

**Задача:** реализовать функционал (функция, класс, еще что либо) позволяющий на вход передать набор данных в формате 1, а на выходе получить расширенный набор данных в формате 2.

**Входящие данные (формат 1)** - описание цены товара в зависимости даты создания заказа и выбранной даты доставки. На вход должны приходить одна или несколько цен для одного или нескольких товаров. Одна цена содержит следующие данные:

position\_id - идентификатор товара  
order\_date\_from - дата заказа с которой актуальна цена  
delivery\_date\_from - дата доставки с которой актуальна цена  
price - цена

**Пример (пример 1) входящих данных:**

position_id	order_date_from	delivery_date_from	price
1	2019-02-01	2019-03-01	100
1	2019-02-10	2019-03-10	200
1	2019-02-20	2019-02-25	130

**Исходящие данные (формат 2)** денормализованные данные для более удобного поиска цены по ним. Одна цена на выходе содержит следующие данные:

position\_id - идентификатор товара  
order\_date\_from - дата заказа с которой актуальна цена  
order\_date\_to - дата заказа до которой актуальна цена  
delivery\_date\_from - дата доставки с которой актуальна цена  
delivery\_date\_to - дата доставки до которой актуальна цена  
price - цена

**Пример исходящих данных (такие данные должны получиться для входных данных из примера 1):**

position_id	order_date_from	order_date_to	delivery_date_from	delivery_date_to	price
1	2019-02-01	2019-02-09	2019-03-10		100
1	2019-02-01		2019-03-01	2019-03-09	100
1	2019-02-10		2019-03-10		200
1	2019-02-20		2019-02-25	2019-02-28	130

Предполагается что и входящие данные и исходящие данные могут быть использованы для определения цены товара. При поиске цены товара у нас есть следующие входящие данные:

position\_id - идентификатор товара  
order\_date - дата заказа товара  
delivery\_date - дата доставки

Выбор цены для входящего и исходящего набора данных должен давать одинаковый результат. Это условие правильно сформированного исходящего набора данных.

Принцип выбора цены для товара по входящему набору данных:

1. выбираем цену с ближайшим delivery\_date\_from (меньшим чем или равным delivery\_date товара) при этом order\_date\_from должен быть меньше или равен order\_date заказа
2. из полученных цен выбираем ближайшую по order\_date\_from (меньшим чем или равным order\_date товара)

**Пример:** для вариантов (order\_date, delivery\_date) и предложенных в примере 1 данных должны выбираться следующие цены:

2019-02-05	2019-03-15	100
2019-02-15	2019-03-15	200
2019-02-25	2019-03-15	200
2019-02-25	2019-03-05	100
2019-02-25	2019-02-28	130

Принцип выбора цены для товара по исходящему набору данных:

1. выбираем цену у которой delivery\_date\_from <= delivery\_date <= delivery\_date\_to и order\_date\_from <= order\_date <= order\_date\_to. На выходе должна получиться только одна цена. Если цен несколько то преобразование работает не верно.

**Условия для корректно сформированных исходящих данных:**

1. для пары order\_date, delivery\_date должна быть **только одна** запись при выборке по интервалам order\_date\_from - order\_date\_to и delivery\_date\_from - delivery\_date\_to соответственно
2. результат должен соответствовать тому что давал изначальный алгоритм выборки

Пожелания:

1. это должно работать
2. понятность кода
3. использование ООП
4. без лишнего кода