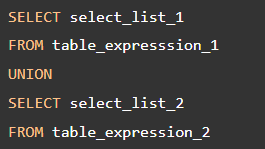
PostgreSQL UNION

The UNION operator combines result sets of two or more [SELECT](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-select/) statements into a single result set.

The following illustrates the syntax of the UNION operator that combines result sets from two queries:



To combine the result sets of two queries using the UNION operator, the queries must conform to the following rules:

* The number and the order of the columns in the select list of both queries must be the same.
* The data types must be compatible.

The UNION operator removes all duplicate rows from the combined data set. To retain the duplicate rows, you use the the UNION ALL instead.

The following Venn digram illustrates how to the UNION works:



### **PostgreSQL UNION with ORDER BY clause**

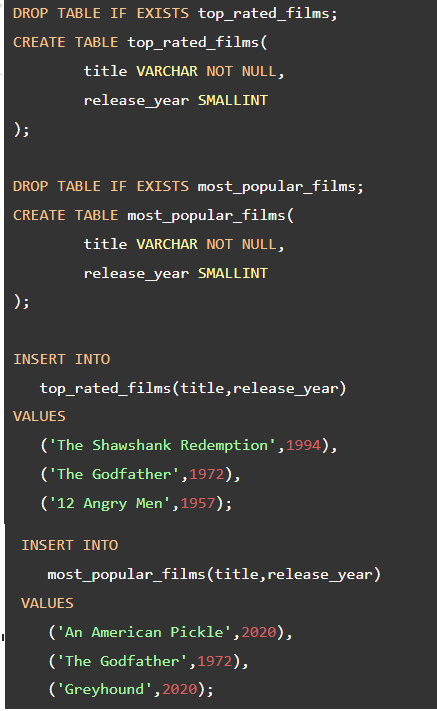
The UNION operator may place the rows from the result set of the first query before, after, or between the rows from the result set of the second query.

To sort rows in the final result set, you use the [ORDER BY](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-order-by/) clause in the second query.

In practice, you often use the UNION operator to combine data from similar tables, which are not perfectly normalized, in the data warehouse or business intelligence systems.

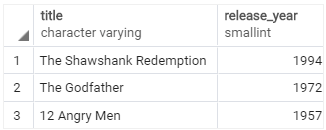
## **Setting up sample tables**

The following statements create two tables: top\_rated\_films and most\_popular\_films, and insert data into these tables:



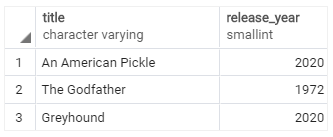
The following shows the data from the top\_rated\_films table:





The following statement returns the data from the most\_popular\_films table:



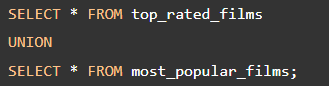


## **PostgreSQL UNION examples**

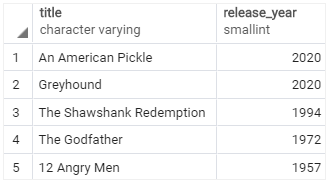
Let’s take some examples of using the PostgreSQL UNION operator.

### **1) Simple PostgreSQL UNION example**

The following statement uses the UNION operator to combine data from both tables:



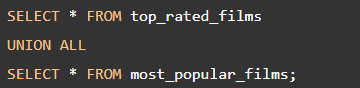
The query returns the following result:

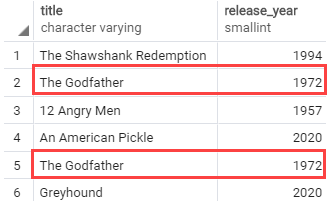


The result set includes five rows in the result set because the UNION operator removes one duplicate row.

### **2) PostgreSQL UNION ALL example**

The following statement uses the UNION ALL operator to combine result sets from the top\_rated\_films and most\_popular\_films tables:

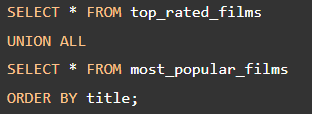


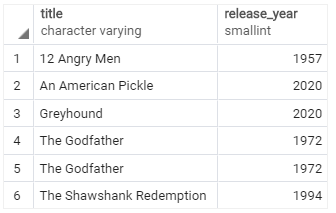


In this example, the duplicate row is retained in the result set.

### **3) PostgreSQL UNION ALL with ORDER BY clause example**

To sort the result returned by the UNION operator, you place the ORDER BY clause end of the last query like this:





If you place the ORDER BY clause at the end of each query, the combined result set will not be sorted as you expected.

Because when UNION operator combines the sorted result sets from each query, it does not guarantee the order of rows in the final result set.

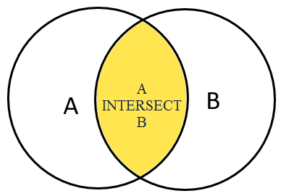
In this tutorial, you have learned how to use the PostgreSQL UNION and UNION ALL to combine the result sets from multiple queries into a single result set.

# PostgreSQL INTERSECT Operator

Like the [UNION](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-union/) and [EXCEPT](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-tutorial/postgresql-except/) operators, the PostgreSQL INTERSECT operator combines result sets of two or more [SELECT](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-select/) statements into a single result set.

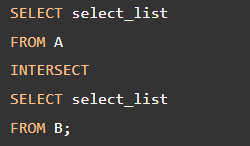
The INTERSECT operator returns any rows that are available in both result sets.

The following illustration shows the final result set produced by the INTERSECT operator.



The final result set is represented by the yellow area where circle A intersects with circle B.

The following illustrates the syntax of the INTERSECT operator:

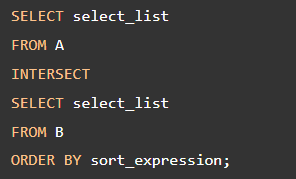


To use the INTERSECT operator, the columns that appear in the SELECT statements must follow the folowing rules:

1. The number of columns and their order in the SELECT clauses must be the same.
2. The [data types](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-data-types/) of the columns must be compatible.

### **PostgreSQL INTERSECT with ORDER BY clause**

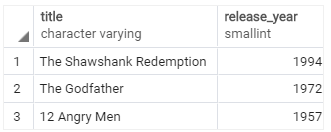
If you want to sort the result set returned by the INTERSECT operator, you place the ORDER BY at the final query in the query list like this:



## **PostgreSQL INTERSECT operator examples**

We’ll use the top\_rated\_films and most\_popular\_films tables created in the [UNION tutorial](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-union/):

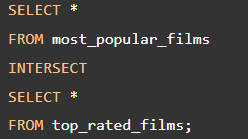
The top\_rated\_films table:



The most\_popular\_films table:



To get popular films which are also top rated films, you use the INTERSECT operator as follows:





The result set returns one film that appears on both tables.

In this tutorial, you have learned how to use the PostgreSQL INTERSECT operator to combine result sets returned by multiple queries.

References

https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-union/

https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-intersect/