

DESCRIPTION

Table: Products

+-----+-----+		
Column Name	Type	
+-----+-----+		
product_id	int	
new_price	int	
change_date	date	
+-----+-----+		

(product_id, change_date) is the primary key (combination of columns with unique values) of this table.

Each row of this table indicates that the price of some product was changed to a new price at some date.

Initially, all products have price 10.

Write a solution to find the prices of all products on the date 2019-08-16.

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

The result format is in the following example.

Example 1:

Input:

Products table:

+-----+-----+-----+		
product_id	new_price	change_date
+-----+-----+-----+		
1	20	2019-08-14
2	50	2019-08-14
1	30	2019-08-15
1	35	2019-08-16
2	65	2019-08-17
3	20	2019-08-18
+-----+-----+-----+		

Output:

```

+-----+-----+
| product_id | price |
+-----+-----+
| 2          | 50    |
| 1          | 35    |
| 3          | 10    |
+-----+-----+

```

SOLUTION**MySQL:**

- Using WITH t1, select product_id with most recent date (**MAX**(change_date)) where change_date <= "2019-08-16"
- Select product_id, new_price where product_id, change_date are in t1
- Select product_id, 10 as price which are not in T1
- Union the above two tables to output the desired table

```

WITH t1 AS (SELECT product_id, MAX(change_date) recent_date
FROM Products
WHERE change_date <= "2019-08-16"
GROUP BY product_id)
SELECT product_id, new_price price
FROM Products
WHERE (product_id, change_date) IN (SELECT * FROM t1)
UNION
SELECT product_id, 10 price
FROM Products
WHERE product_id NOT IN(SELECT product_id FROM t1);

```

PostgreSQL:

- Using WITH t1, select product_id with most recent date (**MAX**(change_date)) where change_date <= "2019-08-16"
- Using WITH t2, select product_id, new_price by joining Products and t1
- Select product_id, 10 as price which are not in T1
- Union t2 and t3 to output the desired table

```

WITH t1 AS(
  SELECT product_id, MAX(change_date) change_date
  FROM Products
  WHERE change_date <= '2019-08-16'

```

```
        GROUP BY 1),
t2 AS(
    SELECT t1.product_id, p.new_price price
    FROM t1
    LEFT JOIN Products p
    ON p.product_id = t1.product_id AND p.change_date = t1.change_date),
t3 AS(
    SELECT product_id, 10 price
    FROM Products
    WHERE product_id NOT IN (SELECT product_id FROM t1))
SELECT *
FROM t2
UNION
SELECT *
FROM t3;
```