

Devoir Surveillé



Eric BERTHOMIER
eric.berthomier@free.fr

29 mars 2017

Première partie

Année 2015

Devoir Surveillé- Le Retour des Volets Roulants

Consignes

- l’anonymat entraîne l’anonymisation de la note (0)
- le brouillon n’est pas une option, toute copie ressemblant à un brouillon ne sera pas corrigée (0)
- les commentaires ne sont pas des fioritures
- la clarté et la précision des schémas sont essentielles
- lorsqu’il vous est demandé d’écrire une requête SQL, il vous est demandé de :
 1. donner la requête
 2. donner le résultat de la requête
- il vous est possible de réaliser votre devoir sur traitement de texte. Les schémas pourront quant à eux être donnés sous forme manuscrite.

Intitulé du devoir

L’entreprise TuxFactory vient d’équiper tout son département administratif de fenêtres avec volets roulants pilotables à distance. Elle souhaite ainsi faire des économies d’énergie en pilotant intelligemment les volets roulants lorsque les bureaux sont vides.

Son idée est d’essayer de conserver une température interne du bâtiment de 20°C et de limiter ainsi l’usage de la climatisation ou du chauffage.

Caractéristiques d’un volet roulant

Un volet roulant se caractérise par les informations suivantes :

- Hauteur de tablier
- Largeur de tablier
- Numéro de série (16 caractères)
- Nom du fabricant du volet
- Une localisation du volet roulant

Il n’existe que 3 fabricants de volets roulants avec lesquels TuxFactory travaille :

- centflap
- tuxflap
- wesflap

sachant qu’une majorité de volets sont de la marque tuxflap.

Un moteur de volet roulant se caractérise par les informations suivantes :

- Marque du moteur
- Numéro de série du moteur
- Date d’achat du moteur
- Adresse IP de l’interface du moteur

Il n’existe que 4 fabricants de moteur de volets roulants avec lesquels TuxFactory travaille :

- winmotor
- gnumotor
- debianmotor
- tuxmotor

sachant que la plupart des volets actuels sont de la marque tuxmotor.

Le **volet roulant** est couplé à un capteur thermique qui permet d’obtenir la température extérieure et intérieure sur simple demande via le réseau domotique informatique.

Volets roulants (7 points)

Exercice 1 (3 points) - Modélisation

Modéliser le problème sous forme d'une base de données. Vous prendrez soin à bien contrôler que vous avez repris l'ensemble des informations données et à éviter toute redondance. Il est possible de procéder en 2 étapes, une première sans prise en compte des contraintes entre les tables, une seconde en les prenant en compte.

Remarque

✓ **Après un délai incompressible de 30 minutes**, si vous n'arrivez pas à créer ce modèle, celui-ci vous sera fourni par votre examinateur mais aucun point ne vous sera accordé pour cet exercice

Exercice 2 (2 points) - Insertion

Intégrer les données sus-citées dans votre base avec la méthode de votre choix.¹

Menuiserie 1	
Volet roulant	
Hauteur (cm)	135
Largeur (cm)	120
Numero de serie	1285564100600149
Nom du fabricant	tuxflap
Localisation	Stockage petit materiel
Moteur	
Marque	tuxmotor
Numéro de série	TU30008U510256F
Date d'achat	12/03/2003
Adresse IP	192.168.1.128

Menuiserie 2	
Volet roulant	
Hauteur (cm)	200
Largeur (cm)	240
Numero de serie	1285564100600150
Nom du fabricant	tuxflap
Localisation	Local Stockage
Moteur	
Marque	tuxmotor
Numéro de série	TU30008U510245F
Date d'achat	12/03/2003
Adresse IP	192.168.1.129

Menuiserie 3	
Volet roulant	
Hauteur (cm)	200
Largeur (cm)	240
Numero de serie	1285564100600151
Nom du fabricant	tuxflap
Localisation	Local Stockage
Moteur	
Marque	tuxmotor
Numéro de série	TU30008U510232F
Date d'achat	12/03/2003
Adresse IP	192.168.1.145

Menuiserie 4	
Volet roulant	
Hauteur (cm)	200
Largeur (cm)	240
Numero de serie	1285564100600152
Nom du fabricant	tuxflap
Localisation	Local Stockage
Moteur	
Marque	tuxmotor
Numéro de série	TU30008U510256E
Date d'achat	07/04/2005
Adresse IP	192.168.1.171

1. Un fichier texte contenant toutes les données est à votre disposition.

Menuiserie 5	
Volet roulant	
Hauteur (cm)	135
Largeur (cm)	120
Numero de serie	01101234545
Nom du fabricant	centflap
Localisation	Stockage petit materiel
Moteur	
Marque	winmotor
Numéro de série	WI30008U519024F
Date d'achat	17/05/2006
Adresse IP	192.168.1.173

Menuiserie 6	
Volet roulant	
Hauteur (cm)	155
Largeur (cm)	150
Numero de serie	01101234546
Nom du fabricant	centflap
Localisation	Salle de Preparation
Moteur	
Marque	gnumotor
Numéro de série	V3309-8CMA002
Date d'achat	19/07/2012
Adresse IP	192.168.1.3

Menuiserie 7	
Volet roulant	
Hauteur (cm)	120
Largeur (cm)	100
Numero de serie	320AG5B7320
Nom du fabricant	wesflap
Localisation	Bureau 1
Moteur	
Marque	debmotor
Numéro de série	U1-93-02-00121
Date d'achat	01/03/2014
Adresse IP	192.168.1.7

Menuiserie 8	
Volet roulant	
Hauteur (cm)	120
Largeur (cm)	100
Numero de serie	320AG5B7321
Nom du fabricant	wesflap
Localisation	Bureau 2
Moteur	
Marque	debmotor
Numéro de série	U1-93-02-00122
Date d'achat	01/05/2011
Adresse IP	192.168.1.185

Menuiserie 9	
Volet roulant	
Hauteur (cm)	120
Largeur (cm)	100
Numero de serie	320AG5B7322
Nom du fabricant	wesflap
Localisation	Secretariat
Moteur	
Marque	debmotor
Numéro de série	U1-93-02-00123
Date d'achat	13/08/2003
Adresse IP	192.168.1.172

Menuiserie 10	
Volet roulant	
Hauteur (cm)	65
Largeur (cm)	60
Numero de serie	01101234548
Nom du fabricant	centflap
Localisation	Vestiaire
Moteur	
Marque	gnumotor
Numéro de série	V3307-8CMA001
Date d'achat	13/09/2007
Adresse IP	192.168.1.178

Exercice 3 (2 points) - Requêtes

Écrire les différentes requêtes associées à ses assertions.

1. Donner la liste des fabricants de volets roulants (0,5 points)
2. Donner la liste des marques de moteur dont la date d'achat est supérieure au 1er Janvier 2014 (0,75 points)
3. Donner la liste des adresses IP associées au numéro de série des volets triées de de la plus petite à la plus grande (0,75 point)

Exercice 4 (3 points) - Programmation

Écrire un programme Python qui permet d'afficher les volets roulants **avec leur localisation** qui ne répondent pas sur le réseau (commande ping).

Le programme devra afficher la listes des adresses IP des volets roulants sous la forme suivante (sans la mise en page) :

IP,Numéro de série,Localisation du volet,is up / is down !

En Python, il est possible de tester la réponse d'une adresse ip par le code suivant :

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import os

ip = "192.168.1.150"
response = os.system("ping -c 1 " + ip)

if response == 0:

    print ip, 'is up!'

else:

    print ip, 'is down!'
```

Quel peut être l'intérêt d'avoir affiché les résultats sous la forme xxx,yyy,zzz ?

Aide

✎ Afin de tester votre programme, changer, **de manière temporaire** l'adresse IP d'un volet pour celle d'une machine existante.

ipconfig sous Windows), ifconfig sous Linux).

Températures (2 points)

Exercice 1 (1 point) - Insertion

Le capteur numéro 1 associé au volet roulant avec le numéro de série : 320AG5B7320 a remonté les températures suivantes :

19 Avril 2015		
Heure	Température Extérieure	Température Intérieure
06h00	10.8°C	17,7°C
08h00	14.1°C	17,7°C
12h00	18.5°C	18°C
13h00	17.8°C	17.8°C
15h00	16.8°C	18.1°C
17h00	14.4°C	18.2°C
19h00	11.2°C	18.4°C
22h00	6.2°C	17.7°C

Le capteur numéro 2 associé au volet roulant avec le numéro de série : 320AG5B7321 a remonté des températures identiques.

Insérer ces données dans votre base de données.

Exercice 2 (1 point) - Calculs

Traduire ces assertions en requêtes SQL :

- Quelles sont les températures minimales et maximales intérieures pour le volet roulant avec le numéro de série : 320AG5B7320? Les valeurs n'apparaîtront pas forcément telles que définies selon votre définition de champs.
- Quelle est la température moyenne extérieure pour ce volet?

Dépannage (5 points)

Contente de votre travail sur l'énergie TuxFactory vous demande de réaliser une traçabilité des interventions de dépannage. En effet quand un volet est en panne, la régulation énergétique ne peut se faire, de l'argent est donc perdu. TuxFactory ne travaille qu'avec un seul prestataire titulaire du marché.

- Un dépannage peut se faire ou non sous garantie.
- Seule, la date du dépannage est enregistrée.
- Une description sommaire du dépannage doit être noté

Exercice 1 (1 point) - Modélisation

Modéliser le problème.

Exercice 2 (1 point) - Insertion

Insérer les dépannages suivants.

Date	Numéro de série du volet	Description	Garantie
11/08/2005	1285564100600149	Deblocage volet roulant	N
12/01/2006	1285564100600149	Réagrafage tablier	N
12/07/2010	1285564100600149	Changement de genouillère	N
01/02/2015	320AG5B7320	Changement de treuil	O
13/04/2015	1101234545	Changement du moteur	O

TABLE 1.1 : Liste de dépannages

Exercice 3 (3 points) - Remplacement

Sauvegarder votre base de données!

Le dernier dépannage a nécessité le remplacement du moteur par un nouveau moteur de la marque winmotor dont le numéro de série est : WI30008U9113084F.

Écrire l'ensemble des requêtes nécessaires à cette action.

Donner l'ensemble des moteurs qui ont été utilisés pour ce volet.

Exercice 4 (1 point) - Requête évoluée

Quel est le numéro de série du volet qui a le plus de dépannage?



FIGURE 1.1 – Durée de l'épreuve : **6h00**

Bilan - Corrigé Devoir - Le Retour des Volets Roulants

Volets roulants (7 points)

Exercice 1 (4 points) - Modélisation

```
DROP DATABASE IF EXISTS voletroulant;
CREATE DATABASE voletroulant;
USE voletroulant;

--
-- Structure de la table `moteur`
--

CREATE TABLE `moteur` (
  `moteur_sn` varchar(20) NOT NULL,
  `marque` enum('tuxmotor','winmotor','gnumotor','debmotor') NOT NULL,
  `moteur_date_achat` date NOT NULL,
  `moteur_ip` varchar(15) NOT NULL,
  `fk_volet` varchar(20) NOT NULL,
  PRIMARY KEY (`moteur_sn`)
);

--
-- Structure de la table `temperature`
--

CREATE TABLE `temperature` (
  `id_capteur` int NOT NULL,
  `t_interieure` float NOT NULL,
  `t_exterieure` float NOT NULL,
  `quand` DATETIME NULL,
  `fk_volet` varchar(16) NOT NULL
);

--
-- Structure de la table `volet`
--

CREATE TABLE `volet` (
  `volet_sn` varchar(20) NOT NULL,
  `hauteur` int NOT NULL COMMENT 'en cm',
  `largeur` int NOT NULL COMMENT 'en cm',
  `fabriquant` enum('centflap','tuxflap','wesflap') NOT NULL DEFAULT 'tuxflap',
  `localisation` text NOT NULL,
  PRIMARY KEY (`volet_sn`)
);

-- Creation des contraintes liees aux cles etrangeres
ALTER TABLE `moteur` ADD FOREIGN KEY (fk_volet) REFERENCES volet (volet_sn);
ALTER TABLE `temperature` ADD FOREIGN KEY (fk_volet) REFERENCES volet (volet_sn);
```

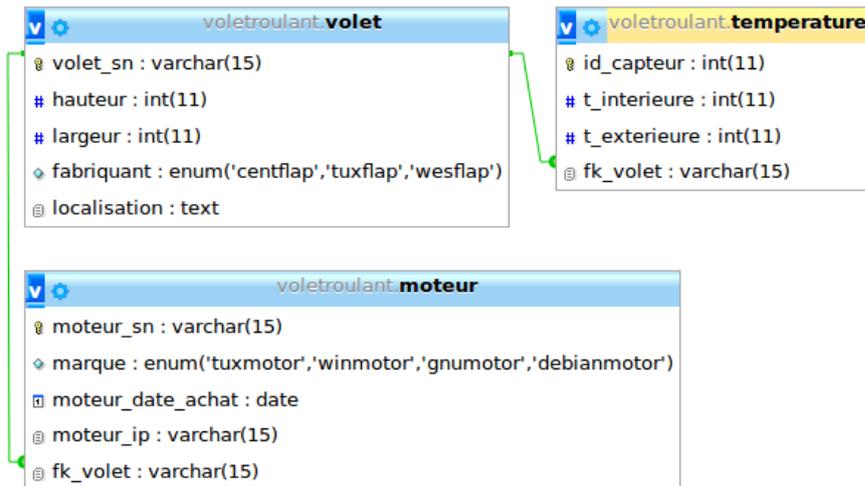


FIGURE 2.1 – Schéma de la base de données "VoletRoulant"

Exercice 2 (2 points) - Insertion

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import sys

def lire_ligne (f):
    valeur = f.readline()
    return valeur.rstrip()

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                    user="vr", # utilisateur
                    passwd="vr", # mot de passe
                    db="voletroulant") # nom de la base de données

# Traitement des requêtes aussitôt
conn.autocommit(True)

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes
cur = conn.cursor()

# On ouvre le fichier "garniture.txt"
f = open('volets_roulants.txt', 'r')

cur.execute("DELETE FROM moteur")
cur.execute("DELETE FROM volet")

for i in range (10):
    hauteur=lire_ligne(f)
    largeur=lire_ligne(f)
    snv=lire_ligne(f)
    fab=lire_ligne(f)
    loc=lire_ligne(f)

    marque=lire_ligne(f)
    snm=lire_ligne(f)
    dachat=lire_ligne(f)
    ip=lire_ligne(f)

    print ("INSERT INTO volet VALUES (%s,%s,%s,%s,%s)"% (snv, hauteur, largeur, fab, loc))

    try:
        cur.execute("INSERT INTO volet VALUES (%s,%s,%s,%s,%s)", (snv, hauteur, largeur, fab, loc))

    except mdb.Error, e:
        print "Erreur %d: %s" % (e.args[0],e.args[1])
        sys.exit(1)
```

```

print ("INSERT INTO moteur VALUES (\'%s\',\'%s\',%s,\'%s\',\'%s\')"% (snm, marque, dachat, ip, snv))

try:
    cur.execute("INSERT INTO moteur VALUES (\'%s\',\'%s\',\'%s\',\'%s\',\'%s\')"% (snm, marque, dachat, ip, snv))

except mdb.Error, e:
    print "Erreur %d: %s" % (e.args[0],e.args[1])
    sys.exit(1)

cur.close()
conn.close()

f.close ()

```

Exercice 3 (2 points) - Requêtes

```

mysql> select fabricant from volet group by fabricant;
+-----+
| fabricant |
+-----+
| centflap |
| tuxflap |
| wesflap |
+-----+
3 rows in set (0.01 sec)

```

```

mysql> select marque from moteur where moteur_date_achat > '2014-01-01' group by marque;
+-----+
| marque |
+-----+
| debmotor |
+-----+
1 row in set (0.00 sec)

```

```

mysql> select moteur_ip,moteur_sn from moteur order by moteur_ip desc;
+-----+-----+
| moteur_ip | moteur_sn |
+-----+-----+
| 192.168.1.7 | U1-93-02-00121 |
| 192.168.1.3 | V3309-8CMA002 |
| 192.168.1.185 | U1-93-02-00122 |
| 192.168.1.178 | V3307-8CMA001 |
| 192.168.1.173 | WI30008U519024F |
| 192.168.1.172 | U1-93-02-00123 |
| 192.168.1.171 | TU30008U510256E |
| 192.168.1.145 | TU30008U510232F |
| 192.168.1.129 | TU30008U510245F |
| 192.168.1.128 | TU30008U510256F |
+-----+-----+
10 rows in set (0.00 sec)

```

Exercice 4 (3 points) - Programmation

```

#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import MySQLdb.cursors
import sys
import os

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                    user="vr", # utilisateur
                    passwd="vr", # mot de passe
                    db="voletroulant", # nom de la base de données
                    cursorclass=MySQLdb.cursors.DictCursor # Dictionnaire de données
                )

# Traitement des requêtes aussitôt
conn.autocommit(True)

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes

```

```

cur = conn.cursor()

try:
    cur.execute("SELECT * FROM moteur")

except mdb.Error, e:
    print "Erreur %d: %s" % (e.args[0],e.args[1])
    sys.exit(1)

rows = cur.fetchall()

for row in rows:

    # On recupere le commentaire du volet roulant
    cur2 = conn.cursor()

    try:
        cur2.execute("SELECT * FROM volet WHERE volet_sn=%s", row['fk_volet'])

    except mdb.Error, e:
        print "Erreur %d: %s" % (e.args[0],e.args[1])
        sys.exit(1)

    row2 = cur2.fetchone()

    hostname = row['moteur_ip']
    response = os.system("ping -c 1 " + hostname)

    if response == 0:
        print (row['moteur_ip'] + "," + row['moteur_sn'] + "," + row2['localisation'] + ",up")
    else:
        print (row['moteur_ip'] + "," + row['moteur_sn'] + "," + row2['localisation'] + ",down")

cur.close()
conn.close()

```

Températures (2 points)

Exercice 1 (1 point) - Insertion

```

INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '10.8', '17.7', '320AG5B7320', '2015-04-19 06:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '14.1', '17.7', '320AG5B7320', '2015-04-19 08:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '18.5', '18.0', '320AG5B7320', '2015-04-19 12:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '17.8', '17.8', '320AG5B7320', '2015-04-19 13:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '16.8', '18.1', '320AG5B7320', '2015-04-19 15:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '14.4', '18.2', '320AG5B7320', '2015-04-19 17:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '11.2', '18.4', '320AG5B7320', '2015-04-19 19:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
1', '6.2', '17.7', '320AG5B7320', '2015-04-19 22:00:00');

INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
2', '10.8', '17.7', '320AG5B7321', '2015-04-19 06:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
2', '14.1', '17.7', '320AG5B7321', '2015-04-19 08:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
2', '18.5', '18.0', '320AG5B7321', '2015-04-19 12:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
2', '17.8', '17.8', '320AG5B7321', '2015-04-19 13:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
2', '16.8', '18.1', '320AG5B7321', '2015-04-19 15:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
2', '14.4', '18.2', '320AG5B7321', '2015-04-19 17:00:00');
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('
2', '11.2', '18.4', '320AG5B7321', '2015-04-19 19:00:00');

```

```
INSERT INTO `voletroulant`.`temperature` (`id_capteur`, `t_interieure`, `t_exterieure`, `fk_volet`, `quand`) VALUES ('2', '6.2', '17.7', '320AG5B7321', '2015-04-19 22:00:00');
```

Exercice 2 (1 point) - Calculs

```
mysql> SELECT min( `t_interieure` ), max( `t_exterieure` ) FROM temperature;
+-----+-----+
| min( `t_interieure` ) | max( `t_exterieure` ) |
+-----+-----+
| 6.199999809265137 | 18.399999618530273 |
+-----+-----+
```

Il faut déclarer le champs en Décimal pour éviter le stockage du float de cette façon.

```
mysql> SELECT avg( `t_interieure` ), avg( `t_exterieure` ) FROM temperature;
+-----+-----+
| avg( `t_interieure` ) | avg( `t_exterieure` ) |
+-----+-----+
| 13.724999785423279 | 17.950000286102295 |
+-----+-----+
```

Dépannage (5 points)

Exercice 1 (1 point) - Modélisation

```
DROP TABLE IF EXISTS `depannage`;

CREATE TABLE `depannage` (
  `id_depannage` int(11) NOT NULL AUTO_INCREMENT,
  `garantie` tinyint(1) NOT NULL DEFAULT '0',
  `quand` date NOT NULL,
  `description` text NOT NULL,
  `fk_volet` varchar(20) NOT NULL,
  PRIMARY KEY (`id_depannage`),
  FOREIGN KEY (`fk_volet`) REFERENCES volet (`volet_sn`);
);
```

Exercice 2 (1 point) - Insertion

```
INSERT INTO `voletroulant`.`depannage` (`garantie`, `quand`, `description`, `fk_volet`) VALUES ('0', '2005-08-11', 'Deblocage volet roulant', '1285564100600149');
INSERT INTO `voletroulant`.`depannage` (`garantie`, `quand`, `description`, `fk_volet`) VALUES ('0', '2006-01-12', 'Reagrafage tablier', '1285564100600149');
INSERT INTO `voletroulant`.`depannage` (`garantie`, `quand`, `description`, `fk_volet`) VALUES ('0', '2006-01-12', 'Changement de genouillere', '1285564100600149');
INSERT INTO `voletroulant`.`depannage` (`garantie`, `quand`, `description`, `fk_volet`) VALUES ('1', '2015-02-20', 'Changement de treuil', '320AG5B7320');
INSERT INTO `voletroulant`.`depannage` (`garantie`, `quand`, `description`, `fk_volet`) VALUES ('1', '2015-04-13', 'Changement du moteur', '1101234545');
```

Exercice 3 (2 points) - Remplacement

Analyse

Il ne faut surtout pas remplacer la donnée existante, mais en créer une nouvelle pour l'associer. Un nouveau champs dans la table moteur est donc nécessaire pour conserver la traçabilité, en service qui contiendra la valeur Oui ou Non afin d'identifier si le moteur est encore en service ou non.

En service sauf le moteur WI30008U519024F

```
mysql> ALTER TABLE `moteur` ADD `en_service` BOOLEAN NOT NULL DEFAULT '1';
mysql> UPDATE `voletroulant`.`moteur` SET `en_service` = '0' WHERE `moteur`.`moteur_sn` = 'WI30008U519024F';
```

Un nouveau moteur

```
mysql> INSERT INTO `voletroulant`.`moteur` (`moteur_sn`, `marque`, `moteur_date_achat`, `moteur_ip`, `fk_volet`, `en_service`) VALUES ('WI30008U9113084F', 'winmotor', '2015-04-13', '192.168.1.173', '1101234545', '1');
```

Historique moteur

```
mysql> SELECT * FROM `moteur` WHERE `fk_volet` = '1101234545';
+-----+-----+-----+-----+-----+-----+
| moteur_sn | marque | moteur_date_achat | moteur_ip | fk_volet | en_service |
+-----+-----+-----+-----+-----+-----+
| WI30008U519024F | winmotor | 2006-05-17 | 192.168.1.173 | 1101234545 | 0 |
| WI30008U9113084F | winmotor | 2015-04-13 | 192.168.1.173 | 1101234545 | 1 |
+-----+-----+-----+-----+-----+-----+
```

Requête évoluée (1 point)

On recherche le volet dont le nombre de dépannage est le plus grand

```
mysql> SELECT `fk_volet`, COUNT(*) as c FROM depannage GROUP BY `fk_volet` ORDER BY c DESC LIMIT 1;
+-----+-----+
| fk_volet | c |
+-----+-----+
| 1285564100600149 | 3 |
+-----+-----+
```

Deuxième partie

Année 2017

Devoir Surveillé- Composants Électroniques

Préambule

Une base de données avec une table Composant avec un champ "nom_composant" de type varchar(50).

Contenu des données - Fichier :

Afficheurs	Batteries	Blocs d'alimentation	Cables et fils
Capteurs	Cartes à puces	Circuits imprimés	Circuits intégrés
Coffrets	Condensateurs	Connecteurs	Cordons
Diacs	Diodes	Domestique	Energie solaire et éolien
Fusibles	Hauts-Parleur	Interrupteurs	Inverseurs
Kits et modules	Lampes	Leds	Loupes
Maison	Mesures	Micros	Modules Arduino
Modules émissions-receptions	Optoelectronique	Outillage	Percage
Piles	Potentiometres	Programmateurs	Quartz
Relais	Resistances	Selfs	Sirenes
Soudage	Supports circuits-integres	Thyristors	Torches à Leds
Transformateurs	Transistors	Triacs	Visseries

Question 1 : Insertion - 1 pt

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import sys

f=open ("composants.txt", "r")

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                    user="composant", # utilisateur
                    passwd="composant", # mot de passe
                    db="composant") # nom de la base de données

conn.autocommit (True)

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes
curs = conn.cursor()

for ligne in f:
    composant = ligne.strip()
    requete =
    curs.execute (requete)

curs.close()
conn.close()

f.close()
```

Compléter le programme pour que celui-ci insère l'ensemble des données contenues dans le fichier dans la base de données.

Question 2 : SELECT

Barème : 1 pt / -1 si réponse fautive / 0 si pas de réponse

Lors d'un SELECT, si aucun élément ne correspond à la demande effectuée, il est retourné :

- a) NULL
- b) un tableau vide

- c) 0
- d) Stack Overflow Error

Question 3 : Interrogation d'une base de données

Soit le programme suivant, compléter les assertions indiquées.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import sys

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                   user="composant", # utilisateur
                   passwd="composant", # mot de passe
                   db="composant") # nom de la base de données

conn.autocommit (True)

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes
curs = conn.cursor()

requete = "... "

curs.execute (requete)
rows = curs.fetchall()
for row in rows:
    print row[0]

curs.close()
conn.close()
```

- Tous les éléments de la table (0,5 pt)
- Tous les éléments comportant la chaîne de caractère "io" (1 pt)
- Tous les éléments de 8 caractères (1,5 pt)

Question 4 - 5 points

Soit le programme suivant

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import sys

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                   user="composant", # utilisateur
                   passwd="composant", # mot de passe
                   db="composant") # nom de la base de données

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes
curs = conn.cursor()

# Demande du composant
composant = <1> ("Votre composant ? ")
print ("Composant : " + composant)

# Recherche du composant
requete = <2>
curs.execute(requete)
rows = curs.fetchall()

# Nombre de résultats
nbre_resultats = len (rows)

if (nbre_resultats != 0 ):
    for row in rows:
        print row[0]
else:
    print ("Désolé, nous n'avons pas : " + (composant) + "\n")
```

```
# Recherche du composant
requete2 = <3>
curs.execute(requete2)
tableau = curs.fetchall()

# Nombre de résultats
nbre_resultats = len (tableau)

if ( <4> ):
    print "Cependant nous pouvons vous proposer : "
    for ligne in tableau:
        <5>
else:
    print "Nous ne pouvons rien vous proposer qui ressemblerait à votre demande."

curs.close()
conn.close()
```

Compléter remplacer les champs <1> <2> <3> <4> <5> par les bonnes commandes (1 pt par commande).

Devoir Surveillé- Composants Électroniques Corrigé

Préambule

Une base de données avec une table Composant avec un champ "nom_composant" de type varchar(50).

Contenu des données - Fichier :

Afficheurs	Batteries	Blocs d'alimentation	Cables et fils
Capteurs	Cartes à puces	Circuits imprimés	Circuits intégrés
Coffrets	Condensateurs	Connecteurs	Cordons
Diodes	Diodes	Domestique	Energie solaire et éolien
Fusibles	Haut-parleur	Interrupteurs	Inverseurs
Kits et modules	Lampes	Leds	Loupes
Maison	Mesures	Micros	Modules Arduino
Modules émissions-réceptions	Optoélectronique	Outils	Percage
Piles	Potentiomètres	Programmateurs	Quartz
Relais	Résistances	Selfs	Sirenes
Soudage	Supports circuits-intégrés	Thyristors	Torches à Leds
Transformateurs	Transistors	Triacs	Visseries

Question 1 : Insertion - 1 pt

Compléter le programme pour que celui-ci insère l'ensemble des données contenues dans le fichier dans la base de données.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import sys

f=open("composants.txt", "r")

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                   user="composant", # utilisateur
                   passwd="composant", # mot de passe
                   db="composant") # nom de la base de données

conn.autocommit (True)

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes
curs = conn.cursor()

for ligne in f:
    composant = ligne.strip()
    requete = "INSERT INTO Composant SET nom_composant=\""+ composant + "\""
    print requete
    curs.execute (requete)

curs.close()
conn.close()

f.close()
```

Question 2 : SELECT

Barème : 1 pt / -1 si réponse fautive / 0 si pas de réponse
Lors d'un SELECT, si aucun élément ne correspond à la demande effectuée, il est retourné :

- b) un tableau vide

Question 3 : Interrogation d'une base de données

Soit le programme suivant, compléter les assertions indiquées.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import sys

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                   user="composant", # utilisateur
                   passwd="composant", # mot de passe
                   db="composant") # nom de la base de données

conn.autocommit (True)

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes
curs = conn.cursor()

requete = "... "

curs.execute (requete)
rows = curs.fetchall()
for row in rows:
    print row[0]

curs.close()
conn.close()
```

- Tous les éléments de la table (0,5 pt)

```
requete = "SELECT * FROM Composant"
```

- Tous les éléments comportant la chaîne de caractère "io" (1 pt)

```
requete = "SELECT * FROM Composant WHERE nom_composant LIKE \"%io%\""
```

- Tous les éléments de 8 caractères (1,5 pt)

```
requete = "SELECT * FROM Composant WHERE nom_composant LIKE \"_____\""
```

Question 4 - 5 points

Soit le programme suivant

Compléter remplacer les champs <1> <2> <3> <4> <5> par les bonnes commandes (1 pt par commande).

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import MySQLdb as mdb
import sys

conn = mdb.connect( host="localhost", # votre hôte, normalement le serveur local
                   user="composant", # utilisateur
                   passwd="composant", # mot de passe
                   db="composant") # nom de la base de données

# Il est nécessaire de créer un objet de type Curseur pour réaliser des requêtes
curs = conn.cursor()

# Demande du composant
composant = raw_input ("Votre composant ? ") # <1>
print ("Composant : " + composant)

# Recherche du composant
requete = "SELECT * FROM Composant WHERE nom_composant=\"\" + composant + "\"" # <2>
curs.execute(requete)
rows = curs.fetchall()
```

```
# Nombre de résultats
nbre_resultats = len (rows)

if (nbre_resultats != 0 ):
    for row in rows:
        print row[0]
else:
    print ("Désolé, nous n'avons pas : " + (composant) + "\n")

# Recherche du composant
requete2 = "SELECT * FROM Composant WHERE nom_composant LIKE \"%\" + composant + \"%\" # <3>"
curs.execute(requete2)
rows = curs.fetchall()

# Nombre de résultats
nbre_resultats = len (rows)

if (nbre_resultats != 0 ): # <4>
    print "Cependant nous pouvons vous proposer : "
    for row in rows:
        print "\t" + row[0] # <5>
else:
    print "Nous ne pouvons rien vous proposer qui ressemblerait à votre demande."

curs.close()
conn.close()
```