

RESULTAT DES TESTS

```
===== test session starts =====
platform darwin -- Python 3.6.2, pytest-5.2.2, py-1.8.0, pluggy-0.13.0
rootdir: /Users/emiliendurif/Dropbox/cpge/ipt_mpsi_ds/DS08/correction_auto/script/DS_08_mostefaoui
plugins: timeout-1.3.4
timeout: 60.0s
timeout method: signal
timeout func_only: False
collected 22 items
```

```
test_TP.py F...F...FF.F.F....FFFF [100%]
```

```
===== FAILURES =====
_____ test_Q1_req _____
```

```
def test_Q1_req ():
>     assert requete(sol_Q1_req) == requete(Q1_req)
E       AssertionError: assert [('DUPONT', '...', 'M.')] == [('M.', 'DUPO..
.'Alain'), ...]
E         At index 0 diff: ('DUPONT', 'Alain', 'M.') != ('M.', 'DUPONT', 'Alain'
)
E         Use -v to get the full diff
```

```
test_TP.py:98: AssertionError
```

```
_____ test_Q4_req _____
```

```
def test_Q4_req ():
>     assert requete(sol_Q4_req) == requete(Q4_req)
E       AssertionError: assert [('DUHAMEL', ... 'Mme.')] == [('Melle.', '..
.onique'), ...]
E         At index 0 diff: ('DUHAMEL', 'Evelyne', 'Melle.') != ('Melle.', 'DUHAM
EL', 'Evelyne')
E         Use -v to get the full diff
```

```
test_TP.py:110: AssertionError
```

```
_____ test_Q7_req _____
```

```
def test_Q7_req ():
>     assert requete(sol_Q7_req) == requete(Q7_req)
```

```
test_TP.py:122:
```

```
-----
req = "CREATE TABLE CLI_TEL AS SELECT T_CLIENT.CLI_NOM,T_TELEPHONE.TEL_NUMERO,T_
TELEPHONE.TEL_LOCALISATION FROM T_CLIENT,T_TELEPHONE WHERE TYP_CODE='TEL' AND T_
CLIENT.CLI_ID=T_TELEPHONE.CLI_ID;"
```

```
def requete (req):
    conn = sqlite3.connect('../bdd/hotel_'+convert(alpha)+' .db')
    c=conn.cursor()
>     c.execute(req)
E       sqlite3.OperationalError: table CLI_TEL already exists
```

```
test_TP.py:25: OperationalError
```

```
_____ test_Q8_req _____
```

```
def test_Q8_req ():
>     assert requete(sol_Q8_req) == requete(Q8_req)
```

```
test_TP.py:125:
```

```
-----
req = 'SELECT CLI_NOM, as nb_occ FROM T_CLIENT WHERE COUNT(CLI_NOM) > 1;'
```

```
def requete (req):
    conn = sqlite3.connect('../bdd/hotel_'+convert(alpha)+' .db')
```

```
c=conn.cursor()
> c.execute(req)
E sqlite3.OperationalError: near "as": syntax error
```

```
test_TP.py:25: OperationalError
```

```
test_Q10_res
```

```
def test_Q10_res ():
> assert sol_Q10_res.upper() == Q10_res.upper()
E AssertionError: assert '(82.0, 116.0)' == '82.0,116.0'
E - (82.0, 116.0)
E ? - -
E + 82.0,116.0
```

```
test_TP.py:131: AssertionError
```

```
test_Q11_res
```

```
def test_Q11_res ():
> assert sol_Q11_res.upper() == Q11_res.upper()
E AssertionError: assert '(82.0, 116)' == '82,116'
E - (82.0, 116)
E + 82,116
```

```
test_TP.py:137: AssertionError
```

```
test_Q15_res
```

```
def test_Q15_res ():
> assert sol_Q15_res.upper() == Q15_res.upper()
E assert "('MEDARD', 'JACQUES', 1392)" == 'NOM,PRENOM,MONTANT'
E - ('MEDARD', 'JACQUES', 1392)
E + NOM,PRENOM,MONTANT
```

```
test_TP.py:152: AssertionError
```

```
test_Q15_req
```

```
def test_Q15_req ():
> assert requete(sol_Q15_req) == requete(Q15_req)
E AssertionError: assert [('MEDARD', 'Jacques', 1392)] == []
E Left contains one more item: ('MEDARD', 'Jacques', 1392)
E Use -v to get the full diff
```

```
test_TP.py:155: AssertionError
```

```
test_Q15_res2
```

```
def test_Q15_res2 ():
> assert sol_Q15_res2.upper() == Q15_res.upper()
E assert "('GARREAU', 'PAUL', 22)" == 'NOM,PRENOM,MONTANT'
E - ('GARREAU', 'PAUL', 22)
E + NOM,PRENOM,MONTANT
```

```
test_TP.py:158: AssertionError
```

```
test_Q15_req2
```

```
def test_Q15_req2 ():
> assert requete(sol_Q15_req2) == requete(Q15_req)
E AssertionError: assert [('GARREAU', 'Paul', 22)] == []
E Left contains one more item: ('GARREAU', 'Paul', 22)
E Use -v to get the full diff
```

```
test_TP.py:161: AssertionError
```

```
===== 10 failed, 12 passed in 0.75s =====
```

```
NOM = "MOSTEFAOUI"  
Prenom = "Rayan"  
Classe = "MPSI2"  
alpha="67"
```

```
## Question 1
```

```
Q1_req = "SELECT TIT_CODE,CLI_NOM,CLI_PRENOM FROM T_CLIENT;"
```

```
## Question 2
```

```
Q2_req = "SELECT COUNT(*) as nbclients FROM T_CLIENT;"
```

```
Q2_res = "89"
```

```
## Question 3
```

```
Q3_req = "SELECT CLI_NOM,CLI_PRENOM FROM T_CLIENT WHERE TIT_CODE='Mme.';"
```

```
## Question 4
```

```
Q4_req = "SELECT TIT_CODE,CLI_NOM,CLI_PRENOM FROM T_CLIENT WHERE TIT_CODE='Mme.'  
OR TIT_CODE='Melle.';"
```

```
## Question 5
```

```
Q5_req = "SELECT COUNT(*) as nbclientes FROM (SELECT TIT_CODE,CLI_NOM,CLI_PRENOM  
FROM T_CLIENT WHERE TIT_CODE='Mme.' OR TIT_CODE='Melle.');
```

```
Q5_res = "17"
```

```
## Question 6
```

```
Q6_req = "SELECT CLI_NOM as Noms, CLI_PRENOM as Prénoms FROM (SELECT TIT_CODE,CL  
I_NOM,CLI_PRENOM FROM T_CLIENT WHERE TIT_CODE='Mme.' OR TIT_CODE='Melle.') ORDER  
BY Noms ASC, Prénoms ASC;"
```

```
## Question 7
```

```
Q7_req = "CREATE TABLE CLI_TEL AS SELECT T_CLIENT.CLI_NOM,T_TELEPHONE.TEL_NUMERO  
,T_TELEPHONE.TEL_LOCALISATION FROM T_CLIENT,T_TELEPHONE WHERE TYP_CODE='TEL' AND  
T_CLIENT.CLI_ID=T_TELEPHONE.CLI_ID;"
```

```
## Question 8
```

```
Q8_req = "SELECT CLI_NOM, as nb_occ FROM T_CLIENT WHERE COUNT(CLI_NOM) > 1;"
```

```
## Question 9
```

```
Q9_req = "SELECT CLI_NOM, COUNT(*) as nb_occ FROM T_CLIENT GROUP BY CLI_NOM HAVI  
NG nb_occ > 1;"
```

```
Q9_res = "MARTIN,2"
```

```
## Question 10
```

```
Q10_req = "SELECT AVG(LIF_REMISE_POURCENT) as moyenne_remise,AVG(LIF_REMISE_MONT  
ANT) as moyenne_montant FROM T_LIGNE_FACTURE;"
```

```
Q10_res = "82.0,116.0"
```

```
## Question 11
```

```
Q11_req = "SELECT MAX(LIF_REMISE_POURCENT) as moyenne_remise,MAX(LIF_REMISE_MONT  
ANT) as moyenne_montant FROM T_LIGNE_FACTURE;"
```

```
Q11_res = "82,116"
```

```
## Question 12
```

```
Q12_req = "SELECT DISTINCT T_LIGNE_FACTURE.FAC_ID as fac_id1 FROM T_LIGNE_FACTUR  
E WHERE LIF_REMISE_MONTANT NOTNULL OR LIF_REMISE_POURCENT NOTNULL;"
```

```
## Question 13
```

```
Q13_req = "SELECT DISTINCT T_FACTURE.CLI_ID FROM T_FACTURE JOIN T_LIGNE_FACTURE  
ON T_FACTURE.FAC_ID=T_LIGNE_FACTURE.FAC_ID WHERE LIF_REMISE_MONTANT NOTNULL OR L  
IF_REMISE_POURCENT NOTNULL;"
```

```
## Question 14
```

```
Q14_req = "SELECT T_FACTURE.CLI_ID FROM T_FACTURE EXCEPT SELECT DISTINCT T_FACTU  
RE.CLI_ID FROM T_FACTURE JOIN T_LIGNE_FACTURE ON T_FACTURE.FAC_ID=T_LIGNE_FACTUR  
E.FAC_ID WHERE LIF_REMISE_MONTANT NOTNULL OR LIF_REMISE_POURCENT NOTNULL;"
```

```
## Question 15  
Q15_req = ""  
Q15_res ="nom,prenom,montant"
```