TD - Séance 4

Part 1 - Creating a database

Question 1 : Create a database called « world »

```
CREATE DATABASE world;
```

Question 2 : Write a SQL statement to create a simple table `countries` including columns `country_id`, `country_name` and `region_id`, where the country is made of 1 to 3 characters, the name cannot exceed 255 characters and the region ID is an integer

```
CREATE TABLE countries (
    country_id CHAR(3) PRIMARY KEY,
    country_name VARCHAR(255) NOT NULL,
    region_id INT
);
```

Question 3 : Write a query to insert the country `France`, identified by the id `FRA`, with region id `1`.

```
INSERT INTO countries (country_id, country_name, region_id)
VALUES ('FRA', 'France', 1);
```

Question 4 : Write a query to insert the country `Mexico`, identified by the id `MEX`, no region specified

```
INSERT INTO countries (country_id, country_name, region_id)
VALUES ('MEX', 'Mexico', NULL);
```

Question 5 : Alter the table `countries` to add the column `country_id` to the primary key (Internet is your friend). You can use command line or the menus of the Workbench.

```
ALTER TABLE countries
ADD PRIMARY KEY (country_id);
```

Question 6 : Try to insert the country `France` as in question 3

```
INSERT INTO countries (country_id, country_name, region_id)
VALUES ('FRA', 'France', 1);
```

This will fail because country_id 'FRA' already exists, violating the primary key constraint.

Question 7 : Alter the table countries to set the `country_name` as not null

```
ALTER TABLE countries
MODIFY country_name VARCHAR(255) NOT NULL;
```

Question 8 : Create a table `region` made of two columns, `id` (an auto increment integer) and a mandatory column `region_name` as a string of maximum 60 caracters

```
CREATE TABLE region (
    id INT AUTO_INCREMENT PRIMARY KEY,
    region_name VARCHAR(60) NOT NULL
);
```

Question 9 : Alter the table `countries` to add a foreign key reference from `countries.region_id` to `region.id`. Why isn't it possible ? Find a way.

```
ALTER TABLE countries
ADD CONSTRAINT fk_region_id
FOREIGN KEY (region_id) REFERENCES region(id);
```

The `region_id` column in the `countries` table has values that exist in the `region` table. If not, the foreign key reference will fail because it cannot link invalid values.

We can solve it by ensuring all `region_id` values in `countries` exist in `region` before adding the foreign key constraint.

Question 10 : Write a query to select the country names and their region names

```
SELECT c.country_name, r.region_name
FROM countries c
JOIN region r ON c.region_id = r.id;
```

Question 11 : Complete manually the regions to obtain the following table content

```
INSERT INTO region (region_name)
VALUES ('Europe'), ('Asia'), ('North America'), ('South America'), ('Oceania'),
('Africa');
```

Question 12 : Clear the table countries, i.e. delete all rows

```
DELETE FROM countries;
```

Question 13 : Load the countries from the CSV File : `countries.csv` that is on Mootse

```
LOAD DATA INFILE '/path/to/countries.csv'
INTO TABLE countries
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
(country_id, country_name, region_id);
```

Question 14 : Load the SQL file `cities.sql` using the Workbench interface Tools->Import

This is done using the Workbench interface:

- Open Workbench
- Go to File > Open SQL Script and select cities.sql
- Click on the lightning bolt (Execute) to run the script

Question 15 : Create a query to select the pair of cities of the same country, so that the smallest city has at least 1M (million) population and the largest has at least 1M more population than the smallest one

```
SELECT c1.city_name AS city1, c2.city_name AS city2
FROM city c1
JOIN city c2 ON c1.CountryCode = c2.CountryCode
WHERE c1.population >= 1000000
AND c2.population >= c1.population + 1000000
AND c1.city_id < c2.city_id;
```

Question 16 : How long was the query ? Use the keyword `EXPLAIN` before your query to get information on the query plan. Take a screenshot of it

```
EXPLAIN SELECT c1.city_name AS city1, c2.city_name AS city2
FROM city c1
JOIN city c2 ON c1.CountryCode = c2.CountryCode
```

```
WHERE c1.population >= 1000000
AND c2.population >= c1.population + 1000000
AND c1.city_id < c2.city_id;
```

Question 17 : Alter the table `city` to add an index on `population` then reset the query cache with the command `RESET QUERY CACHE`. Execute the query 15 again, what do you observe on the execution time ? Use explain to get more information

```
ALTER TABLE city ADD INDEX (population);
RESET QUERY CACHE;
EXPLAIN SELECT c1.city_name AS city1, c2.city_name AS city2
FROM city c1
JOIN city c2 ON c1.CountryCode = c2.CountryCode
WHERE c1.population >= 1000000
AND c2.population >= c1.population + 1000000
AND c1.city_id < c2.city_id;
```

Part 2 - Export and import a sqldump

Question 18 : From the command line, export ONE database. How to choose between data and/or structure only ?

```
# Export only structure (schema)
mysqldump -u username -p --no-data world > world_structure.sql

# Export only data
mysqldump -u username -p --no-create-info world > world_data.sql

# Export both data and structure
mysqldump -u username -p world > world_full.sql
```

Question 19 : From the command line, import this fresh dump in another database. Which argument(s) do you need to use ?

```
mysql -u username -p new_database < world_full.sql
```

Question 20 : From the command line, export ALL databases. How may you use the result ? Which databases are missing ?

```
mysqldump -u username -p --all-databases > all_databases.sql
```

Question 21 : From the command line, import this fresh dump. What is the result ?

```
mysql -u username -p < all_databases.sql
```

Question 22 : Some times, keys block the data import. From the command line, which argument permits to bypass this behaviour ?

```
mysqldump -u username -p --disable-keys world > world_no_keys.sql
```

Question 23 : Congratulations, you have a good friend who will give you a sqldump. Import the file on your server instance. What are the risks ? What will you check ?

The risks include importing malicious data or incorrect structure, which could damage your database. Check:

- The authenticity and trustworthiness of the source
- The consistency and integrity of the data
- That it doesn't overwrite important data or structure

```
mysql -u username -p < dump_file.sql
```

Part 3 - User account

Question 24 : From the command line, how to add a new user with the full admin level on all databases ?

```
CREATE USER 'new_user'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'new_user'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Question 25 : From the command line, how to add a new user with the full admin level on one database only ?

```
CREATE USER 'new_user'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON database_name.* TO 'new_user'@'%';
FLUSH PRIVILEGES;
```

Question 26 : From the command line, how to add a new user with read only right on one database ?

```
CREATE USER 'readonly_user'@'%' IDENTIFIED BY 'password';
GRANT SELECT ON database_name.* TO 'readonly_user'@'%';
```

```
FLUSH PRIVILEGES;
```

Part 4 - Transaction

Question 27 : Determine the default isolation level for your SGBD instance

```
SHOW VARIABLES LIKE 'transaction_isolation';
```

Question 28 : Have two MySQL client connected : 1 Workbench and 1 PhpMyAdmin or 2 Workbench, or 1 Workbench and 1 Dbeaver...

To test transaction isolation, follow these steps:

- Open two clients, e.g., Workbench and PhpMyAdmin
- Perform operations like updating or deleting records in one client, and observe the behavior in the second client

Question 29 : First story : READ COMMITTED

On client 1, start a new transaction : SET TRANSACTION ISOLATION LEVEL READ COMMITTED; START TRANSACTION; SELECT * FROM city WHERE CountryCode='FRA'; → How many results ? DELETE FROM city WHERE CountryCode='FRA'; SELECT * FROM city WHERE CountryCode='FRA'; → How many results ?

On client 2 : SELECT * FROM city WHERE CountryCode='FRA'; → How many results ?

On client 1: COMMIT; OR ROLLBACK ; → As you wish, just agree with you and yourself ! SELECT * FROM city WHERE CountryCode='FRA'; → How many results ?

On client 2 : SELECT * FROM city WHERE CountryCode='FRA'; → How many results ?

```
-- Client 1:
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
START TRANSACTION;
SELECT * FROM city WHERE CountryCode='FRA';
-- How many results? (depends on the data)

DELETE FROM city WHERE CountryCode='FRA';
SELECT * FROM city WHERE CountryCode='FRA';
-- How many results?

-- Client 2:
SELECT * FROM city WHERE CountryCode='FRA';
-- How many results?

-- Client 1:
COMMIT;
SELECT * FROM city WHERE CountryCode='FRA';
-- How many results?
```

```
-- Client 2:
SELECT * FROM city WHERE CountryCode='FRA';
-- How many results?
```

Question 30 : Find in documentation and try the different level of isolation : REPEATABLE READ / READ COMMITTED / READ UNCOMMITTED / SERIALIZABLE

```
-- For each isolation level, run:
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
-- and observe the behavior of transactions.
```

Question 31 : What may you do with the two keywords GLOBAL and SESSION ?

```
SET GLOBAL transaction_isolation = 'READ COMMITTED';
SET SESSION transaction_isolation = 'SERIALIZABLE';
```

Part 5 - Deadlock

Question 32 : In on client :

START TRANSACTION; INSERT INTO test_table (message) VALUES ('test'); SELECT SLEEP(20); UPDATE test_table SET message = 'foo' WHERE (number = '1'); COMMIT;

```
START TRANSACTION;
INSERT INTO test_table (message) VALUES ('test');
SELECT SLEEP(20);
UPDATE test_table SET message = 'foo' WHERE (number = '1');
COMMIT;
```

Question 33 : In another client :

This would depend on how you set up deadlock conditions, which can be tested based on conflicting transactions (updating the same row or locking resources).