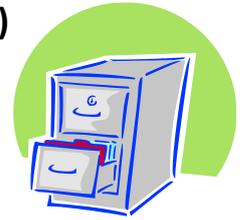




## DECOUVERTE DE SQL SERVEUR (STRUCTURED QUERY LANGUAGE)



### Préambule

Ce TP n'a pour autre objectif que de vous faire découvrir ou **redécouvrir** le langage SQL, langage structuré de requêtes, langage standard et normalisé, destiné à interroger ou à manipuler une base de données relationnelle.

Par conséquent ce TP n'est pas un support théorique, mais une première approche du SERVEUR SQL, au travers de l'interface SQL MANAGEMENT STUDIO et d'un exemple simple.

### Le Langage SQL

Le langage SQL est donc un langage de requêtage qui permet de créer, modifier, interroger, contrôler, traiter une base de données.

Le langage SQL peut se décomposer en 5 parties :

- Le Langage de **D**éfinition de **D**onnées (LDD) : permet de créer et modifier la structure de la base de données
- Le Langage de **M**anipulation de **D**onnées (LMD) : permet d'interroger (consulter) et mettre à jour le contenu de la base de données, donc les données elles-mêmes : ajout, suppression, modification.
  - Le sous-ensemble de commandes qui permet d'interroger la base de données est parfois appelé LID pour **L**angage d'**I**nterrogation de **D**onnées.
- Le Langage de **C**ontrôle de **D**onnées (LCD) : permet de gérer les privilèges, c'est-à-dire les utilisateurs et les actions qu'ils peuvent entreprendre
- Le Langage de **C**ontrôle des **T**ransactions (LCT) : permet de gérer les transactions, une transaction étant un ensemble de commandes indissociables qui doivent pouvoir s'exécuter globalement sans interruption ou pas du tout (explication simplifiée)
- Le Langage **P**rocédural : ensemble d'outils pour développer des procédures, déclencheurs (triggers) et fonctions utilisateurs, pour permettre au SQL de s'interfacer avec des langages de programmation.

Découvrons maintenant l'outil que nous allons utiliser, à savoir SQL SERVEUR MANAGEMENT STUDIO connecté au serveur de Base de données du lycée.

## Présentation de la base de données

Nous allons créer une base de données, dotée initialement d'une seule table. Cette table contiendra les Présidents de la République Française.

ID	Nom	Prenom	Nb Mandats	DebMandats	FinMandats	Naissance	Mort	Parti	Republique
1	BONAPARTE	Louis-Napoléon	1	20/12/1848	02/12/1852	20/04/1808	09/01/1873	Bonapartiste	2
2	THIERS	Adolphe	1	31/08/1871	24/05/1873	15/04/1797	03/09/1877	Orléaniste	3
3	DE MAC MAHON	Patrice	1	24/05/1873	30/01/1879	13/07/1808	08/10/1893	Légitimiste	3
4	GREVY	Jules	2	30/01/1879	02/12/1887	15/08/1807	09/09/1891	Modéré	3
5	CARNOT	Sadi	1	03/12/1887	25/06/1894	11/08/1837	25/06/1894	Modéré	3
6	CASIMIR-PERIER	Jean	1	27/06/1894	16/01/1895	08/11/1847	11/03/1907	Modéré	3
7	FAURE	Félix	1	17/01/1895	16/02/1899	30/01/1841	16/02/1899	Modéré	3
8	LOUBET	Emile	1	18/02/1899	18/02/1906	30/12/1838	20/12/1929	ARD	3
9	FALLIERES	Armand	1	18/02/1906	18/02/1913	06/11/1841	22/06/1931	ARD – PRD	3
10	POINCARRE	Raymond	1	18/02/1913	18/02/1920	20/08/1860	15/10/1934	PRD – ARD	3
11	DESCHANEL	Paul	1	18/02/1920	21/09/1920	13/02/1855	28/04/1922	ARD – PRDS	3
12	MILLERAND	Alexandre	1	23/09/1920	11/06/1924	10/02/1859	07/04/1943	sans étiquette	3
13	DOUMERGUE	Gaston	1	13/06/1924	13/06/1931	01/08/1863	18/06/1937	Radical	3
14	DOUMER	Paul	1	13/06/1931	07/05/1932	22/03/1857	07/05/1932	Radical	3
15	LEBRUN	Albert	2	10/05/1932	11/07/1940	29/08/1871	06/03/1950	AD	3
16	AURIOL	Vincent	1	16/01/1947	16/01/1954	27/08/1884	01/01/1966	SFIO	4
17	COTY	René	1	16/01/1954	08/01/1959	20/03/1882	22/11/1962	CNIP	4
18	DE GAULLE	Charles	2	08/01/1959	28/04/1969	22/11/1890	09/11/1970	UNR – UDR	5
19	POMPIDOU	Georges	1	20/06/1969	02/04/1974	05/07/1911	02/04/1974	UDR	5
20	GISCARD D'ESTAING	Valéry	1	27/05/1974	21/05/1981	02/02/1926		FNRI – UDF	5
21	MITTERAND	François	2	21/05/1981	17/05/1995	26/10/1916	08/01/1996	PS	5
22	CHIRAC	Jacques	2	17/05/1995	16/05/2007	29/11/1932		RPR – UMP	5
23	SARKOZY	Nicolas	1	16/05/2007	16/05/2102	28/01/1955		UMP	5
24	HOLLANDE	François	1	16/05/2102	07/05/2012	12/08/1954		PS	5
25	MACRON	Emmanuel	1	07/05/2017		21/12/1977		EN MARCHE	5

## Présentation de l'outil de requêtage

Nous utiliserons le Système de Gestion de Bases de Données Relationnel (SGBDR) SQL SERVEUR de MICROSOFT.

Pour créer/modifier/utiliser une base de données sur ces serveurs, nous utiliserons Microsoft SQL Server Management Studio. C'est un outil de gestion graphique convivial et gratuit qui permet de gérer SQL.

 Votre login et votre mot de passe doivent être pris en charge par le serveur, cliquez sur « Se connecter ».

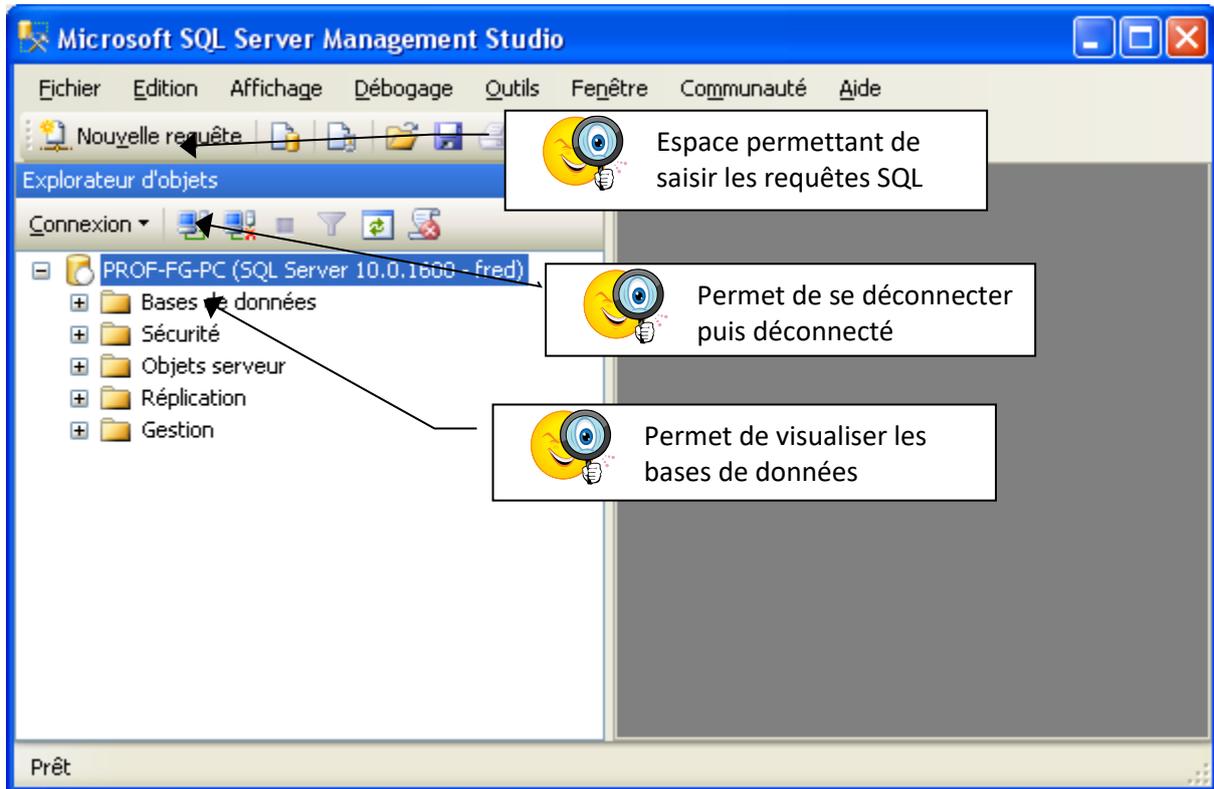
## Remarque :

Il y a deux types de connexion à Microsoft SQL Server.

- le mode **authentification Windows** : il permet de laisser au système la tâche d'authentifier le client. Avantage : le mot de passe ne passe pas au travers des paquets TDS et ne peut donc être "sniffé".
- le mode **authentification SQL** avec un mot de passe et un login propre au SQL server, sans lien avec le système d'exploitation. Avantage : une distinction claire et nette entre les droits sur la base et ceux sur le système d'exploitation.

## Prise en main de l'interface

Une fois connecté, vous devriez obtenir l'interface suivante :



### Création d'une base de données et d'une table – Présentation

Nous allons utiliser cette interface pour créer la table qui contiendra les différents Présidents de la République Française. Pour effectuer cette première mission, nous re-découvrons donc la commande **CREATE TABLE** qui permet comme vous vous en doutez de créer une table (au sens du modèle relationnel, tel que vous l'avez appris).

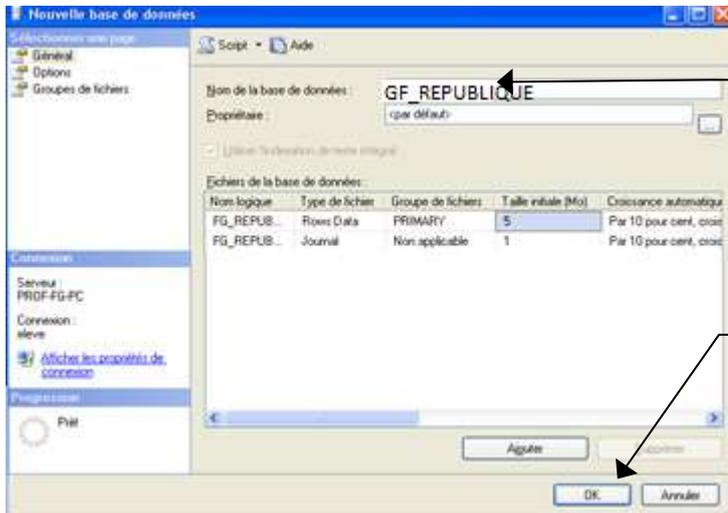
Vous allez créer une table, en définissant sa structure, c'est-à-dire la liste des colonnes qui la composent, avec pour chaque colonne un identificateur, un type, éventuellement une contrainte.

Pour l'instant, je vous donne la commande, avec quelques explications. Vous vous contentez de la comprendre, de la reproduire, et de vérifier qu'elle s'exécute correctement.

### Création effective de la BASE DE DONNES : GF\_REPUBLIQUE

#### Version avec interface :

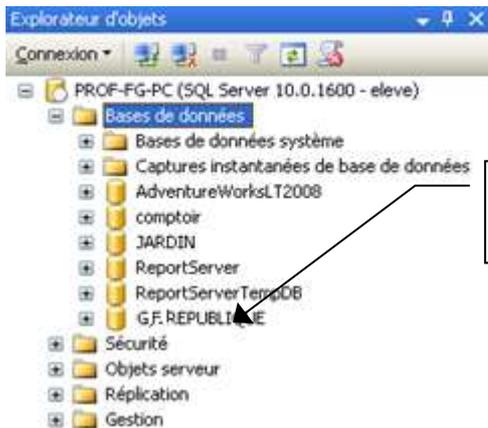
NB : Créer la base de données REPUBLIQUE, **préfixée de vos initiales NOM + PRENOM**, à l'aide d'un clic droit sur Base de données puis Nouvelle base de données et valider par [OK]



 Noter le nom de la BDD

 Valider

La base est créée dans la liste des bases de données.



 La base de données créée

## Version SQL :

```
CREATE DATABASE [FG_PRESIDENT] ON PRIMARY
( NAME = N'test', FILENAME = N'C:\SQL_DATA\DBFILE\GF_REPUBLIQUE.mdf' , SIZE = 5120KB
, FILEGROWTH = 10%) \ il s'agit du chemin sur le serveur
LOG ON
( NAME = N'test_log', FILENAME = N'C:\SQL_DATA\DBLOG\GF_REPUBLIQUE.ldf' , SIZE =
1024KB , FILEGROWTH = 10%)
USE [test] ;
```

## Architecture des fichiers et des groupes de fichiers SQL Server

SQL Server mappe une base de données sur un ensemble de fichiers du système d'exploitation. Les données et les informations des journaux ne figurent jamais dans le même fichier ; par ailleurs, les fichiers spécifiques ne sont utilisés que par une base de données.

Les bases de données SQL Server disposent de trois types de fichiers :

- Fichiers de données primaires  
Le fichier de données primaire constitue le point de départ de la base de données et il pointe vers les autres fichiers de la base de données. Chaque base de données comprend un fichier de données primaire. L'extension de fichier recommandée est **.mdf**
- Fichiers journaux  
Ces fichiers contiennent toutes les informations de suivi nécessaires à la récupération de la base de données. Il doit exister au moins un fichier journal par base de données. L'extension de fichier recommandée est **.ldf**

## Création effective de la table PRESIDENT

NB : Il est préférable de la taper avant dans un éditeur pour effectuer les tabulations correctement et ainsi faciliter la relecture de la requête, et donc minimiser les erreurs :

☞ Dans un Editeur (Ex : Notepad++), taper la commande complète permettant de créer la table PRESIDENT :

RQ : Si on enregistre le fichier avec l'extension « .sql », NotePad++ effectue une coloration syntaxique.

```
1 USE FG_REPUBLIQUE;
2 CREATE TABLE PRESIDENT (
3     ID          INTEGER PRIMARY KEY
4     Nom         VARCHAR(20) NOT NULL,
5     Prenom      VARCHAR(15) NOT NULL,
6     NbMandats   INTEGER,
7     DebMandats  DATE,
8     FinMandats  DATE,
9     Naissance   DATE,
10    Mort        DATE,
11    Parti       VARCHAR(20),
12    Republique  INTEGER
13 );
```

Annotations:

- Indiquer le nom de la base de données à utiliser (pointe vers 'FG\_REPUBLIQUE')
- Une parenthèse ouvrante pour commencer la description des colonnes (pointe vers '(')
- Une virgule après chaque description de colonne... (pointe vers virgules)
- ... excepté après le dernier champ :  
⇒ pas de virgule (pointe vers la fin de la liste de colonnes)
- Une parenthèse pour terminer la description des colonnes, et un point-virgule pour terminer l'instruction SQL (pointe vers ');')

☞ Ensuite Copier/Coller la requête dans l'espace de travail « SQL-STUDIO » :

Annotations:

- PRIMARY KEY Définit la clé primaire (pointe vers 'PRIMARY KEY')
- Contrainte éventuelle associée au champ (pointe vers 'NOT NULL')
- NOT NULL Contraint de renseigner le champ (pointe vers 'NOT NULL')
- Type des champs (cf. Explications) (pointe vers 'INTEGER', 'VARCHAR(20)', 'DATE')
- Nom des champs (pointe vers 'ID', 'Nom', 'Prenom')

☞ Lire et comprendre les explications ci-dessus, puis exécuter la commande en cliquant sur le bouton « Exécuter ».

### Petite explication complémentaire sur la syntaxe

Ci-dessus, la commande SQL avait été tapée dans un éditeur de texte avec des tabulations pour bien montrer les différentes parties qui définissent chaque colonne : le nom du champ, le type du champ, la contrainte éventuelle, puis une virgule. En fait un espace suffit pour séparer nom et type, et type et contrainte.

Noter également que le retour à la ligne n'est pas obligatoire syntaxiquement, mais facilite la lecture de la requête et **SERA DONC OBLIGATOIRE DANS L'ECRITURE DE REQUETE EN COURS OU EN TP.**

## Explications sur les types

---

NB : Nous nous contentons d'expliquer uniquement les types utilisés ci-dessous, mais il y en a d'autres.

<b>INTEGER</b>	Entier
<b>VARCHAR(N)</b>	Chaîne de caractères de longueur variable, mais dont N est la longueur maximal ( $N \leq 2000$ ) RQ : Le type VARCHAR existe, mais c'est un type bogué. Utiliser toujours VARCHAR2.
<b>DATE</b>	Champ de type date

## ERRARE HUMANUM EST

---



En cas d'erreur dans la création de la table, il est possible de :

- Modifier la structure de la table : nous apprendrons comment faire ultérieurement.
- Détruire la table et la recréer correctement. C'est souvent plus facile et plus rapide à faire, surtout si on a conservé le script de création.

Voici donc la commande qui permet de supprimer – en cas de besoin – la table PRESIDENT :

```
DROP TABLE PRESIDENT ;
```

⇒ Supprime complètement la table : contenu et structure !



## Afficher la STRUCTURE d'une TABLE

---

Pour afficher la structure de la table, autrement dit pour visualiser sa **description**, taper la commande suivante :

```
EXEC sp_columns PRESIDENT ;
```

“sp\_columns” est une [procédure stockée](#) SQL



Nous allons maintenant voir comment remplir les tables, et éventuellement modifier leur contenu.

## UN CONSEIL avant de continuer

---

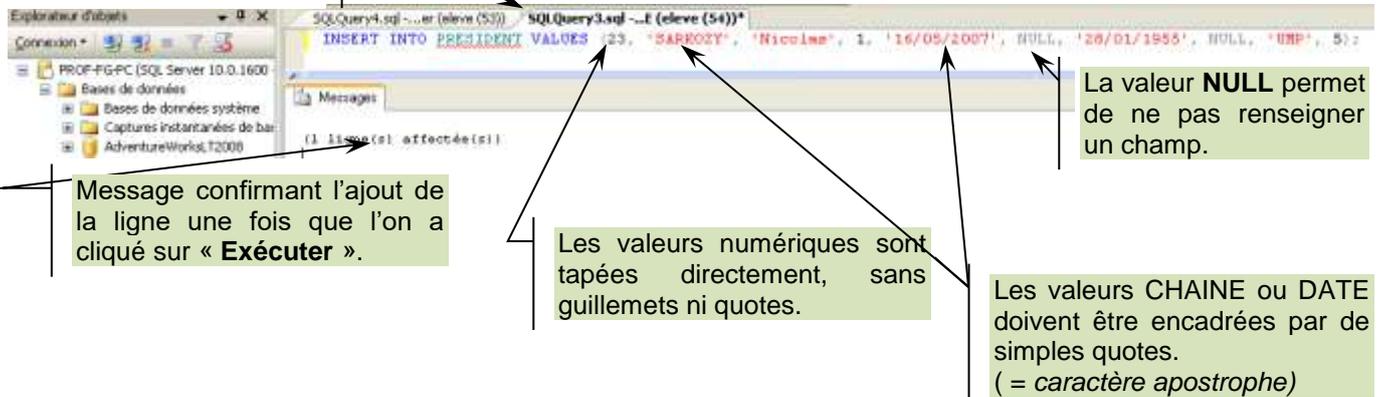


Lorsque l'on crée une base, ou un jeu d'essai, ou surtout quand on fait un TP pour apprendre le SQL, il est vivement conseillé de conserver dans un fichier l'ensemble des scripts qui ont réussi, de manière à être capable de recréer rapidement la même base, le même jeu d'essai, sur un autre serveur.

## Insertion de tuples (lignes) dans la table PRESIDENT

La commande INSERT INTO va nous permettre d'insérer une première ligne dans la table, par exemple celle concernant notre avant avant dernier président: **Nicolas SARKOSY**

Feuille de travail où sont stockées les requêtes, possibilité d'enregistrer



La valeur **NULL** permet de ne pas renseigner un champ.

Les valeurs numériques sont tapées directement, sans guillemets ni quotes.

Les valeurs CHAINE ou DATE doivent être encadrées par de simples quotes. (= caractère apostrophe)

Message confirmant l'ajout de la ligne une fois que l'on a cliqué sur « Exécuter ».

Faites de même pour les autres présidents de la 5<sup>ème</sup> république (les 6 derniers de la liste), en faisant quelques erreurs :

```
INSERT INTO PRESIDENT
VALUES (23, 'CHIRAC', Jacques, 2, '17/05/1995', '16/05/2007', '29/11/1932', NULL, 'RPR – UMP', 5);
```

Quel est le message retourné suite à l'erreur de saisie (pas de quotes autour du prénom) ?

```
INSERT INTO PRESIDENT
VALUES (23, 'CHIRAC', Jacques, 2, '17/05/1995', '16/05/2007', '29/11/1932', NULL, 'RPR – UMP', 5);
```

Quel est le message retourné suite à l'erreur de saisie (le même ID que pour SARKOZY) ?

```
INSERT INTO PRESIDENT
VALUES (24, 'CHIRAC', Jacques', 2, '17/05/1995', '16/05/2007', '29/11/1932', NULL, 'RPR – UMP', 5);
```

Quel est le message retourné ?

## ERRARE HUMANUM EST



Mince alors, je me suis trompé de valeur d'ID ! C'était le 22<sup>ème</sup> et pas le 24<sup>ème</sup>. Eh oui, CHIRAC était président avant SARKOZY et ne le sera probablement pas après ;-)

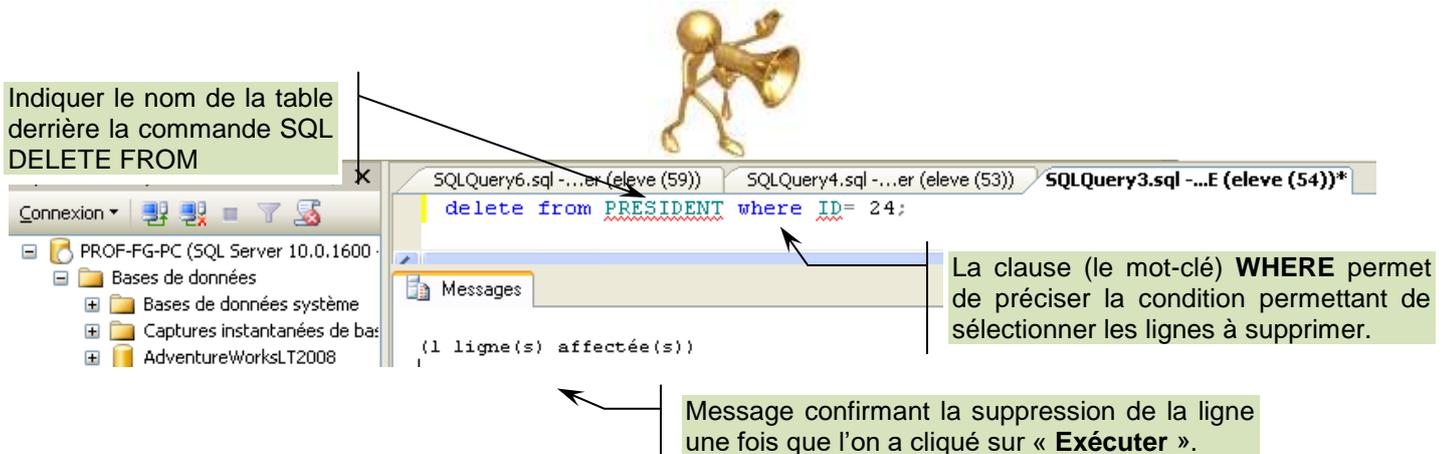
Apprenons à supprimer une ligne...

## ATTENTION !

*Si on ne spécifie pas de condition, toutes les lignes de la table seront supprimées !*

## SUPPRESSION DE LIGNE

La commande DELETE FROM permet de supprimer une (ou plusieurs lignes) :



Indiquer le nom de la table derrière la commande SQL DELETE FROM

```
delete from PRESIDENT where ID= 24;
```

La clause (le mot-clé) **WHERE** permet de préciser la condition permettant de sélectionner les lignes à supprimer.

Message confirmant la suppression de la ligne une fois que l'on a cliqué sur « Exécuter ».

{1 ligne(s) affectée(s)}

Taper maintenant la commande INSERT correcte :

```
INSERT INTO PRESIDENT  
VALUES (22, 'CHIRAC', 'Jacques', 2, '17/05/1995', '16/05/2007', '29/11/1932', '26/09/2019', 'RPR – UMP', 5);
```

Faire encore une erreur pour le président « François MITTERAND » :

```
INSERT INTO PRESIDENT  
VALUES (21, 'MITTERAND', 'Françoise', 2, '21/05/1981', '17/05/1995', '26/10/1916', '08/01/1996', 'PS', 5);
```

Quel est le message retourné ?



En fait il n'y a pas de souci pour ajouter cette ligne, mais nous avons fait une erreur (involontaire ;- ) sur le prénom...

Comment la corriger ?

## MODIFICATION DE LIGNE

La commande UPDATE permet de modifier une (ou plusieurs lignes) :

Indiquer le nom de la table concernée derrière la commande SQL :

**UPDATE**



**ATTENTION !**  
Si On ne spécifie pas de condition, toutes les lignes de la table seront modifiées et tous les présidents se prénommeront « François » !

Espace de travail

Saisissez les instructions SQL, PL/SQL et SQL\*Plus.

```
UPDATE PRESIDENT
SET Prenom='François'
WHERE ID=21;
```

La clause (le mot-clé) **SET** permet de préciser la modification à effectuer : le champ et la nouvelle valeur.  
Si plusieurs champs sont concernés, séparer les modifications à opérer par des virgules.

La clause (le mot-clé) **WHERE** permet de préciser la condition permettant de sélectionner les lignes à modifier.

Exécuter

Charger sc

1 ligne mise à jour.

Message confirmant le nombre de lignes concernées par la mise à jour, une fois que l'on a cliqué sur « Exécuter ».

Poursuivre les insertions pour Messieurs GISCARD, POMPIDOU et DE GAULLE :

```
INSERT INTO PRESIDENT VALUES
(18, 'DE GAULLE', 'Charles', 2, '08/01/1959', '28/04/1969', '22/11/1890', '09/11/1970', 'UNR - UDR', 5);

INSERT INTO PRESIDENT VALUES
(19, 'POMPIDOU', 'Georges', 1, '20/06/1969', '02/04/1974', '05/07/1911', '02/04/1974', 'UDR', 5);

INSERT INTO PRESIDENT VALUES
(20, 'GISCARD D'ESTAING', 'Valéry', 1, '27/05/1974', '21/05/1981', '02/02/1926', NULL, 'FNRI - UDF', 5);
```

Doubler l'apostrophe ici ! ' + '

Et encore SARKOZY et HOLLANDE :

- INSERT INTO PRESIDENT  
VALUES (24, 'HOLLANDE', 'François', 1, '16/05/2012', '07/05/2017', '12/08/1954', NULL, 'PS', 5);
- INSERT INTO PRESIDENT  
VALUES (25, 'MACRON', 'Emmanuel', 1, '07/05/2017', NULL, '21/12/1977', NULL, 'EN MARCHE', 5);



Pour ce dernier ajout, il faut doubler la quote du nom « D'ESTAING » (2 fois l'apostrophe). En effet, du fait qu'en SQL une chaîne est encadrée de quotes simples, une quote simple dans une chaîne provoque une erreur de syntaxe si elle n'est pas doublée, car elle est censée terminer la chaîne.

## CONTENU D'UNE TABLE

Vous aimeriez sans doute vérifier que les lignes que vous avez saisies sont bien présentes dans la table `PRESIDENT`. La commande `SELECT ... FROM ... [WHERE ...]` permet d'afficher le contenu d'une table, en restreignant éventuellement le résultat aux lignes qui respectent un critère de sélection.

### Espace de travail

Saisissez les instructions SQL, PL/SQL et SQL\*Plus :

```
SELECT *  
FROM PRESIDENT;
```

\* Derrière `SELECT` signifie :

Afficher **tous** les champs !

Exécuter    Charger script    Enregistrer script    Annuler

Résultats    Messages

	ID	Nom	Prenom	NbMandats	DebMandats	FinMandats	Naissance	Mort	Parti	Republique
1	18	DE GAULLE	Charles	2	1959-01-08	1969-04-28	1890-11-22	1970-11-09	UNR - UDR	5
2	19	POMPIDOU	Georges	1	1969-06-20	1974-04-02	1911-07-05	1974-04-02	UDR	5
3	20	GISCARD D'ESTAING	Valéry	1	1974-05-27	1981-05-21	1926-02-02	NULL	FNRI - UDF	5
4	21	MITTERAND	François	2	1981-05-21	1995-05-17	1916-10-26	1996-01-08	PS	5
5	22	CHIRAC	Jacques	2	1995-05-17	2007-05-16	1932-11-29	2019-09-26	RPR - UMP	5
6	23	SARKOZY	Nicolas	1	2007-05-16	2102-05-16	1955-01-28	1900-01-01	UMP	5
7	24	HOLLANDE	François	1	2012-05-16	2017-05-07	1954-08-12	NULL	PS	5
8	25	MACRON	Emmanuel	1	2017-05-07	NULL	1977-12-21	NULL	EN MARCHE	5

Par défaut, la largeur des colonnes est calée sur la taille des champs, ce qui peut donc couper les titres.



ET SI je ne veux pas afficher toutes les lignes, mais seulement certains présidents ?

➤ On rajoute un critère de sélection



grâce à la clause ... bien sûr !

🔗 **Exemple 1** : Liste des présidents qui ont fait deux mandats

```
➤ SELECT *  
FROM PRESIDENT  
WHERE NbMandats = 2;
```

🔗 Vérifier le résultat de cette commande SQL : Combien de lignes retournées ?

NOTA BENE : Le langage SQL ne distingue pas les minuscules/majuscules dans sa syntaxe, mais on utilise couramment les MAJUSCULES pour les MOTS-CLES ; cela aide également à la lecture des requêtes, de la même manière que l'écriture sur plusieurs lignes, une par CLAUSE principale.

ATTENTION TOUTEFOIS au fait que si vous recherchez une valeur dans la table, la casse est bien distinguée : « **WHERE** Prenom='François' » est différent de « **WHERE** Prenom='FRANCOIS' ».



ET SI je ne veux pas afficher toutes les colonnes, mais seulement certains champs ?

➤ On spécifie les champs, séparés par des virgules à la place de l'\* !

🔗 **Exemple 2** : Nom et prénom des présidents qui sont encore vivants (autrement dit qui ne sont pas morts)

```
➤ SELECT Nom, Prenom
FROM PRESIDENT
WHERE Mort IS NULL;
```

📖 **IS NULL** signifie a la valeur « NULL », donc « n'est pas renseigné »

🔗 Vérifier le résultat de cette commande SQL : Combien de lignes retournées ? Combien de colonnes ?

Nous n'allons pas passer en revue toutes les possibilités, ni même tous les mots-clés, mais encore quelques exemples, simplement pour donner une idée des possibilités offertes par le langage SQL, et répondre aux questions enthousiastes qui vous viennent, j'en suis certain, à l'esprit.



Peut-on effectuer des calculs, comme par exemple l'âge auquel sont morts les présidents (s'ils ne sont plus en vie bien sûr).

➤ A la place d'un nom de champ, on spécifie le calcul que l'on souhaite faire.

🔗 **Exemple 3** : Nom et prénom des présidents, avec leur âge au moment où ils sont passés de vie à trépas.

The screenshot shows a SQL query in SQL Server Enterprise Manager. The query is:

```
use FG_REPUBLIQUE;
SELECT nom, prenom, YEAR(mort) - YEAR(naissance)
from president;
```

The results table shows the following data:

	nom	prenom	(Aucun nom de colonne)
1	DE GAULLE	Charles	80
2	POMPIDOU	Georges	63
3	GISCARD D'ESTAING	Valéry	NULL
4	MITTERAND	François	80
5	CHIRAC	Jacques	NULL



« *Moi j'y comprends rien...* »

Evidemment ce n'est pas le calcul le plus simple ! Pas de panique, pour l'instant, on ne vous demande pas de savoir la trouver tout seul cette requête !

Cette requête permet de voir 2 choses :

- L'existence de fonctions de conversions
- La nécessité de renommer les colonnes parfois, et notamment lorsqu'il s'agit d'un calcul.



Comment renommer une colonne ou donner un titre à une colonne qui affiche le résultat d'un calcul ?

- On l'indique en utilisant le mot clé AS suivi du titre de colonne ; si celui-ci comporte des espaces, il faut l'encadrer de guillemets (double quotes)

Voici le résultat d'une requête similaire à la précédente, en donnant un titre à la colonne dont le résultat est calculé :

The screenshot shows a SQL query window with the following code:

```

/***** Script de la commande SelectTopNRows à partir de SSMS
use FG_REPUBLIQUE;
SELECT nom, prenom, YEAR(mort) - YEAR(naissance) as âge
from president;

```

The results grid below shows the following data:

	nom	prenom	âge
1	DE GAULLE	Charles	80
2	POMPIDOU	Georges	63
3	GISCARD D'ESTAING	Valéry	NULL
4	MITTERAND	François	80
5	CHIRAC	Jacques	NULL



Est-ce qu'on peut également faire des sommes, des moyennes, trouver une valeur maximale, etc ... ?

- Oui il existe des fonctions d'agrégat (SUM, AVG, MIN, MAX) et même la possibilité de calculer par groupe de lignes, en fonction de la valeur d'un champ.

🔗 **Exemple 4** : Nombre de présidents présents dans la table.

- **SELECT COUNT(\*) AS "Nombre de presidents"**  
**FROM PRESIDENT;**

- 🔗 Vérifier le résultat de cette commande SQL : Combien de lignes retournées ? Combien de colonnes ? Quelle valeur ?

🔗 **Exemple 5** : Quelle est la date de la dernière élection ?

On peut répondre à cette question de plusieurs manières :

- C'est la date de début de mandat pour lequel la date de fin n'est pas renseignée
- C'est la dernière date de début de mandat, autrement dit la plus « grande » ou la plus forte valeur de DebMandats.

Cette 2<sup>ème</sup> version m'intéresse particulièrement. Voici la solution :

- **SELECT MAX(DebMandats) AS "Derniere election"**  
**FROM PRESIDENT;**

- 🔗 Vérifier le résultat de cette commande SQL : Combien de lignes retournées ? Quelle valeur ?

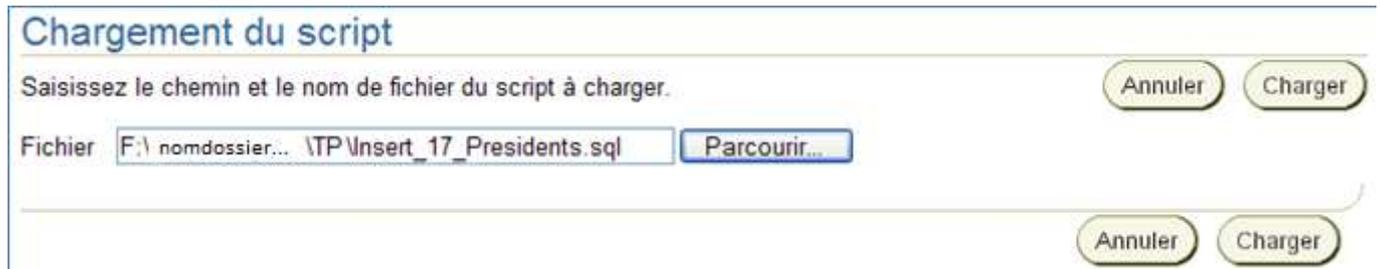
Vous savez écrire la première version : trouvez la requête et inscrivez-la ci-contre

**SELECT**  
**FROM**

Pour pouvoir terminer avec quelques exemples sympatiques, ajoutons les 17 premiers présidents français dans notre table, de BONAPARTE à COTY.

Comme cette tâche serait fastidieuse, je vous invite à charger puis exécuter le script qui se trouve dans le répertoire de ressources du TD.

- Cliquer sur « Fichier » puis « Ouvrir » Pour accéder à la page de chargement d'un script
- Cliquer ensuite sur « Fichier » Pour pouvoir sélectionner le script (dans ressources) : « Insert\_17\_Presidents.sql »



- Cliquer ensuite sur « Ouvrir » Pour revenir à la page principale
- Cliquer enfin sur « Exécuter » Pour lancer l'exécution du script
- Vérifier que la table contient bien maintenant les 23 présidents.

🔗 Quelle commande tapez-vous ?

Nous pouvons maintenant voir encore un ou deux exemples de requêtes. Tout d'abord, dans l'affichage précédent, vous avez sans doute été frustrés de l'ordre d'affichage des présidents. Il est fréquent que l'on veuille un affichage par ordre alphabétique, ou bien en fonction d'un rang, etc...

🔗 **Exemple 6** : Liste des présidents par ordre chronologique.

- On peut utiliser l'ID ou bien la date de début de mandat.
- Il existe effectivement une clause de tri : **ORDER BY**

```
SELECT *  
FROM PRESIDENT  
ORDER By ID;
```

```
SELECT *  
FROM PRESIDENT  
ORDER By DebMandats;
```

🔗 **Exemple 7** : Nombre de présidents pour chacune des « républiques » (deuxième, troisième, quatrième).

NB : Si on veut écrire une seule requête, il faut qu'on ait le moyen de « compter » le nombre de lignes par valeur du champ « République »

Il existe effectivement une clause de regroupement (GROUP BY) qui si elle est spécifiée, permettra de faire un seul affichage pour chaque valeur du champ de regroupement !

```
SELECT République, COUNT(*) AS "Nb president / republique"  
FROM PRESIDENT  
GROUP By République;
```

Le regroupement n'est pas un tri ; si vous voulez les républiques dans l'ordre croissant, il faut rajouter la clause ORDER BY :

```

SELECT      Republique, COUNT(*) AS "Nb presidents / republique"
FROM        PRESIDENT
GROUP By    Republique
ORDER By    Republique;

```

Vous devriez obtenir le résultat ci-contre :

*Maman ! J'ai raté quelque chose ?!?*

REPUBLIQUE	Nb presidents / republique
2	1
3	14
4	2
5	6

**Exemple 8** : Quel est le président actuellement en exercice ?

- Solution 1 : C'est celui dont la date de fin de mandat n'est pas encore connue (simple)  
Ou bien
- Solution 2 : C'est celui dont la date de début de mandat est la dernière (la « maximum »)

#### Solution 1

```

SELECT      Nom, Prenom
FROM        PRESIDENT
WHERE       ...
-- à compléter

```

#### Solution 2

```

SELECT      Nom, Prenom
FROM        PRESIDENT
WHERE       DebMandats = (
SELECT      Max(DebMandats)
FROM        PRESIDENT
)

```

## ADMINISTRATION DE LA BASE DE DONNEES > SAUVEGARDES ET RESTAURATION

Un clic droit sur votre BDD republique et puis Tâches / Sauvegarder ... puis valider le chemin par défaut : C:\Program Files\Microsoft SQL Server\MSSQL13.VAIOFG\_SQL2016\MSSQL\Backup\

On pourrait utiliser la requête de sauvegarde de votre base de données :

```

backup database XX_REPUBLIQUE to DISK=C:\Program Files\Microsoft SQL
Server\MSSQL13.VAIOFG_SQL2016\MSSQL\Backup\XX_REPUBLIQUE.bak '

```

Vous devez obtenir le message de réussite suivant :

*La sauvegarde est effectuée sur le serveur dans le dossier prévu*

Messages

```

184 pages traitées pour la base de données 'FG_REPUBLIQUE', fichier 'FG_REPUBLIQUE' dans le fichier 1.
1 pages traitées pour la base de données 'FG_REPUBLIQUE', fichier 'FG_REPUBLIQUE_log' dans le fichier 1.
BACKUP DATABASE a traité avec succès 185 pages en 0.044 secondes (32.848 Mo/s).

```

Au-préalable une petite requête pour compter le nombre de présidents :

```

SQLQuery2.sql - P...-PC\prof-FG (51))*
SELECT COUNT (*) as NB_PREIDENTS FROM PRESIDENT;

```

Résultats Messages

	nb_president
1	25

Puis une petite requête de suppression (on va dire par erreur) de quelques présidents :



Maman ! Je me suis trompé(e) ??

```
SQLQuery7.sql -...E (eleve (55))* | SQLQuery6.sql -...E (eleve (59))* | SQLQuery5.sql -...E (eleve (55))
Delete from president where id in(1, 2, 3, 7, 23);

Messages

(5 ligne(s) affectée(s))
```

Et comme SQL SERVEUR travaille en **autocommit**, c'est-à-dire :  
Le mode **autocommit** est le mode de gestion par défaut des transactions du Moteur de base de données SQL Server. Chaque instruction Transact-SQL est validée ou restaurée dès qu'elle se termine. Lorsqu'une instruction est exécutée avec succès, elle est validée ; si une erreur se produit, elle est restaurée.  
*Donc ici pas moyen d'annuler la suppression, une seule solution : la restauration puis que sauvegarde il y a eu au préalable.*

Vérification :

Requête pour compter le nombre de présidents :

```
SQLQuery2.sql - P...-PC\prof-FG (51))*
SELECT COUNT(*) as NB_PRESIDENTS FROM PRESIDENT;

Résultats | Messages

nb_president
1 | 20
```



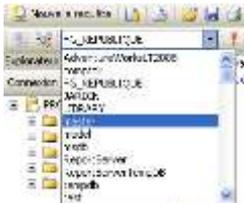
Mais non ! On va restaurer ...

Restauration de la BDD republique en utilisant le clic droit / Tâches / Restaurer ou la requête de restauration suivante :

1<sup>er</sup> essai :

```
restore database AM_REPUBLIQUE from disk=N'C:\Program Files\Microsoft SQL Server\MSSQL13.VAIOFG_SQL2016\MSSQL\Backup\AM_REPUBLIQUE.bak' with replace;
```

Si vous obtenez un message d'erreur, fermer toutes les fenetres des requêtes. Cliquer sur la base de données master selon la copie d'écran suivante :



2<sup>ème</sup> essai :

```
restore database AM_REPUBLIQUE from disk=N'C:\Program Files\Microsoft SQL Server\MSSQL13.VAIOFG_SQL2016\MSSQL\Backup\AM_REPUBLIQUE.bak' with replace;
```

## Requête pour compter le nombre de présidents :

```
SQLQuery2.sql - P...-PC\prof-FG (51))*
SELECT COUNT(*) as NB_PREIDENTS FROM PRESIDENT;
```

Résultats Messages

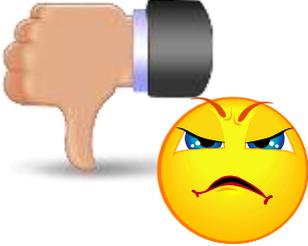
	nb_president
1	25

Il faudra envisager de sauvegarder la base de données régulièrement pour se prémunir des erreurs.

## ET AINSI DE SUITE...



Si vous avez terminé, vous pouvez tenter d'écrire les requêtes suivantes, par imitation et intuition :

	<p><b>ATTENTION ! IL EST INTERDIT DE CONSULTER LE CONTENU DES TABLES (LES VALEURS DES ENREGISTREMENTS) POUR FAIRE LA REQUETE</b></p> <p><i>La consultation doit juste vous permettre de vérifier le résultat, mais elle ne doit en aucun cas vous aider à écrire les critères. L'énoncé de la question doit suffire !</i></p>
--	---

Vous êtes invité à noter votre requête.

1. Quels sont les présidents (Nom, Date de naissance) qui se sont prénomnés Paul ?

2. Pour quel(s) président(s) (Nom, Prenom, Date de leur mort) le mandat a-t-il été interrompu par leur mort ? (fin de mandat et mort le même jour)

.....

3. Combien de présidents ont été du parti dit « Modéré » ? (renommer judicieusement la colonne)

.....

4. Quels sont les présidents qui ont vécu moins de 50 ans ? (idem)

.....



5. Nombre total de mandats (sur l'ensemble des présidents) (idem)

.....

6. Nombre total de mandats depuis le début de la 5<sup>ème</sup> république (idem)

.....

.....

7. *Nombre de mandats par république ? (renommer la colonne nb mandats)*

.....

8. *Qui (Nom et prénom) a connu Emile LOUBET ? (Né avant Emile Loubet et mort après sa naissance)*

.....

9. *Qui (Nom prénom) a succédé à Jacques CHIRAC (Date de début de mandat de ce président à Date de fin de Jacques CHIRAC)*

.....

10. *Qui (nom, prénom, dates de mandat) a succédé à Charles DE GAULLE (en servant du numéro de président)*

.....

Empty dotted-line box for text entry.



11. *Quel sera le nom du prochain président (nom, prénom, dates de mandat) élu en mai 2017*

Large empty dotted-line box for text entry.