Practical Exam Sample: Pet Supplies

PetMind is a retailer of products for pets. They are based in the United States.

PetMind sells products that are a mix of luxury items and everyday items. Luxury items include toys. Everyday items include food.

The company wants to increase sales by selling more products for some animals repeatedly.

They have been testing this approach for the last year.

They now want a report on how repeat purchases impact sales.

Data

The data is available in the table `pet_supplies`.

The dataset contains the sales records in the stores last year.

Column Name	Criteria
product_id	Nominal. The unique identifier of the product. Missing values are not possible due to the database structure.
category	Nominal. The category of the product, one of 6 values (Housing, Food, Toys, Equipment, Medicine, Accessory). Missing values should be replaced with "Unknown".
animal	Nominal. The type of animal the product is for. One of Dog, Cat, Fish, Bird. Missing values should be replaced with "Unknown".
size	Ordinal. The size of animal the product is for. Small, Medium, Large. Missing values should be replaced with "Unknown".
price	Continuous. The price the product is sold at. Can be any positive value, round to 2 decimal places. Missing values should be replaced with the overall median price.
sales	Continuous. The value of all sales of the product in the last year. This can be any positive value, rounded to 2 decimal places. Missing values should be replaced with the overall median sales.
rating	Discrete. Customer rating of the product from 1 to 10. Missing values should be replaced with 0.
repeat_purchase	Nominal. Whether customers repeatedly buy the product (1) or not (0). Missing values should be removed.

Certification Pet Supplies DB DataFrame as `pet_supplies`

```sql
SELECT * FROM public.pet_supplies
```

Task 1

From taking a quick look at the data, you are pretty certain it isn't quite as it should be. You need to make sure all of the data is clean before you start your analysis. The table below shows what the data should look like.

Write a query to return a table that matches the description provided.

Do not update the original table.

Column Name	Criteria
product_id	Nominal. The unique identifier of the product. Missing values are not possible due to the database structure.
category	Nominal. The category of the product, one of 6 values (Housing, Food, Toys, Equipment, Medicine, Accessory). Missing values should be replaced with “Unknown”.
animal	Nominal. The type of animal the product is for. One of Dog, Cat, Fish, Bird. Missing values should be replaced with “Unknown”.
size	Ordinal. The size of animal the product is for. Small, Medium, Large. Missing values should be replaced with “Unknown”.
price	Continuous. The price the product is sold at. Can be any positive value, round to 2 decimal places. Missing values should be replaced with 0.
sales	Continuous. The value of all sales of the product in the last year. This can be any positive value, rounded to 2 decimal places. Missing values should be replaced with the overall median sales.
rating	Discrete. Customer rating of the product from 1 to 10. Missing values should be replaced with 0.
repeat_purchase	Nominal. Whether customers repeatedly buy the product (1) or not (0). Missing values should be removed.

Certification Pet Supplies DB   DataFrame as `clean_data`

```
SELECT
    product_id,
    CASE WHEN category='-' THEN 'Unknown' ELSE category END AS category,
    animal,
    INITCAP(size) AS size,
    CASE WHEN price = 'unlisted' THEN 0 ELSE ROUND(CAST (price AS numeric),2) END AS price,
    COALESCE(sales, AVG(sales) ) AS sales,
    COALESCE (rating, 0) AS rating,
    repeat_purchase
FROM pet_supplies
WHERE repeat_purchase IS NOT NULL
GROUP BY product_id,category,animal,size,price, rating, repeat_purchase,sales
```

index	product_id	category	animal	size	price	sales	rating	repeat_purchase
0	1207	Equipment	Cat	Small	27.98	1039.91	8	
1	339	Food	Bird	Medium	41	1208.27	5	
2	590	Equipment	Fish	Small	23	704.78	4	
3	988	Toys	Cat	Small	34.06	1348.07	8	
4	721	Housing	Cat	Medium	20.83	612.54	4	
5	152	Food	Dog	Large	41.01	1464.8	4	
6	572	Toys	Cat	Small	34.23	1359.67	7	
7	424	Housing	Bird	Small	41.01	1191.23	5	
8	1006	Food	Dog	Small	36.09	1129.37	7	
9	1489	Equipment	Fish	Small	23.25	713.67	3	
10	823	Accessory	Cat	Small	23.13	717.93	6	
11	453	Toys	Fish	Small	28.98	973.92	6	
12	394	Medicine	Cat	Medium	0	745.83	0	
13	833	Toys	Cat	Small	33.88	1349.47	3	
14	346	Accessory	Bird	Medium	33.15	860.51	6	
15	306	Toys	Cat	Small	33.97	1349.03	6	

Rows: 1,500   Expand

Task 2

You want to show whether sales are higher for repeat purchases for different animals. You also want to give a range for the sales.

Write a query to return the `animal`, `repeat_purchase` indicator and the `avg_sales`, along with the `min_sales` and `max_sales`. All values should be rounded to whole numbers.

You should use the original `pet_supplies` data for this task.

Certification Pet Supplies DB DataFrame as `animal_sales`

```
SELECT
    animal,
    repeat_purchase,
    ROUND(AVG(sales)) AS avg_sales,
    ROUND(MIN(sales)) AS min_sales,
    ROUND(MAX(sales)) AS max_sales
FROM pet_supplies
GROUP BY animal, repeat_purchase
```

index	animal	repeat_purchase	avg_sales	min_sales	max_sales
0	Fish	1	693	287	
1	Bird	0	1380	858	
2	Dog	0	1084	574	
3	Dog	1	1038	574	
4	Cat	0	1035	512	
5	Bird	1	1408	853	
6	Fish	0	705	288	
7	Cat	1	998	512	

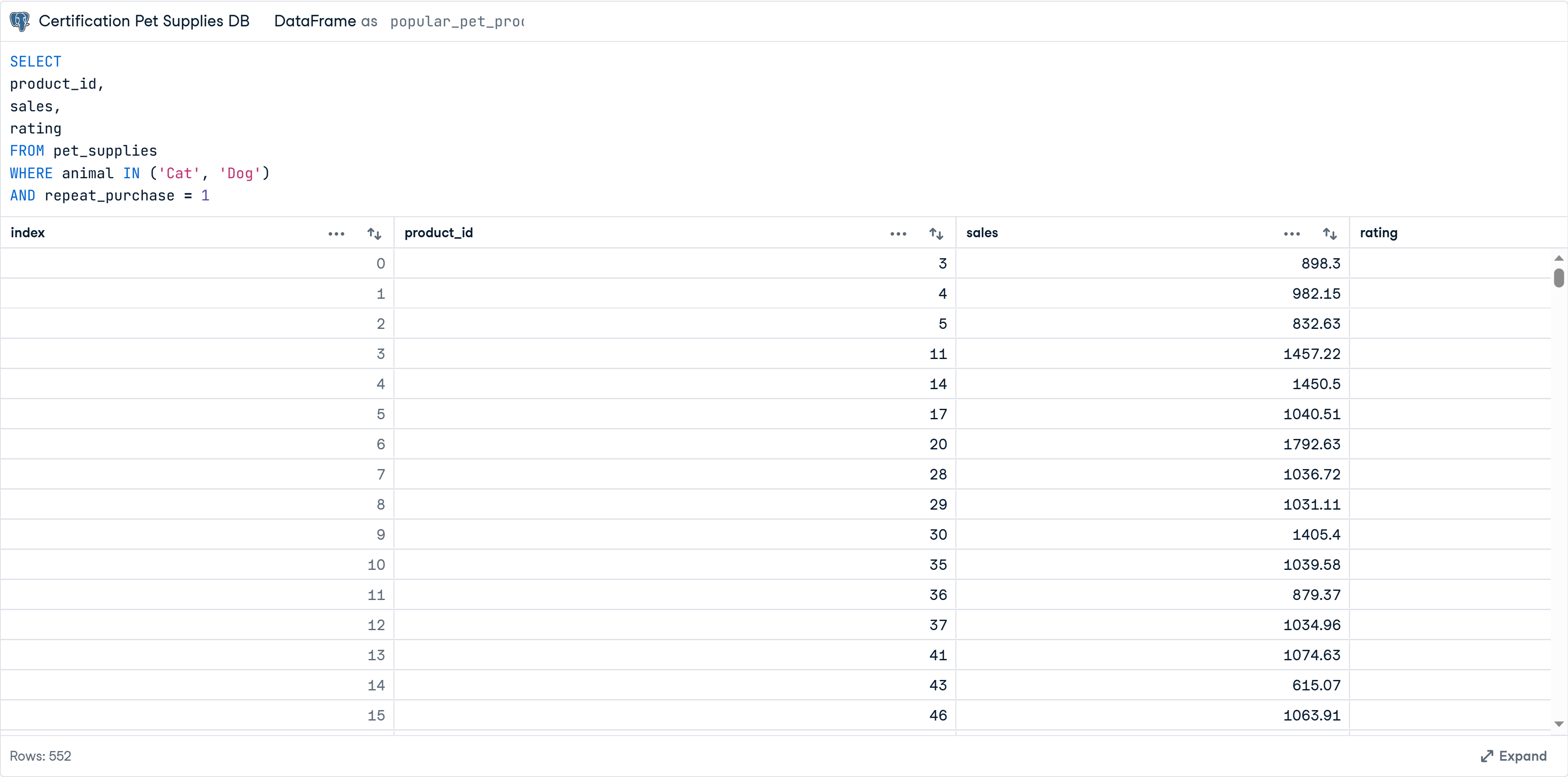
Rows: 8 Expand

Task 3

The management team want to focus on efforts in the next year on the most popular pets - cats and dogs - for products that are bought repeatedly.

Write a query to return the `product_id`, `sales` and `rating` for the relevant products.

You should use the original `pet_supplies` data for this task.



```sql
SELECT
product_id,
sales,
rating
FROM pet_supplies
WHERE animal IN ('Cat', 'Dog')
AND repeat_purchase = 1
```

index	product_id	sales	rating
0	3	898.3	
1	4	982.15	
2	5	832.63	
3	11	1457.22	
4	14	1450.5	
5	17	1040.51	
6	20	1792.63	
7	28	1036.72	
8	29	1031.11	
9	30	1405.4	
10	35	1039.58	
11	36	879.37	
12	37	1034.96	
13	41	1074.63	
14	43	615.07	
15	46	1063.91	

Rows: 552