EXAMEN #1

Votre Nom, Prénom : _____Moreau-Privée, Micaël_____

Écrivez vos réponses de façon lisible. Attention à la présentation de ce document, chaque question et réponse doivent être sur la même page. À la fin de votre document convertissez ce document en format pdf et le mettre dans LEA. our chaque consigne non respectée vous allez perdre 10% de la note final. Vous avez droit à vos notes de cours, exercices et laboratoires. Vous n'avez pas droit d'utiliser un ordinateur.

Soient les tables suivantes :

TABLES

FROMAGES	
Attributs	***Description***
NF (PK)	Numéro du fromage
NOM	Nom du fromage, ex Reblochon
TYPE	Type de fromage, ex persillée
LAIT	Par exemple de brebis, vache, etc
PRIX	Le prix

PERSONNES	
Attributs	***Description***
NP (PK)	Numéro de la personne
NOM	Nom de la personne
PRENOM	Prénom de la personne
VILLE	Ville de provenance

DÉGUSTATION	
Attributs	***Description***
NP (FK)	Numéro de la personne
NF (FK)	Numéro du fromage
DATE	Date de la dégustation
QTE	Quantité

PK-Primary Key
FK-Foreign Key

Règles à respecter

- Si dans une question on ne spécifie pas le nom des attributs alors vous afficherez tous les attributs (*)
- Les mots de SQL en majuscules
- Les noms des tables et attributs en minuscules
- Chaque énoncé de SQL (Exemple : SELECT, FROM, WHERE, …) doit être sur une nouvelle ligne. Utilisez les apostrophes (') pour encadrer les chaines de caractères.

1) Donner le code pour la création des 3 tables. Pour chaque table vous devez indiquer les contraintes de clé primaire et étrangère (si elles existent) via des ***contraintes nommées***. **(30%)**

a) Table **FROMAGES**

```
CREATE TABLE fromages (
    nf      INT NOT NULL,
    nom     VARCHAR(25),
    type    VARCHAR(25),
    lait    VARCHAR(25),
    prix    DECIMAL(6,2),
CONSTRAINT fromages_PRIMARY_KEY PRIMARY KEY (nf));
```

b) Table **PERSONNES**

```
CREATE TABLE personnes (
    np      INT NOT NULL,
    nom     VARCHAR(25),
    prenom  VARCHAR(25),
    ville   VARCHAR(25),
CONSTRAINT personnes_PRIMARY_KEY PRIMARY KEY (np));
```

c) Table **DÉGUSTATION**

```
CREATE TABLE degustations (
    np      INT NOT NULL,
    nf      INT NOT NULL,
    date    DATE,
    qte     INT NOT NULL,
CONSTRAINT np_personnes_FK FOREIGN KEY (np) REFERENCES personnes (np),
CONSTRAINT nf_fromages_FK FOREIGN KEY (nf) REFERENCES fromages (nf));
```

À partir d'ici chaque question vaut 5%

2) Donner la commande SQL pour afficher la structure de la table **DÉGUSTATION**

```
MariaDB [fromages]> desc degustations;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| np    | int(11) | NO   | MUL | NULL    |       |
| nf    | int(11) | NO   | MUL | NULL    |       |
| date  | date    | YES  |     | NULL    |       |
| qte   | int(11) | NO   |     | NULL    |       |
+-------+---------+------+-----+---------+-------+
4 rows in set (0.012 sec)
```

3) Donner le **nom** des fromages dont lait est *brebis.*

```
MariaDB [fromages]> select * from fromages where lait='brebis';
+----+------------------+------+--------+------+
| nf | nom              | type | lait   | prix |
+----+------------------+------+--------+------+
|  3 | Donkey Stiffener | gras | brebis | 9.50 |
|  4 | Le Sweet         | salé | brebis | 6.86 |
+----+------------------+------+--------+------+
2 rows in set (0.000 sec)
```

>>>

```
MariaDB [fromages]> select nom from fromages where lait='brebis';
+------------------+
| nom              |
+------------------+
| Donkey Stiffener |
| Le Sweet         |
+------------------+
2 rows in set (0.001 sec)
```

4) Afficher le contenu de la table **PERSONNES**

```
MariaDB [fromages]> select * from personnes;
+----+--------+---------+-----------------+
| np | nom    | prenom  | ville           |
+----+--------+---------+-----------------+
|  1 | Trump  | Donald  | Boucherville    |
|  2 | Manson | Marilyn | Verdun          |
|  3 | Lama   | Dalai   | Longueuil Beach |
|  4 | Dion   | Caline  | Orlando         |
+----+--------+---------+-----------------+
4 rows in set (0.000 sec)
```

5) Donner les **numéros** et les **noms** des fromages de lait de **vache**, dont le type est **pâte persillée** et dont le **prix** est inférieur à 12 $ le kilo. Le résultat doit être trié par **numéro de fromage**.

Désolé, nous n'avons plus de type 'pâte persillée', en espérant que le type 'jaune' saura vous plaire…

```
MariaDB [fromages]> select nf, nom from fromages where lait='vache' and type='jaune' and prix < 12;
+----+---------------+
| nf | nom           |
+----+---------------+
|  2 | Tranche Orange |
+----+---------------+
1 row in set (0.003 sec)
```

6) Donner la liste de tous les fromages (éliminer les doublons)

```
MariaDB [fromages]> select distinct nom from fromages;
+--------------------+
| nom                |
+--------------------+
| Grue Hier De Crotte |
| Tranche Orange     |
| Donkey Stiffener   |
| Le Sweet           |
+--------------------+
4 rows in set (0.001 sec)
```

7) Donner **3** façons pour afficher le **nom** des fromages dont le **numéro** est compris entre **5** et **8** inclusivement.

a) Façon 1

J'ai mis entre 2 et 3 inclusivement >>>

```
MariaDB [fromages]> select nom from fromages where (nf >= 2) and (nf <= 3);
+-----------------+
| nom             |
+-----------------+
| Tranche Orange  |
| Donkey Stiffener |
+-----------------+
2 rows in set (0.001 sec)
```

ou bien >>>

```
MariaDB [fromages]> select nom from fromages group by nom having (min(nf) >= 2) and (max(nf) <= 3);
+-----------------+
| nom             |
+-----------------+
| Donkey Stiffener |
| Tranche Orange  |
+-----------------+
2 rows in set (0.001 sec)
```

b) Façon 2

```
MariaDB [fromages]> select nom from fromages where nf in(2,3);
+------------------+
| nom              |
+------------------+
| Tranche Orange   |
| Donkey Stiffener |
+------------------+
2 rows in set (0.001 sec)
```

c) Façon 3

```
MariaDB [fromages]> select nom from fromages where nf between 2 and 3;
+------------------+
| nom              |
+------------------+
| Tranche Orange   |
| Donkey Stiffener |
+------------------+
2 rows in set (0.001 sec)
```

8) Donner les fromages dont le nom commence par la lettre '**R**' ou '**r**'.

```
MariaDB [fromages]> select nom from fromages where lower(nom) like 't%';
+----------------+
| nom            |
+----------------+
| Tranche Orange |
+----------------+
1 row in set (0.001 sec)

MariaDB [fromages]> select nom from fromages where lower(nom) like 'r%';
Empty set (0.001 sec)
```

9) Donner le **numéro** (attribut NP) des personnes ayant dégusté du fromage dont le nom est **Reblochon**.

J'ai pris soin de mettre notre fromage le plus fin, soit le 'Tranche Orange' >>>

```
MariaDB [fromages]> select degustations.np from degustations, fromages where fromages.nf = degustations.nf and fromages.nom='Tranc
he Orange';
+----+
| np |
+----+
|  2 |
|  4 |
+----+
2 rows in set (0.001 sec)
```

10) Pour chaque personne donner son **nom**, le **nom** des fromages que cette personne a dégustés ainsi que la **date** de dégustation.

‘Caline Dion’ est venu plus d’une fois (une vraie cochonne) >>>

```
MariaDB [fromages]> select concat(personnes.prenom, ' ', personnes.nom) as 'Goûteur', fromages.nom as 'Fromage', degustations.date
 as 'Quand?' from personnes, fromages, degustations where personnes.np = degustations.np and fromages.nf = degustations.nf;
+----------------+---------------------+------------+
| Goûteur        | Fromage             | Quand?     |
+----------------+---------------------+------------+
| Donald Trump   | Donkey Stiffener    | 2022-06-10 |
| Marilyn Manson | Tranche Orange      | 2022-06-10 |
| Dalai Lama     | Grue Hier De Crotte | 2022-06-10 |
| Caline Dion    | Grue Hier De Crotte | 2022-06-10 |
| Caline Dion    | Tranche Orange      | 2022-06-10 |
| Caline Dion    | Donkey Stiffener    | 2022-06-10 |
| Caline Dion    | Le Sweet            | 2022-06-10 |
+----------------+---------------------+------------+
7 rows in set (0.001 sec)
```

11) Donner le nombre de lignes de la table fromages.

```
MariaDB [fromages]> select count(*) as 'Nb. total de fromages' from fromages;
+----------------------+
| Nb. total de fromages |
+----------------------+
|                    4 |
+----------------------+
1 row in set (0.000 sec)
```

12) Supprimer tous les fromages.

```
MariaDB [fromages]> delete from fromages where nf like '%';
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`fromages`.`degustations`, CONSTRAINT `n
f_fromages_FK` FOREIGN KEY (`nf`) REFERENCES `fromages` (`nf`))
MariaDB [fromages]>
```

La table ‘dégustations’ dépend du contenu de la table ‘fromages’, j’ai donc supprimé ‘degustations’ >>>

```
MariaDB [fromages]> drop table degustations;
Query OK, 0 rows affected (0.008 sec)

MariaDB [fromages]> delete from fromages where nf like '%';
Query OK, 4 rows affected (0.003 sec)

MariaDB [fromages]>
```

13) Insertion d'une ligne dans la table **FROMAGES**. Les valeurs selon votre choix.

```
MariaDB [fromages]> INSERT INTO fromages VALUES (5, 'Le pas de goût en bâton', 'blanc', 'vache', 1.99);
Query OK, 1 row affected (0.001 sec)

MariaDB [fromages]>
```

Avant de remettre vérifiez la mise en page de votre document.

ANNEXE

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[ORDER BY   column];
```

```
SELECT   deptno, max(sal)
FROM     emp
GROUP BY deptno
HAVING   max(sal)>2900;
```

```
SQL> SELECT    job, SUM(sal) PAYROLL
  2  FROM      emp
  3  WHERE     job NOT LIKE 'SALES%'
  4  GROUP BY  job
  5  HAVING    SUM(sal)>5000
  6  ORDER BY  SUM(sal);
```

```
SELECT   table1.column, table2.column
FROM     table1, table2
WHERE    table1.column1 = table2.column2;
```

```
SELECT Personnes.nom, nblivres
FROM Personnes INNER JOIN Bibliothèque
USING (nom)
```

```
SELECT Personnes.nom, nblivres
FROM Personnes INNER JOIN Bibliothèque
ON Personnes.nom = Bibliothèque.nomMembre
```

```
SELECT Personnes.nom, nblivres
  FROM Personnes LEFT JOIN Bibliothèque
  ON Personnes.nom = Bibliothèque.nom
```

```
INSERT INTO   table [(column [, column...])]
VALUES        (value [, value...]);
```

```
UPDATE        table
SET           column = value [, column = value]
[WHERE        condition];
```

```
DELETE [FROM]   table
[WHERE          condition];
```

- AVG : Moyenne
- Min : Minimum
- Max : Maximum
- Sum : Somme

COUNT(*)

COUNT (expr)

Fonctions de Conversion de Casse	Fonctions de Manipulation de Caractères
LOWER	CONCAT
UPPER	SUBSTR
INITCAP	LENGTH
	INSTR
	LPAD

- ROUND: Arrondie un nombre
  - ROUND(45.926, 2) ⟶ 45.93
- TRUNC: Tronque
  - TRUNC(45.926, 2) ⟶ 45.92

```
SELECT ename, DATEDIFF(CURDATE(),hiredate)/7 WEEKS
 FROM emp WHERE deptno = 10;    MySQL
```

Opérateur	
BETWEEN ...AND...	
IN(liste)	
LIKE	
IS NULL	

```
NOT IN (...)

NOT BETWEEN ... AND ...

NOT LIKE ' ... '

... IS NOT NULL
```

- % signifie 0 ou plusieurs caractères
- _ signifie exactement un caractère

```
SQL> SELECT   CONCAT(ename,job) AS "Employees"
  2  FROM     emp;                          MySQL
```

FIN DE L'EXAMEN.