addons presta

Database Cleaner & Optimizer

USER MANUAL · Version 1.0.0 · PrestaShop 1.7.8 – 9.x

1. What this module does

Database Cleaner & Optimizer shows the tables that make your PrestaShop database grow — visitor statistics, logs, sent e-mails, guests, abandoned carts — and cleans them by hand or by cron: everything, or everything but the last days or the newest rows. It also optimizes tables, finds orphan rows and checks that the core tables have the indexes of your PrestaShop version. A smaller database means faster backups, less disk space and quicker pages that read these tables.

Works with **PrestaShop 1.7.8, 8.x and 9.x** (PHP 7.2.5 or newer) and with multistore. Tested on MySQL 8.0.

2. Installation

1. Download the module from **My account** → **Order history** at addonspresta.com.
2. In your Back Office, open **Modules** → **Module Manager** and click **Upload a module**.
3. Drop the `ap_dboptimizer_v1.0.0.zip` file and wait for the confirmation.
4. Click **Configure** on the module card.

If you have shell access, you can also unpack the zip into `modules/` and run `php bin/console prestashop:module install ap_dboptimizer` from the shop folder.

The installation deletes nothing: it saves the default rules, with the cron off for every group.

3. How it works

The module reads the size and the number of rows of every table from the database server and groups the tables that grow on their own into cards. Each card has a colour, based on the number of rows in its tables: green means OK, yellow large, red very large. You decide what to delete, card by card, or let a cron do it every night with the rules you set.

Make a backup of the database before you clean anything. The module does not make backups and cannot restore deleted rows. The page links PrestaShop's backup page (**Advanced Parameters** → **Database** → **DB Backup**), and every deletion asks you to tick **I have a current backup of the database** first.

What the module never deletes: orders, customers, addresses, products and other business data, carts that became an order, discount codes that are still valid, specific prices without an end date, and your own Back Office session. Abandoned carts, discount codes and specific prices are deleted through PrestaShop's own functions, so uploaded files are removed with them and other modules are told about it.

Large tables are cleaned in small chunks. When the time limit of one run ends, the card says how many rows were deleted and that more remain; you click **Continue**, and the cron goes on at its next call. After a cleaning the module refreshes the statistics of the tables, so the card shows the new numbers at once.

The module does not repair tables or change the settings of your database server. It does not connect to addonspresta.com or any other server.

4. Features and settings

Open **Modules** → **Module Manager** → **Database Cleaner & Optimizer** → **Configure**. The **Help** box at the top links this manual, the module page at addonspresta.com and our support address, and shows the installed version. Below it, a panel reminds you to make a backup, with the link **Open the backup page of PrestaShop**.

Database overview

The size of the database, the number of tables and **The 10 largest tables** with their rows, size and engine. When some tables do not use InnoDB, a warning lists them: such tables lock the whole table on every write and do not recover after a crash.

The row numbers of InnoDB tables are estimates given by MySQL; small tables are counted exactly.

Data to clean

One card per data group, with its tables, rows, size and colour:

- **Visitor statistics** — every visit with its referrer and viewed pages (`connections` and the tables that go with it), used by the statistics of the Back Office.
- **Page view counters** — how many times each page was viewed in each period of the statistics.
- **Guests** — visitor profiles (browser, system). Only guests without a customer account, a cart or a recorded visit are deleted.
- **Logs** — the entries of **Advanced Parameters** → **Logs**.
- **E-mail log** — the list of sent e-mails in **Advanced Parameters** → **E-mail**.
- **Pages not found** — addresses that answered "page not found", recorded by the statistics module of that name. The card appears only when the module's table exists.
- **Shop searches** — what visitors typed into the shop search, recorded by the search statistics module. The card appears only when its table exists.
- **Abandoned carts** — carts that never became an order. A cart with an order or changed in the last 7 days is never deleted. The card carries a warning: abandoned-cart reminder modules and

statistics use these carts, so clean them only when you do not need them.

- **Expired discount codes** — cart rules whose end date has passed, for example one-time codes created by modules.
- **Expired specific prices** — specific prices whose end date has passed. A price without an end date is never deleted.
- **Old sessions** — login sessions of customers and employees. Deleting a session signs its user out, so only old sessions or sessions of deleted accounts are removed. Your own session always stays. On PrestaShop 1.7.8 only sessions of deleted accounts can be removed.
- **Smarty cache in the database** — entries of the template cache. They are cleared with PrestaShop's own cache clearing and rebuild themselves.

On each card you choose how to clean. Depending on the group:

- **Delete everything**;
- **Keep the last days (number)** — everything older than the number of days you enter is deleted;
- **Keep the newest rows (number)** — only the given number of the newest rows stays;
- for carts **Abandoned for more than (days)** (at least 7), for discount codes and specific prices **Expired more than (days) ago**, for sessions **Not used for more than (days)** or **Sessions of deleted accounts**, for guests **All unused guests**, for the Smarty cache **The whole cache**.

The buttons of a card:

- **Preview** — shows **Rows this cleaning would delete:** with the exact number, without deleting anything.
- **Clean** — deletes. Tick **I have a current backup of the database** first, otherwise nothing is deleted. When the time limit ends before the work is done, the button becomes **Continue**.
- **Optimize tables** — rebuilds the tables of the card so that the freed space goes back to the disk. Deleting rows alone makes the tables emptier, not smaller on disk. A table larger than the heavy statement limit is not optimized from the page: the card shows the SQL statement to run from a database console (phpMyAdmin, the console of your hosting).

Orphan rows

Rows whose parent record no longer exists: cart lines without a cart, translations of deleted records, stock and price rows of deleted products. The module checks 43 such relations. Orders, customers, addresses and images are never touched.

1. Click **Scan for orphan rows**. The result lists the relations where orphan rows were found, with their number; it is kept on the page until the next scan, with its date under **Last scan:**.
2. Tick **I have a backup** and click **Delete** next to a relation, or tick **I have a current backup of the database** and click **Delete all orphan rows**.
3. When the time limit ends before all rows are gone, click **Delete** again to go on.

Orphan rows are deleted only from this panel, never by the cron.

Indexes

The module compares the indexes of the core tables with those a fresh installation of your PrestaShop version creates, and says which list it used (**Compared with the index list of a fresh PrestaShop ... installation**). An index with another name, or a longer one starting with the same columns, counts as present. Indexes added by modules are not checked.

When nothing is missing the panel says **Every listed index is in place.** Otherwise each missing index is shown in red with **SQL to add it**. For a plain index on a table under the heavy statement limit, tick **I have a backup** and click **Add index**. A unique or primary key, or a larger table, shows **SQL only**: a unique key fails when the table holds duplicate values, and a large table would block the page, so run the statement from a database console.

Automatic cleaning (cron)

The panel shows the cron address and a ready line for the crontab of your server, for example:

```
0 3 * * * curl -fsSL "https://your-shop.com/module/ap_dboptimizer/cron?token=…"
>/dev/null
```

Copy the address from the panel and call it from a scheduled task of your server or your hosting panel, for example once a night. Each call cleans the groups ticked in the rules, each with its rule, and continues at the next call when the time limit ends. The panel shows **Last run:** with the rows deleted per group, or **The cron has not run yet.**

New token creates a new address: the old one stops working at once, so replace it in your scheduled task. Use it when the address may have leaked.

Rules and settings

One row per data group:

- **Clean by cron** — the cron cleans this group. Off for every group after installation; for abandoned carts, think twice before you tick it.
- **How** and **Days or rows** — the cleaning mode and its number, used by the cron and offered on the card.
- **Yellow from (rows)** and **Red from (rows)** — the thresholds of the colour, counted in rows of the group's tables. Yellow must be lower than red.

Below the rows:

- **Optimize the tables after a cron cleaning (gives the freed space back to the disk)** — after a cron run that deleted at least 10 000 rows of a group, its tables are optimized (only those under the heavy statement limit). Off by default.
- **Time limit of one run (seconds)** — 5 to 120, default 20. A cleaning stops after this time and continues when you click **Continue** or at the next cron call. Keep it below the time limit of your web server.

- **Heavy statement limit (MB)** — default 300. Optimizing a table or adding an index cannot be stopped once it started. Tables larger than this are not optimized or indexed from the page or the cron: the SQL statement is shown instead.

Click **Save**.

5. Multistore

The database tables hold the data of every shop, so the module has one set of rules for the whole installation, whatever shop is selected in the Back Office header. Cleaning or changing anything needs an employee who manages all shops; others see the page with a notice. The cron cleans the whole installation. One licence covers every shop of the installation.

6. Uninstall

Module Manager → Database Cleaner & Optimizer → Uninstall removes the module's settings. Rows already deleted stay deleted and indexes it added stay in place, as they belong to PrestaShop's own structure. **Disable** keeps the settings and stops the cron address.

7. FAQ and troubleshooting

I cleaned a group, but the database on disk is not smaller. MySQL keeps the space of deleted rows inside the table for new rows. Click **Optimize tables** on the card to give it back to the disk, or turn on the optimization after a cron cleaning.

The card shows an SQL statement instead of optimizing the table. The table is larger than **Heavy statement limit (MB)**. Optimizing it can take minutes and cannot be stopped, so the module leaves it to you: run the statement from phpMyAdmin or the database console of your hosting at a quiet time, or raise the limit if your server handles it.

The cleaning stopped with a message that more rows remain. The time limit of one run ended. Click **Continue** until the card says it is done, or let the cron go on. Raise **Time limit of one run (seconds)** only if your web server allows longer requests.

The number of rows on the card differs from the preview. The card shows the estimate MySQL keeps for large tables; **Preview** counts the rows exactly.

Can I clean abandoned carts safely? Carts with an order and carts changed in the last 7 days are never deleted. Modules that send reminders about abandoned carts and statistics of abandoned carts need older carts too: keep enough days for them, and leave the cron off for carts if you are not sure.

A customer or an employee was signed out after cleaning sessions. Deleting a session signs its user out. The module removes only sessions older than the lifetime of the login cookie and the number of days you set, or sessions of deleted accounts, and never yours.

The cron address answers "invalid token". The address is not the current one, for example after **New token**. Copy it again from the **Automatic cleaning (cron)** panel.

The cron answers "busy". Another cleaning was running at the same moment (the cron itself or someone on the page). The next call does the work.

The index panel shows a missing index. Is my shop broken? No, but queries on that table may be slower. A missing index is often left over from an old upgrade. Add it with the button or the SQL, at a quiet time and after a backup.

I deleted something by mistake. The module keeps no copy of what it deletes. Restore your backup.

8. Updates and support

Updates are free for life: download new versions from **My account → Order history** at addonspresta.com and upload them like a new module. Your rules and settings are kept. The module's code passes our AI-assisted security audit.

Support is included for 12 months from your order: write to **mail@addonspresta.com** or use the contact form at addonspresta.com (Monday to Friday, 10:00–16:00; we usually answer within 2 business days). Please include your PrestaShop version, the module version, the message the page showed and the steps that lead to the problem.