

Sure, here are 10 tasks of varying difficulty covering the topics in the SQL text manipulation handout:

1. Easy

- Write a query to concatenate the `first_name` and `last_name` columns from the `customers` table, separating them with a space.
- Find the starting position of the '@' character in the `email` column of the `customers` table.
- Extract the first 3 characters from the `product_code` column in the `products` table.

2. Moderate

- Create a new column called `formatted_phone` that formats phone numbers from the `phone` column in the `customers` table as `(XXX) XXX-XXXX`.
- Write a `CASE` statement that categorizes products in the `products` table into 'Electronics', 'Furniture', and 'Other' based on keywords in the `product_name` column.
- Extract the initials (first initial and last initial) from the `first_name` and `last_name` columns in the `employees` table.

3. Challenging

- Pad the `order_id` column in the `orders` table with leading zeros to ensure a minimum length of 6 characters.
- Replace all occurrences of the substring '@example.com' in the `email` column of the `customers` table with an empty string to extract the usernames.
- Write a nested `CASE` statement that categorizes orders in the `orders` table based on the `order_status` and `ship_method` columns (e.g., 'Open', 'In Progress - FedEx', 'In Progress - USPS', 'Closed').

4. Advanced

- Split the `tags` column in the `products` table into separate rows using the `STRING_TO_TABLE()` function, and select the first 3 tags for each product.
- Create a function that takes a product description as input and returns the product category ('Camera', 'Lens', 'Tripod', 'Other') based on the presence of keywords in the description.