

```

use App\Models\Backend\MstOrder;
use App\Models\Backend\Log;

class ProcessOrderCustomer
{
    /**
     * Process order
     * @return void
     */
    public function processOrder($fromDate, $toDate, $maxProfitRateNew) {

        $modelOrder = new MstOrder;

        //return 100,000 records
        $dataOrder = $modelOrder->getOrder($fromDate, $toDate);

        foreach ($dataOrder as $order) {
            $modelLog = new Log; $priceInvoice = $order->price_invoice;
            $maxProfitRateOld = $order->max_profit_rate;
            if ($maxProfitRateNew != $maxProfitRateOld)
            {
                $maxProfitPrice = $priceInvoice / (1 - $maxProfitRateNew);
                $modelOrder->update($order, $maxProfitPrice);
                $modelLog->insert($order, $maxProfitPrice);
            }
        }
    }
}

```

- Nếu ta để processOrder() ở trong vòng lặp, trong mỗi lần lặp sẽ gây lãng phí tài nguyên và giảm hiệu suất.
- Hàm getOrder() trả về 100.000 bản ghi cùng lúc nó sẽ làm chậm vì trả về quá nhiều record thay vào đó ta có thể chia nhỏ để tăng hiệu suất.
- Trong quá trình update và thêm có thể gây lỗi, vì vậy ta có thể thêm try catch để tránh các lỗi trong lúc chỉnh sửa dữ liệu.

Sau khi xử lý code:

```

<?php
use App\Models\Backend\MstOrder;
use App\Models\Backend\Log;

class ProcessOrderCustomer
{
    /**
     * Process order
     * @return void
     */
    public function processOrder($fromDate, $toDate, $maxProfitRateNew)
    {
        $modelOrder = new MstOrder;
        $modelLog = new Log; // Tạo model Log bên ngoài vòng lặp

        // Xử lý dữ liệu từng phần nhỏ
        // Xử lý 1000 bản ghi mỗi lần
        $chunkSize = 1000;
        $offset = 0;

        $dataOrder = $modelOrder->getOrder($fromDate, $toDate, $offset, $chunkSize);

        foreach ($dataOrder as $order) {
            try {
                $priceInvoice = $order->price_invoice;
                $maxProfitRateOld = $order->max_profit_rate;
                if ($maxProfitRateNew != $maxProfitRateOld) {
                    $maxProfitPrice = $priceInvoice / (1 - $maxProfitRateNew);
                    $modelOrder->update($order, $maxProfitPrice);
                    $modelLog->insert($order, $maxProfitPrice);
                }
            } catch (Exception $e) {
                // Xử lý lỗi
            }
        }
    }
}

```