	TransBus Enviro30	200	20	40	20
2	tram	Alstom Citadis 401	300	100	150	150

TABLE 3 – Vehicle

id_traveler	anonymous	name	surname	birth_year	gender	address	nationality	subscription_type	subscription_fees	ticket_price
1	<i>true</i>	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	10
2	<i>true</i>	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1.60
100	<i>false</i>	Jean	Toto	1995	M	16 avenue de titi	France	contrat mobilité jeune	196	NULL
101	<i>false</i>	Jane	Tutu	1987	F	2 boulevard de nyehe	Angleterre	contrat mobilité pour tous	481.50	NULL

TABLE 4 – Traveler

id_date	date	year	month_year	month_calendar	day_month	day_calendar	day_week	day_year	holiday_indicator	weekday_indicator
1	21/10/2018	2018	10	Octobre	21	Dimanche	0	294	Non Holiday	Weekend
2	22/10/2018	2018	10	Octobre	22	Lundi	1	295	Non Holiday	Weekday
3	31/10/2018	2018	10	Octobre	31	Mercredi	3	295	Holiday	Weekday

TABLE 5 – Date

id_time	full_time_description	hours	minutes	seconds	AM_PM_indicator	day_part_segment
1	12 :00 :00	12	0	0	AM	Midnight
2	12 :00 :00	12	0	0	PM	Midday
3	8 :30 :12	8	30	12	PM	Afternoon

TABLE 6 – Time

id_travel	id_date	id_time	id_traveler	id_line	id_station	id_vehicle	travel_cost
1	1	1	1	201	3	2	1
2	2	2	100	101	1	1	1.5

TABLE 7 – Travel

Pour le *data mart* de « maintenance »

id_driver	name	surname	birth_year	gender	address	nationality	contract_type	contract_start_year	salary	experience_years	tram_driver	bus_driver
1	Jeff	Lol	1985	M	3 avenue de MDR	France	CDD	2018	1700	1	<i>true</i>	<i>false</i>
2	Christy	DePelouse	1982	F	10 boulevard de non	France	CDI	2016	2100	2	<i>true</i>	<i>true</i>

TABLE 8 – Driver

id_technician	name	surname	birth_year	gender	address	nationality	contract_type	contract_start_year	salary	experience_years	specialty
1	Sammuel	Bro	1987	M	20 rue de riz	France	CDI	2017	1600	2	Mécanicien
2	Anaïs	Jeune	1983	F	12 avenue de rodez	France	CDD	2018	1400	1	Électricien

TABLE 9 – Technician

id_technical_area	address	type_maintained_vehicles	vehicles_capacity	surface	monthly_costs	equipment_level	workforce_size
1	5 boulevard de marianne	bus	50	5	1400	3	200
1	12 avenue de ludovic	tram	20	20	2000	4	400

TABLE 10 – TechnicalArea

<i>id_date</i>	<i>id_time</i>	<i>id_vehicle</i>	<i>id_driver</i>	<i>id_technician</i>	<i>id_technical_area</i>	cost	estimated_time
2	2	1	1	1	1	100	45
3	2	1	1	1	1	50	20
2	2	2	2	2	2	150	60
3	2	2	2	2	2	100	40

TABLE 11 – Maintenance