Aquí tienes **tres casos de uso para cada función de cadena**, aplicados a **análisis de datos**, más allá de la limpieza y transformación:

---

1. UPPER() y LOWER()

a. **Análisis de tendencias en opiniones**

- Convertir texto a mayúsculas/minúsculas para contar palabras clave en comentarios de clientes.

```
SELECT LOWER(review_text) AS comentario
FROM product_reviews;
```

b. **Comparación de datos sin distinción de mayúsculas/minúsculas**

- Identificar clientes duplicados con nombres escritos de distintas maneras.

```
SELECT customer_name
FROM orders
GROUP BY LOWER(customer_name)
HAVING COUNT(*) > 1;
```

c. **Clasificación de productos según iniciales**

- Agrupar productos que empiezan con la misma letra para análisis de ventas.

```
SELECT UPPER(LEFT(product_name, 1)) AS inicial, COUNT(*) AS cantidad
FROM products
GROUP BY inicial;
```

---

2. TRIM(), LTRIM(), RTRIM()

a. **Análisis de registros de acceso (logs)**

- Eliminar espacios extra en los nombres de usuarios antes de calcular actividad.

```
SELECT TRIM(username), COUNT(*) AS sesiones
FROM user_logs
GROUP BY TRIM(username);
```

b. **Detección de fraudes por inconsistencias en datos**

- Identificar datos mal ingresados donde haya espacios en números de tarjetas.

```
SELECT card_number
FROM payments
WHERE LENGTH(TRIM(card_number)) != 16;
```

c. **Optimización de consultas con índices**

- Asegurar que los valores en claves de búsqueda no tengan espacios que afecten el JOIN.

```
SELECT *
FROM sales s
JOIN customers c ON TRIM(s.customer_id) = TRIM(c.customer_id);
```

---

3. LEFT() y RIGHT()

a. **Análisis de segmentación geográfica**

- Extraer códigos de país de números de teléfono para analizar clientes por región.

```
SELECT LEFT(phone_number, 3) AS country_code, COUNT(*) AS total_clientes
FROM customers
GROUP BY country_code;
```

b. **Evaluación de estacionalidad en ventas**

- Extraer el mes de una fecha para detectar patrones de ventas.

```
SELECT RIGHT(sale_date, 2) AS mes, SUM(total_amount) AS ingresos
FROM sales
GROUP BY mes;
```

c. **Análisis de nombres de productos**

- Identificar tendencias en productos con nombres similares.

```
SELECT LEFT(product_name, 5) AS categoria, COUNT(*)
FROM products
GROUP BY categoria;
```

---

4. SUBSTRING()

a. **Extracción de códigos de productos**

- Obtener los primeros caracteres de un SKU para análisis de categoría.

```
SELECT SUBSTRING(sku, 1, 4) AS categoria, COUNT(*)
FROM inventory
GROUP BY categoria;
```

b. **Segmentación de clientes por dominio de correo**

- Analizar qué dominios de email son más usados por los clientes.

```
SELECT SUBSTRING(email, LOCATE('@', email) + 1) AS dominio, COUNT(*)
FROM customers
GROUP BY dominio;
```

c. **Análisis de documentos de identidad**

- Extraer el año de nacimiento de números de identificación.

```
SELECT SUBSTRING(id_number, 1, 4) AS año_nacimiento, COUNT(*)
FROM users
GROUP BY año_nacimiento;
```

---

5. REPLACE()

a. **Corrección de formatos de datos**

- Reemplazar guiones en números de teléfono para estandarización.

```
SELECT REPLACE(phone_number, '-', '') AS phone_standardized
FROM customers;
```

b. **Análisis de tendencias en comentarios**

- Reemplazar sinónimos en opiniones para facilitar análisis de sentimientos.

```
SELECT REPLACE(review_text, 'excelente', 'bueno') AS comentario_normalizado
FROM product_reviews;
```

c. **Cambio de moneda en reportes financieros**

- Convertir los símbolos de moneda en reportes de ventas.

```
SELECT REPLACE(price, '$', 'USD ') AS precio_en_dolares
FROM sales;
```

---

6. LOCATE()

a. **Detección de palabras clave en texto**

- Encontrar menciones de productos en opiniones de clientes.

```
SELECT review_text
FROM product_reviews
WHERE LOCATE('calidad', review_text) > 0;
```

b. **Identificación de anomalías en datos**

- Buscar caracteres inesperados en códigos de clientes.

```
SELECT customer_id
FROM customers
WHERE LOCATE(' ', customer_id) > 0;
```

c. **Análisis de tendencias en nombres de marcas**

- Contar cuántos productos contienen un nombre específico en su descripción.

```
SELECT COUNT(*)
FROM products
WHERE LOCATE('premium', product_name) > 0;
```

---

7. CONCAT()

a. **Generación de identificadores únicos**

- Crear un ID único combinando fecha y número de pedido.

```
SELECT CONCAT(order_date, '-', order_id) AS unique_order_id
FROM orders;
```

b. **Creación de reportes personalizados**

- Formatear nombres completos con título para informes.

```
SELECT CONCAT('Sr./Sra. ', first_name, ' ', last_name) AS nombre_formal
FROM customers;
```

c. **Comparación de nombres y apellidos para detectar duplicados**

- Verificar si hay clientes repetidos con el mismo nombre completo.

```
SELECT CONCAT(first_name, ' ', last_name) AS nombre_completo, COUNT(*)
FROM customers
GROUP BY nombre_completo
HAVING COUNT(*) > 1;
```

Estos **21 casos de uso** muestran cómo las funciones de cadena no solo sirven para **limpieza de datos**, sino también para **análisis, segmentación y generación de insights clave**. 🚀