**Migration de base de données d’un SGDB vers un autre**

*Un SGBD (Système de Gestion de Base de Données) est un ensemble de logiciels qui sert à la manipulation des bases de données. Il sert à effectuer des opérations ordinaires telles que consulter, modifier, construire, organiser, transformer, copier, sauvegarder ou restaurer des bases de données.*

*Il existe plusieurs SGBD (postgres, mysql, oracle, sqlite, access, …) et chacun a sa syntaxe pour la connexion, la gestion des données, … donc on ne peut pas passer directement d’une base de données vers une autre.*

*Notre objectif est de mettre en place un outil de migration des schémas et des données d’une BD de n’importe quel SGBD vers une autre.*

1. Analyse
2. Mysql
   1. Les types des attributs

|  |  |
| --- | --- |
| **type** | **Description (permet de définir…)** |
| TINYINT | un entier de -128 à 127 |
| TINYINT UNSIGNED | un entier de 0 à 255 |
| SMALLINT | un entier de -32768 à 32767 |
| SMALLINT UNSIGNED | un entier de 0 à 65665 |
| MEDIUMINT | un entier de  -8388608 à 8388607 |
| MEDIUMINT UNSIGNED | un entier de  0 à 16777215 |
| INTEEGER | un entier de -2147483648 à 2147483647 |
| INT EGER UNSIGNED | un entier de 0 à 4294967295 |
| BIGINT | un entier de  -9223372036854775808 à 9223372036854775807 |
| BIGINT UNSIGNED | un entier de  0 à 18446744073709551615 |
| FLOAT | un entier de  -3.402823466E+38 à 1.175494351E-38 |
| DOUBLE | un entier de -1.7976931348623157E+308 à 2.2250738585072014E-308 |
| CHAR(M) | une chaîne de taille fixée à M, où 1<M<255, complétée avec des espaces si nécessaire. |
| CHAR(M) BINARY | Idem, mais insensible à la casse lors des tris et recherches |
| VARCHAR(M) | une chaîne de taille variable, de taille maximum M, où 1<M<255, complété avec des espaces si nécessaire. |
| VARCHAR(M) BINARY | Idem, mais insensible à la casse lors des tris et recherches. |
| TINYTEXT | Longueur maximale de 255 caractères. |
| TEXT | Longueur maximale de 65535 caractères. |
| MEDIUMTEXT | Longueur maximale de 16777215 caractères. |
| LONGTEXT | Longueur maximale de 4294967295 caractères. |
| DECIMAL(M,D) | Simule un nombre flottant de D chiffres après la virgule et de M chiffres au maximum. Chaque chiffre ainsi que la virgule et le signe moins (pas le plus) occupe un caractère. |
| DATE | Date au format anglophone AAAA-MM-JJ. |
| DATETIME | Date et heure au format anglophone AAAA-MM-JJ HH:MM:SS. |
| TIMESTAMP | Affiche la date et l’heure sans séparateur : AAAAMMJJHHMMSS. |
| TIMESTAMP(M) | Idem mais M vaut un entier pair entre 2 et 14. Affiche les M premiers caractères de TIMESTAMP. |
| TIME | Heure au format HH:MM:SS. |
| YEAR | Année au format AAAA. |

* 1. Les requêtes SQL

|  |  |  |  |
| --- | --- | --- | --- |
| **Type instruction** | **Instruction** | **Description** | **Exemple** |
| Creation base de données | Create Database <nom\_base> | Permet de créer une base de données de nom <nom\_base> | Create database MaBase; |
| Création de table | Create table <nom\_table>(<nom\_champ> <type\_attr>…) | Permet de créer une table avec comme champ <nom\_champ> de type <type\_attr>. On peut créer plusieurs champ | Create table personne(nom varchar(20), prenom varchar(40)… |
| Listing de table | Show tables ; | Permet de lister les tables d’une base de données | Show tables ; \*apres s’etre connecté à la base de données |
| Listing des bdd | Show databases | Permet de lister les bases de données | Show databases ; |
| Insertion données | Insert into <table> valuies(<valeur1>, …) | Permet d’inserer des données dans la table <table> | Insert into personne(« diop », « abdoulaye ») ; |
| Spécification des clés primaires | Primary key | Permet d’avoir un enregistrement à partir d’une des attributs composant la clé primaire | Primary key(<attr1>, <attr2> |

1. Postgres
   1. Les types des attributs

|  |  |  |
| --- | --- | --- |
| **Nom** | **Alias** | **Description** |
| bigint | int8 | Entier signé sur 8 octets |
| bigserial | serial8 | Entier sur 8 octets à incrémentation automatique |
| bit [ (n) ] |  | Suite de bits de longueur fixe |
| bit varying [ (n) ] | varbit | Suite de bits de longueur variable |
| boolean | bool | Booléen (Vrai/Faux) |
| character varying [ (n) ] | varchar [ (n) ] | Chaîne de caractères de longueur variable |
| character [ (n) ] | char [ (n) ] | Chaîne de caractères de longueur fixe |
| date |  | Date du calendrier (année, mois, jour) |
| double precision | float8 | Nombre à virgule flottante de double précision (sur huit octets) |
| integer | int, int4 | Entier signé sur 4 octets |
| numeric [ (p, s) ] | decimal [ (p, s) ] | Nombre exact dont la précision peut être précisée |
| real | float4 | Nombre à virgule flottante de simple précision (sur quatre octets) |
| smallint | int2 | Entier signé sur 2 octets |
| serial | serial4 | Entier sur 4 octets à incrémentation automatique |
| text |  | Chaîne de caractères de longueur variable |
| time [ (p) ] [ without time zone ] |  | Heure du jour (pas du fuseau horaire) |
| time [ (p) ] with time zone | timetz | Heure du jour, avec fuseau horaire |
| timestamp [ (p) ] [ without time zone ] |  | Date et heure (pas du fuseau horaire) |
| timestamp [ (p) with time zone | timestamptz | Date et heure, avec fuseau horaire |
| txid\_snapshot |  | image de l'identifiant de transaction au niveau utilisateur |
| xml |  | données XML |
| decimal |  | no limit |

* 1. Les requêtes SQL

|  |  |  |  |
| --- | --- | --- | --- |
| **Type instruction** | **Instruction** | **Description** | **Exemple** |
| Création base de données | CREATE DATABASE <nom\_base> | Permet de créer une base de données de nom <nom\_base> | CREATE DATABASE MaBase; |
| Création de table | CREATE TABLE <nom\_table>(<nom\_champ> <type\_attr> NULL?…) | Permet de créer une table avec comme champ <nom\_champ> de type <type\_attr>. On peut créer plusieurs champ | CREATE TABLE products (    product\_no integer,    name text,    price numeric ); |
| Listing de table | SELECT tablename FROM pg\_tables WHERE tablename !~ '^pg\_' AND tablename !~ '^sql\_' | Permet de lister les tables d’une base de données | Show tables ; \*apres s’etre connecté à la base de données |
| Listing des bdd | select datname from pg\_database | Permet de lister les bases de données | Show databases ; |
| Insertion données | INSERT INTO <table> VALUES (<valeur1>, …) | Permet d’inserer des données dans la table <table> | INSERT INTO products VALUES (1, 'Cheese', 9.99); |
| Spécification des clés primaires | serial | Permet d’avoir un enregistrement à partir d’une des attributs composant la clé primaire | CREATE TABLE personnes (  id serial NOT NULL,  nom character varying(25) NOT NULL DEFAULT 'camara'::character varying ) |
| Récupération de toutes les données d'une table | SELECT \* FROM <nom\_table>; | Permet de récupérer les données de la table nom\_table | SELECT \* FROM table1; |

1. Implémentation
2. Déploiement (sous ubuntu)
3. Packet à installer

* Installer ruby 1.8+ : sudo apt-get install ruby1.8.7
* Installer le packet ruby-dev : sudo apt-get install ruby-dev
* Installer rubygems (<http://rubygems.org/>)
* Installer le paquet libmysqlclient-dev : sudo apt-get install libmysqlclient-dev
* Installer la gem mysql2 : sudo gem install mysql2
* Installer la gem postgres : sudo gem install postgres
* Installer la gem sinatra : sudo gem install sinatra

1. Lancement de l’application

* Ouvrir le terminal
* Se déplacer jusqu’au répertoire /lib du projet
* Lancer l’application avec la commande : ruby main.rb
* Ensuite vous ouvrez votre navigateur : http://localhost:4567