

Rapport Base de données

Augustin Fourmy - Elazar Cohen

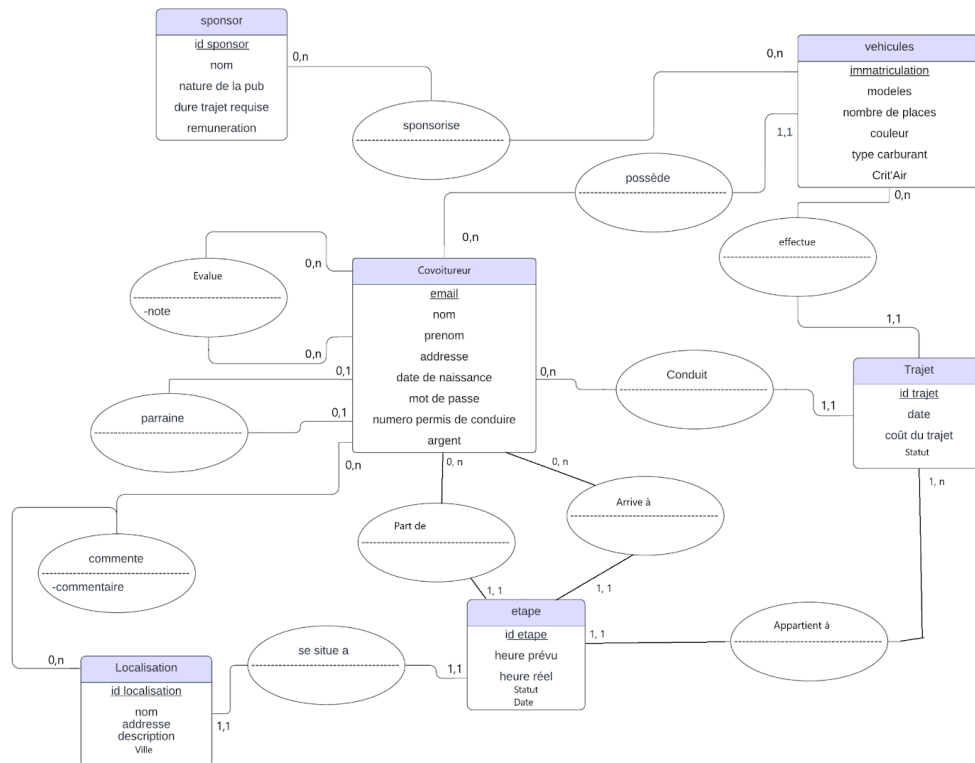
15 décembre 2024

Table des matières

1	Rendu 1	2
2	Rendu 2	4
2.1	Traduction du Schéma Entité-Association	5
2.2	Implémentation	5
2.2.1	Schéma de la base et Justification des choix	5
2.2.2	Contraintes non Garantie	7
2.3	Demande supplémentaire	8
3	Rendu 3	9
3.1	Ajout par rapport au deuxième rapport	10
3.1.1	Dans le dump	10
3.1.2	Génération du dump	10
3.2	Explication du site	10
3.2.1	Menu	10
3.2.2	Accueil	10
3.2.3	Connexion	10
3.2.4	Profil	10
3.2.5	Trajet	11
3.2.6	Trajet en cours	11
3.2.7	Déconnexion	11
3.2.8	Statistiques	11
3.3	Explication du code	11
3.3.1	La clé secrète	11
3.3.2	L'accueil	12
3.3.3	Éviter de réécrire la base de la template	12
3.3.4	Petits icons	13

Chapitre 1

Rendu 1



Changements effectuer:

Nous avons changé la cardinalité de "se situe à" qui relie Localisation et Étape de su coté de localisation de 1,n à 1,1

Nous avons ajouté l'attribut ville à localisation, l'attribut date à Etape

Nous avons renommé certaine association, attribut, ou entité.

Nous avons supprimé les associations "Propose étape", "commence a" et "arrive a". A la place, nous avons créé deux associations qui relient Covoitureur et Étape : "Part de" et "Arrive à". Qui indique l'emplacement de départ et d'arrivée d'un covoitureur. Nous avons également ajouté l'association "Appartient à" entre Etape et Trajet. Cela nous sert à savoir quelle étape concerne quel trajet.

Grâce à ce changement, on peut désormais s'assurer que chaque covoitureur propose deux et seulement deux étapes par trajet. Et les points de départ et d'arrivée du trajet ne sont plus que les points de départ et d'arrivée du conducteur. Cependant, un inconvénient dès cette modélisation est que l'on ne peut pas s'assurer que l'utilisateur qui propose le point de départ est le même que celui qui propose le point d'arrivée, on ne peut également pas restreindre que le conducteur d'un trajet proposé obligatoirement 1 étape. Cela serait réglé plus haut dans le projet dans le projet.

Chapitre 2

Rendu 2

2.1 Traduction du Schéma Entité-Association

En traduisant le schéma entité-association on obtient les tables suivantes :

- Covoitureur(email, nom, prenom, adresse, date_de_naissance, mdp, num_permis, argent, parrain)
- Localisation(id_localisation, nom, adresse, description, ville)
- Etape(id_etape, heure_prevue, heure_reel, statut, date, trajet, est_depart_de, est_arriver_de, localisation)
- Trajet(id_trajet, date, cout, statut, vehicule, conducteur, nb_km, duree)
- Vehicules(immatriculation, modele, nb_place, couleur, carburant, crit_air, proprietaire)
- Sponsor(id_sponsor, nom, nature, duree_requise, remuneration)
- Sponsorise(id_sponsor, immatriculation)
- Commente(email, id_localisation, commentaire)
- Evalue(source, cible, note)

On a les clefs étrangères :

- Sponsorise(id_sponsor) fais référence à Sponsor(id_sponsor)
- Sponsorise(immatriculation) fais référence à Vehicules(immatriculation)
- Commente(email) fais référence à Covoitureur(email)
- Commente(id_localisation) fais référence à Localisation(id_localisation)
- Evalue(source) fais référence à Covoitureur(email)
- Evalue(cible) fais référence à Covoitureur(email)
- Covoitureur(parrain) fais référence à Covoitureur(email)
- Vehicule(proprietaire) fais référence à Covoitureur(email)
- Trajet(vehicule) fais référence à Vehicules(immatriculation)
- Trajet(conducteur) fais référence à Covoitureur(email)
- Etape(trajet) fais référence à Trajet(id_trajet)
- Etape(est_depart_de) fais référence à Covoitureur(email)
- Etape(est_arriver_de) fais référence à Covoitureur(email)
- Etape(localisation) fais référence à Localisation(id_localisation)

2.2 Implémentation

2.2.1 Schéma de la base et Justification des choix

Covoitureur		
Nom	Type	Contrainte
email	varchar(100)	PRIMARY KEY
nom	varchar(50)	NOT NULL
prenom	varchar(50)	NOT NULL
adresse	varchar(100)	NOT NULL
date_de_naissance	date	NOT NULL
mdp	char(64)	NOT NULL
num_permis	char(12)	
argent	numeric(11, 2)	default 0
parrain	varchar(100)	REFERENCES Covoitureur(email) ON DELETE SET NULL
CHECK (email <> parrain)		

Localisation		
Nom	Type	Contrainte
id_localisation	serial	PRIMARY KEY
nom	varchar(50)	NOT NULL
adresse	varchar(100)	NOT NULL
description	varchar(500)	
ville	varchar(64)	NOT NULL

Etape		
Nom	Type	Contrainte
id_etape	serial	PRIMARY KEY
heure_prevue	time	NOT NULL
heure_reel	time	
status	bool	default false
date	date	NOT NULL
trajet	int	NOT NULL REFERENCES trajet(id_trajet) ON DELETE CASCADE
est_depart_de	varchar(100)	REFERENCES Covoitureur(email) ON DELETE CASCADE
est_arriver_de	varchar(100)	REFERENCES Covoitureur(email) ON DELETE CASCADE
localisation	int	NOT NULL REFERENCES Localisation(id_localisation) ON DELETE CASCADE
CHECK((est_depart_de IS NULL AND est_arriver_de IS NOT NULL) OR (est_depart_de IS NOT NULL AND est_arriver_de IS NULL))		

Vehicules		
Nom	Type	Contrainte
immatriculation	char(7)	PRIMARY KEY
modele	varchar(55)	NOT NULL
nb_places	int	NOT NULL
couleur	varchar(60)	NOT NULL
carburant	varchar(50)	NOT NULL
crit_air	char(1)	NOT NULL CHECK(crit_air IN ('E', '1', '2', '3', '4', '5'))
proprietaire	varchar(100)	NOT NULL REFERENCES Covoitureur(email) ON DELETE CASCADE ON UPDATE CASCADE

Trajet		
Nom	Type	Contrainte
id_trajet	serial	PRIMARY KEY
date	date	NOT NULL
cout	numeric(9, 2)	NOT NULL
statut	bool	default false
vehicule	char(7)	NOT NULL REFERENCES Vehicules(immatriculation) ON DELETE CASCADE
localisation	int	NOT NULL REFERENCES Localisation(id_localisation) ON DELETE CASCADE
nb_km	int	CHECK(nb_km >= 0)
duree	interval	

Sponsor		
Nom	Type	Contrainte
id_sponsor	serial	PRIMARY KEY
nom	varchar(75)	NOT NULL
nature	varchar(500)	NOT NULL
duree_requise	interval	NOT NULL
remuneration	numeric(11, 2)	NOT NULL

Sponsorise		
Nom	Type	Contrainte
id_sponsor	int	REFERENCES Sponsor(id_sponsor)
immatriculation	char(7)	REFERENCES Vehicules(immatriculation)
PRIMARY KEY (id_sponsor, immatriculation)		

Commente		
Nom	Type	Contrainte
email	varchar(100)	REFERENCES Covoitureur(email) ON DELETE CASCADE ON UPDATE CASCADE
id_localisation	int	REFERENCES Localisation(id_localisation)
commentaire	varchar(500)	NOT NULL
PRIMARY KEY (email, id_localisation)		

Evaluer		
Nom	Type	Contrainte
source	varchar(100)	REFERENCES Covoitureur(email) ON DELETE CASCADE ON UPDATE CASCADE
cible	varchar(100)	REFERENCES Covoitureur(email) ON DELETE CASCADE ON UPDATE CASCADE
note	int	NOT NULL
PRIMARY KEY (source, cible)		
CHECK (source <> cible)		

Nous avons fait certains choix sur les types :

- Les mots de passe sont stockés comme des chaînes de caractères d'exactly 64 caractères car ils seront hachés avant que leur forme hexadécimale soit stockée.
- Nous séparons la date et les heures pour ne pas permettre de renseigner le décalage de deux jours
- Nous avons ajouté un attribut longueur pour pouvoir réaliser les demandes supplémentaires

2.2.2 Contraintes non Garantie

Il y a des contraintes que nous n'avons pas pu implémenter dans le schéma :

- Il nous est impossible de vérifier que l'e-mail de l'utilisateur est valide
- Il en est de même pour les plaques d'immatriculations
- Nous ne pouvons pas nous assurer que les covoitureurs déposent exactement deux étapes pour un trajet

2.3 Demande supplémentaire

Nous avons créer des vues pour répondre aux contraintes supplémentaire qui nous ont été demander

- Obtenir les 10 premiers points de rendez-vous les plus f

```
CREATE VIEW TopLocation AS (  
    SELECT adresse, ville, COUNT(id_etape) FROM etape  
    NATURAL JOIN Localisation  
    GROUPE BY adresse, ville  
    ORDER BY COUNT(id_etape) DESC  
    LIMIT 10  
);
```

- Le nombre de kilomètre parcourus pour chaque sponsor

```
CREATE VIEW SponsorKm AS (  
    SELECT id_sponsor, nom, SUM(nb_km) FROM Sponsor  
    NATURAL JOIN Sponsorise NATURAL JOIN Vehicules  
    JOIN Trajet ON vehicules = immatriculation  
    GROUP BY id_sponsor, nom  
);
```

Chapitre 3

Rendu 3

3.1 Ajout par rapport au deuxième rapport

3.1.1 Dans le dump

1. dans la table covoitureur nous avons mis la taille des mot de passe a 64 or pour hacher le mot de passe la taille du mot de passe est de 60.
2. dans la table trajet, nous avons enlevé une relation redondante qui était l'association conduit car nous pouvons y accéder avec un *JOIN* entre le trajet et les véhicules dans le schema entité association.
3. nous avons enlever la date dans étape, en effet c'est la même information que la date dans trajet, elle est donc inutile.

3.1.2 Génération du dump

Dans le rapport 2 nous avons générer quelque données avec une IA. Mais cette fois nous avons créer un fichier python qui permet de générer un dump aléatoire avec autant de données que souhaité. vous pourrez trouver le fichier pour générer ce dump dans un dossier Dumpmaker et il faut lancer le fichier dumpmaker.py. (la génération des covoitureurs avec le hashage des mot de passe est un peu long)

3.2 Explication du site

3.2.1 Menu

- Si vous n'êtes pas connecte vous avez accès que a l'accueil, a connexion et les statistiques.
- Si vous êtes connecte en revanche vous pourrez apercevoir l'accueil, votre profil, les trajets, les trajets en cours, se déconnecter et les statistiques .

3.2.2 Accueil

en arrivant sur le site vous pourrez trouver une belle page d'accueil ou vous pourrez a votre guise regarder les trajets existants. vous trouverez aussi un menu avec une page de connexion et les statistiques du site.

3.2.3 Connexion

vous pouvez maintenant vous connecter avec votre compte si vous en avez un, ou dans le cas contraire vous inscrire en rentrant votre email, nom,prénom,etc. une fois inscrits vous pouvez vous connecter a votre compte qui vous dirige vers l'accueil. Vous avez maintenant accès a l'ensemble du menu pour naviguer dans le site.

3.2.4 Profil

Comme dit plus haut si vous êtes connecte vous pouvez aller sur votre profil. Le profil est divisé en 4 parties :

1. Les données de l'utilisateur : elles permettent de voir les données comme le nom, le prénom,l'argent,si on a le permis pouvoir l'enlever ou le rajouter si il nous a été retiré par les policiers ou le FBI. On peut aussi voir son parrain si on en a un et les filleuls.
2. La deuxième partie est pour voir les véhicules en sélectionnant la plaque d'immatriculation et on peut aussi ajouter un véhicule. Des qu'on voit un véhicule :
 - on peut voir les données de son véhicule avec le nombre de place ou encore la couleur
 - on peut aussi modifier son véhicule en modifiant la plaque ou la couleur, etc .
 - la troisième partie du véhicules est pour sponsoriser son véhicule en choisissant le sponsor parmi les différents sponsor
3. la troisième partie du profil est la liste des différents trajets déjà effectué.
4. la dernière partie c'est les opérations sur votre compte en raison des couts des trajets ou si vous avez conduit certains trajets vous recevez de l'argent.



Comme vous pouvez le voir sur l'image, j'ai utilisé un peu de JavaScript pour pouvoir naviguer parmi les trajets passés.

TABLE 3.1 – Navigation avec JavaScript

3.2.5 Trajet

Sur le menu il y'a une page de trajet, elle permet de proposer un nouveau trajet avec une étape et d'arrivée. on peut aussi tout les trajets avec les information nécessaire ou si on veut en savoir plus on peut cliquer sur plus d'information pour voir précisément le trajet et pouvoir rajouter une étape au trajet. On peut aussi noter le conducteur ou encore allez dans plus d'info sur l'étape et écrire un commentaire de la localisation.

Si un utilisateur propose une nouvelle étape le covoitureur depuis son compte peut voir l'étape qui a été proposé, l'accepter a sa guise et finalisée le trajet. Il peut aussi le réouvrir pour permettre a un utilisateur de proposer une nouvelle étape.

un covoitureur (caractérisé par son permis de conduire) peut créer un nouveau trajet en renseignant une etape de départ et d'arriver avec le cout du trajet, le nombre de kilomètre que fais le trajet le temps espéré etc.

3.2.6 Trajet en cours

La page trajet en cours permet a l'utilisateur de voir ses trajets et de pouvoir trier pour voir seulement les trajets en cours ou les trajets passées, etc.

3.2.7 Déconnexion

ce n'est pas une page mais on peut se déconnecter ce qui permet de supprimer de la session l'utilisateur et le redirige vers l'accueil sans toutes les options.

3.2.8 Statistiques

les statistiques permettent de voir les lieux les 10 lieux les plus visités du site et le nombre de km parcourus au total et avec des voitures sponsorisés

3.3 Explication du code

nous allons expliquer quelques bout de code important

3.3.1 La clé secrète

Pour générer la clé secrète nous avons utilisé le module secrets de python.

```
secrets.token_urlsafe(100)
```

Ce qui nous a permis de générer une clé secrète de 100 caractères.

3.3.2 L'accueil

nous avons écrit une route supplémentaire pour arriver directement sur l'accueil au lancement du site

```
@app.route("/")
def main()->None:
    return redirect(url_for("accueil"))
```

3.3.3 Éviter de réécrire la base de la template

pour éviter de réécrire dans chaque template la base du html nous avons créé une page *base.html* qui nous sert de base à toutes les templates.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
    initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="../static
    /base.css">
{%block css%} {%endblock%}
<title>{%block title%} {%endblock%} </title>
</head>
<header>
<nav>
<ul>
<li><a href="/accueil">Accueil</a></li>
{%block menu%}
<li><a href="/connexion">Connexion</a></li>
{%endblock%}
<li><a href="/statistiques">Statistique</a></li>

</ul>
</nav>
</header>
<body>
<div id = "content">
{%block content%} {%endblock%}

</div>
</body>
</html>
```

En faisant ça cela nous permet d'avoir une template sans avoir à réécrire toute la base html et surtout si nous voulons modifier le menu ou le css du menu nous avons pas besoin de le modifier dans toutes les pages. Nous avons juste à écrire comme ci dessous.

```

{% extends "base.html" %}
{% block css%} <link rel="stylesheet" type="text/css"
    href="../static/connexion.css"> {%endblock%}
{% block menu%}
<li><a href="/connexion">Connexion</a></li>
{%endblock%}

{% block content%}
{%endblock%}

```

3.3.4 Petits icons

Dans le profil nous avons ajouté des petits icons pour rendre le visuel plus sympathique.