

Carnel, Guide d'utilisation :

Administrateur

1) Administrateur de Base de Données

La Base de Données utilisée lors de la Saé23 se présente sous la forme suivante :

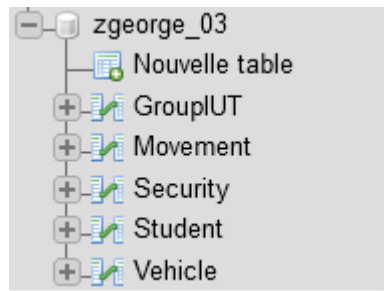


Figure 1 : Toutes les tables dans notre base de données

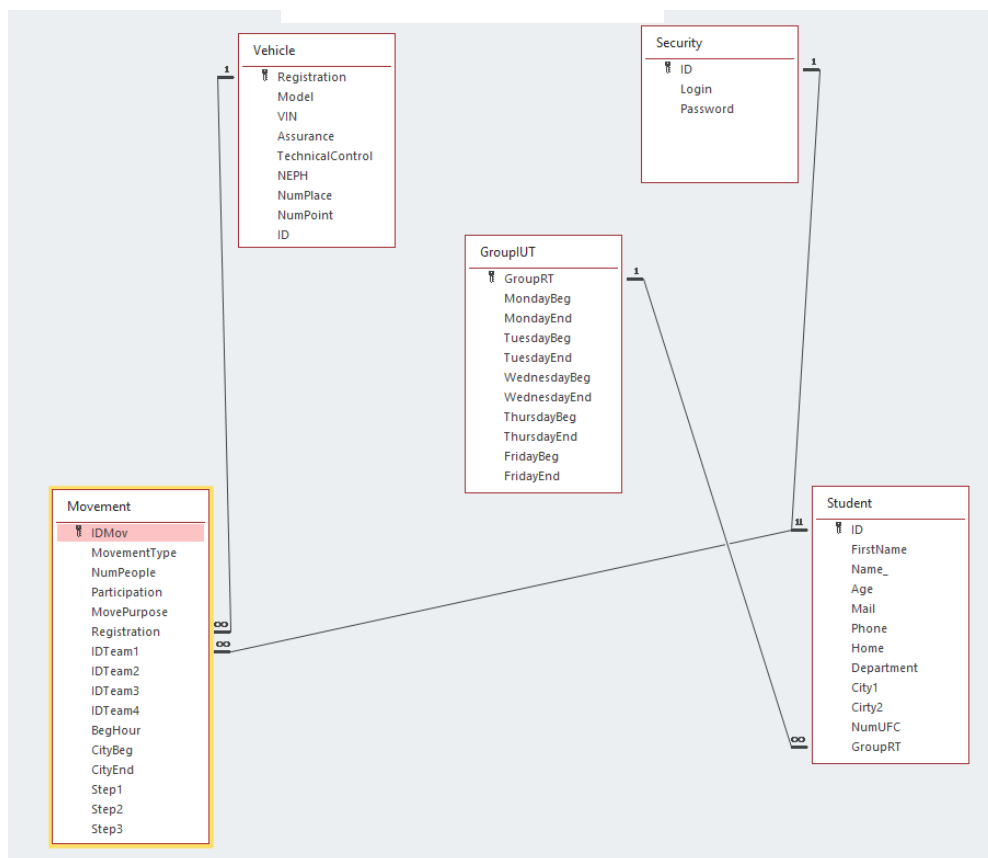



Figure 2 : Base de données sous la forme de MPD faite sous Microsoft Accès

L'administrateur se connecte avec les identifiants suivants :




Identifiant : zgeorge

Mot de passe : ZA12*\$za

Table GroupIUT :

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1	GroupRT 	varchar(15)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	2	MondayBeg	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	3	MondayEnd	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	4	TuesdayBeg	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	5	TuesdayEnd	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	6	WednesdayBeg	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	7	WednesdayEnd	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	8	ThursdayBeg	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	9	ThursdayEnd	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	10	FridayBeg	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	11	FridayEnd	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		

La table GroupIUT contient les différents groupes qui existent auxquels les utilisateurs peuvent être associés. La clé primaire de cette table est la clé GroupRT de type varchar, cette clé est utilisée dans la table Student pour définir l'appartenance d'un étudiant à un groupe. On retrouve également les horaires de début et de fin de chaque jour de la semaine, également de type varchar.

	GroupRT	MondayBeg	MondayEnd	TuesdayBeg	TuesdayEnd	WednesdayBeg	WednesdayEnd	ThursdayBeg	ThursdayEnd	FridayBeg	FridayEnd
<input type="checkbox"/> Éditer  Copier  Supprimer	admin	0	0	0	0	0	0	0	0	0	0
<input type="checkbox"/> Éditer  Copier  Supprimer	RT1-LK2	09h30	18h00	08h00	18h00	08h00	18h00	08h00	18h00	08h00	18h00

A l'heure actuelle, il existe 5 groupes :

- RT1-GB1
- RT1-GB2
- RT1-LK1
- RT1-LK2
- Admin

Sur l'image ci-dessus, on peut voir à quoi ressemble les données associées à cette table. Il faut également savoir que cette table est de préférence rempli manuellement sur phpmyadmin, cependant nous pouvons utiliser le code du fichier json (add.json) ainsi que le script exécutable (json.php) afin de remplir cette table.

Table Student :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1 ID	int(3)			Non	Aucun(e)		AUTO_INCREMENT
<input type="checkbox"/>	2 FirstName	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	3 Name	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	4 Age	int(3)			Non	Aucun(e)		
<input type="checkbox"/>	5 Mail	varchar(50)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	6 Phone	varchar(10)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	7 Home	varchar(50)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	8 Department	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	9 City1	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	10 City2	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	11 NumUFC	varchar(8)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	12 GroupRT	varchar(7)	utf8mb4_general_ci		Non	Aucun(e)		

La table Student permet d'enregistrer les élèves dans la Base de Données dans la Base de Données. Un élève est défini par un ID unique et auto incrémental. On peut aussi renseigner de nombreuses informations le concernant comme son Groupe Réseaux et Télécommunications (GroupRT) ou son domicile. La clé primaire de cette table est utilisée dans les autres tables pour faire le lien entre un élève et les informations le concernant. Les variables sont pratiquement toutes de type varchar pour simplifier leur traitement.

	ID	FirstName	Name	Age	Mail	Phone	Home	Department	City1	City2	NumUFC	GroupRT
<input type="checkbox"/>	1	Admin	Admin	0	admin@carnel.com	0000000000	0	0	0	0	0	admin
<input type="checkbox"/>	31	Thomas	Raynaud	19	traynau2@carnel.com	0781230626	5 rue de la piscine	68250	Rouffach	Montbéliard	142537	RT1-LK2

Cette dernière table est de préférence remplie à l'aide du fichier json et de son script exécutable.

Table Security :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1 ID	int(3)			Non	Aucun(e)		
<input type="checkbox"/>	2 Login	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	3 Password	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	4 ID_Security	int(11)			Non	Aucun(e)		AUTO_INCREMENT

La table Security permet aux utilisateurs de se connecter au site grâce à un login unique et mot de passe. ID_Security permet de différencier deux utilisateurs au cas où ils auraient des logins très similaires et illisibles pour un humain. Le premier ID est récupéré grâce à la clé ID de la table Student.

				ID	Login	Password	ID_Security
<input type="checkbox"/>		Éditer		Copier		Supprimer	1 admin thomas 1
<input type="checkbox"/>		Éditer		Copier		Supprimer	31 traynau2 SAE23 4

Cette table est aussi remplie avec le fichier json, le lien avec la clé étrangère ID se fait alors plus simplement.

Table Vehicle :

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1	Registration	varchar(10)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	2	Model	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	3	VIN	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	4	Assurance	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	5	TechnicalControl	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	6	NEPH	varchar(12)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	7	NumPlace	varchar(2)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	8	NumPoint	varchar(2)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	9	ID	int(3)			Non	Aucun(e)		

La table Vehicle permet de définir les véhicules et toutes les informations les concernant. Un véhicule est défini de manière unique par son immatriculation (Registration). On renseigne des informations comme le nombre de places (NumPlace) qui seront utiles sur le site web. On trouve aussi l'ID de l'étudiant à qui appartient le véhicule.

		Registration	Model	VIN	Assurance	TechnicalControl	NEPH	NumPlace	NumPoint	ID			
		Éditer		Supprimer	AP-RT6-P5	206	547392624	TGFEY67Z	12 Janvier 2020	556787534565	2	1	30

Cette dernière peut aussi être remplie avec le fichier json et son exécutable, si un utilisateur ne possède simplement pas de voiture. Il faut dans ce cas commenter la partie du code dans le fichier json.php.

Table Movement :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1 IDMov	int(3)			Non	Aucun(e)		AUTO_INCREMENT
<input type="checkbox"/>	2 MovementType	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	3 NumPeople	int(2)			Non	Aucun(e)		
<input type="checkbox"/>	4 Participation	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	5 MovePurpose	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	6 Registration	varchar(9)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	7 IDTeam1	int(3)			Non	Aucun(e)		
<input type="checkbox"/>	8 IDTeam2	int(3)			Non	Aucun(e)		
<input type="checkbox"/>	9 IDTeam3	int(3)			Non	Aucun(e)		
<input type="checkbox"/>	10 IDTeam4	int(3)			Non	Aucun(e)		
<input type="checkbox"/>	11 BegHour	varchar(5)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	12 CityBeg	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	13 CityEnd	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	14 Step1	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	15 Step2	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		
<input type="checkbox"/>	16 Step3	varchar(30)	utf8mb4_general_ci		Non	Aucun(e)		

La table Movement permet de définir les différents trajets que vont effectuer un groupe de personnes. On utilise IDMov comme clé primaire, cette clé est de type int et auto incrémentale.

On renseigne le nombre de places que l'on a pour effectuer un trajet (NumPeople), et avec qui on va l'effectuer (IDTeam1, IDTeam2...). On définit également les villes de départ, d'arrivée, et éventuellement les étapes. Cette table relie les tables Student et Vehicule dans lesquelles on va chercher l'immatriculation de la voiture ou l'id d'un participant par exemple.

← T →	IDMov	MovementType	NumPeople	Participation	MovePurpose	Registration	IDTeam1	IDTeam2	IDTeam3	IDTeam4	BegHour	CityBeg	CityEnd	Step1	Step2	Step3
<input type="checkbox"/> Éditer Copier Supprimer	8	voiture	4	2€	IUT	AP-RT6-P5	2	3	4	31	14h30	Montbéliard	Montbéliard	0	0	0

Cette table peut aussi être remplie avec le fichier json et son exécutable. Tout autant que manuellement.

Bonne utilisation du fichier Json et de son exécutable :

Le code étant commenté, il est assez simple à prendre en main. Cependant certaines règles doivent être respecté quant à sa bonne utilisation.

Le fichier contenant les informations sous la forme de Json porte le nom « add.json ». Afin d'ajouter un utilisateur, il faut commencer par l'utilisateur 0, puis le 1, puis le 2 etc. Cette règle doit être respecté afin de faciliter le traitement par le script exécutable nommé « json.php ».

```
{
  "0": [
    {
      "Informations": {
        "FirstName": "Thomas",
        "Name": "Raynaud",
        "Age": "19",
        "Mail": "traynau2@carnel.com",
        "Phone": "0781230626",
        "NumUFC": "142537",

```

Figure 3 : add.json , encadré en rouge la déclaration du premier utilisateur

Dans ce cas on fait la déclaration du premier utilisateur, par la suite il suffit de suivre le plan du document afin d'ajouter des informations de ce dernier.

Dans le cas du second utilisateur :

```
    }
  ],
  "1": [
    {

```

Figure 4 : Déclaration du second utilisateur

Afin d'éviter de commettre des erreurs, il est préférable de garder la même forme et le même ordre présent sous un utilisateur : Information, Data, Vehicule et déplacement. Il faut ainsi garder les mêmes noms de variables :

```
"MovementType": "Voiture",
"Participation": "2$",
"MovePurpose": "Cours",
"CityBeg": "Rouffach",
"CityEnd": "Montébliard",
```

Figure 5 : Nom de variable à gauche et à droite valeur de la variable

Vis-à-vis de l'exécutable « json.php », il faut si nous ne souhaitons pas renseigner une partie commenter la requête, par exemple :

```

/si on veut ajouter les groupes avec le json, il faut décommenter cette requête
/*$groupeiut = substr($groupeiut,0,-2);
$groupeiutkey = substr($groupeiutkey,0,-2);
//echo "<br>" . $groupeiut;
//echo "<br>" . $groupeiutkey;
$sql5= "INSERT INTO GroupIUT ($groupeiutkey) VALUES ($groupeiut)";
//echo "<br>" . "<br>".$sql5."<br><br>";
if (mysqli_multi_query($conn, $sql5)) {
| echo "New records created successfully";
} else {
| echo "Error: " . $sql5 . "<br>" . mysqli_error($conn);
}*/

```

Figure 6 : add.php, si on ne veut pas ajouter de groupe avec le fichier json.

2) Administrateur web/du site

Le site se présente sous la forme d'une page de connexion qui permet d'accéder aux autres pages du site. Si un utilisateur n'est pas connecté et essaie d'accéder à une page, il sera redirigé vers la page de connexion.

L'ID de l'administrateur est 1, et le login et le mot de passe peuvent être ajustés directement dans la base de données.

Lors de la connexion à la page, des requêtes SQL vont vérifier si le login entré existe et si le mot de passe choisi est bien le bon, si c'est le cas on peut se connecter au site. On a ensuite la possibilité de voir les covoiturages auxquels on participe, on peut également accéder à une page permettant de visionner tous les covoiturages et affiner ses recherches avec des filtres. Enfin on peut accéder à une page de déconnexion.

Un traitement est réalisé afin de rediriger l'utilisateur sur des pages différentes en fonction de son ID. Si celui-ci correspond à celui de l'administrateur, alors il est redirigé vers des pages avec des options supplémentaires. Par exemple la page permettant de voir les covoiturages enregistrés affichera toutes les valeurs même si elles n'ont pas de lien avec l'utilisateur administrateur. L'administrateur pourra aussi supprimer des covoiturages en cliquant sur la ligne correspondante :

Vous êtes : admin. Vous avez donc des droits supplémentaires, visionner tous les covoiturages ainsi que de les supprimer.

Type de transport	Type de participation	Ville de départ	Ville d'arrivée	Heure de départ	Participant1	Participant2	Participant3	Participant4	Supprimer
Voiture	2\$	Rouffach	Montébliard	12h30	admin admin	Thomas Raynaud	X	X	<input type="button" value="X"/>
Voiture	2\$	Rouffach	Montébliard	12h30	X	X	X	X	<input type="button" value="X"/>

Un utilisateur lambda ne verra que les covoiturages dans lesquels il est impliqué et ne pourra pas les supprimer, mais il pourra se désinscrire de certains trajets :

Voici les covoiturages auxquels vous êtes inscrits :

Mode de déplacement	But du déplacement	Nombre de personne(s)	Participation	Personne 1	Personne 2	Personne 3	Personne 4	Se désinscrire
Voiture	Voiture	Cours	2	2\$	admin admin	Thomas Raynaud	X	X

Plusieurs parties pourraient être améliorées, comme donner l'heure et le jour d'un covoiturage, que l'utilisateur en ajoute un et aussi lier ces derniers aux emplois du temps qui sont déjà présent dans la base de données. La base de données présente permet alors l'évolutivité de l'application web ainsi qu'une perspective d'amélioration future.