Report TP1 SQL :

#1. Create a new database named Supplier.

```
CREATE DATABASE Supplier;
```

#2. Create tables with primary keys.
#3Create the foreign key in each table.
#4Build the schema for this database.

```
CREATE TABLE Customer (
  Id_client INT PRIMARY KEY,
  Last_name VARCHAR(255) NOT NULL,
  First_name VARCHAR(255) NOT NULL,
  address VARCHAR(255) NOT NULL,
  city VARCHAR(255) NOT NULL,
  age INT NOT NULL
);

CREATE TABLE Product (
  Id_prod INT PRIMARY KEY,
  nomProd VARCHAR(255) NOT NULL,
  Qnty_Stock INT NOT NULL
);

CREATE TABLE Ordered (
  Id_com INT PRIMARY KEY,
  date_com DATE NOT NULL,
  Quantity INT NOT NULL,
  Id_customer INT NOT NULL,
  Id_prod INT NOT NULL,
  FOREIGN KEY (Id_customer) REFERENCES Customer (Id_client),
  FOREIGN KEY (Id_prod) REFERENCES Product (Id_prod)
);
```

#5 Fill the tables (5 records are enough). (Age 20, 40,50, 60) (city: rabat/casa/ Fes)

```
INSERT INTO Customer (Id_client, Last_name, First_name, address, city, age)
VALUES (1, 'Smith', 'John', '123 Main St', 'Rabat', 20),
       (2, 'Johnson', 'Jane', '456 Park Ave', 'Casa', 40),
       (3, 'Williams', 'Bob', '789 Maple St', 'Fes', 50),
       (4, 'Jones', 'Alice', '321 Oak St', 'Rabat', 60);
```

#6. Apply a list of choices on the name of the products (“computer”, “mouse”, “keyboard”).

```
ALTER TABLE Product
ADD CONSTRAINT nomProd_check CHECK (nomProd IN ('computer', 'mouse', 'keyboard'));
```

#7. Show all customers over the age of 30.

```
SELECT * FROM Customer WHERE age > 30;
```

#8. Display by a command the number of rows you have filled in each table.

```
SELECT COUNT(*) FROM Product;
SELECT COUNT(*) FROM Ordered;
SELECT COUNT(*) FROM Customer;
```

#9. Display the customer whose name begins with "A".

```
SELECT * FROM Customer WHERE First_name LIKE 'A%';
```

#10. Select customers whose age = 50 and name contains “M".

```
SELECT * FROM Customer WHERE age = 50 AND First_name LIKE '%M%';
```

#11. Display the list of products from largest quantity to smallest quantity. 1p

```
SELECT * FROM Product ORDER BY Qnty_Stock DESC;
```

#12. Give the orders that date from 2000.

```
SELECT * FROM Ordered WHERE date_com >= '2000-01-01' AND date_com < '2001-01-01';
```

#13. Give orders for the month of December. (If they don't exist, add a new record, then test).

```
INSERT INTO Ordered (Id_com, date_com, Quantity, Id_customer, Id_prod)
VALUES (1, '2020-12-01', 5, 1, 1);

SELECT Id_com, date_com, Quantity FROM Ordered WHERE MONTH(date_com) = 12;
```

#14. Display the average order quantity, maximum order quantity, and minimum order quantity.

```
SELECT AVG(Quantity) AS 'Average Quantity' FROM Ordered;
SELECT MAX(Quantity) AS 'Maximum Quantity' FROM Ordered;
SELECT MIN(Quantity) AS 'Minimum Quantity' FROM Ordered;
```

#15. Modify the quantity of the first product by putting 15 in place of the existing quantity.

```
UPDATE Product SET Qnty_Stock = 15 WHERE Id_prod = 1;
```

#16. Multiply the quantity of the products by two.

```
UPDATE Product SET Qnty_Stock = Qnty_Stock * 2 WHERE Id_prod = 1;
```

#17. Delete the address column from the customer table.

```
ALTER TABLE Customer
DROP COLUMN address;
```

#18. Give customers who ordered a keyboard.

```
SELECT c.Id_client, c.Last_name, c.First_name
FROM Customer c
JOIN Ordered o ON c.Id_client = o.Id_customer
JOIN Product p ON o.Id_prod = p.Id_prod
WHERE p.nomProd = 'Keyboard';
```