Determine and manage catalog rules

Primary tables:

- `catalogrule` - dates from/to, conditions serialized, actions serialized, simple action, discount amount
- `catalogrule_website` - rule-website
- `catalogrule_customer_group` - rule-customer group

Index tables:

- `catalogrule_product` - time from/to, customer group, action operator/amount
- `catalogrule_product_price` - customer group, rule date, rule price, latest start date, earlier end date
- `catalogrule_group_website` - rule to customer group, website. Just unique rule/customer group/websites that currently exist in index table `catalogrule_product` for *today*.

Replica tables - used to fast switch new index data:

- `catalogrule_product_replica`
- `catalogrule_product_price_replica`
- `catalogrule_group_website_replica`

Apply rules

- *Cron job* 01:00 am invalidates `catalogrule_rule` indexer.
- Clicking "Apply rules", "Save and apply" in admin panel invalidates indexer as well.
- Cron job `indexer_reindex_all_invalid`, running every minute, will pick it up and call indexer
    - Magento\CatalogRule\Model\Indexer\AbstractIndexer::executeFull
    - Magento\CatalogRule\Model\Indexer\IndexBuilder::reindexFull

index builder.doReindexFull:

- clear tables `catalogrule_product`, `catalogrule_product_price`

- for every rule, reindex rule product. "one day, maybe today, maybe in a year, this rule will match these products"

    - this is regardless of dates, dates are simply written in table for later uage
    - get matching product IDs
    - insert into `catalogrule_product` (`_replica`) combinations: product/website/customer group
- reindex product prices `catalogrule_product_price` (`_replica`). For yesterday, today and tomorrow reads "planned" products, and executes price update starting from initial product.price attribute value. If multiple rules match in same date, they will all apply (unless stop flag).

    - dates = [yesterday; today; tomorrow]
    - for each website:
    - read ALL planned products from `catalogrule_product` from all rules, join `price` product attribute
        - \Magento\CatalogRule\Model\Indexer\RuleProductsSelectBuilder::build
    - programmatically filter dates (from/to)
    - calculates price

        - \Magento\CatalogRule\Model\Indexer\ProductPriceCalculator::calculate
        - to fixed/to percent/by fixed/by percent
    - groups by [date, product id, website id, customer group id]
    - persists to `catalogrule_product_price`
        - \Magento\CatalogRule\Model\Indexer\RuleProductPricesPersistor::execute
- reindex *rule relations* with *customer groups and websites* `catalogrule_group_website` (`_replica`)

    - `catalogrule_group_website` - distinct combinations of customer group and website matched for *today*
- switches tables from replica to active

Catalog rule price info

Magento\CatalogRule\Pricing\Price\CatalogRulePrice::getValue

- product.data('catalog_rule_price') can be preselected by collection from index. Used only by bundle and is deprecated.
    - \Magento\CatalogRule\Model\ResourceModel\Product\CollectionProcessor::addPriceData - join catalogrule_product_price
- Magento\CatalogRule\Model\ResourceModel\Rule::getRulePrice by (date, website, customer group, product)
- rule.getRulePrices
- loads directly from DB `catalogrule_product_price`

Identify how to implement catalog price rules.

Matching products - conditions and actions serialized, are implemented on foundation of abstract rules Magento\Rule\Model\AbstractModel:

- conditions - Magento\Rule\Model\Condition\AbstractCondition, combine
- actions - Magento\Rule\Model\Action\AbstractAction, collection

Admin form:

- \Magento\CatalogRule\Block\Adminhtml\Promo\Catalog\Edit\Tab\Conditions
- conditions renderer \Magento\Rule\Block\Conditions
- \Magento\Rule\Model\Condition\AbstractCondition::asHtmlRecursive:
    - TypeElementHtml
    - AttributeElementHtml
    - OperatorElementHtml
    - ValueElementHtml
    - RemoveLinkHtml
    - ChooserContainerHtml

Matching products:

- Magento\CatalogRule\Model\Rule::getMatchingProductIds
- get product collection
- for each product, run rule::callbackValidateProduct
- check if product matches conditions Magento\Rule\Model\Condition\Combine::validate

When would you use catalog price rules?

- set up discount 30% on Helloween that starts in a month
- for b2b website, discount 10% from all prices
- VIP customer group members get

How do they impact performance?

- one SQL query on every final price request
- reindexing seems not optimized, dates match in PHP and not SQL

How would you debug problems with catalog price rules?

- Check indexes `catalogrule_product` - find row by product id, ensure found - rule matched
- Ensure needed rule matched and unwanted rules didn’t match, check sort_order, check from_time and to_time, ensure correct
- check `catalogrule_product_price` contains product within ±1 day