

Sure, here's the handout with explanations and code samples for the additional string manipulation topics:

## SQL String Manipulation Handout

### Basic Operations with Strings

#### Concatenating Strings

Concatenating strings is a fundamental operation in SQL. You can use the `CONCAT()` function or the `||` operator to combine multiple string values into a single string.

```
-- Using CONCAT()
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM customers;

-- Using the || operator
SELECT first_name || ' ' || last_name AS full_name
FROM customers;
```

Both of these queries will create a new `full_name` column that combines the `first_name` and `last_name` columns, separated by a space.

#### Finding Position of a Substring

The `POSITION()` (or `INSTR()`) function can be used to find the starting position of a substring within a string.

```
SELECT POSITION('@' IN email) AS at_position
FROM customers;

SELECT INSTR(email, '@') AS at_position
FROM customers;
```

These queries will return the position of the '@' character within the `email` column.

#### Extracting Substrings

The `SUBSTRING()` (or `SUBSTR()`) function can be used to extract a portion of a string, based on the starting position and length.

```
SELECT SUBSTRING(email, 1, 5) AS first_5_chars
FROM customers;

SELECT SUBSTR(phone, 1, 3) AS area_code
FROM customers;
```

The `SUBSTRING()` function takes three arguments: the column name, the starting position (1-based index), and the number of characters to extract. The `SUBSTR()` function works similarly.

### Conditional Expressions: CASE

The `CASE` expression in SQL allows you to implement conditional logic and transform data based on specific conditions.

```
SELECT order_id,
       CASE order_status
         WHEN 'Pending' THEN 'Open'
         WHEN 'Shipping' THEN 'In Progress'
         WHEN 'Delivered' THEN 'Closed'
         ELSE 'Unknown'
       END AS order_status_text
FROM orders;
```

In this example, the `CASE` statement checks the value of the `order_status` column and returns a more user-friendly text representation.

You can also use `CASE WHEN` to check for multiple conditions:

```
SELECT product_name,
       CASE
         WHEN product_name LIKE 'Camera%' THEN 'Camera'
         WHEN product_name LIKE '%Lens' THEN 'Lens'
         WHEN product_name LIKE '%Tripod' THEN 'Tripod'
         ELSE 'Other'
```

```
END AS product_category  
FROM products;
```

This query categorizes products based on keywords in the `product_name` column.

Here's a breakdown of the `CASE` expression syntax:

```
CASE WHEN condition THEN result  
      WHEN condition THEN result  
      ...  
      ELSE result  
END
```

The `CASE` expression evaluates each `WHEN` condition in order and returns the corresponding `THEN` result when a condition is true. If no `WHEN` conditions are true, it returns the `ELSE` result (or `NULL` if no `ELSE` is provided).