

1. Requêtes SELECT simples : La liste des bureaux (adresse et ville) triés par pays décroissant puis par état

```
SELECT addressLine1, addressLine2, city, country, state
FROM offices
ORDER BY country DESC, state
```

/\* RESULTAT ==> 7 lignes / 100 Market Street \*/

2. Clause WHERE : La liste des avions (code et nom) triés par vendeur et par quantité en stock décroissants

```
SELECT productCode, productName
FROM products
WHERE productLine = 'Planes'
ORDER BY productVendor DESC, quantityInStock DESC
```

/\* RESULTAT ==> 12 lignes / 1900s Vintage Tri-Plane \*/

3. La liste des produits (nom, vendeur et prix de vente) qui sont vendus au moins 132\$ triés par nom du produit

```
SELECT productName, productVendor, MSRP
FROM products
WHERE MSRP >= 132
ORDER BY productName
```

/\* RESULTAT ==> 24 lignes / 1903 Ford Model A \*/

4. La liste des produits (code, nom, échelle et quantité) qui ont une échelle soit de 1:10, soit de 1:18 triés par quantité en stock décroissante

```
SELECT productCode, productName, productScale, quantityInStock
FROM products
WHERE productScale IN ('1:10', '1:18') /* équivalent à WHERE productScale = '1:10' OR productScale = '1:18' */
ORDER BY quantityInStock DESC
```

/\* RESULTAT ==> 48 lignes / 1995 Honda Civic \*/

5. La liste des produits (code, nom et prix d'achat) des produits achetés au moins 60\$ au plus 90\$ triés par prix d'achat

```
SELECT productCode, productName, buyPrice
FROM products
WHERE buyPrice BETWEEN 60 AND 90 /* équivalent à WHERE buyPrice >= 60 AND buyPrice <= 90 */
ORDER BY buyPrice
```

/\* RESULTAT ==> 34 lignes / 1937 Lincoln Berline \*/

6. Colonnes calculées : La liste des motos (nom, vendeur, quantité et marge) triés par marge décroissante

```
SELECT productName, productVendor, quantityInStock, (MSRP - buyPrice) AS margin
FROM products
WHERE productLine = 'Motorcycles'
ORDER BY margin DESC
```

/\* RESULTAT ==> 13 lignes / 2003 Harley-Davidson Eagle Drag Bike \*/

7. La liste des commandes (numéro de commande, date de commande, date d'expédition, écart en jours entre la date de commande et la date d'expédition, statut de la commande) soit qui sont en cours de traitement, soit qui ont été expédiées plus de 10 jours après la date de commande triés par écart décroissant puis par date de commande

```
SELECT orderNumber, orderDate, shippedDate, (shippedDate - orderDate) AS processTime, status
FROM orders
WHERE status = 'In Process' OR (status = 'Shipped' AND (shippedDate - orderDate) > 10)
ORDER BY processTime DESC, orderDate
```

/\* RESULTAT ==> 33 lignes / 10165 \*/

8. La liste des produits (nom et valeur du stock à la vente) des années 1960

```
SELECT productName, (quantityInStock * MSRP) AS stockValue FROM products
WHERE productName LIKE '196%'
```

/\* RESULTAT ==> 16 lignes / 1969 Harley Davidson Ultimate Chopper \*/

9. Clause GROUP BY : Le prix moyen d'un produit vendu par chaque vendeur triés par prix moyen décroissant

```
SELECT productVendor, AVG(MSRP) AS averagePrice
FROM products
GROUP BY productVendor
ORDER BY averagePrice DESC
```

/\* RESULTAT ==> 13 lignes / Welly Diecast Productions / 113.9325 \*/

10. Le nombre de produits pour chaque ligne de produit

```
SELECT productLine, COUNT(productCode)
FROM products
GROUP BY productLine
```

```
/* RESULTAT ==> 7 lignes / Classic Cars / 38 */
```

11. Clause GROUP BY : Le total du stock et le total de la valeur du stock à la vente de chaque ligne de produit pour les produits vendus plus de 100\$ trié par total de la valeur du stock à la vente

```
SELECT productLine, SUM(quantityInStock) AS totalStock, SUM(quantityInStock * MSRP) AS totalStockValue FROM products
WHERE MSRP > 100
GROUP BY productLine
ORDER BY totalStockValue
```

```
/* RESULTAT ==> 7 lignes / Ships / 429177.74 */
```

12. La quantité du produit le plus en stock de chaque vendeur trié par vendeur

```
SELECT productVendor, MAX(quantityInStock) AS maxInStock
FROM products
GROUP BY productVendor
ORDER BY productVendor
```

```
/* RESULTAT ==> 13 lignes / Autoart Studio Design / 9354 */
```

13. Le prix de l'avion qui coûte le moins cher à l'achat

```
SELECT MIN(buyPrice) AS cheapestPricePlane
FROM products
WHERE productLine = 'Planes'
```

```
/* RESULTAT ==> 1 ligne / 29.34$ */
```

14. Le crédit des clients qui ont payé plus de 20000\$ durant l'année 2004 trié par crédit décroissant

```
SELECT customerNumber, SUM(amount) AS totalCredit
FROM payments
WHERE paymentDate BETWEEN '2004-01-01' AND '2004-12-31'
GROUP BY customerNumber
HAVING totalCredit > 20000
ORDER BY totalCredit DESC
```

```
/* RESULTAT ==> 69 lignes / 141 / 293 765.51 */
```

15. Clause INNER JOIN : <https://sql.sh/cours/jointures> La liste des employés (nom, prénom et fonction) et des bureaux (adresse et ville) dans lequel ils travaillent

```
SELECT lastName, firstName, jobTitle, addressLine1, addressLine2, city
FROM employees
INNER JOIN offices ON offices.officeCode = employees.officeCode
```

```
/* RESULTAT ==> 23 lignes / Diane Murphy */
```

16. La liste des clients français ou américains (nom du client, nom, prénom du contact et pays) et de leur commercial dédié (nom et prénom) triés par nom et prénom du contact

```
SELECT customerName, contactLastName, contactFirstName, country, lastName, firstName
FROM customers
INNER JOIN employees ON employees.employeeNumber = customers.salesRepEmployeeNumber
WHERE country IN ('France', 'USA')
ORDER BY contactLastName, contactFirstName
```

```
/* RESULTAT ==> 48 lignes / Miguel Barajas */
```

17. La liste des lignes de commande (numéro de commande, code, nom et ligne de produit) et la remise appliquée aux voitures ou motos commandées triées par numéro de commande puis par remise décroissante

```
SELECT orderNumber, orderdetails.productCode, productName, productLine, (MSRP - priceEach) AS discount
FROM orderdetails
INNER JOIN products ON products.productCode = orderdetails.productCode
WHERE productLine IN ('Classic Cars', 'Vintage Cars', 'Motorcycles')
ORDER BY orderNumber, discount DESC
```

```
/* RESULTAT ==> 2026 lignes / 34 */
```

18. Requetes select complexes : Le total des paiements effectués de chaque client (numéro, nom et pays) américain, allemand ou français de plus de 50000\$ trié par pays puis par total des paiements décroissant

```
SELECT customers.customerNumber, customerName, country, SUM(amount) AS totalPayment
FROM customers
```

```
INNER JOIN payments ON payments.customerNumber = customers.customerNumber
WHERE country IN ('France', 'Germany', 'USA')
GROUP BY customers.customerNumber, customerName, country
HAVING totalPayment > 50000
ORDER BY country, totalPayment DESC
```

/\* RESULTAT ==> 38 lignes / 146 / 130305.35 \*/

19. Le montant total de chaque commande (numéro et date) des clients New-Yorkais (nom) trié par nom du client puis par date de commande

```
SELECT customerName, orders.orderNumber, orderDate, SUM(quantityOrdered * priceEach) AS totalAmount
FROM customers
INNER JOIN orders ON orders.customerNumber = customers.customerNumber
INNER JOIN orderdetails ON orders.orderNumber = orderdetails.orderNumber
WHERE city = 'NYC'
GROUP BY city, customerName, orderNumber, orderDate
ORDER BY customerName, orderDate
```

/\* RESULTAT ==> 16 lignes / Classic Legends / 10115 / 21665.98 \*/