

Optimiser l'utilisation de sa base de données avec SQL - premiers pas avec SQL

Implémentation d'une Base de données

Stefan.Gaget@cnrs.fr

Attention !

- Stratégies pour la conception du modèle
 - Raisonnement global
- Stratégie pour l'implémentation
 - Travail incrémental

Stratégie pour la conception d'une DB

- Raisonnement global
- Prévoir et structurer l'ensemble des données
- Anticiper les besoins
- Valider le modèle à partir
 - Des données disponibles et prévue
 - Est-ce que je peux mettre toutes mes données ?
 - Des fonctionnalités souhaitées
 - Est-ce que je peux écrire des requêtes pour répondre à toutes mes questions ?
- Choisir le niveau de détail en fonction des questions

Stratégie pour l'implémentation d'une DB

- Avancer pas à pas : mode incrémental
- Commencer par 1 table
 - Créer la table, la remplir, tester quelques requêtes
- Ajouter une seconde table
 - Créer la table et les relations avec la/les tables précédentes, remplir la 2ème table, tester des requêtes avec ces 2 tables
- Etc ...

Dans quel ordre choisir les tables ?

- Commencer par les tables les plus « externes »
 - Avec le moins de clés étrangères
- Puis choisir les tables en relations avec les précédentes
- Cette approche permet de détecter les problèmes au fur et à mesure

Avantage de l'approche incrémentale

- Déetecter des problèmes au fur et à mesure
- Exemple de problème
 - Nom de table ou de champ incorrect
 - Type différent entre clé primaire et clé étrangère
 - Ex clé primaire INT, clé étrangère VARCHAR
 - Mauvais type de champ
 - Ex : type VARCHAR au lieu de date => year(T1.C1)

Choix des index

- Adapter le schéma aux données disponibles
- Si les données possèdent un champ utilisable comme index
- Il peut être plus judicieux de l'utiliser

Implémentation des requêtes

- Select simple
 - => valide les noms des tables et des champs
- Puis ajouter count, group by
- Puis ajouter order, titres

Debug / Validation des requêtes

- Requête sur 5 tables => Erreur : Comment débugger ?
 - Erreur de syntaxe ? (+ facile)
 - Erreur de logique ? Pas de message d'erreur, mais la requête ne produit pas le résultat attendu
 - Utilisation de la mauvaise table ?
 - Utilisation erronée d'une relation ?
- 1) Connaître à l'avance les résultats pour une requête
 - A calculer « à la main » sur un petit jeu d'essai
- 2) Écrire la requête de manière incrémentale (cf diapo précédente)
- 3) Écrire les requêtes dans fichier .sql pour les exécuter directement à partir de celui-ci (cf commande \i de psql)

Les erreurs courantes

- Table inconnue
- Champ inconnu
- Type de champ
 - Date, Texte sont différentes
- Taille des champs (texte)
- Type de champ différents entre PK et FK
- ' et " ne sont pas synonymes

D'un point de vue pratique ?

- Comment intégrer vos données ?
 - A la main, en ligne de commande
 - Par des scripts
 - Avec un outil d'administration de base de données

D'un point de vue pratique ?

- Comment intégrer vos données ?
 - **A la main, en ligne de commande**
 - Par des scripts
 - Avec un outil d'administration de base de données

En ligne de commande

- (**CREATE TABLE**
... pour créer la
table vide)
- **COPY** pour charger
les valeurs à partir
du fichier .CSV ou
.txt (; ou TAB)
 - COPY data_table
(colonne1, colonne2, ...)
FROM nom_fichier WITH
options ;
- **COPY n'est pas un
mot clé standard
de SQL ...**

Exemple fichier texte avec délimiteurs

D:\Profils\gaget\Desktop\test-data-MODYPSC-patient.csv - Sublime Text (UNREGISTERED)

```
1 id_pat;id_fam;dna;sex;bmi;age_diab;age_exam;gly0;ttt_diab;statut_diab
2 1;8;49;feminin;22.656;27;54;8.2;Aucun;MODY
3 2;8;50;feminin;29.402;52;71;16.5;;MODY
4 3;8;51;feminin;30.119;80;84;9.1;Aucun;MODY
5 4;8;58;feminin;19.723;15;28;6.1;Regime;MODY/IG
6 5;8;59;feminin;21.99;33;51;7.4;Sulfamides;MODY
7 6;8;256;feminin;24.69;49;60;8.2;Aucun;MODY
8 7;8;257;masculin;26.543;22;33;7.7;Aucun;MODY/IG
9 8;8;1239;masculin;21.107;60;79;5.1;Sulfamides;MODY
10 9;8;1248;masculin;23.457;38;44;5.5;Regime;MODY/IG
11 10;8;1267;masculin;20.761;11;30;8;Biguanides;MODY
12 11;8;1270;feminin;20.761;13;26;7;Regime;MODY/IFG
13 12;8;1496;masculin;19.37;17;17;6.6;Aucun;MODY
14 13;8;1535;feminin;20.83;28;55;8.4;Regime;MODY
15 14;8;1911;masculin;22.835;28;28;8.2;Aucun;MODY
16 15;8;1990;feminin;29.136;56;60;7.7;Bs;MODY
17 16;8;2004;masculin;24.725;35;35;6.8;Aucun;MODY/IFG
18 17;8;2006;masculin;17.313;8;8;6.6;Aucun;MODY/IFG
19 18;8;2085;feminin;14.863;5;6;6.8;Aucun;MODY/IFG
20 19;8;2101;masculin;14.863;5;6;6.9;Aucun;MODY/IFG
21 20;8;2978;feminin;0;33;33;6.7;;MODY/IG
22 21;28;115;feminin;14.605;13;13;6.8;Aucun;MODY/IG
23 22;28;117;masculin;17.746;7;15;6.5;Regime;MODY/IFG
```

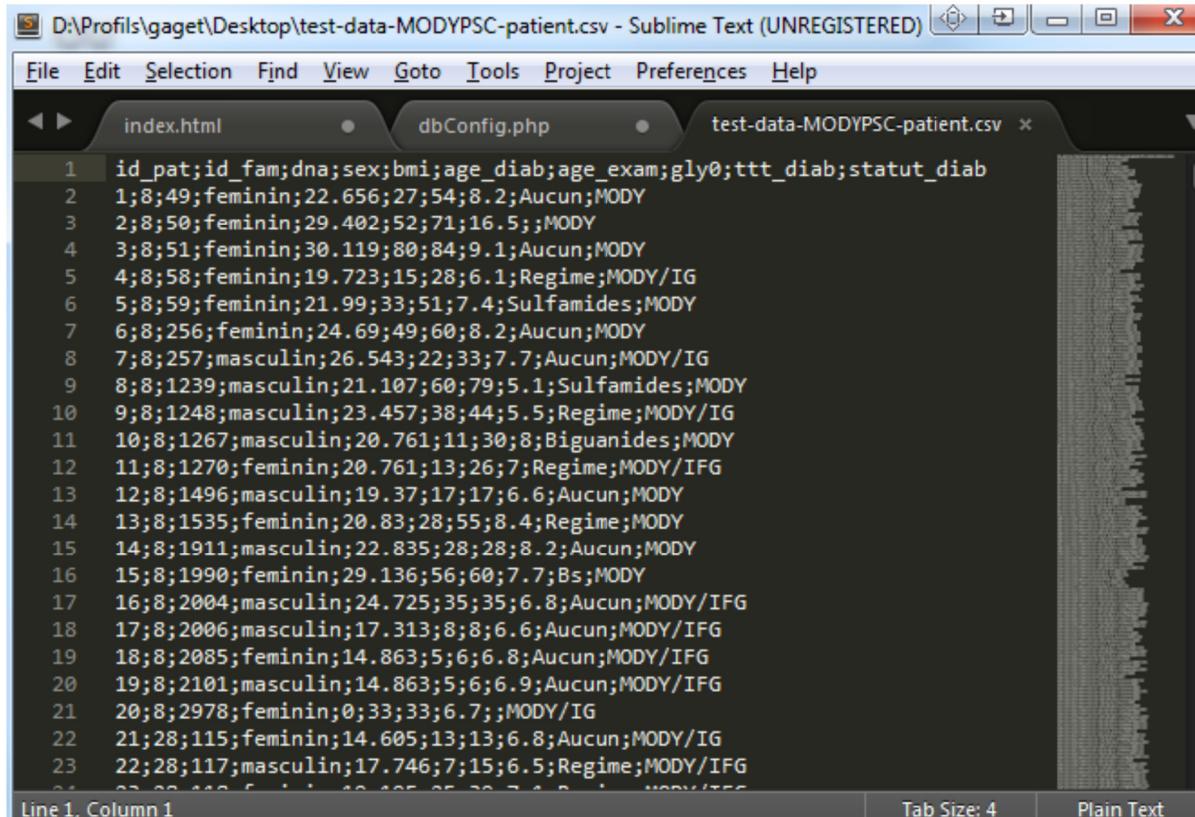
En ligne de commande

- **Les options de COPY :**

- **CSV HEADER** permet de sauter la 1ère ligne (contenant les noms de colonne)
- **DELIMITER ‘ ;’** ou **DELIMITER ‘\t’** pour définir le délimiteur de champ
- **NULL ‘NA’** pour définir une chaîne équivalente à NULL (par défaut : “”)
- **ENCODING ‘UTF8’** ou **ENCODING ‘WIN1252’** pour définir l'encodage (UTF8 ou ANSI)
 - Notepad++ permet de vérifier l'encodage
 - Dans Excel : « Texte (séparateur : tabulation)(*.txt) », « CSV (séparateur : pointvirgule)(*.csv) » donnent des fichiers ANSI
 - Dans Excel : « Texte Unicode (*.txt) » donne des fichiers UTF8

En ligne de commande

- Exemple avec un fichier CSV et UTF8 :



The screenshot shows a Sublime Text window with three tabs: 'index.html', 'dbConfig.php', and 'test-data-MODYPSC-patient.csv'. The 'test-data-MODYPSC-patient.csv' tab contains the following data:

	id_pat	id_fam	dna	sex	bmi	age_diab	age_exam	gly0	ttt_diab	statut_diab
1	1;8;49	feminin	22.656	27;54	8.2	Aucun	MODY			
2	2;8;50	feminin	29.402	52;71	16.5					MODY
3	3;8;51	feminin	30.119	80;84	9.1	Aucun	MODY			
4	4;8;58	feminin	19.723	15;28	6.1	Regime	MODY/IG			
5	5;8;59	feminin	21.99	33;51	7.4	Sulfamides	MODY			
6	6;8;256	feminin	24.69	49;60	8.2	Aucun	MODY			
7	7;8;257	masculin	26.543	22;33	7.7	Aucun	MODY/IG			
8	8;8;1239	masculin	21.107	60;79	5.1	Sulfamides	MODY			
9	9;8;1248	masculin	23.457	38;44	5.5	Regime	MODY/IG			
10	10;8;1267	masculin	20.761	11;30	8	Biguanides	MODY			
11	11;8;1270	feminin	20.761	13;26	7	Regime	MODY/IFG			
12	12;8;1496	masculin	19.37	17;17	6.6	Aucun	MODY			
13	13;8;1535	feminin	20.83	28;55	8.4	Regime	MODY			
14	14;8;1911	masculin	22.835	28;28	8.2	Aucun	MODY			
15	15;8;1990	feminin	29.136	56;60	7.7	Bs	MODY			
16	16;8;2004	masculin	24.725	35;35	6.8	Aucun	MODY/IFG			
17	17;8;2006	masculin	17.313	8;8	6.6	Aucun	MODY/IFG			
18	18;8;2085	feminin	14.863	5;6	6.8	Aucun	MODY/IFG			
19	19;8;2101	masculin	14.863	5;6	6.9	Aucun	MODY/IFG			
20	20;8;2978	feminin	0;33	33;6	7.7		MODY/IG			
21	21;28;115	feminin	14.605	13;13	6.8	Aucun	MODY/IG			
22	22;28;117	masculin	17.746	7;15	6.5	Regime	MODY/IFG			
23	23;28;118									MODY/IFG

```
COPY data_table (id_pat, id_fam, dna, bmi, gly0, ttt_diab, statut_diab) FROM 'D:\test-data-MODYPSC-patient.csv' WITH CSV HEADER DELIMITER ';' ENCODING 'UTF8';
```

```
COPY data_table FROM 'D:\test-data-MODYPSC-patient.csv' WITH CSV HEADER DELIMITER ';' ENCODING 'UTF8';
```

En ligne de commande

- Attention, **COPY** nécessite les droits administrateur. Utilisez **\copy** à la place
- !!! psql s'exécute sur le serveur, le chemin spécifié pour le fichier doit donc être celui sur le serveur.
(dans le cadre de la formation, en monoposte, pas de soucis) !!!

En ligne de commande

- **Application pratique :**
 - Créez un fichier csv avec les données suivantes

id_v	nom
2	Guines
3	Licques
4	Saint-Amand-les-Eaux

- Alimentez la table **ville**, de la base **espacenat**, (utilisateur **tom**...) avec psql et l'instruction **COPY**

En ligne de commande



```
SQL Shell (psql)
espacenat=>
espacenat=>
espacenat=> COPY ville FROM 'D:/Profils/gaget/ville-light.csv' WITH CSV HEADER DELIMITER ';' ENCODING 'UTF8';
ERREUR:  doit ûtre super-utilisateur pour utiliser COPY à partir ou vers un fichier
ASTUCE : Tout le monde peut utiliser COPY vers stdout ou à partir de stdin.
La commande \copy de psql fonctionne aussi pour tout le monde.
espacenat=> \copy ville FROM 'D:/Profils/gaget/ville-light.csv' WITH CSV HEADER DELIMITER ';' ENCODING 'UTF8';
COPY 3
espacenat=> SELECT * FROM ville;
 id_v |          nom
-----+
 0    | Marquise
 1    | Samer
 2    | Guines
 3    | Licques
 4    | Saint-Amand-les-Eaux
(5 lignes)

espacenat=>
```

D'un point de vue pratique ?

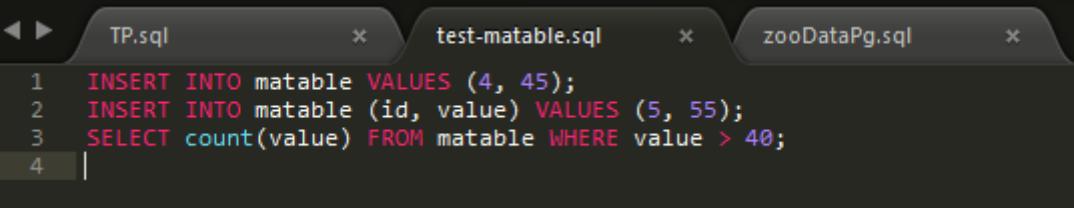
- Comment intégrer vos données ?
 - A la main, en ligne de commande
 - **Par des scripts**
 - Avec un outil d'administration de base de données

Par des scripts (SQL)

- **Avec la Commande \i :**
 - Création des commandes SQL dans un fichier texte (extension .sql)
 - Utilisation de l'instruction psql \i suivi du nom (+chemin) du fichier .sql
 - Attention : mettre des " pour le fichier et les chemin (windows) avec des /
 - ex : \i 'd:/MonProfile/scripts/test1.sql'

Par des scripts (SQL)

- Exemple :



```
TP.sql      * test-matable.sql      * zooDataPg.sql      *
1  INSERT INTO matable VALUES (4, 45);
2  INSERT INTO matable (id, value) VALUES (5, 55);
3  SELECT count(value) FROM matable WHERE value > 40;
4  |
```

```
mabase=> select * from matable;
+-----+
| id | value |
+-----+
| 1  | 5     |
| 2  | 65    |
| 2  | 65    |
| 4  | 45    |
| 5  | 55    |
| 4  | 45    |
| 5  | 55    |
| 4  | 45    |
| 5  | 55    |
| 4  | 45    |
| 5  | 55    |
<11 lignes>

mabase=> \i 'd:/Profils/gaget/test-matable.sql'
INSERT 0 1
INSERT 0 1
count
-----
 12
<1 ligne>

mabase=>
```

Par des scripts (SQL)

- **Application pratique :**

- Créez un fichier de commande sql
 - insérant les données suivantes dans la table **espaceNat**

id_v	nom
5	Raimes
6	Marchiennes
7	Avesnes-sur-Helpe

- calculant le nombre d'enregistrements contenus dans la table **ville**
- calculant la moyenne, la valeur minimum et la valeur maximum du champ **id_v** de la table **ville**
- Exécutez les instructions de ce fichier avec `psql`

Par des scripts (SQL)

```
TP.sql * test-matable.sql * zooDataPg.sql

1 INSERT INTO ville VALUES (5, 'Raismes');
2 INSERT INTO ville VALUES (6, 'Marchiennes');
3 INSERT INTO ville VALUES (7, 'Avesnes-sur-Helpe');
4 SELECT MIN(id_v), MAX(id_v), AVG(id_v) FROM ville;
5

SQL Shell (psql) X

espacenat=> SELECT * FROM ville;
 id_v | nom
-----+
 0    | Marquise
 1    | Samer
 2    | Guines
 3    | Licques
 4    | Saint-Amand-les-Eaux
<5 lignes>

espacenat=> \i 'D:/Profils/gaget/test-matable.sql';
INSERT 0 1
INSERT 0 1
INSERT 0 1
 min | max |      avg
-----+
 0   |  7  | 3.50000000000000000000
<1 ligne>

espacenat=> SELECT * FROM ville;
 id_v | nom
-----+
 0    | Marquise
 1    | Samer
 2    | Guines
 3    | Licques
 4    | Saint-Amand-les-Eaux
 5    | Raismes
 6    | Marchiennes
 7    | Avesnes-sur-Helpe
<8 lignes>

espacenat=> _
```

D'un point de vue pratique ?

- Comment intégrer vos données ?
 - A la main, en ligne de commande
 - Par des scripts
 - **Avec un outil d'administration de base de données**

Avec un outil d'administration de base de données

- Interface client
- Multi-plateformes, multi-installations
- Génération de code automatique
- Proscrire l'écriture d'instructions SQL (« click and go »)

Avec un outil d'administration de base de données

- Les fonctionnalités
 - Création, éditions de bases, tables, ...
 - Édition, exécution d'instructions SQL
 - Importer/Exporter données et structures
- PgAdmin, PhpPgAdmin, SqlWorkbench, DBeaver, HeidiSql, Adminer...

HeidiSQL

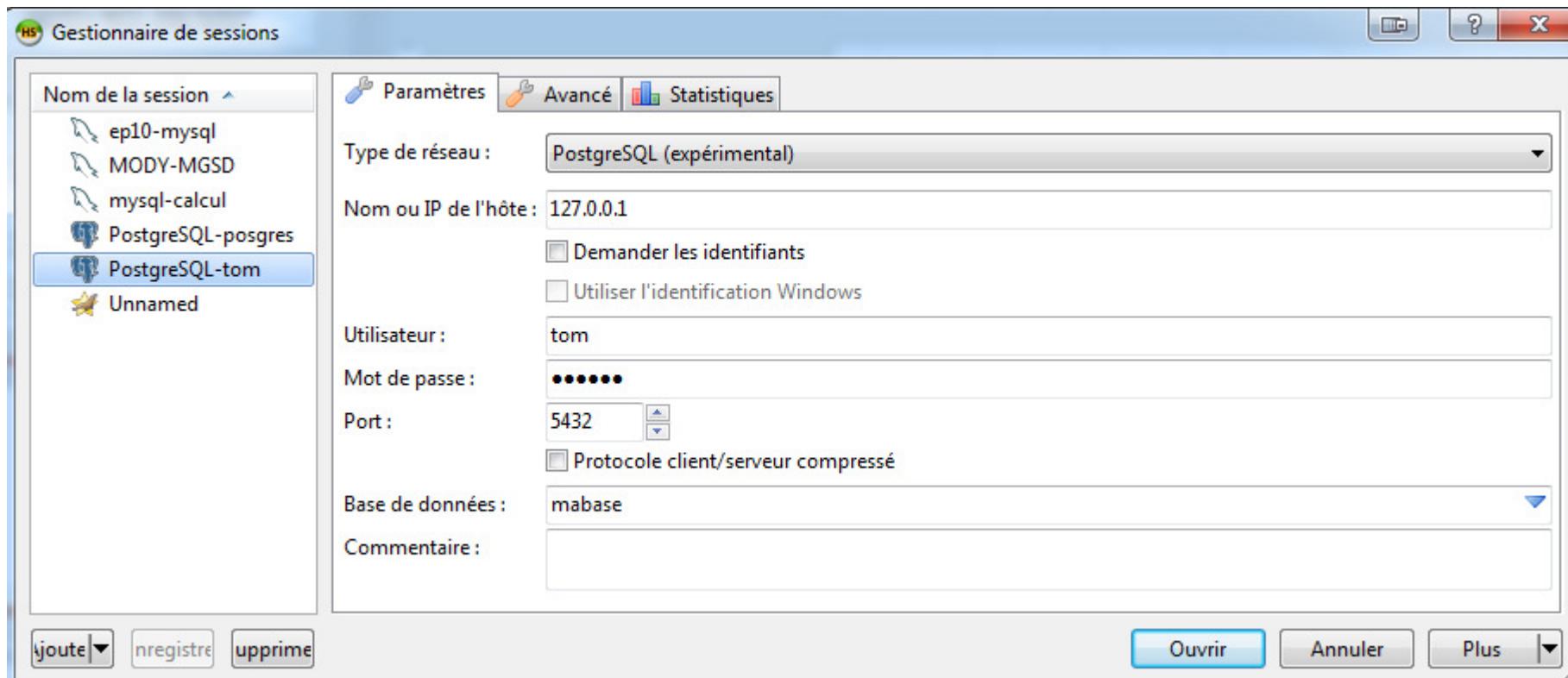
- <http://www.heidisql.com/>
- Projet libre depuis 2006, version 9.4 en octobre 2016
- Communauté, forum des utilisateurs
- Installation locale sur poste de travail
- Windows (installeur ou portable)
- (linux et Mac OSX avec Wine)



- Développé et optimisé pour MySQL
- Connexion avec Mysql, Microsoft SQL et **Postgresql**

HeidiSQL

- Connexion au serveur :



- Mêmes paramètres de connexions que pour psql

HeidiSQL

• Vue générale (1) :

The screenshot shows the HeidiSQL interface for PostgreSQL. The left sidebar lists databases: 'PostgreSQL-tom', 'information_schema', 'pg_catalog', 'pg_temp_1', 'pg_toast', 'pg_toast_temp_1', and 'public'. The 'public' database is selected, showing a table named 'matable' with a size of 16,0 KiB. The main window displays the 'Table: matable' configuration screen. The 'Nom:' field is set to 'matable'. The 'Colonnes:' section contains two columns: 'id' (Type: INTEGER) and 'value' (Type: INTEGER). The bottom pane shows a series of SQL commands related to the current session.

#	Nom	Type de données	Taille/Ensemble	Non signé	NULL autorisé	ZEROFILL	Par défaut	Commentaire	Collation
1	id	INTEGER		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
2	value	INTEGER		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

```
4 SELECT VERSION();
5 /* Connecté. ID du processus : 6096 */
6 SET statement_timeout TO 0;
7 SELECT EXTRACT(EPOCH FROM CURRENT_TIMESTAMP - pg_postmaster_start_time())::INTEGER;
8 SELECT "nsname" FROM "pg_catalog"."pg_namespace" ORDER BY "nsname";
9 /* Ouverture de la session « PostgreSQL-tom » */
10 SELECT *, pg_table_size(QUOTE_IDENT(t.TABLE_SCHEMA) || E'.' || QUOTE_IDENT(t.TABLE_NAME))::bigint AS data_length, pg_relation_size(QUOTE_IDENT(t.TABLE_SCHEMA) || E'.' || QUOTE_IDENT(t.TABLE_NAME))::bigint AS index_length,
11 /* Type de donnée inconnu oid n°1034 pour « nspat ». Repli vers UNKNOWN. */
12 /* Type de donnée inconnu oid n°1034 pour « relact ». Repli vers UNKNOWN. */
13 /* Type de donnée inconnu oid n°1009 pour « reloptions ». Repli vers UNKNOWN. */
14 SELECT "p"."proname" FROM "pg_catalog"."pg_namespace" AS "n" JOIN "pg_catalog"."pg_proc" AS "p" ON "p"."oid" WHERE "n"."nsname"=E'public';
15 SET search_path TO E'public';
16 SELECT DISTINCT a.attname AS column_name, a.attnum, a.atttypid, a.atttypmod, FORMAT_TYPE(a.atttypid,a.atttypmod) AS data_type, CASE a.attnotnull WHEN false THEN E'YES' ELSE E'NO' END AS IS_NULLABLE, com.description AS colu
17 WITH ndx_list AS ( SELECT pg_index.indexrelid, pg_class.oid FROM pg_index, pg_class WHERE pg_class.relname = E'matable' AND pg_class.oid = pg_index.indrelid ), ndx_cols AS ( SELECT pg_class.relname, UNNEST(
```

HeidiSQL

• Vue générale (2) :

The screenshot shows the HeidiSQL interface for a PostgreSQL connection named "PostgreSQL-tom\public\matable". The main window is divided into several panes:

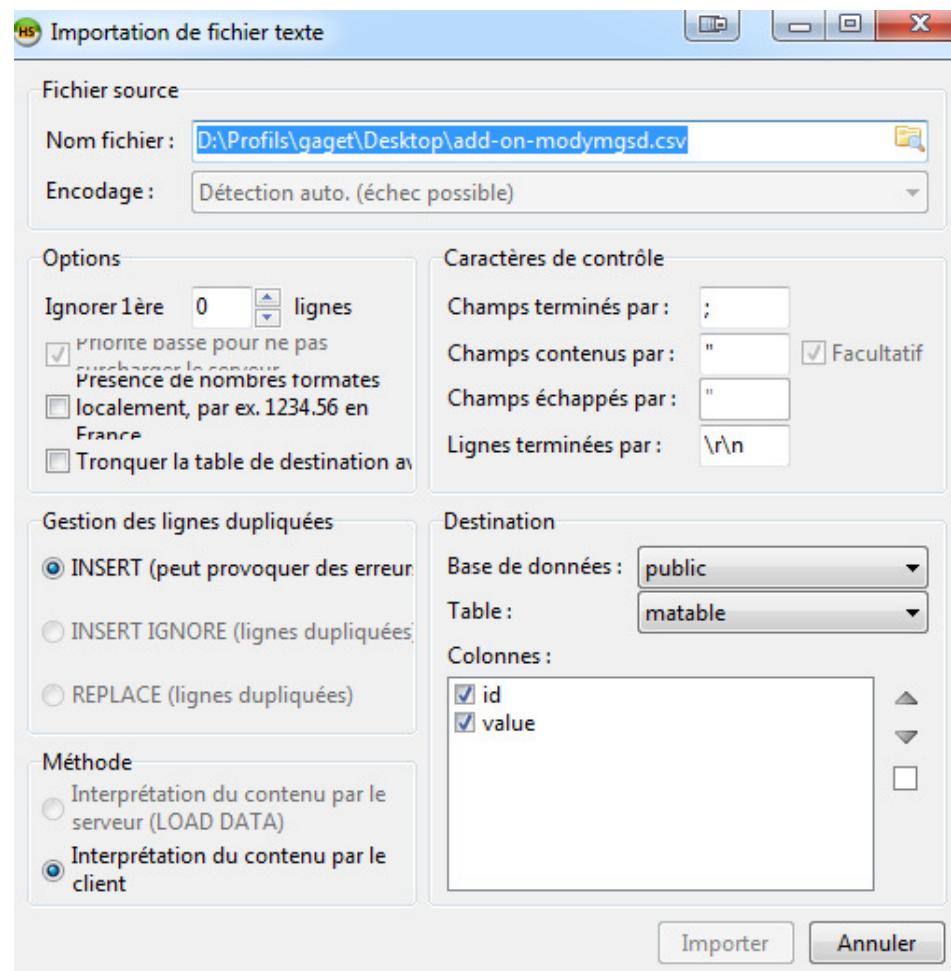
- Left pane (Object browser):** Shows the database schema with the "matable" table selected.
- Top center pane (Query editor):** Displays the SQL query: `1 select * from matable;`
- Middle right pane (Table viewer):** Shows the data in the "matable" table:

id	value
1	5
2	65
2	65
4	45
5	55
4	45
5	55
4	45
5	55
4	45
4	45
4	45
4	45

- Bottom pane (Log/Output):** Displays the full SQL query and its execution results, including statistics like "Lignes trouvées: 13" and "Durée pour 1 requête: 0,000 sec."

HeidiSQL

- L'outil **Importer** dans HeidiSQL
(Outils / Importer un fichier csv...):



HeidiSQL

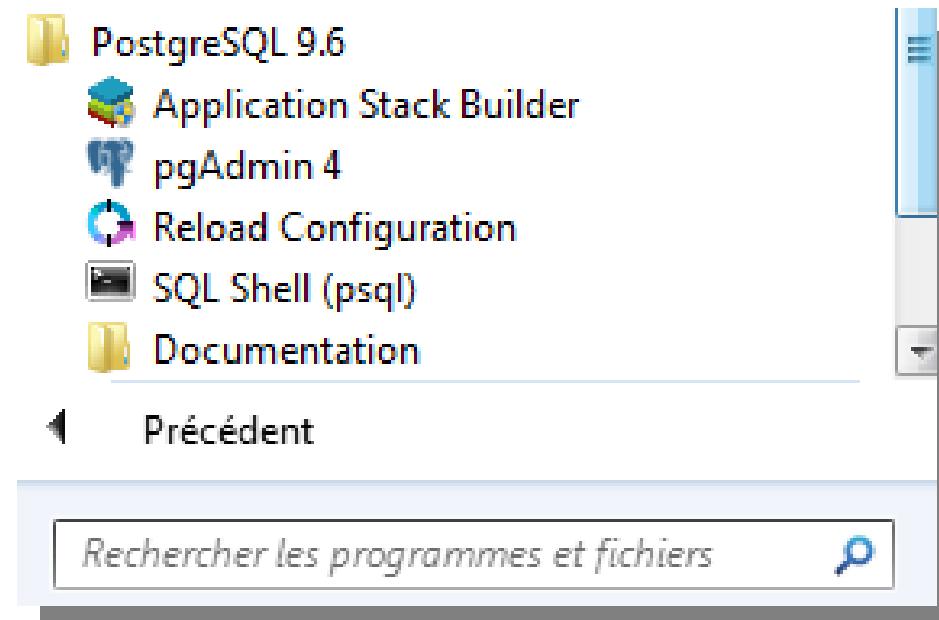
- Supprimer les données de la table **ville**
- Utiliser le fichier « ville-light.csv » afin de re-charger les données dans la table **ville** avec HeidiSQL

HeidiSQL

- Supprimer les données de la table **ville**
- Utiliser le fichier « table_ville_all.sql » afin de charger l'ensemble des données dans la table **ville** avec HeidiSQL
 - Avec l'interface d'exécution de commandes sql

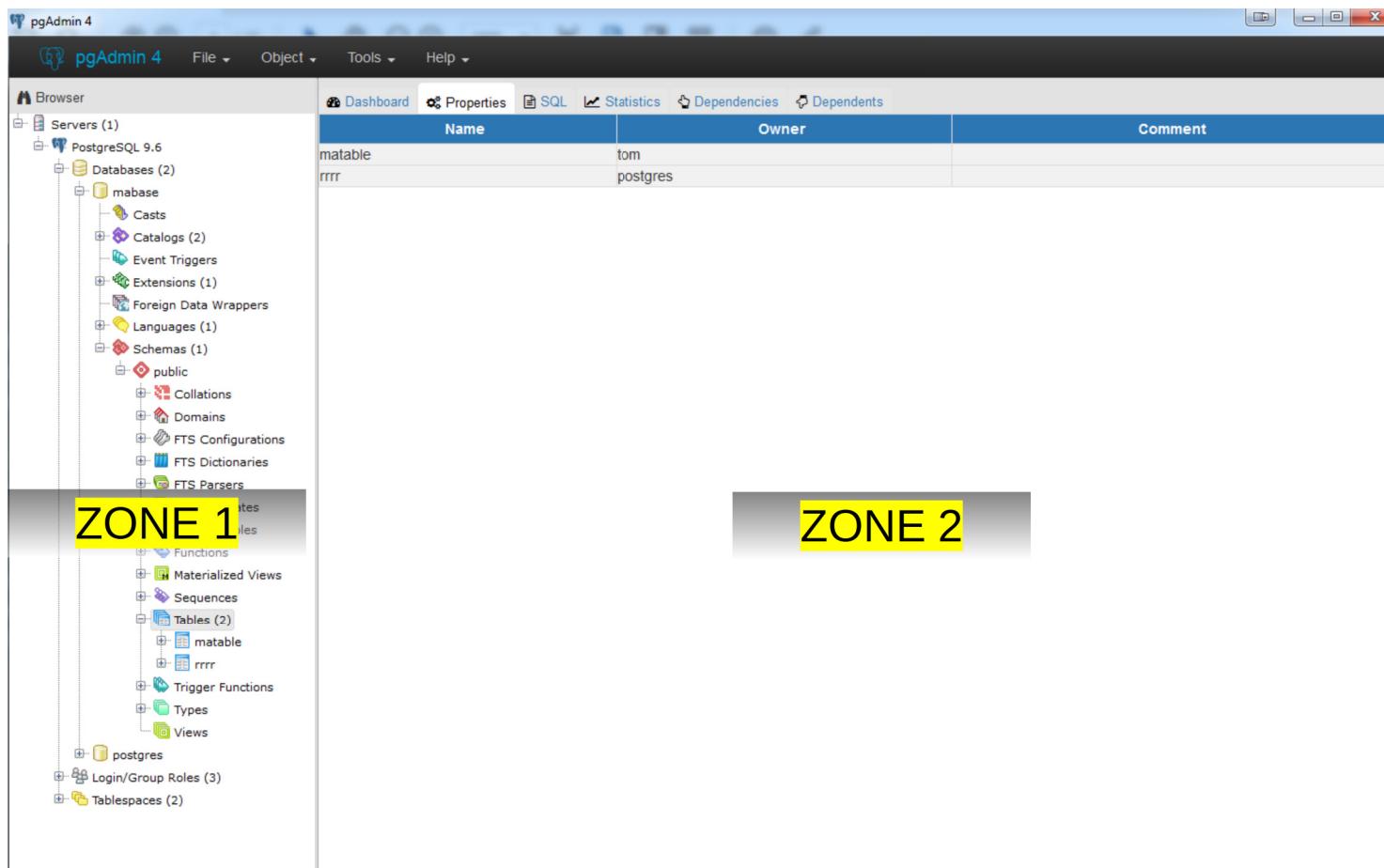
pgAdmin

- Outil d'administration de PostgreSQL
- Se lance depuis le menu démarrer, comme pour psql



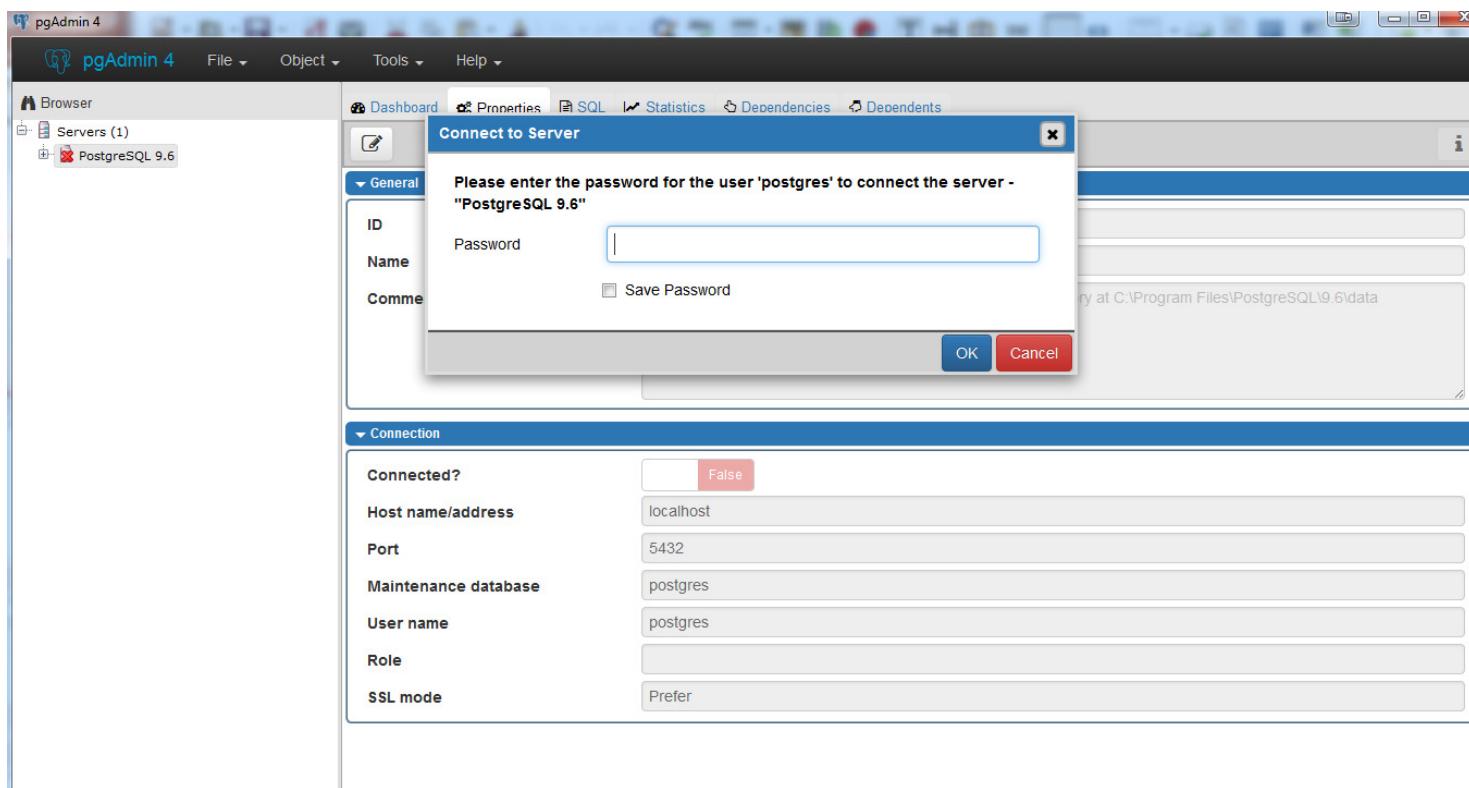
pgAdmin

- La zone 1, présente un navigateur hiérarchisé des objets : Un Groupe de Serveurs -> Serveurs -> des instances de serveur -> des Bases de données -> Schémas -> Tables -> Attributs.
- La zone 2, affichera les propriétés de l'objet sélectionné en zone 1, sera la zone de travail, d'information ...



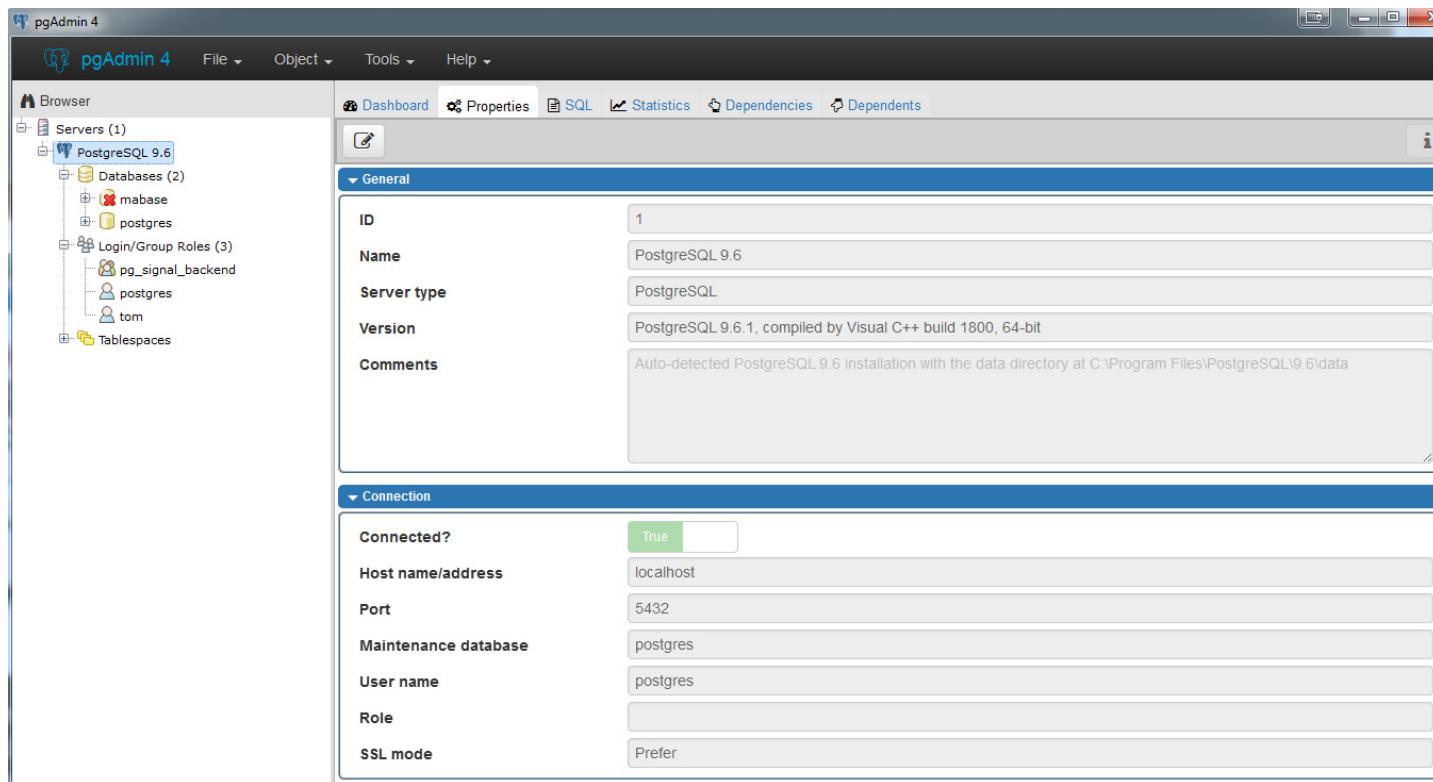
pgAdmin

- Double clic sur le serveur (croix rouge) en zone1
- La fenêtre de connexion s'affiche. Il faut ensuite saisir le mot de passe de l'utilisateur postgres (celui que vous avez défini à l'installation du serveur)



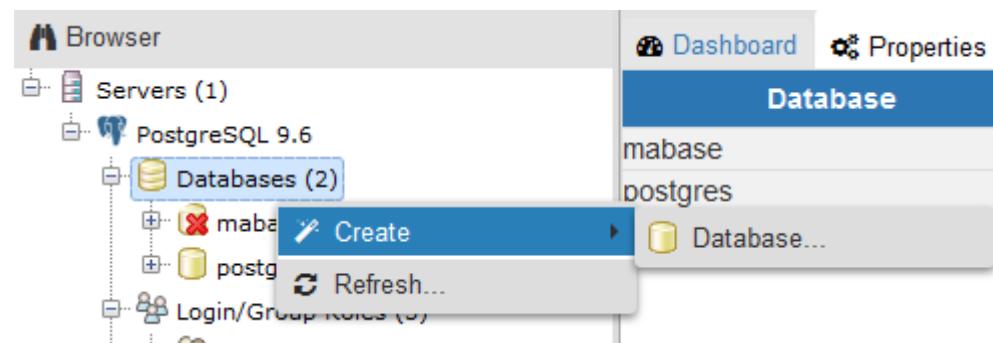
pgAdmin

- La connexion est effective
- Le navigateur affiche les objets disponibles Dans la cas présent nous avons :
 - 2 bases de données (matable et postgres créée lors de l'installation)
 - 2 rôles de connexions (tom et postgres)
 - 0 rôle de groupe



pgAdmin

- **Création d'une base de données** : clic-droit sur « Base de données » puis « ajouter une base de données »

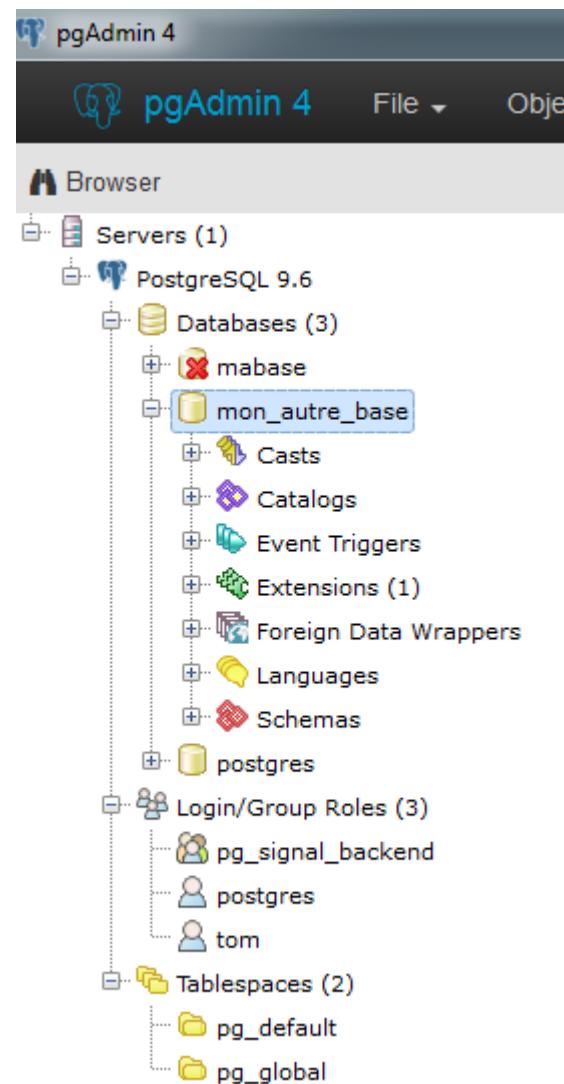
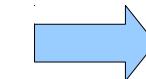
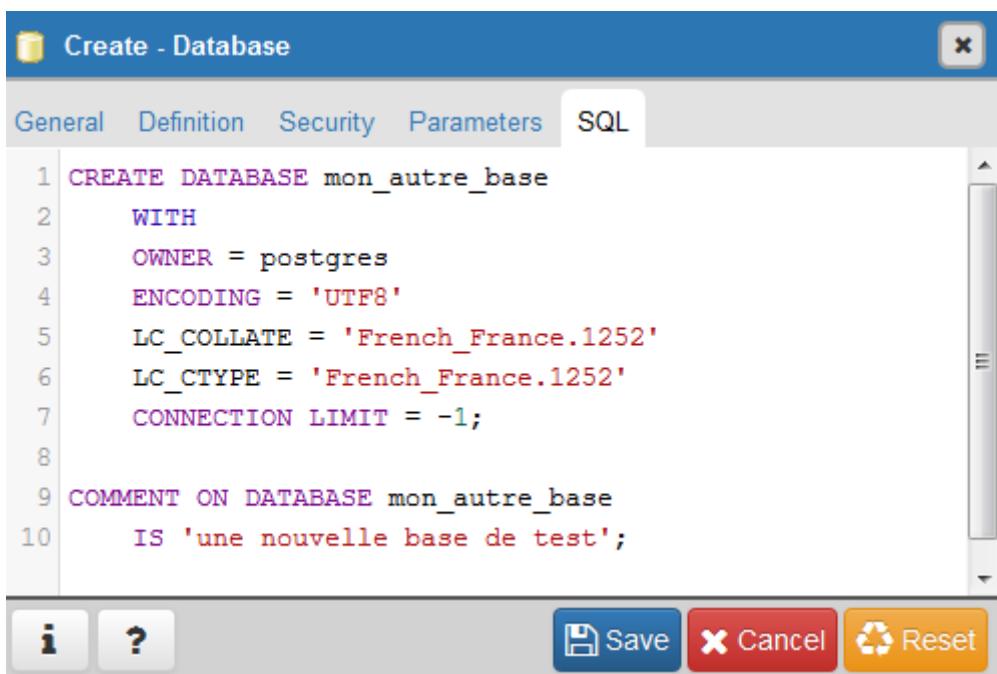
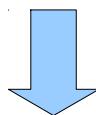


- Remplir ensuite les onglets « Général » et « définition »

Two screenshots of the 'Create - Database' dialog. The first screenshot shows the 'General' tab with fields: 'Database' set to 'mon_autre_base', 'Owner' set to 'postgres', and 'Comment' set to 'une nouvelle base de test'. The second screenshot shows the 'Definition' tab with fields: 'Encoding' set to 'UTF8', 'Template' set to 'Select from the list', 'Tablespace' set to 'Select from the list', 'Collation' set to 'French_France.1252', 'Character type' set to 'French_France.1252', and 'Connection limit' set to '-1'. Blue arrows point from the 'General' tab to the 'Definition' tab, indicating the flow of configuration.

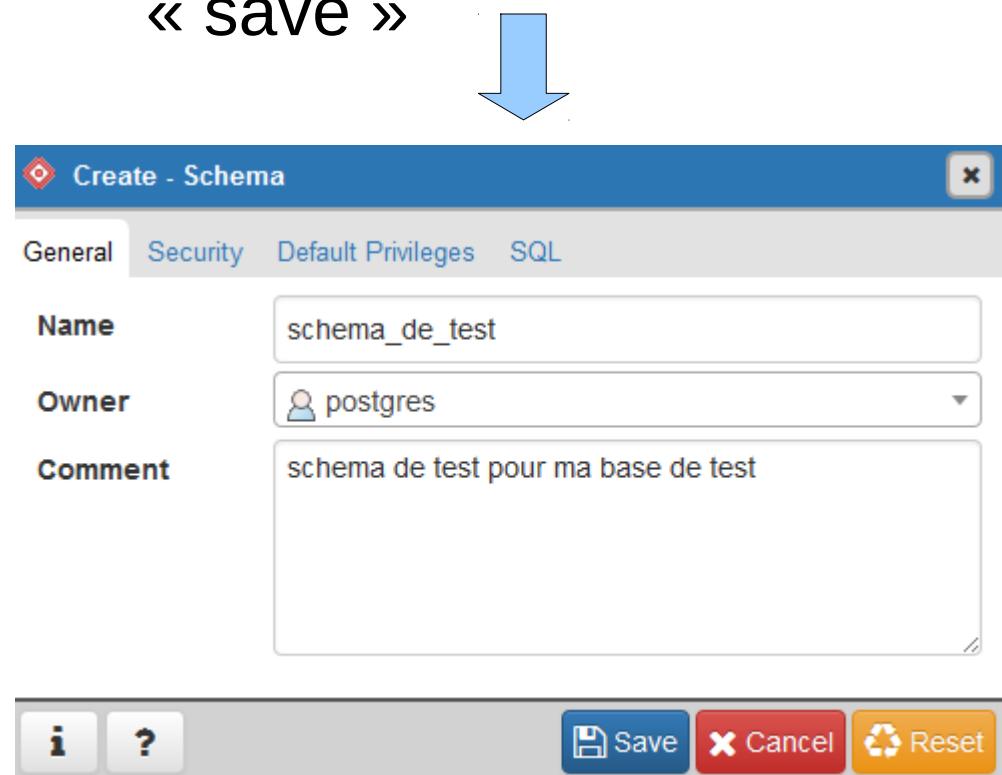
pgAdmin

- L'onglet « SQL » donne les instructions SQL qui vont être exécutées.
- Appuyez sur « sauver (OK) »



pgAdmin

- **Création d'un Schéma** : clic-droit sur « Schemas », puis « creat » ... remplir l'onglet « général » ... terminer par « save »

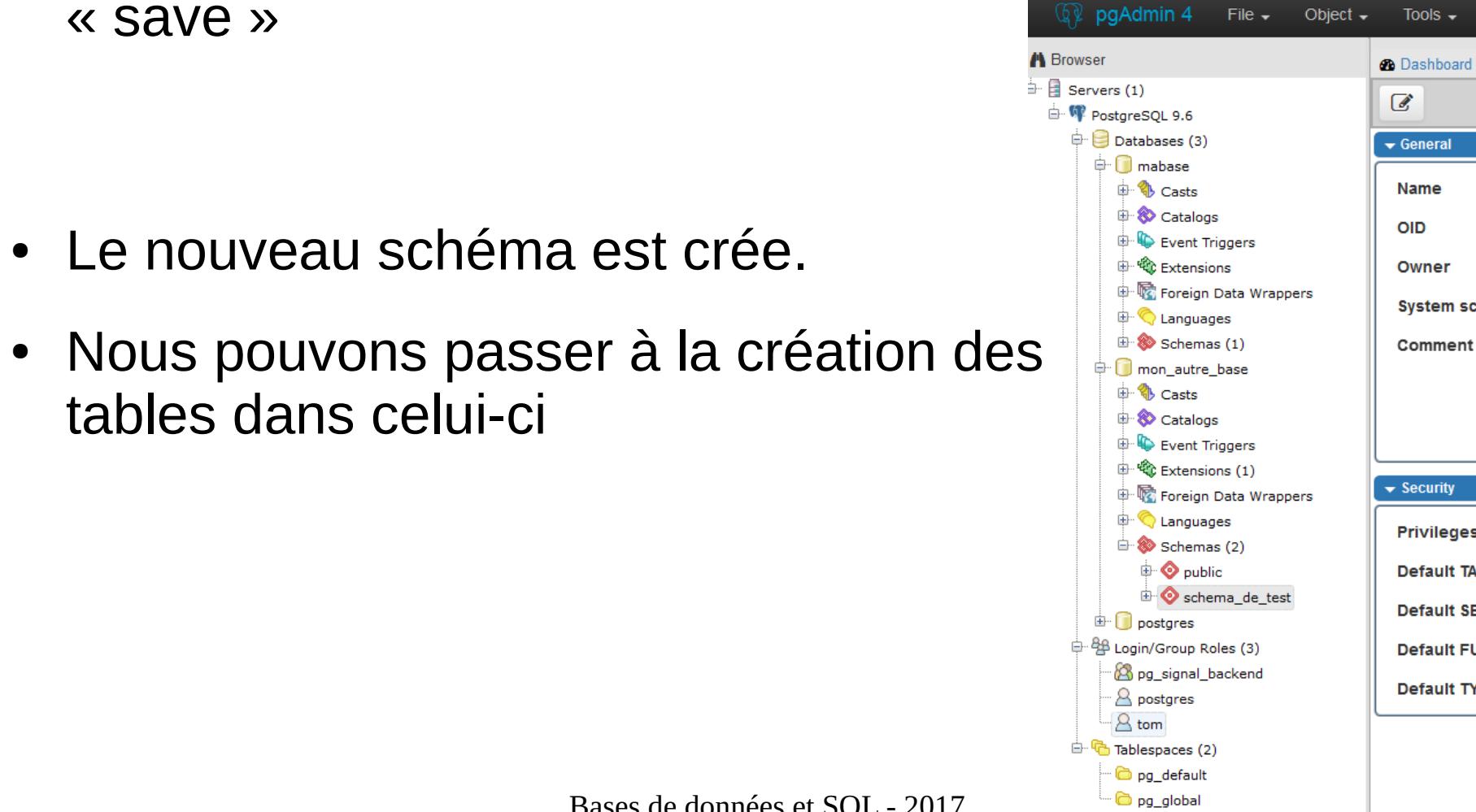


'les Schémas' extrait de la doc PostgreSQL :

Une base de données contient un ou plusieurs schéma(s) nommé(s) qui, eux, contiennent des tables. Les schémas contiennent aussi d'autres types d'objets nommés (types de données, fonctions et opérateurs, par exemple). Le même nom d'objet peut être utilisé dans différents schémas sans conflit ; par exemple, schema1 et mon_schema peuvent tous les deux contenir une table nommée ma_table. À la différence des bases de données, les schémas ne sont pas séparés de manière rigide : un utilisateur peut accéder aux objets de n'importe quel schéma de la base de données à laquelle il est connecté, sous réserve qu'il en ait le droit.

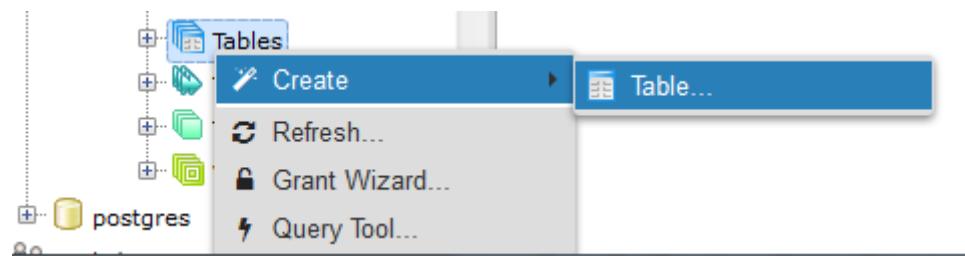
pgAdmin

- **Création d'un Schéma** : clic-droit sur « Schemas », puis « creat » ... remplir l'onglet « général » ... terminer par « save »

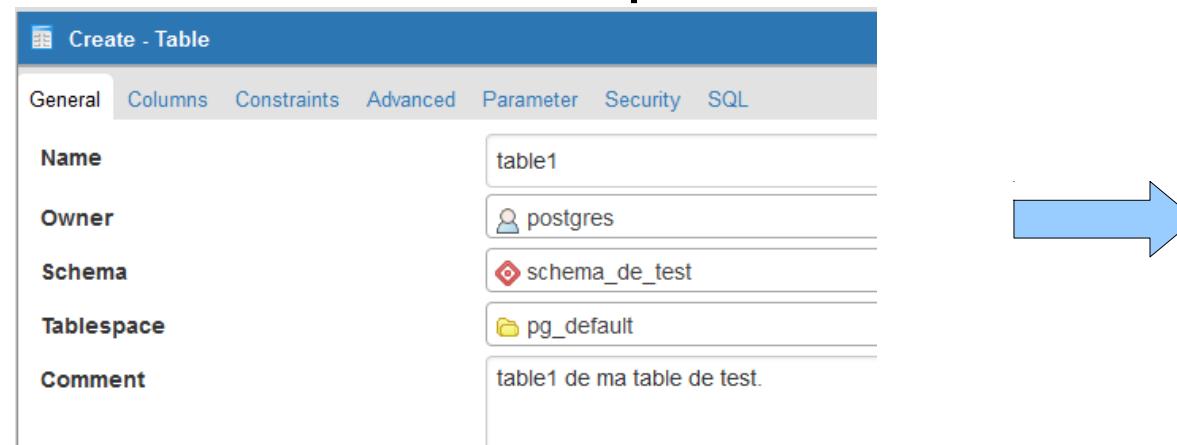


pgAdmin

- **Création d'une table** : Dans le schéma choisi, clic-droit sur « Tables » puis « ajouter une table »



- Remplir ensuite l'onglet « Général » et dans l'onglet « colonnes », cliquer sur « ajouter » ou « + » pour passer à la création des différents champs



pgAdmin

Create - Table

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s) Select to inherit from...

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/> <input type="button" value="Delete"/>		Select from the list			<input type="button" value="No"/>	<input type="button" value="No"/>

Create - Table

General Columns Constraints Advanced Parameter Security SQL

Inherited from table(s) Select to inherit from...

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/> <input type="button" value="Delete"/>	ma_clef	integer			<input type="button" value="No"/>	<input checked="" type="button" value="Yes"/>
<input checked="" type="checkbox"/> <input type="button" value="Delete"/>	nom	character	20		<input type="button" value="No"/>	<input type="button" value="No"/>
<input checked="" type="checkbox"/> <input type="button" value="Delete"/>	prenom	character	25		<input type="button" value="No"/>	<input type="button" value="No"/>

pgAdmin

The screenshot shows the pgAdmin interface with the following details:

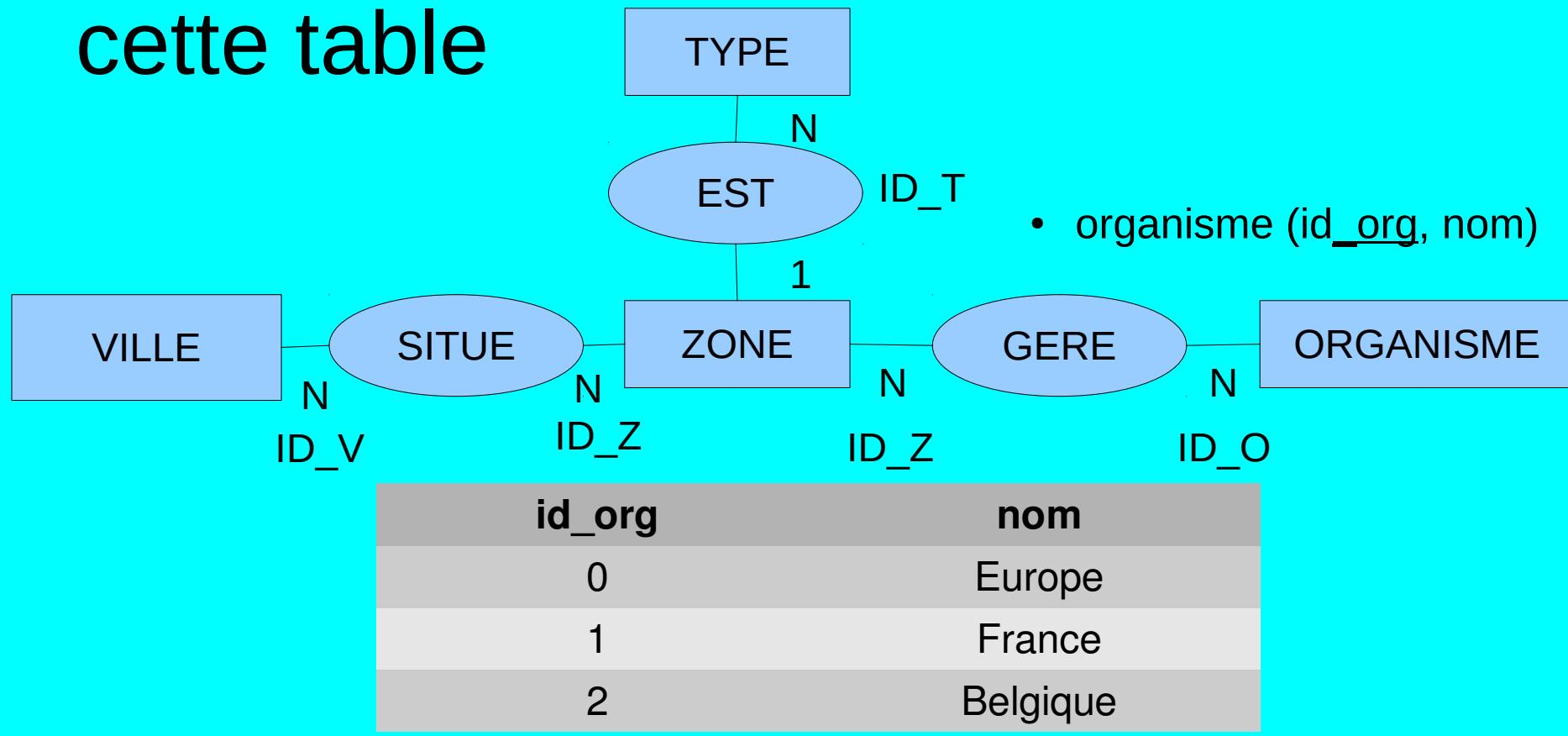
- Browser Panel:** Displays a tree view of database objects under the schema `schema_de_t`. The `Tables` node has one item, `table1`.
- Context Menu:** A right-clicked context menu is open over the `table1` entry. The `View Data` option is highlighted.
- Properties Panel:** On the right, the `General` tab is selected. It contains the following properties:
 - Name:** table1
 - OID:** 1
 - Of type:** (empty)
 - Has OIDs?**: Unchecked
 - Unlogged?**: Unchecked
 - Primary key**: (empty)
 - Rows (estimated)**: (empty)
 - Rows (counted)**: (empty)
 - Inherits table?**: (empty)
- Top Bar:** Includes tabs for `Dashboard`, `Properties`, `SQL`, and other icons.

pgAdmin

The screenshot shows the pgAdmin 4 application interface. On the left, the 'Browser' panel displays a tree view of a PostgreSQL 9.6 server. Under 'Databases', the 'mibase' database is selected, showing its schema objects like Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, and Schemas. One of the schemas, 'public', is expanded to show tables such as 'matable' and 'rrrr'. Other databases listed are 'mon_autre_base' and 'postgres'. At the bottom of the browser panel, 'Login/Group Roles' and 'Tablespaces' are shown. The main workspace on the right contains a 'Query-1' tab with a SQL editor showing a simple SELECT statement: 'SELECT * FROM public.matable'. Below the editor is a 'Data Output' grid displaying the results of the query. The results show two columns: 'id' (integer) and 'value' (integer). The data consists of ten rows with values (1, 5), (2, 65), (2, 65), (4, 45), (5, 55), (4, 45), (5, 55), (4, 45), (5, 55), and (4, 45). A tooltip 'Download as CSV (F8)' is visible over the download icon in the toolbar.

pgAdmin

- Créer la table **organisme** de la base **espacenat** et charger les données de cette table



APPLICATIONS

1 – Étude des espaces naturels

Espaces naturels

Sur une carte régionale, on a représenté des zones correspondant aux différents Parcs Naturels Régionaux (PNR), Parcs Naturels Transfrontaliers (PNT) et Espaces Naturels Régionaux (ENR) des Hauts-De-France. Chacune de ces zones est repérée par un numéro à 2 chiffres. On souhaite associer des informations complémentaires à ces zones :

- nom de la zone (PNR des Caps et Marais d'Opale, ENR Lille Métropole...)
- type (PNR, PNT, EN)
- liste de villes se trouvant sur le territoire de la zone
- liste des organismes prenant part à la gestion du parc

Voici les zones se trouvant sur votre carte (les listes de villes et d'organismes sont volontairement non exhaustives) :

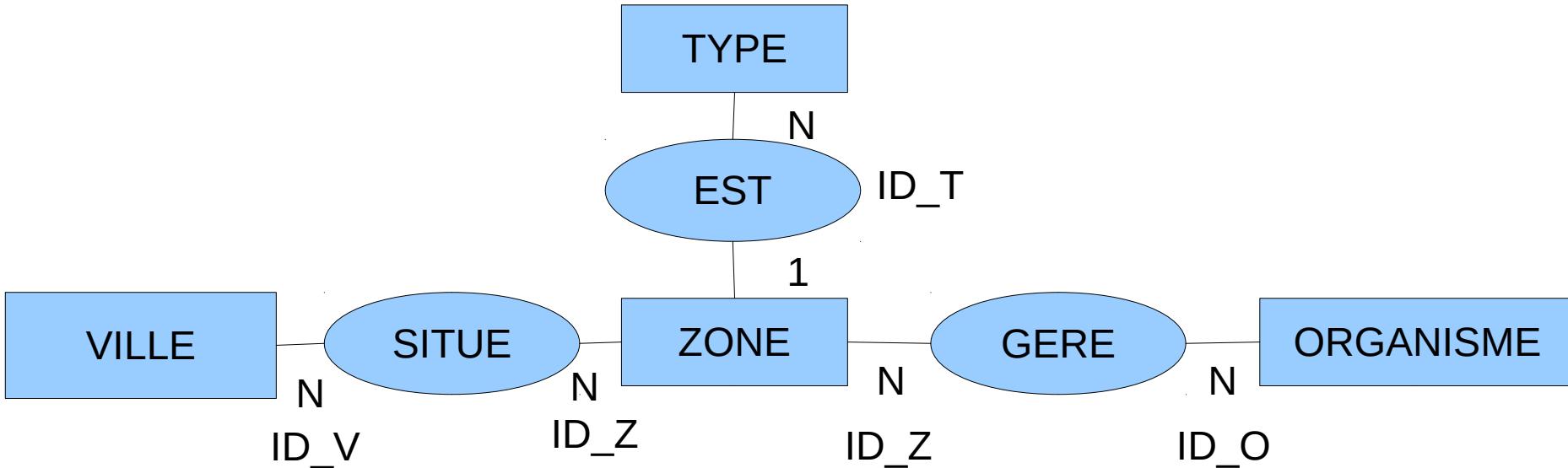
N°	Nom	Type	Villes	Organismes
01	PNR des Caps et Marais d'Opale	PNR	Marquise, Samer, Guines, Licques	France, Région HdF, Département du Pas-de-Calais
02	PNR Scarpe-Escaut	PNR	Saint-Amand-les-Eaux, Raismes, Marchiennes	France, Région HdF, Département du Nord
03	PNR Avesnois	PNR	Maroilles, Avesnes/Helpes, Le Quesnoy	France, Région HdF, Département du Nord
04	PNT du Hainaut	PNT	Saint-Amand-les-Eaux, Raismes, Marchiennes	Europe, France, Région HdF, Département du Nord, Belgique, Région Wallone
05	ENR Lille Métropole	ENR	Croix, Leers, Marcq-en-Barœul, Roubaix, Tourcoing, Wasquehal, Wattrelos	Etat, Région HdF, Département du Nord, Métropole Européenne de Lille

Construisez le modèle entité-relation

Vous organiserez au mieux vos données pour éviter les redondances d'informations et faciliter les recouplements d'informations...

Indiquez pour chacune des entités et des relations la liste des attributs

Espaces Naturels



- VILLE(ID_V, NOM)
- ORG(ID_O, NOM)
- TYPE(ID_T, NOM)
- ZONE(ID_Z, NOM, ID_T)
- SITUE(ID_Z, ID_V)
- GERE(ID_O, ID_Z)

Espaces naturels

Écrivez en SQL les requêtes donnant les résultats ci-dessous

- Liste des types de zones
- Liste des villes se trouvant dans un PNR
- Liste des organismes s'occupant de la gestion de PNT
- Liste des noms de zones dont le Département du Nord assure la gestion (seul ou non)

Espaces Naturels

- Créez la base **espacenat2** pour l'utilisateur **tom**
- Créez les tables et insérez-les données à partir des instructions sql du fichier « **espacenat2.sql** »
- Créez les requêtes demandées et exécutez-les

D'un point de vue pratique ?

- Comment exporter des données ?
 - A la main, en ligne de commande
 - Par des scripts
 - Avec un outil d'administration de base de données

D'un point de vue pratique ?

- Comment exporter des données ?
 - A la main, en ligne de commande
 - Par des scripts
 - Avec un outil d'administration de base de données

En ligne de commande

- **COPY :**

- Une table entière ou à partir d'un **SELECT** pour des données d'une ou plusieurs tables
- Dans psql :
 - **COPY** (SELECT ... FROM ... [WHERE ...]) TO 'd:/test.csv' DELIMITER ',' CSV HEADER ;
 - **COPY** matable TO 'd:/test.csv' WITH (FORMAT CSV, HEADER); (depuis V9.5)
 - **\copy** matable TO 'd:/test.csv' WITH (FORMAT CSV, HEADER);
- Attention, **COPY** nécessite les droits administrateur. Utilisez **\copy** à la place
- !!! psql s'exécute sur le serveur, le chemin spécifié pour le fichier doit donc être celui sur le serveur. (dans le cadre de la formation, en monoposte, pas de soucis) !!!

D'un point de vue pratique ?

- Comment exporter des données ?
 - A la main, en ligne de commande
 - **Par des scripts**
 - Avec un outil d'administration de base de données

Par des scripts (SQL)

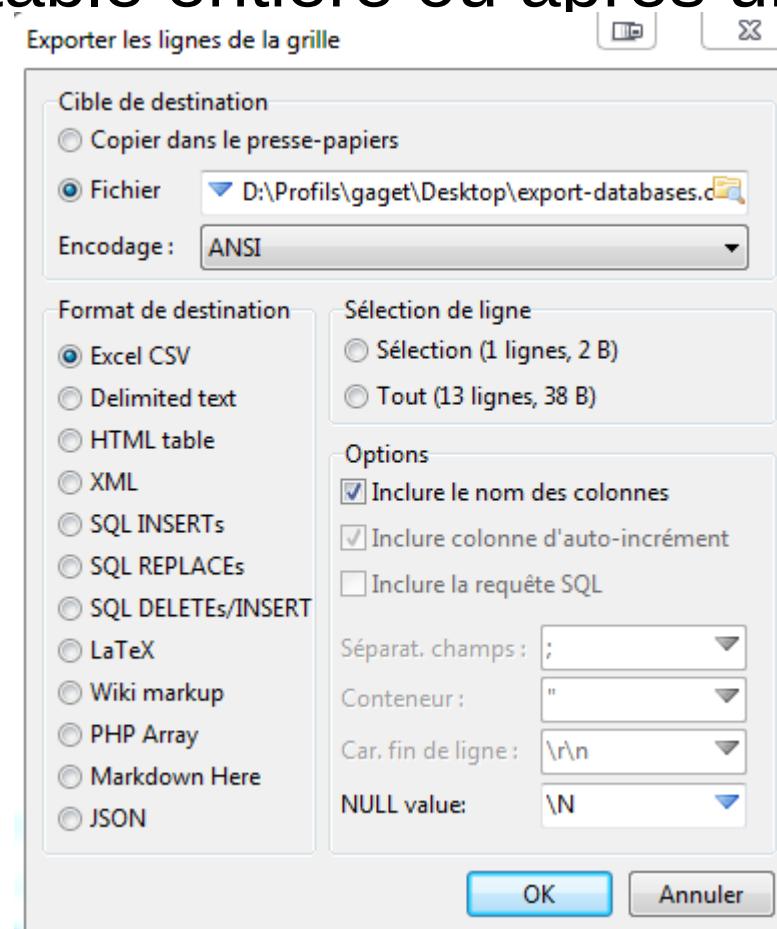
- **Avec la Commande \i :**
 - Création de la commande **COPY** ou **\copy** dans un fichier texte (extention .sql)
 - Utilisation de l'instruction psql **\i** suivi du nom (+chemin) du fichier .sql
 - Attention : mettre des " pour le fichier et les chemin (windows) avec des /
 - ex : **\i 'd:/MonProfile/scripts/export1.sql'**

D'un point de vue pratique ?

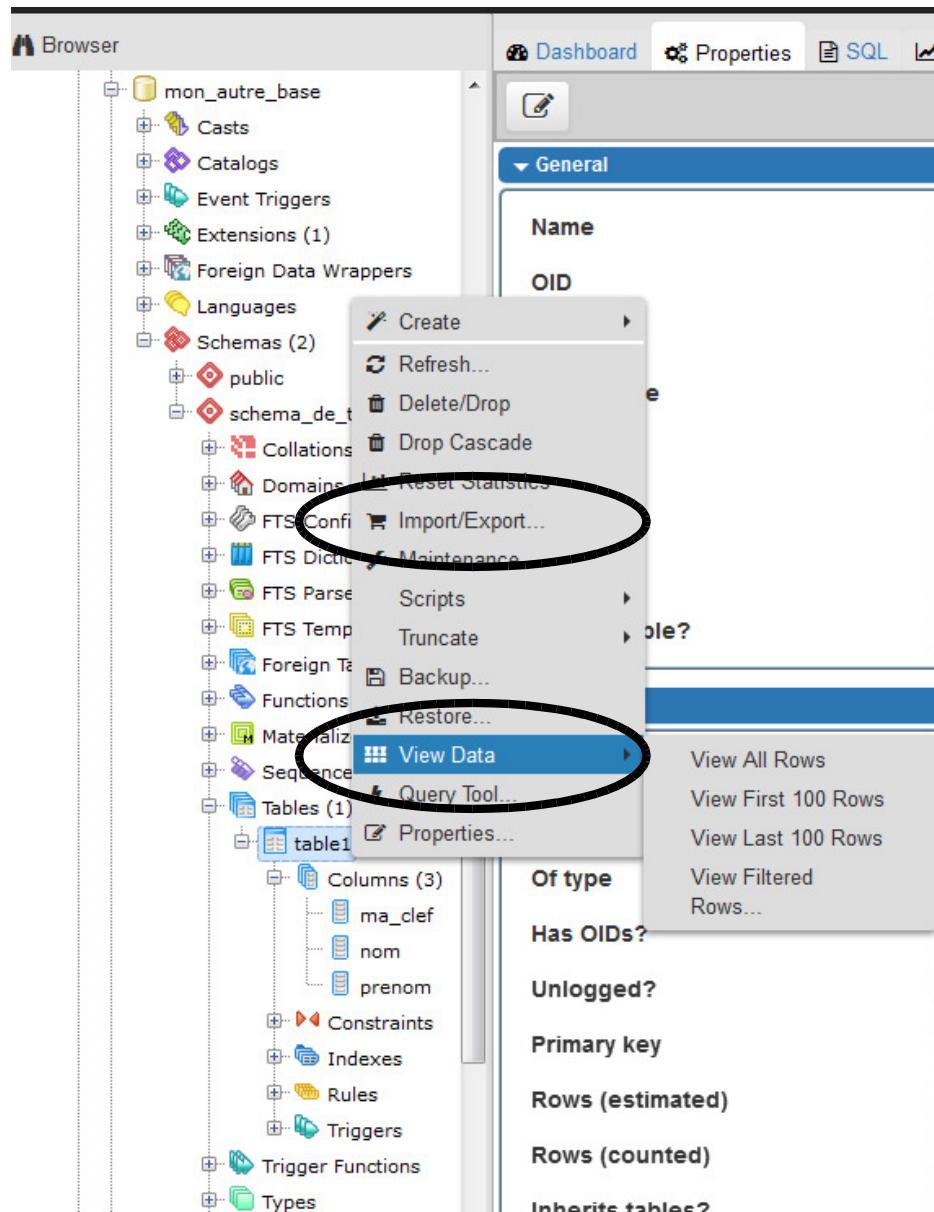
- Comment exporter des données ?
 - A la main, en ligne de commande
 - Par des scripts
 - **Avec un outil d'administration de base de données**

HeidiSQL

- L'outil **Exporter** dans HeidiSQL (Outils / Exporter les lignes de la grille ...)
- A partir de la table entière ou après une requête sql (select)



pgAdmin



pgAdmin

The screenshot shows the pgAdmin 4 interface. On the left is the Browser pane, which displays the database structure of a PostgreSQL 9.6 server. The 'mabase' database is selected, showing its contents: Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, and Schemas. The 'public' schema is expanded, showing Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Sequences, Tables (containing 'matable' and 'rrrr'), Trigger Functions, Types, and Views. Other databases listed are 'mon_autre_base' and 'postgres'. At the bottom of the Browser pane, there are 'Login/Group Roles' and 'Tablespaces (2)'. The main pane shows a query results grid for the 'matable' table. The SQL query is:

```
1 SELECT * FROM public.matable
2
```

The results grid has two columns: 'id' (integer) and 'value' (integer). The data is as follows:

	id	value
1	1	5
2	2	65
3	2	65
4	4	45
5	5	55
6	4	45
7	5	55
8	4	45
9	5	55
10	4	45
11	5	55

A black oval highlights the 'Download as CSV (F8)' button in the toolbar above the results grid.

pgAdmin

Import/Export data - table 'matable'

Options Columns

Import/Export Export

File Info

Filename

Format CSV

Encoding Select from the list

Miscellaneous

OID No

Header No

Delimiter :

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Quote "

Escape Select from list...

OK Cancel

Espaces Naturels

- Exporter au format csv, la liste des organismes s'occupant de la gestion d'un parc naturel transfrontalier

```
SELECT o.nom  
  
FROM type t, zone z, gerepar g, org o  
  
WHERE t.nom = 'PNT'  
  
AND t.id_type = z.id_type  
  
AND z.id_zone = g.id_zone  
  
AND g.id_org = o.id_org
```

Hôte : 127.0.0.1 | Base de données : public | Table : org | Données | Requête*

```
1 SELECT o.nom
2 FROM type t, zone z, gerepar g, org o
3 WHERE t.nom = 'PNT'
4 AND t.id_type = z.id_type
5 AND z.id_zone = g.id_zone
6 AND g.id_org = o.id_org
```

org (1×6)

nom
Europe
France
Belgique
Region HDF
Dept Nord
Region Wallone

ropole européenne de Lille' WHERE "id org"=7;

Exporter les lignes de la grille

Cible de destination

Copier dans le presse-papiers
 Fichier D:\Profils\gaget\Desktop\export-databases.csv

Encodage : ANSI

Format de destination

Excel CSV
 Delimited text
 HTML table
 XML
 SQL INSERTs
 SQL REPLACEs
 SQL DELETEs/INSERT
 LaTeX
 Wiki markup
 PHP Array
 Markdown Here
 JSON

Sélection de ligne

Sélection (0 lignes, 0 B)
 Tout (6 lignes, 53 B)

Options

Inclure le nom des colonnes
 Inclure colonne d'auto-incrémentation
 Inclure la requête SQL

Séparat. champs : ;

Conteneur : "

Car. fin de ligne : \r\n

NULL value: \N

OK Annuler

- Avec psql :

- \copy (SELECT o.nom FROM type t, zone z, gerepar g, org o WHERE t.nom = 'PNT' AND t.id_type = Z.id_type AND z.id_zone = g.id_zone AND g.id_org = o.id_org) TO 'D:/Profils/gaget/export-select.csv' DELIMITER ';' CSV HEADER;

The screenshot shows the pgAdmin 4 interface. On the left, the Browser pane displays a database structure under the schema 'espacenet2'. In the center, the main window contains a query tool with a SQL editor containing the following code:

```
1 SELECT o.nom
2 FROM type t, zone z, gerepar g, org o
3 WHERE t.nom = 'PNT'
4 AND t.id_type = z.id_type
5 AND z.id_zone = g.id_zone
6 AND g.id_org = o.id_org
```

The toolbar above the query editor includes several icons: a magnifying glass, a save icon, a refresh icon, a trash icon, a dropdown menu, a 'No limit' button, and a 'Download as CSV (F8)' button, which is highlighted with a blue arrow and circled in black.

APPLICATIONS

3 – Immeubles

Immeubles

On veut constituer une base de données d'immeubles en construction. Pour chacun de ces bâtiments, on connaît :

- le numéro de permis de construire
- le nom du bâtiment
- la ville dans laquelle se réalise le projet
- le porteur du projet de construction
- les architectes
- les entreprises qui participent au chantier

Voici la liste des bâtiments :

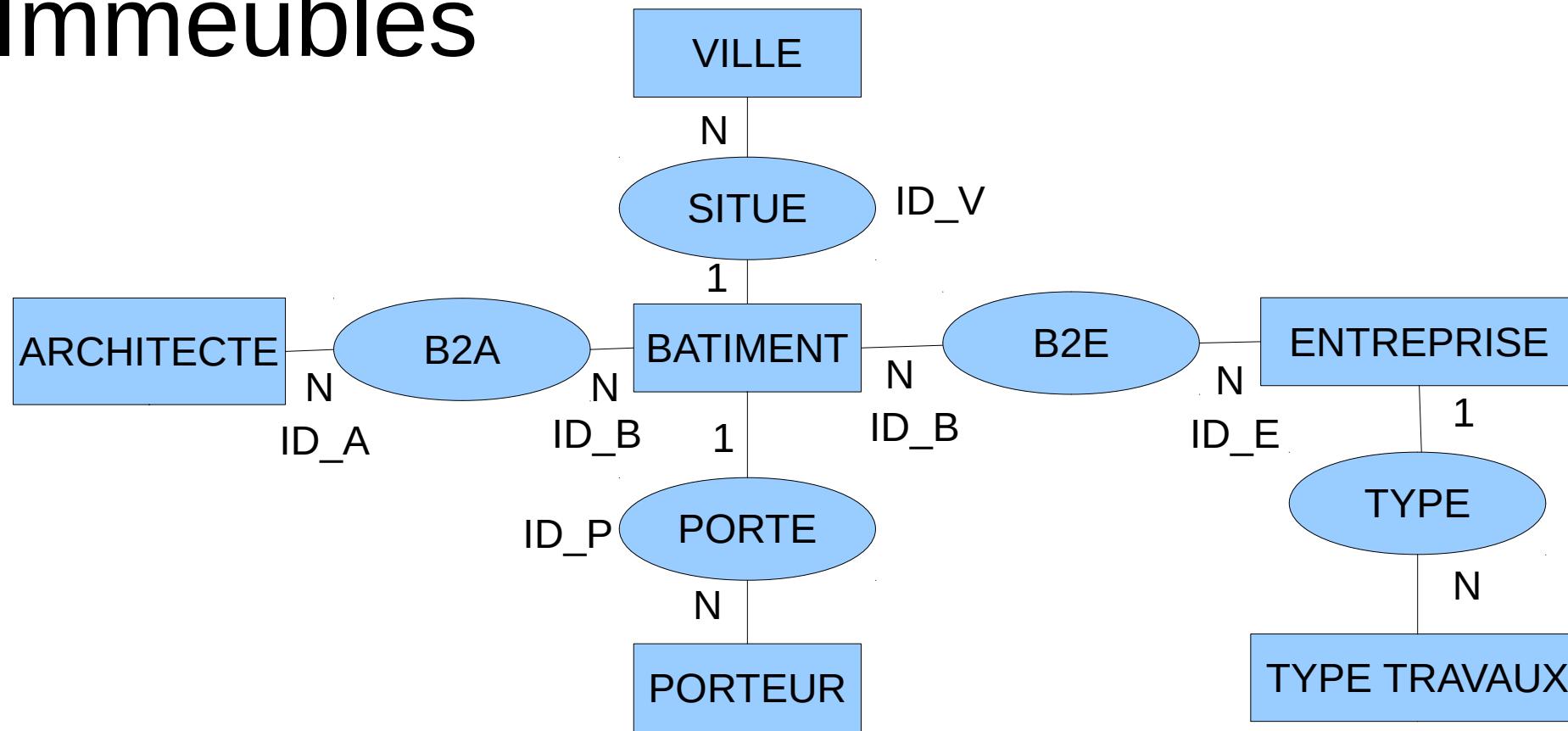
N° de permis	Nom du bâtiment	Ville	Porteur du projet	Architectes	Entreprises
45XV5	Résidence Camus	Lille	M. Dupont	Cabinet Lorem	SGEN (électricité), SGO (gros-œuvre), SPN (plomberie)
23GC4	Résidence Vinci	Douai	Mme Durand	Cabinet Sic	DOUELEC (électricité), SGO (gros-œuvre), PLOMBONOR (plomberie)
01SD6	Résidence Europe	Lille	M. Paul	Cabinet Ipsum Cabinet Lorem	DOUELEC (électricité), GROUVRE (gros-œuvre), SPMPLM (plomberie)
87PX9	Résidence Soleil	Arras	M. Dupont	Cabinet Sic	SGEN (électricité), SGO (gros-œuvre), PLOMBONOR (plomberie)

Construisez le modèle entité-relation

Vous organiserez au mieux vos données pour éviter les redondances d'informations et faciliter les recoupements d'informations...

Indiquez pour chacune des entités et des relations la liste des attributs

Immeubles



- **VILLE(ID_V, NOM)**
- **ARCH(ID_A, NOM)**
- **PORTEUR(ID_P, NOM)**
- **TT(ID_TT, NOM)**
- **ENTR(ID_E, NOM, ID_TT)**
- **BAT(ID_B, NOM, ID_P, ID_V, PERMIS)**
- **B2A(ID_B, ID_A)**
- **B2E(ID_B, ID_E)**

Immeubles

Écrivez en SQL les commandes de création des tables et insertions de quelques une des données

Écrivez en SQL les requêtes donnant les résultats ci-dessous

- Liste des porteurs de projets
- Liste des bâtiments en construction à Lille
- Liste des architectes travaillant sur un projet sur lequel intervient la société SGEN
- Liste des entreprises d'électricité intervenant sur des projets à Lille
- Nombre de bâtiments en construction à Lille
- Nombre de bâtiments en construction par ville

Immeubles

- Créez la base ***immeubles*** pour l'utilisateur ***tom***
- Créez les tables et insérez-les données à partir des instructions sql du fichier « immeubles.sql »
- Créez les requêtes demandées et exécutez-les

APPLICATIONS

2 – Étude de lignes ferroviaires

Lignes ferroviaires

On veut constituer une base de données avec les lignes ferroviaires de la région. Chaque ligne est repérée par un numéro.

On souhaite associer des informations complémentaires à ces lignes :

- nom de la ligne (exemples : "Lille-Dunkerque", "Lille-Calais-Londres"...)
- type de la ligne (exemples : "TER", "TGV"...)
- liste des gares desservies par la ligne, et pour chaque gare, la ville dans laquelle elle se trouve

Voici la liste des lignes (les listes de gares sont volontairement non exhaustives) :

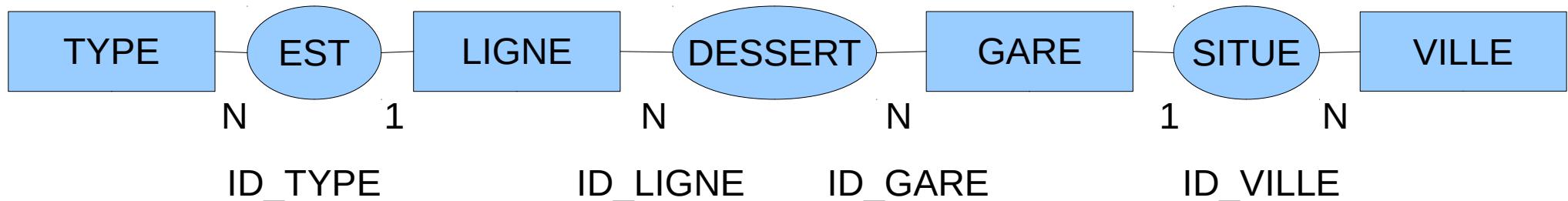
N°	Nom	Type	Gares (Ville)
08	Lille-Dunkerque	TER	Lille Flandres (Lille), Armentières (Armentières), Hazebrouck (Hazebrouck), Dunkerque (Dunkerque)
101	Lille-Paris	TGV	Lille Flandres (Lille), Paris Nord (Paris)
02	Lille-Douai-Arras-Amiens-Rouen	TER	Lille Flandres (Lille), Douai (Douai), Arras (Arras), Amiens (Amiens), Rouen Rive Droite (Rouen)
201	Lille-Calais-Londres	Eurostar	Lille Europe (Lille), Calais Fréthun (Fréthun), Londres Waterloo (Londres)
12	Lille-Hazebrouck-Calais-Boulogne	TER	Lille Flandres (Lille), Armentières (Armentières), Hazebrouck (Hazebrouck), Calais Ville (Calais), Calais Fréthun (Fréthun), Boulogne Tintelleries (Boulogne-sur-Mer), Boulogne Ville (Boulogne-sur-Mer)

Construisez le modèle entité-relation

Vous organiserez au mieux vos données pour éviter les redondances d'informations et faciliter les recoupements d'informations...

Indiquez pour chacune des entités et des relations la liste des attributs

Lignes ferroviaires



- TYPE(ID_TYPE, NOM)
- VILLE(ID_VILLE, NOM)
- LIGNE(ID_LIGNE, NOM, NUMERO, ID_TYPE)
- GARE(ID_GARE, NOM, ID_VILLE)
- DESSERT(ID_LIGNE, ID_GARE)

Lignes ferroviaires

- Créez la base ***lignes_ferroviaires*** pour l'utilisateur ***tom***
- Créez les tables et insérez-les données à partir des instructions sql du fichier « lignes_ferroviaires.sql »
- Créez les requêtes demandées et exécutez-les

