**[57-sql-problems](https://github.com/rgerhardt/57-sql-problems)**

1. Which shippers do we have?

We have a table called Shippers. Return all the fields from all the shippers

select \*

from shippers;

2. Certain fields from Categories In the Categories table, selecting all the fields using this SQL:

Select \* from Categories…will return 4 columns. We only want to see two columns, Category Name and Description.

select

category\_name,

description

from categories;

3. Sales Representatives

We’d like to see just the FirstName, LastName, and HireDate of all the employees with the Title of Sales Representative. Write a SQL statement that returns only

those employees.

select

first\_name,

last\_name,

Hire\_date

from employees;

4.Sales Representatives in the United States

Now we’d like to see the same columns as above, but only for those employees that both have the title of Sales Representative, and also are in the United States.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Select  first\_name,  last\_name,  Hire\_date  From employees  Where title = ‘sales Representative’ |  |  |
| And country =’USA’ |  |  |
| 5. Orders placed by specific EmployeeID  Show all the orders placed by a specific employee. The Employee ID for this Employee (Steven Buchanan) is 5.  SELECT  order\_id,  order\_date  FROM orders  where employee\_id= 5;  6. Suppliers and ContactTitles  In the Suppliers table, show the Supplier ID, Contact Name, and ContactTitle for those Suppliers whose ContactTitle is *not* Marketing Manager.  select  supplier\_id,  contact\_name,  contact\_title  from suppliers  where contact\_title!= 'Marketing Manager'; |  |  |
| 7. Products with “queso” in ProductName  In the products table, we’d like to see the Product ID and Product Name for those products where the ProductName includes the string “queso”. |  |  |
| select  product\_id,  product\_name  from products  where product\_name like '%queso%';  8. Orders shipping to France or Belgium  Looking at the Orders table, there’s a field called Ship Country. Write a query that shows the OrderID, Customer ID, and Ship Country for the orders where the ShipCountry is either France or Belgium. |  |  |
|  |  |  |

select

order\_id,

customer\_id,

ship\_country

from orders

whereship\_country = 'France' or 'Belgium';

9. Orders shipping to any country in Latin America

Now, instead of just wanting to return all the orders from France of Belgium, we want to show all the orders from any Latin American country. But we don’t have a list of Latin American countries in a table in the Northwind database. So, we’re going to just use this list of Latin

American countries that happen to be in the Orders table:

Brazil

Mexico

Argentina

Venezuela

It doesn’t make sense to use multiple Or statements anymore, it would

get too convoluted. Use the In statement.

SELECT

order\_id,

customer\_id,

ship\_country

FROM

orders

WHERE

ship\_country in ('Brazil', 'Mexico', 'Argentina', 'Venezuela');

10. Employees, in order of age

For all the employees in the Employees table, show the First Name, LastName, Title, and BirthDate. Order the results by Birth Date, so we have the oldest employees first.

select

first\_name,

last\_name,

title,

birth\_date

from employees

order by birth\_date desc;

11. Showing only the Date with a Date Time field

In the output of the query above, showing the Employees in order of Birth Date, we see the time of the Birth Date field, which we don’t want. Show only the date portion of the Birth Date field.

SELECT

first\_name,

last\_name,

title,

birth\_date

FROM employees;

12. Employees full name

Show the First Name and Last Name columns from the Employees table, and then create a new column called Full Name, showing First Name and Last Name joined together in one column, with a space in-between.

SELECT

first\_name,

last\_name,

first\_name +''+last\_name as full\_name

FROM

employees;

13. Order Details amount per line item

In the Order Details table, we have the fields Unit Price and Quantity. Create a new field, Total Price,that multiplies these two together. We’ll ignore the Discount field for now. In addition, show the Order ID, Product ID, Unit Price, and Quantity. Order by Order ID and Product ID.

SELECT

order\_id,

product\_id,

unit\_price,

quantity, (unit\_price \* quantity) as total\_price

FROM

order\_details

ORDER BY

order\_id, product\_id;

14. How many customers?

How many customers do we have in the Customers table? Show one value only, and don’t rely on getting the record count at the end of a result set.

select

count (customer\_id)

from customers;

15. When was the first order?

Show the date of the first order ever made in the Orders table.

select

min(order\_date)

from orders;

16. Countries where there are customers

Show a list of countries where the North wind company has customers.

select

country

from customers;

17. Contact titles for customers

Show a list of all the different values in the Customers table for Contact Titles. Also include a count for each Contact Title. This is similar in concept to the previous question “Countries where there are customers”, except we now want a count for each Contact Title.

select

contact\_title,

count(contact\_title) as total

from customers

group by contact\_title

order by total DESC;

18. Products with associated supplier names

We’d like to show, for each product, the associated Supplier. Show the Product ID, Product Name, and the Company Name of the Supplier. Sort by Product ID.

This question will introduce what may be a new concept, the Join clause in SQL. The Join clause is used to join two or more relational database tables together in a logical way.

Select

p.product\_id,

p.product\_name,

s.company\_name

from product p

inner join suppliers s ON p.supplier\_id = s.supplier\_id;

19.Orders and the Shipper that was used

We’d like to show a list of the Orders that were made, including the Shipper that was used. Show the Order ID, Order Date (date only), and Company Name of the Shipper, and sort by Order ID. In order to not show all the orders (there’s more than 800), show only those rows with an Order ID of less than 10300.

select

o.order\_id,

o.order\_date,

s.company\_name

from orders o

inner join shippers s on o.order\_id = shippers\_id;