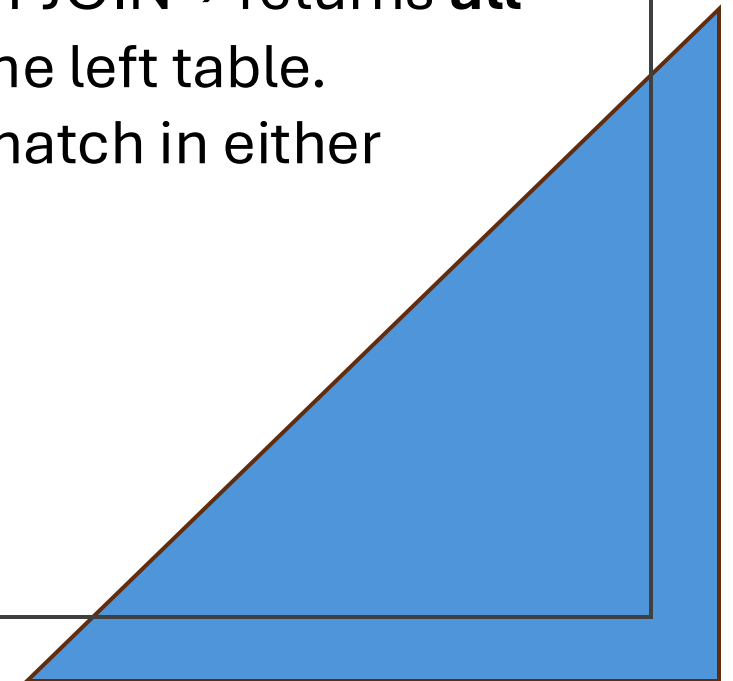


# JOINS

1. **INNER JOIN:** Returns rows where there is a match in **both tables**.
2. **LEFT JOIN** (or LEFT OUTER JOIN): Returns **all rows from the left table**, and matching rows from the right table. If no match, right-side columns are NULL.
3. **RIGHT JOIN** (or RIGHT OUTER JOIN): Opposite of LEFT JOIN → returns **all rows from the right table**, and matching rows from the left table.
4. **FULL OUTER JOIN:** Returns all rows when there is a match in either table. Non-matching rows will have NULLs.

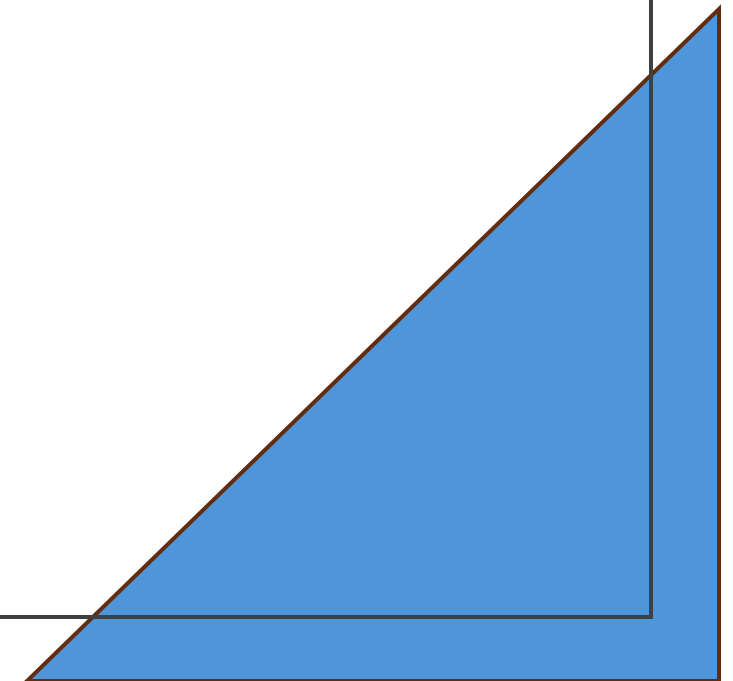


# Common Built-in Functions and Expressions

- 1. String Functions:** LEN(string), TRIM(string), UPPER(string) / LOWER(string), SUBSTRING(string, start, length), CONCAT(a, b, ...)
- 2. Date & Time Functions:** GETDATE(), DATEADD(part, number, date), DATEDIFF(part, start, end)
- 3. Aggregate Functions:** COUNT(\*), SUM(column), AVG(column), MIN(column)
- 4. Conversion & Formatting Functions:** CAST(expression AS datatype), FORMAT(value, format)
- 5. Logical Functions:** ISNULL(expr, replacement), COALESCE(expr1, expr2, ...), CASE WHEN ... THEN ... ELSE ... END

# Window Functions

- **ROW\_NUMBER()** → sequential numbering of rows
- **RANK()** → ranking with gaps for ties
- **DENSE\_RANK()** → ranking without gaps
- **NTILE(n)** → split rows into n groups
- **LAG(column, offset)** → look at previous row's value
- **LEAD(column, offset)** → look at next row's value



# Common Table Expression & Subquery

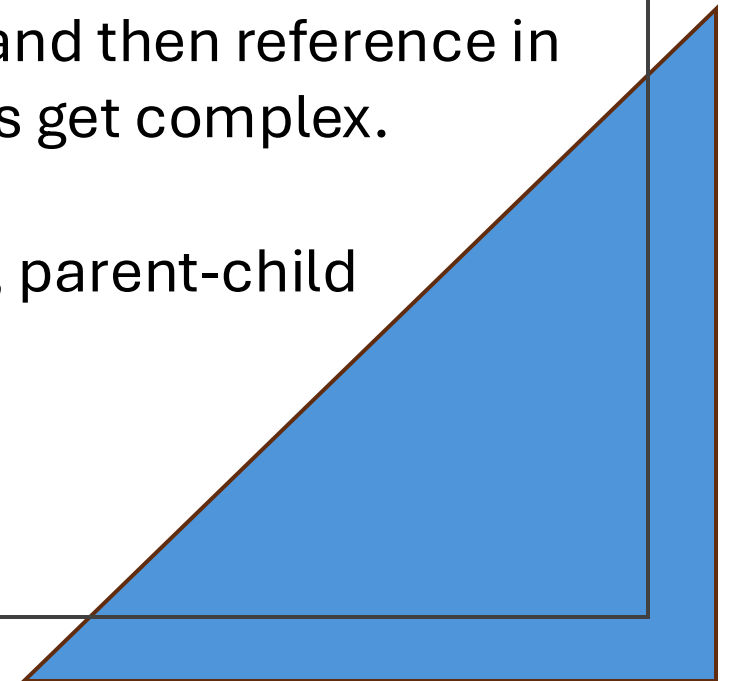
## Subqueries

A subquery is a query nested inside another query. It can appear in the SELECT, FROM, or WHERE clause.

## Common Table Expressions (CTEs)

A CTE is like a temporary result set you define with WITH and then reference in your query. They're cleaner than subqueries when queries get complex.

1. Simple CTE
2. Recursive CTE: Useful for hierarchical data (org charts, parent-child relationships).



# TASK

## Insert a blacklisted customer with no ID

Insert a record into the customers table where the customer\_id is NULL and the blacklisted flag is set to TRUE.

## Find percentage of free deliveries per customer

For each customer, calculate the percentage of their orders that were delivered **for free**.

Assume free delivery applies when the **order total is at least \$150**.

## List pending deliveries (excluding blacklisted customers)

The result should include: order\_id, customer\_address, & carrier

## Find the second highest purchase per customer

For each customer, return the order with the **second highest total\_amount**.

If a customer has fewer than 2 orders, exclude them.

<https://github.com/Stephen-Data-Engineer-Public/BDPN/tree/main/Dataset/Query>