Shopizer data model

SHOPIZER

Data model

Table of Contents

1

Shopizer data model

Shopizer's runtime data is maintained in a relational database. Runtime data exists for the configuration of the system and for the operation of the system. Metadata also exists in various files in Shopizer's project. Please see Shopizer file metadata for more details. This first section describes configuration data maintained in various database tables. Information is described

- Reference objects
- Store configuration
- User management
- Shipping configuration
- Payment configuration
- Tax configuration
- Product management
- Shopping cart
- Orders

Information is required for the configuration of the system and a store. When login in the system you are logged in the default store (DEFAULT) in which you need to configure *global system information* and *store details information*.

The following section describes basic information required by the system for being able to run adequately.

References objects

Reference objects

Reference objects are objects that are used in multiple objects as repeating reference important values necessary for the creation of a given entity

- *Country*

  - All country codes are defined in sm-core SchemaConstants

```
public static final String[] COUNTRY_ISO_CODE = { "AF","AX","AL","DZ",
"AS","AD","AO","AI","AQ","AG","AR","AM","AW","AU","AT","AZ","BS","BH",
"BD","BB","BY","BE","BZ","BJ","BM","BT","BO","BA","BW","BV","BR","IO",
"BN","BG","BF","BI","KH","CM","CA","CV","KY","CF","TD","CL","CN","CX",
"CC","CO","KM","CG","CD","CK","CR","CI","HR","CU","CY","CZ","DK","DJ",
"DM","DO","EC","EG","SV","GQ","ER","EE","ET","FK","FO","FJ","FI","FR",
"GF","PF","TF","GA","GM","GE","DE","GH","GI","GR","GL","GD","GP","GU",
"GT","GG","GN","GW","GY","HT","HM","VA","HN","HK","HU","IS","IN","ID",
"IR","IQ","IE","IM","IL","IT","JM","JP","JE","JO","KZ","KE","KI","KP",
"KR","KW","KG","LA","LV","LB","LS","LR","LY","LI","LT","LU","MO","MK",
"MG","MW","MY","MV","ML","MT","MH","MQ","MR","MU","YT","MX","FM","MD",
"MC","MN","ME","MS","MA","MZ","MM","NA","NR","NP","NL","AN","NC","NZ",
"NI","NE","NG","NU","NF","MP","NO","OM","PK","PW","PS","PA","PG","PY",
"PE","PH","PN","PL","PT","PR","QA","RE","RO","RU","RW","SH","KN","LC",
"PM","VC","WS","SM","ST","SA","SN","RS","SC","SL","SG","SK","SI","SB",
"SO","ZA","GS","ES","LK","SD","SR","SJ","SZ","SE","CH","SY","TW","TJ",
"TZ","TH","TL","TG","TK","TO","TT","TN","TR","TM","TC","TV","UG","UA",
"AE","GB","US","UM","UY","UZ","VU","VE","VN","VG","VI","WF","EH",
"YE","ZM","ZW" };
```

Other country if required can be added to this java class which will at schema creation create the entry. Country description are derived from java Locale class and then inserted to CountryDescription

- *Zone*

  - All zones are initially defined in sm-core – resources – zoneconfig.json and inserted in Zone table during initial data creation.

```
{
  "en":[
        {
                "zoneCode": "AL",
                "zoneName": "Alabama",
                "countryCode": "US"
        },
        {
                "zoneCode": "AK",
                "zoneName": "Alaska",
                "countryCode": "US"
        },
        {
                "zoneCode": "AS",
                "zoneName": "American Samoa",
                "countryCode": "US"
        },
…
```

Other zones to country association can be created to zoneconfig.json which will at schema creation create the new entry

- *Geozone*

    - Geozone allows grouping country under a unique geo locality (geozone). This can be used for instance for creating country included in European Union geozone. Data in this table needs to be created manually.

- *Currency*

    - Currency are created from a static list of currency code defined in SchemaConstants

```
static {
                CURRENCY_MAP.put("EUR", "Euro");
                CURRENCY_MAP.put("ALL", "Lek");
                CURRENCY_MAP.put("DZD", "Algerian Dinar");
                CURRENCY_MAP.put("USD", "US Dollar");
                CURRENCY_MAP.put("AOA", "Kwanza");
                CURRENCY_MAP.put("XCD", "East Caribbean Dollar");
                CURRENCY_MAP.put("ARS", "Argentine Peso");
                CURRENCY_MAP.put("AMD", "Armenian Dram");
                CURRENCY_MAP.put("AWG", "Aruban Florin");
        …
```

Final insertion is based on current java.util.Currency implementation which will proceed with creation if the Currency definition exists in the JDK used at runtime.

- *Language*

    - Supported languages are defined in SchemaConstants. Additionnal supported languages have to be created in SchemaConstants before the initial schema is created.

public static final String[] LANGUAGE_ISO_CODE = {"en", "fr", "ru"};

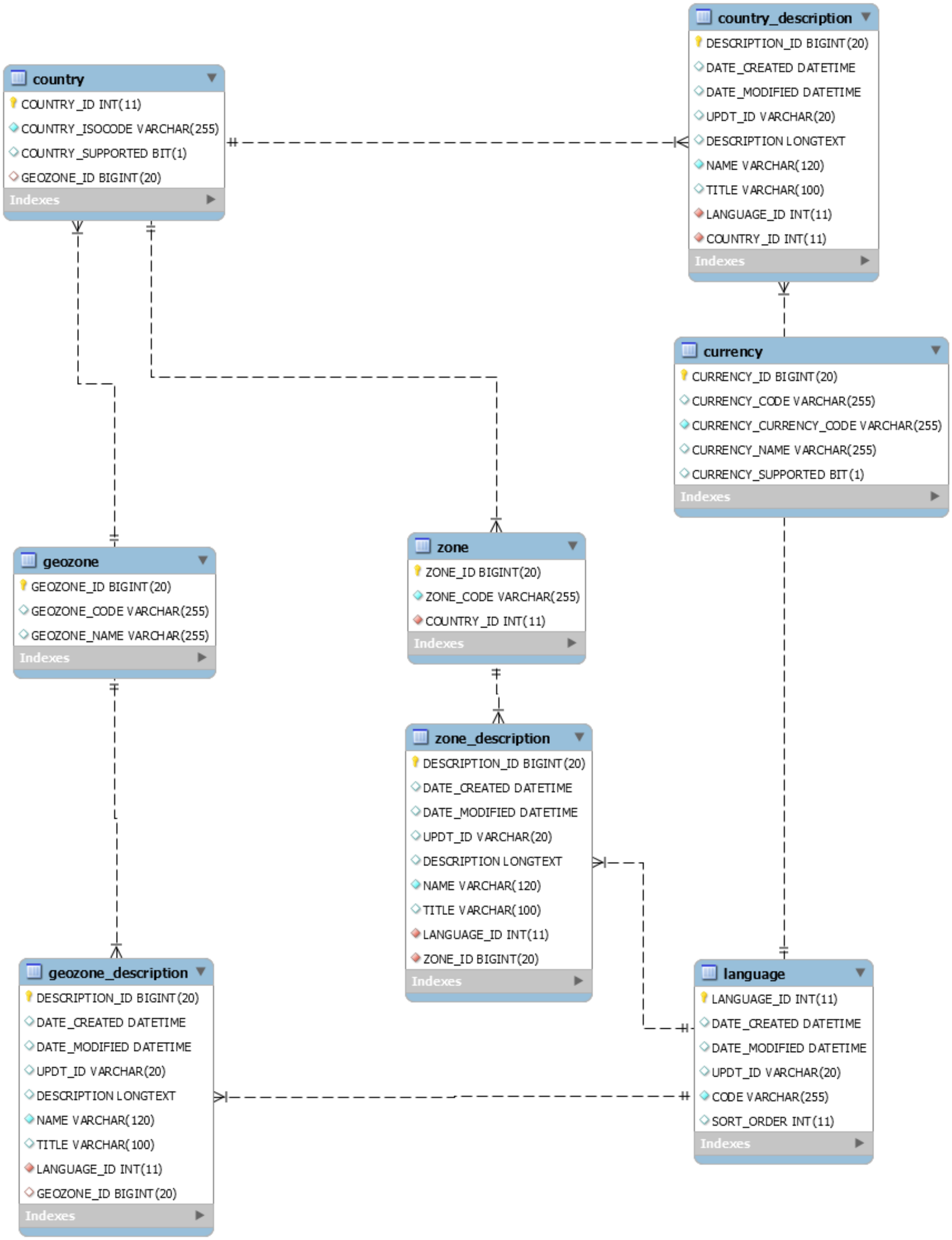


Figure 1 Reference objects

Country object has one to many CountryDescription children according to the number of languages supported globally in the system.

Country has 0 to multiple zones which represent states and provinces pertaining to this Country. Zone object has one to many ZoneDescription children according to the number of languages supported globally in the system.

Geozone is a group of country pert of a particular set of country, for example European Union.

Groups and permissions

Shopizer's security is implemented by Spring Security. See configurations section on the content of spring security files and annotations. Authorization is managed through Group and Permissions principle. Users and customers can be assigned one to multiple groups. Each group has a set of associated permissions defined in shopizer\sm-shop\src\main\resources\permission\permission.json. Permission file is inserted in PERMISSION table at startup. Each functionality, method and resource can be configured to be granted for specific permission or group. For instance and administrator will have access to all administration functionality, while a content manager will only be able to manage content in administration section and customer will be authorized when signin in to the customer portal.

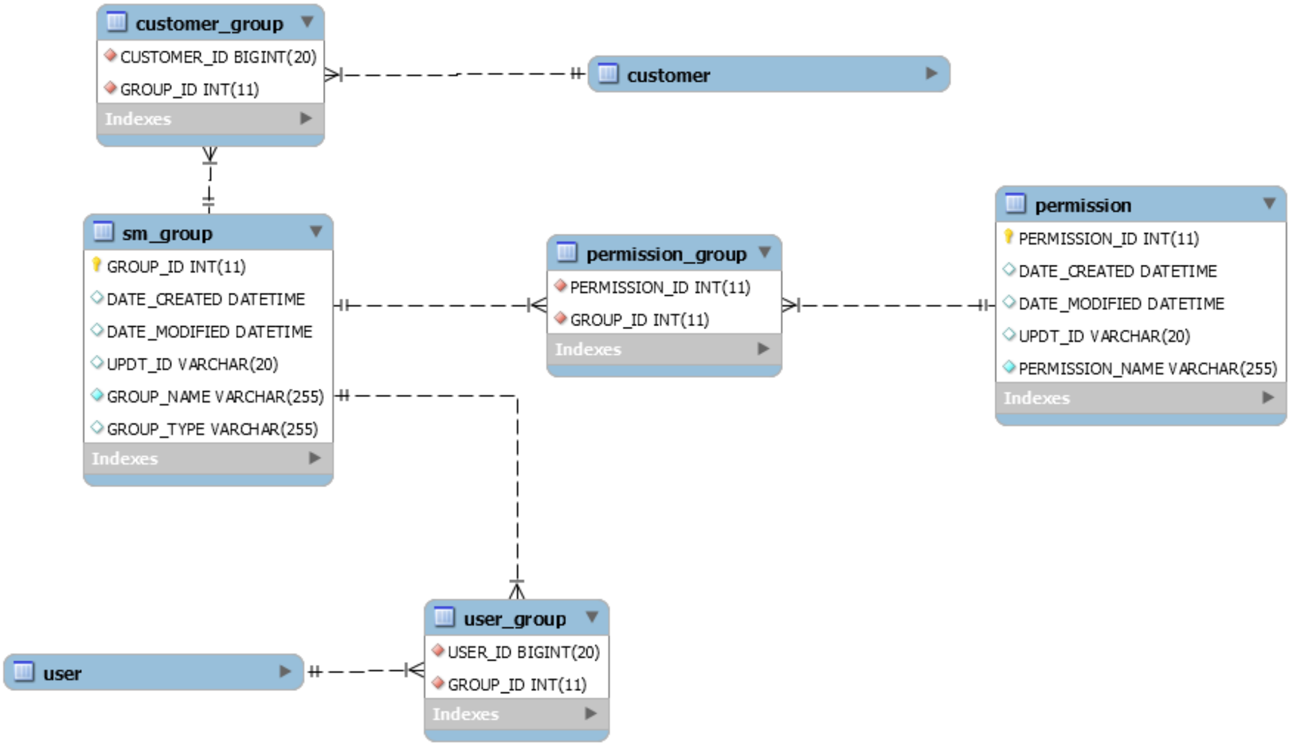


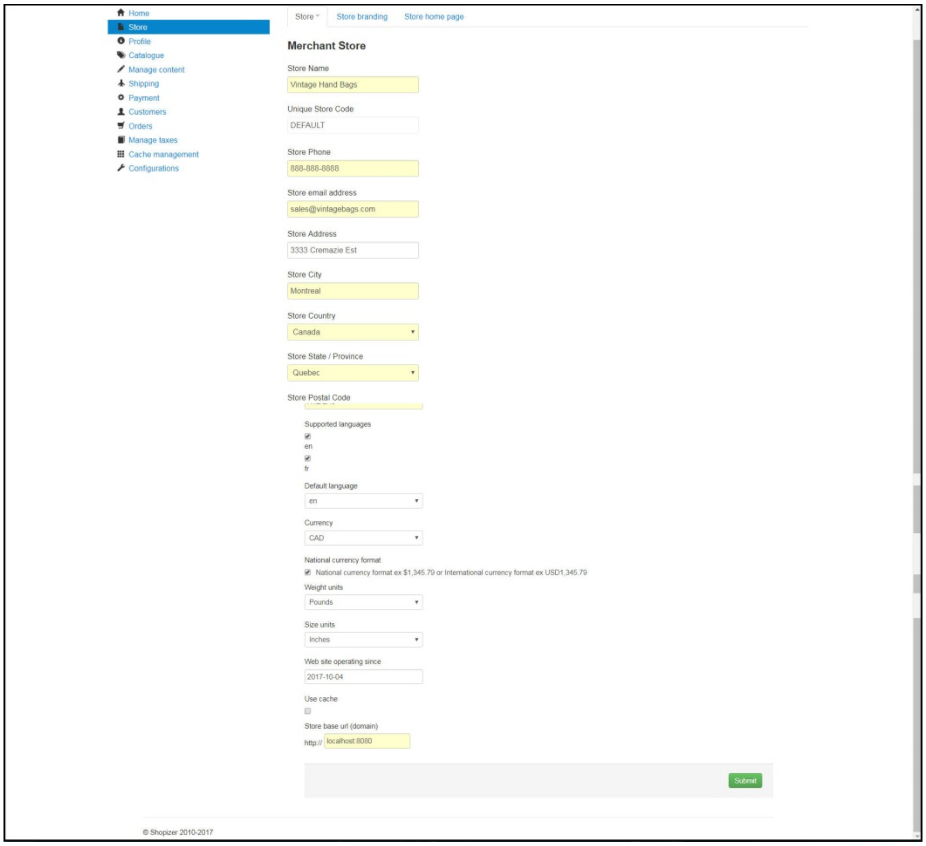
Figure 2 – Groups and permissions

Customers and Users are added to groups allowing access to functionalities restricted to certain permissions. Multiple groups can be added to customers and users.

Store configuration

Store information (Merchant Store)

- **Edit Store**



Merchant Store

Store Name
Vintage Hand Bags

Unique Store Code
DEFAULT

Store Phone
888-888-8888

Store email address
sales@vintagebags.com

Store Address
3333 Cremazie Est

Store City
Montreal

Store Country
Canada

Store State / Province
Quebec

Store Postal Code

Supported languages
☑
en
☑
fr

Default language
en

Currency
CAD

National currency format
☑ National currency format ex $1,345.79 or International currency format ex USD1,345.79

Weight units
Pounds

Size units
Inches

Web site operating since
2017-10-04

Use cache
□

Store base url (domain)
http:// localhost:8080

© Shopizer 2010-2017

Description of the 'Store' fields

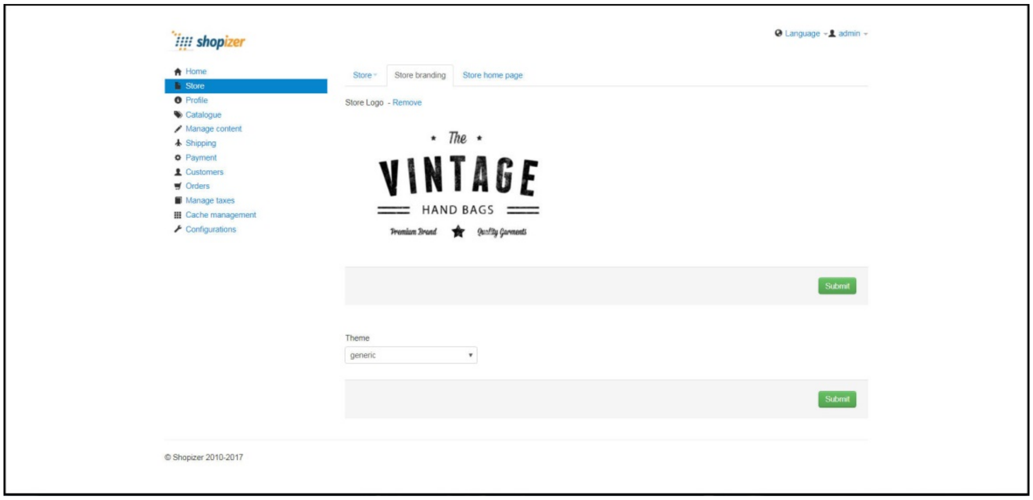
Store Name	Name of the store that will be displayed on invoices, Contact page and payment receipts.
Unique store code	Unique code generated at the creation of a state. Should be DEFAULT
Store Phone	Phone number displayed on the contact us page.

Store email address	Email used when contact us form is used. This email is also displayed on email confirmation.
Store Address	Store address displayed in contact us page
Store City	Store city displayed in contact us page
Store Country	Store country displayed in contact us page
Store State / Province	Store state/province displayed in contact us page
Store Postal Code	Store postal code displayed in contact us page. The postal code is used when calculating shipping fees between package origin and destination.
Supported languages	The system currently supports English and French. Those languages allow language switching when using store front web application.
Default language	Default selected and displayed language.
Currency	Store currency
National currency format	This flag indicates the system to use the national currency symbol ie ($,¥…) instead of currency ISO format (CAD, USD…)
Weight units	Pounds or kilograms. This selection will be displayed in product details and when getting shipping quotes.
Size units	Inches or centimeters. This selection will be displayed in product details and when getting shipping quotes.
Web site operating since	Will be displayed in store front footer

Use cache	Use this option to optimize website performance. When the products are properly configured, this option makes use of caching for all section of the system.
Store base url (domain)	Domain name the website responds (default is localhost:8080)

- **Store branding**

Login -> Store -> Store Branding



Store Logo - Remove

Theme: generic

Description of the 'Store' fields

Store logo	Upload your store logo. Will be used in store front web application and confirmation emails sent to the customer.
Theme *	Different store front themes are available,

(*) Shopizer comes out of the box configured with pre-defined themes. It is possible to extend the system with additional themes. Themes can be found in Shopizer project under this location shopizer/sm-shop/src/main/webapp/pages/shop/templates

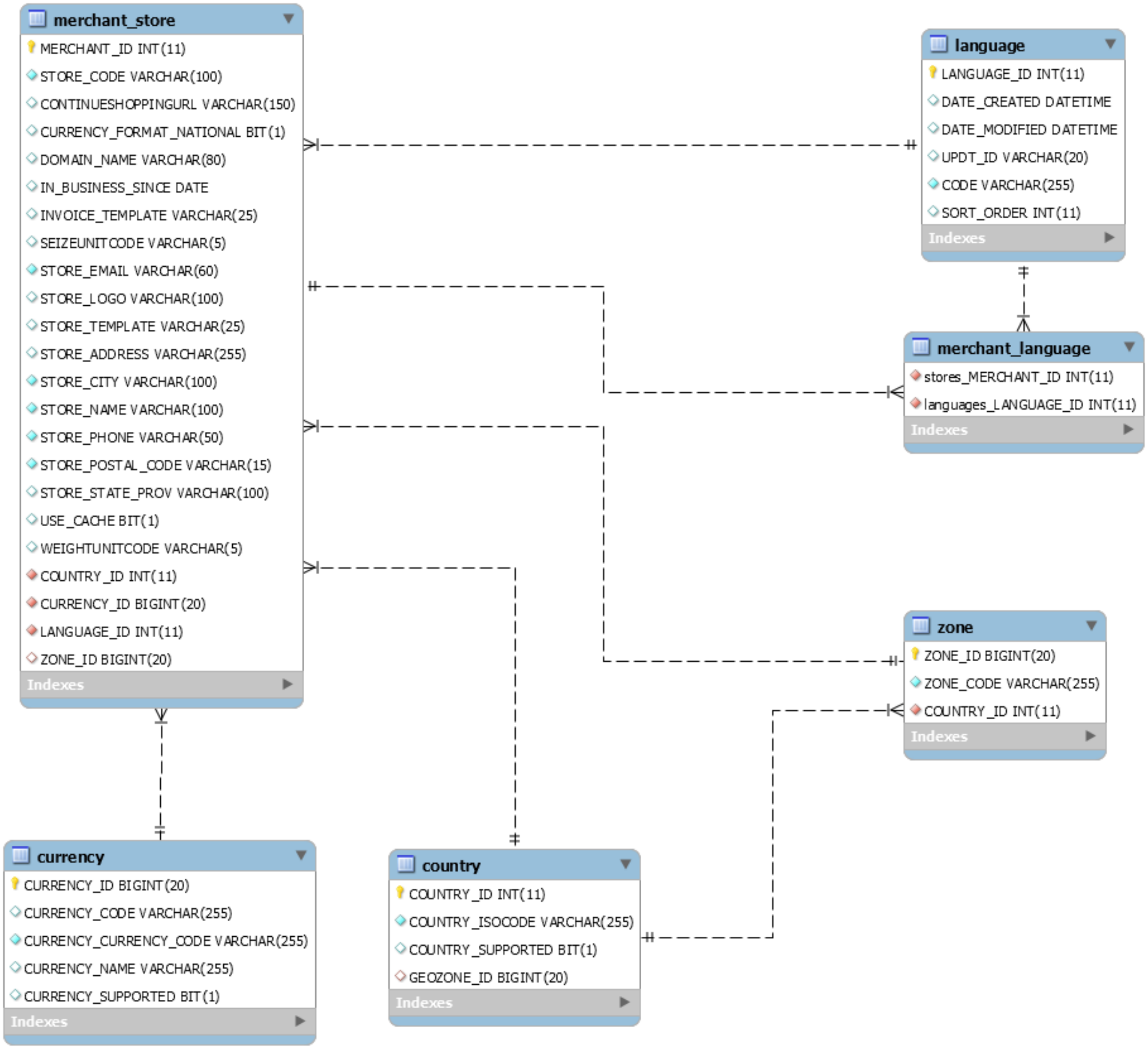


Figure 3 Merchanr

MerchantStore has core fields allowing the definition of a specific Merchant. A Merchant has a default Language but supports multiple Languages. A MerchantStore supports a single Currency and is defined in a single Zone.

Integration Modules definition objects

Shopizer integrates with multiple systems in order to support functionality such as Payment and Shipping. Initial configuration of supported Payment and Shipping modules is defined in sm-core – resources – reference/**integrationmodules.json** which is then inserted to Shopizer initial schema created when Shopizer is started for the first time. Any additional integration module will have to be added to integrationmodules.json and then inserted to the runtime module tables by POSTING a request to <host>:<port>/services /private/system/module. See `SystemAPITest createIntegrationModule for an example.`

Shipping module Meta definition (example of a single module definition)

```
{
        "module": "SHIPPING",
        "code": "usps",
        "version":"",
        "regions": ["US"],
        "image":"usps.jpg",

"configuration":[{"env":"TEST","scheme":"http","host":"testing.shippingapis.com","port":"80","uri":"/ShippingAPI.dll"},{"env":"PROD","scheme":"http","host":"production.shippingapis.com","port":"80","uri":"/ShippingAPI.dll"}]

}
```

Payment module Meta definition (example of a single module definition)

```
{
        "module": "PAYMENT",
        "code": "stripe",
        "type":"creditcard",
        "version":"",
        "regions": ["US","CA","GB","AU","FI","DK","IE","NO","SE"],
        "image":"stripe.png",

"configuration":[{"env":"TEST","scheme":"https","host":"www.stripe.com","port":"443","uri":"/"},{"env":"PROD","scheme":"https","host":"www.stripe.com","port":"443","uri":"/"}]

}
```

- Module * : Supported module (PAYMENT | SHIPPING_
- Code * : Unique code
- Type : sub type
- Version : Define a specific version
- Regions * : Supported country code
- Image * : Logo of the integration system that will appear in Shopizer admin console
- Configuration * : Production and Sandboxes environment urls

When Shopizer is started for the first time, a process reads integrationmodules.json and inserts data in MODULE_CONFIGURATION table. Meta definitions cannot be edited once loaded in Shopizer tables. Administration pages will be used in order to configure runtime data for each module required by the application. Runtime data is defined in MERCHANT_CONFIGURATION in a key value format. See specific configuration sections below for more details.

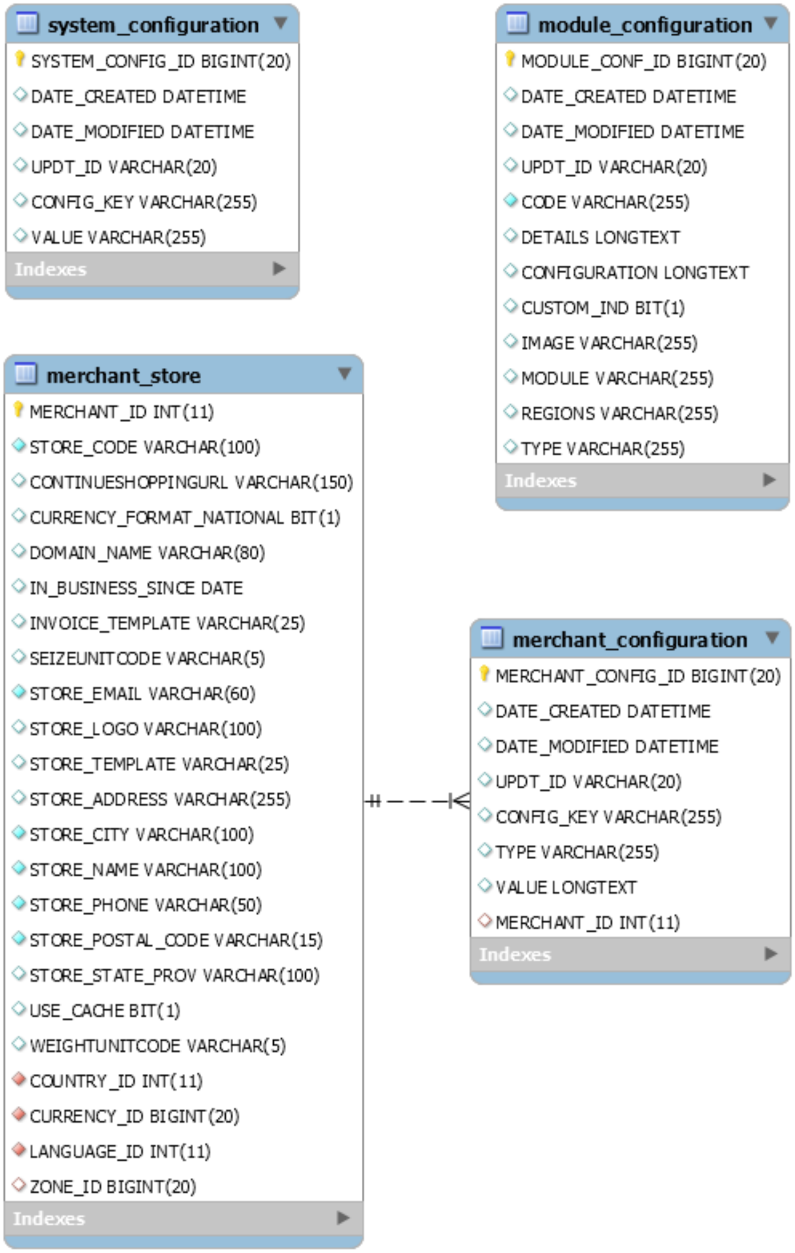
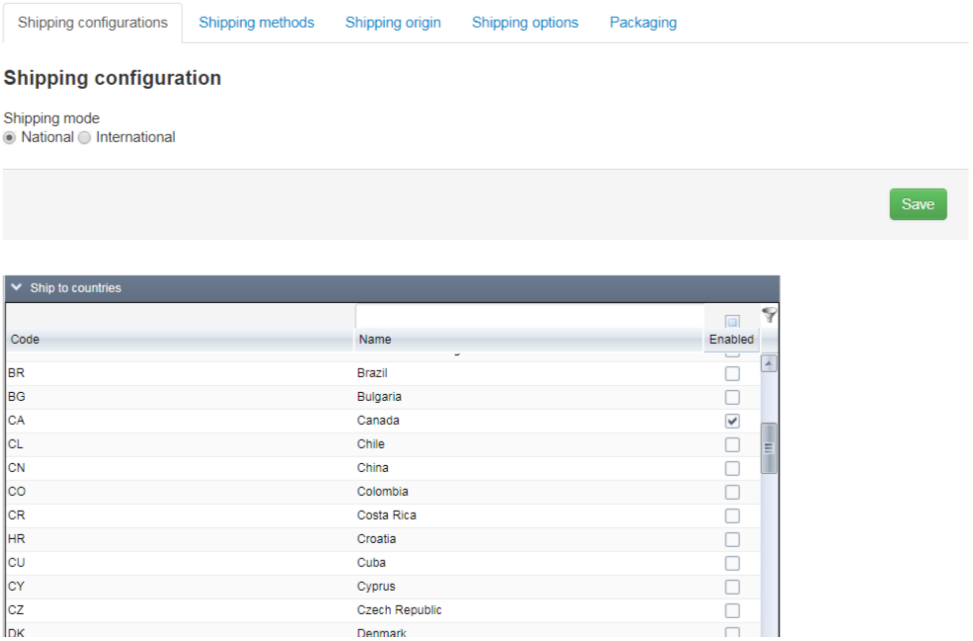


Figure 4 Shopizer configuration

Shipping configuration

Login -> Shipping -> Shipping configurations

Allows specifying if shipping is national or international. When international is selected, it is necessary to select for which country



Shipping configuration

Shipping mode
● National ○ International

Code	Name	Enabled
BR	Brazil	□
BG	Bulgaria	□
CA	Canada	☑
CL	Chile	□
CN	China	□
CO	Colombia	□
CR	Costa Rica	□
HR	Croatia	□
CU	Cuba	□
CY	Cyprus	□
CZ	Czech Republic	□
DK	Denmark	□

- **Shipping methods**

  Login -> Shipping -> Shipping methods

  Specify the shipping methods to be used when requesting a shipping quote. Some of the shipping method requires creating an account with their quotation gateway system. When creating items an attribute specifies if the item requires to be shipped or not (default not). If the item requires shipping and no shipping method is configured then it won't be possible to complete checkout flow.

- **Shipping origin**

  Login -> Shipping -> Shipping origin

  Allows specifying a shipping address origin different than store address for getting shipping quotations.

- **Shipping options**

  Login -> Shipping -> Shipping options

  Allows adding handling fees and to create 'free shipping' rules.

- **Packaging**

  Login -> Shipping -> Packaging

Allows specifying if the items are shipped individually or if boxes are used to pack items together.

When items are shipped individually the system takes the product dimension and weight for getting the appropriate shipping quote.

When items are packed in a box the system will determine how many boxes are required based on the volume of items and box size. The final box count and dimension will be used to calculate the quote.

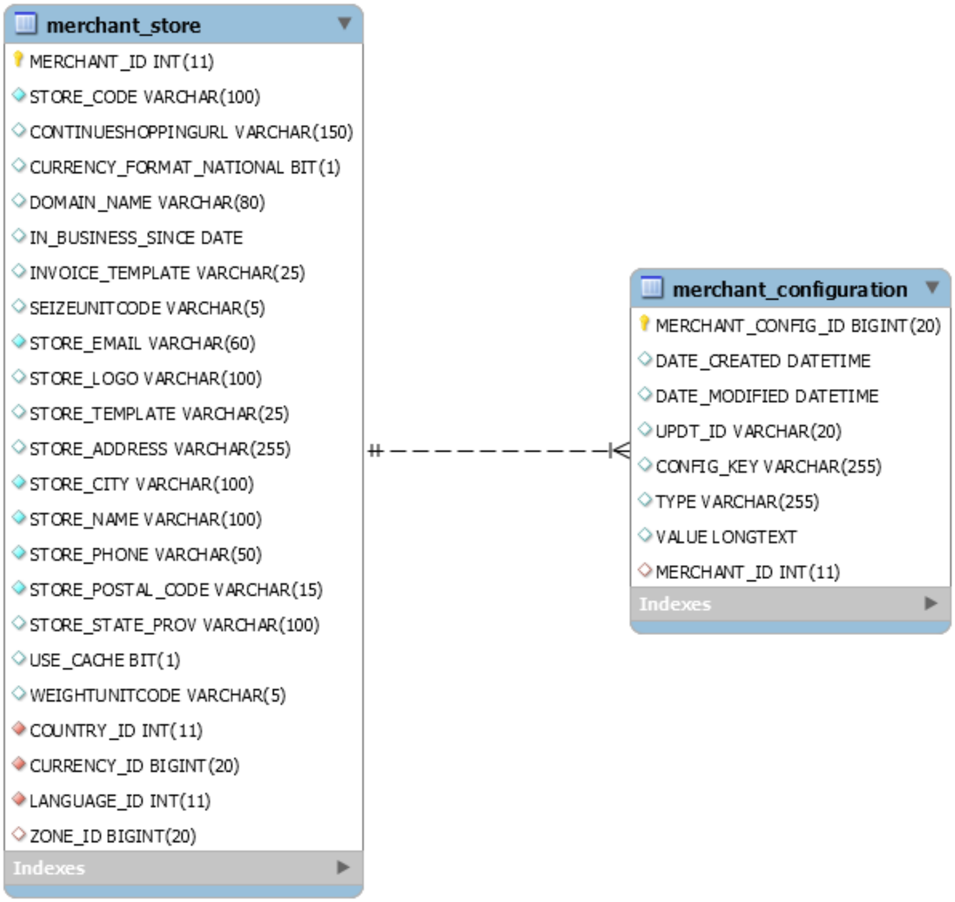


Figure 5 – Shipping configuration

Merchant store saves encrypted configuration in generic key value merchant_configuration table.

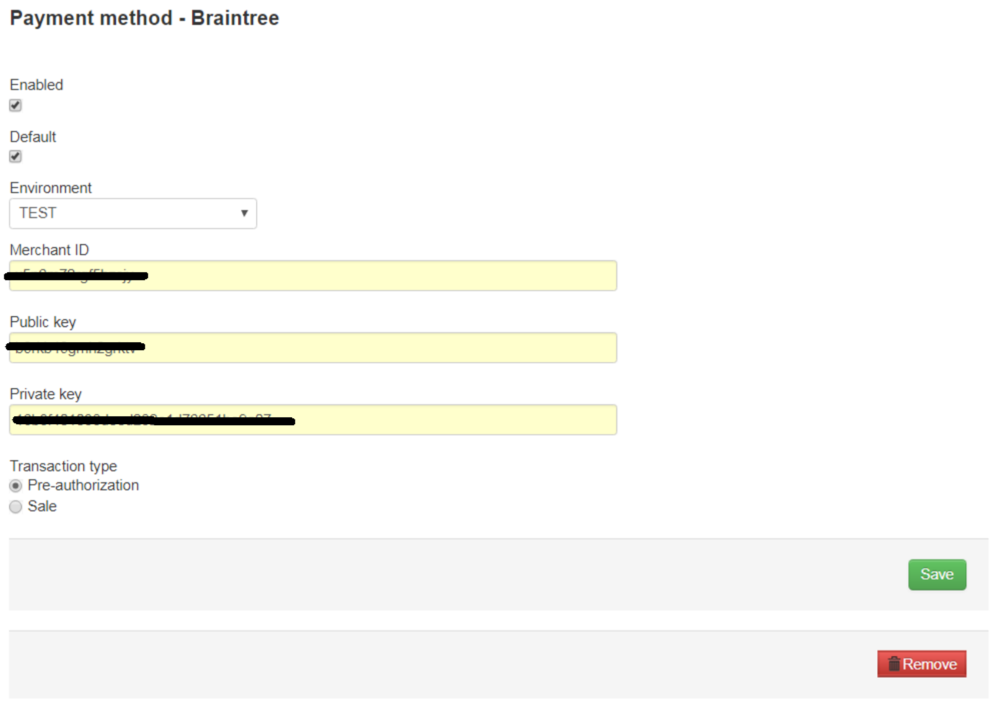
MERCHANT_CONFIG_ID	DATE_CREATED	DATE_MODIFIED	UPDT_ID	CONFIG_KEY	TYPE	VALUE	MERCHANT_ID
ID	2018-01-07 20:36	2018-01-07 20:36:	<null	SUPPORTED_CNTR	INTEGRATION	["CA","IN","CN","RU","US"]	1

	:41.328	41.328	>				
101	2018-01-07 20:40:43.788	2018-01-07 20:49:14.527	<null>	SHIPPING_CONFIG	INTEGRATION	{"boxLength":0,"orderTotalFreeShipping":null,"shipBaseType":"SHIPPING","shipOptionPriceType":"LEAST","maxWeight":0.0,"boxHeight":0,"taxOnShipping":false,"shipPackageType":"ITEM","freeShippingEnabled":false,"boxWeight":0.0,"shipType":"INTERNATIONAL","boxWidth":0,"handlingFees":null,"shipDescription":"SHORT_DESCRIPTION"}	101
102	2018-01-07 20:41:37.719	2018-01-07 20:41:53.829	<null>	SHIPPING	INTEGRATION	<ENCRYPTED>	1
103	2018-01-07 20:41:53.831	2018-01-07 20:47:48.68	<null>	weightBased	INTEGRATION	{"moduleCode":"weightBased","active":true,"regions":[{"customRegionName":"CANADA","countries":["CA"],"quoteItems":[{"price":19.99,"maximumWeight":200}]},{"customRegionName":"US","countries":["US"],"quoteItems":[{"price":19.99,"maximumWeight":200}]},{"customRegionName":"INDIA","countries":["IN"],"quoteItems":[{"price":19.99,"maximumWeight":200}]},{"customRegionName":"CHINA","countries":["CN"],"quoteItems":[{"price":19.99,"maximumWeight":200}]},{"customRegionName":"RUSSIA","countries":["RU"],"quoteItems":[{"price":19.99,"maximumWeight":200}]}]}	1

Payment configuration

Login -> Payment

Payment methods (except cheque) require the creation of an account with the selected payment method. Usually 2 payment methods are offered (credit card & Paypal).



Payment method - Braintree

Enabled
☑

Default
☑

Environment
TEST ▾

Merchant ID
[redacted]

Public key
[redacted]

Private key
[redacted]

Transaction type
● Pre-authorization
○ Sale

Description of Braintree payment method fields

Enabled	Upload your store logo. Will be used in store front web application and confirmation emails sent to the customer.
Default	Default payment method selected in store front web application
Environment	TEST or PRODUCTION. Usually account information are different between test and production
Merchant ID	Goto https://www.braintreepayments.com/ Signup for the sandbox
Public Key	Goto https://www.braintreepayments.com/ Signup for the sandbox
Private Key	Goto

	https://www.braintreepayments.com/ Signup for the sandbox
Transaction type	Pre-authorization and Sale. ** When pre-authorization is selected, an action must be made from the order details to capture the transaction. An admin API is also available to remotely trigger capture action on a specific order.

Content management

Content objects are attached to MERCHANT_STORE. MERCHANT_STORE can define "0 to many" unique content identified by a unique code used as directive in the ui files when needing to embed specific content. Content have "1 to many" content description for each supported language.

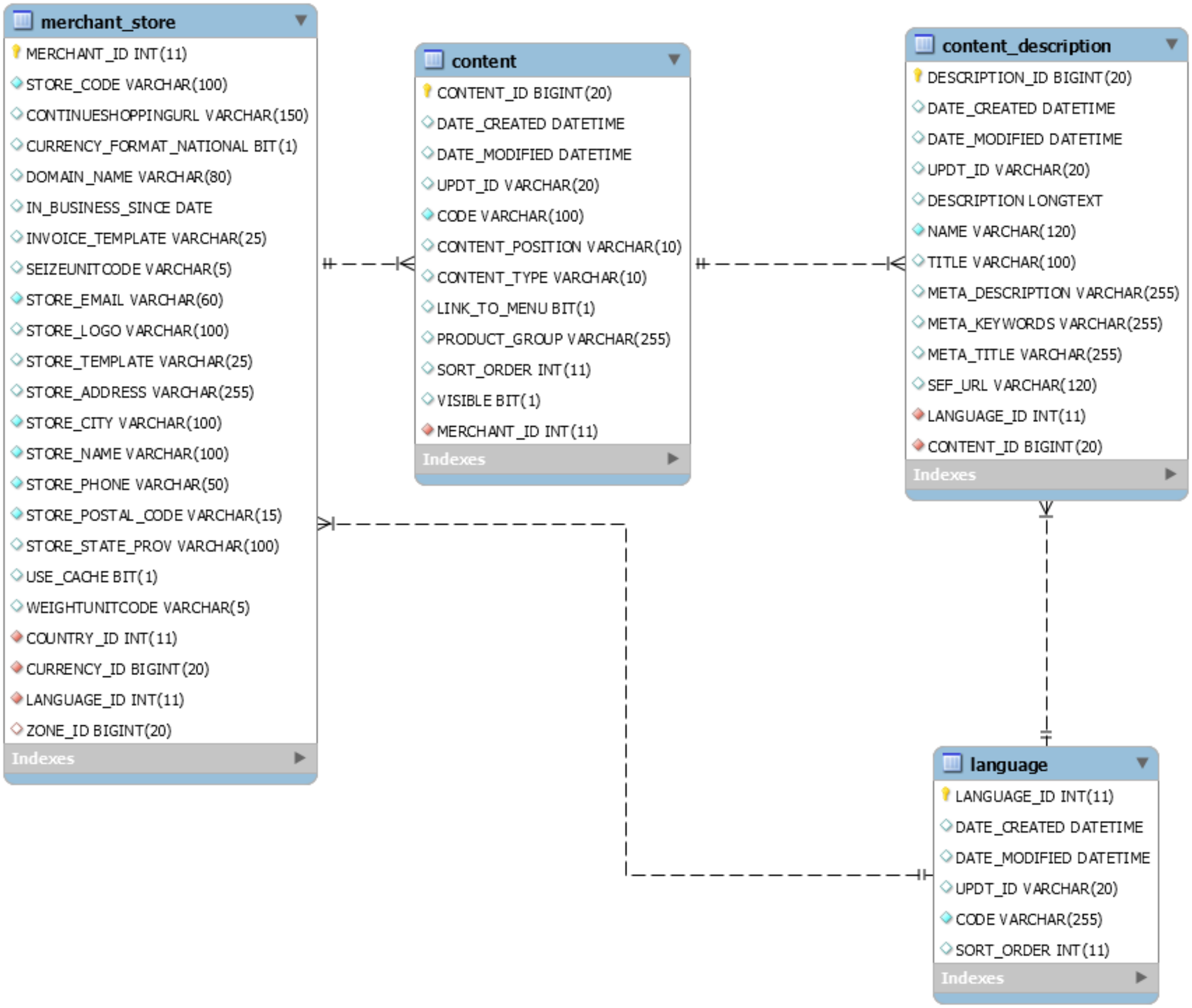


Figure 6 – Content management

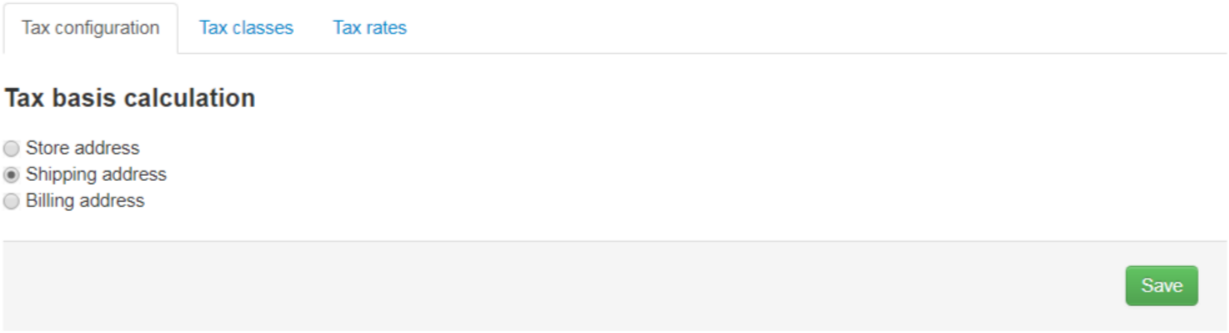
Tax calculation

Shopizer manages taxes by country and zone (state/province). During checkout the system calculates taxes based on different scenarios in the following order:

- Customer is already authenticated and the system can retrieve billing and shipping address.

- Customer is anonymous but has filled in billing / shipping address.
- "Geolocation" component has successfully detected customer country and province.

Login -> Tax configuration



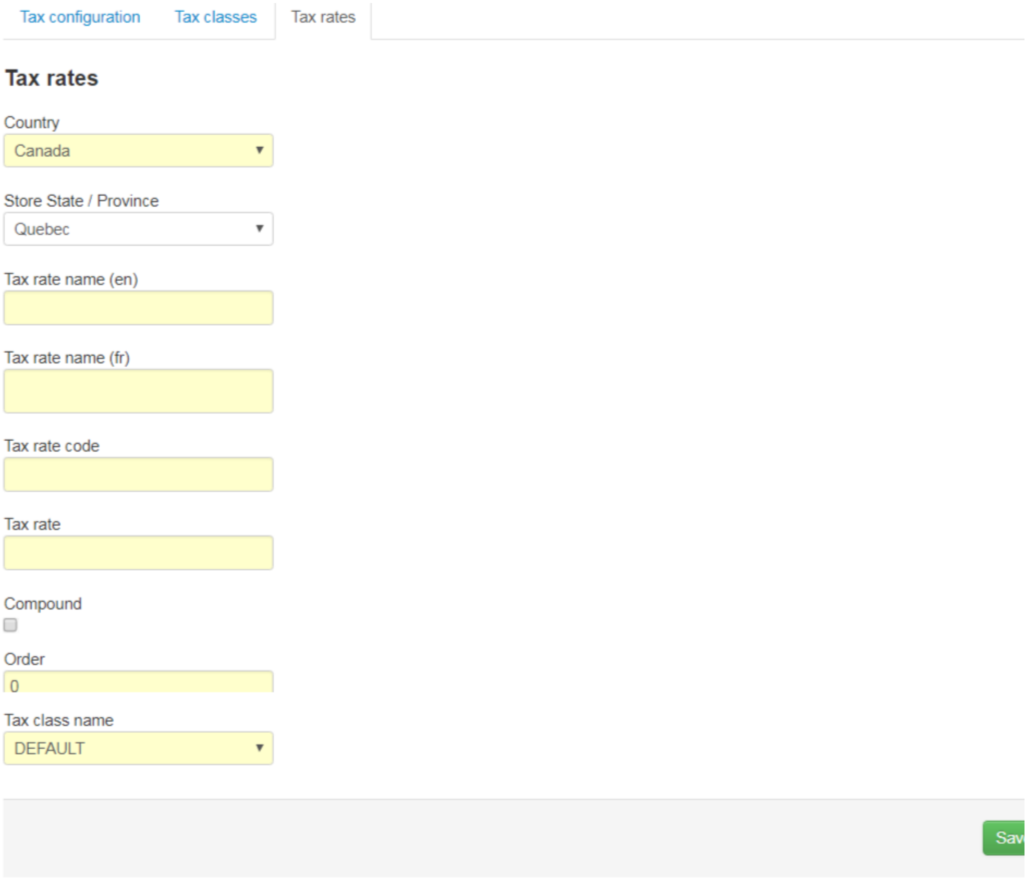
Tax basis calculation

○ Store address
● Shipping address
○ Billing address

Tax calculation may be calculated based on shipping address, billing address of from store address.

Tax Rates

Login -> Tax Rates



Tax rates

Country: Canada

Store State / Province: Quebec

Tax rate name (en):

Tax rate name (fr):

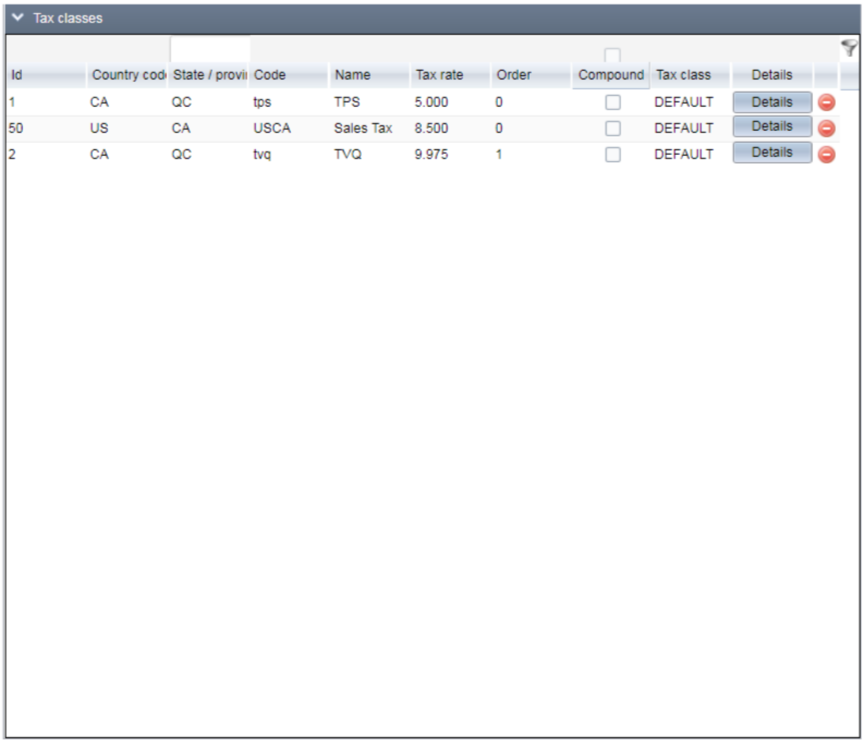
Tax rate code:

Tax rate:

Compound: □

Order: 0

Tax class name: DEFAULT



Id	Country cod	State / provi	Code	Name	Tax rate	Order	Compound	Tax class	Details	
1	CA	QC	tps	TPS	5.000	0	□	DEFAULT	Details	⊖
50	US	CA	USCA	Sales Tax	8.500	0	□	DEFAULT	Details	⊖
2	CA	QC	tvq	TVQ	9.975	1	□	DEFAULT	Details	⊖

Specifies tax rate based on the customer shipping or billing address.

Country	Country for which tax rate applies
State / Province	State / Province for which tax rate applies
Tax rate (by language)	Tax rate name to be displayed to buying customer
Tax rate code	Unique tax rate code
Tax rate	Tax rate (decimal value) ie 5.5
Compounded	Tax rate applied on top of all tax rates preceding. For example tax rate order 1 is 5% and tax rate order 2 is 7% and compounded. Tax calculation will be price * tax rate 1 * tax rate 2 instead of price * tax rate 1 plus (+) price * tax rate 2
Order	Order of calculation when multiple taxes apply for a region (example state and country tax)
Tax class	Associated to a product. A country / state may have different rates depending on the product class.

Tax classes are different types of class that can be used for instance to calculate different rates for specific products. For example in some country clothes might not be taxable while shoes are taxable. In the catalog, clothes would use a given tax class with a rate being 0 and shoes would use default tax class with appropriate rates configured.

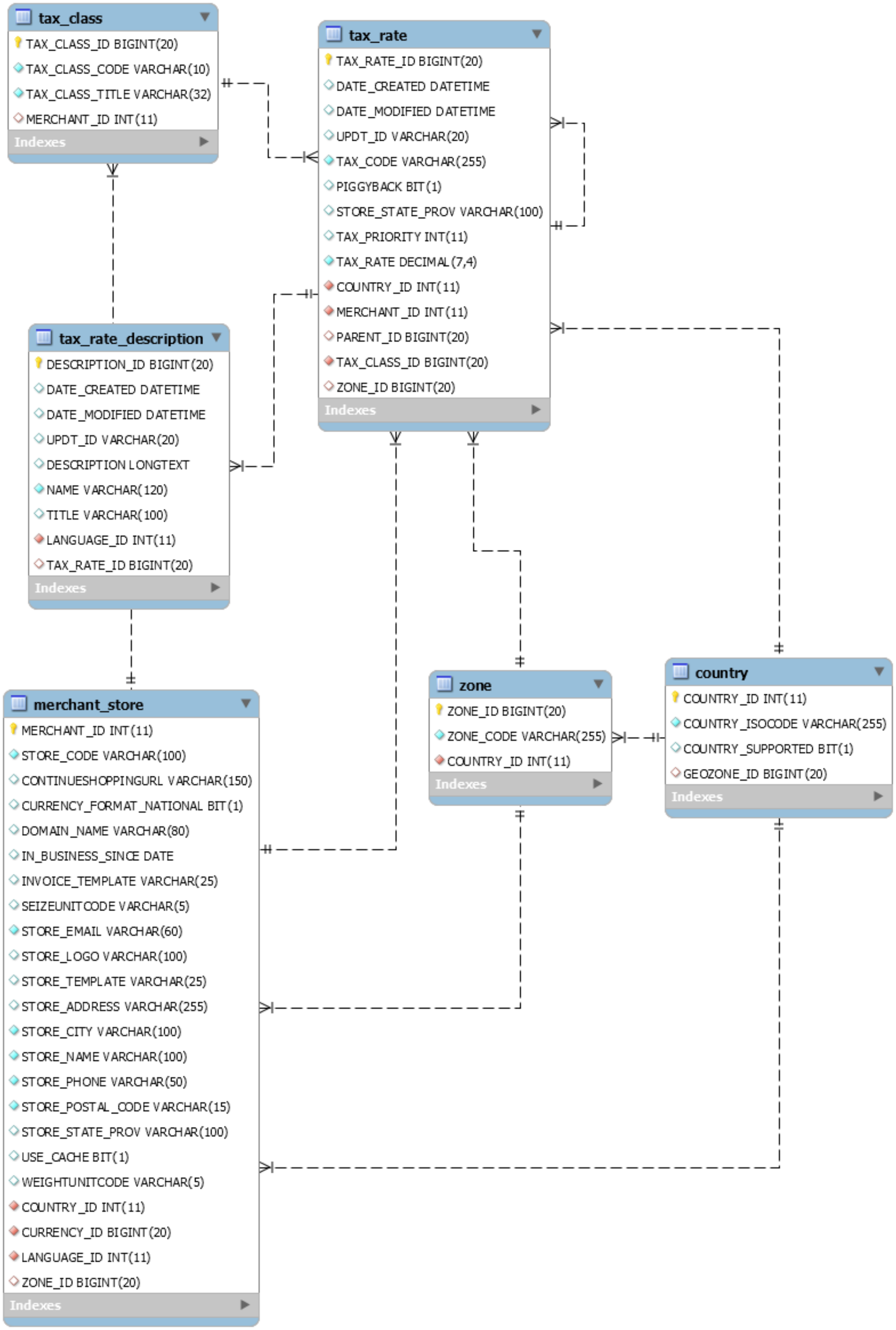
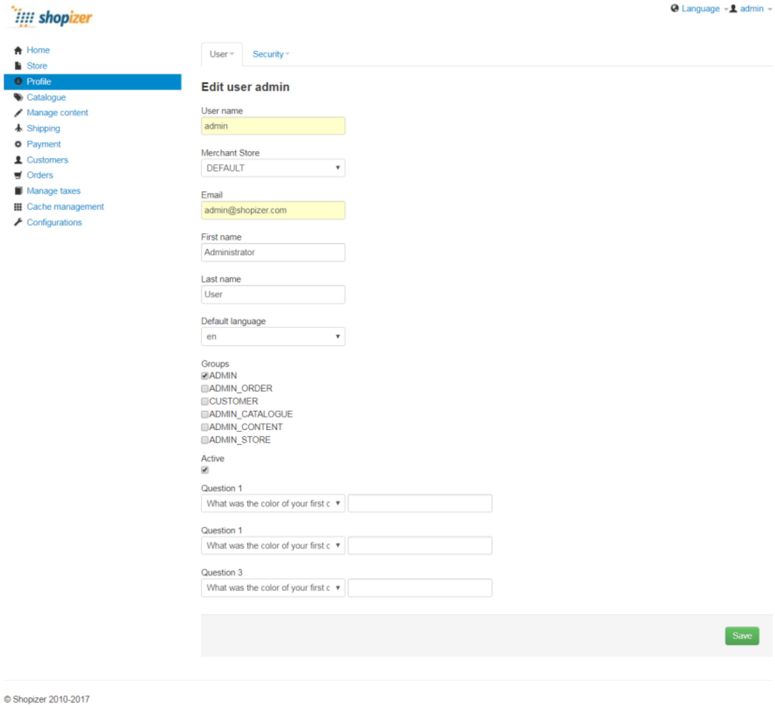

tax_class
TAX_CLASS_ID BIGINT(20)
TAX_CLASS_CODE VARCHAR(10)
TAX_CLASS_TITLE VARCHAR(32)
MERCHANT_ID INT(11)
Indexes
tax_rate
TAX_RATE_ID BIGINT(20)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
TAX_CODE VARCHAR(255)
PIGGYBACK BIT(1)
STORE_STATE_PROV VARCHAR(100)
TAX_PRIORITY INT(11)
TAX_RATE DECIMAL(7,4)
COUNTRY_ID INT(11)
MERCHANT_ID INT(11)
PARENT_ID BIGINT(20)
TAX_CLASS_ID BIGINT(20)
ZONE_ID BIGINT(20)
Indexes
tax_rate_description
DESCRIPTION_ID BIGINT(20)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
DESCRIPTION LONGTEXT
NAME VARCHAR(120)
TITLE VARCHAR(100)
LANGUAGE_ID INT(11)
TAX_RATE_ID BIGINT(20)
Indexes
merchant_store
MERCHANT_ID INT(11)
STORE_CODE VARCHAR(100)
CONTINUESHOPPINGURL VARCHAR(150)
CURRENCY_FORMAT_NATIONAL BIT(1)
DOMAIN_NAME VARCHAR(80)
IN_BUSINESS_SINCE DATE
INVOICE_TEMPLATE VARCHAR(25)
SEIZEUNITCODE VARCHAR(5)
STORE_EMAIL VARCHAR(60)
STORE_LOGO VARCHAR(100)
STORE_TEMPLATE VARCHAR(25)
STORE_ADDRESS VARCHAR(255)
STORE_CITY VARCHAR(100)
STORE_NAME VARCHAR(100)
STORE_PHONE VARCHAR(50)
STORE_POSTAL_CODE VARCHAR(15)
STORE_STATE_PROV VARCHAR(100)
USE_CACHE BIT(1)
WEIGHTUNITCODE VARCHAR(5)
COUNTRY_ID INT(11)
CURRENCY_ID BIGINT(20)
LANGUAGE_ID INT(11)
ZONE_ID BIGINT(20)
Indexes
zone
ZONE_ID BIGINT(20)
ZONE_CODE VARCHAR(255)
COUNTRY_ID INT(11)
Indexes
country
COUNTRY_ID INT(11)
COUNTRY_ISOCODE VARCHAR(255)
COUNTRY_SUPPORTED BIT(1)
GEOZONE_ID BIGINT(20)
Indexes

Figure 7 – Tax rates

Users

Login -> User -> Create a user

Use this step for creating administration users. Those users can be used to log in from the administration web page (localhost:8080/admin) or from REST API */private/* administration services.



Edit user admin

User name: admin

Merchant Store: DEFAULT

Email: admin@shopizer.com

First name: Administrator

Last name: User

Default language: en

Groups
☑ ADMIN
□ ADMIN_ORDER
□ CUSTOMER
□ ADMIN_CATALOGUE
□ ADMIN_CONTENT
□ ADMIN_STORE

Active
☑

Question 1: What was the color of your first c

Question 1: What was the color of your first c

Question 3: What was the color of your first c

Description of the 'User' fields

User name	This is the username used for login. The username is unique.
Merchant Store	Username should be associated to 'DEFAULT' store.
Email	Email address used for communication through the system (password reset, new user created…)
First name	First name of the user
Last name	Last name of the user

Default language	Will be used when logging in the administration console
Groups	Restrict activity by selecting the desired user role. ‘Admin’ permits all access. For the API, admin should be selected.
Active	Flag the user account active or inactive
Security questions	Should be answered in order to support reset password self-serve reset password functionality

Once a user is created, it is possible to change the user password Login -> Profile -> Change Password

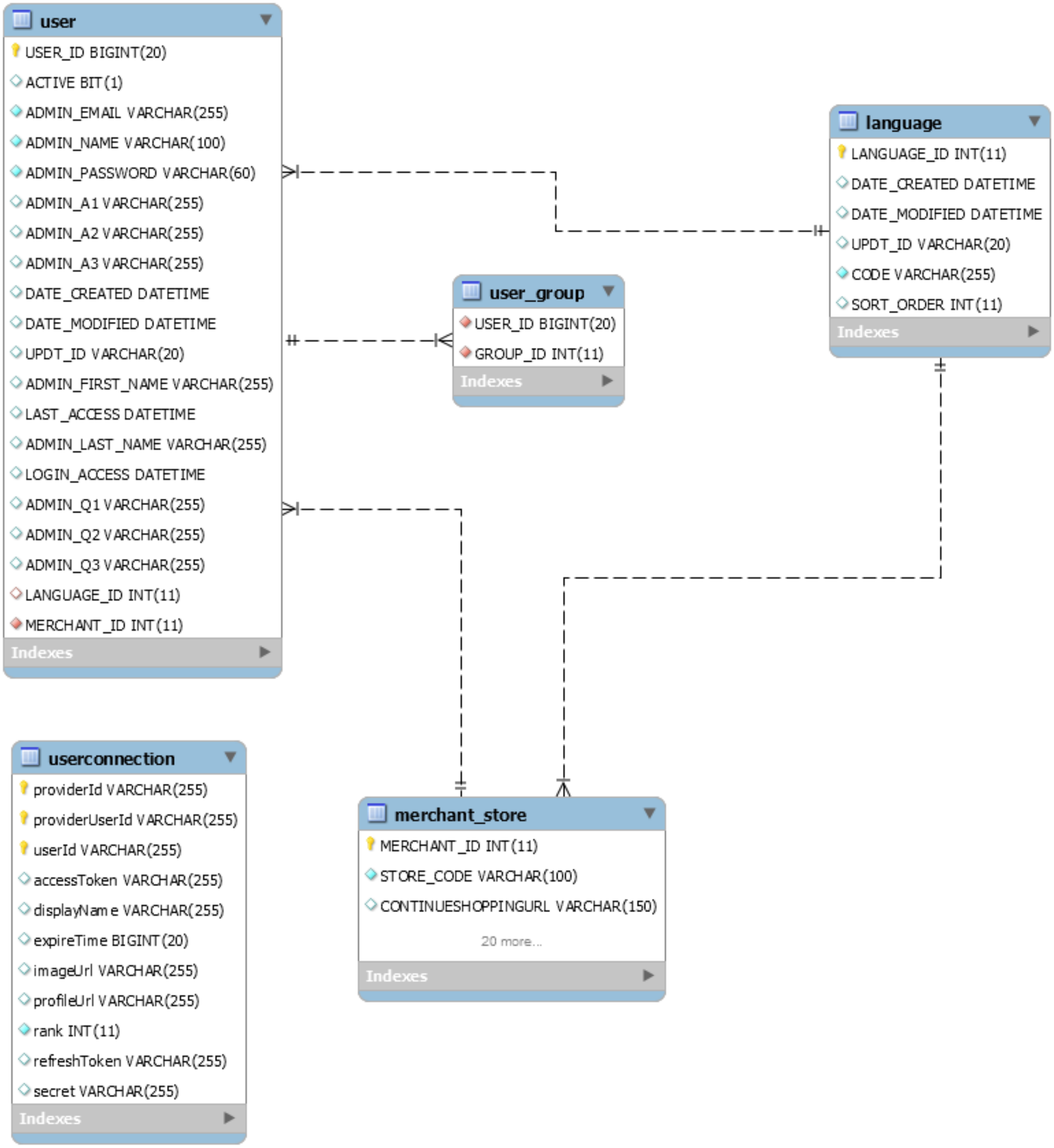
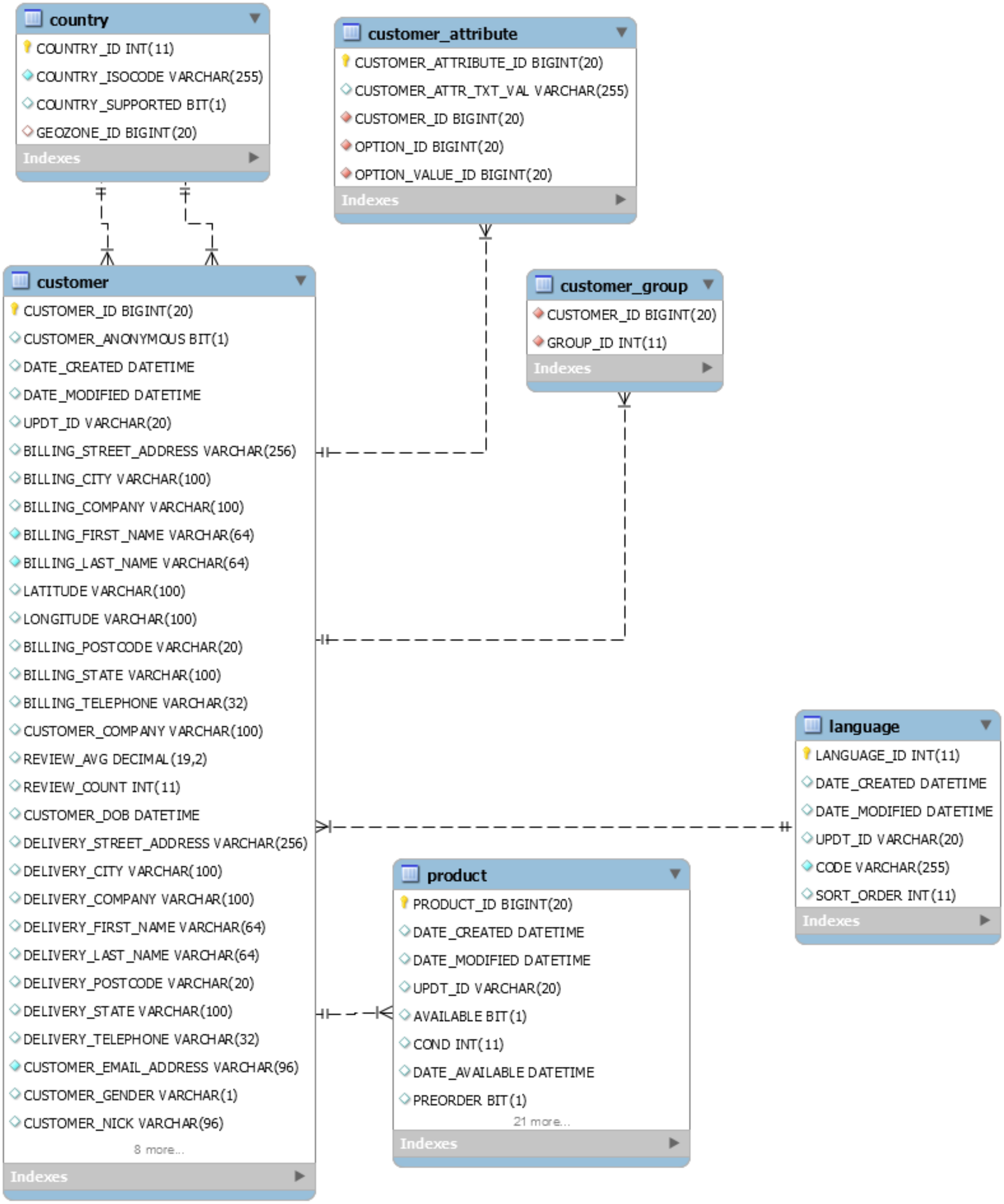


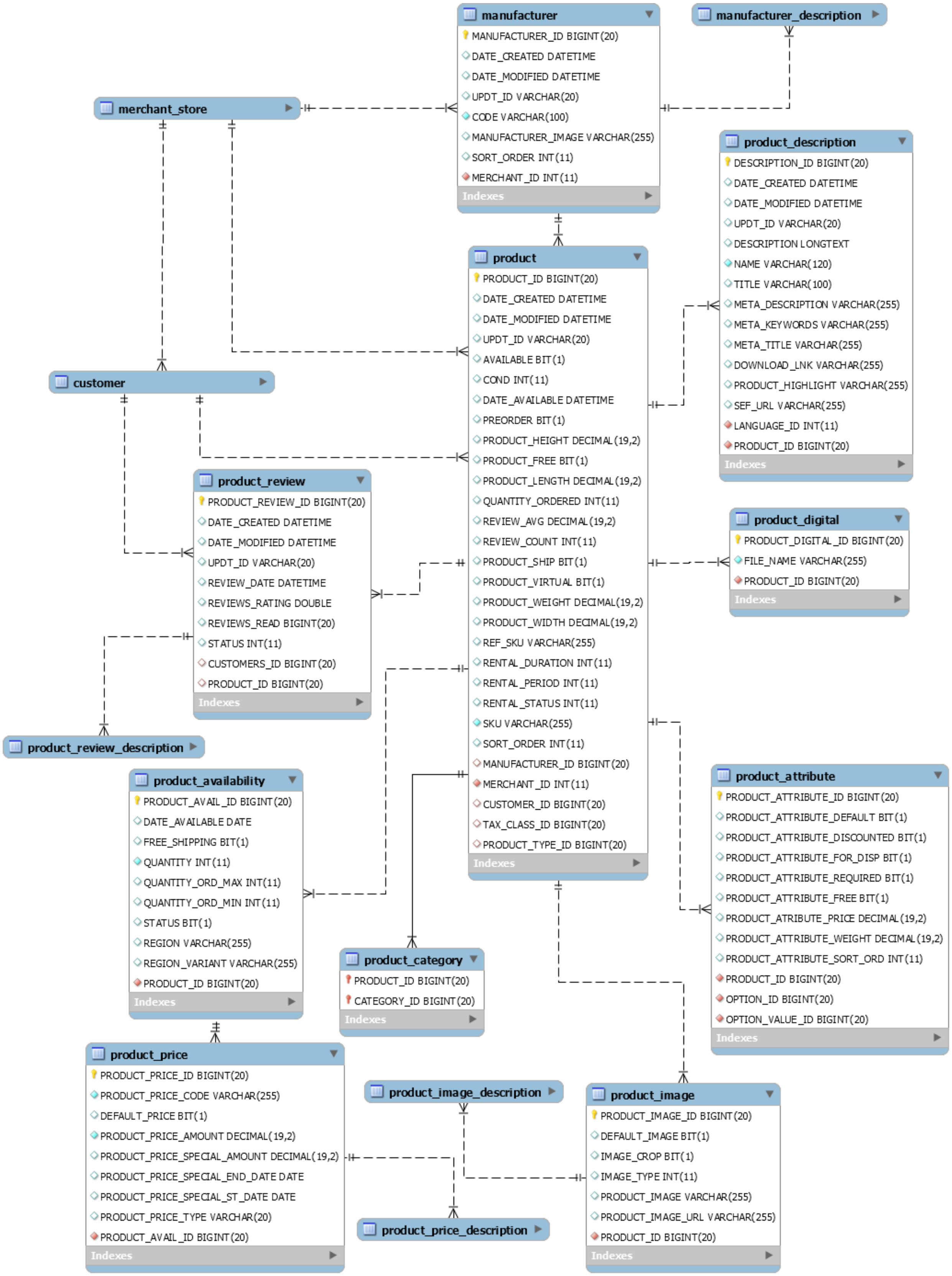
Figure 8 User model

A user has a default language and is created in a single store context. Users can belong to multiple groups. User connection is a transient representation of the user logged on from social networks applications such as facebook.

Customer


country
COUNTRY_ID INT(11)
COUNTRY_ISOCODE VARCHAR(255)
COUNTRY_SUPPORTED BIT(1)
GEOZONE_ID BIGINT(20)
Indexes
customer_attribute
CUSTOMER_ATTRIBUTE_ID BIGINT(20)
CUSTOMER_ATTR_TXT_VAL VARCHAR(255)
CUSTOMER_ID BIGINT(20)
OPTION_ID BIGINT(20)
OPTION_VALUE_ID BIGINT(20)
Indexes
customer
CUSTOMER_ID BIGINT(20)
CUSTOMER_ANONYMOUS BIT(1)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
BILLING_STREET_ADDRESS VARCHAR(256)
BILLING_CITY VARCHAR(100)
BILLING_COMPANY VARCHAR(100)
BILLING_FIRST_NAME VARCHAR(64)
BILLING_LAST_NAME VARCHAR(64)
LATITUDE VARCHAR(100)
LONGITUDE VARCHAR(100)
BILLING_POSTCODE VARCHAR(20)
BILLING_STATE VARCHAR(100)
BILLING_TELEPHONE VARCHAR(32)
CUSTOMER_COMPANY VARCHAR(100)
REVIEW_AVG DECIMAL(19,2)
REVIEW_COUNT INT(11)
CUSTOMER_DOB DATETIME
DELIVERY_STREET_ADDRESS VARCHAR(256)
DELIVERY_CITY VARCHAR(100)
DELIVERY_COMPANY VARCHAR(100)
DELIVERY_FIRST_NAME VARCHAR(64)
DELIVERY_LAST_NAME VARCHAR(64)
DELIVERY_POSTCODE VARCHAR(20)
DELIVERY_STATE VARCHAR(100)
DELIVERY_TELEPHONE VARCHAR(32)
CUSTOMER_EMAIL_ADDRESS VARCHAR(96)
CUSTOMER_GENDER VARCHAR(1)
CUSTOMER_NICK VARCHAR(96)
8 more...
Indexes
customer_group
CUSTOMER_ID BIGINT(20)
GROUP_ID INT(11)
Indexes
language
LANGUAGE_ID INT(11)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
CODE VARCHAR(255)
SORT_ORDER INT(11)
Indexes
product
PRODUCT_ID BIGINT(20)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
AVAILABLE BIT(1)
COND INT(11)
DATE_AVAILABLE DATETIME
PREORDER BIT(1)
21 more...
Indexes

Products


manufacturer
MANUFACTURER_ID BIGINT(20)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
CODE VARCHAR(100)
MANUFACTURER_IMAGE VARCHAR(255)
SORT_ORDER INT(11)
MERCHANT_ID INT(11)
Indexes
manufacturer_description
merchant_store
product_description
DESCRIPTION_ID BIGINT(20)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
DESCRIPTION LONGTEXT
NAME VARCHAR(120)
TITLE VARCHAR(100)
META_DESCRIPTION VARCHAR(255)
META_KEYWORDS VARCHAR(255)
META_TITLE VARCHAR(255)
DOWNLOAD_LNK VARCHAR(255)
PRODUCT_HIGHLIGHT VARCHAR(255)
SEF_URL VARCHAR(255)
LANGUAGE_ID INT(11)
PRODUCT_ID BIGINT(20)
Indexes
product
PRODUCT_ID BIGINT(20)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
AVAILABLE BIT(1)
COND INT(11)
DATE_AVAILABLE DATETIME
PREORDER BIT(1)
PRODUCT_HEIGHT DECIMAL(19,2)
PRODUCT_FREE BIT(1)
PRODUCT_LENGTH DECIMAL(19,2)
QUANTITY_ORDERED INT(11)
REVIEW_AVG DECIMAL(19,2)
REVIEW_COUNT INT(11)
PRODUCT_SHIP BIT(1)
PRODUCT_VIRTUAL BIT(1)
PRODUCT_WEIGHT DECIMAL(19,2)
PRODUCT_WIDTH DECIMAL(19,2)
REF_SKU VARCHAR(255)
RENTAL_DURATION INT(11)
RENTAL_PERIOD INT(11)
RENTAL_STATUS INT(11)
SKU VARCHAR(255)
SORT_ORDER INT(11)
MANUFACTURER_ID BIGINT(20)
MERCHANT_ID INT(11)
CUSTOMER_ID BIGINT(20)
TAX_CLASS_ID BIGINT(20)
PRODUCT_TYPE_ID BIGINT(20)
Indexes
customer
product_review
PRODUCT_REVIEW_ID BIGINT(20)
DATE_CREATED DATETIME
DATE_MODIFIED DATETIME
UPDT_ID VARCHAR(20)
REVIEW_DATE DATETIME
REVIEWS_RATING DOUBLE
REVIEWS_READ BIGINT(20)
STATUS INT(11)
CUSTOMERS_ID BIGINT(20)
PRODUCT_ID BIGINT(20)
Indexes
product_digital
PRODUCT_DIGITAL_ID BIGINT(20)
FILE_NAME VARCHAR(255)
PRODUCT_ID BIGINT(20)
Indexes
product_review_description
product_availability
PRODUCT_AVAIL_ID BIGINT(20)
DATE_AVAILABLE DATE
FREE_SHIPPING BIT(1)
QUANTITY INT(11)
QUANTITY_ORD_MAX INT(11)
QUANTITY_ORD_MIN INT(11)
STATUS BIT(1)
REGION VARCHAR(255)
REGION_VARIANT VARCHAR(255)
PRODUCT_ID BIGINT(20)
Indexes
product_attribute
PRODUCT_ATTRIBUTE_ID BIGINT(20)
PRODUCT_ATTRIBUTE_DEFAULT BIT(1)
PRODUCT_ATTRIBUTE_DISCOUNTED BIT(1)
PRODUCT_ATTRIBUTE_FOR_DISP BIT(1)
PRODUCT_ATTRIBUTE_REQUIRED BIT(1)
PRODUCT_ATTRIBUTE_FREE BIT(1)
PRODUCT_ATRIBUTE_PRICE DECIMAL(19,2)
PRODUCT_ATTRIBUTE_WEIGHT DECIMAL(19,2)
PRODUCT_ATTRIBUTE_SORT_ORD INT(11)
PRODUCT_ID BIGINT(20)
OPTION_ID BIGINT(20)
OPTION_VALUE_ID BIGINT(20)
Indexes
product_category
PRODUCT_ID BIGINT(20)
CATEGORY_ID BIGINT(20)
Indexes
product_price
PRODUCT_PRICE_ID BIGINT(20)
PRODUCT_PRICE_CODE VARCHAR(255)
DEFAULT_PRICE BIT(1)
PRODUCT_PRICE_AMOUNT DECIMAL(19,2)
PRODUCT_PRICE_SPECIAL_AMOUNT DECIMAL(19,2)
PRODUCT_PRICE_SPECIAL_END_DATE DATE
PRODUCT_PRICE_SPECIAL_ST_DATE DATE
PRODUCT_PRICE_TYPE VARCHAR(20)
PRODUCT_AVAIL_ID BIGINT(20)
Indexes
product_image_description
product_image
PRODUCT_IMAGE_ID BIGINT(20)
DEFAULT_IMAGE BIT(1)
IMAGE_CROP BIT(1)
IMAGE_TYPE INT(11)
PRODUCT_IMAGE VARCHAR(255)
PRODUCT_IMAGE_URL VARCHAR(255)
PRODUCT_ID BIGINT(20)
Indexes
product_price_description

Shopping cart

Orders

TDB

chapter

2

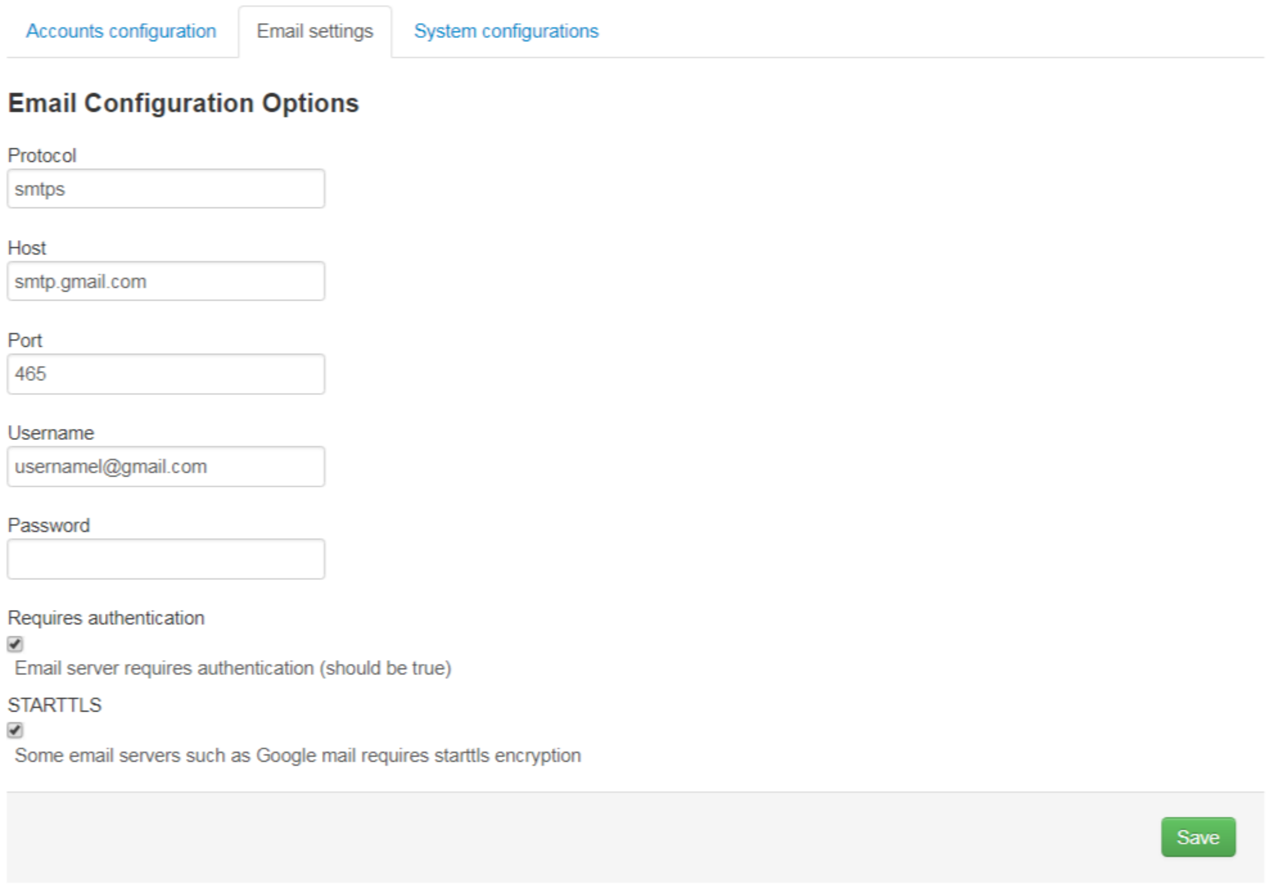
Configurations

Configurations that happen at the system level for being able to process adequately during order processing, sending emails, cache actions and displaying images.

System configuration

- **Emails settings**

Login -> Configuration -> Email settings



Email Configuration Options

Protocol: smtps

Host: smtp.gmail.com

Port: 465

Username: usernamel@gmail.com

Password:

Requires authentication
☑ Email server requires authentication (should be true)

STARTTLS
☑ Some email servers such as Google mail requires starttls encryption

Works with all email gateways as well as Google smtp server when having an existing email account.