

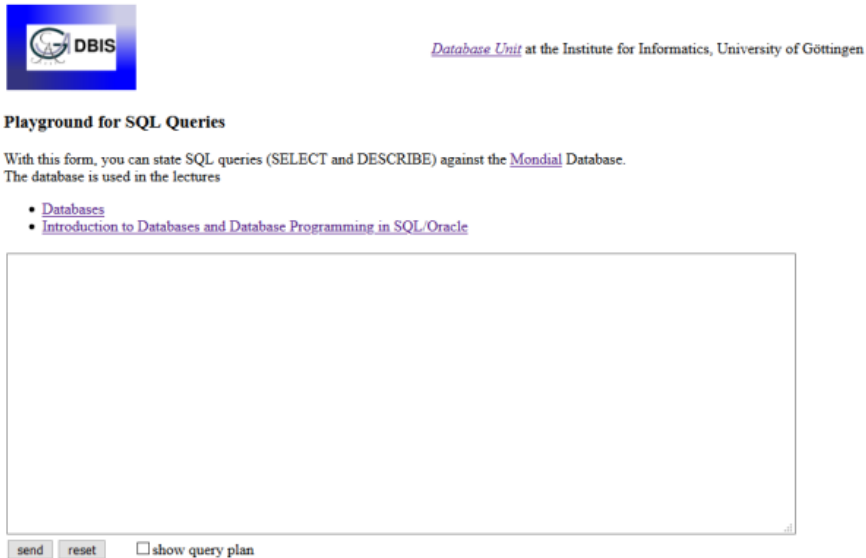
Le langage SQL

Nous avons défini les contours d'une base de données relationnelle, nous allons maintenant apprendre à interroger une base de données.

Nous utiliserons à titre d'exemple la base de données MONDIAL.

Il s'agit d'une **BDR** qui compile un certain nombre de données géographiques et qui est gérée par l'université de Göttingen.

Il est possible d'interagir avec elle en utilisant un formulaire que l'on trouve à l'adresse : <http://www.semwebtech.org/sqlfrontend/> (<http://www.semwebtech.org/sqlfrontend/>).



Nous sommes en présence d'une architecture trois tiers : la première couche (le client) est représentée en HTML pour être exploitée par un navigateur web et sert d'interface entre l'homme et la machine. La seconde couche (le serveur d'application) est un serveur web qui reçoit des données textuelles de la part du client, les transmet sous la forme de requêtes SQL au serveur de la base puis actualise la page web du client pour y intégrer la réponse du serveur. Enfin, la troisième couche est un SGBD, ici Oracle Database, qui gère la base de données et répond aux sollicitations du serveur d'application.

Méthode

Obtenir la liste des tables

Nous allons écrire une requête qui affichera la liste des tables accessible par un utilisateur de cette BDR.

```
1 SELECT table_name FROM user_tables;
```

Les mots-clés SELECT ... FROM réalisent l'interrogation.

À faire : Testez cette requête, vous devez obtenir une liste de 47 noms de tables.

Méthode

Obtenir la description d'une table

La requête suivante permet d'afficher les attributs et leurs domaines.

On considère la table COUNTRY

```
1 DESC COUNTRY;
```

L'attribut NULLABLE (Yes ou No) permet de définir si lors de l'enregistrement, l'information est obligatoire ou pas.

À faire : Tester cette requête et expliquer l'affichage obtenu.

Méthode

Exploration d'une table

On considère la table COUNTRY dont les attributs sont NAME, CODE, CAPITAL, PROVINCE, AREA et POPULATION.

Commençons par extraire de cette table le nom de tous les pays qu'elle contient :

```
1 SELECT name FROM country
```

Remarque : la requête suivante produit le même effet :

```
1 SELECT NAME FROM COUNTRY
```

Les mots-clés `SELECT ... FROM` réalisent l'interrogation de la table. Dans le cas de l'exemple ci-dessus on ne liste qu'un seul des attributs de la table, pour en avoir plusieurs on sépare les attributs par une virgule.

```
1 SELECT name , capital FROM country
```

Pour les avoir tous on les désigne par une étoile.

```
1 SELECT * FROM country
```

À faire : tester ces différentes requêtes

Méthode

Filtrage de données

Si par exemple on souhaite afficher la capitale du Kiribati, on utilisera le mot-clé `WHERE` qui permet de filtrer les données qui répondent à un critère de sélection :

```
1 SELECT capital FROM country WHERE name = 'Kiribati'
```

À faire : Vérifier que la capitale du Kiribati est Tarawa

Il se peut que certains attributs d'un enregistrement soient manquants ; dans ce cas la valeur de cet attribut est `NULL`.

Par exemple, dans la table `country` un territoire ne possède pas de capitale ; pour le connaître on produit la requête :

```
1 SELECT name FROM country WHERE capital IS NULL
```

À faire : Vérifier que ce pays est Gaza Strip

À faire : Que produit cette requête ?

```
1 SELECT name, population FROM country WHERE population < 1000000;
```

Les conditions utilisables :

- `=` (égal)
- `<>` (pas égal)
- `!=` (pas égal)
- `>` (supérieur)
- `<` (inférieur)
- `>=`
- `<=`
- `IS NULL` (valeur nulle - sans valeur)
- `IS NOT NULL`

On peut également trier les résultats obtenus, la requête suivante trie les résultats par ordre croissant des noms :

```
1 SELECT name FROM country ORDER BY name ASC;
```

Celle-ci trie la liste par ordre décroissant des noms :

```
1 SELECT name FROM country ORDER BY name DESC;
```

À faire : Que produit cette requête ?

```
1 SELECT name,population FROM country ORDER BY population DESC;
```

Simulation

Exercice 1

Rédiger une requête SQL pour obtenir :

1. la liste des pays dont la population excède 60 000 000 d'habitants
2. la même liste triée par ordre alphabétique
3. la liste des pays et de leurs populations respectives, triée par ordre décroissant de leur superficie
4. la liste des pays ayant moins de 10 habitants au km²

Jointures

L'intérêt d'une base de données réside en particulier dans la possibilité de croiser des informations présentes dans plusieurs tables par l'intermédiaire d'une jointure.

Dans la base de données qui nous occupe on trouve une table nommée *encompassees* qui possède trois attributs : Country Continent Percentage

Le premier attribut d'un enregistrement est le code du pays, le deuxième le nom du continent et le dernier la portion du pays présente sur ce continent. La clé primaire de cette table est le couple (Country;Continent), et la valeur du troisième argument ne peut pas être nulle.

Cette seconde table possède un attribut en commun avec la première table : l'attribut Country de la table *encompassees* est identique à l'attribut Code de la table *country* et va nous permettre par son intermédiaire de croiser les informations de ces deux tables.

Le croisement de deux tables, revient à créer une table virtuelle contenant les informations des deux tables qui possède huit attributs :

NAME	CODE	CAPITAL	PROVINCE	AREA	POPULATION	COUNTRY	CONTINENT	PERCENTAGE
Bulgaria	BG	Sofia	Bulgaria	110910.	7284552	BG	Europe	100
Romania	RO	Bucuresti	Bucuresti	237500.	20121641	RO	Europe	100
Turkey	TR	Ankara	Ankara	780580.	75627384	TR	Asia	97
Turkey	TR	Ankara	Ankara	780580.	75627384	TR	Europe	3
Denmark	DK	Kobenhavn	Hovedstaden	43070.	5580516	DK	Europe	100

Un extrait de la table virtuelle créée

Méthode

Réaliser une jointure entre deux tables

Le principe d'une requête de jointure est : `SELECT * FROM table1 JOIN table2 ON expression_logique`

Par exemple cette requête affiche tous les attributs de la jointure des tables *country* et *encompassees* dont une fraction au moins est en Europe.

```
1 SELECT * FROM country JOIN encompassees ON country.code = encompassees.country WHERE
   encompassees.continent = 'Europe';
```

On peut l'écrire sur plusieurs ligne pour en faciliter la lecture...

```
1 SELECT * FROM country
2 JOIN encompassees ON country.code = encompassees.country
3 WHERE encompassees.continent = 'Europe';
```

Et celle-ci n'affiche que la liste des pays avec les mêmes contraintes :

```
1 SELECT country.name FROM country
2 JOIN encompassees ON country.code = encompassees.country
3 WHERE encompassees.continent = 'Europe';
```

À faire : Que produit cette requête ?

```
1 SELECT * FROM country
2 JOIN encompassees ON country.code = encompassees.country
3 WHERE encompassees.continent = 'Europe' AND encompassees.percentage < 100;
```

Simulation

Exercice 2

Rédiger une requête SQL pour obtenir :

1. le nom des pays qui sont à cheval sur plusieurs continents
2. les pays du continent américain qui comptent moins de 10 habitants par km²

Complément

Une jointure de 3 tables

Dans la base de données figure une table nommée *city* qui possède les attributs suivants : Name Country Province Population Longitude Latitude

Nous allons déterminer les capitales européennes situées à une latitude supérieure à 60, pour cela nous allons réaliser une jointure des trois tables *country* *city* et *encompassees*

Voici la requête :

```
1 SELECT country.name, country.capital FROM country
2 JOIN city ON country.code = city.country
3 JOIN encompassees ON country.code = encompassees.country
4 where encompassees.continent='Europe' AND city.name = country.capital AND city.latitude > 60
```

Simulation

Exercice 3

La table *Language* possède les attributs suivants : Country Name Percentage

L'attribut Country est le code du pays, Name le nom d'une langue parlée dans celui-ci, et Percentage la proportion d'habitants dont c'est la langue maternelle.

Rédiger une requête SQL pour obtenir :

1. les pays du monde où on parle le Français (French)
(on peut faire afficher le résultat d'un calcul : `SELECT ..., language.percentage*country.population,..FROM ..`)
2. les langues parlées par moins de 30 000 personnes dans le monde et leur pays respectif
3. la liste ordonnée pour le continent africain des langues (et du pays concerné), on précisera pour chacune d'entre elles le nombres de personnes qui la parlent.

Complément À faire

Dans la BDR mondial se trouve une table `economy` qui possède les attributs suivants : `Country` (le code du pays), `GDP` (le PIB, en millions de dollars), `agriculture` (la part de l'agriculture dans le PIB, en pourcentage), `Service` (la part des services dans le PIB), `Industry` (la part de l'industrie dans le PIB), `Inflation` (le taux d'inflation) et `Unemployment` (le taux de chômage).

Par exemple, la requête suivante affiche les pays majoritairement agricole.

```
1 SELECT country.name, economy.agriculture FROM country
2 JOIN economy ON country.code = economy.country
3 where economy.agriculture>economy.industry AND economy.agriculture > economy.service;
```

Imaginez une requête qui utilise cette nouvelle table, et réalisez là.