

Bases de données

TDs MACHINE

Enseignant

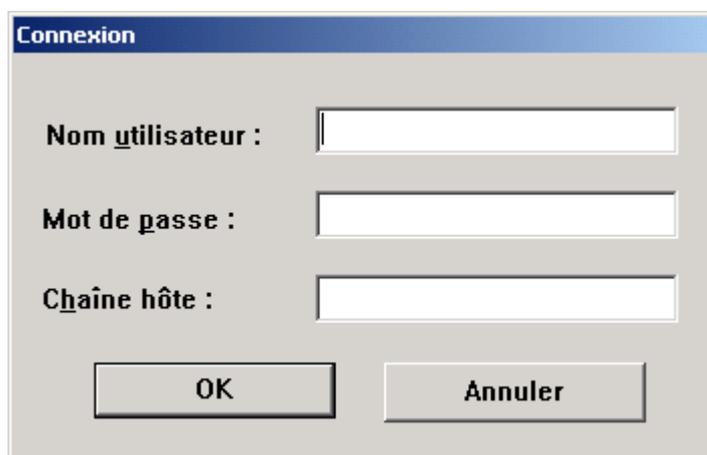
Jean-Yves ANTOINE

(Jean-Yves.Antoine@univ-tours.fr)

Connection SGBD Oracle (installation IUP Blois)

Travail sous l'utilitaire SQLplus

1. Logez vous sous votre compte étudiant habituel
2. Connectez-vous sous l'utilitaire SQL+ d'Oracle à partir de la barre Démarrer : menu Programmes → Oracle OraClient → Development Application → SQLPlus



Nom utilisateur votre nom de compte

Mot de passe votre nom de compte

Chaîne hôte db001_Oracle

Il existe une interface graphique (SQLPlus Worksheet) pour utiliser Oracle de manière plus conviviale. Nous n'utiliserons cependant ici que l'interface minimale SQLPlus : l'objectif de ces Tps n'est en effet pas de se former à Oracle, mais à SQL qui se retrouve sur tous les SGBD commerciaux.

TDM 1 – SQL et LDD

1. Présentation

Plus fort que de savoir qui de Zidane ou Materazzi était le plus fautif en finale de Coupe du Monde de football, ce qui a passionné le microcosme journalistique en 2007 était l'affrontement entre Ségolène Royal et Nicolas Sarkozy pour devenir calife à la place du calife (l'avez-vous oublié ? Jacques Chirac était président de la République, si si je vous assure !). Cette surmédiation de deux candidats est la conséquence de la peopolisation de la politique française, quinze ans après son émergence aux Etats Unis d'Amérique. C'est ainsi que tous les jours, la ménagère de 50 ans put suivre sur TF1 les amours tumultueux de Cécilia et Nicolas. Et après l'élection, cette ménagère n'a bien sûr par manqué d'acheter le numéro de *Voici* où l'on pouvait tout savoir sur la séparation de Ségolène et de François Hollande. Passionnant tout cela..



Les autres candidats, à commencer par François Bayrou qui parla de collusion entre les puissances médiatico-financières et les deux partis dominants de la vie politique (ce qui existe effectivement aux USA), virent dans cette surmédiation les raisons de leurs échecs.

Avai-ils raison ? Voilà une question à laquelle nous ne répondrons pas dans ce TP. Nous nous contenterons de prendre ce thème comme prétexte à une étude de la définition de bases de données sous Oracle, à l'aide du langage SQL.

On considère donc une base de données regroupant différentes informations sur les hommes politiques de notre pays, et en particulier leur cote de popularité estimée par différents instituts de sondages. Soient donc les relations suivantes :

- `personne(num_h, nom, prenom, parti)` liste des hommes politiques,
- `parti(num_p, sigle, nom)` liste des partis,
- `sondage(institut, n_hom, cote, date)` listes des côtes de popularité.

personne	attribut	Définition
	num_h	référence de l'homme politique, entier
	nom	nom de la personne, chaîne 30 caractères, obligatoirement renseigné
	prenom	prénom de la personne, chaîne de 20 caractères
	parti	référence au parti auquel appartient la personne

parti	attribut	Définition
	num_p	référence de parti, entier positif
	sigle	sigle du parti, chaîne de 10 caractères, obligatoirement renseigné
	nom	nom complet du parti, chaîne de 30 caractères

sondages	attribut	Définition
	institut	nom de l'institut de sondage
	n_hom	référence à la personne concernée
	cote	cote de popularité (% compris entre 0 et 100), entier, renseigné
	dates	date du sondage, date, obligatoirement renseigné

On donne la définition des dépendances fonctionnelles correspondant à cette BD:

- relation PERSONNE $\text{num_h} \rightarrow \text{nom, prenom, parti}$
- relation PARTI $\text{num_p} \rightarrow \text{sigle, nom}$
- relation SONDAGES $\text{institut, n_hom, date} \rightarrow \text{cote}$

On donne de même les dépendances d'inclusion correspondantes:

- $\Pi_{\text{parti}}(\text{PERSONNE}) \subseteq \Pi_{\text{num_p}}(\text{PARTI})$
- $\Pi_{\text{n_hom}}(\text{SONDAGES}) \subseteq \Pi_{\text{num_h}}(\text{PERSONNE})$

2. Travail demandé

1 — Déterminez l'ensemble des clés primaires et étrangères de la base de données. Dans quel ordre devront nécessairement être créées vos relations ? Justifiez votre réponse.

2 — Ecrire **sur papier** les requêtes SQL de création des relations `parti` et `personne`.

Faites valider ce schéma conceptuel et ces requêtes par votre enseignant. Si tout va bien, vous pouvez alors vous connecter à Oracle pour réaliser le TP proprement dit.

3 — Créez sous Oracle l'ensemble des relations de votre base de données.

4 — A l'aide de la commande `describe`, ainsi qu'en parcourant le dictionnaire de données Oracle, vérifiez que les relations de la base de données ont bien été créées dans leur ensemble. Quelles caractéristiques de la base de données ne peuvent être vérifiées par ce type de commande ?

Nous verrons l'an prochain (étude plus approfondie du dictionnaire Oracle) comment vérifier exhaustivement l'ensemble des propriétés du schéma conceptuel.

Etant directement responsable de l'action gouvernementale, tout ministre est une personnalité politique plus exposée en terme d'impopularité. Pour étudier ce phénomène, il serait intéressant de rajouter à la relation `personne` un attribut `fonction` qui décrit la fonction principale (président, ministre, député etc...) assurée par la personne considérée. Cet attribut correspondra à une chaîne de 20 caractères maximal et ne sera pas obligatoirement renseignée (un homme politique peut n'assurer aucune fonction particulière).

5 — Modifiez la définition de la relation `personne` pour ajouter ce nouvel attribut. Vérifiez ensuite que la structure de la relation modifiée est correcte.

Il existe en France plus d'une trentaine de partis politiques : connaissez-vous certaines formations régionales comme l'UDB (Union Démocratique Bretonne, gauche régionaliste) ou AA (Alsace d'Abord, extrême droite alsacienne) ? En dehors des principaux partis nationaux, un sigle ne représente rien pour le citoyen lambda. Afin de rendre notre base de données plus lisible, il semble donc important d'imposer une contrainte de renseignement sur l'attribut `nom` de la relation `parti`.

6 — Modifiez la définition de la relation `parti` pour ajouter cette nouvelle contrainte d'intégrité. Comment pourriez-vous vérifier (partiellement...) que cette contrainte a bien été associée à la relation considérée.

Si on fait souvent de la politique en famille (François Hollande et Ségolène Royal...), il y a tout de même peu de chance de que deux hommes politiques aient le même nom et le même prénom (avec, certes, l'exception des Bush père et fils...). On peut donc imaginer que le couple d'attributs (`nom`, `prenom`) joue le rôle de clé primaire pour la relation `personne`.

7 — Modifiez cette relation en définissant comme nouvelle clé primaire le couple (`nom`, `prenom`) et en vous contentant de définir une contrainte d'unicité sur l'ancienne clé `num_h`. Qu'observez-vous ?

8 — Essayez de supprimer la clé primaire de la relation `parti`. Qu'observez-vous ? Commentaire?.

9 — Remplissez les deux premières relations de votre base de données avec les enregistrements suivants :

PERSONNE	Num_h	Nom	Prenom	Parti	Fonction
	1	Chirac	Jacques	1	président
	2	Sarkozy	Nicolas	1	ministre
	3	Straus-Kahn	Dominique	2	—

PARTI	Num_p	Sigle	Nom
	1	UMP	Union pour la Majorite Presidentielle
	2	PS	Parti Socialiste

Qu'observez vous au cours des différents chargements ? Ces réponses du système étaient-elles prévisibles ?

On observe rapidement la lourdeur de cette saisie enregistrement par enregistrement. A l'occasion du prochain TP, nous verrons qu'ORACLE, comme tout SGBD qui se respecte, dispose d'un utilitaire (appelé ici SQLoader) pour faciliter l'importation d'informations à partir d'un fichier de données.

Pour le moment, vous pouvez également utiliser la "bête" commande SQL d'insertion de tuples pour vérifier la bonne définition de vos contraintes d'intégrités (tentative d'insertion d'enregistrements qui enfreignent les contraintes d'intégrité).

3. Pour aller plus loin...

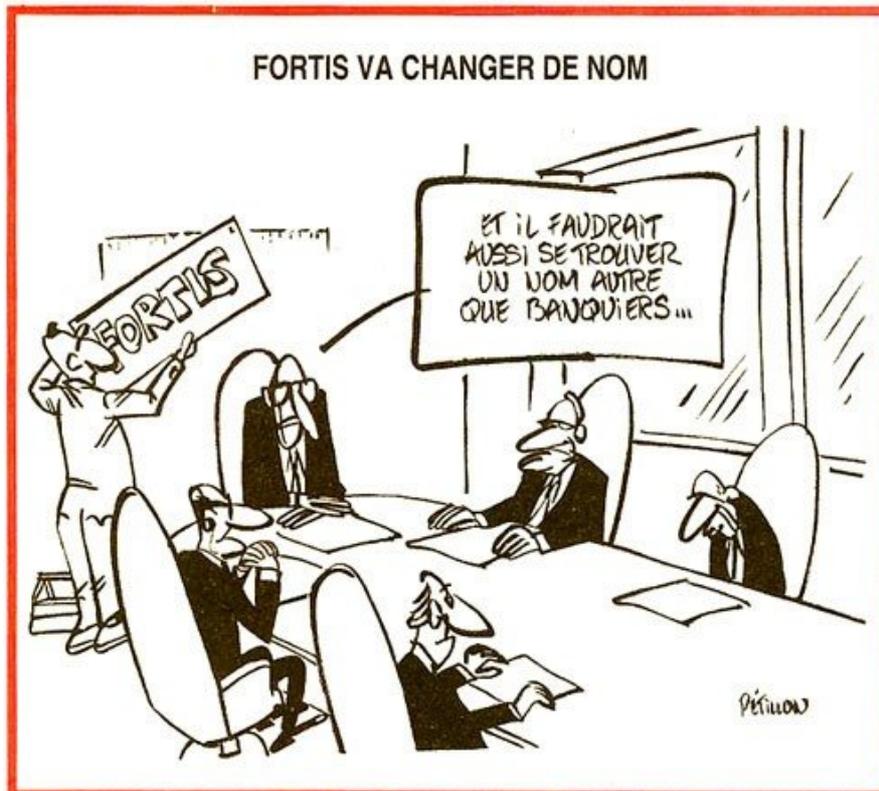
Afin de terminer ce TP, et en préparation du TP à venir, nous allons maintenant interroger la base de données « sondages politiques » que vous avez créée.

- 1 — On veut connaître la cote de popularité moyenne des membres de chaque partis. Les résultats seront donnés par ordre décroissant de popularité.
- 2 — On accuse parfois les instituts de sondage de ne pas être politiquement neutres, c'est à dire de gonfler ou au contraire d'atténuer volontairement la cote de popularité de certains homme politiques ? Pour traquer les éventuels favoritismes, on désire obtenir tous les couples de cotes de popularité émanant d'instituts différents et présentant une différence supérieure à 10%. On donnera à chaque fois en réponse le nom de l'homme politique concerné, les 2 cotes de popularités relevées et les instituts de sondages qui leur correspondent.
- 3 — Un institut de sondage a-t-il évalué la cote de popularité de **l'ensemble** des hommes politiques présents dans la base de données ? Créer une requête qui réponde à cette question.
- 4 — Les petits partis politiques accusent souvent les médias et les instituts de sondages de favoriser, en leur accordant plus de temps d'antenne ou plus de sondages, les hommes politiques des partis dits "de gouvernement". Afin de vérifier la pertinence de ce reproche, on désire savoir quel est le nombre moyen de cotes de popularité des hommes politiques de chaque parti ? Créer une requête qui donne donc, pour chaque parti, le nombre moyen de sondages par homme politique.

TDM 2 –SQL : interrogation de BD et LMD

2. Présentation

L'informatisation des différentes places boursières mondiales, ainsi que leur mise en réseau, ont entraîné la constitution de gigantesques bases de données fourmillant d'informations et consultables à tout moment sur Internet. Ces interconnexions, couplées à la mise en place de systèmes experts d'achat et de vente automatique d'actions, ont eu d'ailleurs un effet démultiplicateur dans la crise boursière asiatique en 1999, puis plus près de nous au cours de l'effondrement de la bulle spéculative autour d'Internet et en 2008-2009 celle des *subprimes* : les systèmes experts automatiques couplés à ces bases de données se sont en effet précipités aveuglément dans une stratégie aveugle de vente qui n'a fait qu'accroître la chute des cours...



Dans le cadre de ce TP, nous allons considérer une petite base de données consacrée précisément au domaine boursier. Cette base de données (simplifiée de manière presque caricaturale) concerne l'ensemble des places boursières mondiales et donne plusieurs types d'informations sur les valeurs qui y sont cotées :

- tout d'abord, des informations sur les cours de ses valeurs (actions), leur évolution etc...
- ensuite, des informations plus subjectives, correspondant aux prévisions d'évolution et conseils d'achats donnés par différents journaux financiers.

La base de données est composée de 4 relations (clés primaires en gras, étrangères soulignées):

- valeurs(**nom**, marché, varia_an) liste des actions,
- cours(**valeur**, **date**, cote, volume, varia_jr) cotations pour chaque clôture journalière,
- prévisions(**valeur**, **expert**, **date**, objectif, conseil, risque) prévisions associées à chaque valeur

Valeurs Table regroupant les informations sur chaque valeur cotée:

Champ	Commentaires	Type de données
Valeur	Code de la valeur (nom de l'action)	chaîne de caractères - obligatoirement renseigné
Marché	Code du marché correspondant	chaîne de caractères - obligatoirement renseigné
Varia_an	Variation depuis le début de l'année	pourcentage

Cours Table regroupant les informations sur le cours de chaque valeur clôture journalière :

Champ	Commentaires	Type de données
Valeur	Référence de la valeur cotée	chaîne de caractères
Dates	Date du jour	type DATE
Volume	Nombre d'actions échangées ce jour	entier - obligatoirement renseigné
Cote	Cotation de la valeur à la clôture	réel - obligatoirement renseigné
Varia_jr	Variation journalière	pourcentage - obligatoirement renseigné

Prévisions Table regroupant les informations sur les prévisions associées à chaque valeur :

Champ	Commentaires	Type de données
Valeur	Référence de la valeur cotée	chaîne de caractères - obligatoirement renseigné
Expert	Journal responsable de la prévision	chaîne de caractères - obligatoirement renseigné
Dates	Date du jour	type DATE
Objectif	Objectif de cote à moyen terme	réel - obligatoirement renseigné
Conseil	Conseil d'achat	caractère pris parmi les valeurs suivantes : A (Achat), V (Vente), C (Conserver), B (Prise de bénéfice) - obligatoirement renseigné
Risque	Indice de confiance en la prévision	entier compris entre 0 (risqué) et 5 (très sûr).

3. Travail demandé : interrogation de base de données

- Cette BD a déjà été créée sous le serveur ORACLE de l'IUP sous forme de *synonymes publics* (notion que vous étudierez en Master 1). Récupérez le contenu de chacune des relations de la base afin de pouvoir vérifier ultérieurement la bonne exécution de vos requêtes.
- Jusqu'à la chute des valeurs automobiles du fait de la crise actuelle, Renault était une valeur phare de la Bourse de Paris. Donnez les valeurs du marché parisien ayant progressé plus rapidement, sur l'année, que cette action. On donnera la réponse en respectant le format d'affichae suivant :

Valeur	Marché de cotation	Variation annuelle
=====	=====	=====
ATOS	Paris	+26
- Donnez les valeurs dont la cote a dépassé au moins un objectif proposé par un journal financier : il est en effet urgent de réfléchir à leur vente... On affichera donc la valeur, son marché, sa cote, l'objectif de cote, le nom du journal financier et son conseil d'achat / vente.
- Une place boursière réalise-t-elle de meilleures performances que les autres? Pour répondre à cette question, réalisez une requête qui calcule la variation annuelle de chaque marché, calculée comme la moyenne des variations de ses valeurs cotées. Affichage par ordre décroissant de variation.
- Les conseils des experts financiers sont-ils fiables ? Pour répondre à cette question, réalisez une requête qui donne, pour chaque journal financier, la moyenne d'évolution journalière des valeurs que cette publication conseille à l'achat.
- **Requête paramétrée** — Reprenez la requête de la question 2 pour la créer une vue paramétrée permettant de demander la place boursière à laquelle on s'intéresse (et non plus Paris obligatoirement) et la valeur qui doit servir de référence (et non plus Renault obligatoirement).
- **Division euclidienne** — Peut-on jouer en bourse sans risque? Pour du moins limiter les dégats, écrire une requête qui donne leurs valeurs (nom, marché) qui ont toujours fait l'objet d'une recommandation d'achat.
- **Requête intermédiaire** — Quelles sont les valeurs phares de chaque marché? Pour répondre à cette question, réalisez une requête qui donne pour chaque marché la valeur ayant connu la plus grosse progression journalière.

4. Travail demandé : LMD

Politique fiction : suite à l'élection présidentielle de 2007 de Nicolas Sarkozy, le gouvernement libéral au pouvoir décide de privatiser l'enseignement supérieur. L'IUP Blois se retrouve coté au Nouveau Marché sur la bourse de Paris, la rémunération du directeur de l'IUP étant directement liée à l'évolution du cours de l'action (gloups !).

9 — Ecrire **sur papier** (compte-rendu) une nouvelle action IUP dans les relations VALEUR et COURS. Pour la première journée de cotation, l'action IUP a terminé la séance à 17,8 Euros, en progression de 25,6 % (on est très fort) et un volume échangé de 56 283 actions (tous les anciens étudiants de l'IUP se sont précipités sur cette valeur prometteuse).

Fiction économique : après des années de déficits abyssaux financés par l'économie mondiale, le dollar a perdu tout crédit et se trouve fortement dévalué en cette année 2012 : 1 \$US = 0,12 €. Acculé à la banqueroute, le gouvernement du président Georges W. Bush II Jr (le petit fils de Georges W. Bush réélu après avoir envoyé 100 000 Marines envahir le Liechtenstein soupçonné de produire du chocolat suisse bactériologique) se voit obligé d'abandonner la monnaie nationale pour utiliser l'euro. Les cotations à Wall Street (New-York) doivent donc être maintenant exprimées en euros.

10 — Ecrire **sur papier** une requête pour modifier la cote de toutes les valeurs new-yorkaises pour les exprimer désormais en Euros.

1. Présentation

En ce début de XXI^e siècle, la population urbaine a, pour la première fois dans l'histoire, dépassé en nombre la population rurale. Nous assistons ici à une inquiétante accélération de l'évolution de l'humanité. Première hyper-mégalopole de l'histoire, la Rome antique ne réunissait qu'une population de 800 000 habitants environ. Au XVI^e siècle, le conquistador Hernan Cortès découvrait avec stupéfaction **Techniclan** (actuelle Mexico) au milieu de sa lagune. Avec plus d'un million d'habitants, la capitale de l'empire aztèque dépassait de loin la population des plus grandes villes européennes. La révolution industrielle allait conduire à un premier mouvement d'exode rural : à la fin du XIX^e, plusieurs métropoles dépassaient déjà le million d'habitants. Une telle population faisait toutefois d'une agglomération une ville mondiale. A l'orée du XXI^e c'est par un facteur par 10 qu'il faudrait au moins multiplier ce chiffre, et on ne compte plus les centaines de villes de plus d'un million d'habitant ignorées du grand public : connaissez-vous Kisinov (Chisnau en allemand), en Moldavie, Lodz en Pologne, Tula en Russie, Lvov ou Krijov en Ukraine ?

Ce sont des villes que nous allons rencontrer au cours d'un TP dont l'objectif est de rapprocher vos connaissances sur les bases de données, leur modélisation et leur interrogation, de problèmes réels d'utilisation. En particulier, nous allons voir :

1. Comment importer automatiquement de grandes masses de données dans le SGBD Oracle. Nous utiliserons pour le cela l'utilitaire Sql*loader.
2. Comment exprimer des requêtes avancées avec des directives SQL un peu spécifiques (ANY, ALL, SOME, IN) mais très utiles en pratique.
3. Comment utiliser certains opérateurs (LIKES) et fonctions de manipulation de données propres à Oracle (manipulation de dates, travail sur les formats de chaînes de caractères, fonctions numériques...).

2. Création et chargement de la base de données

Au cours de ce TP, nous allons travailler sur une base de données qui recense l'ensemble des villes européennes d'importance en donnant entre autres leur pays, leur population (ville principale et non agglomération) et leur position géographique (géolocalisation d'une base de données). Les villes françaises étant moins bien décrites que les autres (uniquement quelques agglomérations importantes) dans les données que nous utiliserons, les informations les concernant seront sauvegardées à part.

La base de données est composée des 2 relations suivantes :

- Ville(id, nom_ville, pays, statut, pop) information sur chaque ville non française
- Villefr(nom_ville, statut, pop) information sur chaque ville française
- Loc_ville(nom_ville, pays, latitude, longitude) géolocalisation des villes

Ville et **Villefr** Table regroupant les informations sur chaque ville

Champ	Commentaires	Type de données
Id	Identificateur associé à la ville	Chaîne de caractères
Nom_ville	Nom de la ville	Chaîne de caractères - obligatoirement renseigné
Pays	Pays où se situe la ville (table VILLE)	Chaîne de caractères - obligatoirement renseigné
Statut	Capitale ou non	Chaîne de caractères
Pop	Population de la ville	Entier
Date_pop	Date du recensement de population	Date

Loc_ville Table regroupant les informations sur la localisation de chaque ville :

Champ	Commentaires	Type de données
Id	Identificateur associé à la ville	Chaîne de caractères
Nom_ville	Nom de la ville	Chaîne de caractères - obligatoirement renseigné
Pays	Pays où est situé la ville	Chaîne de caractères - obligatoirement renseigné
Latitude	Latitude N ou S de la ville	Chaîne de caractères - obligatoirement renseigné
Longitude	Latitude E ou W de la ville	Chaîne de caractères - obligatoirement renseigné

On donne la définition des dépendances fonctionnelles correspondant à cette BD :

- relations VILLE et VILLE_FR id → nom_ville, pays, statut, pop
- relation LOC_VILLE id → nom_ville, pays, latitude, longitude

L'attribut id de chaque relation sera formé de la concaténation du nom du pays et de la ville concernée (Par exemple, l'identificateur de Paris sera paris_france). On évite ainsi que deux villes (Brest en France et en Biélorussie par

exemple) aient le même identificateur. Les dépendances fonctionnelles précisées ci-dessus sont donc bien satisfaisables.

On donne de même les dépendances d'inclusion correspondantes:

- $\Pi_{id}(VILLE) \subseteq \Pi_{id}(LOC_VILLE)$
- $\Pi_{id}(ILLE_FR) \subseteq \Pi_{id}(LOC_VILLE)$

- 1 — Créez sous Oracle la base de données correspondant à la spécification décrite ci-dessus. Pour dimensionner les attributs alphanumériques de chaque relation, vous pouvez jeter un coup d'oeil aux fichiers de données que vous aurez à importer : fichiers `EUROPE_POP_VILLE.csv` et `EUROPE_LOC_VILLES.csv` disponibles sur la page WWW de cet enseignement.
- 2 — Vérifiez que votre base de données est bien créée en consultant en particulier le dictionnaire Oracle.
- 3 — A l'aide de *SQL Loader*, chargez la base à l'aide des données qui sont présentes dans les fichiers `EUROPE_POP_VILLE.csv` et `EUROPE_LOC_VILLES.csv`. Le nom des pays sera enregistré en majuscules.
- 4 — Toutes les données ont-elles été correctement importées ? Consultez pour cela les différents fichiers de log générés par *SQL Loader* lors de l'importation.

3. Requêtes SQL avancées : prédicats d'appel de sous-requêtes

Nous allons maintenant interroger cette base de données. Ce sera l'occasion de réaliser des requêtes SQL mettant en jeu les prédicats d'appel de sous-requêtes ANY, ALL, SOME, (NOT) IN et EXISTS.

- 1 — Créez une requête qui donne la ville européenne la plus peuplée. Cette requête devra donner en réponse le nom de la ville, son pays et sa population.
- 2 — Créez une requête qui donne la liste des pays comportant au moins une ville de plus d'un million d'habitants.
- 3 — Créez une requête qui donne la liste des pays dont la capitale n'est pas la ville la plus peuplée.
- 4 — Créez une requête qui donne la liste des villes belges géolocalisées mais non présentes dans la table `loc_ville`.

4. Fonctions de manipulation de chaînes ou de dates

Nous allons maintenant interroger la base de données en utilisant les fonctions de manipulation de chaînes ou de formats date.

- 1 — Les données de population dont nous disposons datent d'un référendum réalisé en 1990. Modifiez le contenu des tables `villes` et `ville_fr` en ajoutant la date au 1er janvier 1990 pour le champ `Date_pop`.
- 2 — Existe-t-il plusieurs villes de même nom en Europe ? Pour répondre à cette question, créer une requête qui donne tous les couples de villes de même nom mais de pays différent : on précisera à chaque fois le toponyme en question, suivi du nom des deux pays correspondant. Faire en sorte d'éviter les doublons dans les réponses.
- 3 — Le codage de la base de données n'est pas cohérent : en effet, les villes étrangères les plus connues ont leur nom écrit en français (Londres et non pas London), tandis que les plus confidentielles sont écrites dans leur langue d'origine. Au final, il est difficile de savoir a priori quelle langue utiliser pour interroger la base de données. *Prague* s'écrit *Praha* en tchèque et *Praga* dans d'autres langues. Pourriez-vous créer une requête qui permette de donner toutes les informations (population, localisation) sur cette ville sans avoir à préciser ces trois écritures possibles.
- 4 — De même, il existe plusieurs manières de dire *Saint* ou *Sainte* dans les noms de villes, suivant la langue européenne considérée : *San*, *Santa*, *Santi*, *Sankt* par exemple. Créez une clause SQL qui remplace automatiquement tous ces vocables par un simple S.(S suivi d'un point).
- 5 — Les données entrées dans cette base ont pris en considération les conséquences de la chute du bloc soviétique. Les derniers soubressauts qu'ont connus les Balkans depuis ne sont toutefois pas tous intégrés. Par exemple, la Serbie n'est pas séparée, dans cette base, du Monténégro, ainsi que du Kosovo dont l'indépendance n'est pas encore reconnue de tous les pays. Afin d'attribuer automatiquement aux villes monténégrines leur vraie identité, écrire une requête qui attribue le pays Montenegro à toutes les villes serbes dont la latitude est inférieure à 42° N et la longitude inférieure à 20°.
- 4 — Ecrire une clause SQL qui transforme automatiquement toutes les occurrences de *Yougoslavie* en *Serbie*.