



SOFTWARESILOS

Customer Specific Prices

PDF Manual

MODULE

customerprice

LAST UPDATED

2026-03-12

Time-Based Pricing

Customer Specific Prices · /customerprice/features/time-based-pricing

Overview

Configure temporary customer prices with precise date ranges. Schedule promotional prices, seasonal adjustments, or contract renewals in advance that automatically activate and deactivate.

How It Works

Time-based pricing uses two optional date fields:

- **from_date**: Price becomes active on this date (inclusive)
- **to_date**: Price expires after this date (inclusive)

The system checks these dates during price resolution and only applies prices where the current date falls within the specified range.

Date Validation Logic

```
Current Date: 2025-02-15
```

```
Price Entry 1: from_date=NULL, to_date=NULL
```

```
→ Always active
```

```
Price Entry 2: from_date=2025-01-01, to_date=NULL
```

```
→ Active (started Jan 1, no end date)
```

```
Price Entry 3: from_date=2025-03-01, to_date=2025-03-31
```

```
→ Not yet active (starts March 1)
```

```
Price Entry 4: from_date=2025-01-01, to_date=2025-01-31
```

```
→ Expired (ended Jan 31)
```

```
Price Entry 5: from_date=2025-02-01, to_date=2025-02-28
```

```
→ Active (within range)
```

Use Cases

Promotional Campaign

Scenario: Q1 2025 promotion with special pricing

```
Customer: ACME Corp
```

```
Product: Widget ABC (SKU: WGT-ABC)
```

```
Quantity: 1
```

```
Price: 85.00
```

```
From Date: 2025-01-01
```

To Date: 2025-03-31

- Before Jan 1: Standard price applies
- Jan 1 - Mar 31: \$85.00 promotional price applies
- After Mar 31: Returns to standard price

Seasonal Pricing

Example: Holiday season pricing

```
Winter Season 2025:  
- Customer: Beta Ltd  
- Product: Widget ABC  
- Quantity: 1  
- Price: 95.00  
- From Date: 2025-12-01  
- To Date: 2026-02-28
```

```
Spring Season 2026:  
- Customer: Beta Ltd  
- Product: Widget ABC  
- Quantity: 1  
- Price: 90.00  
- From Date: 2026-03-01  
- To Date: 2026-05-31
```

Contract Renewal

Scenario: Transitioning from old to new contract pricing

```
Old Contract (expiring):  
- Price: 100.00  
- From Date: 2024-01-01  
- To Date: 2024-12-31
```

```
New Contract (starting):  
- Price: 95.00  
- From Date: 2025-01-01  
- To Date: 2025-12-31
```

Admin Setup

Step 1: Navigate to Customer Prices

Go to: **Catalog > Customer Prices > Add New**

Step 2: Fill Basic Fields

```
Customer: Select customer  
Product: Select product
```

Quantity: 1
Price: 85.00

Step 3: Set Date Range

From Date:

- Click calendar icon
- Select start date: 2025-01-01
- Format: YYYY-MM-DD

To Date:

- Click calendar icon
- Select end date: 2025-03-31
- Format: YYYY-MM-DD

Step 4: Save

Click **Save** to schedule the price.

Advanced Scenarios

Multiple Time-Based Tiers

Combine time-based pricing with quantity tiers:

```
Q1 2025 Promotional Tiers:  
Entry 1: qty=1, price=$90, from=2025-01-01, to=2025-03-31  
Entry 2: qty=10, price=$85, from=2025-01-01, to=2025-03-31  
Entry 3: qty=50, price=$80, from=2025-01-01, to=2025-03-31  
  
Q2 2025 Standard Tiers:  
Entry 4: qty=1, price=$95, from=2025-04-01, to=2025-06-30  
Entry 5: qty=10, price=$90, from=2025-04-01, to=2025-06-30  
Entry 6: qty=50, price=$85, from=2025-04-01, to=2025-06-30
```

Permanent Price with Temporary Override

```
Permanent Price:  
- qty=1, price=$100, from_date=NULL, to_date=NULL  
  
Summer Sale Override:  
- qty=1, price=$80, from_date=2025-07-01, to_date=2025-08-31
```

During July and August, the \$80 price applies. Outside that period, \$100 applies.

Overlapping Dates

Avoid overlapping date ranges for the same customer + product + quantity combination.

Bad Example:

```
Entry 1: qty=1, price=$90, from=2025-01-01, to=2025-06-30
Entry 2: qty=1, price=$85, from=2025-03-01, to=2025-09-30
```

Problem: March-June overlap creates ambiguity.

Good Example:

```
Entry 1: qty=1, price=$90, from=2025-01-01, to=2025-02-28
Entry 2: qty=1, price=$85, from=2025-03-01, to=2025-09-30
```

Best Practices

Planning

Do:

- Schedule prices in advance
- Use full date ranges (start and end)
- Document campaigns in customer contracts
- Set reminders for upcoming price changes
- Verify dates in customer timezone

Don't:

- Create overlapping date ranges
- Use ambiguous dates (e.g., to_date without from_date)
- Forget to set end dates for temporary campaigns
- Schedule dates in the past for new entries

Date Management

Open-Ended Pricing

```
from_date=2025-01-01, to_date=NULL
```

Starts on date, never expires. Good for permanent new pricing.

Campaign Pricing

```
from_date=2025-01-01, to_date=2025-03-31
```

Fixed duration. Good for promotions and seasonal pricing.

Historical Pricing

```
from_date=2024-01-01, to_date=2024-12-31
```

Past dates. Keep for audit trail and reporting.

Database Structure

Schema

```
CREATE TABLE pricesystem_customerprice (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  entity_id INT NOT NULL,  
  customer_id INT NOT NULL,  
  qty DECIMAL(12,4) NOT NULL,  
  value DECIMAL(12,4) NOT NULL,  
  from_date DATE NULL,  
  to_date DATE NULL,  
  website_id INT NOT NULL DEFAULT 0,  
  CONSTRAINT UNIQUE (entity_id, customer_id, qty, from_date, to_date,  
  website_id)  
);
```

Unique Constraint

The combination must be unique:

- entity_id + customer_id + qty + from_date + to_date + website_id

This means you can have:

- Same product/customer/qty with different dates
- Same product/customer/dates with different qty
- Exact duplicate of all six fields

Frontend Display

Product Page

Shows available time-based pricing:

Current price: \$100.00

Special Offers:

- From Jan 1, 2025: \$85.00
- Expires: Mar 31, 2025

Cart

Active time-based price is applied:

Widget ABC
Quantity: 1
Unit Price: \$85.00 (Q1 2025 Promotion)

Customer Account

"Download My Prices" CSV includes date ranges:

```
Product,SKU,Quantity,Price,From Date,To Date
Widget ABC,WGT-ABC,1,85.00,2025-01-01,2025-03-31
Widget ABC,WGT-ABC,1,100.00,,
```

Troubleshooting

Price Not Activating

Check:

1. Current date is \geq from_date
2. Current date is \leq to_date (if set)
3. Date format is correct (YYYY-MM-DD)
4. Server timezone matches expected timezone
5. Cache has been cleared

Debug Query:

```
SELECT * FROM pricesystem_customerprice
WHERE entity_id = 123
      AND customer_id = 456
      AND (from_date IS NULL OR from_date <= CURDATE())
      AND (to_date IS NULL OR to_date >= CURDATE());
```

Price Not Expiring

Verify:

1. to_date is set correctly
2. to_date is in the past
3. No other active price entry exists
4. Cache has been cleared

Wrong Price Applied

Common Causes:

1. Multiple entries with overlapping dates
2. Quantity tier overriding time-based price
3. Website filter excluding current store
4. Cache showing old price

Solution: Review all entries for customer + product:

```
SELECT qty, value, from_date, to_date, website_id
FROM pricesystem_customerprice
WHERE entity_id = 123 AND customer_id = 456
```

```
ORDER BY from_date DESC, qty ASC;
```

Combining Features

Time-Based + Quantity Tiers + Multi-Website

```
US Website (website_id=1), Q1 2025:  
qty=1, price=$90, from=2025-01-01, to=2025-03-31, website_id=1  
qty=50, price=$85, from=2025-01-01, to=2025-03-31, website_id=1  
  
EU Website (website_id=2), Q1 2025:  
qty=1, price=€80, from=2025-01-01, to=2025-03-31, website_id=2  
qty=50, price=€75, from=2025-01-01, to=2025-03-31, website_id=2
```

Reporting

Active Campaigns Query

```
SELECT  
  c.email as customer,  
  p.sku,  
  cp.qty,  
  cp.value,  
  cp.from_date,  
  cp.to_date  
FROM pricesystem_customerprice cp  
JOIN customer_entity c ON cp.customer_id = c.entity_id  
JOIN catalog_product_entity p ON cp.entity_id = p.entity_id  
WHERE cp.from_date <= CURDATE()  
  AND (cp.to_date IS NULL OR cp.to_date >= CURDATE())  
ORDER BY cp.to_date ASC;
```

Upcoming Campaigns Query

```
SELECT * FROM pricesystem_customerprice  
WHERE from_date > CURDATE()  
ORDER BY from_date ASC;
```

Expired Campaigns Query

```
SELECT * FROM pricesystem_customerprice  
WHERE to_date < CURDATE()  
ORDER BY to_date DESC  
LIMIT 100;
```

Next Steps

- [Quantity-Based Pricing Tiers](#)
- [Multi-Website Configuration](#)
- [Bulk Import with Add-On](#)