

SAE 3 — Livrable BD

Partie Bases de Données



Groupe G-K

RESTOUT Loïc

BASIN Léanne

ARNOULD Noa

LEMAIRE Clément

CARPENTIER Louka

LEPEUVE Maxence

I. Conception générale	3
1. Description des objets fonctionnels	3
2. MCD et dictionnaire de données	8
II. Conception détaillée	10
1. MLD, clés primaires et étrangères	10
3. Choix de la technologie	11
4. MPD	12
5. Scripts de création des tables	13

I. Conception générale

1. Description des objets fonctionnels

Description des objets/tables

USERS (Utilisateurs)

Description : Représente tous les utilisateurs de la plateforme (conducteurs et passagers). Plus précisément, un utilisateur est une entité physique inscrite sur la plateforme pour proposer ou rechercher des solutions de mobilité. Cet objet agit comme l'identité numérique centrale au sein du système : il regroupe les informations d'état civil nécessaires à l'identification, les coordonnées de contact et l'affiliation académique qui est indispensable pour garantir l'appartenance à la communauté universitaire.

Attributs clés & Spécificités techniques :

- **Identifiant unique (user_id) :** Clé primaire auto-incrémentée.
- **Informations personnelles :** Comprend first_name, last_name, phone et email (ce dernier possède une contrainte **UNIQUE** pour servir de login).
- **Informations académiques :** student_id et university (qui a pour valeur par défaut 'IUT Amiens').
- **Profil visuel :** Chemins vers profile_picture et banner_picture.
- **Statut du compte (Flags) :** Utilisation de indicateurs en VARCHAR(1) avec une contrainte **CHECK ('Y', 'N') :**
 - is_verified_flag (Défaut 'N') : Email validé ou non.
 - is_active_flag (Défaut 'Y') : Compte activé ou banni.
 - is_admin_flag (Défaut 'N') : Droits d'administration.

VEHICLES (Véhicules)

Description : Représente les véhicules appartenant aux conducteurs. Fonctionnellement, un véhicule est la ressource matérielle indispensable à la création d'une offre de covoiturage. C'est l'outil de travail du conducteur qui doit être clairement identifiable par le passager lors du rendez-vous (grâce à la marque, la couleur et surtout la plaque d'immatriculation). Cet objet porte des contraintes physiques (le nombre de places) qui limitent le nombre de réservations possibles.

Attributs clés & Spécificités techniques :

- **Informations du véhicule** : brand, model, color, license_plate et l'année (year_vehicle).
- **Capacité (seats)** : Nombre de places total, défini par défaut à **4** et ne peut être nul.
- **Statut (is_active_flag)** : Indicateur 'Y'/'N' (Défaut 'Y') pour rendre un véhicule indisponible sans le supprimer.
- **Relation** : Lié à un utilisateur via user_id. La suppression du propriétaire entraîne la suppression du véhicule (**ON DELETE CASCADE**).
-

TRIPS (Trajets)

Description : Représente les trajets proposés par les conducteurs. Un trajet matérialise une offre de service ponctuelle : c'est l'engagement d'un conducteur à partager son véhicule sur un itinéraire précis et à un horaire donné. C'est l'objet central de la recherche pour les passagers. Il ne se limite pas à un point A et un point B ; il inclut une notion de temps (date de départ, durée estimée) et une dimension économique (prix par place).

Attributs clés & Spécificités techniques :

- **Localisation** : Adresses textuelles + Coordonnées GPS précises (departure_latitude, departure_longitude, etc.) stockées en **DECIMAL(10,8)/(11,8)**.
- **Planification** : departure_datetime et estimated_duration (en minutes, avec contrainte **CHECK > 0**).
- **Disponibilité & Tarif** : available_seats et price_per_seat.
- **Statut (status)** : Chaîne de caractères avec contrainte stricte : '**active**', '**completed**', '**cancelled**' (Défaut : 'active').
- **Relations** : Lié à un conducteur (driver_id) et un véhicule (vehicle_id).

BOOKINGS (Réservations)

Description : Représente les réservations effectuées par les passagers. Une réservation est le “contrat” établi entre un passager et un conducteur pour un trajet spécifique. Elle concrétise la rencontre entre l'offre et la demande en réservant une ou plusieurs places dans le véhicule, décrémentant ainsi la disponibilité du trajet.

Attributs clés & Spécificités techniques :

- **Référence** : Lie un `passenger_id` à un `trip_id`.
- **Nombre de places (`seats_booked`)** : Par défaut **1**, ne peut être nul.
- **Prix total** : Calculé et stocké dans `total_price`.
- **Statut de la réservation (`status`)** : Contrainte stricte : '**pending**', '**confirmed**', '**cancelled**', '**completed**' (Défaut : '**pending**').
- **Statut du paiement (`payment_status`)** : Contrainte stricte : '**pending**', '**paid**', '**refunded**'.

REVIEWS (Évaluations)

Description : Représente les évaluations entre conducteurs et passagers. Une évaluation est l'outil de régulation de la plateforme. Elle intervient a posteriori pour construire la réputation des membres. Cet objet permet de quantifier la qualité du service et du comportement (via une note de 1 à 5 et des “★”) et de qualifier l'expérience humaine (via un commentaire).

Attributs clés & Spécificités techniques :

- **Note (`rating`)** : Entier avec contrainte **CHECK (BETWEEN 1 AND 5)**.
- **Commentaire textuel** : Champ comment de type TEXT.
- **Type d'évaluation** : Contrainte stricte sur type : '**driver**' ou '**passenger**'.
- **Unicité** : Contrainte **UNIQUE(booking_id, reviewer_id)** empêchant un utilisateur de noter plusieurs fois la même réservation.

MESSAGES (Messages)

Description : Représente les messages échangés concernant un trajet. Contrairement à une messagerie instantanée classique, cet objet est strictement lié à un trajet : il sert quasi exclusivement à la coordination logistique (préciser le point de rendez-vous exact, prévenir d'un retard, etc.).

Attributs clés & Spécificités techniques :

- **Contenu du message** : Champ message de type TEXT, ne peut être vide.
- **Statut de lecture (is_read_flag)** : Indicateur 'Y'/'N', défini par défaut à 'N' (Non lu) à la création.
- **Horodatage** : created_at automatique.
- **Relations** : Lié à l'expéditeur (sender_id) et au contexte du trajet (trip_id).

Règles métier

1. **Disponibilité des places** : Le nombre de places réservées ne peut pas dépasser le nombre de places disponibles
Le système doit effectuer une vérification bloquante avant chaque confirmation de réservation : il calcule la somme des places déjà réservées (seats_booked) pour le trajet et vérifie que l'ajout de la nouvelle demande ne dépasse pas la capacité totale du véhicule (available_seats ou seats).
2. **Évaluation unique** : Un utilisateur ne peut évaluer qu'une seule fois une même réservation
Elle empêche un utilisateur malveillant ou mécontent de "bombarder" un autre utilisateur avec plusieurs mauvaises notes pour faire chuter sa moyenne, ou inversement de gonfler artificiellement une note.
3. **Statut cohérent** :
 - Un trajet complété ne peut plus être modifié
 - Une fois le statut passé à 'completed', les détails deviennent des archives inaltérables pour garantir que la prestation correspond bien à ce qui a été vendu.
 - Une réservation annulée ne peut pas être confirmée
 - Une réservation 'cancelled' est une voie sans issue. Si le passager change d'avis, il doit créer une nouvelle réservation, assurant ainsi une traçabilité claire des actions.

4. **Prix cohérent** : $\text{total_price} = \text{seats_booked} \times \text{price_per_seat}$
 Cette règle impose une transparence totale sur la facturation. Le prix total d'une réservation ne doit jamais être une valeur saisie arbitrairement par l'utilisateur, mais toujours le résultat d'un calcul mathématique.
5. **Double évaluation** : Après un trajet, le conducteur peut évaluer le passager ET le passager peut évaluer le conducteur
 Le système doit permettre deux avis distincts pour une même réservation, permettant de construire deux profils de réputation séparés pour chaque utilisateur.
6. **Vérification** : Seuls les utilisateurs vérifiés peuvent proposer des trajets
 Cette règle agit comme une barrière de sécurité. Avant de pouvoir assumer la responsabilité de transporter des tiers, un conducteur doit prouver la validité de son identité numérique.
7. **Dates** : La date de départ d'un trajet doit être dans le futur lors de la création
 Il serait illogique de proposer un trajet pour une date passée. Le système doit bloquer la création de toute offre dont le "departure_datetime" est antérieur à l'instant présent.
8. **Messages** : Seuls les participants d'un trajet (conducteur et passagers ayant réservé) peuvent envoyer des messages
 Cette règle protège la vie privée des utilisateurs et évite le harcèlement. L'espace de discussion (messages) n'est pas un chat public ; c'est un canal logistique privé.

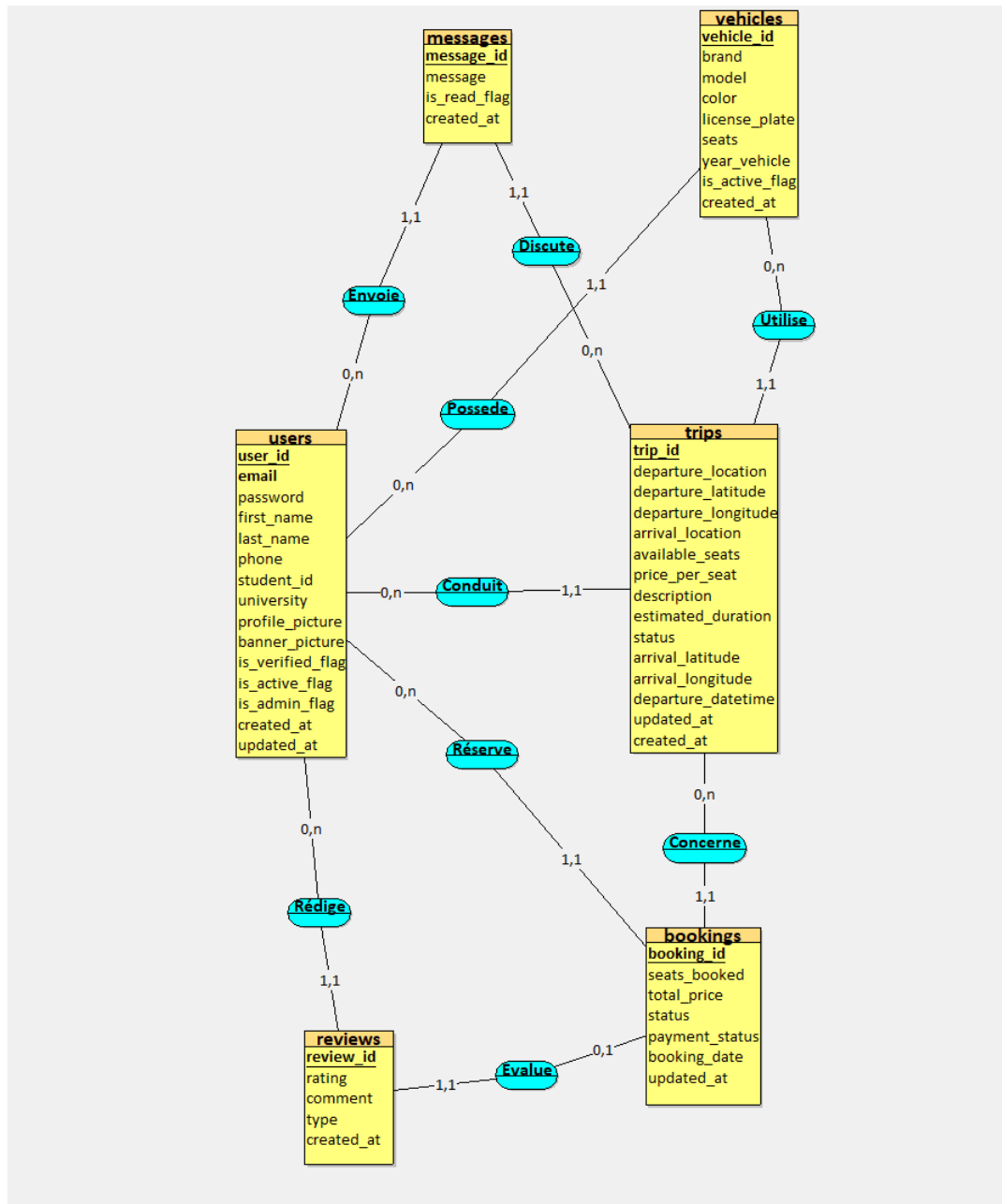
Optimisations

Index créés :

Index	Table	Colonnes	Objectif
idx_trips_status	trips	status	Optimiser les recherches de trajets par statut
idx_trips_departure_datetime	trips	departure_datetime	Optimiser les recherches de trajets par date
idx_bookings_trip_id	bookings	trip_id	Optimiser les recherches de réservations par trajet
idx_bookings_passenger_id	bookings	passenger_id	Optimiser les recherches de réservations par passager
idx_bookings_status	bookings	status	Optimiser les recherches de réservations par statut
idx_messages_trip_id	messages	trip_id	Optimiser les recherches de messages par trajet
idx_vehicles_user_id	vehicles	user_id	Optimiser les recherches de véhicules par utilisateur

2. MCD et dictionnaire de données

MCD



Dictionnaire de données

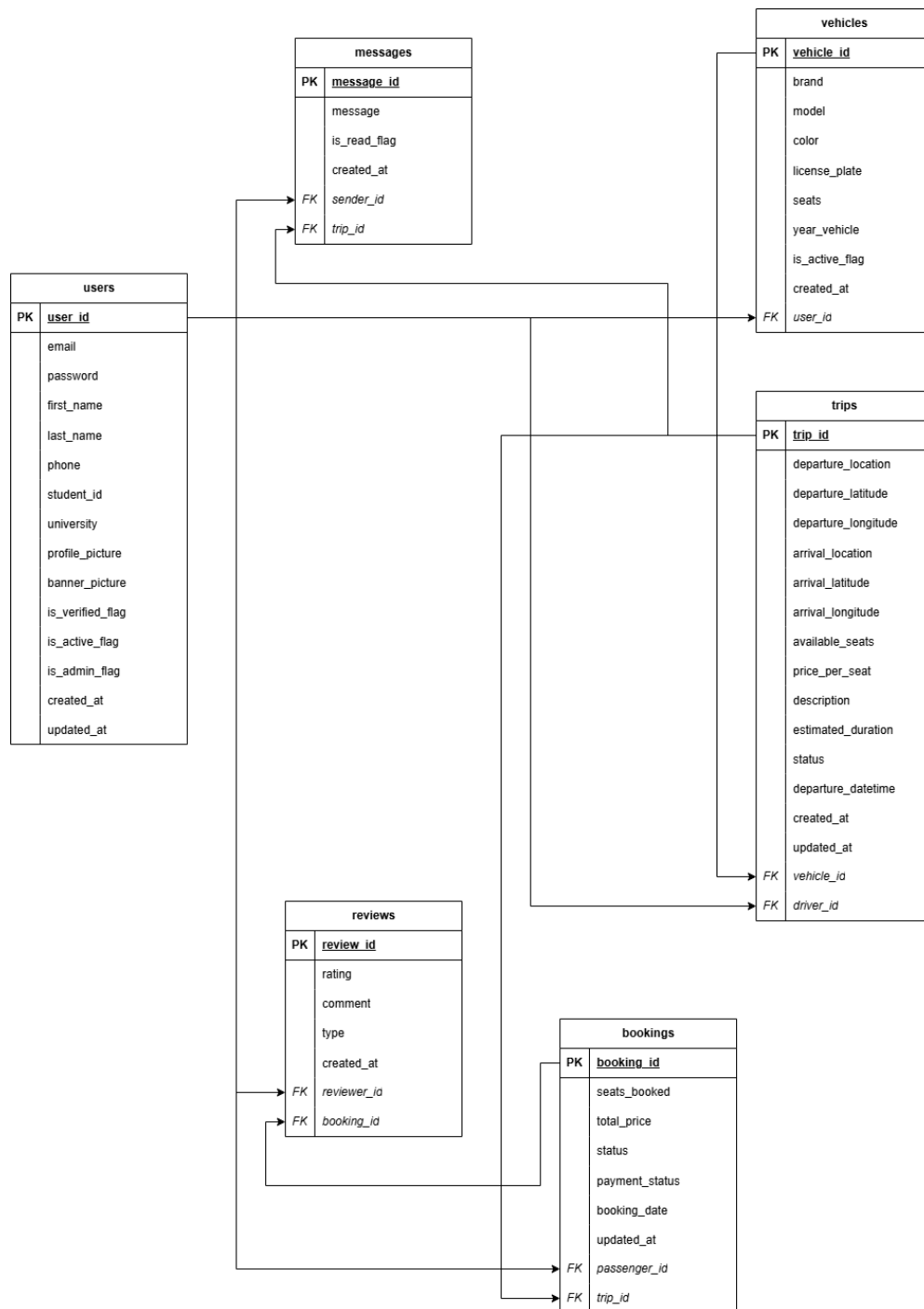
(Voir le fichier DictionnaireDonneesG-K.pdf)

Règles de gestion du dictionnaire

1. Un utilisateur peut être conducteur et/ou passager
2. Un conducteur peut posséder plusieurs véhicules
3. Un trajet est associé à un seul conducteur et un seul véhicule (optionnel)
4. Un passager peut réserver plusieurs places sur un même trajet (seats_booked)
5. Un passager ne peut pas réserver deux fois le même trajet (contrainte UNIQUE)
6. Une réservation peut avoir plusieurs statuts : pending, confirmed, cancelled, completed
7. Une évaluation est liée à une réservation terminée
8. Chaque participant (conducteur/passager) peut évaluer l'autre après le trajet
9. Les messages sont liés à un trajet spécifique
10. La suppression d'un utilisateur entraîne la suppression de ses trajets, véhicules, réservations (CASCADE)
11. La suppression d'un véhicule ne supprime pas les trajets associés (SET NULL)

II. Conception détaillée

1. MLD, clés primaires et étrangères

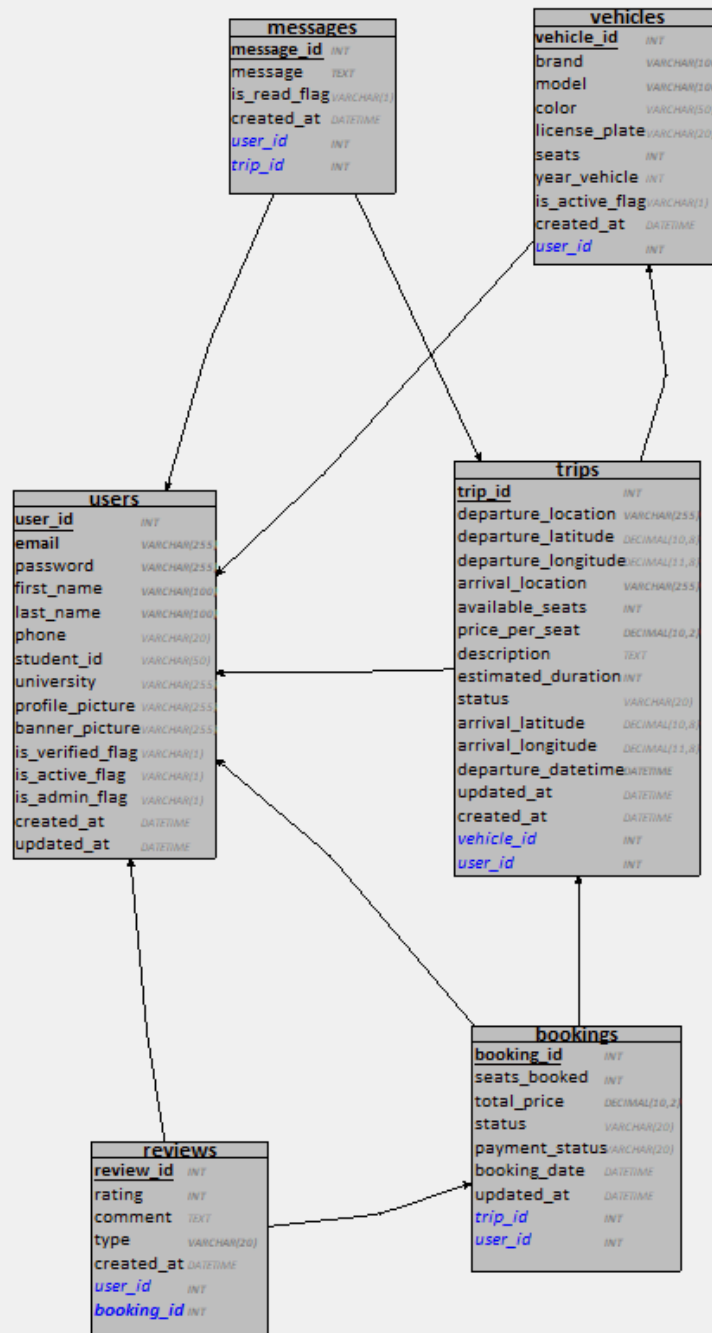


3. Choix de la technologie

Nous avons choisi SQLite pour ce projet car c'est la solution la plus simple à partager. Comme la base de données est un simple fichier inclus dans notre dossier, le projet marchera tout de suite sur n'importe quel ordinateur, sans avoir besoin d'installer un serveur compliqué. Cela garantit que notre démonstration fonctionnera toujours, même dans plusieurs mois.

Cependant, nous savons que pour lancer le site "pour de vrai" avec beaucoup d'utilisateurs, il faudrait passer sur PostgreSQL. En effet, SQLite a un défaut : il ne gère pas bien quand plusieurs personnes essaient d'écrire des données en même temps (comme réserver un trajet au même moment). Cela ralentirait le site. PostgreSQL, lui, est conçu pour supporter des centaines de connexions simultanées sans bloquer, ce qui est indispensable pour une vraie application publique.

4. MPD



Note technique : Sur le schéma généré par le logiciel de modélisation (Looping), les clés étrangères portent le nom par défaut de la clé primaire (user_id). Cependant, dans notre script SQL de production, nous avons renommé ces clés (driver_id, passanger_id, sender_id) afin d'apporter une meilleure clarté sémantique aux rôles de chacun.

5. Scripts de création des tables

(Voir le fichier ScriptSQLG-K.sql)