



INF3710 –Bases de données

Automne 2020

TP No. 1

Groupe 4

1947025 – Cassy, Charles

1898357 – Dimitry Fumtim, Kamga

Soumis à : Philippe Maisonneuve

Mardi 18 Septembre 2020

I. Création d'une base de données

1. Créez la base de données nommée « BD_TP1 ».

Fait.

2. Accédez à la BD déjà créé.

Fait.

II. Création des tables

3. Créez les 4 tables (Departement, Trajet, Camion, Chauffeur) en exécutant les commandes CREATE TABLE Et en justifiant l'ordre de création

Les tables ont été créés dans l'ordre suivant : DEPARTEMENT →CHAUFFEUR →CAMION →TRAJET

La table DEPARTEMENT a été créé la première car elle est indépendante des autres tables c'est-à-dire elle ne dépend d'aucune table. La table CHAUFFEUR a été créé par la suite car, par la ligne « FOREIGN KEY (idEmployeur) REFERENCES DEPARTEMENT(idDepartement) », on voit qu'elle dépend de DEPARTEMENT. La table CAMION a été créé après CHAUFFEUR dont elle est dépendante « FOREIGN KEY (idChauffeur) REFERENCES CHAUFFEUR(idChauffeur) ». La table TRAJET est la dernière table créé car elle dépend de la table CAMION « FOREIGN KEY (idCamion) REFERENCES CAMION(vin) ».

La table TRAJET n'aurait pas pu être créée avant CAMION car étant enfant de cette dernière, les fonctionnalités de CAMION doivent être déjà créées. De même pour CAMION qui ne peut être créé avant CHAUFFEUR, car CHAUFFEUR doit exister pour que la table CAMION soit créée et DEPARTEMENT doit aussi être mis en place avant CHAUFFEUR.

Affichage des tables créées :

DEPARTEMENT :

BD_TP1

- Casts
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Schemas (1)
 - public
 - Collations
 - Domains (1)
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables (4)
 - camion
 - chauffeur
 - departement**
 - trajet

Query Editor Query History Scratch Pad

```
1 SELECT * FROM public.departement
2 ORDER BY iddepartement ASC
```

Data Output Explain Messages Notifications

nom	iddepartement
character varying (20)	[PK] integer

CHAUFFEUR :

Databases (2)

- BD_TP1
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Schemas (1)
 - public
 - Collations
 - Domains (1)
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables (4)
 - camion
 - chauffeur**
 - departement
 - trajet

Query Editor Query History Scratch Pad

```
1 SELECT * FROM public.chauffeur
2 ORDER BY idchauffeur ASC
```

Data Output Explain Messages Notifications

idchauffeur	nomcomplet	sexe	datedenaissance	idemployeur
[PK] character varying (20)	character varying (30)	character (1)	date	integer

CAMION :

Databases (2)

- BD_TP1
 - Casts
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Schemas (1)
 - public
 - Collations
 - Domains (1)
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - Sequences
 - Tables (4)
 - camion**
 - chauffeur
 - departement
 - trajet

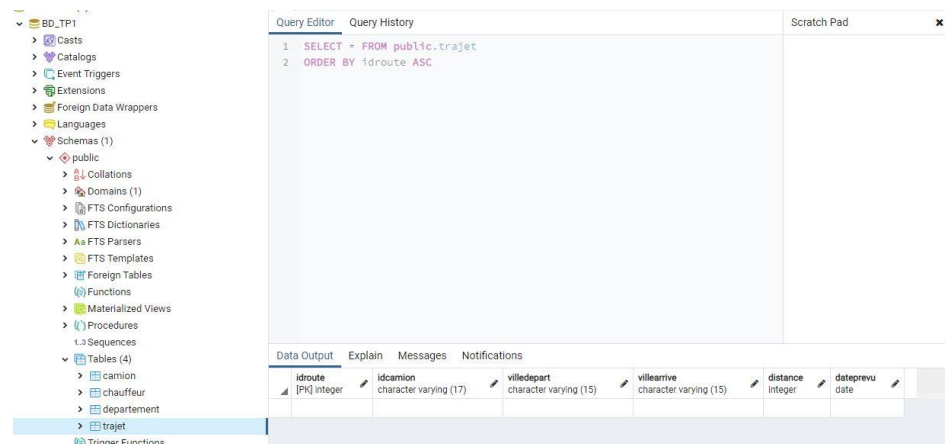
Query Editor Query History Scratch Pad

```
1 SELECT * FROM public.camion
2 ORDER BY vin ASC
```

Data Output Explain Messages Notifications

vin	idchauffeur	marque	annee de production
[PK] character varying (17)	character varying (5)	character varying (30)	integer

TRAJET :



4. Si vous essayez de créer une deuxième fois la table « Camion» avec la commande. Que se passe-t-il? Quel est la cause à votre avis? Quelle commande manque-t-il?

Si on essaie de créer une deuxième fois la table Camion avec la commande donnée, un message d'erreur s'affiche « ERROR: relation "camion" already exists SQL state: 42P07 ». Cette erreur est causée parce que la table CAMION existe déjà. La commande manquante est **CREATE TABLE IF NOT EXISTS CAMION**.

5. Expérimentez la commande SQL : SELECT * FROM CAMION; A quoi sert cette commande ?

La commande SELECT * FROM CAMION affiche la table CAMION et son contenu.

III. Peuplement des tables


6. En se basant toujours sur la base de données «BD-TP1-populate », peuplez les tables dans l'ordre en utilisant la requête Insert INTO

Fait.

7. Affichez le contenu de chaque table

Les tables affichent selon les images suivantes :

DEPARTEMENT :



BD_TP1/postgres@PostgreSQL 12

Query Editor

Query History

1

SELECT * FROM DEPARTEMENT

2

Data Output

Explain

Messages

Notifications

	nom character varying (20)	iddepartement [PK] integer
1	Long Trajets	0
2	Court Trajets	1

CHAUFFEUR :

BD_TP1/postgres@PostgreSQL 12

Query Editor

Query History

1

SELECT * FROM CHAUFFEUR

Data Output

Explain

Messages

Notifications

	idchauffeur [PK] character varying (20)	nomcomplet character varying (30)	sexe character (1)	datedenaissance date	idemployeur integer
1	XAW32	Charles Lebeau	M	1968-05-16	0
2	XAW21	Henriette Garceau	F	1980-03-26	0
3	ZAQ87	Liam Charlevoix	M	1983-09-19	1
4	ZAQ92	Belange Leningard	F	1969-07-12	1
5	WOP96	Charles Fumtim	M	1999-12-17	1

CAMION :

BD_TP1/postgres@PostgreSQL 12

Query Editor

Query History

1

SELECT * FROM CAMION

Data Output

Explain

Messages

Notifications

	vin [PK] character varying (17)	idchauffeur character varying (5)	marque character varying (30)	annee de production integer
1	2G1WL52M8X9253039	XAW32	Mercedes	2010
2	SALAK2D46BA548114	XAW21	IVECO	2008
3	1FTRW08L91KC11319	ZAQ87	MAN	2016
4	JN8A55MT0BW145107	ZAQ92	TESLA	2004

BD_TP1/postgres@PostgreSQL 12

Query Editor

Query History

Scratch Pad

1

SELECT * FROM TRAJET








2

Data Output

Explain

Messages

Notifications

 idroute [PK] integer	 idcamion character varying (17)	 villeddepart character varying (15)	 villearrive character varying (15)	 distance integer	 dateprevu date	
1	0	2G1WL52M8X9253039	Toronto	Montreal	541	2020-09-10
2	1	2G1WL52M8X9253039	Montreal	New York	595	2020-10-05
3	2	SALAK2D46BA548114	Quebec	Miami	2872	2020-08-11
4	3	SALAK2D46BA548114	Detroit	Edmonton	1917	2020-12-20
5	4	1FTRW08L91KC11319	Toronto	Hamilton	68	2020-11-24
6	5	1FTRW08L91KC11319	Long Island	New York	51	2021-01-17

TRAJET :

8. Ajoutez correctement à la table CHAUFFEUR une ligne avec les données de votre choix et fournir la requête utilisée. Puis, affichez le contenu de la table.

Nous avons ajouté la requête suivante :

INSERT INTO CHAUFFEUR VALUES ('WOP96', 'Charles Fumtim', 'M', TO_DATE('17-12-1999', 'DD:MM:YYYY'), 1);

BD_TP1/postgres@PostgreSQL 12		
Query Editor		Scratch Pad
<pre>1 INSERT INTO CHAUFFEUR VALUES ('WOP96', 'Charles Fumtim', 'M', TO_DATE('17-12-1999', 2 </pre>		

Affichage de la nouvelle table CHAUFFEUR :

BD_TP1/postgres@PostgreSQL 12					
Query Editor					Scratch Pad
<pre>1 SELECT * FROM CHAUFFEUR</pre>					
Data Output	Explain	Messages	Notifications		
idchauffeur [PK] character varying (20)	nomcomplet character varying (30)	sexe character (1)	datedenaissance date	idemployeur integer	
1 XAW32	Charles Lebeau	M	1968-05-16	0	
2 XAW21	Henriette Garceau	F	1980-03-26	0	
3 ZAQ87	Liam Charlevoix	M	1983-09-19	1	
4 ZAQ92	Belange Leningard	F	1969-07-12	1	
5 WOP96	Charles Fumtim	M	1999-12-17	1	