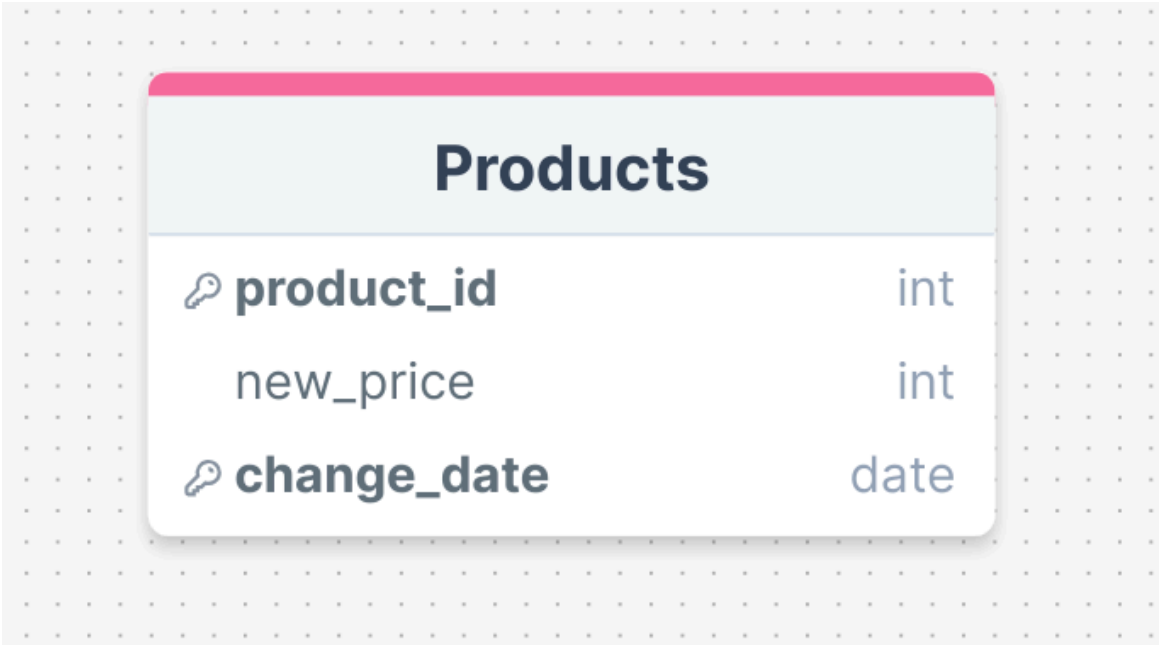
SQL CODE
WRITE-UP

PRODUCT PRICE AT A GIVEN DATE

APR 2024	
Advanced Select, Joins & Clauses	Each row of this table indicates that the price of some product was changed to a new price at some date. Write a solution to find the prices of all products on 2019-08-16. Assume the price of all products before any change is 10. Return the result table in **any order**.
Leetcode	
Advanced	

INPUT FORMAT

The main source tables IS **PRODUCTS.**



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

| 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 |

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

CODE SOLUTION

```
SELECT DISTINCT

p.product_id,

COALESCE(upd.new_price,10) AS price

FROM products p

LEFT JOIN

    (SELECT * FROM

    products p

    WHERE (product_id, change_date) IN

        (

        SELECT

        product_id,

        MAX(change_date) AS mdate

        FROM products

        WHERE change_date <= '2019-08-16'

        GROUP BY 1

        ORDER BY 1

        )

    ) AS upd

ON upd.product_id = p.product_id

ORDER BY 1
```

SOLUTION PROCESS

- Innermost Select statement: Identifies product IDs whose change occurs before specified by using aggregate function to extract most recent while defining parameters using a WHERE clause
- Mid-select statement AS upd: Extracting records from original table that meet criteria of nested selecting all fields using IS IN clause to ensure that no aggregate fields are present for ease of analysis
- Outermost Select statement: LEFT JOIN used to identify products that did not undergo price change and therefore remain at set price 10. Null values resolved COALESCE function within select statement and duplicates resolved using DISTINCT.

OUTPUT

product_id	price
1	35
2	50
3	10