

**REPONSES IPPOLITO LUCAS**

* **Créer et peupler la base de données**

Dans un premier temps vous devez installer la base de données dans votre environnement. Sont disponibles un fichier SQL avec la structure de la base ainsi qu’une archive contenant les données de la base sous forme de fichiers CSV.

**2. Analyser la base de données**

**2.1 Analyse structurelle**

Une fois la base de données installée et peuplée, vous effectuerez l’analyse structurelle de la base. Sont attendus une description de chaque table (à quoi sert-elle ?) ainsi que les relations entre celles-ci.

**Customers: Ce sont les clients listés, elle sert à avoir toutes les infos nécéssaires sur eux.**

**Employees: La liste des employés de classicmodelcars, des infos nécéssaires comme leur email et leur fonction.**

**Offices: Ce sont les bureux de l'entreprise, avec adresses.**

**orderdetails: Commandes passées et détaillées dans le prix & quantité, elle est reliée à la table order via son orderNumber.**

**orders: Commande globale avec numéro de commande, date d'envoie, date requise et commentaire sur la commande.**

**payments: Table qui permet de voir le numéro de client, la date à laquelle il a payé et le montant de sa commande.**

**productlines: Grande famille qui vont regrouper différents modèles.**

**products: Produits dans leur précision avec le nom exactes, leur famille, le vendeur du produit, sa description, sa quantité en stock son prix.**

**2.2 Analyse de données**

De quel genre d’entreprises est tirée cette base de données ? Fournissez le dictionnaire de données des tables offices et customers.

**Cette base de données serait tirée d'une entreprise qui vend différents modèles de jouets partout dans le monde.**

**(Voir dico PDF)**

**3. Manipulations**

**3.1 Interrogations - Requêtes**

* Combien y-a-t-il d'enregistrements dans les tables customers, employees, offices, orders, payments et products ?

**customers: 122**

**employees: 23**

**offices: 7**

**orders: 326**

**payments: 273**

**products: 110**

* Lister par quantité (quantityInStock) décroissante les productName de la table des produits (products) contenant "*Harley*".

**SELECT productName FROM products**

**WHERE productName LIKE '%Harley%'**

**ORDER BY quantityInStock DESC**

* Lister par le prénom en ordre croissant dans la table des clients (customers) les clients dont le prénom a un **a** en deuxième position.

**SELECT contactFirstName FROM customers**

**WHERE contactFirstName LIKE '\_a%'**

**ORDER BY customerNumber ASC**

* Combien y en a-t-il ?

**33 rows in set, donc 33**

**Ou on rajoute un COUNT**

* Dans la table products dont nous voulons n'afficher que productName et

buyPrice sous les dénominations respectives de "Les articles" et "Les prix"

; lister les articles dont les prix sont compris entre 50 et 65.

**SELECT productName AS 'Les articles', buyPrice AS 'Les prix' FROM products**

**WHERE buyPrice BETWEEN 50 AND 65**

* Afficher la somme, que vous nommerez "Total [07/2004]", de tous les paiements effectués dans la table payments depuis le 1 juillet 2004.

**SELECT SUM(amount) AS 'Total [07/2004]' FROM payments**

**WHERE paymentDate BETWEEN '2004-06-30' AND '2004-08-01'**

**Total de juillet: 290611.32**

* Sélectionner dans les détails de commande (orderDetails), les commandes dont la quantité est supérieure ou égale à 50 que l'on veut grouper par numéro d'ordre.

**SELECT \* FROM orderDetails**

**WHERE quantityOrdered >= 50**

**GROUP BY orderNumber**

* Sélectionner les clients (customerName) qui n'ont pas encore passé commande et trier par ordre alphabétique ascendant.

**SELECT customerName FROM customers**

**LEFT JOIN orders ON customers.customerNumber = orders.customerNumber**

**WHERE orders.customerNumber IS NULL**

**ORDER BY customerName ASC**

* Combien d'employés (Nom, prenom, employeeNumber, jobTitle) ne sont rattachés à aucun bureau (officeCode) ?

**SELECT (\*) FROM employees**

**LEFT JOIN offices ON employees.officeCode = offices.officeCode**

**WHERE offices.officeCode IS NULL**

**Soit: zéro**

* Combien y-a-t-il de villes différentes dans la table customers ?

**SELECT COUNT(DISTINCT city) FROM customers;**

**Réponse: 95**

* Quel est le nom et le numéro du client qui a payé le plus dans la table payments. On souhaite avoir la somme totale réellement payée pour chaque client dont on ne veut que le nom (pas le numéro).

**1ere requête:**

**SELECT SUM(amount), customerName, customers.customerNumber**

**FROM customers**

**INNER JOIN payments ON customers.customerNumber = payments.customerNumber**

**GROUP BY payments.customerNumber**

**ORDER BY SUM(amount) DESC LIMIT1**

**Rep: 715738.98 Euro+ Shopping Channel Number:141**

**2eme requête:**

**SELECT SUM(amount), customerName FROM customers**

**INNER JOIN payments ON customers.customerNumber = payments.customerNumber**

**GROUP BY payments.customerNumber**

**ORDER BY SUM(amount) DESC;**

* Dans la table des produits (products), nous voulons la quantité commandée, qui se trouve dans les détails des ordres (orderDetails), totale, et la somme des ventes, depuis le 01/01/2005, classées par ligne de produit (productLine).

**SELECT SUM(quantityOrdered), productLine, SUM(quantity\*priceEach) AS Chiffres**

**FROM orderdetails**

**INNER JOIN products ON orderdetails.productCode = products.productCode**

**INNER JOIN orders ON orderdetails.orderNumber = orders.orderNumber**

**WHERE orderDate >= '2005-01-01'**

**GROUP BY products.productLine;**

**+----------------------+------------------+-----------+----------------+**

**| SUM(quantityOrdered) | productLine | Chiffres |**

**+----------------------+------------------+-----------+**

**| 6735 | Classic Cars | 715953.54 |**

**| 2771 | Motorcycles | 245273.04 |**

**| 2219 | Planes | 172881.88 |**

**| 1379 | Ships | 104490.16 |**

**| 409 | Trains | 26425.34 |**

**| 1921 | Trucks and Buses | 182066.45 |**

**| 4156 | Vintage Cars | 323846.30 |**

**+----------------------+------------------+-----------+**

* Afficher les identifiants, les dates et statuts des commandes, le nom du client ayant passé commande, ainsi que le nom, le code et la quantité d'articles commandé.

**SELECT orderNumber, orderDate, status**

**FROM orders**

**UNIONSELECT productName, products.productCode, quantityOrdered**

**FROM products**

**INNER JOIN orderdetails ON products.productCode = orderdetails.productCode;**

* À partir des tables customers, employees, offices, orders, orderDetails, products afficher le nom du client, la ville et le pays du bureau, le prénom, le nom et le poste occupé par l'employé, les articles commandés, leur quantité, la ligne de produit et le prix d'achat.



Alexis ROUYER - formateur

Ludovic DOMENICI – formateur Développement logiciel

Ch. Perrachon – Ingénieure de formation

**Date de mise à jour** : 21/01/20