

## TP n°1

1. Créer une base *monde* ainsi qu'une base *societe*
2. Supprimer la base *societe*
3. Créer 3 tables dans la base *monde*

Une table *pays* permettant de stocker des pays

Nom de la colonne	Type	Divers
id	Entier	Clé primaire auto-incrémentée
nom	Texte	Non null et unique
population	Entier	Valeur par défaut : 1000
continent	Texte	Non null
annee_independance	Année	Nul autorisé

Une table *ville* permettant de stocker des villes

Nom de la colonne	Type	Divers
id	Entier	Clé primaire auto-incrémentée
id_pays	Entier	Clé étrangère sur l'id du pays
nom	Texte	Non null

Une table *langue* permettant de stocker des langues

Nom de la colonne	Type	Divers
id	Entier	Clé primaire auto-incrémentée
id_pays	Entier	Clé étrangère sur l'id du pays
nom	Texte	Non null
pourcentage	Decimal(5,2)	Null possible
groupe	Texte	Non null

4. Ajouter à la table *ville* une colonne *population* de type Entier en fin de table
5. Supprimer la colonne *groupe* de la table *langue*
6. Supprimer la table *ville*
7. Insérer quelques pays

France	67595000	Europe	
Mali	17994837	Afrique	1960
Inde	1251695584	Asie	1947

8. Insérer quelques langues pour le Mali

Bambara	46.3
Peul	9.2

## TP n°2

## MySQL, Prise en main

1. Créer la base world en important le fichier world.sql grâce à la commande suivante à exécuter dans le répertoire qui contient le fichier world.sql

```
mysql -u root -p -e "source world.sql"
```
2. Analyser les 3 tables de cette base grâce aux commandes `SHOW TABLES` et `DESC nom_table`
3. Effectuer les requêtes suivantes sans jointure
  1. En utilisant les tris, quels sont les 3 pays les plus peuplés ?
  2. En utilisant les tris, quel est le plus petit pays du monde ?
  3. Quel est le code pays du Canada ?
  4. Grâce au résultat précédent, quelles sont les villes du Canada triées par population décroissante ?
  5. Quels sont les pays dont le nom commence par 'Z' ?
  6. Quel est le PNB (GNP) de l'Ukraine en Hryvnia ? (1 USD = 35 UAH)
  7. Combien y a-t-il de pays dans le monde ?
  8. Quel est le PNB total du monde ?
  9. Quels sont les 3 pays les plus densément peuplés d'Asie (Asia) ?
  10. Quelle est la population de chaque continent du monde ?
4. Effectuer les requêtes suivantes avec des jointures internes
  1. Quelles sont les villes situées au Japon (Japan) ?
  2. Quels pays ont une ville nommée « Kingston » ?
  3. Quels sont les villes de plus de 5.000.000 d'habitants situées en Asie ?
  4. Quelle est la population totale des villes africaines ?
5. Effectuer les requêtes suivantes avec une requête imbriquée
  1. Quels sont les pays qui ont une population supérieure à la moyenne mondiale ?
  2. Quels sont les pays les plus grands de chaque région du monde ?
6. Effectuer les modifications suivantes dans la base
  1. Modifier la France en mettant « Emmanuel Macron » comme chef d'état (colonne HeadOfState)
  2. Ajouter 10% de population dans les pays d'Europe
  3. Supprimer les villes répondant au nom de Paris
  4. Supprimer les villes situées en Afrique

## TP n°3

1. Effectuer les opérations suivantes
  1. Désactiver le mode auto-commit
  2. Supprimer les villes ayant pour code pays 'FRA' (France)
  3. Supprimer les villes ayant pour code pays 'DEU' (Allemagne)
  4. Vérifier que les villes d'Allemagne et de France ont été supprimées
  5. Annuler les modifications
  6. Vérifier que les villes d'Allemagne et de France existent
  7. Supprimer les villes ayant pour code pays 'FRA' (France)

## MySQL, Prise en main

8. Valider les modifications et vérifier le résultat
  9. Activer le mode auto-commit
2. Effectuer les opérations suivantes
1. Désactiver le mode auto-commit
  2. Supprimer les villes ayant pour code pays 'CAN' (Canada)
  3. Créer un jalon
  4. Supprimer les villes ayant pour code pays 'CHN' (Chine)
  5. Vérifier que les villes du Canada et de Chine ont été supprimées
  6. Revenir au jalon créé à l'étape 3
  7. Valider les modifications et vérifier le résultat

### TP n°4

1. Créer une procédure stockée permettant d'augmenter la population de tous les pays d'un pourcentage donné en paramètre
2. Créer une procédure stockée qui calcule la superficie mondiale en utilisant les curseurs
3. Créer un trigger pour l'insertion d'un nouveau pays qui remplace la population de ce nouveau pays par 1.000.000.000 si la population donnée est supérieure à 1.000.000.000
4. Faire ce même trigger pour la mise à jour d'un pays. Ce trigger devra aussi vérifier que la différence de population n'est pas supérieure à 10%, auquel cas la population ne sera pas mise à jour

### TP n°5

1. Créer un utilisateur world avec un mot de passe ne pouvant se connecter que depuis la machine locale (localhost)
2. Créer un utilisateur world2 pouvant se connecter depuis n'importe où
3. Modifier le mot de passe de l'utilisateur world2
4. Donner à l'utilisateur world les droits SELECT et UPDATE sur la base world
5. Donner à l'utilisateur world2 tous les droits sur la base world
6. Tester l'utilisateur world
7. Tester l'utilisateur world2
8. Enlever le privilège UPDATE à l'utilisateur world
9. Afficher les droits de l'utilisateur world
10. Supprimer l'utilisateur world2

### TP n°6

1. Exporter la base world
2. Exporter uniquement les tables country et city de la base world
3. Importer la sauvegarde effectuée à l'étape 1

### Correction TP n°1

```
-- 1
CREATE DATABASE monde;
CREATE DATABASE societe;

-- 2
DROP DATABASE societe;

-- 3
USE monde;

CREATE TABLE pays (
  id INTEGER AUTO_INCREMENT PRIMARY KEY,
  nom VARCHAR(30) NOT NULL UNIQUE,
  population INTEGER DEFAULT 1000,
  continent VARCHAR(15) NOT NULL,
  annee_independance YEAR);

CREATE TABLE ville (
  id INTEGER AUTO_INCREMENT PRIMARY KEY,
  id_pays INTEGER NOT NULL,
  nom VARCHAR(30) NOT NULL,
  CONSTRAINT fk_id_pays_ville FOREIGN KEY(id_pays) REFERENCES pays(id));

CREATE TABLE langue (
  id INTEGER AUTO_INCREMENT PRIMARY KEY,
  id_pays INTEGER NOT NULL,
  nom VARCHAR(30) NOT NULL,
  pourcentage DECIMAL(5,2),
  groupe VARCHAR(15) NOT NULL,
  CONSTRAINT fk_id_pays_langue FOREIGN KEY(id_pays) REFERENCES pays(id));

-- 4
ALTER TABLE ville ADD population INTEGER;

-- 5
ALTER TABLE langue DROP groupe;

-- 6
DROP TABLE ville;

-- 7
INSERT INTO pays(nom, population, continent, annee_independance)
VALUES ('France', 67595000, 'Europe', NULL),
('Mali', 17994837, 'Afrique', 1960),
('Inde', 1251695584, 'Asie', 1947);

-- 8
INSERT INTO langue(id_pays, nom, pourcentage)
VALUES(2, 'Bambara', 46.3),
(2, 'Peul', 9.2);
```

### Correction TP n°2

```
-- 3-1 Chine, Inde et Etats-Unis
SELECT * FROM country
ORDER BY population DESC;

-- 3-2 Vatican
SELECT * FROM country
ORDER BY surfacearea ASC;

-- 3-3 CAN
SELECT * FROM country
WHERE name='Canada';

-- 3-4 Montréal, Calgary, Toronto,...
SELECT * FROM city
WHERE countrycode="CAN"
ORDER BY population DESC;

-- 3-5 Zambia et Zimbabwe
SELECT name FROM country
WHERE name LIKE 'Z%';

-- 3-6 1.475.880,00
SELECT gnp*35 as gnp_uah FROM country
WHERE name="Ukraine";

-- 3-7 239
SELECT COUNT(*) AS nombre FROM country;

-- 3-8 29.354.907,90
SELECT SUM(gnp) as total FROM country;

-- 3-9 Macao, Hong-Kong et Singapore
SELECT name FROM country
WHERE continent='Asia'
ORDER BY population/surfacearea DESC;

-- 3-10 Asie, 3.705.025.700 / Europe, 730.074.600,...
SELECT continent, SUM(population) as population
FROM country
GROUP BY continent;

-- 4-1 Tokyo, Jokohama, Osaka,...
SELECT city.name FROM city
INNER JOIN country ON city.countrycode = country.code
WHERE country.name = 'Japan';

-- 4-2 Jamaica et Norfolk Island
SELECT country.name FROM country
INNER JOIN city ON city.countrycode = country.code
WHERE city.name = 'Kingston';
```

## MySQL, Prise en main

```
-- 4-3 Shanghai, Peking, Chongqing...
SELECT city.name FROM city
INNER JOIN country ON city.countrycode = country.code
WHERE country.continent='Asia'
AND city.population > 5000000;

-- 4-4 135.838.579
SELECT SUM(city.population) as population FROM city
INNER JOIN country ON city.countrycode = country.code
WHERE country.continent='Africa';

-- 5-1 Argentina, Bangladesh, Brazil,...
SELECT name FROM country
WHERE population > (
    SELECT AVG(population) FROM country
);

-- 5-2 Antartica, Australia, Brazil, Canada, China,...
SELECT country.name, country.surfacearea FROM
(
    SELECT continent, MAX(SurfaceArea) as surface
    FROM country
    GROUP BY continent
) max_surface
INNER JOIN country
ON country.SurfaceArea = max_surface.surface
AND country.continent = max_surface.continent;

-- 6-1
UPDATE country SET HeadOfState='Emmanuel Macron' WHERE `Code`='FRA';

-- 6-2
UPDATE country SET population=population*1.1
WHERE continent='Europe';

-- 6-3
DELETE FROM city
WHERE name='Paris';

-- 6-4
DELETE city.* FROM city
INNER JOIN country ON city.countrycode = country.code
WHERE country.continent = 'Africa';
```

## Correction TP n°3

```
-- 1-1
SET autocommit=0;

-- 1-2
DELETE FROM city WHERE countrycode='FRA';

-- 1-3
DELETE FROM city WHERE countrycode='DEU';
```

## MySQL, Prise en main

```
-- 1-4
SELECT * FROM city WHERE countrycode='FRA';
SELECT * FROM city WHERE countrycode='DEU';

-- 1-5
ROLLBACK;

-- 1-6
SELECT * FROM city WHERE countrycode='FRA';
SELECT * FROM city WHERE countrycode='DEU';

-- 1-7
DELETE FROM city WHERE countrycode='FRA';

-- 1-8
COMMIT;
SELECT * FROM city WHERE countrycode='FRA';
SELECT * FROM city WHERE countrycode='DEU';

-- 1-9
SET autocommit=1;

-- 2-1
SET autocommit=0;

-- 2-2
DELETE FROM city WHERE countrycode='CAN';

-- 2-3
SAVEPOINT after_canada;

-- 2-4
DELETE FROM city WHERE countrycode='CHN';

-- 2-5
SELECT * FROM city WHERE countrycode='CAN';
SELECT * FROM city WHERE countrycode='CHN';

-- 2-6
ROLLBACK TO after_canada;

-- 2-7
COMMIT;
SELECT * FROM city WHERE countrycode='CAN';
SELECT * FROM city WHERE countrycode='CHN';
```

## Correction TP n°4

```
-- 1
DELIMITER //
CREATE PROCEDURE augmenter_population(IN pourcentage decimal)
BEGIN
    SET SQL_SAFE_UPDATES = 0;
    UPDATE country SET population = population * (1 + pourcentage/100);
    SET SQL_SAFE_UPDATES = 1;
END//
```

## MySQL, Prise en main

```
-- 2
DELIMITER //
CREATE PROCEDURE calc_superficie_mondiale()
BEGIN

    DECLARE superficie_mondiale DECIMAL(15,2) DEFAULT 0;
    DECLARE varea FLOAT(10,2);
    DECLARE fin_cursor BOOLEAN DEFAULT FALSE;
    DECLARE c_country CURSOR FOR SELECT SurfaceArea FROM country;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin_cursor = TRUE;

    OPEN c_country;

    boucle_country: LOOP
        FETCH c_country INTO varea;
        SET superficie_mondiale = superficie_mondiale + varea;

        IF fin_cursor THEN LEAVE boucle_country; END IF;
    END LOOP;

    CLOSE c_country;

    SELECT CONCAT('Superficie mondiale : ', superficie_mondiale) AS 'Résultat';

END//

-- 3
DELIMITER //
CREATE TRIGGER b_ins_country_max_pop
BEFORE INSERT ON country FOR EACH ROW
BEGIN
    IF NEW.population > 1000000000 THEN
        SET NEW.population = 1000000000;
    END IF;
END//

--4
DELIMITER //
CREATE TRIGGER b_upt_country_max_pop
BEFORE UPDATE ON country FOR EACH ROW
BEGIN
    IF NEW.population > 1000000000 THEN
        SET NEW.population = 1000000000;
    END IF;

    IF NEW.population > OLD.population * 1.1 OR NEW.population < OLD.population
    * 0.9 THEN
        SET NEW.population = OLD.population;
    END IF;
END//
```



### Correction TP n°5

```
-- 1
CREATE USER 'world'@'localhost' IDENTIFIED BY 'azerty';

-- 2
CREATE USER 'world2'@'%';

-- 3
SET PASSWORD FOR 'world2'@'%' = 'azerty';

-- 4
GRANT SELECT, UPDATE ON world.* TO 'world'@'localhost';

-- 5
GRANT ALL PRIVILEGES ON world.* TO 'world2'@'%';

-- 6 En se connectant avec l'utilisateur world
USE world;
SELECT * FROM country;
UPDATE country SET population = population-1;
DELETE FROM city WHERE population < 1000;

-- 7 En se connectant avec l'utilisateur world2
USE world;
SELECT * FROM country;
UPDATE country SET population = population-1;
DELETE FROM city WHERE population < 1000;

-- 8
REVOKE UPDATE ON world.* FROM 'world'@'localhost';

-- 9
SHOW GRANTS FOR 'world'@'localhost';

-- 10
DROP USER 'world2'@'%';
```

### Correction TP n°6

1. `mysqldump -u root -p world -r world.sql`
2. `mysqldump -u root -p world country city -r country.sql`
3. `mysql -u root -p world < world.sql`