

PROJET CUBE : MY SQL



NUMÉRISATION DU SYSTÈME D'INFORMATION D'UNE AGENCE DE VOYAGE

SOMMAIRE

- PRÉSENTATION DU PROJET
- LES RÈGLES DE GESTION
- LE DICTIONNAIRE DE DONNÉES
- LE MCD ET SES DÉPENDANCES FONCTIONNELLES
- LE MLD
- LE MLD TEXTUEL
- LE SCRIPT SQL
- LE SCRIPT IMPORT DES DONNÉES
- LES PROCÉDURES
- LES REQUÊTES



PRÉSENTATION DU PROJET

NUMÉRISATION DU SYSTÈME D'INFORMATION
D'UNE AGENCE DE VOYAGE



MySQL®

LES RÈGLES DE GESTION

Les outils utilisés:

- Google Drive et ses outils
 - Looping
- MySQL et son Workbench
- Generate Data.com & Mockaroo.com
- Chat GPT pour générer des données (coordonnées)

Respect de la 3ème forme normale

LE DICTIONNAIRE DE DONNÉES

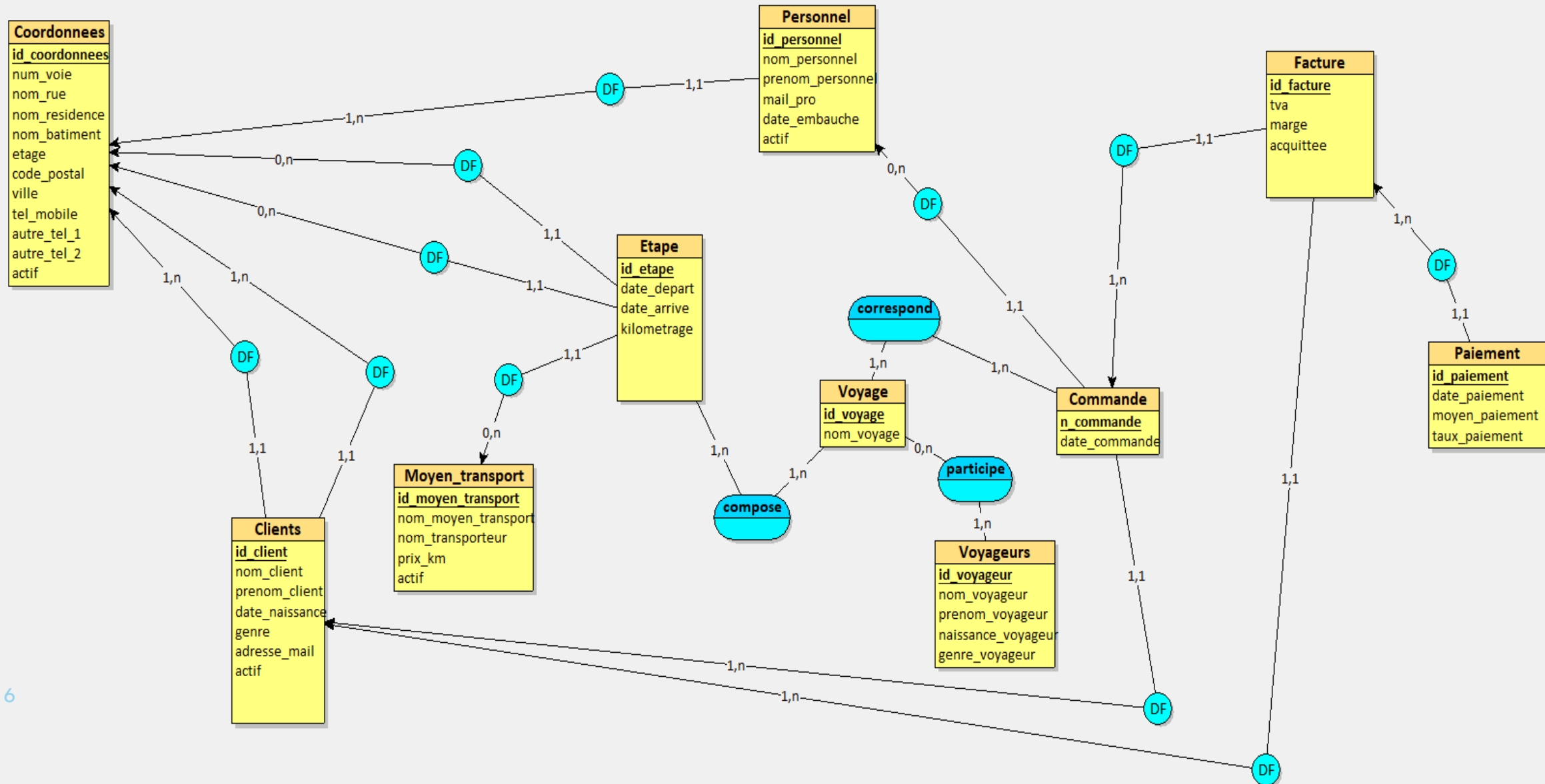
Le dictionnaire de données SQL représente le cœur de la base de données.

Il s'agit d'un ensemble de tables systèmes contenant :

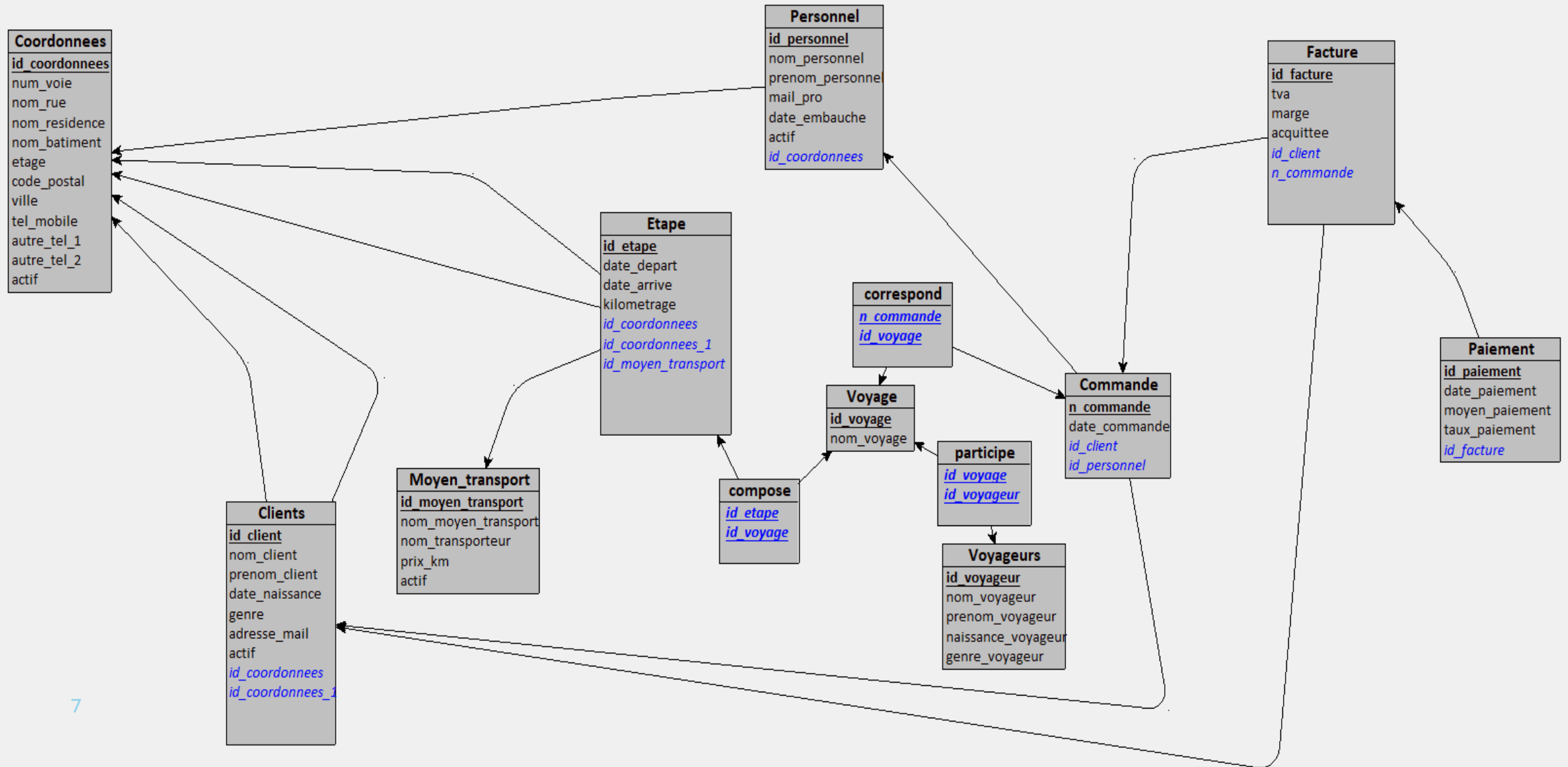
- Codes mnémotechniques
- Désignations
- Types
- Taille
- Obligatoire ou non

Codes mnémotechniques	Désignations	Type	Taille	Obligatoire
id_client	Numéro du client	N	100	O
nom_client	Nom du client	A	50	O
prenom_client	Prénom du client	A	50	O
date_naissance	Date de naissance du client	A	10	O
genre	Genre du client (H/F/N)	A	1	O
adresse_mail	Adresse mail du client	AN	100	O
actif	Savoir si le client est actif (True) ou non (False)	Booléen	1	O
id_coordonnees	Numéro identifiant de l'adresse	N	100	O
num_voie	Numéro de la voie	N	100	O
nom_rue	Nom de la rue (boulevard, avenue,...)	A	150	O
nom_residence	Nom de résidence	A	100	N
nom_batiment	Nom du bâtiment	AN	50	N
etage	Etage	N	10	N
code_postal	Code postal	N	5	O
ville	Ville	A	100	O
tel_mobile	Téléphone principal	N	20	N
autre_tel_1	Autre téléphone n°1	N	20	N
autre_tel_2	Autre téléphone n°2	N	20	N
actif	Savoir si la coordonnée est active (True) ou non (False)	Booléen	1	O
id_personnel	Numéro du salarié	N	100	O
nom_personnel	Nom du salarié	A	50	O
prenom_personnel	Prénom du salarié	A	50	O
date_embauche	Date d'embauche du salarié	N	10	O

MCD



MLD



MLD TEXTUEL

Coordonnees (id_coordonnees, num_voie, nom_rue, nom_residence, nom_batiment, etage, code_postal, ville, tel_mobile, autre_tel_1, autre_tel_2, actif)

Personnel (id_personnel, nom_personnel, prenom_personnel, mail_pro, date_embauche, actif, #id_coordonnees)

Moyen_transport (id_moyen_transport, nom_moyen_transport, nom_transporteur, prix_km, actif)

Etape (id_etape, date_depart, date_arrivee, kilometrage, #id_depart, #id_arrivee, #id_moyen_transport)

Voyage (id_voyage, nom_voyage)

Voyageurs (id_voyageur, nom_voyageur, prenom_voyageur, naissance_voyageur, genre_voyageur)

Clients (id_client, nom_client, prenom_client, date_naissance, genre, adresse_mail, actif, #id_livraison, #id_facturation)

Commande (n_commande, date_commande, #id_client, #id_personnel)

Facture (id_facture, tva, marge, acquittee, #id_client, #n_commande)

Paielement (id_paiement, date_paiement, moyen_paiement, taux_paiement, #id_facture)

compose (#id_etape, #id_voyage)

correspond (#n_commande, #id_voyage)

participe (#id_voyage, #id_voyageur)

SCRIPT SQL

1- DROP TABLE IF EXISTS :

participe;
correspond;
compose;
Paielement;
Facture;
Commande;
Clients;
Voyageurs;
Voyage;
Etape;
Moyen_transport;
Personnel;
Coordonnees;

2- CREATE TABLE :

Coordonnees;
Personnel;
Moyen_transport;
Etape;
Voyage;
Voyageurs;
Clients;
Commande;
Facture;
Paielement;
compose;
correspond;
participe;

SCRIPT IMPORT DES DONNÉES

```
1- DELETE FROM (Table);  
...  
2- ALTER TABLE (Table) AUTO_INCREMENT=1;  
...  
3- INSERT INTO (Table)  
...  
VALUES  
...
```

```
• INSERT INTO Voyage (nom_voyage)  
VALUES  
('Chat'),  
('Serpent'),  
('Mouton'),
```

```
• INSERT INTO Voyageurs (nom_voyageur, prenom_voyageur, naissance_voyageur, genre_voyageur)  
VALUES  
('Dupont', 'Marie', '1950-02-14', 'F'),  
('Martin', 'David', '1989-06-21', 'H'),  
('Leroy', 'Sophie', '2000-03-08', 'F'),  
('Petit', 'Jérôme', '2011-11-16', 'H'),
```

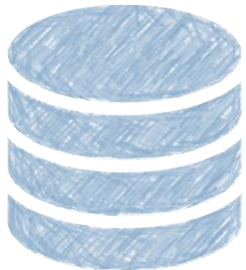
```
• INSERT INTO compose (id_etape, id_voyage)  
VALUES  
(1,1),  
(1,2),  
(2,3),  
(2,4),
```

```
• INSERT INTO Coordonnees (num_voie, nom_rue, nom_residence, nom_batiment, etage, code_postal, ville, tel_mobile, autre_t  
VALUES  
(123, 'Rue du Pont', NULL, NULL, NULL, 75000, 'Paris', '0123456789', NULL, NULL),  
(456, 'Rue de la Montagne', 'Résidence du Soleil', 'Batiment A', 6, 69000, 'Lyon', '0987654321', '0612345678', NULL),  
(789, 'Avenue de la Forêt', NULL, 'Immeuble Le Chêne', NULL, 13000, 'Marseille', '0123456789', '0612345678', '071234567  
(321, 'Rue du Jardin', 'Résidence des Fleurs', NULL, 2, 13100, 'Aix-en-Provence', '0987654321', NULL, NULL),  
(654, 'Boulevard de la Mer', NULL, 'Immeuble Le Phare', 7, 13200, 'Arles', '0123456789', '0612345678', '0712345678'),  
  
• INSERT INTO Personnel (nom_personnel, prenom_personnel, mail_pro, date_embauche, id_coordonnees)  
VALUES  
('Elrick', 'Kacy', 'kelrick0@voyage.fr', '1503697375', 1),  
('Moncaster', 'Ferd', 'fmoncaster1@voyage.fr', '1653697375', 2),  
('Blondell', 'Wynn', 'wblondell2@voyage.fr', '1603697375', 3),  
  
• INSERT INTO Moyen_transport (nom_moyen_transport, nom_transporteur, prix_km)  
VALUES  
('Avion', 'Air France', '0.10'),  
('Avion', 'Air India', '0.11'),  
('Avion', 'Air China', '0.12'),  
('Avion', 'KLM', '0.13'),  
('Train', 'SNCF', '0.16'),  
  
• INSERT INTO Etape (date_depart, date_arrivee, kilometrage, id_depart, id_arrivee, id_moyen_transport)  
VALUES  
(1641229200, 1641235000, 500, 6, 9, 7),  
(1645198400, 1645202000, 1000, 3, 25, 2),  
(1649489600, 1649495000, 856, 33, 22, 8),
```

LES PROCÉDURES

EXEMPLE : LA FACTURE

CRÉATION



- `DROP PROCEDURE IF EXISTS AjoutFacture;`
`DELIMITER //`
- `CREATE PROCEDURE AjoutFacture(` *#Ajout d'une facture*
`IN tva FLOAT,`
`IN marge FLOAT,`
`IN acquittee tinyint(1),`
`IN id_client INT,`
`IN n_commande INT`
`)`
`BEGIN`
`INSERT INTO Facture (tva, marge, acquittee, id_client, n_commande)`
`VALUES`
`(tva, marge, acquittee, id_client, n_commande);`
`END //`
`DELIMITER ;`
- `#CALL AjoutFacture (0.2, 0.1, 0, 1, 2);`
`#SELECT * FROM Facture;`

LES PROCÉDURES

EXEMPLE : LA FACTURE

MODIFICATION



```
DROP PROCEDURE IF EXISTS ModifFacture;
DELIMITER //
• CREATE PROCEDURE ModifFacture
  (IN id_fact INT,
  IN tva FLOAT,
  IN marge FLOAT,
  IN acquittee tinyint(1),
  IN id_client INT,
  IN n_commande INT)
  BEGIN
  UPDATE Facture
  SET tva=tva,
  marge=marge,
  acquittee=acquittee,
  id_client=id_client,
  n_commande=n_commande
  WHERE id_facture=id_fact;
  END //
DELIMITER ;
```

```
• #CALL ModifFacture (29,0.2, 0.15, 0, 1, 2);
#SELECT * FROM Facture;
```

MODIFICATION(DONNÉE)

```
DROP PROCEDURE IF EXISTS ModifDonneeFacture;
DELIMITER //
• CREATE PROCEDURE ModifDonneeFacture
  (IN id_fact INT,
  IN marge FLOAT)
  BEGIN
  UPDATE Facture
  SET
  marge=marge
  WHERE id_facture=id_fact;
  END //
DELIMITER ;

• #CALL ModifDonneeFacture (29, 0.17);
#SELECT * FROM Facture;
```

LES PROCÉDURES

EXEMPLE : LA FACTURE

RECHERCHE



- ```
DROP PROCEDURE IF EXISTS RechercheFacture;
DELIMITER //
```
- ```
CREATE PROCEDURE RechercheFacture
(IN id_fact INT)
BEGIN
SELECT id_facture, tva, marge, acquittee, id_client, n_commande
FROM Facture
WHERE id_facture=id_fact;
END //
```
 - ```
#CALL RechercheFacture (29);
```

# LES PROCÉDURES

EXEMPLE : LA FACTURE

## ÉDITION



```
DROP PROCEDURE IF EXISTS EditionFacture;
DELIMITER //
CREATE PROCEDURE EditionFacture
 (IN id_fact INT,
 IN marge FLOAT)
 BEGIN
 SELECT id_facture, tva, marge, acquittee, id_client, n_commande
 FROM Facture WHERE id_facture=id_fact;
 CALL ModifDonneeFacture(29, 0.18);
 SELECT id_facture, tva, marge, acquittee, id_client, n_commande
 FROM Facture WHERE id_facture=id_fact;
 END //
DELIMITER ;

#CALL EditionFacture(29, 0.18);
```



# LES PROCÉDURES

EXEMPLE : LA FACTURE

## SUPPRESSION



```
DROP PROCEDURE IF EXISTS SuppFacture;
DELIMITER //

• CREATE PROCEDURE SuppFacture
 (IN id_fact INT)
 BEGIN
 DELETE FROM Facture
 WHERE id_facture=id_fact;
 END//
DELIMITER ;

• #CALL SuppFacture (1);
 #SELECT * FROM Facture;
```

# LES PROCÉDURES

## EXEMPLE : LA FACTURE

### SUPPRESSION (CASCADE)



```
DROP PROCEDURE IF EXISTS SuppFactureCascade;
DELIMITER //

• CREATE PROCEDURE SuppFactureCascade
 (IN id_fact INT)
 BEGIN
 WHILE (Select count(*) FROM Paiement WHERE id_facture=id_fact) > 0 do
 CALL SuppPaiement((SELECT id_paiement FROM Paiement WHERE id_facture=id_fact LIMIT 1));
 END WHILE;
 DELETE FROM Facture WHERE id_facture=id_fact;
 END //
DELIMITER ;

• #CALL SuppFactureCascade (14);
 #SELECT * FROM Facture;
```



# LES REQUÊTES

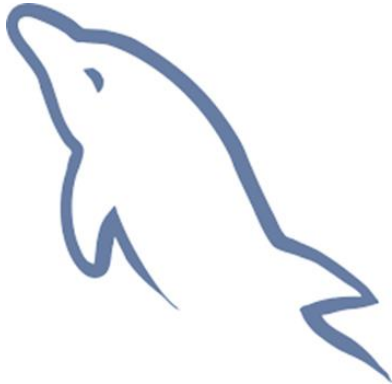
QUELQUES REQUÊTES:

#Requête 1 : Connaître le nombre de clients

```
SELECT count(*) AS Nb_Clients
FROM Clients;
```

Résultat :

|   | Nb_Clients |
|---|------------|
| ► | 21         |



#Requête 2 : Connaître le nombre de clients par villes !

```
SELECT Coordonnees.ville, count(Clients.id_client) AS Clients_par_ville
FROM Clients
INNER JOIN Coordonnees ON Coordonnees.id_coordonnees=Clients.id_facturation
GROUP BY Coordonnees.ville
ORDER BY count(Clients.id_client) DESC;
```

Résultat :

|   | ville       | Clients_par_ville |
|---|-------------|-------------------|
| ► | Vannes      | 2                 |
|   | Arras       | 2                 |
|   | Montpellier | 1                 |
|   | Rennes      | 1                 |
|   | Châteauroux | 1                 |
|   | Grenoble    | 1                 |
|   | ...         | ...               |

# LES REQUÊTES

## QUELQUES REQUÊTES:

#Requête 7 : Connaître le nombre de voyage dont les moyens de transports sont l'avion et le car.

- `DROP VIEW req7;`
- `CREATE VIEW req7 AS SELECT Moyen_transport.nom_moyen_transport, compose.id_voyage, compose.id_etape  
FROM Etape  
INNER JOIN Moyen_transport ON Etape.id_moyen_transport=Moyen_transport.id_moyen_transport  
INNER JOIN Compose ON compose.id_etape=Etape.id_etape;`
- `SELECT count(DISTINCT id_voyage) AS "Nb voyages par avion et car"  
FROM req7  
WHERE id_voyage IN  
(SELECT id_voyage FROM req7 AS T1 WHERE EXISTS  
(SELECT * FROM req7 AS T2 WHERE (T1.nom_moyen_transport = "Avion") AND  
(T2.nom_moyen_transport = "Bus") AND (T1.id_voyage=T2.id_voyage)));`

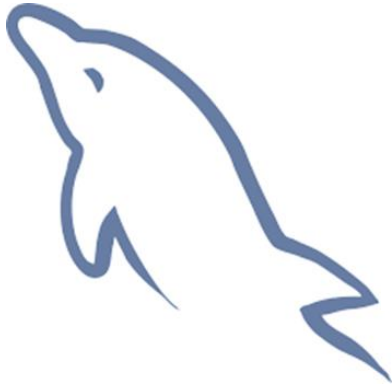
Résultat :

|   | Nb voyages par avion et car |
|---|-----------------------------|
| ► | 2                           |

# LES REQUÊTES

QUELQUES REQUÊTES:

- #Requête 9 : Connaître le moyen de transport le plus utilisé sur les 3 derniers mois
- ```
SELECT moyen_transport.nom_moyen_transport, count(moyen_transport.nom_moyen_transport)
AS "nb de fois utilisé sur les 3 derniers mois"
FROM Etape
INNER JOIN moyen_transport ON etape.id_moyen_transport=
moyen_transport.id_moyen_transport
WHERE Etape.date_depart >unix_timestamp(NOW())-7889400
GROUP BY moyen_transport.nom_moyen_transport
ORDER BY count(moyen_transport.nom_moyen_transport) DESC;
```



Résultat :

	nom_moyen_transport	nb de fois utilisé sur les 3 derniers mois
►	Bus	2
	Avion	1
	Train	1

LES REQUÊTES

QUELQUES REQUÊTES:

#Requête 11: Connaître le taux des personnes par sexe qui voyagent les 6 derniers mois

```
• SELECT genre_voyageur, count(voyageurs.id_voyageur)/  
  (SELECT count(voyageurs.id_voyageur)  
   FROM Voyageurs  
   INNER JOIN participe ON participe.id_voyageur=voyageurs.id_voyageur  
   INNER JOIN voyage ON participe.id_voyage=voyage.id_voyage  
   INNER JOIN compose ON voyage.id_voyage=compose.id_voyage  
   INNER JOIN etape ON compose.id_etape=etape.id_etape WHERE date_depart>  
   unix_timestamp(NOW())-15778800) AS taux FROM Voyageurs  
   INNER JOIN participe ON participe.id_voyageur=voyageurs.id_voyageur  
   INNER JOIN voyage ON participe.id_voyage=voyage.id_voyage  
   INNER JOIN compose ON voyage.id_voyage=compose.id_voyage  
   INNER JOIN etape ON compose.id_etape=etape.id_etape WHERE date_depart>  
   unix_timestamp(NOW())-15778800  
 GROUP BY genre_voyageur;
```

Résultat :

	genre_voyageur	taux
►	N	0.1176
	F	0.5000
	H	0.3824

**Merci de votre
attention!
Des questions?**