



SOFTWARESILOS

Category Specific Prices

PDF Manual

MODULE

categoryprice

LAST UPDATED

2026-03-29

Category Customer Pricing

Category Specific Prices · /categoryprice/features/category-customer-pricing

Set custom prices at the category level for individual customers. One price entry affects all products in the category.

Overview

Category Customer Pricing allows you to assign specific prices to individual customers for entire product categories. This is the most granular level of category pricing control.

Key Characteristics

- **Individual customer targeting** - One customer, one category
 - **Database table:** `pricesystem_categoryprice`
 - **Foreign keys:** `customer_id` → `customer_entity`, `entity_id` → `catalog_category_entity`
 - **Unique constraint:** category + customer + qty + website + dates + application type
 - **Priority system:** Conflicts resolved by priority field or global rule
-

When to Use Customer Pricing

Use Case 1: Contract-Based Pricing

Scenario: Customer signed annual contract for specific pricing on office supplies.

Solution:

- Category: Office Supplies
- Customer: Customer #12345
- Price: \$19.99 (contracted price)
- From Date: 2025-01-01
- To Date: 2025-12-31
- Priority: 30

All office supply products honor the contract price for this customer.

Use Case 2: VIP Customer Rewards

Scenario: Top customer gets exclusive 20% discount on Electronics.

Solution:

- Category: Electronics
- Customer: VIP Customer #7890
- Price Application Type: Discount
- Price: 20 (20% off)
- Priority: 35

The discount applies to all existing and future electronics products.

Use Case 3: Loyalty Program

Scenario: Customer reached Gold tier, gets special pricing on Premium category.

Solution:

- Category: Premium Products
- Customer: Gold Member #4567
- Price: Reduced by 15%
- Priority: 25

Automatically applies when customer browses premium products.

Use Case 4: Make-Good Pricing

Scenario: Customer had service issue, compensated with reduced pricing for 3 months.

Solution:

- Category: All categories (create multiple entries)
- Customer: Affected Customer #3456
- Price Application Type: Discount
- Price: 10 (10% off everything)
- From Date: 2025-03-01
- To Date: 2025-05-31
- Priority: 20

Temporary discount expires automatically.

Database Structure

Table: pricesystem_categoryprice

Schema:

```
CREATE TABLE pricesystem_categoryprice (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  entity_id INT NOT NULL,           -- Category ID  
  customer_id INT NOT NULL,         -- Customer ID  
  qty DECIMAL(12,4) NOT NULL,       -- Quantity tier  
  value DECIMAL(12,4) NOT NULL,     -- Price value  
  priority INT DEFAULT 0,           -- Priority for conflict resolution  
  from_date DATE DEFAULT NULL,      -- Start date (NULL = immediate)  
  to_date DATE DEFAULT NULL,        -- End date (NULL = permanent)  
  website_id INT NOT NULL,          -- Website scope  
  price_application_type VARCHAR(50), -- fixed|discount|adjustment  
  created_at TIMESTAMP,  
  updated_at TIMESTAMP,
```

```
FOREIGN KEY (entity_id) REFERENCES catalog_category_entity(entity_id) ON
DELETE CASCADE,
FOREIGN KEY (customer_id) REFERENCES customer_entity(entity_id) ON DELETE
CASCADE,
FOREIGN KEY (website_id) REFERENCES store_website(website_id) ON DELETE
CASCADE,

UNIQUE KEY (entity_id, customer_id, qty, website_id, from_date, to_date,
price_application_type)
);
```

Key Points:

- `entity_id` is the category, not the product
- `customer_id` is the individual customer
- `qty` enables quantity tiers (1, 10, 50, 100, etc.)
- `priority` field for conflict resolution
- Date fields can be NULL for permanent pricing
- Unique constraint prevents duplicate entries

Admin Interface

Location

Admin Panel > Catalog > Category Prices > Customer Prices

Grid Columns

- **ID** - Database record ID
- **Category** - Category name with breadcrumb
- **Customer** - Customer name and email
- **Qty** - Minimum quantity for this tier
- **Price** - Custom price value
- **Priority** - Conflict resolution priority (0-999)
- **From Date** - Start date (empty = immediate)
- **To Date** - End date (empty = permanent)
- **Website** - Website scope
- **Created** - Record creation timestamp
- **Actions** - Edit/Delete buttons

Grid Features

Filtering:

- Search by customer name/email
- Filter by category
- Filter by date range
- Filter by website
- Filter by priority

Sorting:

- Sort by any column
- Multi-column sorting
- Ascending/descending

Mass Actions:

- Delete selected
- Export to CSV (with add-on)

Pagination:

- 20/50/100/200 per page
 - Jump to page number
-

Creating Customer Prices

Standard Entry Form

Fields:**Category (Required):**

- Type: Category selector (tree)
- Validation: Must exist, must be active
- Example: Electronics > Laptops

Customer (Required):

- Type: Autocomplete search
- Search by: Name, email, customer ID
- Validation: Must exist, must be active
- Example: "john.doe@example.com"

Quantity (Required):

- Type: Decimal (12,4)
- Default: 1.0000
- Min: 0.0001
- Example: 1 (first tier), 10 (bulk tier)

Price (Required):

- Type: Decimal (12,4)
- Validation: Must be positive
- Example: 99.99

Priority (Optional):

- Type: Integer (0-999)
- Default: 0
- Higher = takes precedence

- Example: 30 for VIP pricing

From Date (Optional):

- Type: Date picker
- Format: YYYY-MM-DD
- Default: NULL (immediate)
- Example: 2025-01-01

To Date (Optional):

- Type: Date picker
- Must be $\geq$ From Date
- Default: NULL (permanent)
- Example: 2025-12-31

Website (Required):

- Type: Dropdown
- Options: All websites or specific website
- Default: Default website
- Example: "US Website"

Price Application Type (Required):

- Type: Dropdown
- Options:
 - **fixed** - Absolute price (default)
 - **discount** - Percentage discount
 - **adjustment** - Add/subtract amount
- Example: fixed

Quantity Tiers

Creating Tiers

Create multiple entries for same customer + category with different qty values:

Example: Volume Discount for Customer #12345

Category	Customer	Qty	Price	Priority
Electronics	#12345	1	\$100.00	10
Electronics	#12345	10	\$95.00	10
Electronics	#12345	50	\$90.00	10
Electronics	#12345	100	\$85.00	10

Resolution Logic:

- Customer orders 5 items → \$100 each (Qty 1 tier)
- Customer orders 25 items → \$95 each (Qty 10 tier)

- Customer orders 75 items → \$90 each (Qty 50 tier)
- Customer orders 150 items → \$85 each (Qty 100 tier)

Tier Selection Algorithm

1. Get all prices for customer + category + website
2. Filter by date validity (from_date <= today <= to_date)
3. Filter by qty (price.qty <= cart_qty)
4. Sort by qty DESC
5. Select first match (highest qty <= cart qty)

Best Practices

1. **Use same priority** for all tiers of same customer
 2. **Consistent date ranges** across tiers
 3. **Logical qty progression** (1, 10, 50, 100 - not random)
 4. **Price decreases with qty** (volume discount pattern)
 5. **Test edge cases** (qty exactly at breakpoint)
-

Date-Based Pricing

Permanent Pricing

Configuration:

- From Date: NULL
- To Date: NULL

Meaning: Price always active, never expires.

Use Case: Standard contract pricing.

Open-Ended Start Date

Configuration:

- From Date: 2025-06-01
- To Date: NULL

Meaning: Activates June 1st, never expires.

Use Case: New contract starting future date.

Fixed Date Range

Configuration:

- From Date: 2025-01-01
- To Date: 2025-12-31

Meaning: Active only during 2025.

Use Case: Annual contract renewal.

Past Dates

Configuration:

- From Date: 2024-01-01
- To Date: 2024-12-31

Meaning: Already expired, won't apply.

Use Case: Historical record keeping.

Priority System

Priority Values

Recommended Range: 0-100

Common Values:

- **0** - Default/fallback pricing
- **10** - Standard customer pricing
- **20** - Promotional customer pricing
- **30** - VIP customer pricing
- **40** - Override everything

Conflict Scenarios

Scenario 1: Customer vs Customer (Different Priorities)

Setup:

- Customer #123 + Electronics + Qty 1 + \$100 + Priority 10
- Customer #123 + Electronics + Qty 1 + \$90 + Priority 20

Result: \$90 wins (priority 20 > 10)

Why: Higher priority always wins when both match.

Scenario 2: Customer vs Group Price

Setup:

- Customer #123 + Electronics + Qty 1 + \$95 + Priority 15 (customer price)
- Wholesale Group + Electronics + Qty 1 + \$85 + Priority 25 (group price)
- Customer #123 is in Wholesale group

Result: Depends on configuration:

- **Select by priority:** \$85 (priority 25 > 15)
- **Customer first:** \$95 (ignores priority)
- **Group first:** \$85 (ignores priority)

Multi-Website Support

Website Scope

Each price entry is scoped to one website (or all websites).

Example Setup:

Website	Customer	Category	Price
US Website	#123	Electronics	\$100 USD
EU Website	#123	Electronics	€90 EUR
UK Website	#123	Electronics	£80 GBP

Same customer, same category, different prices per region.

All Websites Option

Set `website_id = 0` (or select "All Websites") to apply globally.

When to use:

- Single-website stores
- Global pricing strategy
- Testing/development

When NOT to use:

- Multi-currency stores
 - Regional pricing differences
 - Tax-inclusive vs exclusive regions
-

CSV Export for Customers

Configuration

Enable Download Link:

Navigate to: **Stores > Configuration > Pricesystem > Categoryprice Settings**

- **Enable Download** = Yes
- **CSV Delimiter** = , (or ; for European) (`pricesystem/categoryprice/csv_delimiter`)
- **CSV Encloser** = " (double quotes) (`pricesystem/categoryprice/csv_encloser`)

Customer Access

Location: Customer Dashboard > My Prices

The download button text depends on your theme and configuration.

CSV Format

```
name,sku,qty,price,price_type
"Widget Pro",SKU-123,1,99.99,categoryprice
"Widget Pro",SKU-123,10,95.00,categoryprice
```

Use Cases

- Customer tracking their agreements
 - Customer planning bulk orders
 - Customer comparing prices across categories
 - Customer verifying contract terms
-

API Integration

Category customer prices can be managed via WebAPI if you install the **SOAP / REST API Add-On**:

- [Categoryprice SOAP / REST API Add-On](#)
- [Pricesystem SOAP / REST API \(full endpoint list\)](#)

Core endpoints use the prefix `/V1/pricesystem/...`. For category customer prices the key endpoints are:

- GET `/V1/pricesystem/categoryprice/search`
 - POST `/V1/pricesystem/categoryprice`
 - PUT `/V1/pricesystem/categoryprice/:id`
 - DELETE `/V1/pricesystem/categoryprice/:id`
-

Import/Export

Bulk import/export is available via the **CSV Import/Export Add-On**:

- [Categoryprice CSV Import/Export Add-On](#)
- [Pricesystem CSV Import/Export \(overview\)](#)

In Magento's importer (System > Data Transfer > Import) the entity type is:

- **Pricesystem Categoryprice**
-

Best Practices

Naming Conventions

Use consistent priority values across your organization:

- 0-9: System defaults
- 10-19: Standard customer pricing
- 20-29: Promotional pricing
- 30-39: VIP pricing
- 40-49: Contract overrides
- 50+: Emergency overrides

Documentation

Document each price entry:

- Why was it created?
- Who approved it?
- When does it expire?
- What's the renewal process?

Keep external documentation linked to customer records.

Regular Audits

Schedule quarterly reviews:

1. Expired prices (to_date < today)
2. Unused prices (no orders)
3. Conflicting prices (same customer+category+qty with different priorities)
4. Orphaned prices (customer deleted, category deleted)

Performance

For large catalogs (10,000+ price entries):

1. Index regularly
2. Use caching
3. Optimize database queries
4. Consider archiving old prices

Troubleshooting

Price Not Applying

Checklist:

1. ✓ Customer logged in?
2. ✓ Customer ID matches?

3. ✓ Category matches?
4. ✓ Product in category?
5. ✓ Quantity meets minimum?
6. ✓ Date range valid?
7. ✓ Website matches?
8. ✓ Priority correctly set?
9. ✓ Cache cleared?

Debug Steps:

1. Check database directly: `SELECT * FROM pricesystem_categoryprice WHERE customer_id = X AND entity_id = Y;`
2. Check Magento logs: `var/log/system.log`, `var/log/exception.log`
3. Enable Pricesystem Core debug mode
4. Test with different customer
5. Test with different category

Wrong Price Applies

Possible Causes:

1. Multiple matching prices (check priority)
2. Group price overriding (check price select rule)
3. Product-level price overriding (check Pricesystem Core order)
4. Tier price on product (check product admin)
5. Special price on product (check product admin)
6. Catalog price rule (check Marketing > Promotions)

Resolution:

1. Review all prices for customer: Check both customer and group prices
2. Check priority configuration: Stores > Config > Pricesystem > Categoryprice
3. Review Pricesystem Core order: Check which price type has precedence
4. Clear cache: `php bin/magento cache:flush`

Can't Delete Price

Possible Causes:

1. Database foreign key constraint
2. Insufficient admin permissions
3. Price currently in use (active order)

Resolution:

1. Check admin ACL permissions
2. Check for active orders referencing this price
3. Archive instead of delete

Advanced Scenarios

Scenario 1: Customer Migration

Goal: Move customer prices from old customer ID to new customer ID.

SQL:

```
UPDATE pricesystem_categoryprice
SET customer_id = 456 -- new ID
WHERE customer_id = 123 -- old ID
AND entity_id = 789; -- specific category
```

Scenario 2: Category Merge

Goal: Category A merged into Category B, move all prices.

SQL:

```
UPDATE pricesystem_categoryprice
SET entity_id = 200 -- Category B
WHERE entity_id = 100 -- Category A
AND customer_id = 123;
```

Scenario 3: Bulk Priority Update

Goal: Increase priority for all VIP customer prices.

SQL:

```
UPDATE pricesystem_categoryprice
SET priority = priority + 10
WHERE customer_id IN (SELECT entity_id FROM customer_entity WHERE group_id
= 4);
-- Assuming group_id 4 is VIP
```

Scenario 4: Expire All Temp Prices

Goal: End all temporary promotional prices.

SQL:

```
UPDATE pricesystem_categoryprice
SET to_date = CURDATE()
WHERE priority = 20 -- promotional priority
AND to_date IS NULL; -- currently no end date
```

Related Documentation

- [Creating Category Prices](#)
- [Category Group Pricing](#)
- [Priority System](#)
- [Import/Export Guide](#)