

# MySQL Limitations on the Flat Catalog (Product)

## Problem

A table in MySQL has a [limitation of 65535 bytes of overall row length](#). It may severely limit the number of columns in a table when it has varchar/char columns.

In Magento starting from 1.3 the products catalog in the "flat" mode suffers from this limitation depending on the number and combination of the product attributes that participate in the flat product index.

## Background

A product attribute gets into the flat tables and gets affected by the issue in the following conditions:

- The flat product catalog is enabled in "Sys Config -> Catalog -> Frontend -> Use Flat Catalog Product"
- When a product attribute is added with the following parameters:
  - Input Type – one of:
    - Text Field
    - Date
    - Yes/No (boolean)
    - Multiple Select
    - Dropdown
    - Price
    - Media Image
    - Fixed Product Tax
  - Frontend Properties – "Yes" one of: "Used in Product Listing", "Use in Layered Navigation", "Used for Sorting in Product Listing"
- When the cache is refreshed and the flat catalog index is refreshed

Result:

```
"SQLSTATE[42000]: Syntax error or access violation: 1118 Row size too large. The maximum row size for the used table type, not counting BLOBs, is 65535. You have to change some columns to TEXT or BLOBs".
```

## Estimated Limits

The limits are calculated with the assumption that the table has utf8 encoding and the columns would fill all 65535 bytes (in reality this number is decreased by the static attribute columns).

Input Type	Type in the Database	Size Formula	Maximum Qty
Text Field Multiple Select Media Image	varchar(255)	$(255 + 1) \times 3 = 766$ bytes	85
Date	date	3 bytes	4096
Yes/No Dropdown	int(11)	4 bytes	4096
Price Fixed Product Tax	decimal(12,4)	$4 + 2 = 6$ bytes	4096

## Solution

### Workarounds

- Minimize usage of the varchar (Text Field, Multiple Select) product attributes on the frontend by avoiding to put them into flat index: "Used in Product Listing" = No, "Use in Layered Navigation" = No, "Used for Sorting in Product Listing" = No
- Use "Dropdown" input type instead of "Multiple Select" (int instead of varchar)

## Optimization

Requires either core code modification or a customization.

Some proposed (not verified) solutions:

- Reduce the "Multiple Select" storage to varchar(128): this would allow to store up to  $65535 / (128 * 3 + 1) = 170$  attributes
- Switch the "Text Field" and "Media Image" types to some blob format – not affected by the 65535 length, but performance will suffer on searching/filtering by this attributes on the frontend
- Do the above and switch the table encoding to ANSI:  $65535 / (128 + 1) = 508$  attributes
- Or a completely alternative solutions:
  - move out the "Multiselect" or all varchar attributes to other table with various optimizations
  - dissolve the "Multiselect" storage into a table with int values (cardinality & implementation logic change)