DESCRIPTION

Table: Customer

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| customer_id | int |

| product_key | int |

+-------------+---------+

This table may contain duplicates rows.

customer_id is not NULL.

product_key is a foreign key (reference column) to Product table.

Table: Product

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| product_key | int |

+-------------+---------+

product_key is the primary key (column with unique values) for this table.

Write a solution to report the customer ids from the Customer table that bought all the products in the Product table.

Return the result table in **any order**.

The result format is in the following example.

Example 1:

Input:

Customer table:

+-------------+-------------+

| customer_id | product_key |

+-------------+-------------+

| 1 | 5 |

| 2 | 6 |

| 3 | 5 |

| 3 | 6 |

| 1 | 6 |

+-------------+-------------+

Product table:

+-------------+

| product_key |

+-------------+

| 5 |

| 6 |

+-------------+

Output:

+-------------+

| customer_id |

+-------------+

| 1 |

| 3 |

+-------------+

Explanation:

The customers who bought all the products (5 and 6) are customers with IDs 1 and 3.

SOLUTION

MySQL:

- In a subquery, select `the number of product_key using COUNT`

- For the customers who bought all the products, select customer_id who has the same number of product_key (from the above subquery) using HAVING, COUNT and DISTINCT

```
SELECT customer_id
FROM Customer
GROUP BY customer_id
HAVING COUNT(DISTINCT product_key) = (SELECT COUNT(product_key) FROM Product);
```

PostgreSQL:

- Same approach as above

```
SELECT customer_id
FROM Customer
GROUP BY 1
HAVING COUNT(DISTINCT product_key) = (SELECT COUNT(*) FROM Product);
```