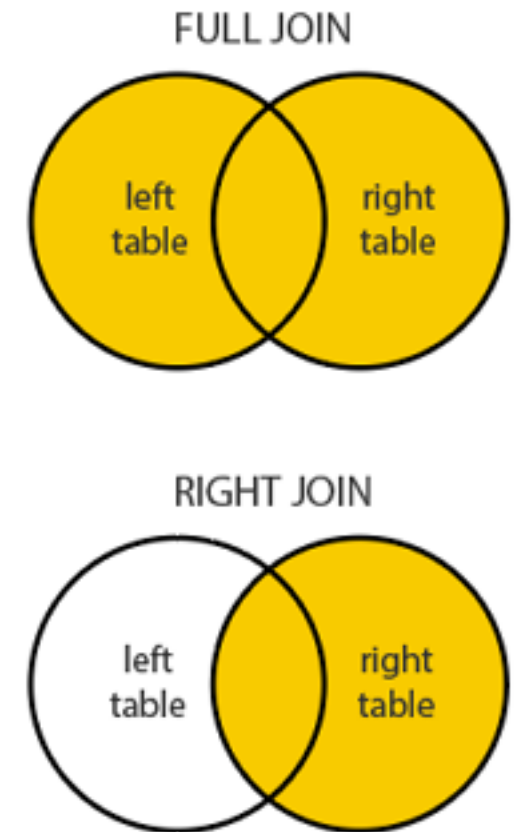
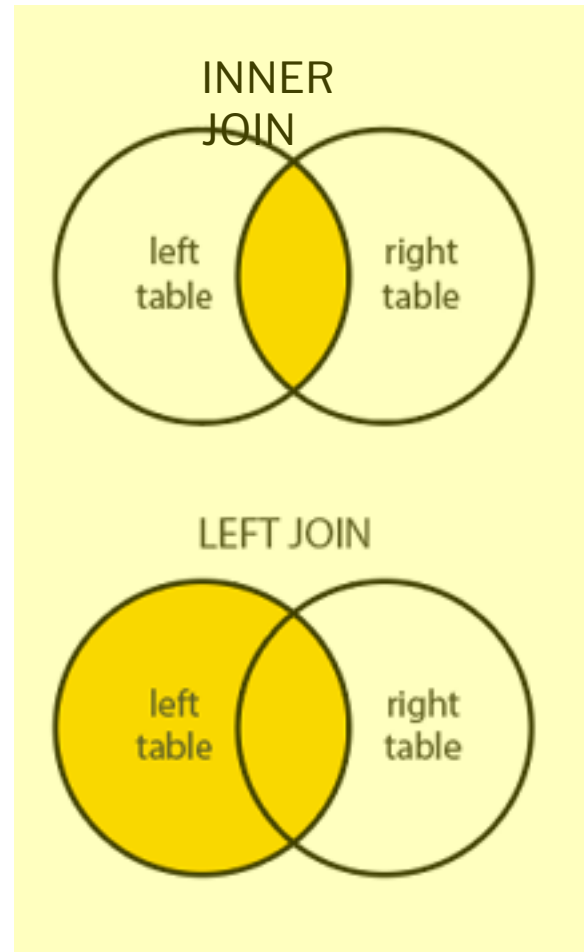


SQL JOINS

JOINING TABLES

- **JOIN** is used to combine rows from two or more tables based on a related column between them.
- There 4 major types of joins.
 - **INNER JOIN** returns rows when there is a match in both tables.
 - **LEFT JOIN** returns all rows from the left table and the matched rows from the right table
 - **RIGHT JOIN** returns all rows from the right table and the matched rows from the left table
 - **FULL (OUTER) JOIN** returns all rows when there is a match in either table. If there is no match



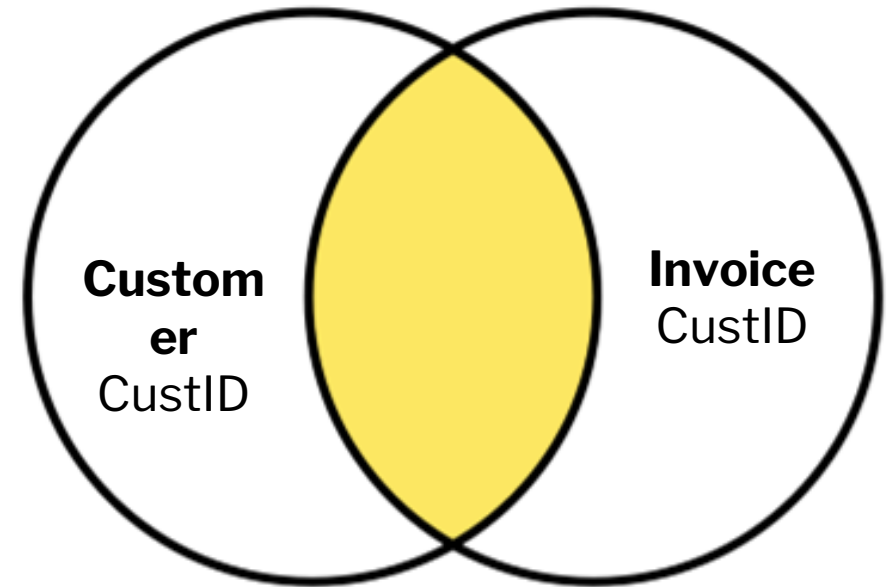
INNER JOIN

- An INNER JOIN returns rows when there is a match in both tables. It excludes rows where the condition is not met.

Retrieve customer names and their invoice totals by joining Customer and Invoice

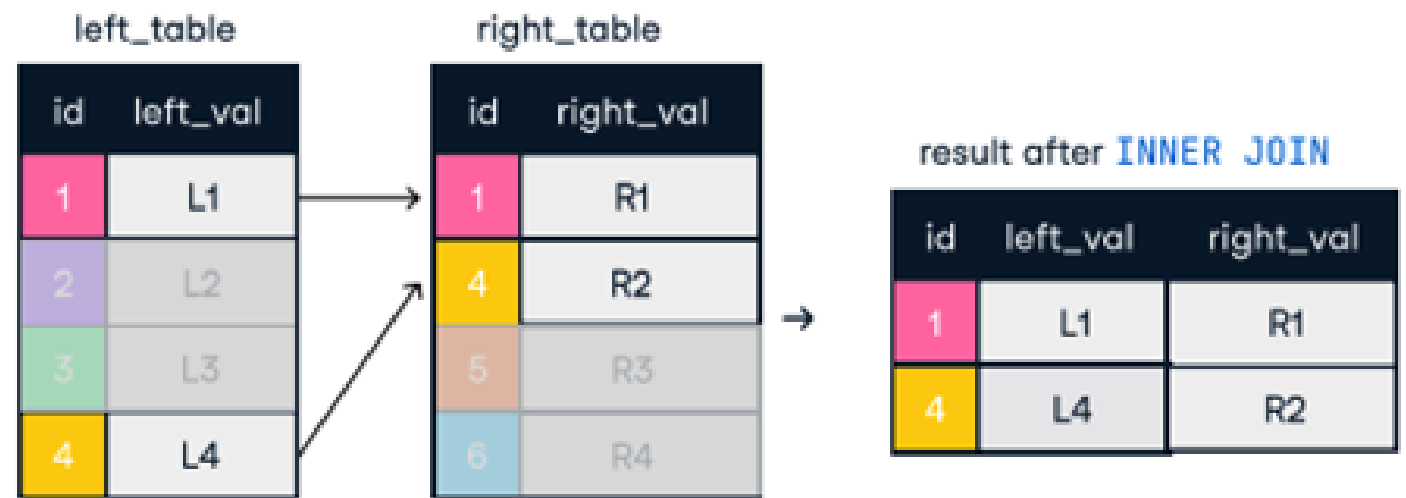
```
SELECT c.FirstName, c.LastName, i.Total  
FROM Customer c  
INNER JOIN Invoice i ON c.CustomerId =  
i.CustomerId;
```

INNER JOIN



INNER JOIN

An inner join between two tables will return only records where a joining field, such as a key, finds a match in both tables.



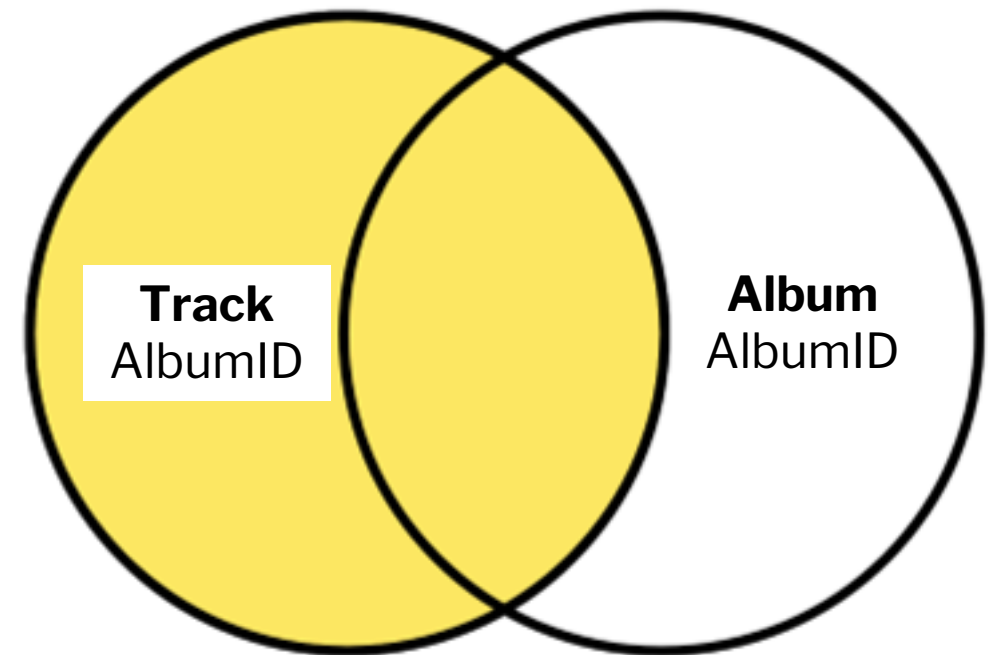
LEFT JOIN

A LEFT JOIN returns all rows from the left table and the matched rows from the right table. If there is no match, the result will include NULL values for the columns from the right table.

List all tracks and their corresponding album titles, including tracks without an album assigned:

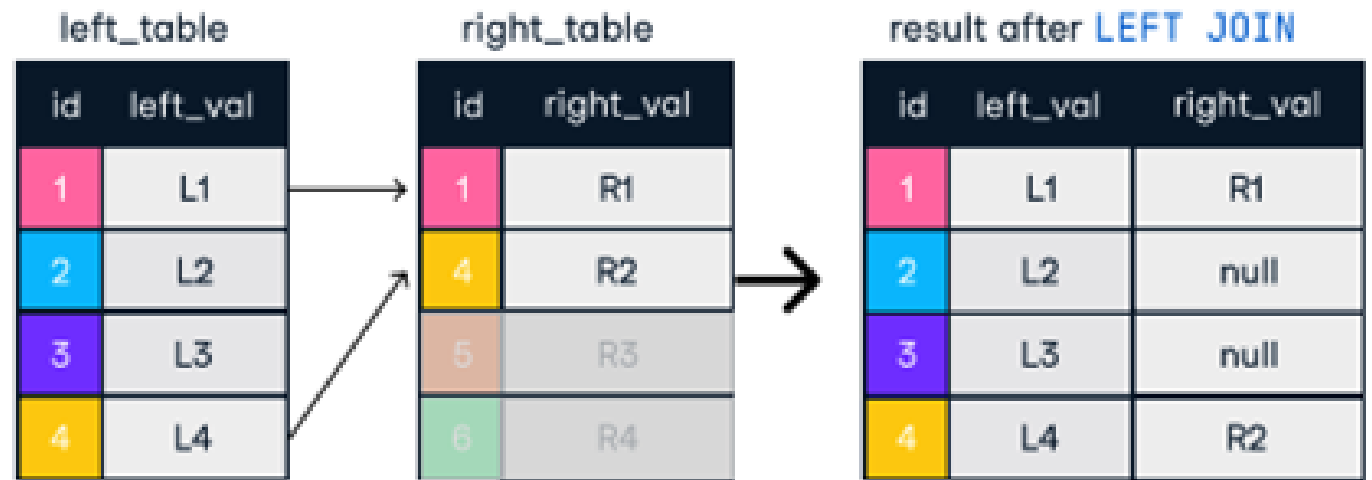
```
SELECT t.Name, a.Title  
FROM Track t  
LEFT JOIN Album a ON t.AlbumId = a.AlbumId;
```

LEFT JOIN



LEFT JOIN

A left join keeps all of the original records in the left table and returns missing values for any columns from the right table where the joining field did not find a match.



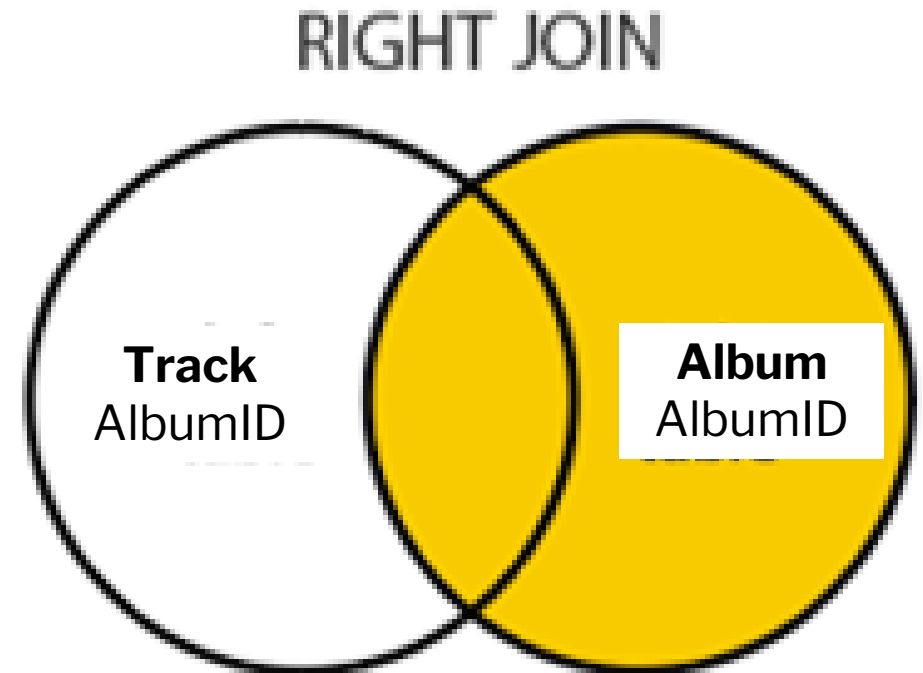
RIGHT JOIN

A RIGHT JOIN returns all rows from the right table and the matched rows from the left table. If there's no match, the result will include NULL values for the left table.

List all albums and the tracks they contain, including albums without tracks

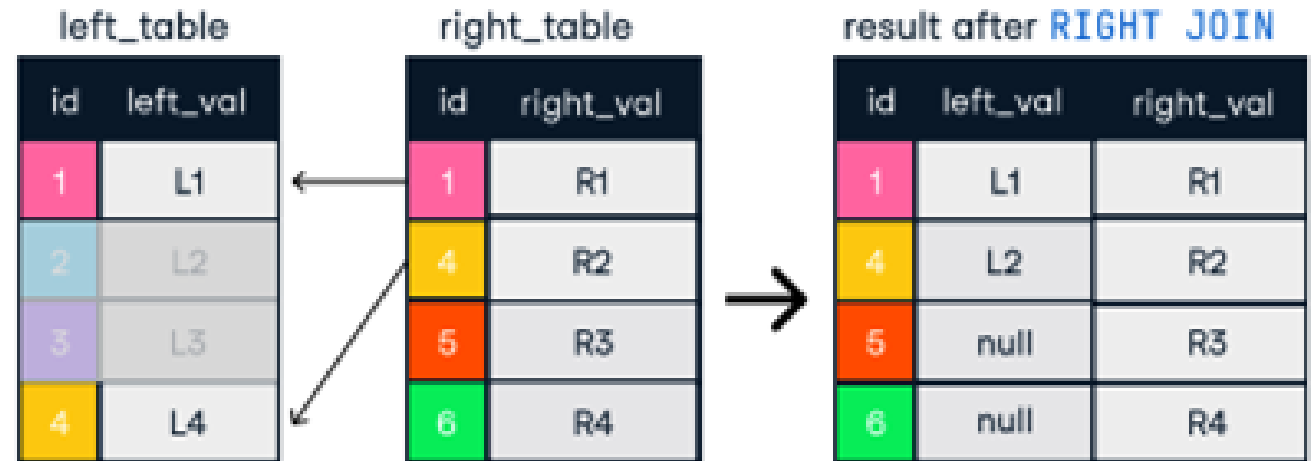
```
SELECT a.Title, t.Name  
FROM Track t  
RIGHT JOIN Album a ON t.AlbumId = a.AlbumId;
```

Can be converted into a LEFT JOIN by reversing the tables



RIGHT JOIN

A right join keeps all of the original records in the right table and returns missing values for any columns from the left table where the joining field did not find a match. Right joins are far less common than left joins, because right joins can always be re-written as left joins.

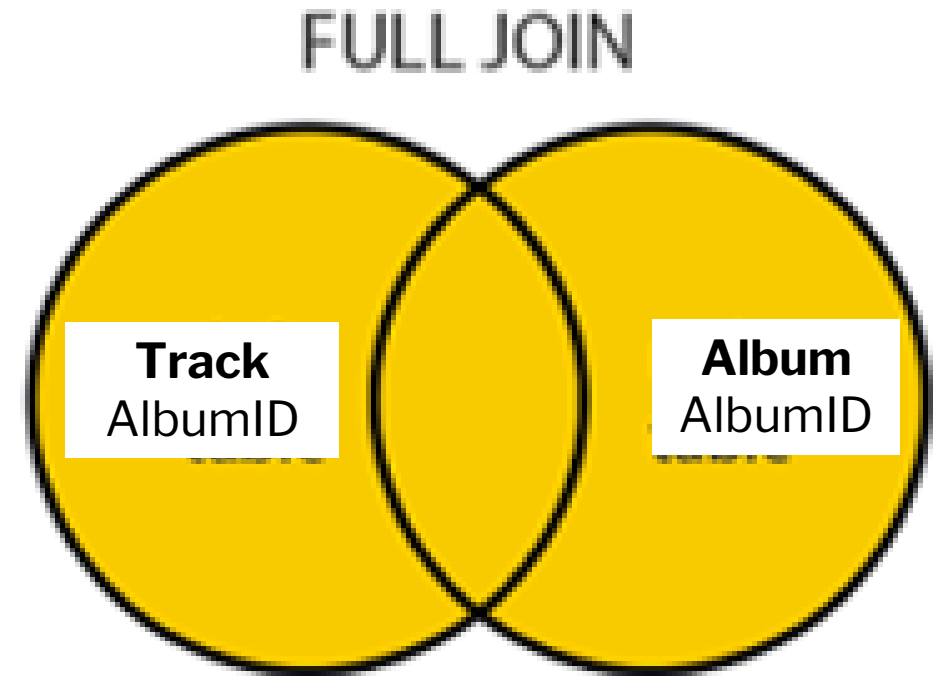


FULL JOIN (AKA FULL OUTER JOIN)

A FULL OUTER JOIN returns all rows when there is a match in either table. If there is no match, the result will include NULL values for the missing data from both tables.

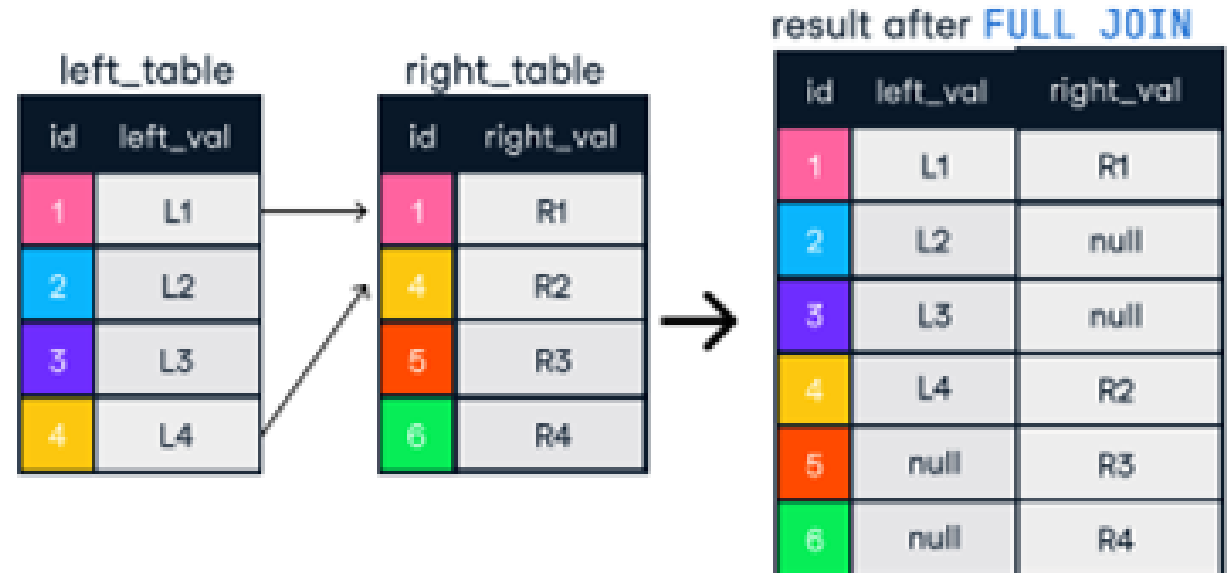
Retrieve all artists and albums, showing nulls where there's no match:

```
SELECT ar.Name AS ArtistName, al.Title AS  
AlbumTitle  
FROM Artist ar  
FULL OUTER JOIN Album al ON ar.ArtistId =  
al.ArtistId;
```



FULL JOIN

A full join combines a left join and right join. A full join will return all records from a table, irrespective of whether there is a match on the joining field in the other table, returning null values accordingly.



UNION

- The UNION operator is used to combine results of two or more SELECT queries into a single result set.
- It removes duplicate rows by default, so the result set contains only unique rows.
- If you want to include all duplicate rows, you can use UNION ALL.

Key Points:

- **Queries must have the same number of columns.**
- **The columns must have compatible data types** in corresponding positions.
- The **column names in the result** will be based on the first SELECT statement.
- By default, UNION removes duplicates, but UNION ALL keeps all rows.

Syntax:

```
SELECT column1, column2, ...  
FROM table1  
UNION  
SELECT column1, column2, ...  
FROM table2;
```

Let's say we have two tables: VIPCustomer (for VIP customers) and RegularCustomer (for regular customers). We want to retrieve a list of all customers, combining both tables.

```
SELECT CustomerID, FirstName, LastName  
FROM VIPCustomer  
UNION  
SELECT CustomerID, FirstName, LastName  
FROM RegularCustomer;
```

UNION

The **UNION** operator is used to vertically combine the results of two **SELECT** statements. For **UNION** to work without errors, all **SELECT** statements must have the same number of columns and corresponding columns must have the same data type. **UNION** does not return duplicates.

