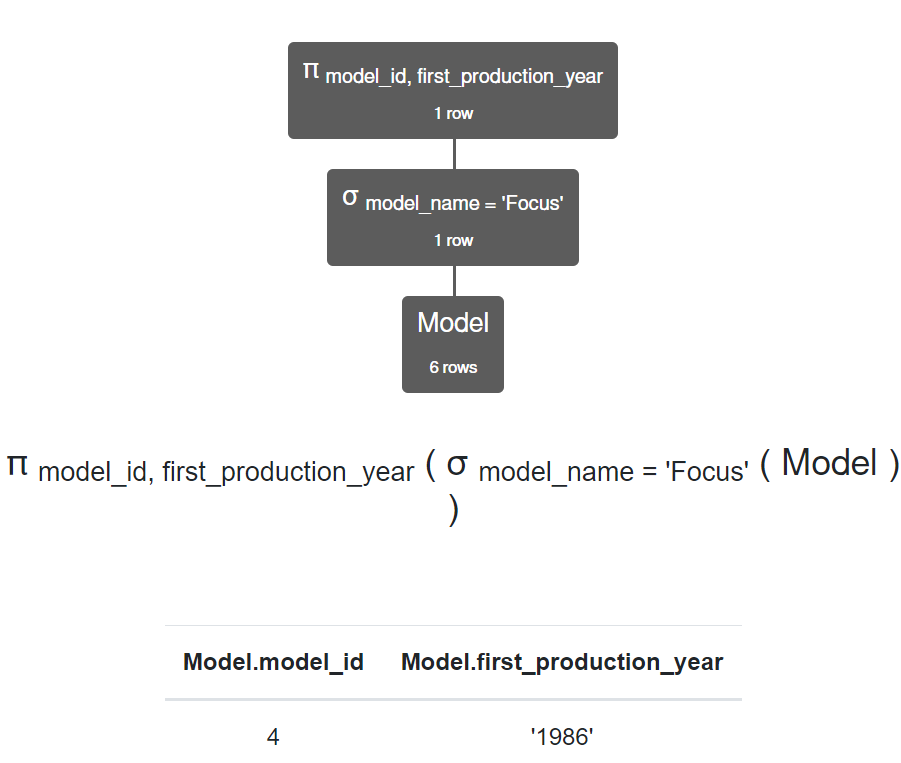
# Project (∏)

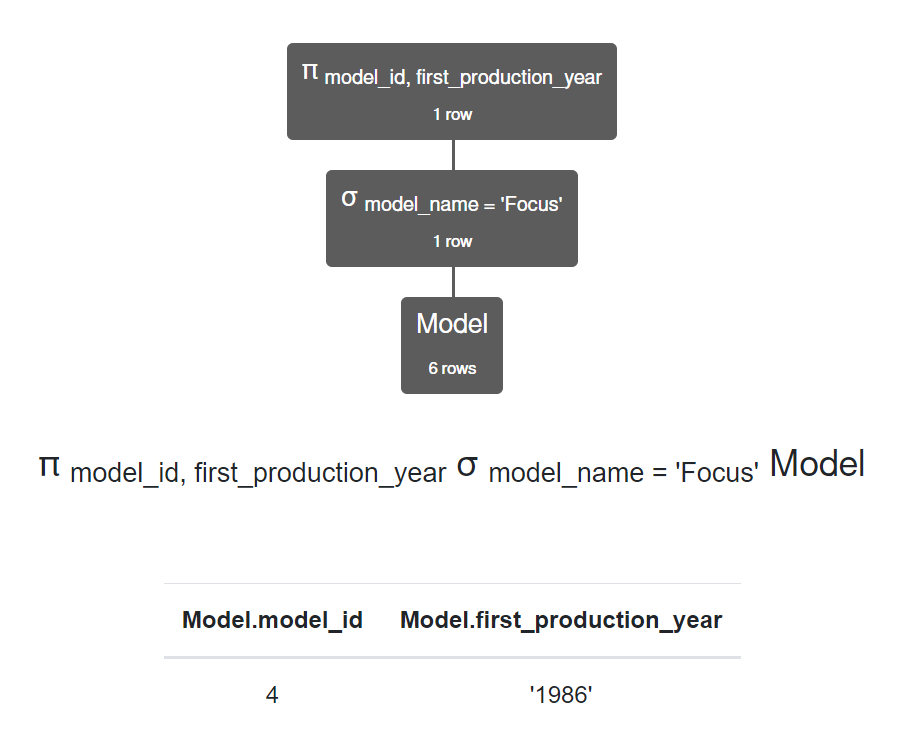
**Relational Algebra**

π model\_id, first\_production\_year (σ model\_name = 'Focus' (Model))



**SQL Query**

SELECT model\_id, first\_production\_year FROM Model WHERE model\_name='Focus';

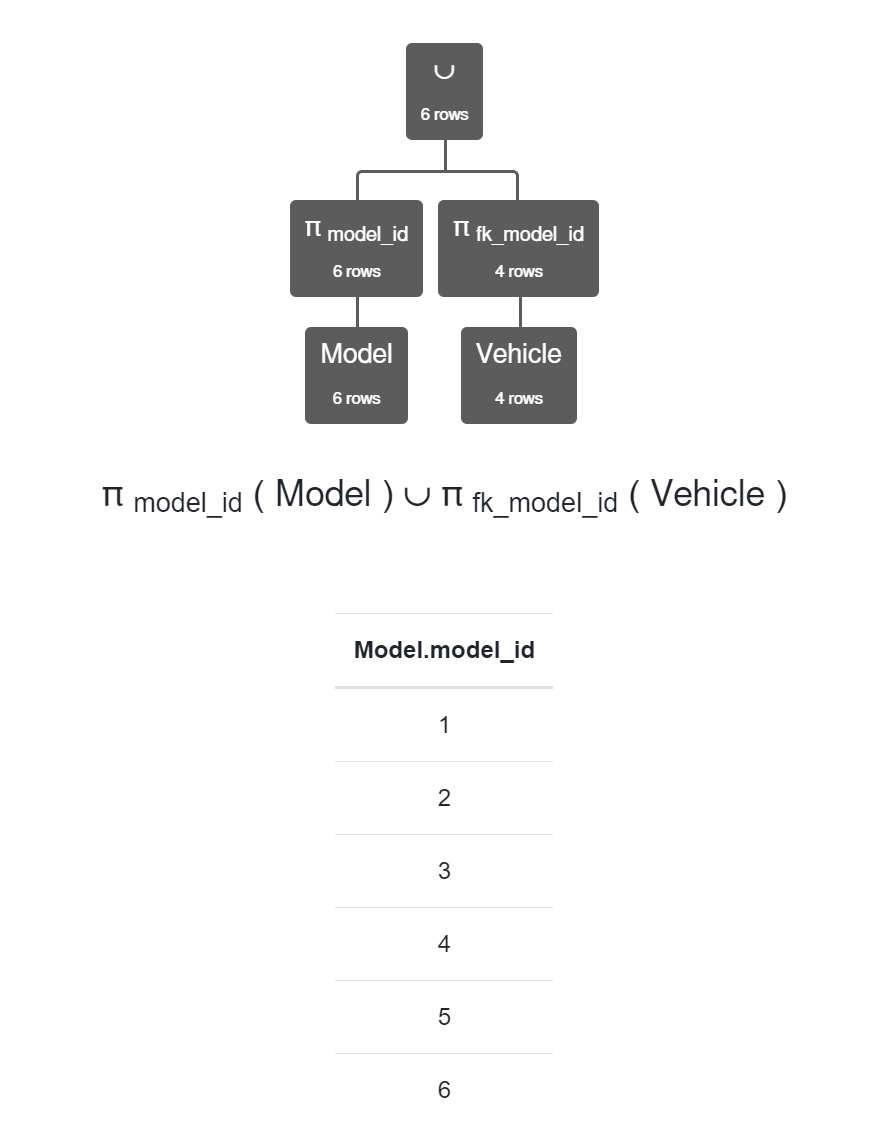


***Explanation:*** The Project operator will select columns to show, in this case the query will display the columns model\_id and first\_production\_year in the selected table where select criteria is met.

# Union (∪)

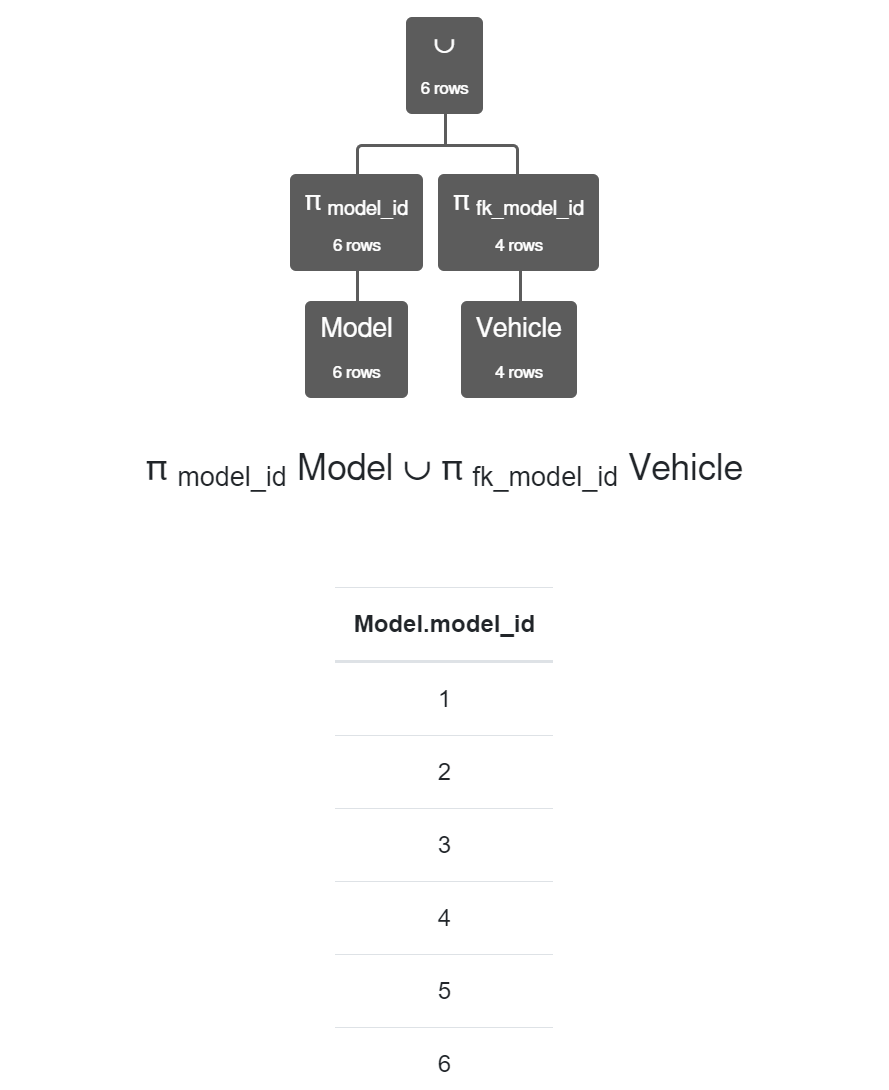
**Relational Algebra**

π model\_id (Model) ∪ π fk\_model\_id (Vehicle)



**SQL Query**

SELECT model\_id FROM Model UNION SELECT fk\_model\_id FROM Vehicle;

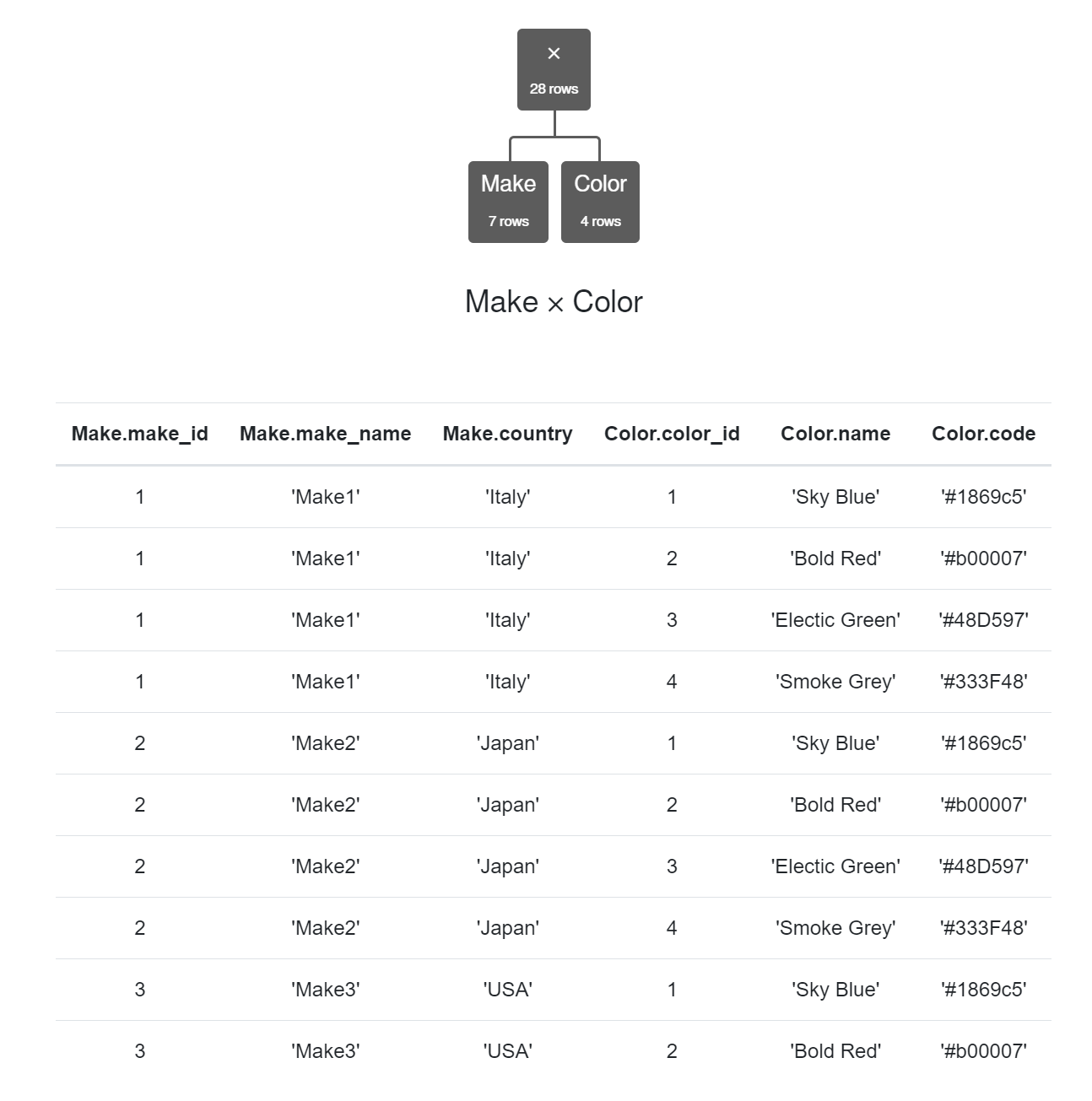


***Explanation:*** The Union operator will combine tables as long as the data types and columns are the same. In the example for tables Model and Vehicle both have a model\_id that and a number data type.

# Cartesian Product (X)

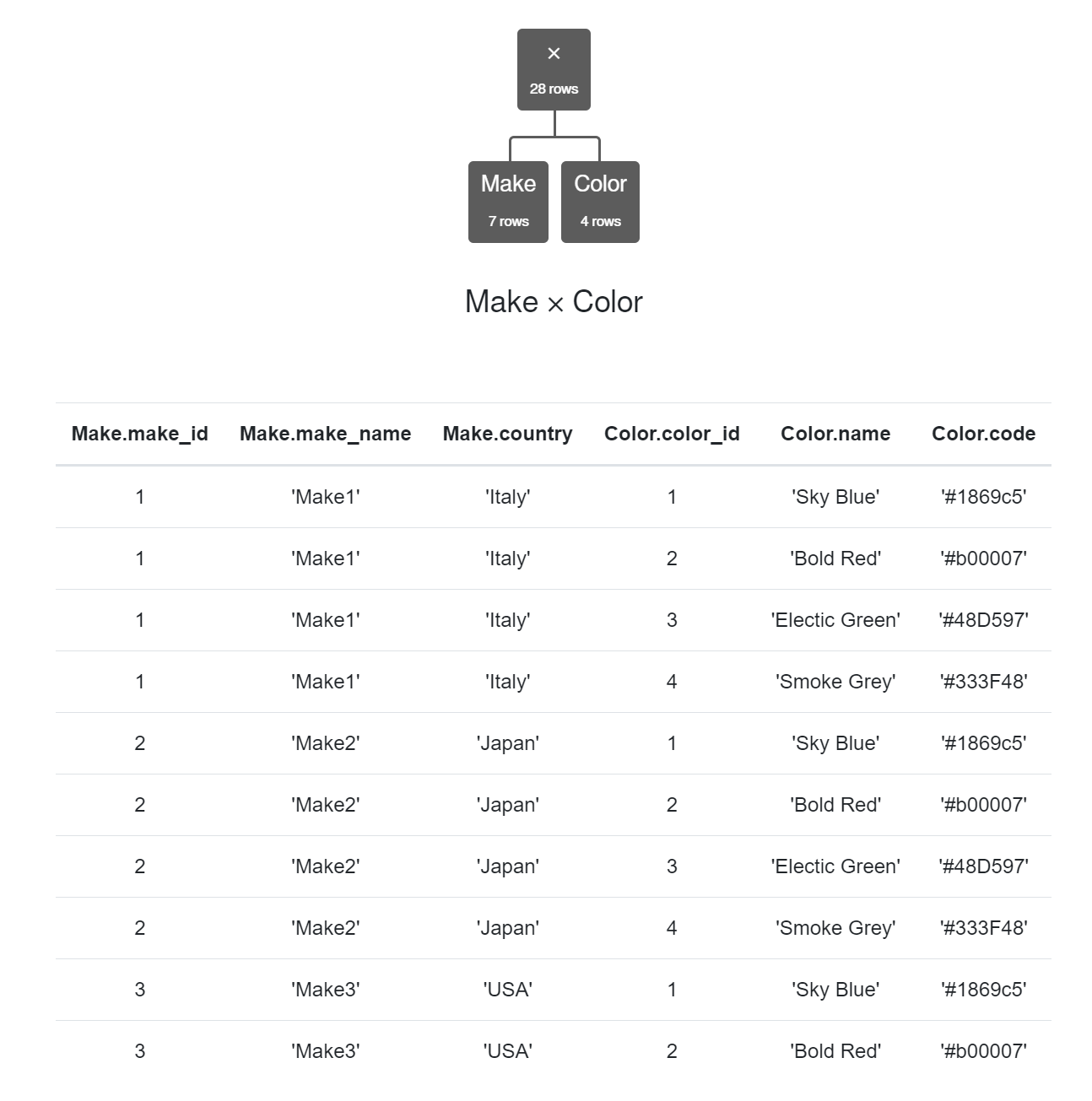
**Relational Algebra**

Make ⨯ Color

****

**SQL Query**

SELECT \* FROM Make, Color**;**

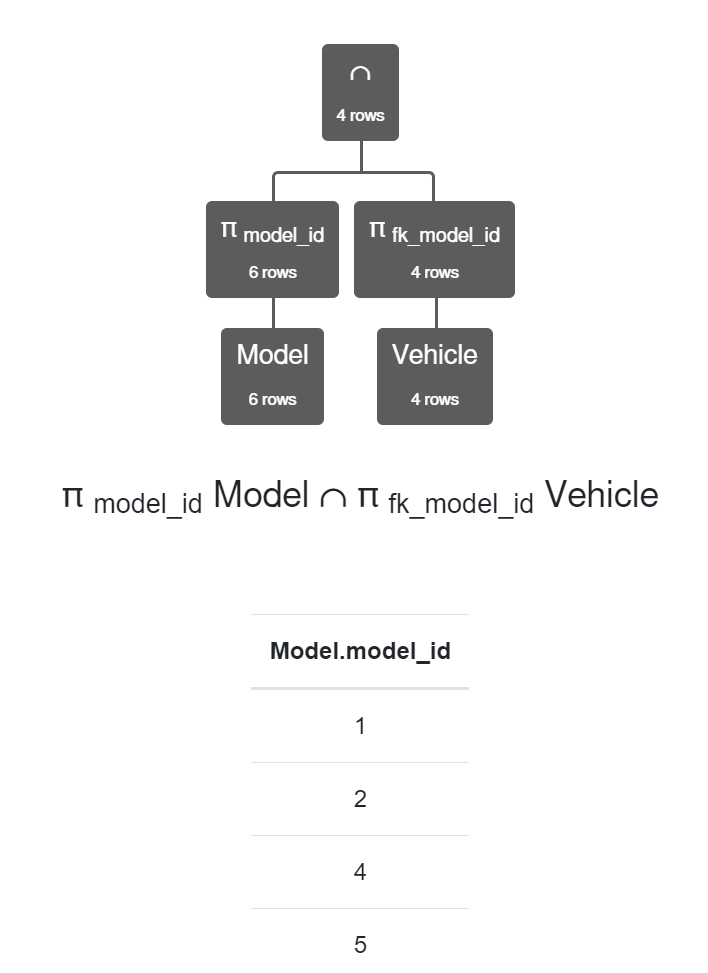
****

***Explanation:*** The Cartesian Product operator will join columns to show the results of both tables by adding all columns from each, in this case the query will display all columns from Make and Color.

# Intersect (∩)

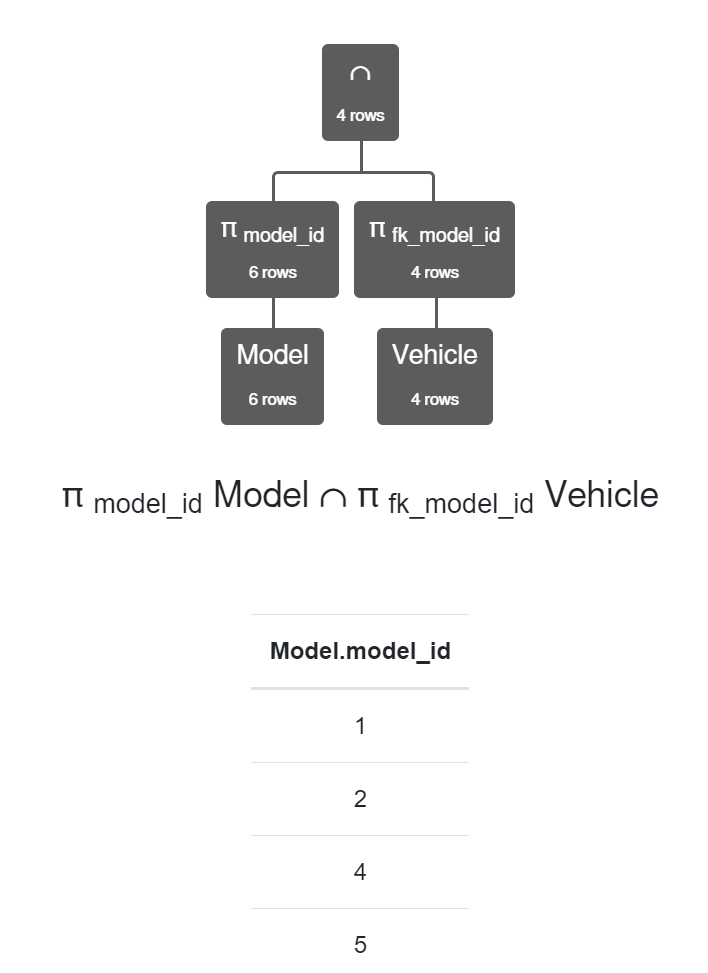
**Relational Algebra**

π model\_id Model ∩ π fk\_model\_id Vehicle

****

**SQL Query**

SELECT model\_id FROM Model INTERSECT SELECT fk\_model\_id FROM Vehicle;

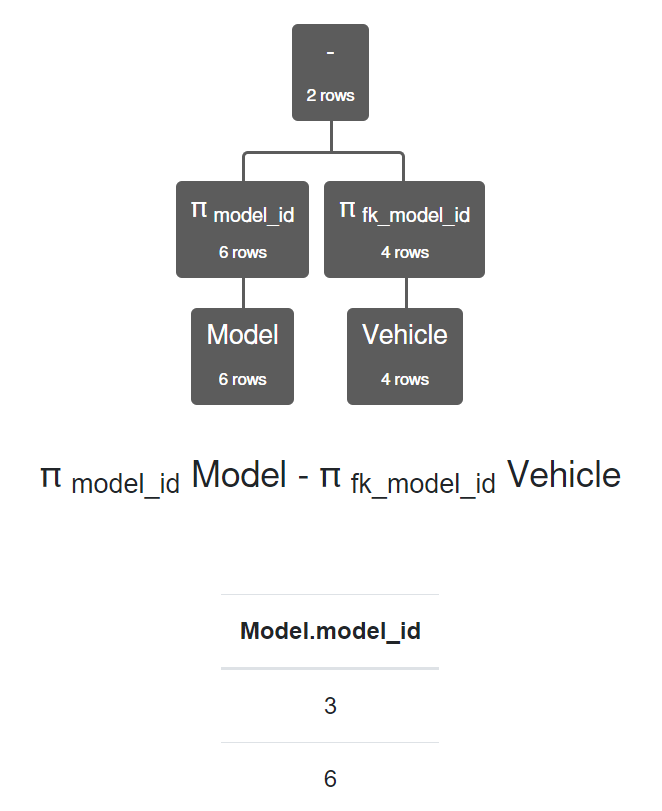
****

***Explanation:*** The Intersect operator combine columns but return only those from that match. In this case it will return all model\_id that are in both Model and Vehicle (1, 2, 4, and 5).

# Difference (-)

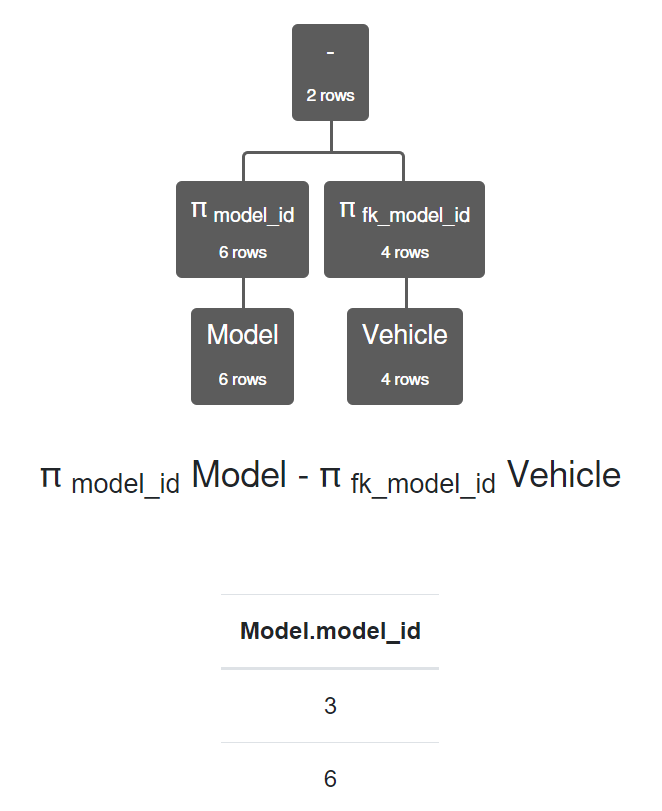
**Relational Algebra**

π model\_id Model - π fk\_model\_id Vehicle



**SQL Query**

SELECT model\_id FROM Model EXCEPT SELECT fk\_model\_id FROM Vehicle

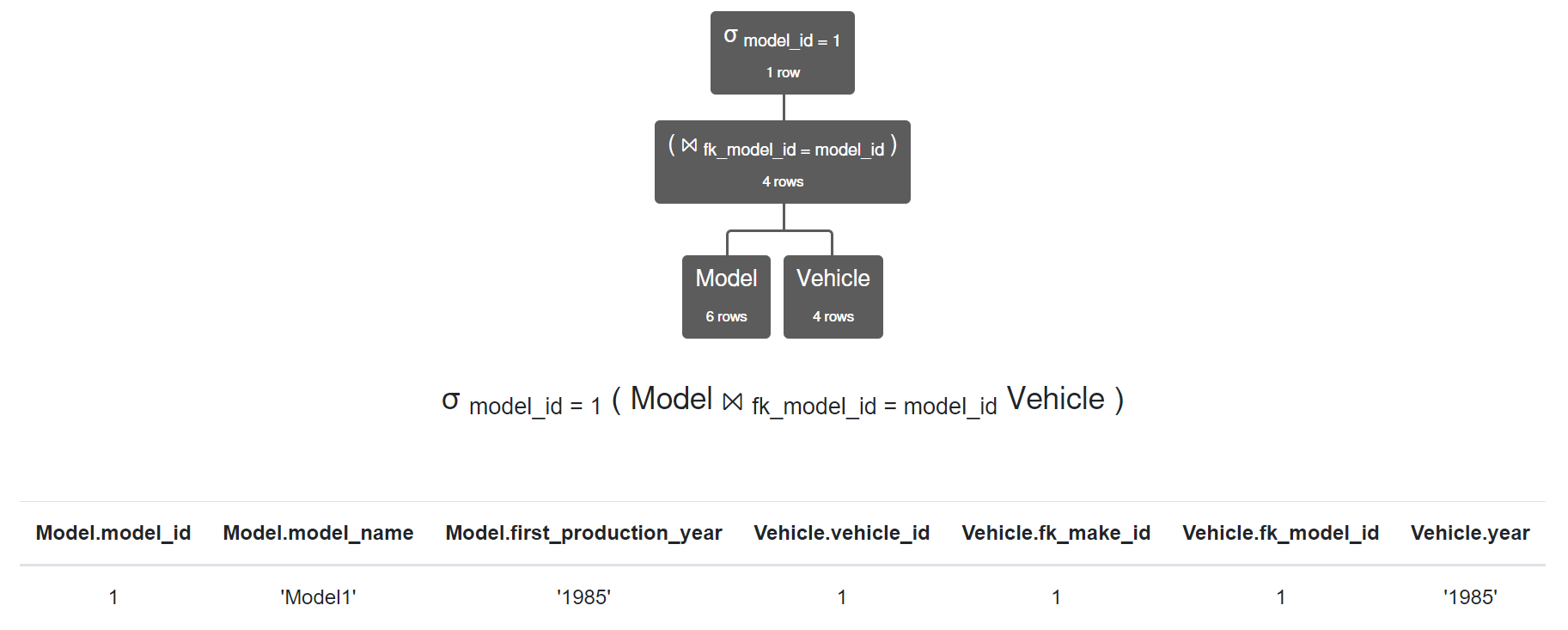


***Explanation:*** The Difference operator will return all that are in the first relation but are not on the second one. In this case it will return all model\_id that are in Model but not in Vehicle (3 and 6).

# Join (⋈, ⟕, ⟖, or ⟗)

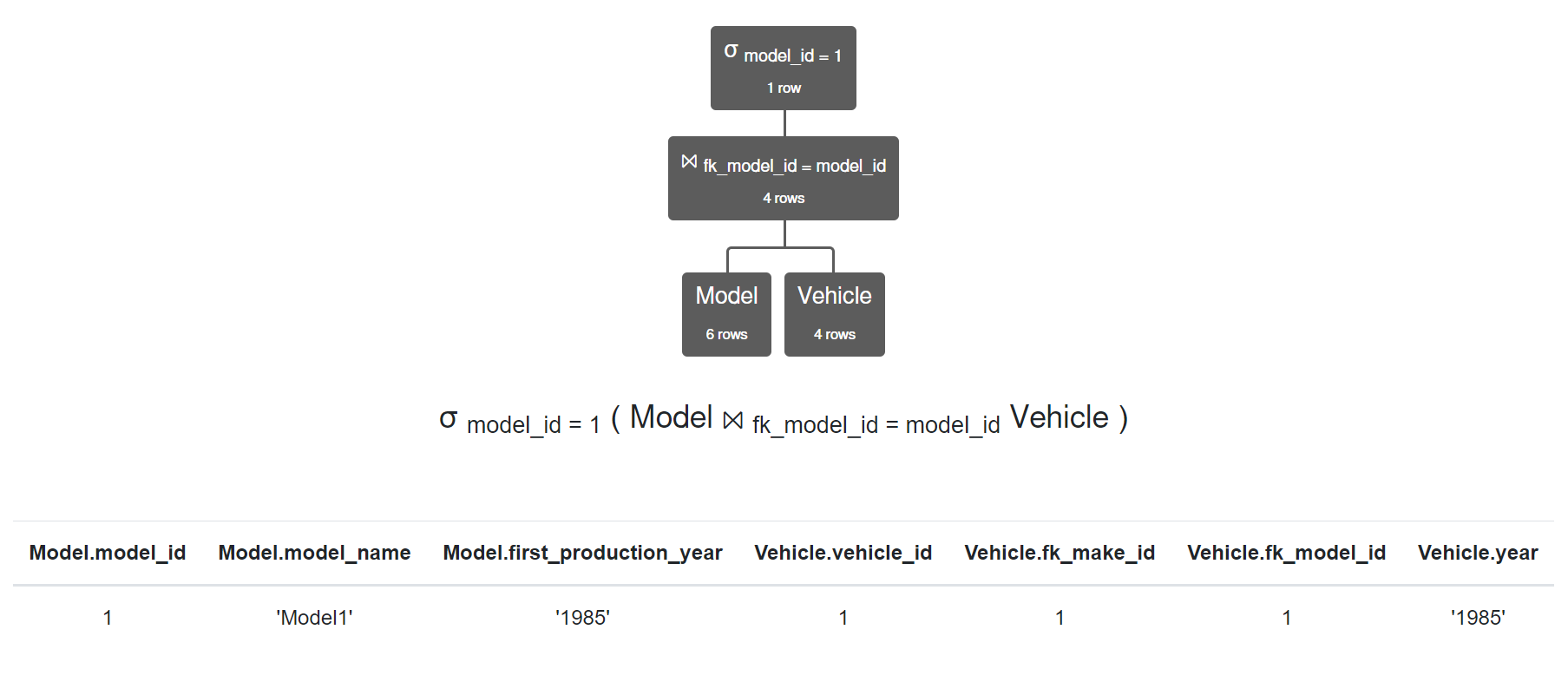
**Relational Algebra**

π model\_id Model - π fk\_model\_id Vehicle



**SQL Query**

SELECT \* FROM Model INNER JOIN Vehicle ON fk\_model\_id = model\_id WHERE model\_id = 1;

****

***Explanation:*** The Join operator shows all column from two tables that have a common relation, in this case model\_id and fk\_model\_id where model\_id is equal to 1 meaning that fk\_model\_id has to be 1.