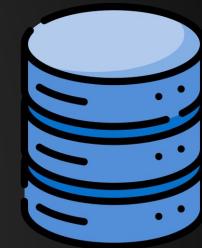
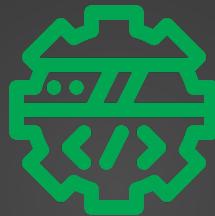


# Cours de DB SQL

Nicolas Triaud  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

# Objectif du cours

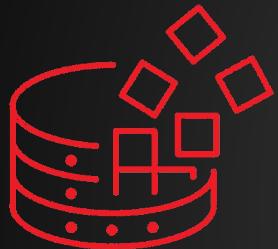


Télécharger un environnement de développement comme XAMPP / MAMP / WAMP

Setup l'environnement

Créer une DB MySQL

# Objectif du cours



Comprendre les  
types de données



010101110101



Requêtes SQL CRUD  
& Avancées

# 1. XAMPP / MAMP / WAMP

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

# XAMPP / MAMP / WAMP

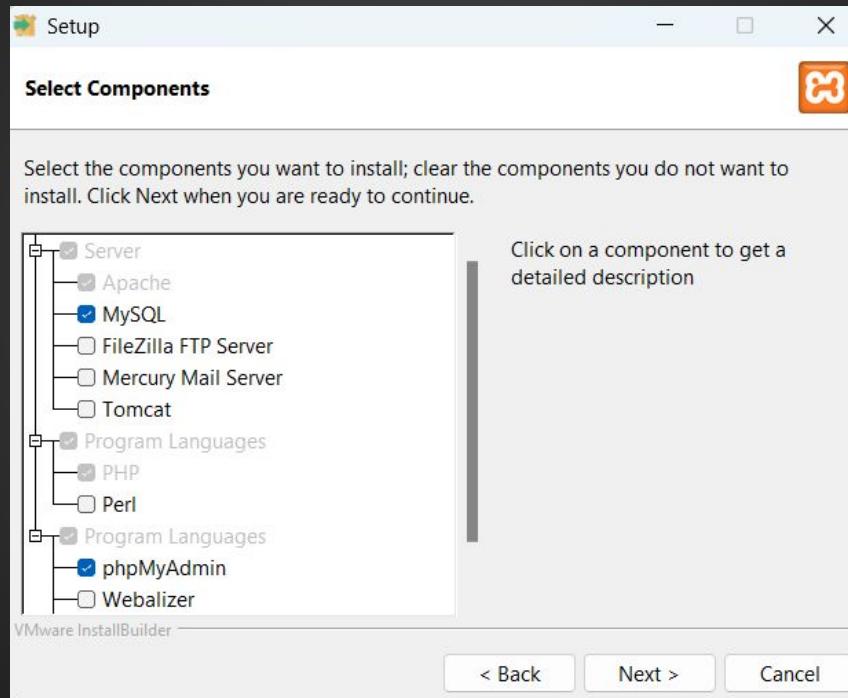
- **Qu'est-ce que c'est ?**

- Paquet de logiciels
- **APACHE** -> Serveur web local
- **MySQL** -> DataBase SGBDR (Système de Gestion de Bases de Données Relationnelles)
- Le logiciel permet aussi de se connecter (via Apache) à une interface graphique du nom de **phpmyadmin**

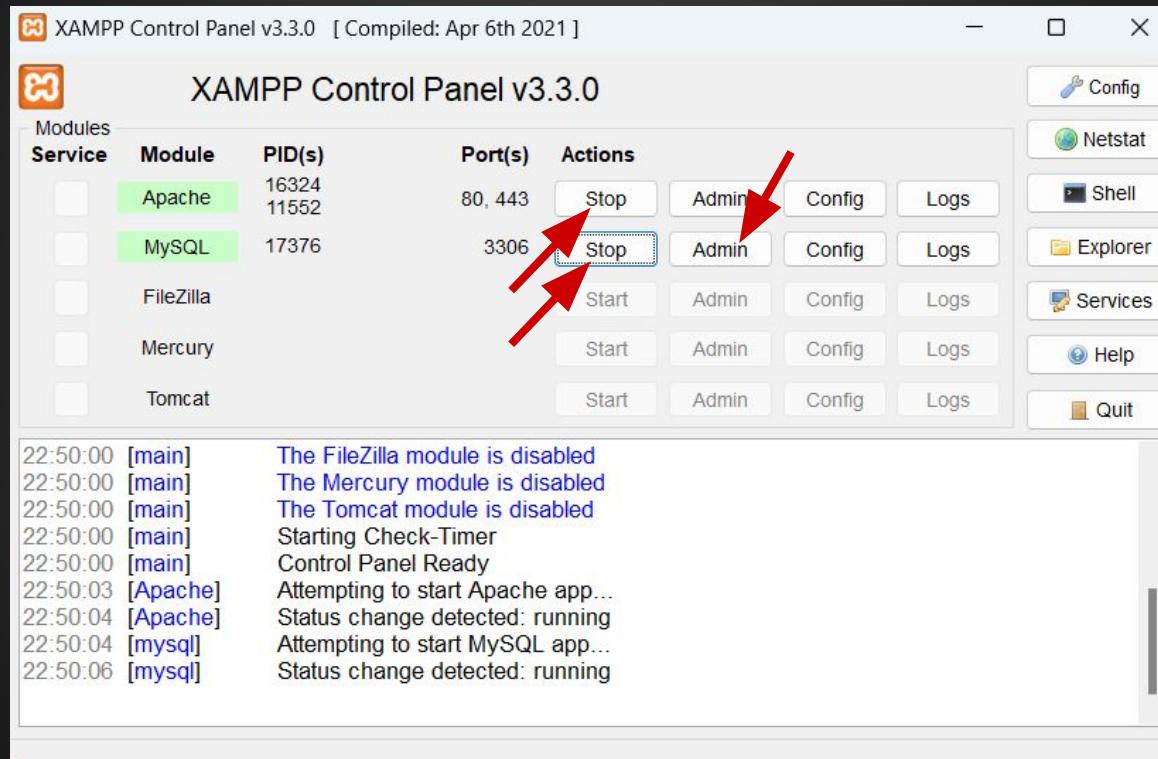
## XAMPP / MAMP / WAMP

*<https://www.apachefriends.org/fr/index.html>*

# XAMPP / MAMP / WAMP



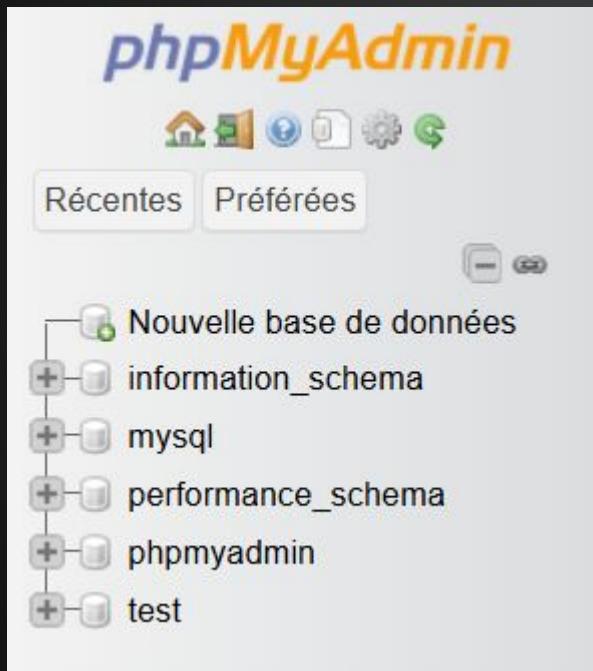
# XAMPP / MAMP / WAMP



# 2. Creation d'une DB

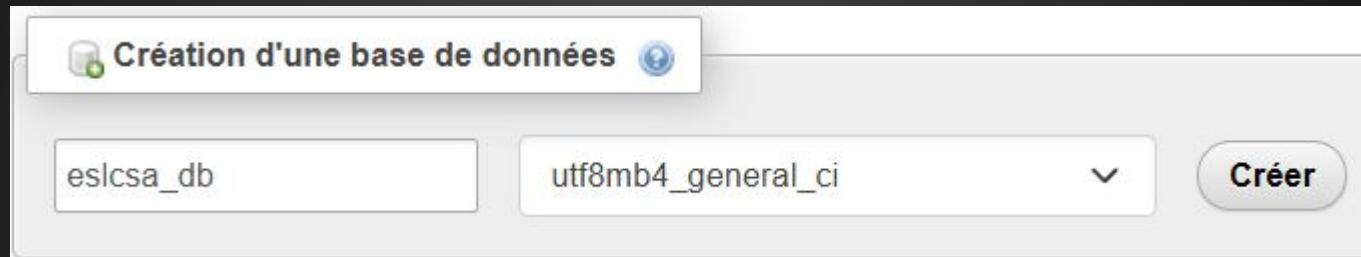
Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

## Création d'une DB



- A gauche de l'UI, les DB sont listées
- Au-dessus, vous pouvez créer une nouvelle DB

## Création d'une DB



- Indiquez le nom de votre DB
- Décider de l'encodage. Vous pouvez laisser celui par défaut

# Création d'une DB

Nom de table: movies Ajouter 1 colonne(s) Exécuter

Structure

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index	A_I	Commentaires
id	INT		Aucun(e)			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
name	VARCHAR	255	Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>	
description	TEXT		NULL			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>	
country	CHAR	50	NULL			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>	

# Création d'une DB

- ***Types de données en SQL***

- Taille s'indique en **N** ou du **tiny/small/medium/int/big**
- Pour les nombres, **INT / FLOAT / DECIMAL**
- Pour les petits textes, **CHAR / VARCHAR**
- Pour les longs textes, **TEXT**
- Côté options récurrentes :
  - **NULL**, indique l'absence de donnée
  - **A\_I (Auto Increment)**, indique un comportement automatique pour la colonne

# Création d'une DB

- ***Type de données relatifs au SGBDR***
  - **Primary Key**, contrainte imposée à la DB pour identifier, de manière unique, une row via la valeur d'une colonne. Un index est automatiquement créé sur cette colonne.
  - **Foreign Key**, fait référence à une clé primaire d'une autre table. Ce qui lie les données entre-elles.

# 3. Le Binaire

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

# Le Binaire

- **Fonctionnement du binaire**

- La plus petite unité du binaire est le **bit**. C'est dans les bits que l'on va insérer une valeur binaire.
- Par convention, un bit est stocké dans un **octet**. Un octet est un rassemblement de 8 bits afin de former une information à 256 possibilités.



# Le Binaire

## - Calculer au-dessus du 256

- A chaque octet, on ajoute 256 possibilités.
- Pour savoir combien on peut calculer en fonction du nombre d'octet (n) on utilise  $2^{8n}$
- Si on calcule sur une plage négative & positive, c'est simplement divisé par 2
- 256 possibilités / 65 536 possibilités / 16 777 216 possibilités

32 768 16 384 8 192 4 096 2 048 1 024 512 256

128 64 32 16 8 4 2 1

0 1 0 1 0 0 1 1

0 1 0 1 0 0 1 1

# Le Binaire

## Comment stocker du texte en binaire

- On utilise une représentation d'une valeur en fonction de celle du binaire
- Dans le contexte de la table ASCII, c'est un standard utilisé dans l'encodage de caractère dans l'informatique, voici la correspondance des valeurs.

## ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1100000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1100001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1100100	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1100101	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1101000	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1101010	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1101100	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1101110	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1110000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1110001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1110100	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1110111	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111000	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111001	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111100	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	,	87	57	1010111	127	W					
40	28	101000	50	(	88	58	1011000	130	X					
41	29	101001	51	)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	\					
44	2C	101100	54	,	92	5C	1011100	134	/					
45	2D	101101	55	-	93	5D	1011101	135	^					
46	2E	101110	56	.	94	5E	1011110	136	_					
47	2F	101111	57	/	95	5F	1011111	137	-					

# 4. CRUD

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

# CRUD

- **C**reate. *Insert Into*. Insère une donnée dans une table
- **R**ead. *Select*. Opération pour lire les données d'une table
- **U**pdate. *Update*. Modifie des données existantes
- **D**elete. *Delete*. Supprime des données existantes

## CRUD - Exemples de Create

SELECT \*

SELECT

INSERT

UPDATE

DELETE

Effacer

Format

```
1 INSERT INTO `movies` SET
2   `name` = 'Gladiator',
3   `description` = 'Un romain énervé qui va se venger',
4   `country` = 'US';
```

```
1 INSERT INTO `movies`(`name`, `description`, `country`) VALUES ('Les Choristes', 'Des gosses qui chantent avec un prof super cool', 'Fr')
```

```
1 INSERT INTO `movies` (`name`, `description`, `country`) VALUES
2   ('Gladiator', 'Un romain énervé qui se venge', 'US'),
3   ('V pour Vendetta', 'Film stylé qui a fait devenir tout le monde anarchiste', 'UK');
```

## CRUD - Exemples du Read

```
1 SELECT *
2 FROM movies
3 WHERE name = 'V pour Vendetta';
```

```
1 SELECT country, description
2 FROM movies
3 WHERE name = 'V pour Vendetta';
```

- **Select** est la requête la plus sûre et la plus utilisée
  - Cette requête peut permettre de faire des tests avant une update
  - Elle est la requête la plus utilisée dans les DBs et doit au maximum être **optimisée** via les colonnes sélectionnées, et les conditions exprimées

## CRUD - Exemples d'Update

```
1 UPDATE movies  
2 SET country = 'Fr'  
3 WHERE name = 'Gladiator';
```

```
1 UPDATE movies  
2 SET country = 'US'  
3     description = 'Un film fait par Ridley Scott'  
4 WHERE name = 'Gladiator';
```

- **UPDATE** apporte une plus grande complexité
  - La syntaxe pour modifier les colonnes
  - L'apport de la conditionnalité afin de **cibler** les lignes à mettre à jour

## CRUD - Exemples du Delete

```
1 DELETE  
2 FROM movies  
3 WHERE name = 'Diplomatie';
```

- **Delete** est la requête la plus dangereuse et la moins utilisée
  - Cette requête se base sur la même logique qu'un select
  - Elle a cependant la possibilité de **tout détruire**

# 5. Structure de DB

## - Table de liaison

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

# Structure DB



- Pour les relations entre DB
  - “one-to-one”, “one-to-many”, fonctionne avec une Foreign Key “classique”. Juste une liaison simple entre une table et une autre
  - Pour le “many-to-many”. Il nous faut une table de liaison plus ou moins détaillées. Elle sert à lister les relations entre deux à X tables

# 6. Modifier une table existante

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

## Modifier une table existante

```
1 ALTER TABLE movies  
2 ADD people_capacity INT;
```

```
1 ALTER TABLE movies  
2 MODIFY people_capacity SMALLINT;
```

- **ALTER TABLE** permet de sélectionner la table à modifier dans la structure
- **ADD / MODIFY**, permettent respectivement d'ajouter/modifier une colonne. Il faut ensuite préciser le type et/ou les paramètres qui changent (NULL DEFAULT “default”...)
- **DROP COLUMN** supprimer tout simplement la colonne choisie

## Modifier une table existante

```
1 | DROP TABLE movies;
```

- **DROP TABLE** permet de supprimer la table

# 7. Import / Export

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

## Importer/Exporter une table



- Dans le cadre des UI (phpmyadmin, DBeaver etc...) il est très facile d'importer/exporter des tables
- Pour phpmyadmin, il faut tout simplement aller dans la section correspondante
- Pour exporter, il suffira de nommer votre export et choisir l'extension

# Importer/Exporter une table

Fichier à importer :

Le fichier peut être compressé (gzip, bzip2) ou non.  
Le nom du fichier compressé doit se terminer par **.[format].[compression]**. Exemple : **.sql.zip**

Parcourir les fichiers : (Taille maximale : 40Mio)

Choisir un fichier    movies.sql

Il est également possible de glisser-déposer un fichier sur n'importe quelle page.

Jeu de caractères du fichier :

utf-8

- Pour import il faut tout simplement sélectionner le fichier SQL
- Il est possible d'importer d'autres extensions, mais il y a d'autres paramètres à entrer afin que le tout soit bien détecté

# 8. Requêtes SQL avancées

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

# Requêtes SQL avancées

- JOIN permet de joindre dans le contexte de la requête une ou plusieurs tables. Il est toutefois préférable la jointure via un ON afin de mieux sélectionner les données des tables annexes

```
1 select g.first_name, r.type
2 from guests g
3 JOIN rooms r
4 ON g.room_number = r.room_number;
```

- LIKE permet de cibler des mots ou des passages de mots dans des chaînes de caractères. L'utilisation du % permet de dire “peu importe” avant ou après un passage précis

```
1 select activity
2 from activity_log
3 where activity LIKE '%kitchen%';
```

# Requêtes SQL avancées

- Des fonctions comme SUM, AVG, COUNT sont directement disponible en SQL et permettent de respectivement additionner, moyenner, compter la donnée reçue dans une colonne

```
1 select count(*)
2 from guests;
```

- Il y a aussi la possibilité, pour des requêtes plus complexes, de faire des SELECT imbriqués. C'est-à-dire que le WHERE regardera le résultat retourné par une autre requête SELECT

```
1 SELECT *
2 from employees
3 where salary > (SELECT AVG(salary) from employees);
```

# 9. Exercice

[sql-course-nine.vercel.app/cours\\_sql.pdf](#)

[sql-course-nine.vercel.app/murder\\_party.sql](#)

Nicolas Triaud  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

## **Exercice murder party**

- 1. Listez les invités
- 2. Listez les invités ayant quitté l'hôtel après le 26 octobre
- 3. Listez les invitées dont le nom de famille commençant par "T"
- 4. Afficher pour chaque invité le type et le prix de sa chambre.
- 5. Quels invités se trouvaient au restaurant entre 20h00 et 20h30 le 24 octobre ?
- 6. Quels invités s'y sont disputés ?
- 7. Donner le numéro de chambre et l'étage de Benjamin Stone.
- 8. Combien d'invités se sont enregistrés avant le 24 octobre ?
- 9. Quel est le prix moyen d'une suite ?
- 10. Quel est le revenu total généré par les paiements des invités ?

## **Exercice murder party**

- 11. Passer le niveau de suspicion de Clara Johnson à 0
- 12. Augmentez de 1 le niveau de suspicion pour tous ceux dont le niveau est déjà  $\geq 3$
- 13. Trouver les invités ayant payé plus que la moyenne des paiements
- 14. Donnez le total des paiements par étage
- 15. Trouvez le nom de l'invité ayant effectué le paiement le plus élevé
- 16. Calculez le salaire moyen par shift.
- 17. Listez les invités occupant une chambre de type “suite”
- 18. Qui se trouvait dans la cuisine à 21h00 le 24 octobre ?
- 19. Calculez le total des niveaux de suspicion par type de suspect
- 20. Identifiez la personne au plus haut niveau de suspicion

# 10. Exercice 2

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau

## **Exercice structure DB**

- 1. Suivi des dépenses et revenus mensuels pour personnes physiques
  - Il faut pouvoir enregistrer des utilisateurs
  - Associer un ou des comptes à un utilisateur
  - Associer des revenus à un compte. Ces revenus auront une catégorie et un libellé
  - Un suivi des transactions financières
  - Table de régulation des montants autorisés par catégorie de compte bancaire par mois.

## **Exercice structure DB**

- 2. Suivi des dépenses avancées
  - Ajout des personnes morales (entreprise / association etc..)
  - Une personne morale peut avoir un ou plusieurs département afin de cibler là où les coûts/revenus ont eu lieu
  - Il doit être possible de tracer les facturations. Personnes impliquées, montant, devise, date etc..
  - Une facturation peut être validée ou non-validée. Il faut pouvoir indiquer qui a, ou non, validé la facturation

# 11. Exercice CC

Nicolas Triau  
Mail : [nicolas.triau@yahoo.fr](mailto:nicolas.triau@yahoo.fr)  
Discord : gillesleveau