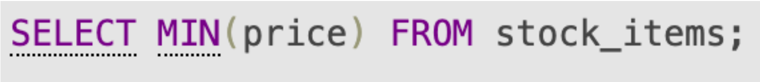
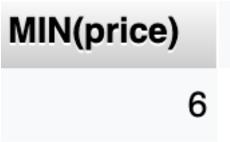
Part 1: MIN() & MAX() (1%)

A. "What is the lowest price of any item?" No need to provide the item, just the lowest price.

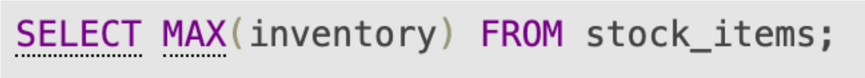


```
SELECT MIN(price) FROM stock_items;
```

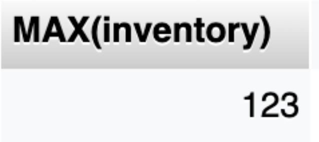


MIN(price)
6

B. "What is the greatest quantity of any item in stock? No need to provide the item, just the largest quantity.



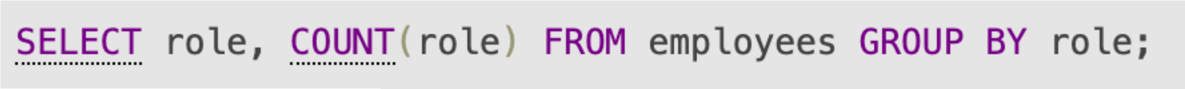
```
SELECT MAX(inventory) FROM stock_items;
```



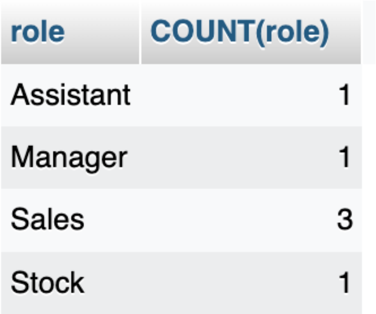
MAX(inventory)
123

Part 2: Get Into Groups (1%)

A. Provide a count of employees grouped by role. Include the role and the count in your results.

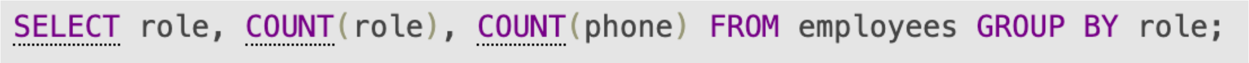


```
SELECT role, COUNT(role) FROM employees GROUP BY role;
```

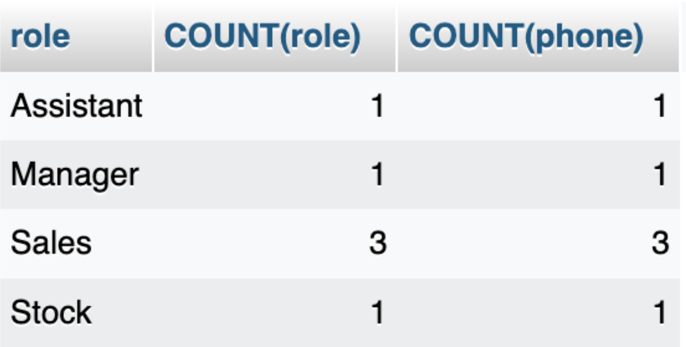


role	COUNT(role)
Assistant	1
Manager	1
Sales	3
Stock	1

B. Checking for contact info: Create a chart of employees grouped by role that shows the role, # of employees in that role, and # of phone numbers for that role.



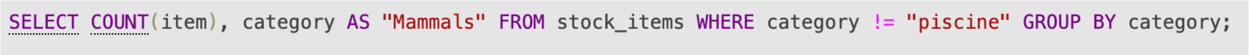
```
SELECT role, COUNT(role), COUNT(phone) FROM employees GROUP BY role;
```



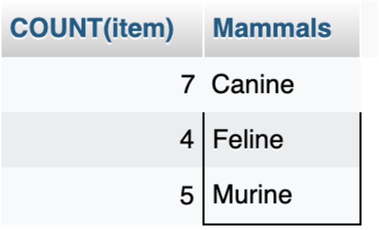
role	COUNT(role)	COUNT(phone)
Assistant	1	1
Manager	1	1
Sales	3	3
Stock	1	1

Part 3: Challenge (2%)

A. Create a count of items by category that excludes the fish category (“piscine”). Put “Mammals” as the heading for the category.

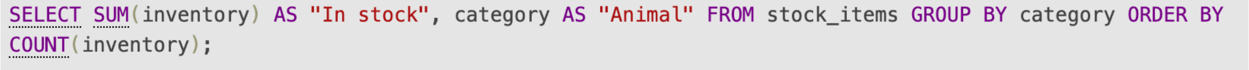


```
SELECT COUNT(item), category AS "Mammals" FROM stock_items WHERE category != "piscine" GROUP BY category;
```

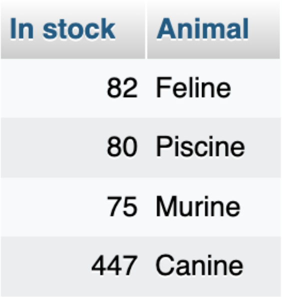


COUNT(item)	Mammals
7	Canine
4	Feline
5	Murine

B. Manager: “Inventory time. I need a total of in-stock items by each animal category, organized by fewest items to the most items. At the top of the chart I want to see ‘In stock’, and ‘Animal’.”

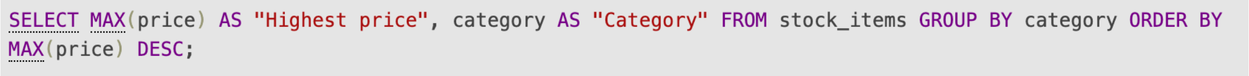


```
SELECT SUM(inventory) AS "In stock", category AS "Animal" FROM stock_items GROUP BY category ORDER BY
COUNT(inventory);
```

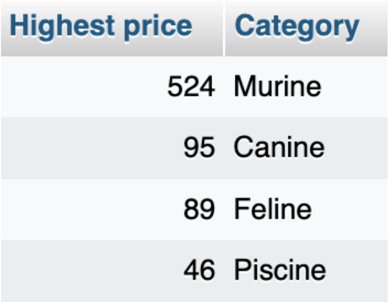


In stock	Animal
82	Feline
80	Piscine
75	Murine
447	Canine

C. Manager: What is the “highest price” from each category organized by the most expensive to least expensive? The results will show : “Highest price”, “Category.”

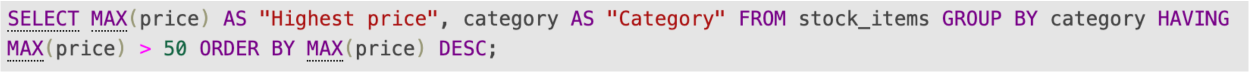


```
SELECT MAX(price) AS "Highest price", category AS "Category" FROM stock_items GROUP BY category ORDER BY
MAX(price) DESC;
```

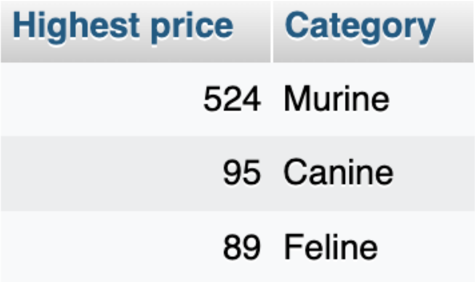


Highest price	Category
524	Murine
95	Canine
89	Feline
46	Piscine

D. Just as you go to the manager’s office with the results from C, the Manager says, Oh, I need the categories where the highest price is more than $50. Note: the result set should show “Highest Price”, “Category” at the top.



```
SELECT MAX(price) AS "Highest price", category AS "Category" FROM stock_items GROUP BY category HAVING
MAX(price) > 50 ORDER BY MAX(price) DESC;
```



Highest price	Category
524	Murine
95	Canine
89	Feline