Problem Statements

Following are the business questions that Emily wants answers to. Use SQL to answer them. All the best.

1. Emily would like to know **how many bikes the shop owns by category**. Can you get this for her?

   Display the **category name** and the **number of bikes the shop owns in each category** (call this column `number_of_bikes`). Show only the categories where the number of bikes is greater than `2`.

2. Emily needs a list of **customer names with the total number of memberships purchased by each**.

   For each customer, display the **customer's name** and the **count of memberships purchased** (call this column `membership_count`). Sort the results by `membership_count`, starting with the customer who has purchased the highest number of memberships.

   Keep in mind that some customers may not have purchased any memberships yet. In such a situation, display `0` for the `membership_count`.

3. Emily is working on a special offer for the winter months. Can you help her prepare a list of **new rental prices**?

   For each bike, display its **ID**, **category**, **old price per hour** (call this column `old_price_per_hour`), **discounted price per hour** (call it `new_price_per_hour`), **old price per day** (call it `old_price_per_day`), and **discounted price per day** (call it `new_price_per_day`).

   Electric bikes should have a **10% discount for hourly** rentals and a **20% discount for daily** rentals. Mountain bikes should have a **20% discount for hourly** rentals and a **50% discount for daily** rentals. All other bikes should have a **50% discount** for all types of rentals.

   Round the new prices to `2` decimal digits.

4. Emily is looking for **counts of the rented bikes and of the available bikes in each category**.

   Display the **number of available bikes** (call this column `available_bikes_count`) and the **number of rented bikes** (call this column `rented_bikes_count`) by bike category.

5. Emily is preparing a sales report. She needs to know the **total revenue from rentals by month, the total by year, and the all-time across all the years**.

   Display the total revenue from rentals for each month, the total for each year, and the total across all the years. **Do not take memberships into account.** There should be 3 columns: `year` , `month` , and `revenue` .

   Sort the results **chronologically**. Display the year total after all the month totals for the corresponding year. Show the all-time total as the last row.

   The resulting table looks something like this:

year	month	revenue
2022	11	200.00
2022	12	150.00
2022	null	350.00
2023	1	110.00
...		
2023	10	335.00
2023	null	1370.00
null	null	1720.00

6. Emily has asked you to get the **total revenue from memberships** for each combination of year, month, and membership type.

   Display the **year**, the **month**, the name of the **membership type** (call this column `membership_type_name`), and the **total revenue** (call this column `total_revenue`) for every combination of year, month, and membership type. Sort the results by year, month, and name of membership type.

7. Next, Emily would like data about **memberships purchased in 2023**, with subtotals and grand totals for all the different combinations of membership types and months.

   Display the **total revenue from memberships purchased in 2023 for each combination of month and membership type**. Generate subtotals and grand totals for all possible combinations. There should be 3 columns:

`membership_type_name` , `month` , and `total_revenue` .

Sort the results by membership type name **alphabetically** and then **chronologically** by month.

8. Now it's time for the final task.

   Emily wants to **segment customers based on the number of rentals** and see the **count of customers in each segment**. Use your SQL skills to get this!

   Categorize customers based on their rental history as follows:

   - Customers who have had more than 10 rentals are categorized as `'more than 10'` .
   - Customers who have had 5 to 10 rentals (inclusive) are categorized as `'between 5 and 10'` .
   - Customers who have had fewer than 5 rentals should be categorized as `'fewer than 5'` .

   Calculate the number of customers in each category. Display two columns: `rental_count_category` (the rental count category) and `customer_count` (the number of customers in each category).

*** Solutions to all these problems can be found in “SOLUTION Bike_Rental_Case_Study.txt” file.