

MySQL Coding Standards and Naming Conventions

General rules for MySQL database tables, fields, indices and queries.

MySQL Naming Convention

Database

Database name must have the same name as project name. If the project name is "My Web Site" database should be created as:

```
CREATE DATABASE mywebsite DEFAULT CHARSET UTF8;
```

Tables

All tables must be UTF8 encoded. All tables in the same package must have prefix. Use 2 or 3 letters that describe the package to prefix database. If the project name is Real Deal Marketing, the most obvious prefix will be "rdm_";

```
CREATE TABLE rdm_affiliates;
```

ALTER TABLE and file versions major.minor[.build[.revision]]

All alterations for all tables should be saved in a **(database_name).sql** file. If database name is "realdealmarketing" filename should be realdealmarketing-1.0.0.sql

1. Initial file must be version 1.0.0
2. If there are only **table alterations** file should have new version increased by 0.0.1
Example: **realdealmarketing-1.0.7.sql**
3. If **new tables are created**, version must be increased by 0.1.
Example: **realdealmarketing-1.2.0.sql**
4. If **tables are dropped**, version must be increased by 1.0.
Example: **realdealmarketing-2.0.0.sql**
5. All **minor revisions** should have appropriate file name and version following this pattern: **major.minor[.build[.revision]]**
6. All versions should be saved in the same directory

Fields

This section explains how to create database table fields and how to choose name for each field.

Field Names

1. Field names must be prefixed with 2-4 letters of table name.
2. Field names are always lowercase with "_" to separate words

Example

```
DROP TABLE IF EXISTS rdm_affiliates;
```

```
CREATE TABLE IF NOT EXISTS rdm_affiliates
```

```
(
```

```
  aff_id          INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
```

```
  aff_url         VARCHAR(120) NOT NULL DEFAULT '',
```

```
  aff_title       VARCHAR(120) NOT NULL DEFAULT '',
```

```
  aff_website     CHAR(10) NOT NULL DEFAULT '',
```

```
  aff_gender      CHAR(8) NOT NULL DEFAULT '',
```

```
  aff_landing_page VARCHAR(120) NOT NULL DEFAULT '',
```

```
  aff_link        VARCHAR(120) NOT NULL DEFAULT '',
```

```
  aff_text        TEXT,
```

```
  PRIMARY KEY (aff_id),
```

```
  INDEX (aff_website)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

All field names must be descriptive, avoid names like "unique", "sort" and reserved words.

NULL and DEFAULT

All fields except AUTO_INCREMENT, TEXT, DATE (and similar) must be defined as **NOT NULL DEFAULT 'value'**

Examples:

```
aff_title       VARCHAR(120) NOT NULL DEFAULT ''
```

```
aff_gender      ENUM('male','female') NOT NULL DEFAULT 'male'
```

```
user_id        INT UNSIGNED NOT NULL DEFAULT 0
```

```
user_birthday   DATE DEFAULT NULL
```

```
user_allow_newsletter TINYINT(1) UNSIGNED NOT NULL DEFAULT 0
```

```
art_text        TEXT
```

Indices

All fields in WHERE and ORDER BY should be defined as INDEX. If you have query like this:

```
SELECT * FROM table1 WHERE a='something' ORDER BY b
```

Then fields a, b should be considered as indices

```
INDEX index_name (a,b)
```

CHAR Indices

If text field (CHAR, VARCHAR) is used as index for larger tables (> 5000 records) should be always defined as CHAR. For example, if you have fields like this:

```
aff_url VARCHAR(255) NOT NULL DEFAULT ''
```

```
...
```

```
INDEX affiliate_url (aff_url)
```

To improve performance, this table should be changed to something like this

```
aff_url CHAR(20) NOT NULL DEFAULT ''
```

```
...
```

```
INDEX affiliate_url (aff_url)
```

Queries

Reserved words should be uppercase to increase readability. Also try to separate long queries in multiple lines, while simple queries should stay in one line. Example:

```
SELECT a.field_name1, a.field_name2, COUNT(a.field3) AS cnt, b.*
```

```
FROM table1 AS a
```

```
LEFT JOIN table2 AS b ON (a.key_field = b.key_field)
```

```
RIGHT JOIN table3 AS c ON (a.key_field2 = c.key_field2)
```

```
WHERE a.field7 = 'something here' AND b.field9 = '45'
```

```
GROUP BY a.field_name1
```

```
ORDER BY a.field_name2 DESC, b.field8
```

```
HAVING cnt > 5
```

Escape

Always escape values in WHERE even when value is integer

PHP variables in queries

PHP variables must be defined and escaped before query.

Avoid:

```
$q = "SELECT * FROM users WHERE email='"._escape($_POST['email'])."' ";
```

Use:

```
$email = _escape($_POST['email']);
```

```
if (empty($email))
```

```
{
```

```
    return false;
```

```
}
```

```
$q = "SELECT * FROM users WHERE email='{$_email}' ";
```