Exercice 2

Les tables qui ne se référent à aucune autres doivent être crées et alimentées en premières. C'est à dire les tables sans clés étrangère . Ces tables peuvent avoir une clé étrangère sur elle mêmes, la sous catégorie doit être nul ou se référer à une catégorie existante.

Vienne ensuite la création et l'alimentation des tables avec des clés étrangères sur d'autres tables déjà crées et alimentées. C'est à dire que lors de l'alimentation de ces tables, la clé étrangère doit exister dans la tables où elles se réfère. Par exemple, pour un client avec un pays, son pays doit déjà être alimenté dans la table pays. En cas de sous catégories, la réglé présenté en 1 s’applique de la même façon.

Pour la suppression, il s'agit d'appliquer un ordre inverse de celui de la création ou bien dans un autre cas, si on supprime une table parente avec des données dont une autre table se réfère, il faudra faire alors une suppression en cascade.

Q1

SELECT CONCAT(emp\_lastname,' ', emp\_firstname)AS employés, emp\_children AS Nbr\_enfants

FROM employees

ORDER BY employés

Q2

SELECT cus\_lastname, cus\_firstname, cus\_countries\_id

FROM customers

WHERE cus\_countries\_id != "FR"

Q3

SELECT cus\_city, cus\_countries\_id, cou\_name

FROM customers

JOIN countries ON customers.cus\_countries\_id = countries.cou\_id

ORDER BY customers.cus\_city;

Q4

SELECT cus\_lastname, cus\_update\_date

FROM customers

WHERE cus\_update\_date IS NOT NULL;

Q5

SELECT cus\_id AS Client\_ID, CONCAT(cus\_lastname,' ',cus\_firstname) AS NOM\_Prénom, cus\_city AS Ville

FROM customers

WHERE cus\_city LIKE "%divos%"

Q6

SELECT pro\_id, pro\_name, pro\_price

FROM products

WHERE pro\_price

IN (SELECT MIN(pro\_price) FROM products)

Q7

SELECT pro\_id, pro\_ref, pro\_name

FROM products

WHERE pro\_id NOT IN

    (SELECT ode\_pro\_id FROM orders\_details)

Q8

SELECT pro\_id, pro\_ref, pro\_color, pro\_name, cus\_id, cus\_lastname, ord\_id, ode\_id

FROM products

JOIN orders\_details ON products.pro\_id = orders\_details.ode\_pro\_id

JOIN orders ON orders\_details.ode\_ord\_id = orders.ord\_id

JOIN customers ON orders.ord\_cus\_id = customers.cus\_id

WHERE cus\_lastname LIKE "Pikatchien"

Q9

SELECT cat\_id, cat\_name, pro\_name

FROM categories

JOIN products ON categories.cat\_id = products.pro\_cat\_id

ORDER BY cat\_name;

Q10

SELECT CONCAT(sup.emp\_lastname, ' ', sup.emp\_firstname) AS Employé , CONCAT(emp.emp\_lastname, ' ', emp.emp\_firstname) AS Supérieur

FROM employees AS sup

RIGHT JOIN employees AS emp ON sup.emp\_superior\_id = emp.emp\_id

WHERE sup.emp\_sho\_id IN

    (SELECT shops.sho\_id FROM shops WHERE shops.sho\_city LIKE 'Compiegne')

ORDER BY Employé

Q11

SELECT pro\_id, CONCAT(pro\_ref, ' ', pro\_name) AS "libellé du produit", ord\_id, ode\_id, ode\_discount

FROM products

JOIN orders\_details ON products.pro\_id = orders\_details.ode\_pro\_id

JOIN orders ON orders\_details.ode\_ord\_id = orders.ord\_id

WHERE ode\_discount IN

    (SELECT MAX(ode\_discount) FROM orders\_details )

Q12

SELECT COUNT(cus\_countries\_id) AS "NOMBRE CLIENT CANADA"

FROM customers

JOIN countries ON customers.cus\_countries\_id = countries.cou\_id

WHERE cus\_countries\_id LIKE 'CA';

Q13

SELECT ode\_id, ode\_unit\_price, ode\_discount, ode\_quantity, ode\_ord\_id, ode\_pro\_id, ord\_order\_date

FROM orders\_details

JOIN orders ON orders\_details.ode\_ord\_id = orders.ord\_id

WHERE YEAR(ord\_order\_date) = "2020"

ORDER BY ord\_order\_date DESC

Q14

SELECT sup\_id, sup\_name, sup\_address, sup\_zipcode, sup\_city, sup\_contact, sup\_phone, sup\_mail

FROM suppliers

JOIN products ON suppliers.sup\_id = products.pro\_sup\_id

JOIN orders\_details ON products.pro\_id = orders\_details.ode\_pro\_id

WHERE ode\_pro\_id IS NOT NULL

GROUP BY sup\_name

Q15

SELECT ROUND(SUM((ode\_unit\_price \* ode\_quantity)-(((ode\_unit\_price \* ode\_quantity)/ 100) \* ode\_discount)), 2) AS "Chiffre d'affaire"

FROM orders\_details

JOIN orders ON orders\_details.ode\_ord\_id = orders.ord\_id

WHERE YEAR(ord\_order\_date) = 2020;

Q16

SELECT ord\_id, cus\_lastname, ord\_order\_date, ode\_quantity, SUM(ode\_quantity \* pro\_price) AS Total

FROM orders\_details

JOIN orders ON orders\_details.ode\_ord\_id = orders.ord\_id

JOIN customers ON orders.ord\_cus\_id = customers.cus\_id

JOIN products ON orders\_details.ode\_pro\_id = products.pro\_id

GROUP BY ord\_id

ORDER BY Total DESC;

Q17

SELECT AVG(total) AS 'Moyenne Panier'

FROM (SELECT SUM(ode\_quantity \* (ode\_unit\_price - ( (ode\_unit\_price \* ode\_discount) / 100) )) AS total

        FROM orders\_details

            GROUP BY ode\_ord\_id) AS Gilb;

Q18

update products

set pro\_ref='camper'

pro\_price = pro\_price \* 0.90

where pro\_id=12

Q19

update products

join categories on products.pro\_cat\_id = categories.cat\_id

set pro\_price = pro\_price \* 1.011

where cat\_name like 'parasols'

Q20

DELETE FROM `products`

where `pro\_cat\_id` = (select `cat\_id` from `categories` where `cat\_name`="Tondeuses électriques") and `pro\_id` not in (select `ode\_pro\_id` from `orders\_details`);