



Introduction to Join Queries in SQL

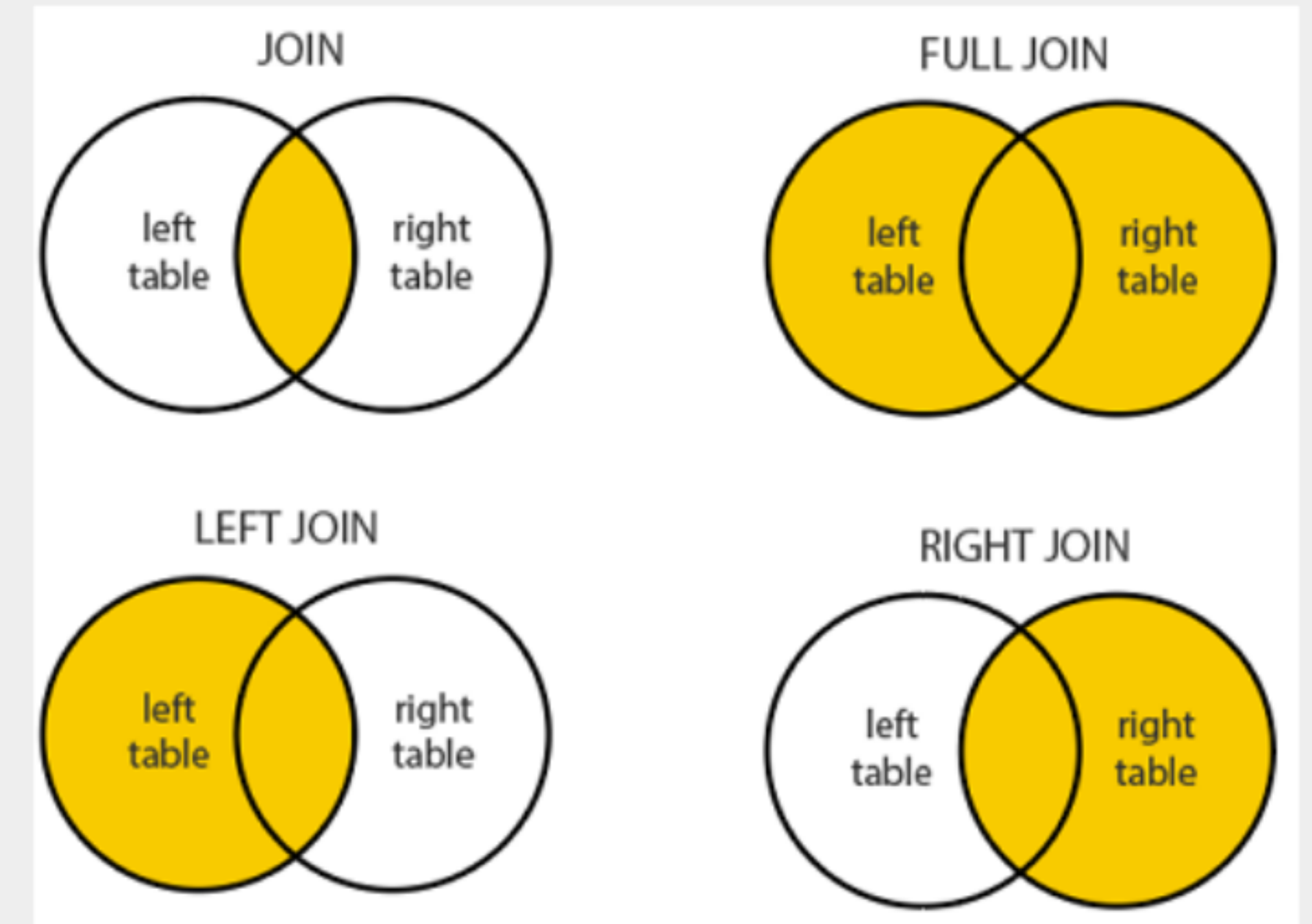
By

Sree Varshan .M

2021242019

What are Join Queries in SQL?

Join queries allow us to combine data from multiple tables based on a common column, providing a powerful tool for data analysis and manipulation.



Types of Join Queries

To describe the different types of Join Queries in SQL

- **Inner Join**

Returns only the matching rows from both tables.

- **Self Join**

joins the table to itself and allows comparing rows within the same table.

- **Right Outer Join**

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

- **Left Outer Join**

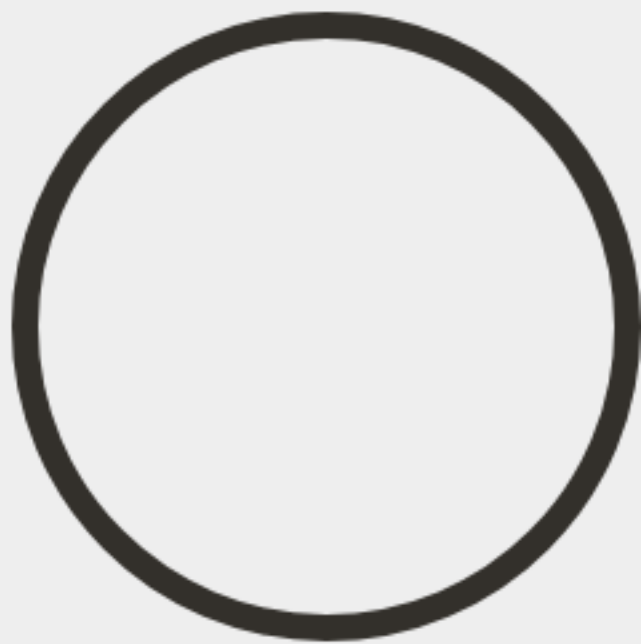
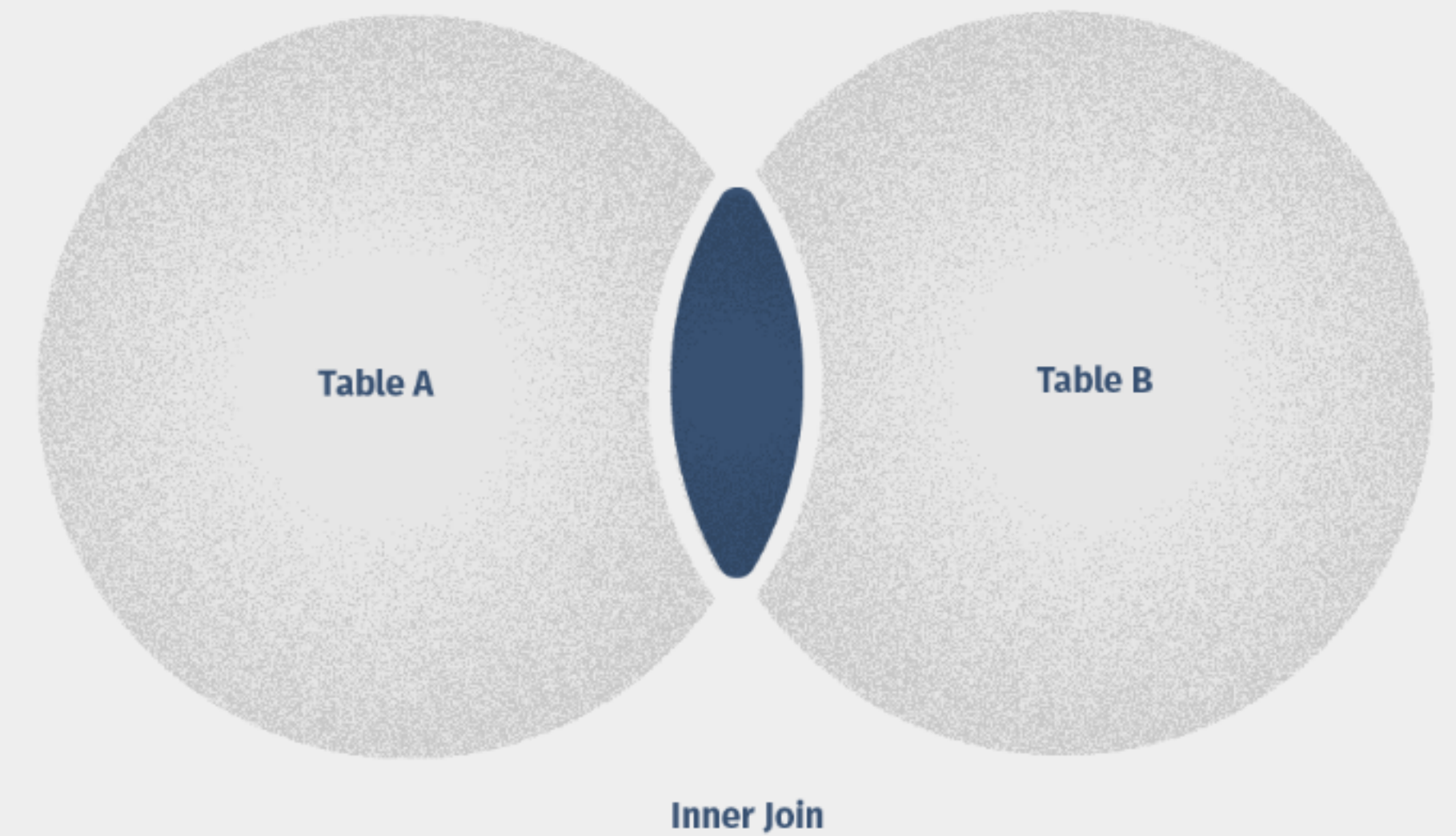
Returns all the rows from the left table and matching rows from the right table.

- **Cross Join**

creates a result table containing paired combination of each row of the first table with each row of the second table.

Inner Join

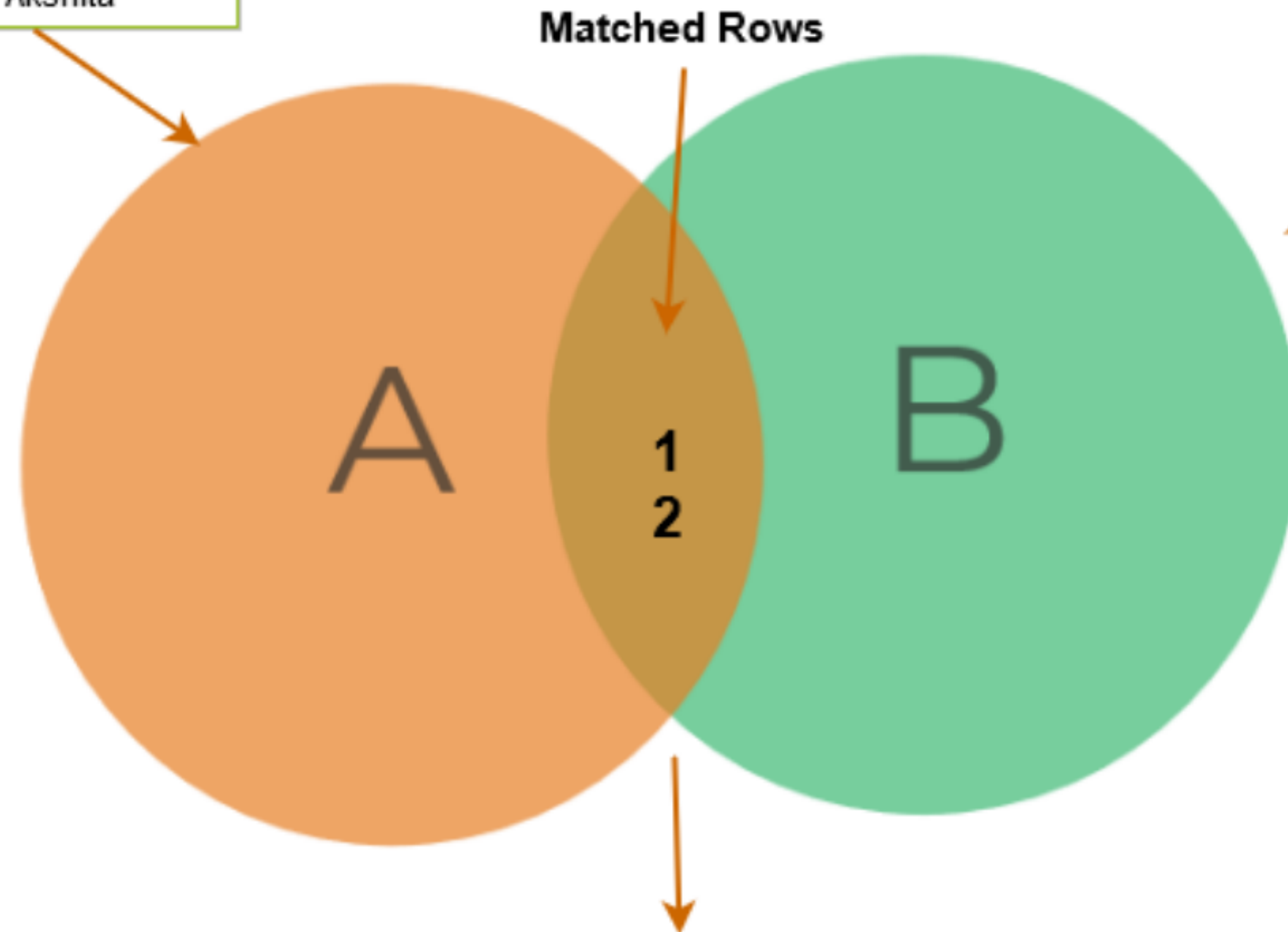
INNER JOIN statement returns only those records or rows that have matching values and is used to retrieve data that appears in both tables.



EmpID	EmpName
1	Rajendra
2	Kusum
3	Akshita

Inner Join

ID	City
1	Jaipur
2	Delhi
7	Raipur
8	Bangalore

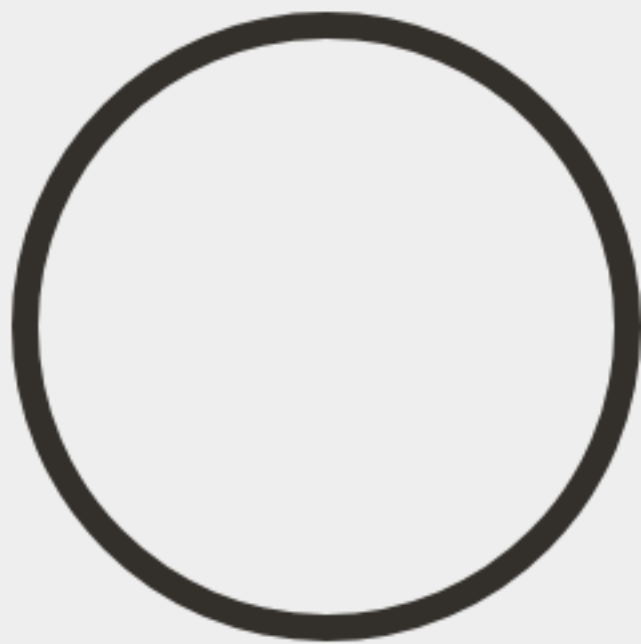
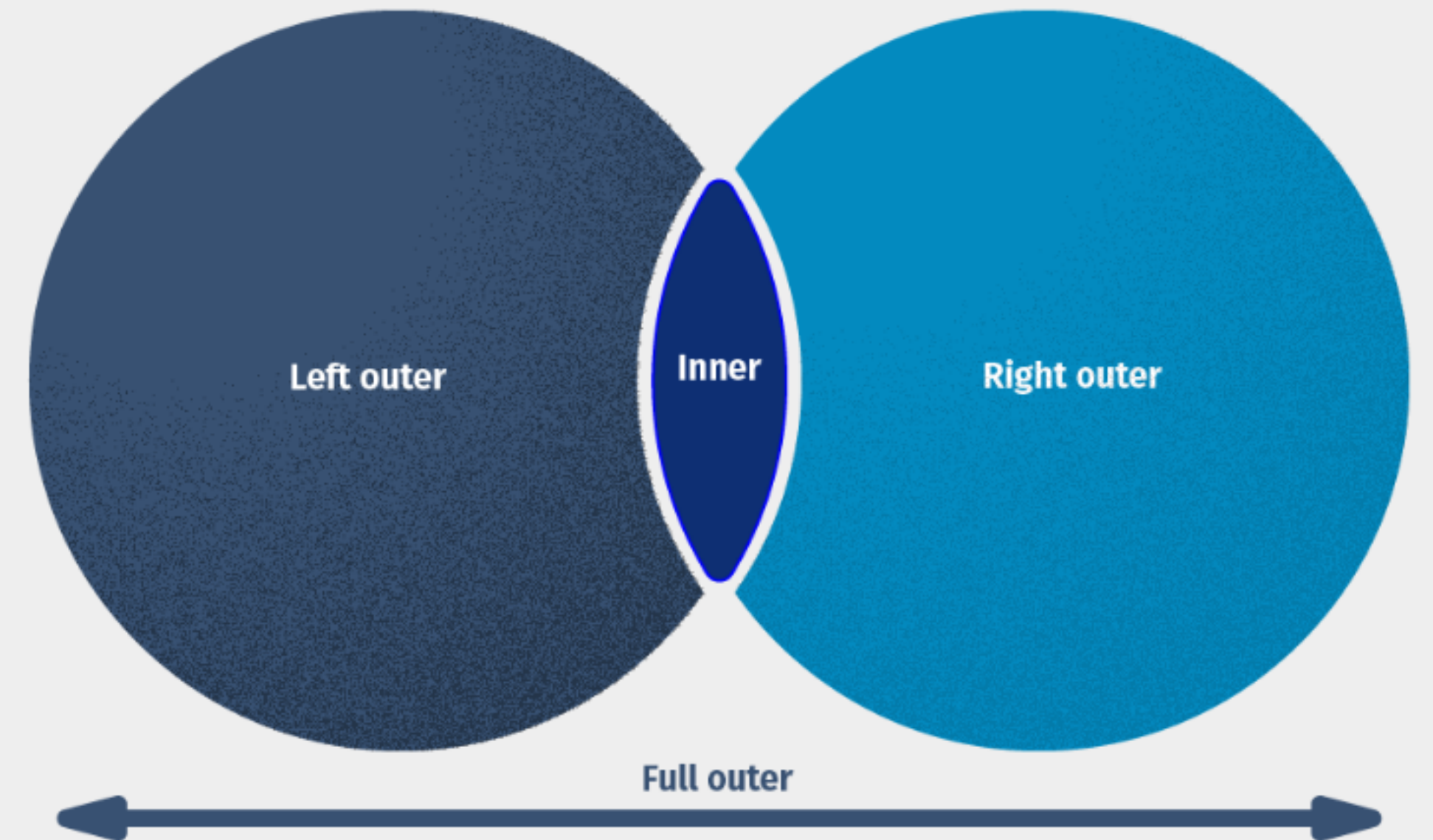


Select EmpName, City from employees e
Inner Join address a
on e.empid=a.id

EmpName	City
Rajendra	Jaipur
Kusum	Delhi

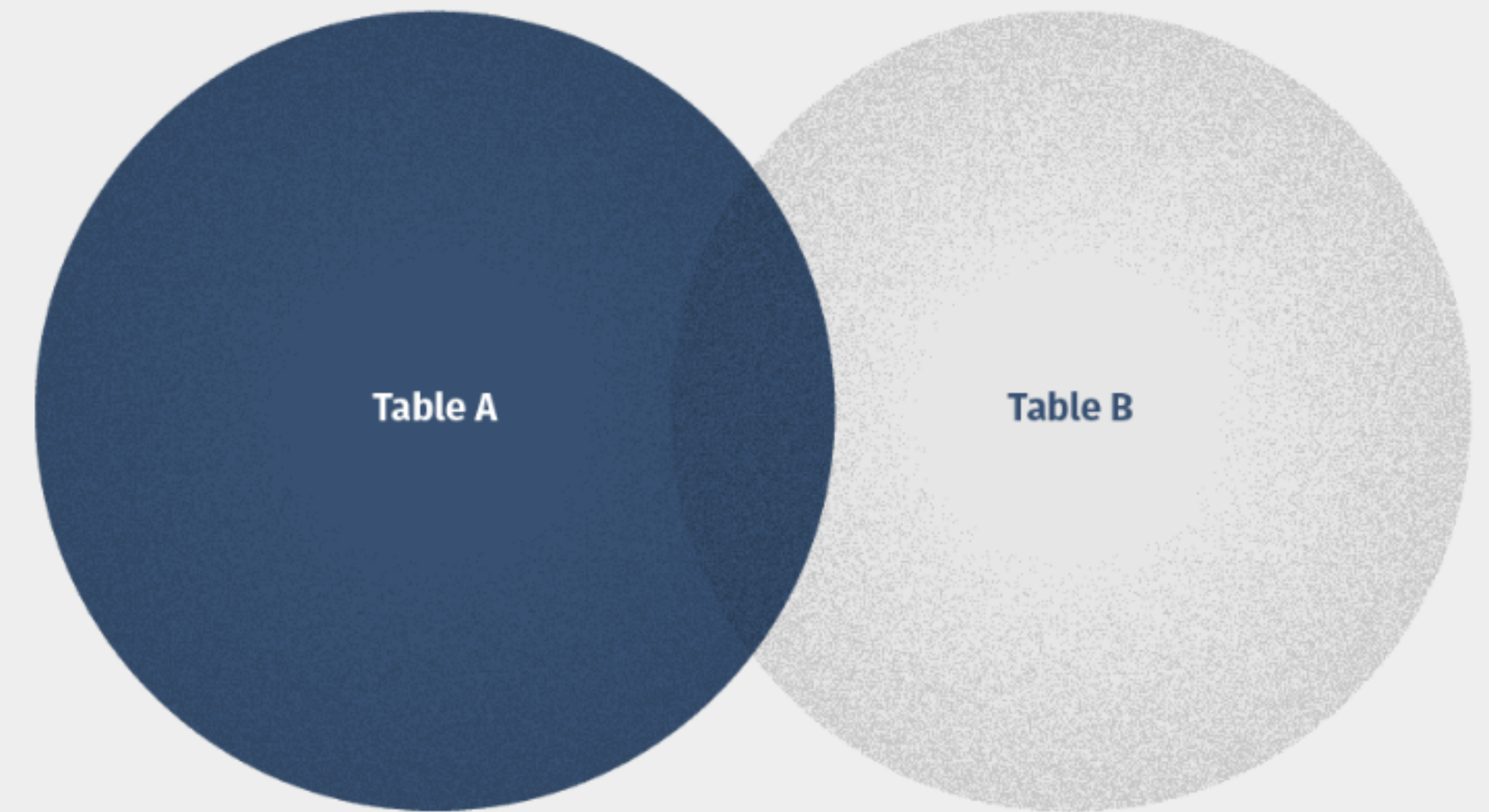
Outer Join

When applying an SQL INNER JOIN, the output returns only matching rows from the stated tables. In contrast, if you use an SQL OUTER JOIN, it will retrieve not only the matching rows but also the unmatched rows as well.

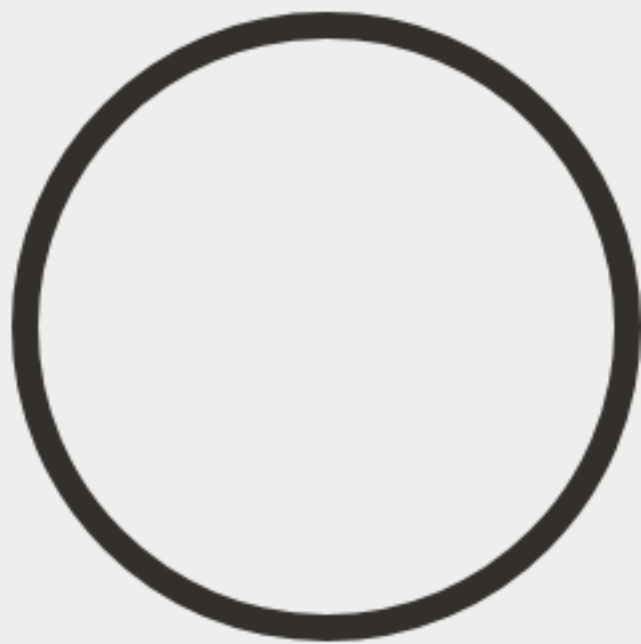


Left Outer Join

The LEFT OUTER JOIN gives the output of the matching rows between both tables. In case, no records match from the left table, it shows those records with null values.



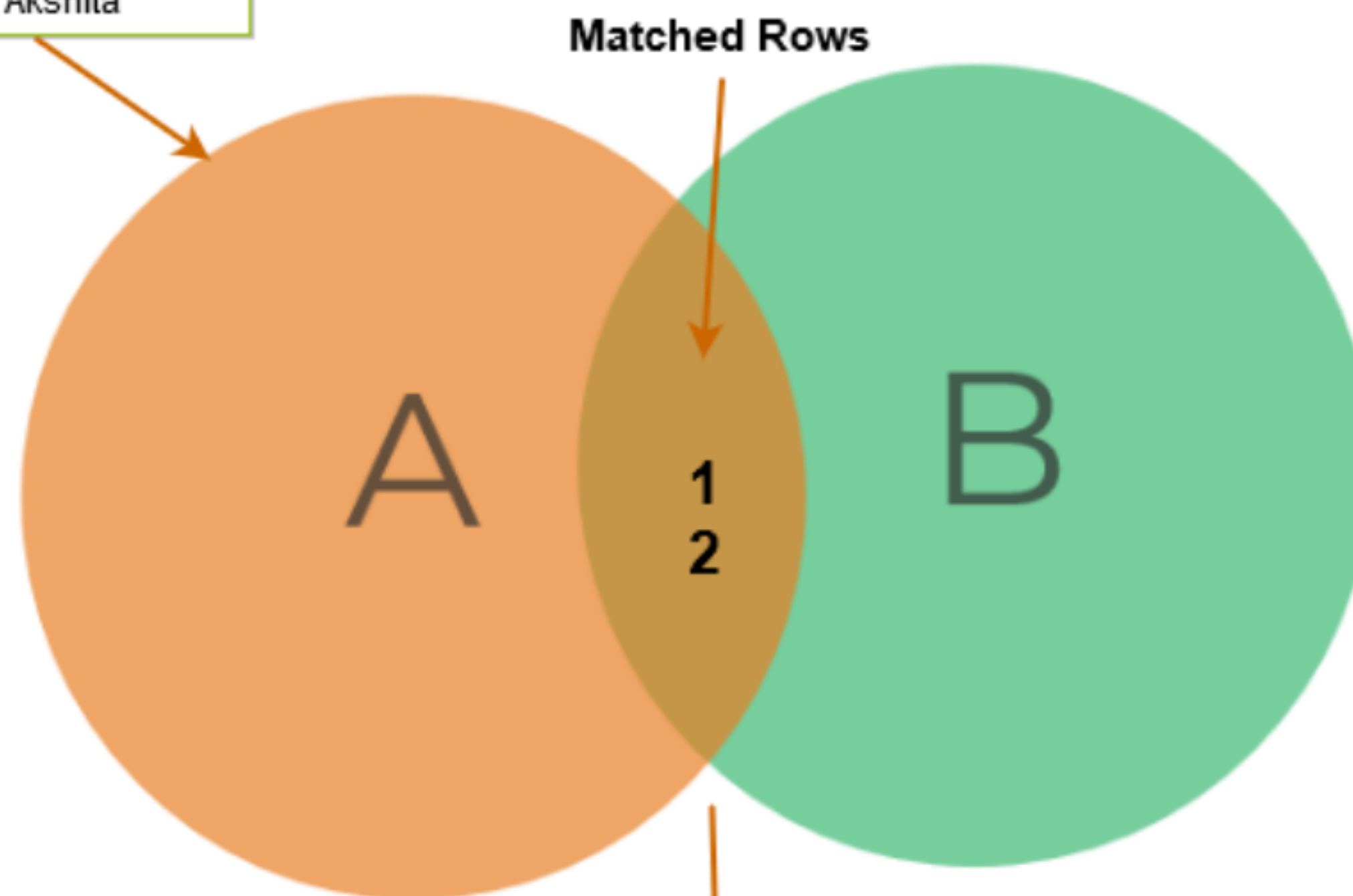
Left Outer Join



EmpID	EmpName
1	Rajendra
2	Kusum
3	Akshita

Left Outer Join

ID	City
1	Jaipur
2	Delhi
7	Raipur
8	Bangalore

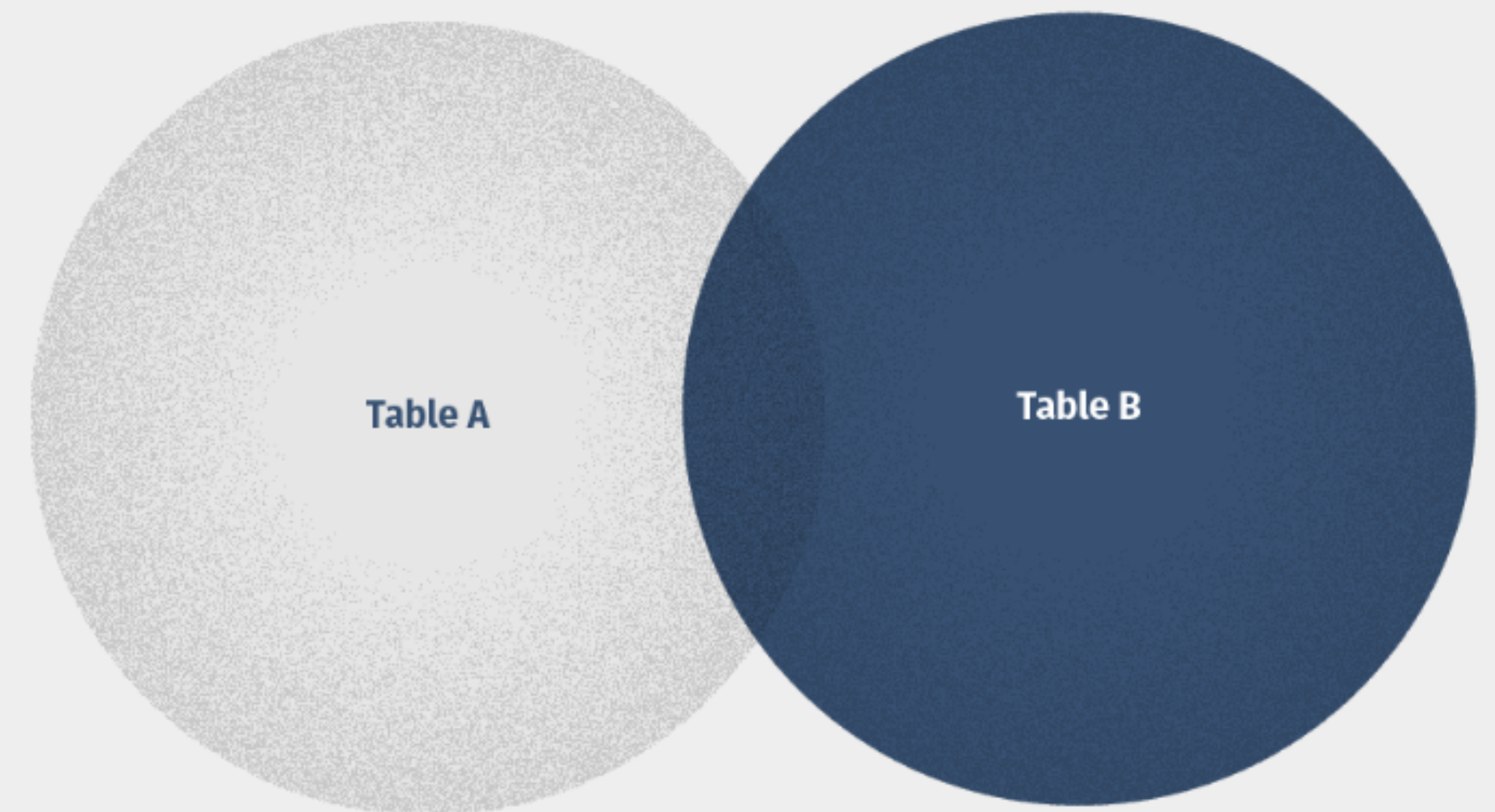


Select EmpName, City from employees e
Left Outer Join address a
on e.empid=a.id

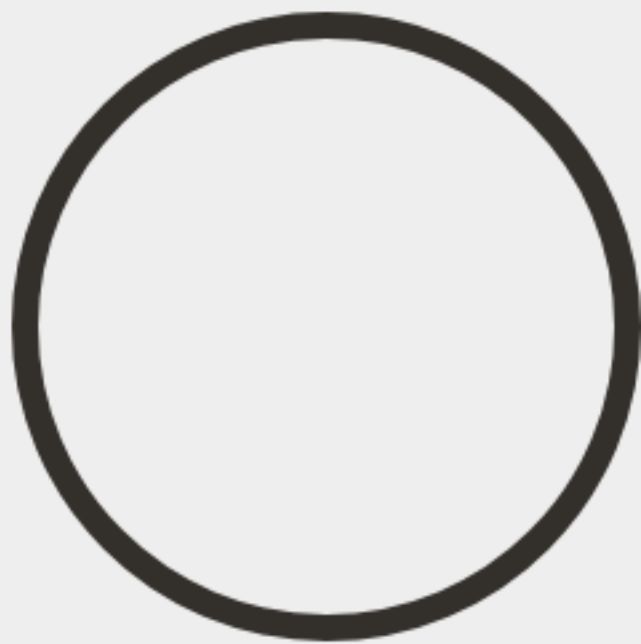
EmpName	City
Rajendra	Jaipur
Kusum	Delhi
Akshita	NULL

Right Outer Join

The RIGHT OUTER JOIN works by the same principle as the LEFT OUTER JOIN. The RIGHT OUTER JOIN selects data from the right table (Table B) and matches this data with the rows from the left table (Table A). The RIGHT JOIN returns a result set that includes all rows in the right table, whether or not they have matching rows from the left table. In case, a row in the right table does not have any matching rows in the left table, the column of the left table in the result set will have nulls.



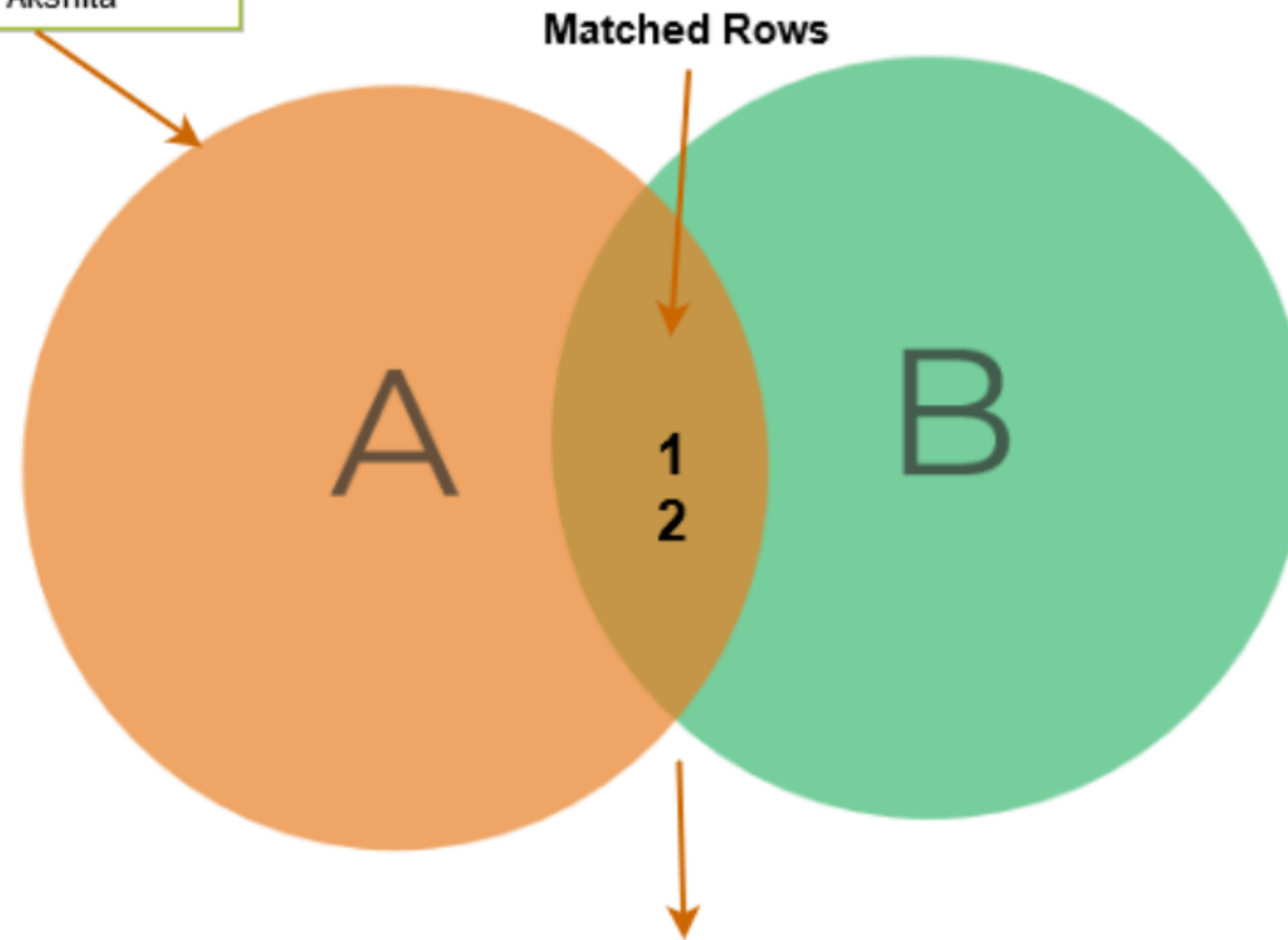
Right Outer Join



EmplID	EmpName
1	Rajendra
2	Kusum
3	Akshita

Right Outer Join

ID	City
1	Jaipur
2	Delhi
7	Raipur
8	Bangalore



Select EmpName, City from employees e
Right Outer Join address a
on e.empid=a.id

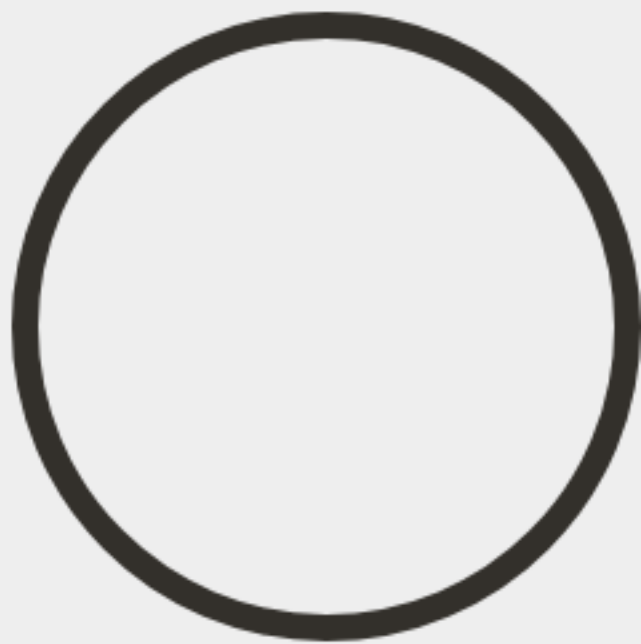
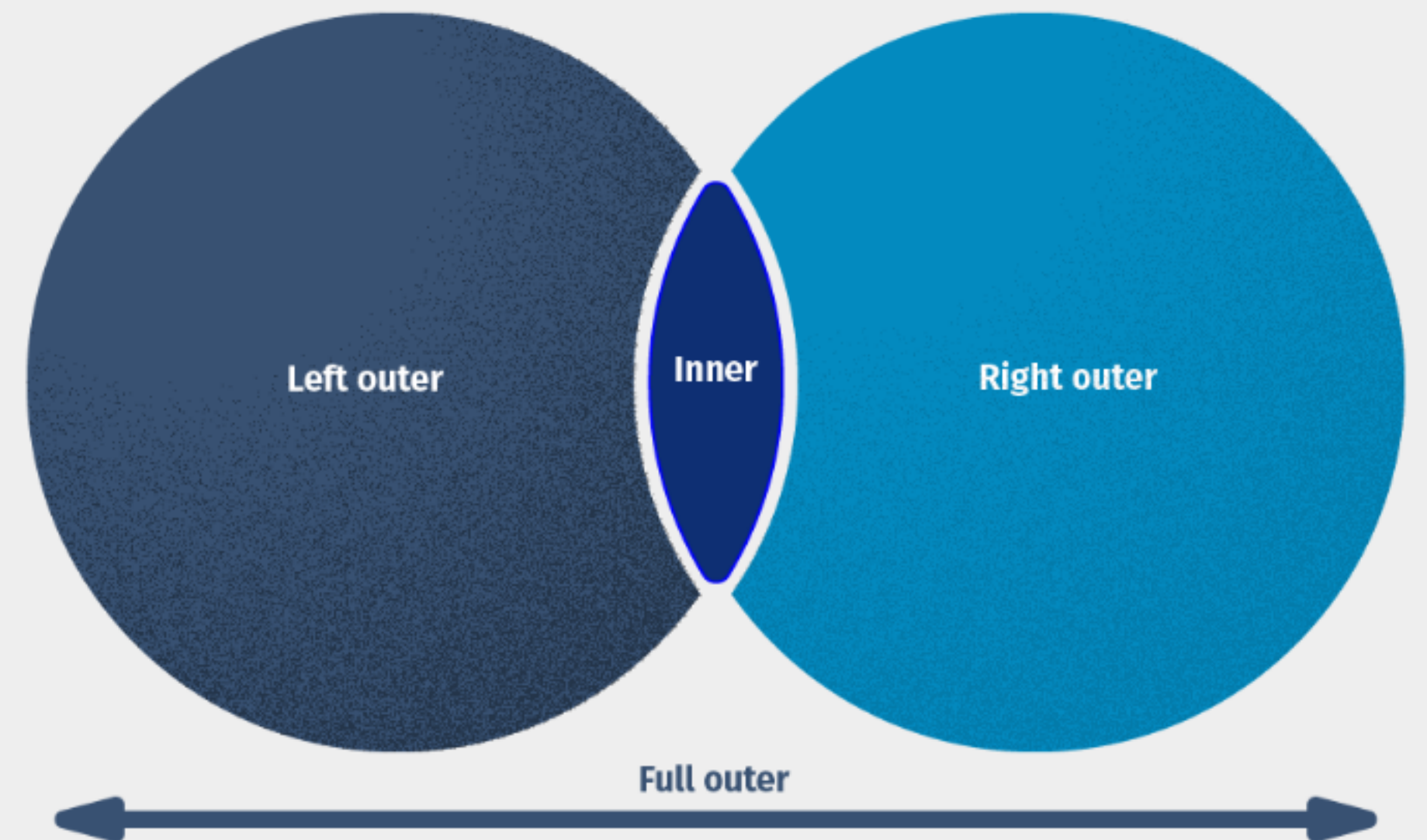
EmpName	City
Rajendra	Jaipur
Kusum	Delhi
NULL	Raipur
NULL	Bangalore

Full Outer Join

A full outer join returns the following rows in the output:
Matching rows between two tables.

Unmatched rows similar to left outer join: NULL values
for unmatched rows from the right table.

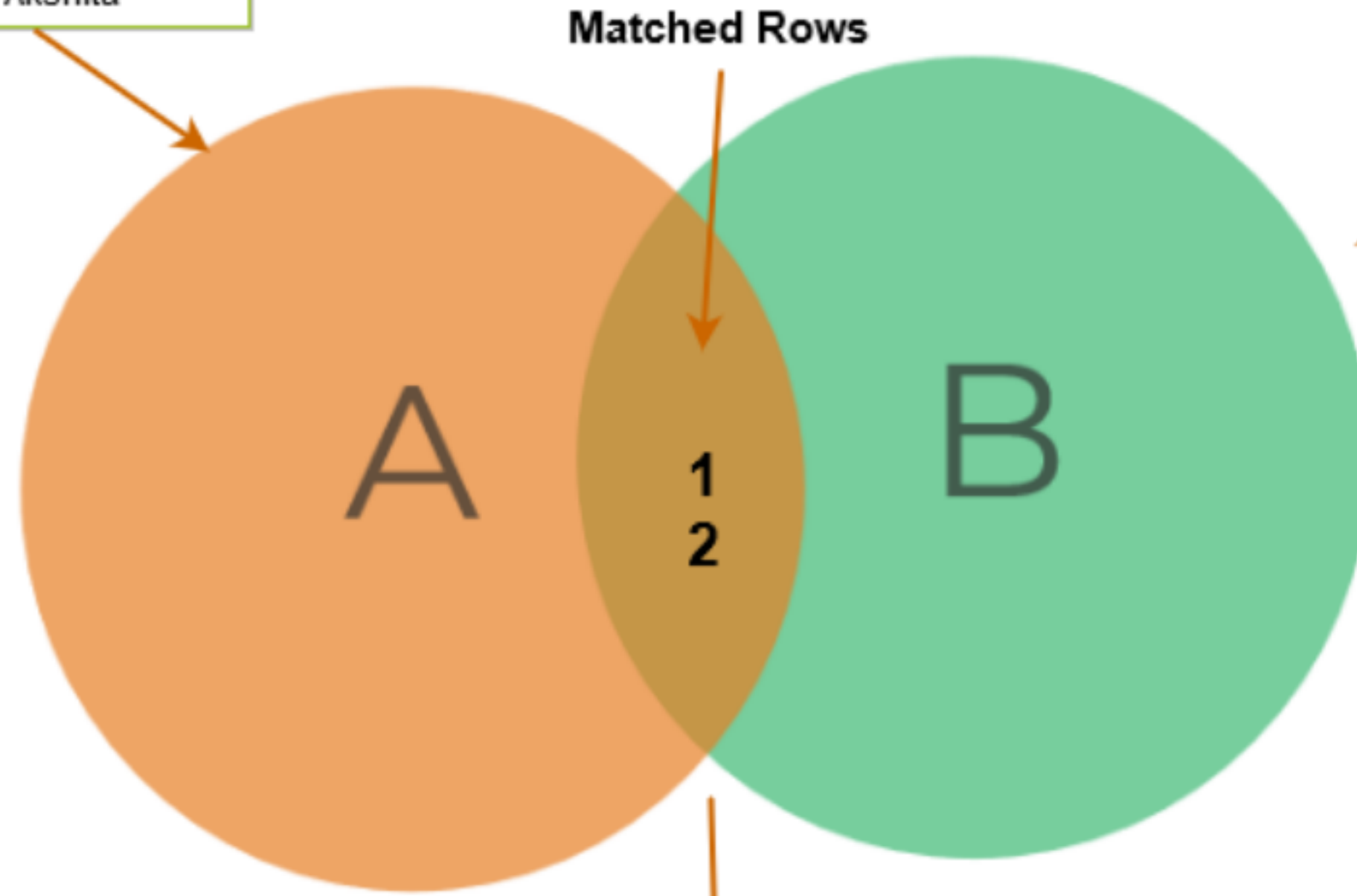
Unmatched rows similar to right outer join: Null values
for unmatched rows from the left table.



EmpID	EmpName
1	Rajendra
2	Kusum
3	Akshita

Full Outer Join

ID	City
1	Jaipur
2	Delhi
7	Raipur
8	Bangalore

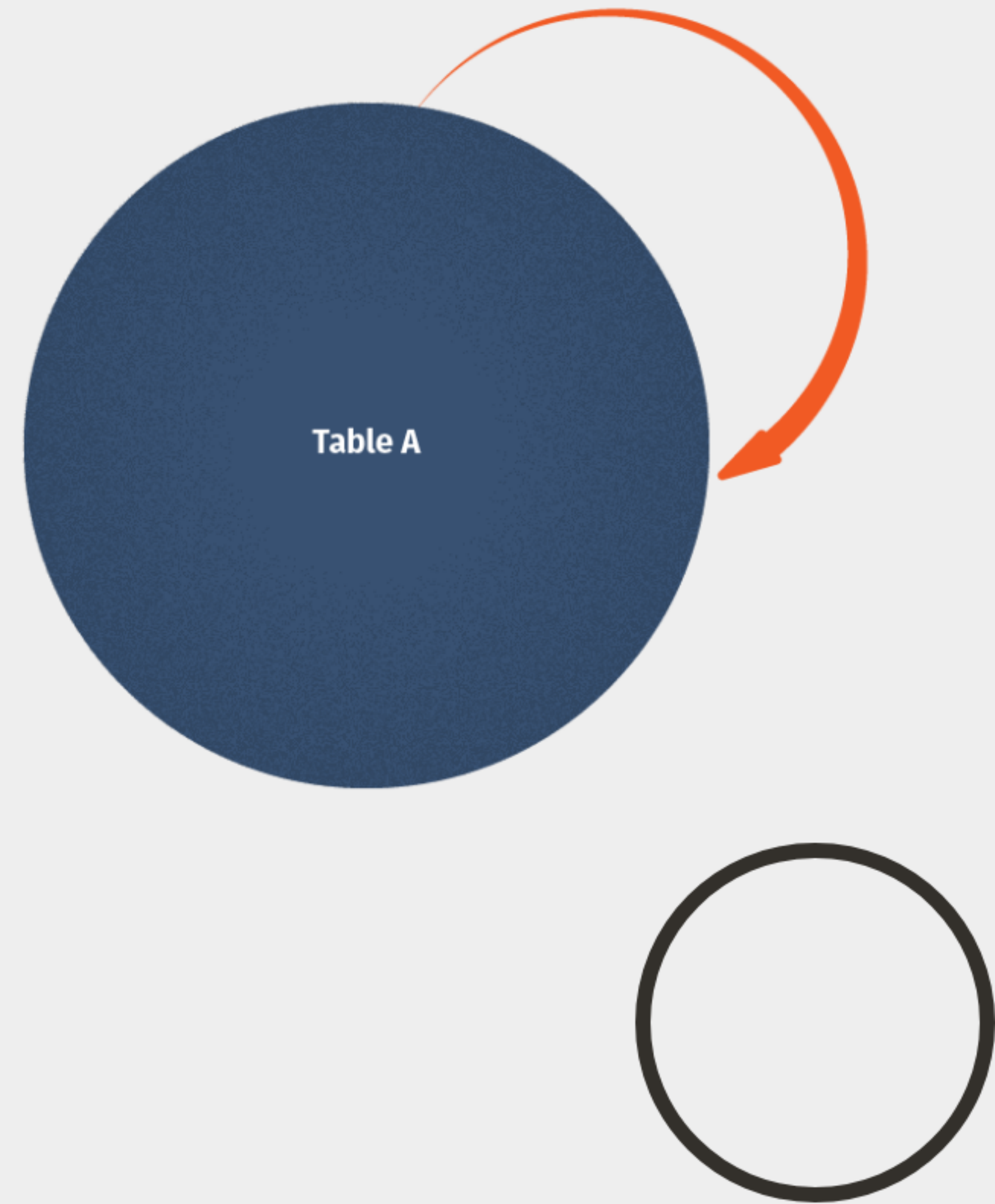


Select EmpName, City from employees e
Full Outer Join address a
on e.empid=a.id

EmpName	City
Rajendra	Jaipur
Kusum	Delhi
Akshita	NULL
NULL	Raipur
NULL	Bangalore

Self Join

The SELF JOIN allows you to join a table to itself. This implies that each row of the table is combined with itself and with every other row of the table. The SELF JOIN can be viewed as a join of two copies of the same table. The table is not actually copied, but SQL performs the command as though it were. This is accomplished by using table name aliases to give each instance of the table a separate name. It is most useful for extracting hierarchical data or comparing rows within the same table.



EmpID	Name	EmpMgrid
1	Rajendra	1
2	Mohan	1
3	Amit	1
4	Manoj	1
5	Manish	2
6	Kapil	2



```
SELECT e.EmpID, e.Name, m.Name as Manager FROM Emp e Inner Join Emp m
On e.EmpMgrid=m.EmpID;
```

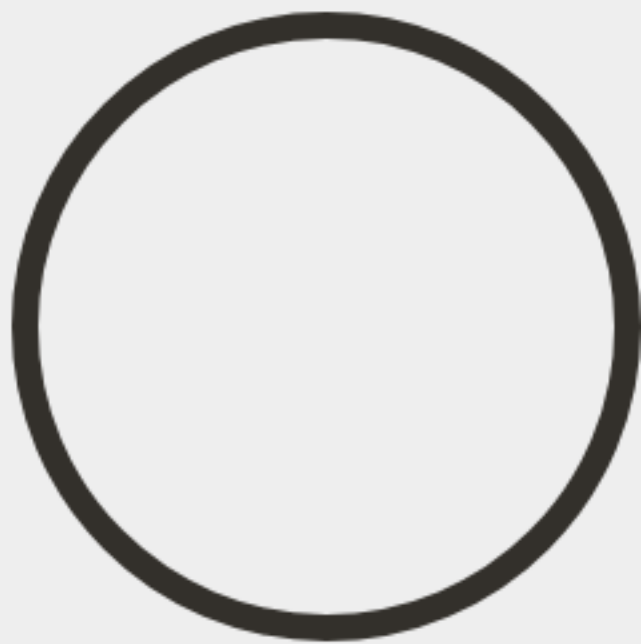
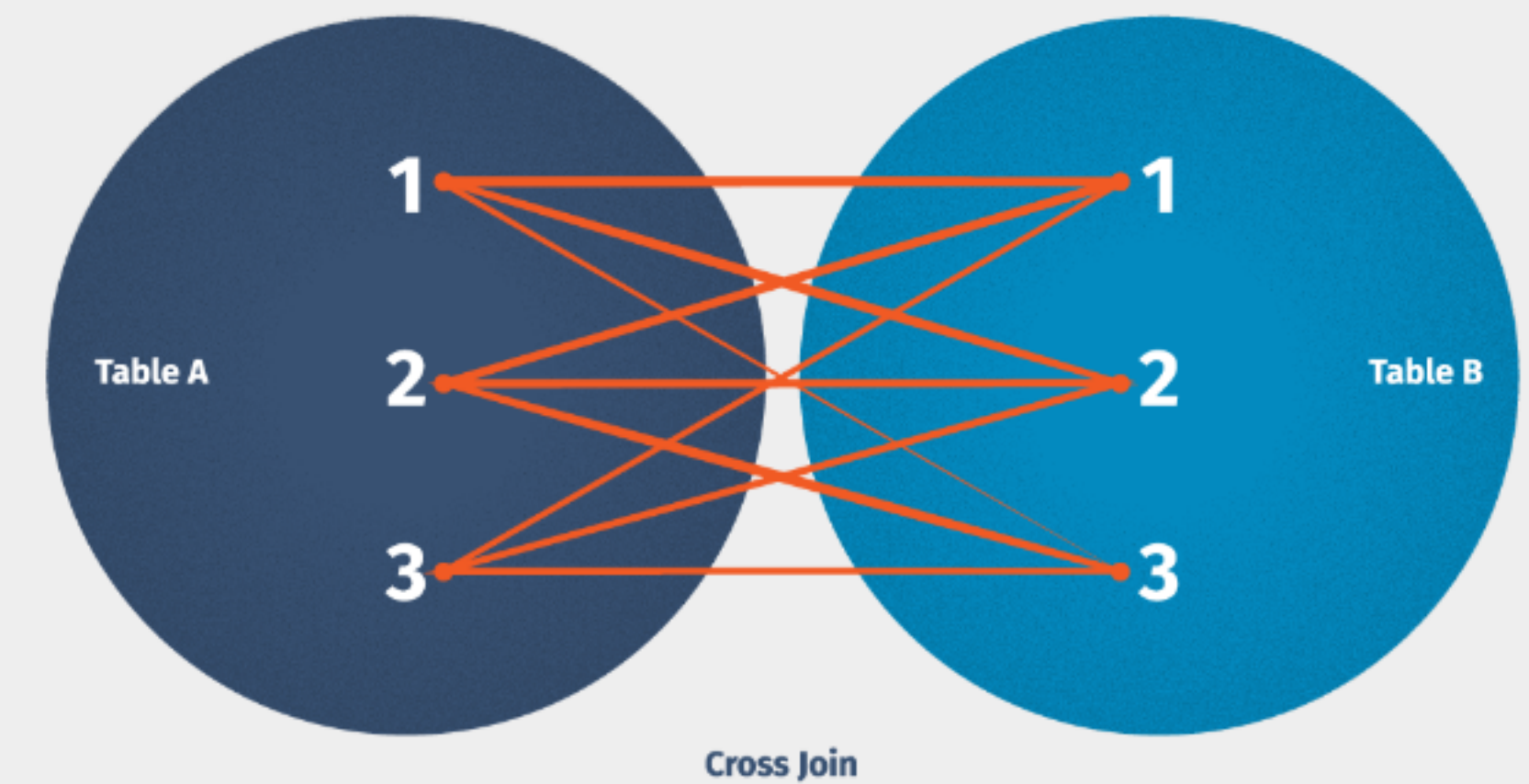
```
SELECT e.EmpID, e.Name,
m.Name as Manager
FROM Emp e Inner Join Emp m
On e.EmpMgrid=m.EmpID;
```

Results Messages

EmpID	Name	Manager
1	Rajendra	Rajendra
2	Mohan	Rajendra
3	Amit	Rajendra
4	Manoj	Rajendra
5	Manish	Mohan
6	Kapil	Mohan

Cross Join

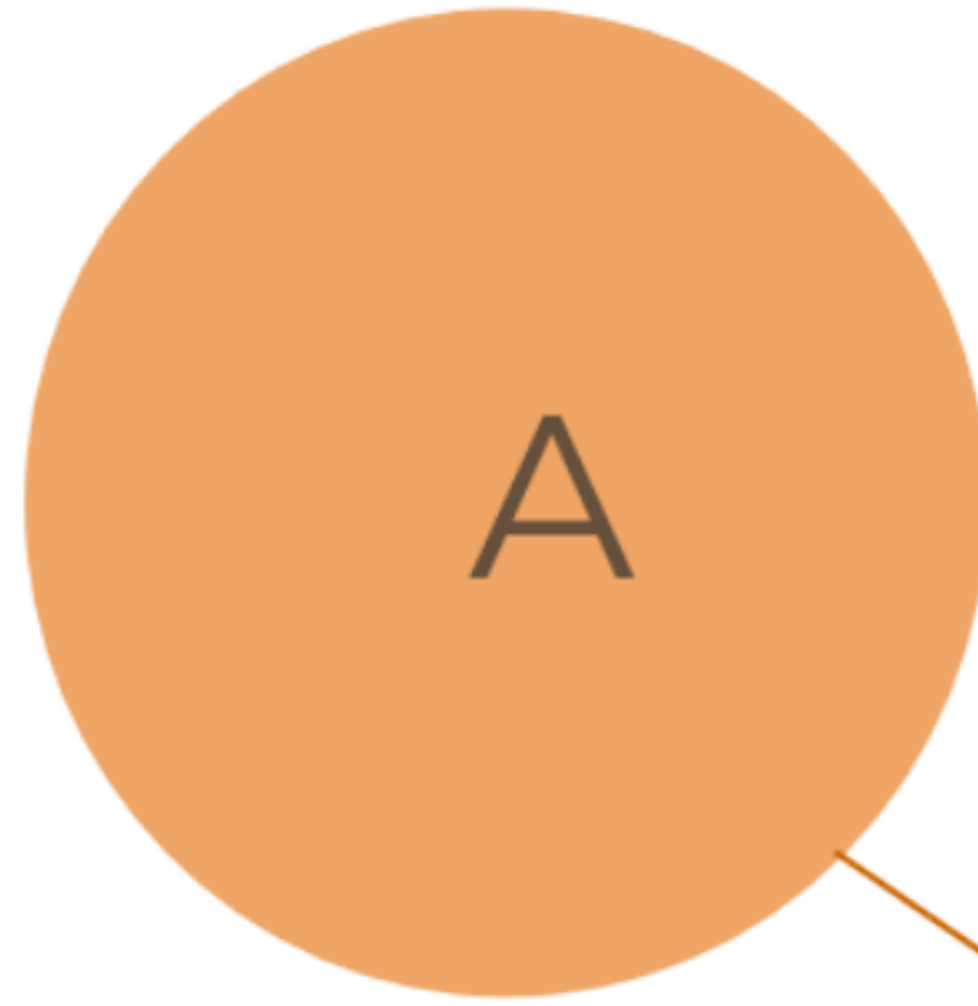
The CROSS JOIN command in SQL, also known as a cartesian join, returns all combinations of rows from each table. Envision that you need to find all combinations of size and color. In that case, a CROSS JOIN will be an asset. Note, that this join does not need any condition to join two tables. In fact, CROSS JOIN joins every row from the first table with every row from the second table and its result comprises all combinations of records in two tables.



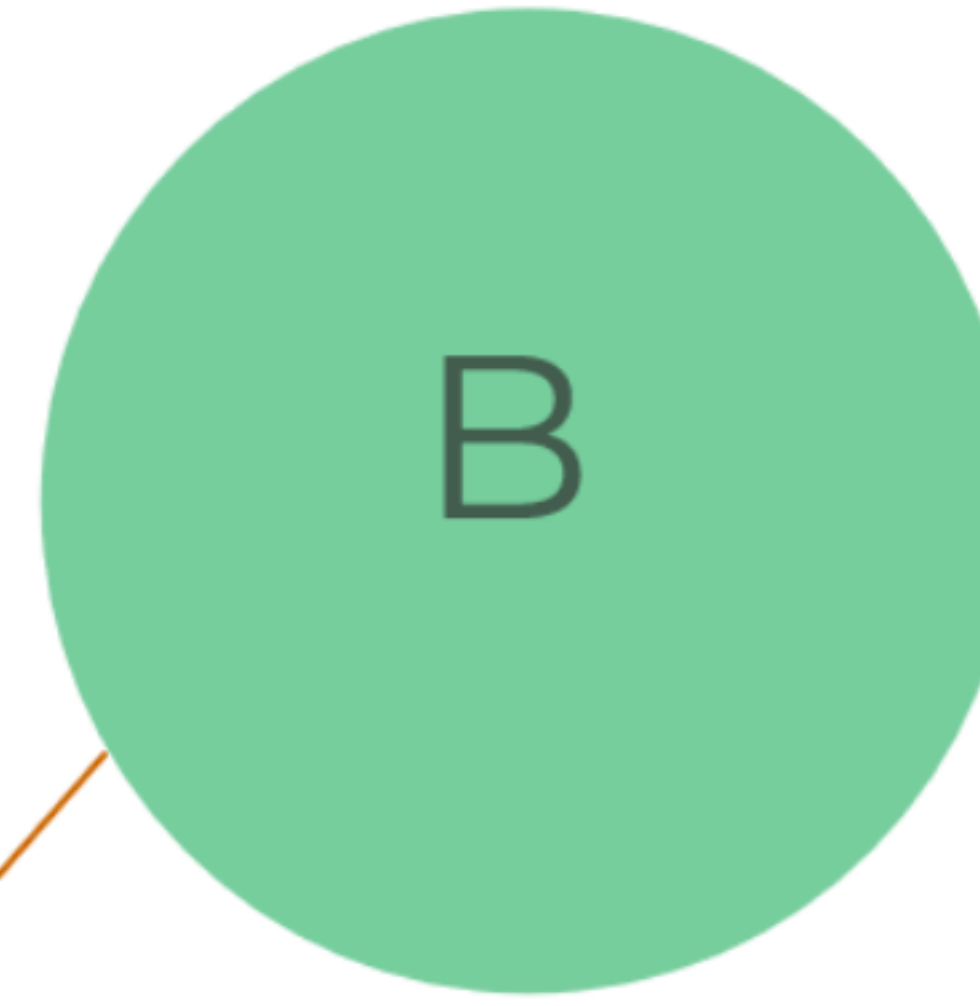
EmpID	EmpName
NULL	Rajendra
2	Kusum
3	Akshita

ID	City
1	Jaipur
2	Delhi
7	Raipur
8	Bangalore

Cross Join



$A * B$



Select EmpName, City
from employees e
CROSS join address a

EmpName	City
Rajendra	Jaipur
Rajendra	Delhi
Rajendra	Raipur
Rajendra	Bangalore
Kusum	Jaipur
Kusum	Delhi
Kusum	Raipur
Kusum	Bangalore
Akshita	Jaipur
Akshita	Delhi
Akshita	Raipur



Benefits of using Join Queries in SQL

■ Improved Query Performance

Join queries allow for the retrieval of data from multiple tables in a single query, which reduces the number of queries needed and improves overall query performance.

■ Increased Flexibility

Join queries allow for the combination of data from multiple tables, providing increased flexibility in querying and reporting on data.

■ Efficient Data Retrieval


Join queries retrieve only the necessary data from multiple tables, reducing the amount of unnecessary data returned and improving overall query efficiency.

■ Simplified Data Analysis

Join queries simplify data analysis by allowing for the combination of data from multiple tables, reducing the need for complex and time-consuming data manipulation.

■ Real-World Applicability

Join queries are widely used in real-world scenarios, including e-commerce, financial services, and healthcare, making them an essential skill for software developers.





Conclusion