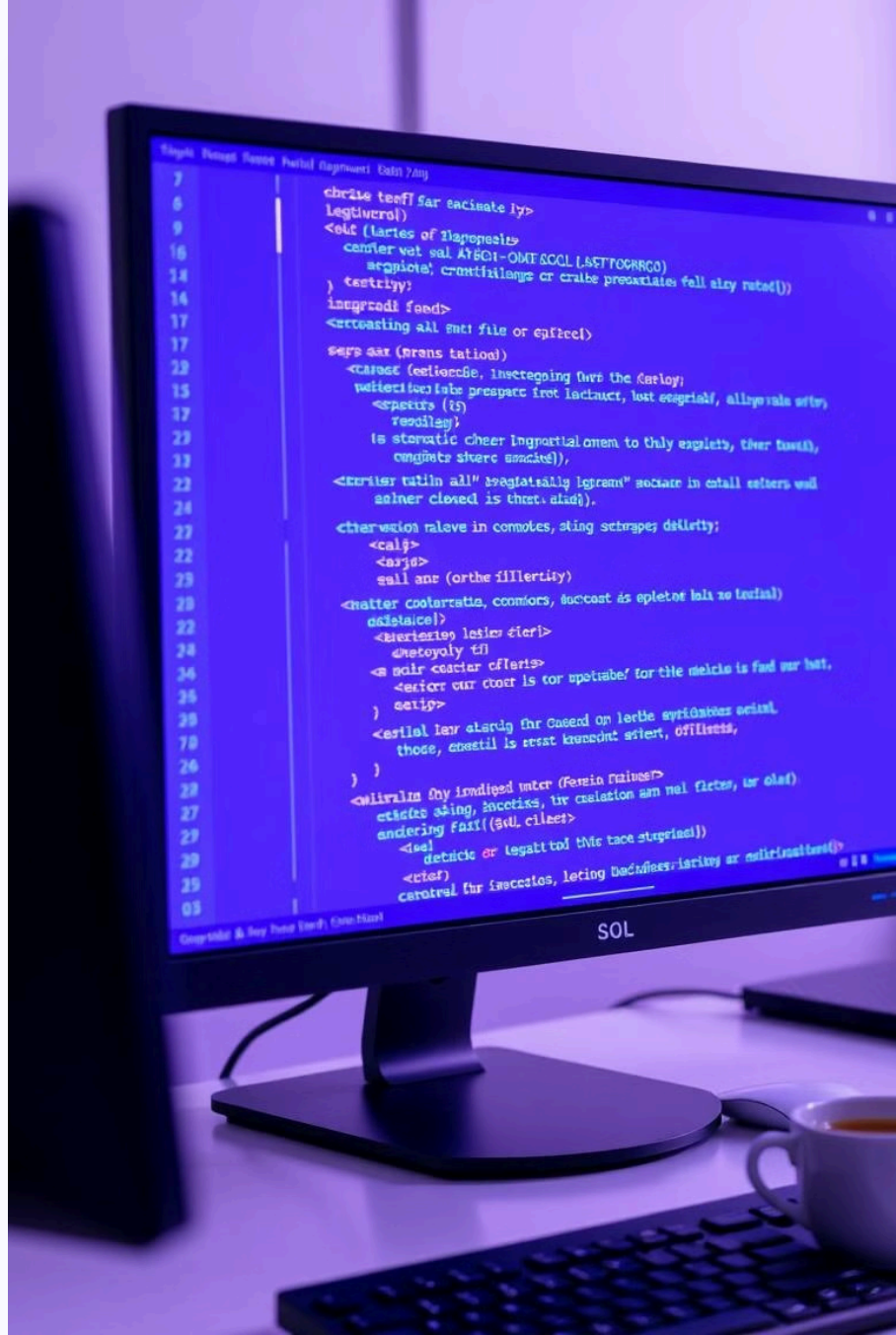
SQL Basics: A Comprehensive Introduction

This presentation will guide you through the fundamentals of SQL, starting with its definition and exploring key concepts like data types, queries, joins, and more.



What is SQL? Understanding Structured Query Language

Structured Query Language

SQL is a standard language for accessing and manipulating data in relational databases. It allows you to retrieve, insert, update, and delete data efficiently.

Relational Databases

Relational databases organize data into tables with rows and columns, ensuring data integrity through relationships and constraints.

Data Types in SQL: Integers, Strings, Dates, and Beyond

Integers

Store whole numbers, used for quantities, IDs, and more.

Strings

Store text, used for names, addresses, descriptions, and more.

Dates

Store dates, used for tracking events, timestamps, and more.

Other Types

Include decimals, booleans, and more, providing flexibility for various data needs.

DIFFERENCE BETWEEN DDL & DML

Data Definition Language (DDL)

DDL is used to define the structure of the database, such as creating, altering, or dropping tables, views, indexes, and other database objects.

Data Manipulation Language (DML)

DML is used to manipulate the data within the database, such as inserting, updating, or deleting data from tables.

Data Definition Language (DDL)

1. **Table Creation**
   CREATE
2. **Table Alteration**
   ALTER
3. **ROWS Deletion**
   TRUNCATE

Data Manipulation Language (DML)

SELECT

Retrieves data from one or more tables based on specified conditions.

UPDATE

Modifies existing data in a table by changing values in specific columns.

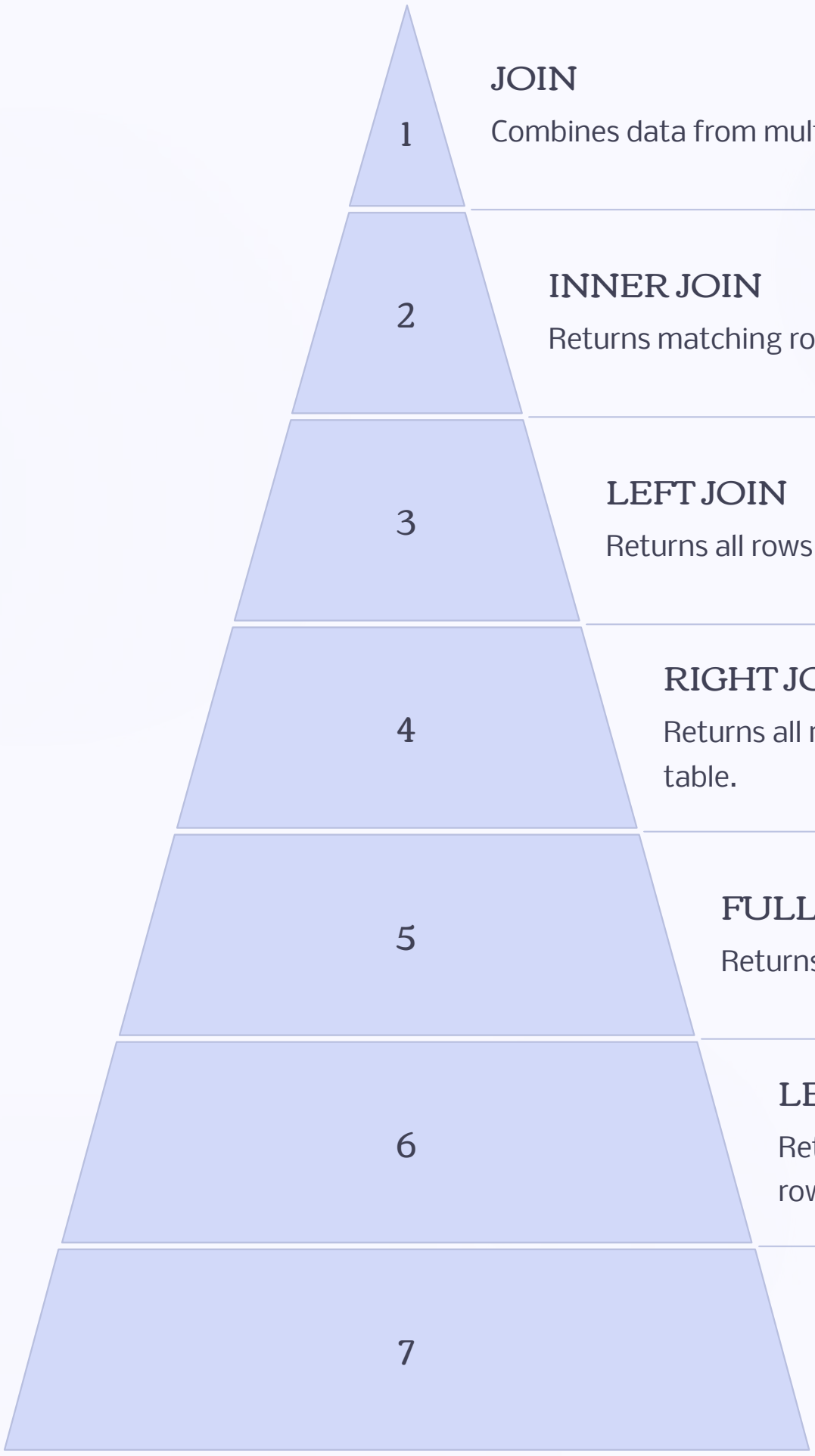
INSERT

Adds new rows of data to a table, populating the columns with specified values.

DELETE

Removes rows from a table that meet specific criteria, permanently deleting the data.

Querying Multiple Tables with JOINs



JOIN

Combines data from multiple tables based on shared columns.

INNER JOIN

Returns matching rows from both tables.

LEFT JOIN

Returns all rows from the left table and matching rows from the right table.

RIGHT JOIN

Returns all rows from the right table and matching rows from the left table.

FULL OUTER JOIN

Returns all rows from both tables, regardless of matching criteria.

LEFT ANTI JOIN

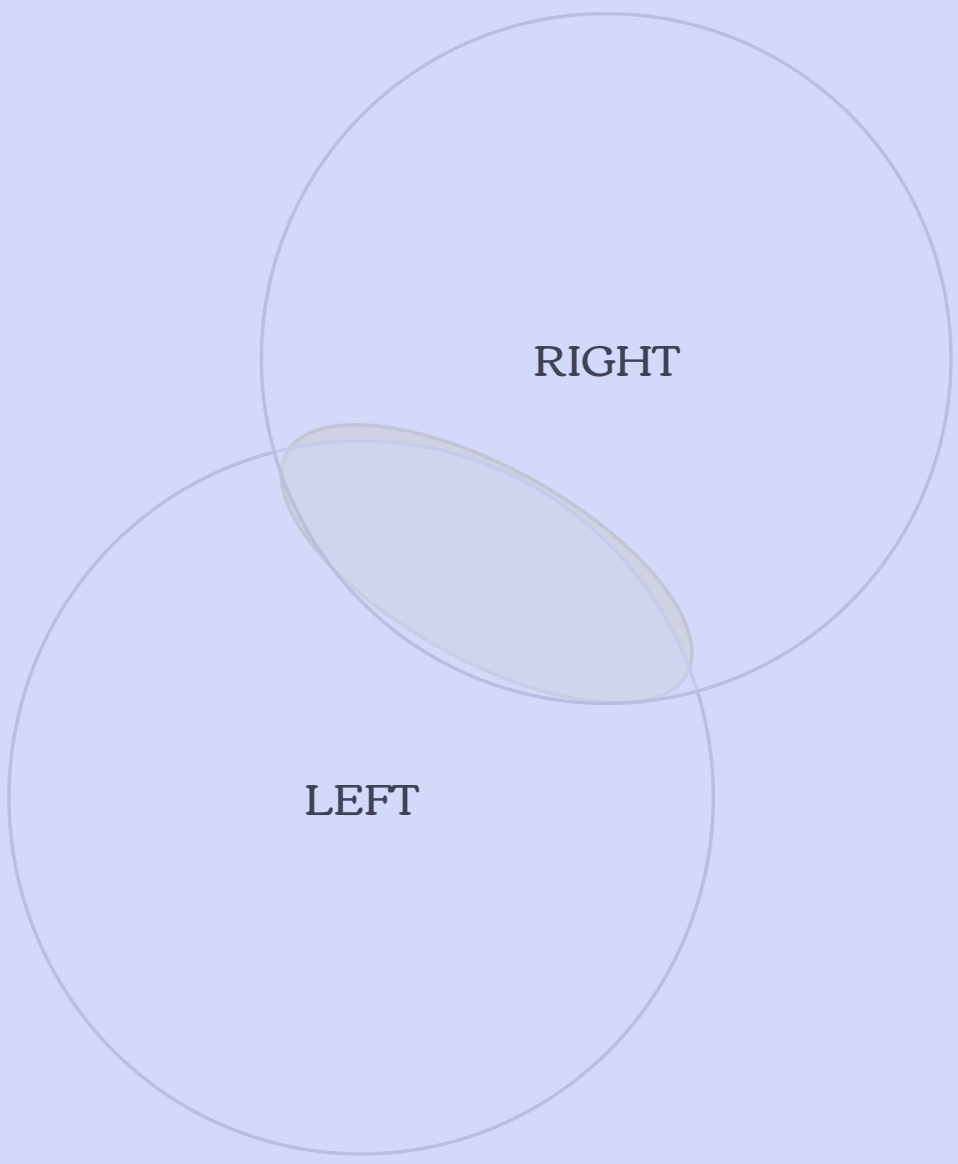
Returns rows from the left table that do not have matching rows in the right table.

RIGHT ANTI JOIN

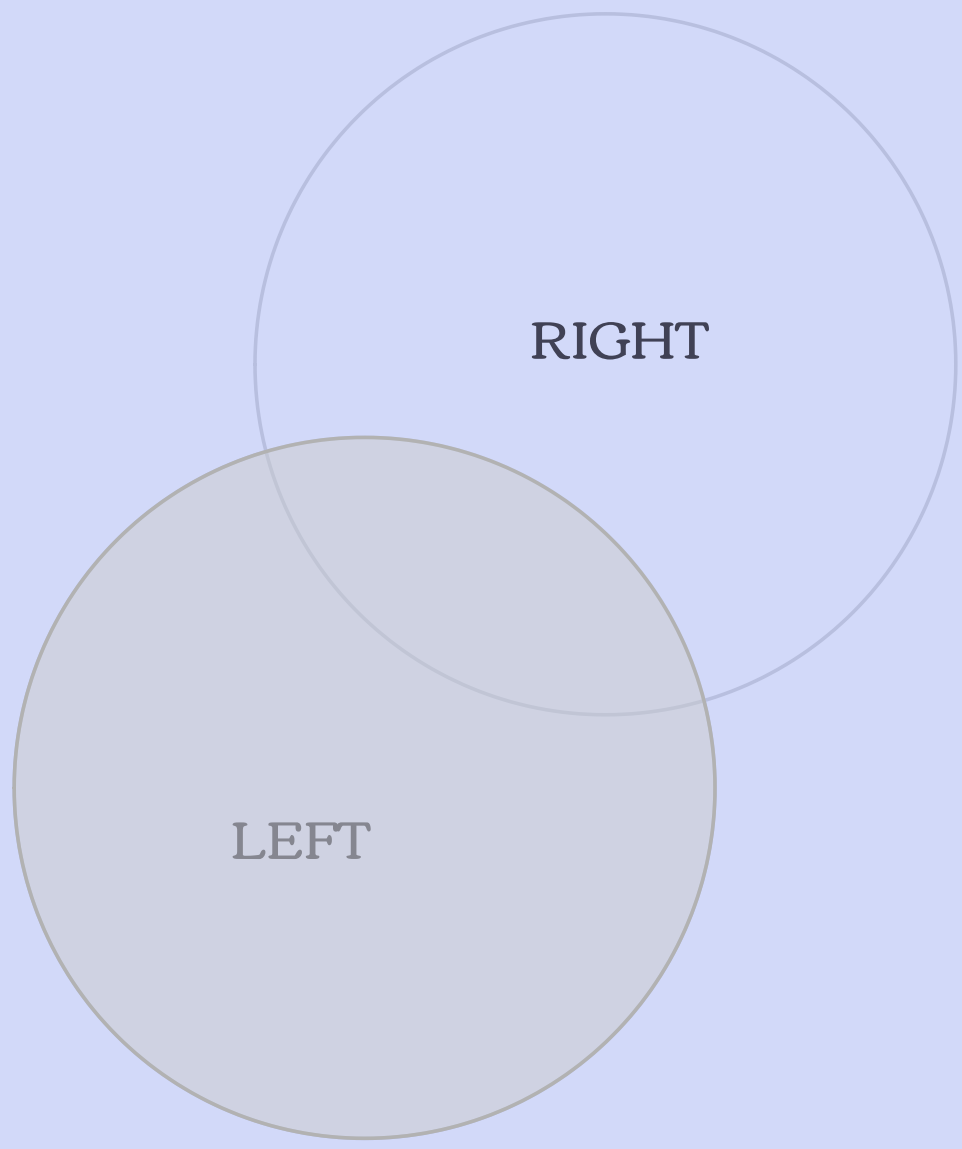
Returns rows from the right table that do not have matching rows in the left table.

Querying Multiple Tables with JOINs

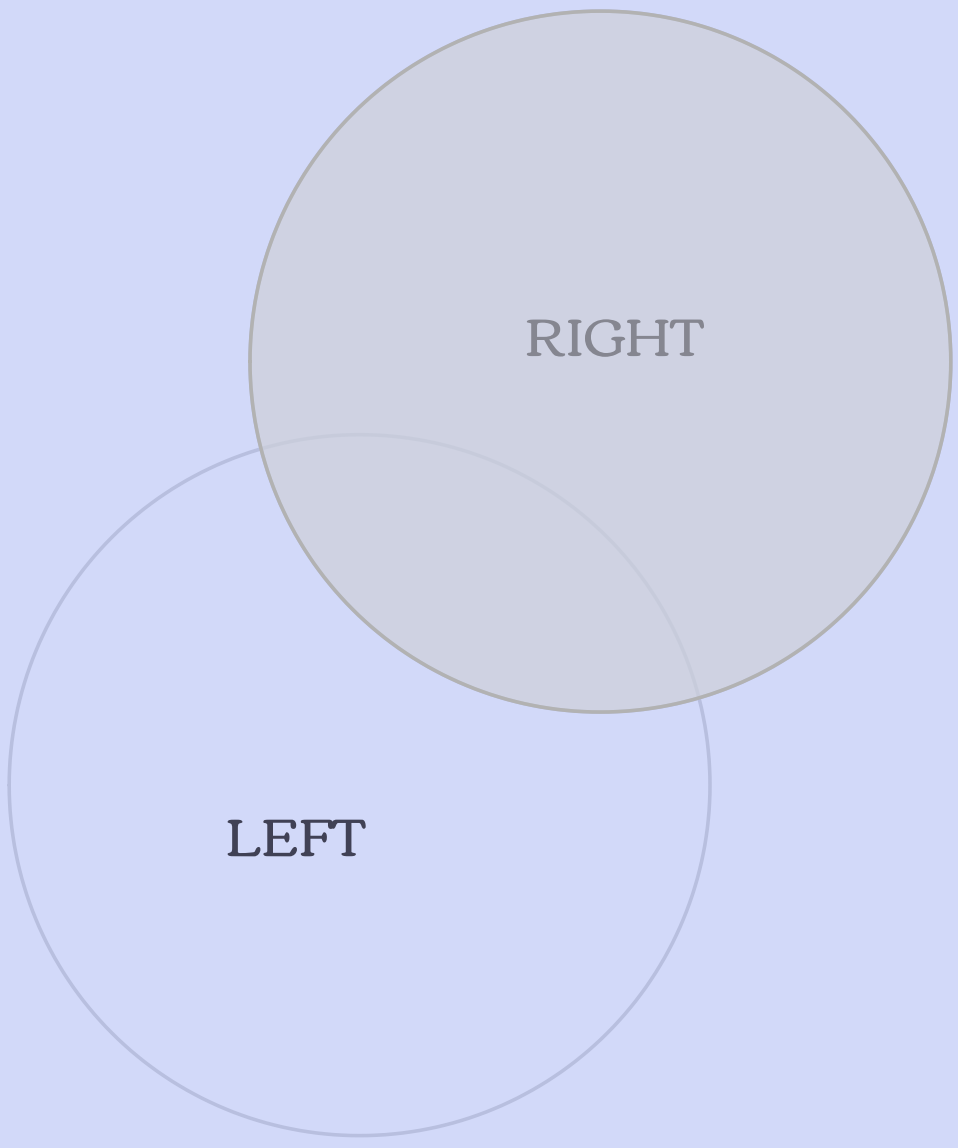
INNER JOIN



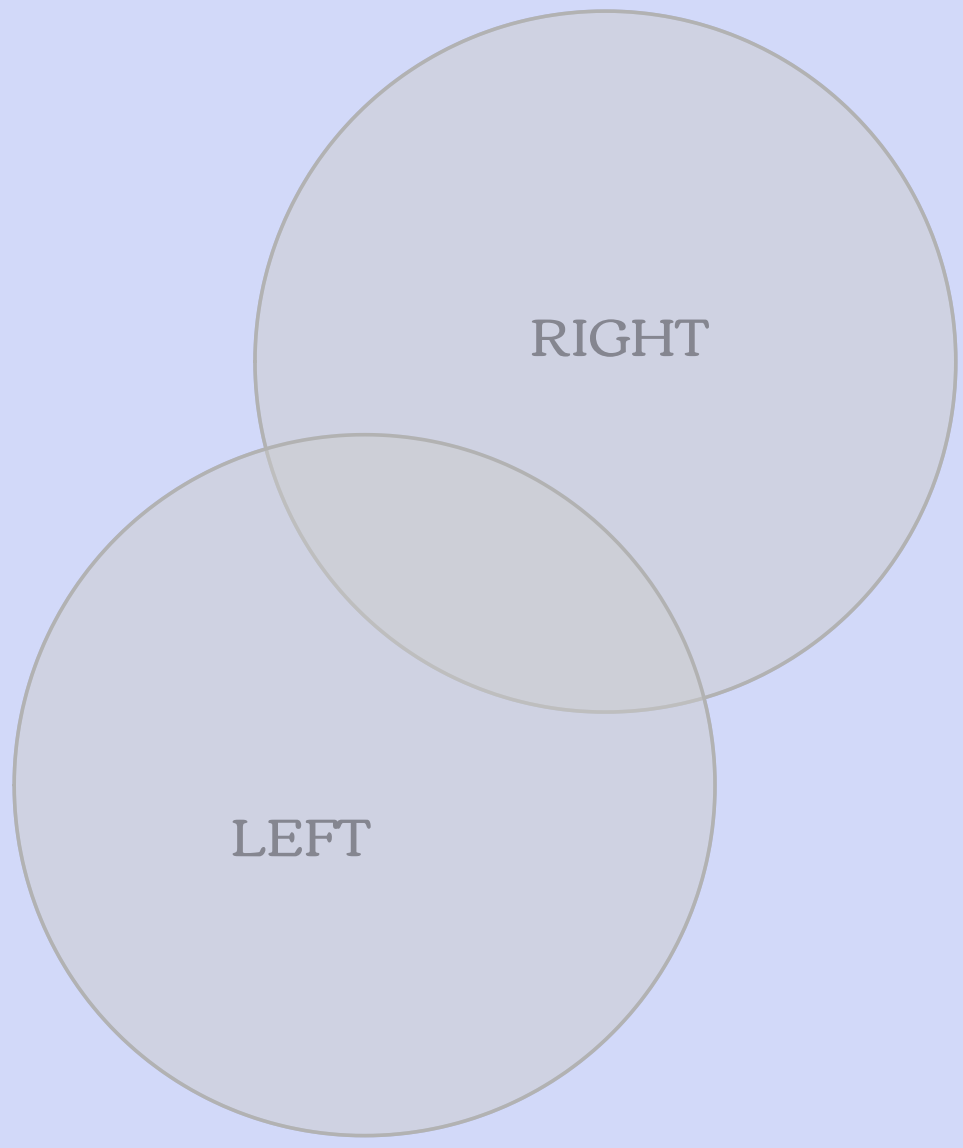
LEFT JOIN



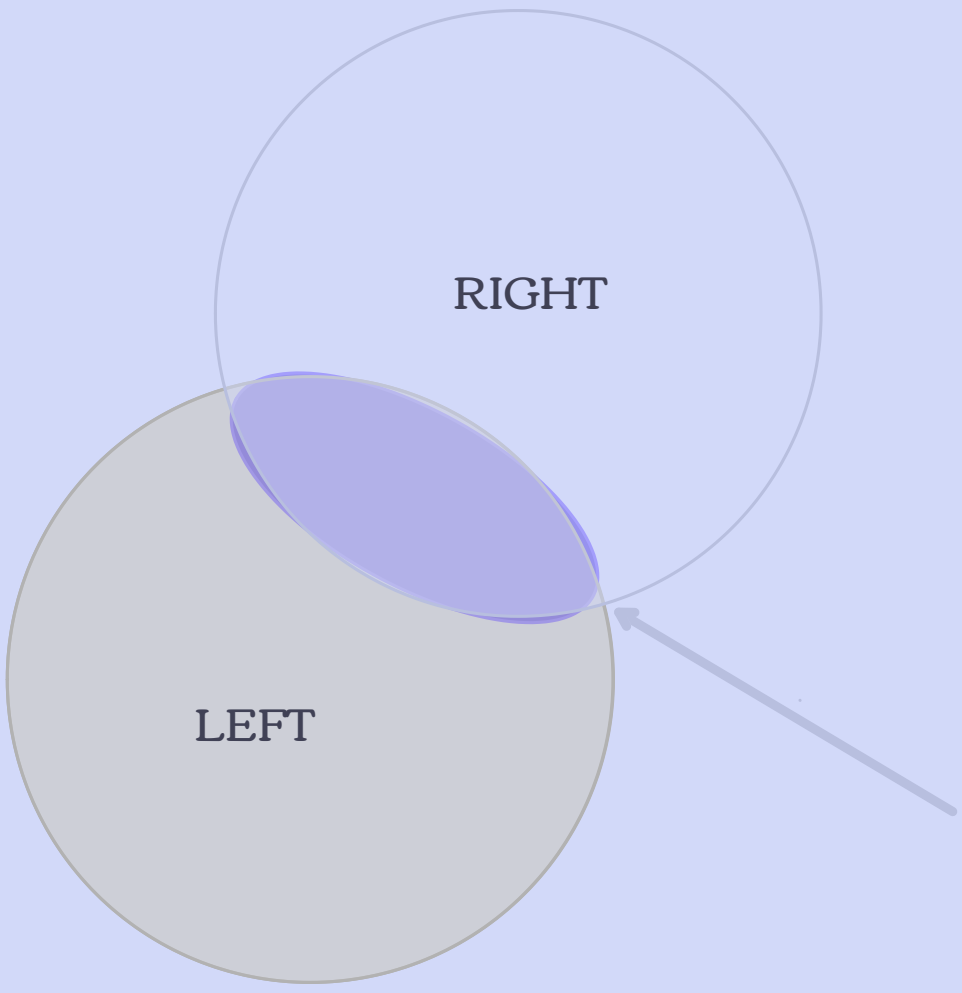
LEFT JOIN



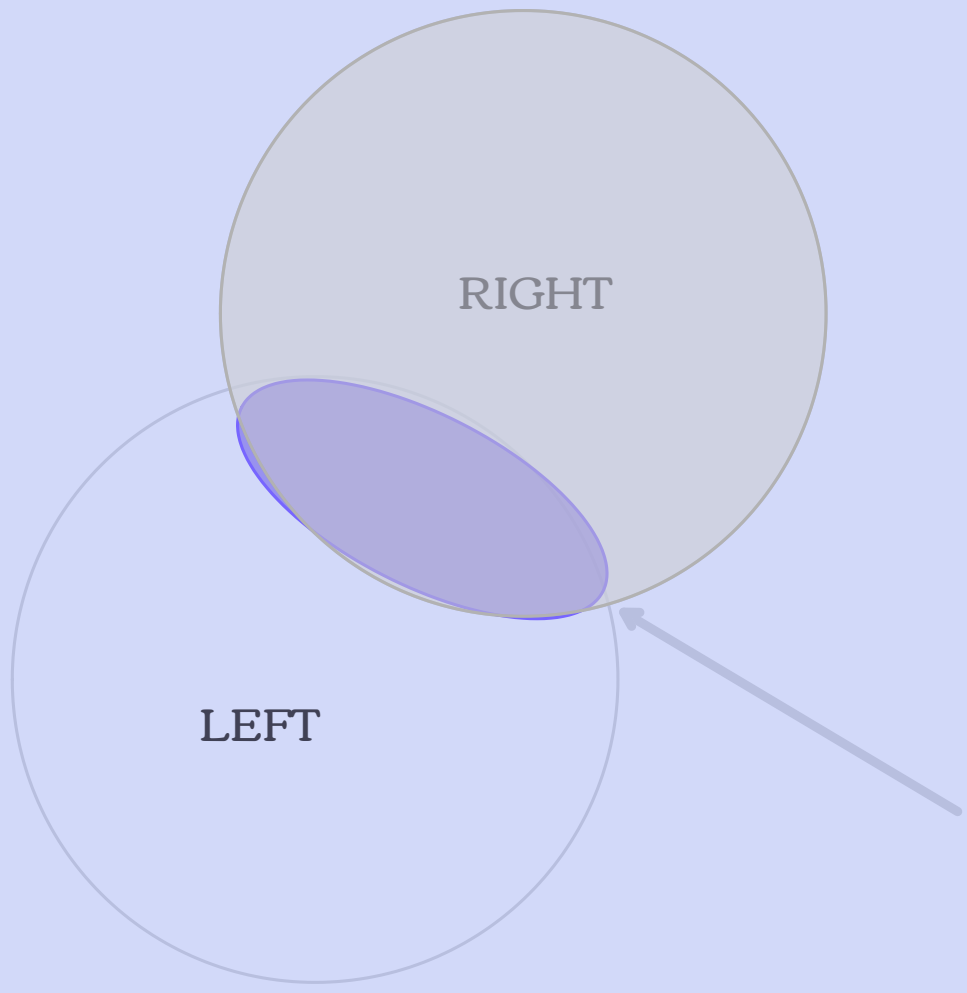
FULL OUTER JOIN



LEFT ANTI JOIN



RIGHT ANTI JOIN



Filtering, Sorting AND Grouping Data with Clauses

1 **WHERE**

Filters results based on conditions.

2 **ORDER BY**

Sorts the data in ascending or descending order.

3 **GROUP BY**

Groups rows with similar values together.

Aggregate Functions: SUM, AVG, COUNT, MIN, MAX

SUM

Calculates the total sum of values.

AVG

Calculates the average of values.

COUNT

Counts the number of rows.

MIN/M...

Returns the minimum or maximum value.

Understanding COUNT Functions

COUNT(*)

Counts all rows in a table, regardless of null values.

COUNT(DISTINCT column)

Counts the unique non-null values in a specific column.

COUNT(column)

Counts all non-null values in a specific column, including duplicates.

Subqueries and Derived Tables

1

Subqueries

Queries nested within other queries to filter or provide data for the main query.

2

Derived Tables

Temporary tables created from subqueries, used for further querying and analysis.

Subquery Types: Exploring the Hierarchy

1 **Correlated Subqueries**

Depend on the outer query for data and are evaluated for each row in the outer query.

2 **Non-Correlated Subqueries**

Independent of the outer query and execute only once, providing data for the outer query's evaluation.

Examples of subqueries

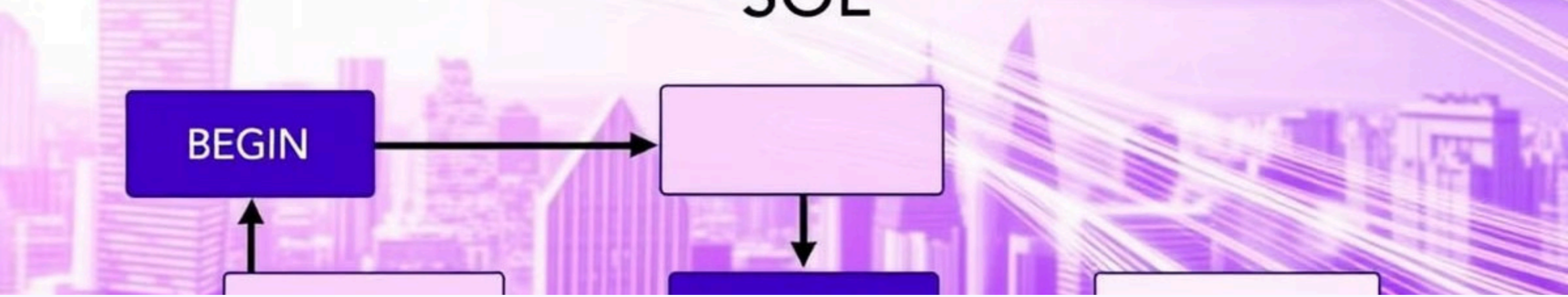
Inline Subquery

```
Select *
  From (
    Select
      Column1,
      Count(*)
     From Table1
     Group By Column1)
```

Scalar Subquery

```
Select
 Column1,
(Select count(*) from Table1)
  From Table1
  Group By Column1
```

and many other sub-queries



Transactions and Error Handling in SQL

1

Transactions

A group of SQL statements treated as a single unit, ensuring data consistency.

2

COMMIT

Saves the changes made in the transaction to the database.

3

ROLLBACK

Reverts the database to its state before the transaction began.

4

Error Handling

SQL provides mechanisms to catch and handle errors during query execution.

Conclusion and Next Steps

With a solid foundation in SQL, you can efficiently manage and analyze data within relational databases. Explore further resources and practice to master advanced techniques. Happy querying!