

1. Obtenir l'utilisateur ayant le prénom "Muriel" et le mot de passe "test11", sachant que l'encodage du mot de passe est effectué avec l'algorithme Sha1.

```
SELECT *  
FROM client  
WHERE prenom = 'Muriel' AND password = encode(digest('test11','sha1'),'hex');
```

	id	prenom	nom	email	ville	password
1	11	Muriel	Dupuis	muriel@examp...	Paris	100c4e57374fc...

2. Obtenir la liste de tous les produits qui sont présent sur plusieurs commandes.

```
SELECT nom, COUNT(nom) AS nombre_produit  
FROM commande_ligne  
GROUP BY nom  
HAVING COUNT(nom) > 1  
ORDER BY nombre_produit DESC;
```

	nom	nombre_produit
1	Produit 6D	4
2	Produit 67	3
3	Produit DD	2
4	Produit 52	2
5	Produit DE	2
6	Produit D9	2
7	Produit 95	2
8	Produit D6	2
9	Produit 2E	2
10	Produit FC	2
11	Produit 00	2
12	Produit 3C	2
13	Produit E1	2
14	Produit 8A	2
15	Produit 12	2
16	Produit 78	2
17	Produit 93	2
18	Produit C4	2
19	Produit 07	2

3. Obtenir la liste de tous les produits qui sont présent sur plusieurs commandes et y ajouter une colonne qui liste les identifiants des commandes associées.

```
SELECT nom, COUNT(nom) AS nombre_produit  
FROM commande_ligne  
GROUP BY nom  
HAVING COUNT(nom) > 1  
ORDER BY nombre_produit DESC;
```

4. Enregistrer le prix total à l'intérieur de chaque ligne des commandes, en fonction du prix unitaire et de la quantité.

```
UPDATE commande_ligne  
SET prix_total = (prix_unitaire * quantite);
```

	id	commande_id	nom	quantite	prix_unitaire	prix_total
1	1	1	Produit 19	3	49.57	148.71
2	2	1	Produit 92	4	81.24	324.96
3	3	1	Produit 68	2	17.48	34.96
4	4	2	Produit 53	4	83.69	334.76
5	5	2	Produit 78	6	5.99	35.94
6	6	3	Produit D9	7	18.91	132.37
7	7	4	Produit A3	8	76.57	612.56
8	8	4	Produit BB	10	86.14	861.4
9	9	4	Produit 7C	4	80.96	323.84
10	10	4	Produit 78	9	26.4	237.6
11	11	4	Produit 07	6	9.13	54.78
12	12	5	Produit 00	10	86.45	864.5
13	13	5	Produit 7A	2	44.86	89.72
14	14	6	Produit E1	9	84.93	764.37
15	15	7	Produit D6	2	50.07	100.14
16	16	7	Produit BD	7	115.55	808.85
17	17	7	Produit D9	3	67.55	202.65
18	18	8	Produit 55	9	111.12	1000.08
19	19	9	Produit C7	10	112.93	1129.3
20	20	10	Produit 2A	2	111.31	222.62

- Obtenir le montant total pour chaque commande et y voir facilement la date associée à cette commande ainsi que le prénom et nom du client associé.

```
SELECT client.prenom, client.nom, commande.date_achat, SUM(prix_total) AS prix_total_commande
FROM commande_ligne
INNER JOIN commande ON commande.id = commande_ligne.commande_id
INNER JOIN client ON client.id = commande.client_id
GROUP BY client.prenom, client.nom, commande.date_achat
ORDER BY client.nom;
```

	nom	prenom	date_achat	prix_total_com...
1	Buisson	Maris	2019-01-18	136.4
2	Buisson	Maris	2019-01-25	1928.59
3	Camus	Emilien	2019-01-14	97.0
4	Camus	Emilien	2019-01-27	995.76
5	Camus	Emilien	2019-02-13	719.54
6	Collin	Gustave	2019-01-03	370.7
7	Collin	Gustave	2019-01-17	1646.31
8	Collin	Gustave	2019-01-21	907.2
9	Collin	Gustave	2019-02-06	700.96
10	Dupuis	Muriel	2019-01-04	132.37
11	Dupuis	Muriel	2019-02-02	362.81
12	Dupuis	Muriel	2019-02-03	673.65
13	Durand	Manon	2019-01-15	482.45
14	Durand	Manon	2019-01-19	1285.81
15	Durand	Manon	2019-02-01	472.82
16	Foucher	Fabrice	2019-01-13	1063.17
17	Foucher	Fabrice	2019-02-09	554.7
18	Huet	Maurice	2019-02-02	784.0
19	Huet	Maurice	2019-02-11	1398.06
20	Jung	Lucas	2019-01-09	764.37
21	Jung	Lucas	2019-01-11	1000.08
22	Jung	Lucas	2019-02-05	751.64
23	Jung	Lucas	2019-02-16	592.32
24	Langlois	Jacinthe	2019-02-14	620.68
25	Langlois	Jacinthe	2019-02-15	1321.91

26	Leon	Joachim	2019-02-05	114.4
27	Leon	Joachim	2019-02-09	185.28
28	Marais	Firmin	2019-02-08	93.68
29	Payet	Amaury	2019-01-16	451.94
30	Payet	Amaury	2019-02-07	441.85
31	Riou	Christiane	2019-01-11	1129.3
32	Riou	Christiane	2019-01-29	238.99
33	Riou	Christiane	2019-02-17	1518.11
34	Riou	Olivier	2019-01-07	2090.18
35	Riou	Olivier	2019-01-16	1223.32
36	Saunier	Patrick	2019-01-08	954.22
37	Saunier	Patrick	2019-01-20	1061.92
38	Saunier	Patrick	2019-01-22	1169.15
39	Saunier	Patrick	2019-01-23	510.07
40	Vallee	Arnaude	2019-01-01	508.63
41	Vallee	Arnaude	2019-02-04	1255.08
42	Vespasien	Valentin	2019-01-10	1111.64
43	Vespasien	Valentin	2019-02-08	810.2
44	Vespasien	Valentin	2019-02-12	1429.16
45	Vespasien	Valentin	2019-02-19	2637.18
46	Vigneron	Armel	2019-02-10	567.13
47	Vigneron	Armel	2019-02-18	611.52

- Enregistrer le montant total de chaque commande dans le champ intitulé "cache_prix_total"

```
UPDATE commande
SET cache_prix_total = a.prix
FROM
(SELECT commande_id, SUM(prix_total) AS prix
FROM commande_ligne
GROUP BY commande_id) AS a
WHERE commande.id = a.commande_id;
```

	id	client_id	date_achat	reference	cache_prix_total
1	1	20	2019-01-01	004214	508.63
2	2	3	2019-01-03	007120	370.7
3	3	11	2019-01-04	002957	132.37
4	4	6	2019-01-07	003425	2090.18
5	5	17	2019-01-08	008255	954.22
6	6	7	2019-01-09	000996	764.37
7	7	2	2019-01-10	000214	1111.64
8	8	7	2019-01-11	008084	1000.08
9	9	12	2019-01-11	009773	1129.3
10	10	16	2019-01-13	004616	1063.17
11	11	4	2019-01-14	003757	97.0
12	12	9	2019-01-15	004939	482.45
13	13	14	2019-01-16	003421	451.94
14	14	6	2019-01-16	002286	1223.32
15	15	3	2019-01-17	001167	1646.31
16	16	15	2019-01-18	008974	136.4
17	17	9	2019-01-19	001369	1285.81
18	18	17	2019-01-20	009924	1061.92
19	19	3	2019-01-21	005510	907.2
20	20	17	2019-01-22	007778	1169.15
21	21	17	2019-01-23	002359	510.07
22	22	15	2019-01-25	008459	1928.59
23	23	4	2019-01-27	005217	995.76
24	24	12	2019-01-29	000706	238.99
25	25	9	2019-02-01	007879	472.82

26	26	8	2019-02-02	007277	784.0
27	27	11	2019-02-02	002745	362.81
28	28	11	2019-02-03	001893	673.65
29	29	20	2019-02-04	001230	1255.08
30	30	10	2019-02-05	000469	114.4
31	31	7	2019-02-05	008653	751.64
32	32	3	2019-02-06	001858	700.96
33	33	14	2019-02-07	003330	441.85
34	34	2	2019-02-08	001074	810.2
35	35	5	2019-02-08	005379	93.68
36	36	16	2019-02-09	003672	554.7
37	37	10	2019-02-09	002220	185.28
38	38	19	2019-02-10	000086	567.13
39	39	8	2019-02-11	003770	1398.06
40	40	2	2019-02-12	008590	856.14
41	41	2	2019-02-12	001639	573.02
42	42	4	2019-02-13	002426	719.54
43	43	13	2019-02-14	007209	620.68
44	44	13	2019-02-15	008768	1321.91
45	45	7	2019-02-16	002213	592.32
46	46	12	2019-02-17	004759	1518.11
47	47	19	2019-02-18	007155	611.52
48	48	2	2019-02-19	001496	2637.18

7. Obtenir le montant global de toutes les commandes, pour chaque mois.

```
SELECT extract(year FROM date_achat) AS annee_achat, extract(month FROM date_achat)
AS mois_achat, SUM(cache_prix_total) AS prix_total_au_mois
FROM commande
GROUP BY annee_achat, mois_achat
ORDER BY mois_achat DESC;
```

	annee_achat	mois_achat	prix_total_au_mois
1	2019.0	2.0	18616.68
2	2019.0	1.0	21259.57

8. Obtenir la liste des 10 clients qui ont effectué le plus grand montant de commandes, et obtenir ce montant total pour chaque client.

```
SELECT nom, prenom, SUM(cache_prix_total) AS montant_commande
FROM client
INNER JOIN commande ON client.id = commande.client_id
GROUP BY nom, prenom
ORDER BY montant_commande DESC LIMIT 10;
```

	nom	prenom	montant_commande
1	Vespasien	Valentin	5988.18
2	Saunier	Patrick	3695.36
3	Collin	Gustave	3625.17
4	Riou	Olivier	3313.5
5	Jung	Lucas	3108.41
6	Riou	Christiane	2886.4
7	Durand	Manon	2241.08
8	Huet	Maurice	2182.06
9	Buisson	Maris	2064.99
10	Langlois	Jacinthe	1942.59

9. Obtenir le montant total des commandes pour chaque date.

```
SELECT date_achat, SUM(cache_prix_total) AS total_commande
FROM commande
GROUP BY date_achat
ORDER BY date_achat DESC;
```

	date_achat	total_commande
1	2019-02-19	2637.18
2	2019-02-18	611.52
3	2019-02-17	1518.11
4	2019-02-16	592.32
5	2019-02-15	1321.91
6	2019-02-14	620.68
7	2019-02-13	719.54
8	2019-02-12	1429.16
9	2019-02-11	1398.06
10	2019-02-10	567.13
11	2019-02-09	739.98

12	2019-02-08	903.88
13	2019-02-07	441.85
14	2019-02-06	700.96
15	2019-02-05	866.04
16	2019-02-04	1255.08
17	2019-02-03	673.65
18	2019-02-02	1146.81
19	2019-02-01	472.82
20	2019-01-29	238.99
21	2019-01-27	995.76
22	2019-01-25	1928.59
23	2019-01-23	510.07
24	2019-01-22	1169.15
25	2019-01-21	907.2
26	2019-01-20	1061.92
27	2019-01-19	1285.81
28	2019-01-18	136.4
29	2019-01-17	1646.31
30	2019-01-16	1675.26
31	2019-01-15	482.45
32	2019-01-14	97.0
33	2019-01-13	1063.17
34	2019-01-11	2129.38
35	2019-01-10	1111.64
36	2019-01-09	764.37
37	2019-01-08	954.22
38	2019-01-07	2090.18
39	2019-01-04	132.37
40	2019-01-03	370.7
41	2019-01-01	508.63

10. Ajouter une colonne intitulée “category” à la table contenant les commandes. Cette colonne contiendra une valeur numérique.

```
ALTER TABLE commande ADD category NUMERIC;
```

11. Enregistrer la valeur de la catégorie, en suivant les règles suivantes

- « 1 » pour les commandes de moins de 200€
- « 2 » pour les commandes entre 200€ et 500€
- « 3 » pour les commandes entre 500€ et 1.000€
- « 4 » pour les commandes supérieures à 1.000€

```
UPDATE commande  
SET category = (  
CASE  
WHEN cache_prix_total < 200 THEN 1  
WHEN cache_prix_total < 500 THEN 2  
WHEN cache_prix_total < 1000 THEN 3  
ELSE 4  
END);
```

12. Créer une table intitulée “commande_category” qui contiendra le descriptif de ces catégories

```
CREATE TABLE commande_category (id INTEGER NOT NULL, descriptif VARCHAR(255));
```

13. Insérer les 4 descriptifs de chaque catégorie au sein de la table précédemment créée

```
INSERT INTO commande_category VALUES  
(1, 'prix commandes inférieures à 200€'),  
(2, 'prix commandes entre 200€ et 500€'),  
(3, 'prix commandes entre 500€ et 1000€'),  
(4, 'prix commandes supérieures à 1000€');
```

	id	descriptif
1	1	prix commandes inférieures à 200€
2	2	prix commandes entre 200€ et 500€
3	3	prix commandes entre 500€ et 1000€
4	4	prix commandes supérieures à 1000€

14. Supprimer toutes les commandes (et les lignes des commandes) inférieur au 1er février 2019. Cela doit être effectué en 2 requêtes maximum

```
DELETE FROM commande_ligne  
WHERE commande_id IN (SELECT id FROM commande WHERE date_achat < '2019-02-01');
```

```
DELETE FROM commande WHERE date_achat < '2019-02-01';
```