

1.SELECT orderNumber, productCode, quantityOrdered FROM orderdetails WHERE quantityOrdered > 30;

2.SELECT officeCode, city FROM `offices` WHERE city like 'L%';

3.SELECT officeCode, postalCode FROM `offices` WHERE postalCode like '9____';

4.SELECT productName, buyPrice FROM `products` WHERE buyPrice BETWEEN 50 AND 100;

5.SELECT officeCode, city FROM `offices` WHERE officeCode = 1 or officeCode = 2 or officeCode = 4;
SELECT officeCode, city FROM `offices` WHERE officeCode IN (1,2,4);

6.SELECT firstName, lastName, email FROM `employees` WHERE firstName not like 'N%';

7.SELECT c.customerName, c.phone, o.orderDate FROM `customers` c
LEFT JOIN orders o using(customerNumber);

8.SELECT customerNumber, year(paymentDate), SUM(amount)
FROM `payments` GROUP BY customerNumber, year(paymentDate)
order BY customerNumber, year(paymentDate);

10.SELECT DISTINCT(orderNumber), status FROM `orders`;

11.SELECT contactFirstName, country, od.quantityOrdered
FROM `customers` JOIN orders o using(customerNumber)
JOIN orderdetails od USING(orderNumber);

12.SELECT p.productName, DAYNAME(orderDate)
FROM products p JOIN orderdetails od USING(productCode)

JOIN orders o using(orderNumber) WHERE DAYNAME(orderDate) = 'Saturday';

13.SELECT c.customerName, od.orderNumber, od.productCode, (od.quantityOrdered*od.priceEach) as
totalPrice

FROM `customers` c JOIN orders o using(customerNumber)

JOIN orderdetails od USING(orderNumber) HAVING totalPrice > 2000;

14.SELECT c.customerName, o.orderNumber, e.firstName, p.paymentDate, (o.orderDate -
p.paymentDate) as Days

FROM `customers` c JOIN orders o using(customerNumber) JOIN employees e ON
c.salesRepEmployeeNumber = e.employeeNumber

JOIN payments p USING(customerNumber) HAVING Days > 30 ORDER BY Days;

15.SELECT c.customerNumber,e.firstName,e.lastName,SUM(p.amount) AS TotalPayments

FROM `employees` e

JOIN customers c on e.employeeNumber=c.salesRepEmployeeNumber

JOIN payments p on c.customerNumber=p.customerNumber

GROUP BY c.customerNumber, e.employeeNumber;

16.

17.SELECT c.customerName, SUM(od.quantityOrdered) AS Total

FROM Customers c

JOIN Orders o ON c.customerNumber = o.customerNumber

JOIN OrderDetails od ON o.orderNumber = od.orderNumber

GROUP BY c.customerNumber, c.customerName

HAVING Total > 1000

ORDER BY Total ASC;

18.SELECT e.employeeNumber,e.firstName,e.officeCode,o.city,

```
COUNT(*) AS employees_in_same_Office,SUM(COUNT(*)) OVER () AS totalEmployees
FROM `employees` e
JOIN offices o on e.officeCode=o.officeCode
GROUP BY e.employeeNumber, e.firstName, e.officeCode, o.city
ORDER BY employees_in_same_Office ASC;
```

```
19.SELECT *,row_number()over(PARTITION BY orderNumber order by quantityOrdered) rownumber
FROM `orderdetails`;
```

```
20.SELECT p.productName,p.productCode,p.quantityInStock,
row_number()over(order by p.quantityInStock ASC) row_num
FROM `products` p;
```

```
21.SELECT orderNumber,productCode,quantityOrdered,
sum(quantityOrdered)over(PARTITION BY orderNumber order by orderNumber) sum
FROM `orderdetails`
ORDER BY orderNumber ASC;
```

```
22.SELECT *,MIN(amount) OVER (order BY customerNumber)minimum_payment,
MAX(amount) OVER (order BY customerNumber)maximum_payment
FROM `payments`;
```

```
23.SELECT *,MIN(amount) OVER (PARTITION BY YEAR(paymentDate) ORDER BY MONTH(paymentDate))
sum_payment
FROM `payments`;
```