

# SQLite Command

**to\_char(istenen\_sütun,'999,999.99') = Bu noktadan sonra 2 hane verir.**

```
SELECT SupplierID, SUM(UnitsInStock)AS "Inventory"
FROM "alanparadise/nw"."products"
GROUP BY SupplierID
ORDER BY 2 DESC LIMIT 1;
# Burada 2 ikinci sütun demektir.
```

```
SELECT country, count(customerid) as "Total"
FROM "alanparadise/nw"."customers"
GROUP BY country
HAVING count(customerid) > 5
ORDER BY 2 DESC;
# WHERE ve HAVING deyimleri arasındaki temel fark şudur: WHERE satırları, gruplar ve
ortak değerler hesaplanmadan önce seçer (ortak değer hesaplamasında kullanılacak
satırları seçer), HAVING deyimini ise ortak değerler hesaplandıktan ve gruplamalar
yapıldıktan sonra işleme sokulur.
```

```
SELECT country, count(customerid) as "Total"
FROM "alanparadise/nw"."customers"
GROUP BY country
HAVING count(customerid) > 5
ORDER BY 2 DESC;
# WHERE ve HAVING deyimleri arasındaki temel fark şudur: WHERE satırları, gruplar ve
ortak değerler hesaplanmadan önce seçer (ortak değer hesaplamasında kullanılacak
satırları seçer), HAVING deyimini ise ortak değerler hesaplandıktan ve gruplamalar
yapıldıktan sonra işleme sokulur.
```

**2.1** List the total (unitprice \* quantity) as "Total Value" by orderid for the top 5 orders. (That is, the five orders with the highest Total Value.)

```
SELECT OrderID,
       SUM(UnitPrice * Quantity) AS "Total Value"
FROM "alanparadise/nw"."orderdetails"
GROUP BY OrderID
ORDER BY 2 DESC LIMIT 5;
```

**2.2** How many products does Northwinds have in inventory?

```
SELECT count(productid)
FROM "alanparadise/nw"."products"
Where unitsinstock>0;
```

**2.3** How many products are out of stock?

```
SELECT count(productid)
FROM "alanparadise/nw"."products"
where unitsinstock=0;
```

## 2.4 From which supplier(s) does Northwinds carry the fewest products?

```
SELECT supplierid, count(productid)
FROM "alanparadise/nw"."products"
GROUP BY supplierid
ORDER BY 2;
```

## 2.5 Which Northwinds employees (just show their employeeid) had over 100 orders ?

```
SELECT employeeid, count(*) as orders
FROM "alanparadise/nw"."orders"
GROUP BY employeeid
HAVING count(*)>100
ORDER BY 2 DESC;
```

```
select productID, productname, unitprice
from "alanparadise/nw"."products"
where UnitPrice = (
select MAX(UnitPrice)
from "alanparadise/nw"."products" )
```

pp

## SQL SubQueries

Find the productID, name and unit price of the highest priced product Northwinds sells.

```
select productID, productname, unitprice
from "alanparadise/nw"."products"
where UnitPrice = (
select MAX(UnitPrice)
from "alanparadise/nw"."products" )
```

Find the CustomerID and the OrderID for all orders with more than 100 units sold.

```
select customerID, orderID
from "alanparadise/nw"."orders"
where orderID in (
select orderID
from "alanparadise/nw"."orderdetails"
where quantity > 100 )
Order by CustomerID
```

List each product name and the total value of that product's orders

```
select productname, (select sum(UnitPrice * quantity)
from "alanparadise/nw"."orderdetails"
where "alanparadise/nw"."orderdetails".productid =
"alanparadise/nw"."products".productid) as "Total"
from "alanparadise/nw"."products"
```

Create a list of all orders with fewer than 100 items sold

```
SELECT orderID
FROM (SELECT orderID, SUM(quantity)
from "alanparadise/nw"."orderdetails"
GROUP BY OrderID
HAVING SUM(quantity) < 100) AS DetailCount;
```

Select all the employees where the employee had orders shipped to customers in the employee's own home city.

```
SELECT O.employeeID, orderID, shipcity, customerID
FROM "alanparadise/nw"."orders" O
WHERE employeeID IN
(SELECT employeeID FROM "alanparadise/nw"."employees" E
WHERE E.City = O.shipcity);
```

**2.1** List the productid, productname, unitprice of the lowest priced product Northwinds sells.

```
SELECT productid, productname, unitprice
FROM "alanparadise/nw"."products" O
WHERE unitprice = (SELECT unitprice FROM "alanparadise/nw"."products"
ORDER BY unitprice limit 1);
```

```
SELECT productid, productname, unitprice
FROM "alanparadise/nw"."products" O
WHERE unitprice = (SELECT unitprice FROM "alanparadise/nw"."products"
ORDER BY unitprice limit 1);
```

**2.2** How many orders in the orders table have a bad customerID (either missing or not on file in the customers table.)

```
SELECT customerid
FROM "alanparadise/nw"."orders"
where orderid in (SELECT orderid FROM "alanparadise/nw"."products"
WHERE reorderlevel=0)
group by customerid;
```

```
SELECT count(*)
FROM "alanparadise/nw"."orders"
```

```
where customerid not in (SELECT customerid
                        FROM "alanparadise/nw"."customers") ;
```

### 2.3 Use a subquery in a SELECT to list something interesting.

```
SELECT customerid
FROM "alanparadise/nw"."orders"
  where orderid in (SELECT orderid FROM "alanparadise/nw"."products"
                  WHERE reorderlevel=0)
group by customerid;
```

### 2.3 Use a subquery in a SELECT to list something interesting.

```
SELECT customerid, (SELECT employeeid*reportsto
                  FROM "alanparadise/nw"."employees"
                  WHERE "alanparadise/nw"."employees".employeeid="alanparadise/nw"."orders".employeeid) as ozel
FROM "alanparadise/nw"."orders";
```

### 2.4 Use a subquery in a FROM to list something interesting.