Analiza

Intelligence - Cyber - Data

SQL - Joins & Union

Kodkod 4

Overview

- Understand what a `JOIN` is and why it’s needed.
- Use `INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN`.
- Write queries that combine data from related tables.
- Understand the difference between `JOIN` and `UNION`.
- Practice writing SQL queries using the school database.

What is a JOIN?

- A `JOIN` combines data from two or more tables based on a related column between them - usually a **foreign key** in one table referencing a **primary key** in another.
- Instead of viewing tables separately, we can see how the data connects.
- Example: By joining the id column from the classes table with the class_id column from the students table, we can display the class name in the student results.

id	full_name	class_id
1	Jennifer Thomas	2
2	Linda Martinez	2
6	Laura Martinez	4
20	Robert Martinez	3

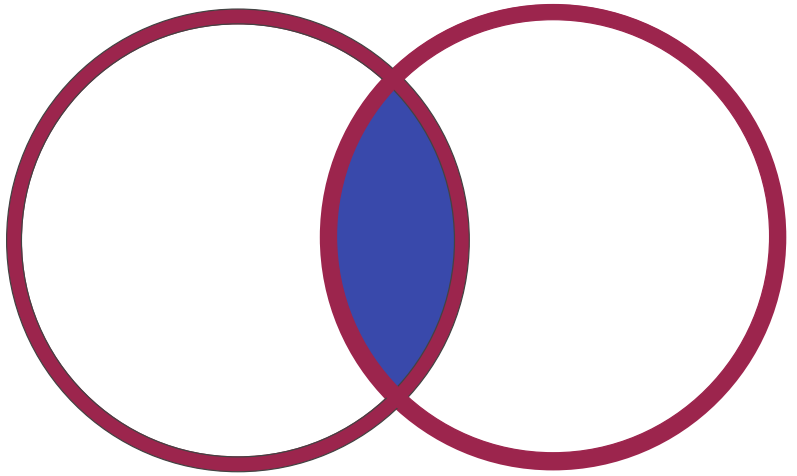
id	full_name	class_id	class
1	Jennifer Thomas	2	B
2	Linda Martinez	2	B
6	Laura Martinez	4	D
20	Robert Martinez	3	C

id	class
1	A
2	B
3	C
4	D

INNER JOIN

Matching Data from Two Tables

- `INNER JOIN` returns only the rows that have matching **values in both tables**.
- If a record exists in one table but not in the other, it will not appear in the result.

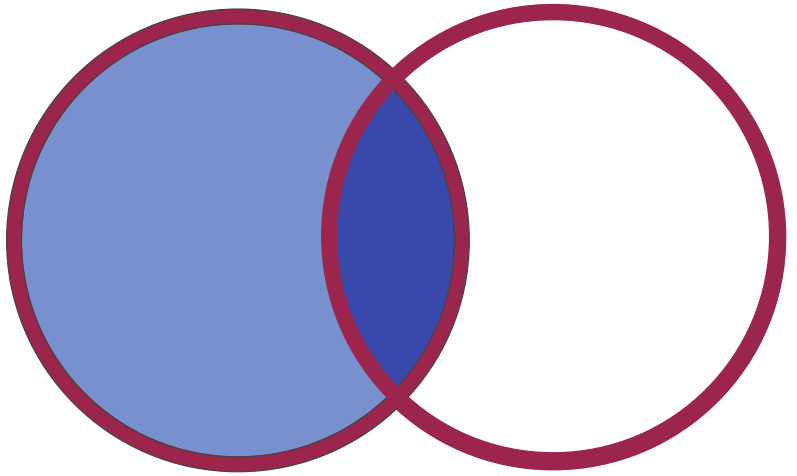


```sql
SELECT students.full_name, classes.class
FROM students
INNER JOIN classes
ON students.class_id = classes.id;
```

LEFT JOIN

All Records from the Left Table

- `LEFT JOIN` returns all rows from the left table, and the matching rows from the right table.
- If there's no match, the result will still include the left row, but with NULL values for the right table's columns.

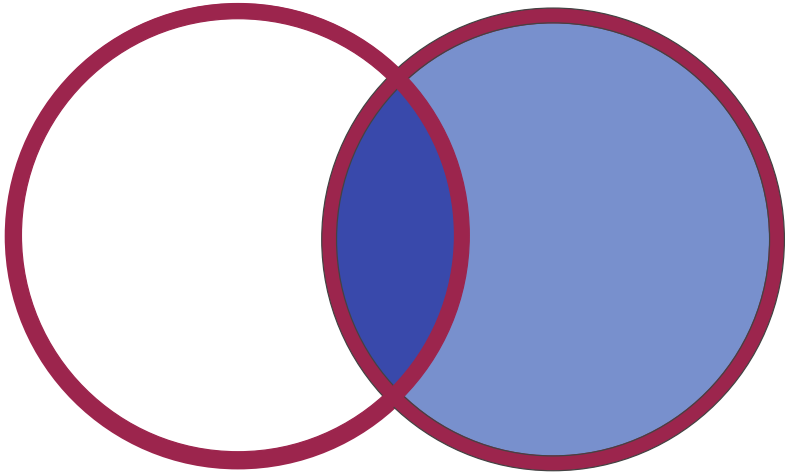


```sql
SELECT students.full_name, classes.class
FROM students
LEFT JOIN classes
ON students.class_id = classes.id;
```

RIGHT JOIN

All Records from the Right Table

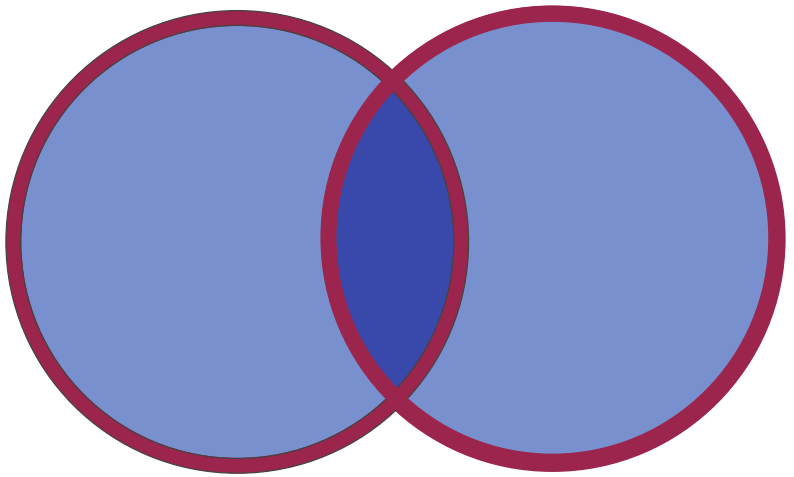
- `RIGHT JOIN` returns all rows from the right table, and the matching rows from the left table.
- If there's no match, the result will still include the right row, but with NULL values for the left table's columns.



```sql
SELECT students.full_name, classes.class
FROM students
RIGHT JOIN classes
ON students.class_id = classes.id;
```

FULL JOIN

All Records from Both Tables



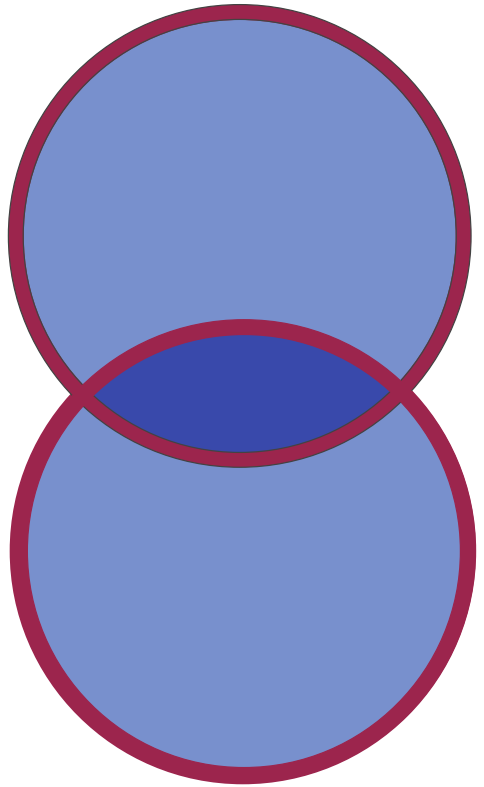
- `FULL JOIN` returns all rows from both tables, whether or not there's a match.
- If a record exists only in one table, the result will still include it - with NULLs for the missing side.
- In most SQL systems (like PostgreSQL, SQL Server, Oracle) you can use FULL JOIN directly.

```sql
SELECT students.full_name, classes.class
FROM students
LEFT JOIN classes
ON students.class_id = classes.id

UNION

SELECT students.full_name, classes.class
FROM students
RIGHT JOIN classes
ON students.class_id = classes.id;
```

UNION

Combine Results from Multiple Queries

- `UNION` is used to combine the results of two or more `SELECT` statements.
- Unlike `JOIN`, which merges tables side by side, `UNION` stacks the results one on top of another.
- `UNION` removes duplicates automatically.
- Each query inside the `UNION` must have:
  - The same number of columns.
  - The same or compatible data types.
  - Columns in the same order.

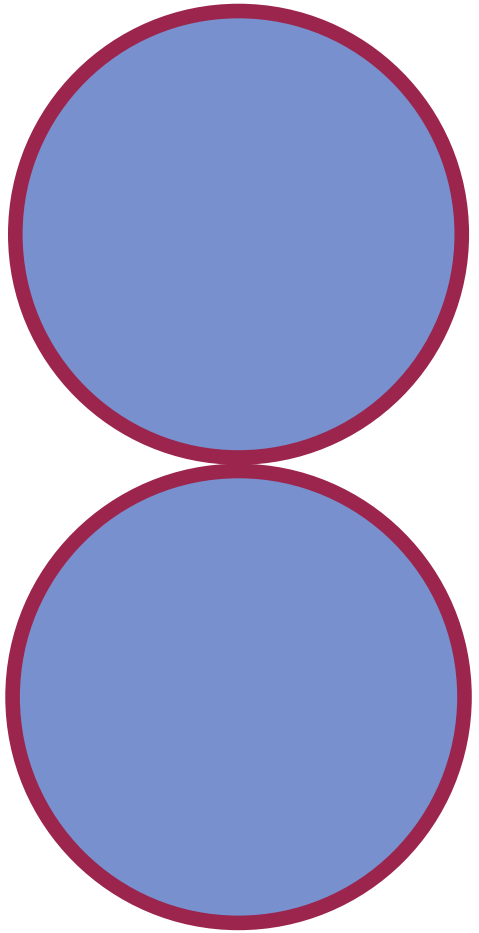


```sql
SELECT id, full_name, class_id FROM students
UNION
SELECT id, full_name, class_id FROM students_new;
```

UNION ALL

Combine Results - Include Duplicate

- `UNION ALL` works just like `UNION`, but it does not remove duplicate rows.
- It simply combines all results from both queries - even if some rows are identical.

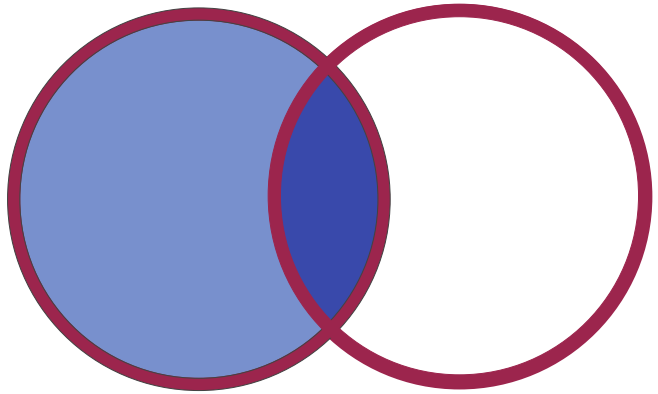


```sql
SELECT column1, column2 FROM table1
UNION ALL
SELECT column1, column2 FROM table2;
```

Type of JOINs & UNIONs?

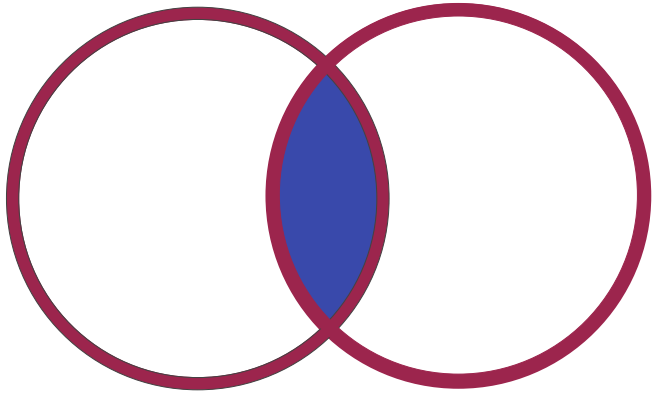
LEFT JOIN

Returns all rows from the left table, and matches from the right.



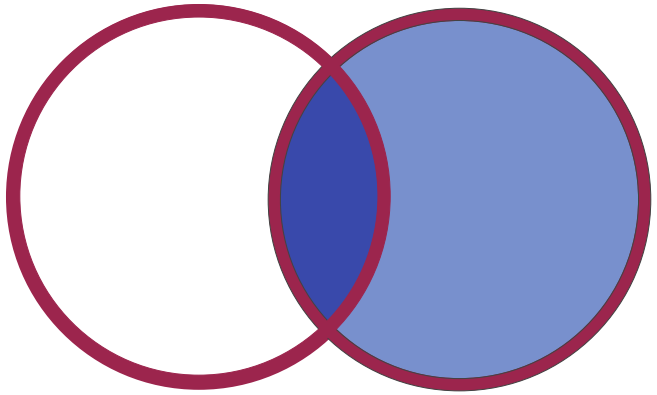
INNER JOIN

Returns rows that exist in both tables.



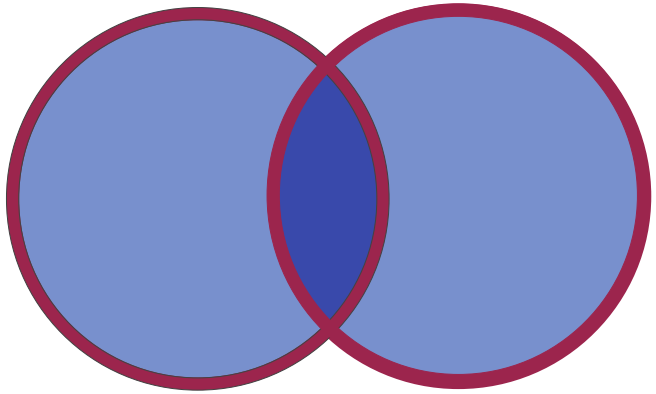
RIGHT JOIN

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



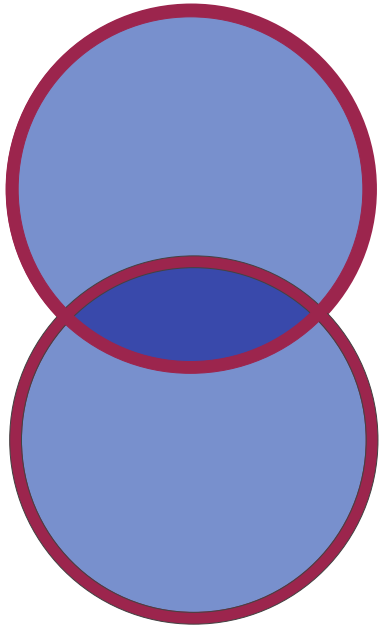
FULL JOIN

Returns all rows from both matched and unmatched rows.



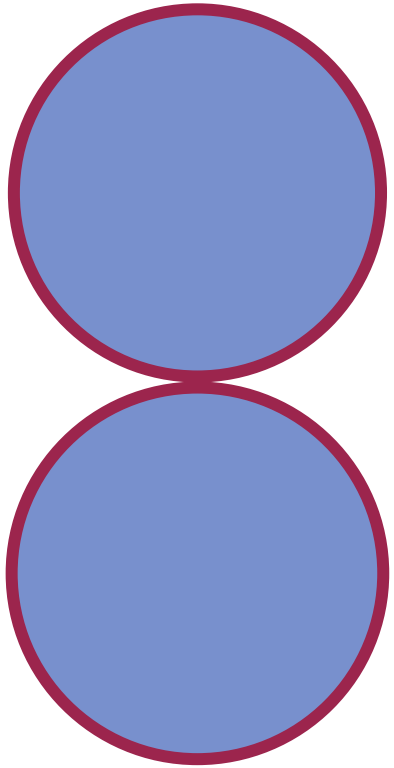
UNION

Combines query results vertically and remove duplicates.



UNION ALL

Combines query results including duplicates.



Class Exercises

1. Get a list of all students and their class names. If a student doesn’t belong to any class, display NULL instead.
2. Get a list of all students with valid classes only. Show each student’s name and class.
3. Combine all students from both tables - students and students_new - into one unified list, without duplicates.

Summary

- Combine data from multiple tables using different types of `JOIN`s to enrich the result with more columns.
- Combine query results vertically with `UNION`.
- Understand that:
  - `JOIN` adds columns (horizontal merge).
  - `UNION` adds rows (vertical merge).