



M0012 – SQL-Reporter

BITTE BEACHTEN SIE: MIT DER INSTALLATION DES MODUL-FRAMEWORKS WERDEN DIE BEIGEFÜGTEN LIZENZBESTIMMUNGEN UND HAFTUNGSREGELUNGEN AUSDRÜCKLICH ZUR KENNTNIS GENOMMEN UND ANERKANNT; DAS FRAMEWORK UND AUCH UNSERE MODULE DÜRFEN NICHT INSTALLIERT UND NICHT BETRIEBEN WERDEN WENN SIE DIESE REGELN NICHT IN VOLLEM UMFANG ANERKENNEN.

UPDATEHINWEISE FÜR SERVICEPACKS UND MASTER-UPDATES

Beachten Sie bei für Ihren Shop geplanten Servicepacks oder Masterupdates, dass auch das Modulframework und auch die eingebauten Module zur jeweiligen Shopversion kompatibel sein müssen und stellen Sie ggf. sicher, dass entsprechende Updates für das Modul-Framework und auch für alle genutzten Module zur Verfügung stehen **bevor** Sie mit der Aktualisierung des Shops beginnen oder diese beauftragen. Wenn Sie die Aktualisierung Ihres Shops durch uns durchführen lassen, aktualisieren wir auch immer vorhandene Xycons-Module und das Modul-Framework, wenn diese für die Shop-Masterversion verfügbar sind.

ZUR SELBSTINSTALLATION BEACHTEN SIE DIE BEIGEFÜGTE INSTALLATIONSANLEITUNG.

DAS MODUL IST KOMPATIBEL ZUR SHOP-VERSION: 4.6.X.X / 4.8.X.X / 5.0.X.X

Die erworbene Nutzungslizenz erweitert Ihren GAMBIO-Shop um folgende Funktionen:

- ➔ Nutzung vorgefertigter oder eigener SQL-Abfragen
- ➔ Cron-Job-Einbindung der SQL-Abfragen
- ➔ Export der Daten als CSV oder XLS-Datei

ACHTUNG!

Das Tool kann - je nach Anwendung - Änderungen an der Datenbank, den Daten oder Datentabellen durchführen. Wir übernehmen **keinerlei Haftung** für Beschädigungen oder Datenverluste, die aus der Nutzung von verändernden Datenbankabfragen oder anderen auf die Datenbank schreibend oder verändernd wirkenden Abfragen herrühren.

Benutzen Sie verändernde SQL-Abfragen nur, wenn Sie sich über die Auswirkungen der jeweiligen Abfrage im Klaren sind und ausreichende Datensicherung durchgeführt haben!

Für die Ergebnisse dieser Abfragen haften Sie grundsätzlich selbst!

Das Setup


Das Setup erlaubt die Einrichtung des Moduls und die Anpassung an den gewünschten Einsatz.
Das Modul kann ein- und ausgeschaltet werden.

Ihre Xycons-Module

Ihr Xycons Kundenkonto

SQL-Reports Setup

IndividualModule



[Xycons News](#)
[Xycons Updates anzeigen](#)
[Xycons Kontakt / Fehler melden](#)
[Handbuch/Dokumentation zum Modul](#)

Ihr Systemschlüssel:

5f1403886779cf96bb49f31d4db6daa8

Modulstatus - Reports

☒

Modul einschalten oder ausschalten

Ihre Lizenz:

Voll-Lizenz (unbegrenzt gültig)

Ihr Lizenzschlüssel

MnBIUkRKZThDSnInSGFiZ0M5R25D0XVUQT11Z0NzZjdPSjZnQ2FyOU85N2JESkdRRDk3N1B3R1JEOUw3UGN1VF80YVNETU9VUDIhB1BNk

Ausgabeformat

☒ XLS

☐ CSV

Wählen Sie XLS (Excel) oder CSV (Comma-Separated-Values) als Ausgabeformat

CSV-Trennzeichen

|

Welches Trennzeichen zwischen den Zellen soll bei der CSV-Ausgabe genutzt werden?
Standard ist: |
Diese Einstellung ist nur beim Ausgabeformat CSV gültig

Text-Trenner

*

Welches Trennzeichen soll als Text-Trenner bei der CSV-Ausgabe genutzt werden?
Standard ist: *
Diese Einstellung ist nur beim Ausgabeformat CSV gültig

Export-Verzeichnis

/export

In welchem Verzeichnis sollen die Exportdateien abgelegt werden?
Das angegebene Verzeichnis MUSS EXISTIEREN und 777er Dateirechte haben und auf "/" enden!

Cron-Schlüssel

XyconsCron

Hinterlegen Sie einen zufälligen Schlüssel, der zum Starten als Cron-Job übereinstimmen muss.
So etwas von der Art wie oben der Systemschlüssel.

Wie gewohnt sind die Funktionen und Wirkungsweisen der Setup-Optionen direkt bei der Option beschrieben.

Modul-Hauptfunktion

Das Modul ermöglicht die Nutzung vorgefertigter und hinterlegter SQL-Abfragen und ermöglicht auch vorgefertigte Abfragen an eigene Bedürfnisse anzupassen oder eigene Abfragen hinzuzufügen. Die im Shop vorbereiteten SQL-Abfragen finden Sie im Menüpunkt „SQL-Reports“ zur Auswahl in dem dortigen Pull-Down-Menü.

Nach Auswahl der gewünschten SQL-Abfrage erscheinen die jeweils auf die Abfrage bezogenen Selektoren, welche eine Eingrenzung nach vorgegebenen Kriterien ermöglicht. Das Modul unterstützt hierbei durch vorbereitete Auswahllisten oder Eingabefelder und leitet die hier gemachten Auswahlen bei der Ausführung an das SQL-Statement weiter.

SQL-Reports

Report: Kunden nach Postleitzahl [customers_by_zip]

Kundennummer Start:

Kundennummer Ende:

PLZ Start:

PLZ Ende:

Sprache: [2] - Deutsch

Nach der Ausführung des Reports erscheinen die durch die Selektionskriterien eingegrenzten Ergebnisse in der Anzeige. Sie können die Ergebnisse nun ansehen oder auch über den Button „EXPORTIEREN“ in dem über das Modul-Setup voreingestellte Formate CSV oder XLS exportieren:

SQL-Reports

Kundennummer	Kunde	Nachname	Vorname	eMail	Kundengruppe	Geschlecht	Telefon
1904					Gast		
1883					Gast		
1928					Gast		
1880					Admin	m	
...							
1520					Neuer Kunde	f	
807					Neuer Kunde		
365					Neuer Kunde		
880					Neuer Kunde		
892					Neuer Kunde		

Wenn Sie die Ergebnisse EXPORTIEREN generiert das Modul eine entsprechende Ausgabedatei und legt diese im Ordner

/export

Des Shops ab. Nach der Erzeugung der Ausgabedatei erscheint auch ein Download-Link, über den die erzeugte Datei direkt heruntergeladen werden kann. Die generierte Datei trägt immer den Kurznamen des Reports gefolgt von Datum und Uhrzeit der Erstellung der Datei:

SQL-Reports

The screenshot shows a web interface for 'SQL-Reports'. It features a 'Report:' label followed by a dropdown menu. The dropdown menu is open, displaying the text: 'Ihr letzter Export: http://127.0.0.1/_GX2.1/swgx20012/export/Rprt_customers_by_zip_31-01-2016_14-34-34.xlsx'. Below this text, a smaller line of text reads: 'Sie finden die Datei auch in folgendem Verzeichnis auf Ihrem Webserver: /export'. At the bottom of the interface, there is a 'RESET' button.

Als Link angezeigt wird immer der letzte erzeugte Report.

Bereits vorher generierte Reports können Sie jederzeit z. B. per FTP aus dem Ordner */export* herunterladen.

Der Abfragen-Editor

Über den SQL-Abfragen-Editor können Sie:

- ➔ Bereits vorhandene (auch die mitgelieferten) Report verändern, anpassen und löschen
- ➔ Neue eigene Reports erstellen und zur Nutzung hinterlegen

SQL-Abfragen-Editor



SQL-Reports

Report:

Beachten Sie bei der Arbeit die jeweiligen Tooltip der möglichen Eingabefelder; diese geben wertvolle Hinweise welche Hinterlegung im jeweiligen Feld vorzunehmen ist.

SQL-Abfragen-Editor



SQL-Reports

Kurzname:

Beschreibung:

Selektoren:

SQL-Befehl:

```
select distinct
    products.products_model as SKU,
    products.products_quantity as SKU_Qty,
    products.description.products_name as Name,
    shipping_status.shipping_status_name as ShippingStatus
from
    products, products_description, shipping_status
where
    products.products_id = products_description.products_id
and
    products.products_shippingtime = shipping_status.shipping_status_id
and
    shipping_status.language_id = '@language@'
and
    products_description.language_id = '2';
```

SQL-Befehl VORHER:

SQL-Befehl HINTERHER:

Stellen Sie sicher, dass ihre SQL-Befehle funktionieren.
Testen Sie diese ggf. vorher z. B. in MyPHPAdmin.
Das Tool führt KEINE Syntax-Prüfung durch; die Befehle müssen syntaktisch und formell korrekt sein.
Verwenden Sie KEINE Befehle, über deren Auswirkungen Sie sich nicht absolut sicher sind.

Welche Selektoren werden benötigt?
Trennen Sie mehrere Selektoren durch ein Semikolon voneinander.
Jeder Selektor erzeugt entsprechende Eingabefelder.

Mögliche Selektoren sind:
product - ein Produkt -> @products_id@
products - Produkt von/bis -> @products_id_start@ / @products_id_end@
products_models - Artikelnummer von/bis -> @products_model_start@ / @products_model_end@
manufacturers - Hersteller von/bis -> @manufacturers_start@ / @manufacturers_end@
dates - Datum von/bis (Kalender) -> @dates_start@ / -> @dates_end@
ndates - Datum (von/bis) direkte Eingabe -> @dates_start@ / @dates_end@
month - Monat von/bis -> @month_start@ / @month_end@
customers - Kunden von/bis -> @customers_id_start@ / @customers_id_end@
customers_group - Kundengruppe von/bis -> @customers_group_start@ / @customers_group_end@
zips - Postleitzahlen von/bis -> @zip_start@ / @zip_end@
orders - Bestellungen von/bis -> @orders_start@ / @orders_end@
orders_status - Bestellstatus von/bis -> @orders_status_start@ / @orders_status_end@
couponcode - 1 Kuponcode -> @coupon_code@
couponcodes - Kuponcodes von/bis -> @coupon_start@ / @coupon_end@
value1 - Eingabefeld Wert1 -> @value1@
value2 - Eingabefeld Wert2 -> @value2@
value3 - Eingabefeld Wert3 -> @value3@

Hinterlegen Sie folgende Werte bzw. bearbeiten Sie die Inhalte bei bereits vorhandenem Report analog.

Das Feld **KURZNAME**

Hier erfassen Sie einen Kurznamen für den Report. Der Kurzname muss eindeutig sein. Unter diesem Kürzel wird der Report verwaltet und der Wert ist Bestandteil des Dateinamens, wenn die Ergebnisse aus diesem Report exportiert werden. Nutzen Sie hier keine Sonderzeichen, da diese sonst bei der Nutzung des Wertes als Dateinamen zu Fehlern führen können. Beschränken Sie sich daher auf den Zeichenvorrat ,0' bis ,9' und ,a' bis ,z' bzw. ,A' bis ,Z', ersetzen Sie Leerzeichen durch Unterstriche.

Das Feld **BESCHREIBUNG**

Hier erfassen Sie eine kurze prägnante Beschreibung des Reports. Dieser Wert wird später als Kurzbeschreibung genutzt.

Das Feld **SELEKTOREN**

Sofern bestimmte Werte zur Selektion der Daten an das SQL-Statement weitergegeben werden sollen, hinterlegen Sie die vorgefertigten Selektoren durch Semikolon getrennt in diesem Feld. Der Tooltip gibt an, welche Selektoren benutzt werden können, da diese über das Modul mitgeliefert wurden:

Welche Selektoren werden benötigt?

Trennen Sie mehrere Selektoren durch ein **Semikolon** voneinander.
Jeder Selektor erzeugt entsprechende Eingabefelder.

Mögliche Selektoren sind:

product - ein Produkt -> @products_id@
products - Produkt von/bis -> @products_id_start@ / @products_id_end@
products_models - Artikelnummer von/bis -> @products_model_start@ / @products_model_end@
manufacturers - Hersteller von/bis -> @manufacturers_start@ / @manufacturers_end@
dates - Datum von/bis (Kalender) -> @dates_start@ / -> @dates_end@
mdates - Datum (von/bis) direkte Eingabe -> @dates_start@ / @dates_end@
month - Monat von/bis -> @month_start@ / @month_end@
customers - Kunden von/bis -> @customers_id_start@ / @customers_id_end@
customers_group - Kundengruppe von/bis -> @customers_group_start@ / @customers_group_end@
zips - Postleitzahlen von/bis -> @zip_start@ / @zip_end@
orders - Bestellungen von/bis -> @orders_start@ / @orders_end@
orders_status - Bestellstatus von/bis -> @orders_status_start@ / @orders_status_end@
couponcode - 1 Kuponcode -> @coupon_code@
couponcodes - Kuponcodes von/bis -> @coupon_start@ / @coupon_end@
value1 - Eingabefeld Wert1 -> @value1@
value2 - Eingabefeld Wert2 -> @value2@
value3 - Eingabefeld Wert3 -> @value3@
value4 - Eingabefeld Wert4 -> @value4@
value5 - Eingabefeld Wert5 -> @value5@

Hinter den Selektoren finden Sie die Felder, welche innerhalb der SQL-Statements als Platzhalter für den jeweiligen Wert genutzt werden können. Haben Sie beispielsweise den Selektor „product“ hinterlegt, so steht Ihnen im folgenden SQL-Befehl ein Platzhalter @products_id@ zur Verfügung, anstelle dieses Platzhalters wird dann bei der Ausführung des Befehls der Wert aus dem Eingabefeld für das Produkt eingefügt. Auf diese Weise können Sie die Eingabewerte der genutzten Selektoren an den SQL-Befehl weitergeben und dort verwenden. Beachten Sie, dass es unterschiedliche Selektoren gibt für Einzelwerte und für Von-/Bis-Werte einer Entität.

Eine Besonderheit sind die 5 frei verfügbaren Selektoren „value1“ bis „value5“:

Wenn Sie diese Selektoren benutzen, erscheinen Eingabefelder, um die jeweiligen Werte einzugeben.
Bei reiner Verwendung der Selektoren erscheinen die Eingabefelder mit der gleichnamigen Beschriftung.

Der Selektor „value2“ erzeugt dann ein Eingabefeld mit der Beschriftung „value2“.

Sie können die Beschriftung bei diesen 5 Selektoren selbst festlegen, wenn Sie direkt hinter dem Selektor folgende Syntax benutzen, z. B.:

value1[MeinFeldname];value2[MeinZweiterFeldname]

Ist diese Syntax befolgt erscheint das Feld für „value1“ mit der angegebenen Beschriftung: „MeinFeldname“ und das Eingabefeld für „value2“ erscheint mit der Beschriftung „MeinZweiterFeldname“.

Das Feld **SQL-Befehl**

In diesem Feld hinterlegen Sie den jeweiligen SQL-Befehl, der zur Sektion der gewünschten Daten führt. Für eigene Reporthinterlegungen sollten Sie entsprechend die Skriptsprache SQL beherrschen.

Verbauen Sie innerhalb des SQL-Befehls die Wertefelder der genutzten Selektoren, damit die Werte benutzt werden können.

Im nachfolgenden Beispiel wurden die Selektoren „manufacturers“ und „language“ verbaut:

Selektoren: manufacturers;language

SQL-Befehl:

```
SELECT
    p.products_model AS ArtikelNr,
    p.products_id AS ArtikelID,
    p.products_name AS ProduktName,
    m.manufacturers_name AS Hersteller
FROM
    products p, products_description pd, manufacturers m
WHERE
    p.products_id = pd.products_id
    AND
    pd.language_id = @language@
    AND
    m.manufacturers_id = @manufacturers_start@
    AND
    m.manufacturers_id <= @manufacturers_end@
ORDER BY
    m.manufacturers_name, products_model
```

Entsprechend der oben dargestellten Selektoren und Wertefelder führt die Nutzung des Selektors „manufacturers“ zu Eingabe-Möglichkeiten für den Hersteller (Von-/bis)

manufacturers - Hersteller von/bis -> @manufacturers_start@ / @manufacturers_end@

Die Eingabefelder werden in den Platzhaltern „@manufacturers_start@“ und „@manufacturers_end@“ an den SQL-Befehl übergeben und wurden daher an der notwendigen Stelle im SQL-Befehl eingebaut, damit sie an dieser Stelle erscheinen.

Hinweise:

Der SQL-Befehl wird später – so wie er ist – ausgeführt nach evtl. Eintragung der Selektoren-Werte. Die hierbei erzeugte Ausgabe entspricht **1:1 dem Report-Ergebnis**. Wenn Sie also eine Tabellenspalte innerhalb der Ergebnismenge mit einem eigenen Namen wiederfinden möchten, dann müssen Sie mit den Mitteln der SQL-Scripting Sprache z. B. die Möglichkeiten der Namensgebung über den Befehl „AS {MeinWertName}“ benutzen.

Im obigen Beispiel wäre das:

```
SELECT
    p.products_model AS ArtikelNr,
    p.products_id AS ArtikelID,
    pd.products_name AS ProduktName,
    m.manufacturers_name AS Hersteller
```

Die Spalte „products_model“ wird als Spalte „ArtikelNr“ ausgegeben.

Die Spalte „products_id“ wird als Spalte „ArtikelID“ ausgegeben.

u.s.w.

Der Editor überprüft NICHT die Syntax des hinterlegten SQL-Befehls. Das Modul übergibt den hinterlegten Befehl so wie er ist später während der Ausführung an die Datenbank. Sie müssen daher in eigenen oder veränderten SQL-Befehlen selbst sicherstellen, dass der Befehl korrekt definiert und somit auch ausführbar ist. Hat der hinterlegte Befehl syntaktische Fehler führt die Ausführung ggf. zu einer leeren Ergebnismenge oder einen Fehler. Eine gute Vorgehensweise ist den Befehl vor der Nutzung in einem Report z. B. über ein Datenbanktool wie „phpMyAdmin“ auf korrekte Syntax und Richtigkeit zu testen.

Das Feld **SQL-Befehl VORHER** und **SQL-Befehl HINTERHER**

In diesen Feldern können Sie in sich geschlossene SQL-Befehle hinterlegen, die VOR und NACH dem Haupt-SQL im Feld „SQL-Befehl“ ausgeführt werden müssen. Die Felder sind dafür vorgesehen z. B. VORHER eine temporäre Tabelle anzulegen, auf die sich dann im Haupt-SQL-Befehl bezogen werden kann. Der Bereich NACHHER löscht und verwirft dann die jeweilige Tabelle wieder. Die Ergebnisse der VORHER und NACHHER -Befehle fließen NICHT in das Ausgabeergebnis ein; dieses wird nur über den Haupt-SQL-Befehl erzeugt. Daher nutzen Sie diese beiden Bereiche nur dafür, wofür sie definiert, wurden: Als VOR- und NACH-LAUF zum Haupt-SQL-Befehl.

Beispiel:

SQL-Befehl VORHER:

```
CREATE TEMPORARY TABLE tmp_unique_ptt ENGINE=INNODB SELECT DISTINCT products_id, categories_id FROM products_to_categories GROUP BY products_id;
```

SQL-Befehl HINTERHER:

```
DROP TABLE tmp_unique_ptt;
```

Im Beispiel wird VOR dem eigentlichen Haupt-SQL-Befehl eine temporäre Tabelle „tmp_unique_ptt“ erzeugt. Die Ergebnismenge dieser temporären Tabelle steht nun zur Nutzung im Haupt-SQL-Befehl zur Verfügung und kann dort genutzt werden. Im Nach-Lauf wird die zuvor erzeugte temporäre Tabelle wieder verworfen.

Grundsätzlich gilt:

Stellen Sie sicher, dass ihre SQL-Befehle funktionieren.

Testen Sie diese ggf. vorher z. B. in MyPHPAdmin.

Das Tool führt KEINE Syntax-Prüfung durch; die Befehle müssen syntaktisch und formell korrekt sein.

Verwenden Sie KEINE Befehle, über deren Auswirkungen Sie sich nicht absolut sicher sind.

Reports als Cron-Job

Sie können die lauffähigen Report-Vorlagen auch als sog. CronJob ausführen lassen. Die jeweiligen Ergebnisdateien werden auch hier in */export* abgelegt, wenn der CronJob ausgeführt wurde.

Es gilt folgende Befehls-Syntax:

{IhreURL}/M0012_sql_reports.php?cron={CronKey}&action=report&report={ReportKurzname}&{SELEKTOREN}

Hierbei gilt:

{CronKey}

Der im **Setup** als „Cron-Schlüssel“ hinterlegte individuelle Kennung:

Cron-Schlüssel

\$xycons2016\$

Hinterlegen Sie einen zufälligen Schlüssel, der zum Starten als Cron-Job übereinstimmen muss.
So etwas von der Art wie oben der Systemschlüssel.

Hier wäre dann der Wert „\$xycons2016\$“ einzufügen als Passwort für den CronJob

{ReportKurzname}

Der Kurzname des gewünschten Reportes

Kurzname:

product_bought_by_cat

Hier wäre dann der Wert „products_bought_by_cat“ anzugeben

{SELEKTOREN}

Alle notwendigen Selektoren des Reports, der als Cron-Job ausgeführt werden soll aneinandergereiht und mit „&“ verbunden gemäß der Syntax zum Aufruf von Skripten mit Übergabeparametern:

Selektor: @products_models@ → Felder @products_model_start@ / @products_model_end@
= &products_model_start=ABC&products_model_end=XYZ

Beispiel für den Report „Liste der Kundengeburtstage von/bis“ vom 01.01.1970 bis 31.12.1979 mit CronKey „1234567890“:

M0012_sql_reports.php?cron=1234567890&action=report&report=customers_dob&start_date=01.01.1970&end_date=31.12.1979

Restriktionen bei Cron:

- Das Befehlsfeld „SQL-Befehl“ darf nur einen SQL-Befehl beinhalten; eine Abtrennung weiterer Befehle mit Semikolon ist hier nicht zulässig

Datenverändernde „Reports“

ACHTUNG!

Das Tool kann - je nach Anwendung - Änderungen an der Datenbank, den Daten oder Datentabellen durchführen. Wir übernehmen keinerlei Haftung für Beschädigungen oder Datenverluste, die aus der Nutzung von verändernden Datenbankabfragen oder anderen auf die Datenbank schreibend oder verändernd wirkenden Abfragen herrühren.

Benutzen Sie verändernde SQL-Abfragen nur, wenn Sie sich über die Auswirkungen der jeweiligen Abfrage im Klaren sind und ausreichende Datensicherung durchgeführt haben!

Für die Ergebnisse dieser Abfragen haften Sie grundsätzlich selbst!

Grundsätzlich können Sie auch SQL-Befehle zur Ausführung hinterlegen, welche zu Datenveränderungen führen. Der SQL-Befehl wird 1:1 so ausgeführt wie hinterlegt. Es spricht also grundsätzlich nichts dagegen auch SQL-Befehle zu nutzen, die z. B. über „UPDATE“-Befehle zu Datenveränderungen oder Datenlöschungen führen oder neue Datensätze anlegen.

Für solche Reports und die damit durchgeführten Datenveränderungen sind Sie grundsätzlich selbst verantwortlich. Beachten Sie, dass Ihre eigenen datenverändernden Befehle zu fehlerhaften Daten, Datenverlust oder Problemen im Shopsystem führen können.

Wir wünschen Ihnen viel Erfolg beim Einsatz dieses Moduls!