**Application REXEL - Requêtes SQL**

L’entreprise REXEL commercialise du matériel électrique à destination d’une clientèle de professionnels (électriciens, thermiciens…). A l’aide des relations suivantes et des extraits de tables, vous êtes chargé de préparer les requêtes suivantes :

**Requêtes**

1. Liste des produits avec les prix de vente et leur prix d'achat
2. Liste des produits dont le prix d'achat est supérieur à 100 €, mais dont le prix de vente est inférieur à 150 €
3. Nom des clients avec le nom du commercial qui les suit
4. Liste des produits dont la quantité en stock est inférieure à 3
5. Liste des clients du département du Rhône
6. Le prix de vente, prix d’achat et la Marge dégagée par le disjoncteur 856 (la marge est la différence entre le prix de vente et le prix d’achat)
7. Liste des produits de la commande n°5
8. Le prix d’achat moyen des disjoncteurs
9. La liste des produits en stock avec le chiffrage pour chaque produit (Qté \* Prix), classé par rayon
10. Liste des produits stockés dans le rayon Bat A-33
11. Liste des clients qui ont passé une commande en 2020
12. Liste des clients qui n’ont passé aucune commande en avril 2020 (vous pouvez regarder le memento SQL pour vous aider)
13. CA total de la commande n°108
14. Liste des clients non affecté à un commercial
15. Liste des produits commandés au 1er trimestre 2020

**TABLES DE LA BASE DE DONNEES DE EBP**

* Client(NumClient, NomClient, CPClt, VilleClt, #NumCial)
* Commande(NumCde, DateCde, #Numclient)
* Produit(Ref, Designation, PrixAchat, PrixVente, #NumRayon, Qté\_stock)
* LigneCommande(#Ref, #NumCde, QtCde)
* Rayon(NumRayon, NomRayon)
* Commercial(Numcial, NomCial, PrenomCial)

**Extrait du contenu des tables**

|  |
| --- |
| **CLIENT** |
| NumClient | NomClient | CPClt | VilleClt | NumCial |
| 524 | ELECTRICITE JANDON | 88000 | GERARDMER | 12 |

|  |  |  |
| --- | --- | --- |
| **COMMANDE** |  | **PRODUIT** |
| **NumCde** | **Date Cde** | **NumClient** |  | **Ref** | **Designation** | **PrixAchat** | **PrixVente** | **NumRayon** | **Quantité en stock** |
| 107 | 17/05/20 | 524 |  | 856 | Disjoncteurs 30mA | 157,53 | 212,32 | 18 | 82 |
| 108 | 18/05/20 | 6520 |  | 857 | Disjoncteurs 60 mA | 265,50 | 357,20 | 17 | 166 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ligne Commande** |  | **RAYON** |  | **COMMERCIAL** |
| **Ref** | **NumCde** | **QteCde** |  | **NumRayon** | **NomRayon** |  | **NumCial** | **NomCial** | **PrenomCial** |
| 856 | 107 | 2 |  | 1 | Bat A-32 |  | 12 | DUPOND | Régis |
| 857 | 108 | 5 |  | 2 | Bat A-33 |  | 13 | FARGE | Louis |
| 823 | 108 | 1 |  |  |  |  |  |  |  |

**Annexe : Mémento SQL**

|  |
| --- |
|  **PROJECTION D’ATTRIBUTS** |
| *Expression* | *Résultat* | *Syntaxe* |
| SELECT | Spécifie les attributs que l’on veut extraire et afficher | SELECT TABLE.Attribut |
| FROM | Spécifie les tables nécessaires à la requête | FROM TABLE1, TABLE2 |
| ; | Indique que la requête est terminée | ; |
|  |  |  |
| **SÉLECTION** |
| *Expression* | *Résultat* | *Syntaxe* |
| WHERE | Précède la première jointure ou sélection | WHERE TABLE.Attribut LIKE chaîne de caractères |
| AND | Succède à WHERE que ce soit pour une sélection ou une jointure | AND TABLE.Attribut = Valeur numérique |
| OR | Précède une sélection (union) | OR TABLE.Attribut = Valeur numérique |
| LIKE / =  | LIKE précède une chaîne de caractères. = précède une valeur numérique. | WHERE TABLE.Attribut LIKE chaîne de caractèresAND TABLE.Attribut = Valeur numérique |
| IS [NOT] NULL | Prédicat de [non] nullité | WHERE TABLE.Attribut IS [NOT] NULL |
| BETWEEN … AND …  | Prédicat d’intervalleÉquivalent à >= … AND <= … | WHERE TABLE.Attribut BETWEEN valeur1 AND valeur 2 |
|  |  |  |
| **TRI** |
| *Expression* | *Résultat* | *Syntaxe* |
| ORDER BY … ASC ou DESC | La hiérarchie des clés de tri est définie par l’ordre des attributs derrière ORDER BY | ORDER BY TABLE.Attribut1, TABLE.Attribut2 ASC |
|  |  |  |
| **CALCULS**  |
| *Expression* | *Résultat* | *Syntaxe* |
| SUM | Retourne la somme des valeurs d’un attribut d’une table | SELECT SUM (TABLE.Attribut) AS NomAlias |
| AVG | Retourne la moyenne des valeurs d’un attribut d’une table | SELECT AVG (TABLE.Attribut) AS NomAlias |
| MAX | Retourne la valeur maximum d’un attribut d’une table | SELECT MAX (TABLE.Attribut) AS NomAlias |
| MIN | Retourne la valeur minimum d’un attribut d’une table | SELECT MIN (TABLE.Attribut) AS NomAlias |
| AS | L’attribut projeté est identifié par le nom de l’alias | SELECT SUM (TABLE.Attribut) AS NomAlias |
|  |  |  |
| **REGROUPEMENT** |
| *Expression* | *Résultat* | *Syntaxe* |
| COUNT | Retourne le nombre de tuples d’une table | SELECT COUNT (TABLE.Attribut) AS NomAlias |
| GROUP BY | Permet de faire porter les fonctions d’agrégat sur des partitions de la table | GROUP BY TABLE.AttributHAVING TABLE.Attribut = Valeur |
| HAVING | Permet d’appliquer des prédicats de condition sur des résultats de regroupement | GROUP BY TABLE.AttributHAVING TABLE.Attribut = Valeur |

**CORRIGE REXEL - Requêtes SQL**

1. *Liste des produits avec les prix de vente et leur prix d'achat*

SELECT ref, Designation, Prix Achat, PrixVente,

FROM PRODUIT ;

1. *Liste des produits dont le prix d'achat est supérieur à 100 €, mais dont le prix de vente est inférieur à 150 €*

SELECT Ref, Designation,PrixAchat, PrixVente

FROM PRODUITS

WHERE PrixAchat>100

AND PrixVente<150 ;

1. *Nom des clients avec le nom du commercial qui les suit*

|  |  |
| --- | --- |
| SELECT NomClient, NomcialFROM CLIENT, COMMERCIALWHERE CLIENT.Numcial = COMMERCIAL.Numcial ; | *SELECT NomClient, Nomcial**FROM CLIENT INNER JOIN COMMERCIAL ON CLIENT.Numcial = COMMERCIAL.Numcial ;* |

1. *Liste des produits dont la quantité en stock est inférieure à 3*

SELECT Ref, Designation, Qté\_Stock

FROM PRODUIT

WHERE Qte\_Stock <3

1. *Liste des clients du département du Rhône*

SELECT NomClient, CPClt

FROM CLIENT

WHERE CPClt LIKE "69\*" ;

1. *Le prix de vente, prix d’achat et la Marge dégagée par le disjoncteur 856 (la marge est la différence entre le prix de vente et le prix d’achat)*

SELECT Ref, Designation, PrixAchat, PrixVente, PrixVente- PrixAchat AS [Marge]

FROM PRODUIT

WHERE ref = 856

1. *Liste des produits de la commande n°5*

|  |  |
| --- | --- |
| SELECT Ref, Designation, NumCdeFROM PRODUIT, LIGNECOMMANDEWHERE PRODUIT.Ref, LIGNECOMMANDE.RefAND NumCde =107 ; | *SELECT Ref, Designation, NumCde**FROM PRODUIT INNER JOIN LIGNECOMMANDE ON PRODUIT.Ref, LIGNECOMMANDE.Ref**WHERE NumCde = 5 ;* |

1. *Le prix d’achat moyen des disjoncteurs*

SELECT AVG(PrixAchat) AS [prix d’achat moyen disjoncteurs]

FROM PRODUIT

WHERE designation LIKE “Disjoncteur\*”

1. *La liste des produits en stock avec le chiffrage pour chaque produit (Qté \* Prix), classé par rayon*

SELECT Ref, Designation, NumRayon, PrixAchat, Qte\_stock, PrixAchat \*Qte\_stock AS [valorisation stock]

FROM PRODUIT

ORDER BY Numrayon

1. *Liste des produits stockés dans le rayon Bat A-33*

SELECT Ref, Designation, NomRayon

FROM PRODUIT, RAYON

WHERE PRODUIT.NumRayon = RAYON.NumRayon

AND Rayon = "Bat A-33" ;

1. *Liste des clients qui ont passé une commande en 2020*

SELECT NumCde, DateCde, NumClient, NomClient

FROM CLIENT, COMMANDE

WHERE CLIENT.NumClient = COMMANDE.NumCde

AND Date Cde >= 01/01/2020

1. *Liste des clients qui n’ont passé aucune commande en avril 2020 (vous pouvez regarder le memento SQL pour vous aider)*

SELECT NomClient,

FROM CLIENT, COMMANDE

WHERE CLIENT.NumClient = COMMANDE.NumCde

AND NumCde IS NULL

AND date Cde BETWEEN 01/04/2020 AND 30/04/2020 ;

1. *CA total de la commande n°108*

SELECT ref, Designation, QteCde, PrixVente, SUM(QteCde\*PrixVente) AS [Montant vente]

FROM PRODUIT, LIGNECOMMANDE

WHERE NumCde = 108

1. *Liste des clients non affecté à un commercial*

SELECT NomClient, Nomcial

FROM CLIENT, COMMERCIAL

WHERE CLIENT.Numcial = COMMERCIAL.Numcial

AND NomCial IS NULL ;

1. *Liste des produits commandés au 1er trimestre 2020*

SELECT Ref, Designation, DateCde

FROM PRODUIT, LIGNECOMMANDE, COMMANDE

WHERE LIGNECOMMANDE.Ref=PRODUIT.Ref

AND LIGNECOMMANDE.Numcde=COMMANDE.NumCde

AND DateCde >= 01/01/2020 AND <=31/03/2020 ;