

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur(s) | Action(s) |
| 1.0 | 23/09/21 | Rouyer Alexis | Création du document |

**Préambule**

**Objectifs**

Évaluer les requêtes SQL, les déclencheurs, fonctions et procédures stockées dans un système de gestion de bases de données

**Méthodologie**

Le temps imparti pour mener à bien le document est de 15h.

* **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.

**La base de donnée est composé de 8 tables.**

**Les clés sont :**

**Table "customers" = Primary = customerNumber**

**Index = salesRepEmployeeNumber**

**Table "employees" = Primary = employeeNumber**

**Index = officeCode, reportsTo**

**Table "offices" = Primary = officeCode**

**Table "orderDetails" = Primary = orderNumber, productCode**

**Index = productCode**

**Table "orders" = Primary = orderNumber**

**Index = customerNumber**

**Table "payments" = Primary = customerNumber, checkNumber**

**Table "productlines" = Primary = productLine**

**Table "Products" = Primary = productCode**

**Index = productLine**

**Description des tables:**

**customers = Cette table contients les informations des clients, nom de société, adresse complète et un numero est attribué à chaque clients.**

**Numéro qui est lié à la table "payments". Il y a aussi un numéro de vendeur attribué, lui est lié à la table "employees".**

**employees = Cette table contient la liste des employé, chaque employé a un numéro de bureau attribué qui est lié à la table "offices".**

**offices = Elle contient la liste de chaque bureau à qui appartient la BDD.**

**ordersdetails = La tableau contient la liste détaillé de toute les commandes, quantité commandé, prix et un numéro de commande est attribué à chaque commande et lié à la table "orders". il y a aussi le code produit qui lui est lié à la table "products".**

**orders = Cette table contients toute les commande avec les dates de celle-ci.**

**le numéro de client est aussi dans cette table pour pouvoir la lié à la table "customers".**

**payments = Cette table contient la liste des paiments, numéroté, daté et avec le prix. Il y a aussi les numéros des clients qui permets une liaison avec la table "customers".**

**productlines = Cette table contients les informations des sociétés qui produisent les produits vendu. La colonne productLine peut être lié avec la table products.**

**products = Cette table contient les informations en détail des produits jusqu'au nombre restant en stock.**

**Comme un inventaire.**

**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.

**3. Manipulations**

**3.1 Interrogations - Requêtes**

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

**SELECT COUNT(\*)**

**FROM nom\_de\_la\_table; (Par exemple: customers)**

**customers:** 122 enregistrements

**employees:** 23 enregistrements

**offices:** 7 enregistrements

**orders:** 326 enregistrements

**payments:** 273 enregistrements

**products:** 110 enregistrements

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

**SELECT quantityInStock, productName**

**FROM products**

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

**ORDER BY quantityInStock DESC;**

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

**| quantityInStock | productName |**

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

**| 7933 | 1969 Harley Davidson Ultimate Chopper |**

**| 5582 | 2003 Harley-Davidson Eagle Drag Bike |**

**| 4357 | 1936 Harley Davidson El Knucklehead |**

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

* 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 contactFirstName;**

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

**| contactFirstName |**

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

**| Carine |**

**| Carmen |**

**| Catherine |**

**| Dan |**

**| Daniel |**

**| Daniel |**

**| Hanna |**

**| Jan |**

**| Janine |**

**| Kalle |**

**| Karin |**

**| Laurence |**

**| Maria |**

**| Marie |**

**| Marta |**

**| Martha |**

**| Mart├¡n |**

**| Martine |**

**| Mary |**

**| Mary |**

**| Matti |**

**| Maurizio |**

**| Palle |**

**| Paolo |**

**| Pascale |**

**| Patricia |**

**| Paul |**

**| Raanan |**

**| Rachel |**

**| Sarah |**

**| Valarie |**

**| Valarie |**

**| Wales |**

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

* Combien y en a-t-il ?

**SELECT COUNT(contactFirstName)**

**FROM customers**

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

**Il y en a 33.**

* 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';**

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

**| Les articles | Les prix |**

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

**| 1957 Chevy Pickup | 55.70 |**

**| 1969 Dodge Charger | 58.73 |**

**| 1940 Ford Pickup Truck | 58.33 |**

**| 1937 Lincoln Berline | 60.62 |**

**| 1948 Porsche 356-A Roadster | 53.90 |**

**| 1932 Model A Ford J-Coupe | 58.48 |**

**| 1999 Indy 500 Monte Carlo SS | 56.76 |**

**| 1913 Ford Model T Speedster | 60.78 |**

**| 1999 Yamaha Speed Boat | 51.61 |**

**| 18th Century Vintage Horse Carriage | 60.74 |**

**| 1985 Toyota Supra | 57.01 |**

**| 1917 Maxwell Touring Car | 57.54 |**

**| 1948 Porsche Type 356 Roadster | 62.16 |**

**| 1941 Chevrolet Special Deluxe Cabriolet | 64.58 |**

**| 1904 Buick Runabout | 52.66 |**

**| 1997 BMW R 1100 S | 60.86 |**

**| 1962 Volkswagen Microbus | 61.34 |**

**| 1969 Chevrolet Camaro Z28 | 50.51 |**

**| 2002 Chevy Corvette | 62.11 |**

**| 1936 Chrysler Airflow | 57.46 |**

**| 1980ÔÇÖs GM Manhattan Express | 53.93 |**

**| 1974 Ducati 350 Mk3 Desmo | 56.13 |**

**| American Airlines: B767-300 | 51.15 |**

**| ATA: B757-300 | 59.33 |**

**| F/A 18 Hornet 1/72 | 54.40 |**

**| The Titanic | 51.09 |**

**| The Queen Mary | 53.63 |**

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

* 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 >= '2004-07-01';**

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

**| Total [07/2004] |**

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

**| 4290600.94 |**

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

* 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 orderNumber, quantityOrdered**

**FROM orderdetails**

**WHERE quantityOrdered >= '50'**

**GROUP BY orderNumber;**

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

**| orderNumber | quantityOrdered |**

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

**| 10100 | 50 |**

**| 10105 | 50 |**

**| 10106 | 50 |**

**| 10113 | 50 |**

**| 10117 | 50 |**

**| 10121 | 50 |**

**| 10123 | 50 |**

**| 10126 | 50 |**

**| 10129 | 50 |**

**| 10131 | 50 |**

**| 10149 | 50 |**

**| 10153 | 50 |**

**| 10159 | 50 |**

**| 10160 | 50 |**

**| 10165 | 50 |**

**| 10168 | 50 |**

**| 10175 | 50 |**

**| 10176 | 50 |**

**| 10177 | 50 |**

**| 10195 | 50 |**

**| 10196 | 50 |**

**| 10197 | 50 |**

**| 10202 | 50 |**

**| 10210 | 50 |**

**| 10214 | 50 |**

**| 10220 | 50 |**

**| 10224 | 50 |**

**| 10229 | 50 |**

**| 10234 | 50 |**

**| 10250 | 50 |**

**| 10251 | 59 |**

**| 10257 | 50 |**

**| 10261 | 50 |**

**| 10268 | 50 |**

**| 10271 | 50 |**

**| 10273 | 50 |**

**| 10276 | 50 |**

**| 10280 | 50 |**

**| 10284 | 50 |**

**| 10288 | 50 |**

**| 10292 | 50 |**

**| 10301 | 50 |**

**| 10306 | 50 |**

**| 10309 | 50 |**

**| 10318 | 50 |**

**| 10322 | 50 |**

**| 10326 | 50 |**

**| 10330 | 50 |**

**| 10332 | 50 |**

**| 10339 | 55 |**

**| 10340 | 55 |**

**| 10341 | 55 |**

**| 10342 | 55 |**

**| 10347 | 50 |**

**| 10356 | 50 |**

**| 10360 | 50 |**

**| 10362 | 50 |**

**| 10363 | 50 |**

**| 10373 | 50 |**

**| 10377 | 50 |**

**| 10382 | 50 |**

**| 10386 | 50 |**

**| 10388 | 50 |**

**| 10390 | 50 |**

**| 10399 | 51 |**

**| 10400 | 64 |**

**| 10401 | 64 |**

**| 10402 | 55 |**

**| 10403 | 66 |**

**| 10404 | 64 |**

**| 10405 | 97 |**

**| 10406 | 61 |**

**| 10407 | 59 |**

**| 10409 | 61 |**

**| 10410 | 65 |**

**| 10412 | 54 |**

**| 10413 | 51 |**

**| 10414 | 56 |**

**| 10415 | 51 |**

**| 10417 | 66 |**

**| 10418 | 52 |**

**| 10419 | 55 |**

**| 10420 | 66 |**

**| 10422 | 51 |**

**| 10424 | 50 |**

**| 10425 | 55 |**

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

* 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;**

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

**| customerName |**

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

**| American Souvenirs Inc |**

**| ANG Resellers |**

**| Anton Designs, Ltd. |**

**| Asian Shopping Network, Co |**

**| Asian Treasures, Inc. |**

**| BG&E Collectables |**

**| Cramer Spezialit├ñten, Ltd |**

**| Der Hund Imports |**

**| Feuer Online Stores, Inc |**

**| Franken Gifts, Co |**

**| Havel & Zbyszek Co |**

**| Kommission Auto |**

**| Kremlin Collectables, Co. |**

**| Lisboa Souveniers, Inc |**

**| Messner Shopping Network |**

**| Mit Vergn├╝gen & Co. |**

**| Nat├╝rlich Autos |**

**| Porto Imports Co. |**

**| Precious Collectables |**

**| Raanan Stores, Inc |**

**| SAR Distributors, Co |**

**| Schuyler Imports |**

**| Stuttgart Collectable Exchange |**

**| Warburg Exchange |**

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

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

**SELECT COUNT(firstName), lastName, employeeNumber, jobTitle**

**FROM employees**

**WHERE officeCode IS NULL;**

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

**| COUNT(firstName) | lastName | employeeNumber | jobTitle |**

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

**| 0 | NULL | NULL | NULL |**

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

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

**SELECT COUNT(DISTINCT city)**

**FROM customers;**

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

**| COUNT(DISTINCT city) |**

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

**| 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).

**SELECT payments.customerNumber, contactLastName, SUM(amount) FROM payments**

**INNER JOIN customers**

**ON payments.customerNumber = customers.customerNumber**

**GROUP BY payments.customerNumber**

**ORDER BY SUM(amount) DESC**

**LIMIT 1;**

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

**| customerNumber | contactLastName | SUM(amount) |**

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

**| 141 | Freyre | 715738.98 |**

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

**2eme parti de la question:**

**SELECT contactLastName, SUM(amount)**

**FROM payments**

**INNER JOIN customers**

**ON payments.customerNumber = customers.customerNumber**

**GROUP BY payments.customerNumber**

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

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

**| contactLastName | SUM(amount) |**

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

**| Freyre | 715738.98 |**

**| Nelson | 584188.24 |**

**| Ferguson | 180585.07 |**

**| Young | 177913.95 |**

**| Natividad | 156251.03 |**

**| Graham | 154622.08 |**

**| Ashworth | 148410.09 |**

**| O'Hara | 137034.22 |**

**| Brown | 132340.78 |**

**| Saveley | 130305.35 |**

**| Rovelli | 127529.69 |**

**| Henriot | 126983.19 |**

**| Labrune | 116949.68 |**

**| Young | 116449.29 |**

**| Snowden | 114497.19 |**

**| Sommer | 112440.09 |**

**| Holz | 108777.92 |**

**| Lee | 107639.94 |**

**| Petersen | 107446.50 |**

**| Shimamura | 105548.73 |**

**| Victorino | 105420.57 |**

**| Hashimoto | 104545.22 |**

**| Bergulfsen | 104224.79 |**

**| Suominen | 103896.74 |**

**| Huang | 101872.52 |**

**| Leong | 98509.25 |**

**| Oeztan | 97562.47 |**

**| Koskitalo | 95706.15 |**

**| Karttunen | 95546.46 |**

**| King | 95424.63 |**

**| Young | 93803.30 |**

**| Huxley | 91655.61 |**

**| Hernandez | 90545.37 |**

**| Moroni | 90143.31 |**

**| Ibsen | 89909.80 |**

**| Pfalzheim | 89223.14 |**

**| Benitez | 88627.49 |**

**| Cruz | 87468.30 |**

**| Bertrand | 86553.52 |**

**| Perrier | 86436.97 |**

**| Pipps | 85060.00 |**

**| Accorti | 82223.23 |**

**| Brown | 80556.73 |**

**| Thompson | 80375.24 |**

**| King | 80180.98 |**

**| Frick | 78432.16 |**

**| Salazar | 77726.59 |**

**| Larsson | 76776.44 |**

**| Keitel | 75937.76 |**

**| Citeaux | 75859.32 |**

**| Nelson | 73533.65 |**

**| Frick | 72497.64 |**

**| Bennett | 71783.75 |**

**| Lebihan | 71547.53 |**

**| Murphy | 70378.65 |**

**| Tamuri | 70122.19 |**

**| Hernandez | 69214.33 |**

**| Klaeboe | 69059.04 |**

**| Fresni├¿re | 68977.67 |**

**| Saavedra | 68520.47 |**

**| Da Silva | 67659.19 |**

**| Lincoln | 66812.00 |**

**| Murphy | 66710.56 |**

**| McRoy | 66704.94 |**

**| Franco | 65541.74 |**

**| Kentary | 62361.22 |**

**| Dewey | 62253.85 |**

**| Ranc├® | 61781.70 |**

**| Devon | 61167.18 |**

**| Roulet | 60483.36 |**

**| Frick | 59551.38 |**

**| Tonini | 58876.41 |**

**| Cervantes | 57939.34 |**

**| MacKinlay | 56662.38 |**

**| Barajas | 55577.26 |**

**| Calaghan | 55190.16 |**

**| Mendel | 51059.99 |**

**| Roel | 50987.85 |**

**| Lewis | 49967.78 |**

**| Cassidy | 49898.27 |**

**| Fernandez | 46751.14 |**

**| Gao | 45480.79 |**

**| Clenahan | 44920.76 |**

**| Thompson | 43748.72 |**

**| Berglund | 43680.65 |**

**| Taylor | 42570.37 |**

**| Chandler | 41506.19 |**

**| Brown | 38281.51 |**

**| Yoshido | 33533.47 |**

**| Tseng | 32198.69 |**

**| Donnermeyer | 31310.09 |**

**| Smith | 29586.15 |**

**| Choi | 29230.43 |**

**| Cartrain | 29217.18 |**

**| Ricotti | 25358.32 |**

**| Schmitt | 22314.36 |**

**| Taylor | 21554.26 |**

**| Young | 7918.60 |**

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

* 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(quantityOrdered\*priceEach)**

**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 productLine;**

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

**| SUM(quantityOrdered) | productLine | SUM(quantityOrdered\*priceEach) |**

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

**| 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 orders.orderNumber, orders.orderDate, orders.status, products.productName, orderdetails.productCode, orderdetails.quantityOrdered, customers.customerName**

**FROM orders**

**INNER JOIN orderdetails**

**ON orders.orderNumber = orderdetails.orderNumber**

**INNER JOIN customers**

**ON orders.customerNumber = customers.customerNumber**

**INNER JOIN products**

**ON orderdetails.productCode = products.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.

**SELECT employees.lastName, employees.firstName, employees.jobTitle, ' ', customers.customerName, customers.city, customers.country, ' ', products.productName, orderdetails.quantityOrdered, SUM(orderdetails.priceEach)**

**FROM employees**

**RIGHT JOIN customers**

**ON employees.employeeNumber = customers.salesRepEmployeeNumber**

**RIGHT JOIN orders**

**ON customers.customerNumber = orders.customerNumber**

**RIGHT JOIN orderdetails**

**ON orders.orderNumber = orderdetails.orderNumber**

**RIGHT JOIN products**

**ON orderdetails.productCode = products.productCode**

**GROUP BY customers.customerName;**



Alexis ROUYER - formateur

Ludovic DOMENICI – formateur Développement logiciel

Ch. Perrachon – Ingénieure de formation

**Date de mise à jour** : 21/01/20