



链滴

MySQL only_full_group_by 1055 报错的 三种解决方案, 临时关闭有影响吗?

作者: [HiJiangChuan](#)

原文链接: <https://ld246.com/article/1638410262855>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



本文首发：[MySQL only_full_group_by 1055报错的三种解决方案，临时关闭有影响吗？](#)

当我们迁移到 MySQL 5.7+ 的版本时，常会碰到 **ERROR 1055 only_full_group_by** 错误，这是 5.7 后 **SQL_MODE** 默认打开了严格模式导致的错误。说明你代码里有地方写的不严谨。

ERROR 1055 (42000): Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'kalacloud.user_id' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by

我看到大多数教程，只写了解决这个问题「术」的部分，并没有讲解什么原因导致这个错误。本教程从原理讲起，先让大家理解为什么会出错。然后给出三种解决方案：「彻底解决」、「临时解决」和「折中解决」，你可根据自己的实际情况进行选择。

SQL_MODE 是什么？

讲 **ONLY_FULL_GROUP_BY** 错误前，我们先来说一下 **SQL_MODE**。理解 MySQL 工作原理能更好帮你理解错误发生的本质原因。

SQL_MODE 是 MySQL 中的一个环境变量，定义了 MySQL 支持的 SQL 语法和数据校验程度。

SQL_MODE 一共三种模式

- ****ANSI 模式**：****宽松模式**。对插入数据进行校验，如不符合定义类型或长度，对保存数据进行截断。
- ****TRADITIONAL 模式**：****严格模式**。对插入数据进行严格校验，保证错误数据不能插入，**ERROR** 报错。用于事物时，事物会进行回滚。
- ****STRICT_TRANS_TABLES 模式**：****严格模式**。对插入数据进行严格校验，错误数据不能插入，**ER OR** 报错。

MySQL 5.7.4 之前, MySQL 默认不开启严格模式

```
卡拉云 kалаcloud.com 低代码开发工具
MariaDB [kалаcloud_demo]> SELECT @@sql_mode;
+-----+
| @@sql_mode |
+-----+
| NO_ENGINE_SUBSTITUTION |
+-----+
1 row in set (0.002 sec)
```

这是 MySQL 升级到5.7.5 之后默认SQL_MODE 为严格模式:

```
卡拉云 kалаcloud.com 低代码开发工具
MariaDB [kалаcloud_demo]> SELECT @@sql_mode;
+-----+
| @@sql_mode |
+-----+
| STRICT_TRANS_TABLES,STRICT_ALL_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,TRADITIONAL, ... |
+-----+
1 row in set (0.003 sec)
```

扩展阅读: [MySQL 连接数过多的处理方法合集 - Too many connections](#)

SQL_MODE 严格模式的意义

在宽松模式下, 即便 insert 一个错误的的数据, MySQL 也会不加判断的全盘接受。

我们来看一个实例

我们首先关闭 SQL_MODE 严格模式

```
set session sql_mode="";
```

然后我们创建一个表并向其中插入一组超范围的数据

```
create table kалаcloud_t1(website char(9));
insert into kалаcloud_t1 values('kалаcloud.com');
```

返回值:

```
卡拉云 kalacloud.com 低代码开发工具

MariaDB [kalacloud_demo]> select * from kalacloud_t1;
+-----+
| website |
+-----+
| kalacloud |
+-----+
1 row in set (0.007 sec)
```

从返回值可以看出，我们向 `website char(9)` 中插入了一条长为 13 的值，没有报错，直接插入，但超 9 的部分，即「.com」被截断丢掉了。

接着我们在严格模式下试试，首先打开 `SQL_MODE` 严格模式：

```
set session sql_mode='TRADITIONAL';

create table kalacloud_t2(website char(9));
insert into kalacloud_t2 values('kalacloud.com');
```

返回值：

```
卡拉云 kalacloud.com 低代码开发工具

MariaDB [kalacloud_demo]> set session sql_mode='TRADITIONAL';
Query OK, 0 rows affected (0.002 sec)

MariaDB [kalacloud_demo]> create table kalacloud_t2(website char(9));
Query OK, 0 rows affected (5.626 sec)

MariaDB [kalacloud_demo]> insert into kalacloud_t2 values('kalacloud.com');
ERROR 1406 (22001): Data too long for column 'website' at row 1
```

我们可以从返回值看出，MySQL 直接报错，告诉你插入的数据有问题。

扩展阅读：[MySQL 配置文件 my.cnf / my.ini 逐行详解](#)

ONLY_FULL_GROUP_BY 问题及解决方案

接着我们来说说 `ONLY_FULL_GROUP_BY` 的问题。当我们数据库迁移至 5.7 或者 8.0 之后，最常见错误就是 `Error 1055 only_full_group_by` 错误。

正如我前文所写，这个错误的关键原因是不规范的 SQL 语法，5.7 之后默认 `SQL_MODE` 变为严格模

。

我们来一起看一个实例，这是一组卡拉云用户点击网页的 log 记录：

```
卡拉云 kалаcloud.com 低代码开发工具  kалаcloud.com

MariaDB [kалаcloud_demo]> create table kалаcloud_website_log (
  → id int auto_increment primary key, page_url varchar(100),
  → user_id int, ts timestamp);
Query OK, 0 rows affected (4.919 sec)

MariaDB [kалаcloud_demo]> insert into kалаcloud_website_log(page_url,user_id,ts)
  → values
  → ('/index.html',1,'2023-07-17 17:21:32'),
  → ('/index.html',2,'2023-07-17 17:21:35'),
  → ('/news.php',1,'2023-07-17 17:22:11'),
  → ('/store_offers.php',3,'2023-07-17 17:22:41'),
  → ('/store_offers.php',2,'2023-07-17 17:23:07'),
  → ('/faq.html',1,'2023-07-17 17:23:22'),
  → ('/index.html',3,'2023-07-17 17:32:25'),
  → ('/news.php',2,'2023-07-17 17:32:38');
Query OK, 8 rows affected (0.065 sec)
Records: 8 Duplicates: 0 Warnings: 0

MariaDB [kалаcloud_demo]> select * from kалаcloud_website_log;
+----+-----+-----+-----+
| id | page_url          | user_id | ts                |
+----+-----+-----+-----+
| 1  | /index.html      | 1       | 2023-07-17 17:21:32 |
| 2  | /index.html      | 2       | 2023-07-17 17:21:35 |
| 3  | /news.php        | 1       | 2023-07-17 17:22:11 |
| 4  | /store_offers.php | 3       | 2023-07-17 17:22:41 |
| 5  | /store_offers.php | 2       | 2023-07-17 17:23:07 |
| 6  | /faq.html        | 1       | 2023-07-17 17:23:22 |
| 7  | /index.html      | 3       | 2023-07-17 17:32:25 |
| 8  | /news.php        | 2       | 2023-07-17 17:32:38 |
+----+-----+-----+-----+
8 rows in set (0.006 sec)
```

现在我们使用 **GROUP BY** 来排序找出访问量最大的网页。

我们先关掉 `sql_mode` 的严格模式来试试：

```
卡拉云 kalacloud.com 低代码开发工具 
MariaDB [kalacloud_demo]> set session sql_mode='';
Query OK, 0 rows affected (0.002 sec)

MariaDB [kalacloud_demo]> SELECT page_url, user_id, COUNT(*) AS visits
→ FROM kalalcloud_website_log
→ GROUP BY page_url ORDER BY COUNT(*) DESC;

+-----+-----+-----+
| page_url      | user_id | visits |
+-----+-----+-----+
| /index.html   | 1       | 3      |
| /store_offers.php | 3       | 2      |
| /blog.php     | 1       | 2      |
| /faq.html     | 1       | 1      |
+-----+-----+-----+
4 rows in set (0.007 sec)
```

在宽松模式下，我们可以看出这个 query 虽然可以查询，但语法和逻辑上稍有问题。我们想对 `page_url` 进行排序，但 query 中也加入了 `user_id`，在返回值中可以发现问题，`index.html` 这个页面不仅 `user_id` 1 的用户访问过，用户 2 和 3 也访问了，那么这张返回的表表格数据就是有问题的。

`user_id` 1 列在返回数据里，到底代表什么？是第一个访问 `index.html` 还是最后一个访问这个页面的意思呢？没人知道，这是个随机盲盒，运行原理未知。

我们打开 `sql_mode` 严格模式跑一下上面这段代码：

```
卡拉云 kalacloud.com 低代码开发工具 
MariaDB [kalacloud_demo]> SET SESSION sql_mode='ONLY_FULL_GROUP_BY';
Query OK, 0 rows affected (0.002 sec)

MariaDB [kalacloud_demo]> SELECT page_url, user_id, COUNT(*) AS visits
→ FROM kalalcloud_website_log
→ GROUP BY page_url ORDER BY COUNT(*) DESC;

ERROR 1055 (42000): 'kalacloud_demo.kalalcloud_website_log.user_id' isn't in GROUP BY
MariaDB [kalacloud_demo]>
```

返回一个 `ERROR 1055` 报错。

`ONLY_FULL_GROUP_BY` 是 `SQL MODE` 中 `TRADITIONAL` 的选项参数，从 5.7 开始默认开启为严格模式。这就是为什么大家迁移到 MySQL 新版会报 1055 错误的原因。

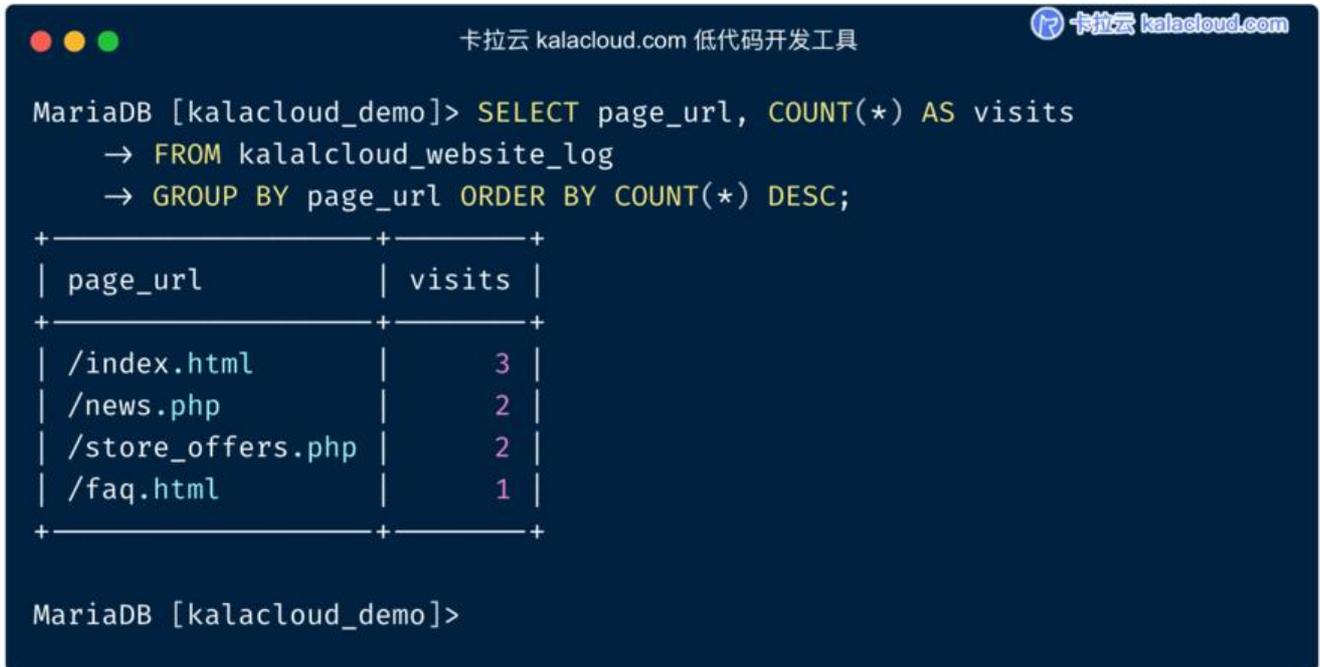
我们已经理解了这个问题的原理原因，接下来，碰到 `ONLY_FULL_GROUP_BY` 报错，我们应该怎么

理呢？

扩展阅读：[如何在 MySQL 中查询数据库中含有某个字段的所有表名](#)

解决方案 1 – 重写代码

找到报错语法中错误的部分，根据逻辑重写 query，本示例中，我们去掉 `user_id` 即可。



```
卡拉云 kалаcloud.com 低代码开发工具
MariaDB [kalacloud_demo]> SELECT page_url, COUNT(*) AS visits
  → FROM kalalcloud_website_log
  → GROUP BY page_url ORDER BY COUNT(*) DESC;
+-----+-----+
| page_url | visits |
+-----+-----+
| /index.html | 3 |
| /news.php | 2 |
| /store_offers.php | 2 |
| /faq.html | 1 |
+-----+-----+

MariaDB [kalacloud_demo]>
```

如果你有大量在旧版 SQL 完成的代码，检查这种错误很有可能是相当浩大的工程。这时就到了你的断时刻了，你有两条路可选：

- (1) 干脆推翻重写，让代码保持严谨和整洁，养成良好的习惯，避免未来出现的未知错误。
- (2) 了解原理。知道可能会出现的问题，关掉严格模式，回到宽松模式，接着用旧版代码。

解决方案 2 – 返回宽松模式

如果你碰到 ERROR 1055 报错，但也不想浩如烟海的代码中纠错，那么你可以回到「宽松模式」着用。

在 MySQL 5.7 及以上版本中 SQL_MODE 包含

ONLY_FULL_GROUP_BY、STRICT_TRANS_TABLES、NO_ZERO_IN_DATE、NO_ZERO_DATE、ERROR_FOR_DIVISION_BY_ZERO、NO_AUTO_CREATE_USER

我们直接在 MySQL 配置文件中更改，或者临时全部关闭：

```
SET GLOBAL sql_mode="";
```

或者单关闭 `ONLY_FULL_GROUP_BY`

```
MariaDB [kalacloud_demo]> SET GLOBAL sql_mode='STRICT_TRANS_TABLES,STRICT_ALL_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,TRADITIONAL,NO_
```

ENGINE_SUBSTITUTION';

解决方案 3 – 使用聚合函数

如果某些特别的原因就是要查询 `user_id`，但你又没空改代码，那么我可使用聚合函数来规避这类错，算是一种折中的方案了，语法改严谨了，代码又不需要大动。

我们可以使用 `MAX()`、`MIN()` 或者 `GROUP_CONCAT()` 聚合函数。

```
卡拉云 kалаcloud.com 低代码开发工具 
```

```
MariaDB [kalacloud_demo]> SET SESSION sql_mode='ONLY_FULL_GROUP_BY';
```

```
MariaDB [kalacloud_demo]> SELECT page_url, MAX(user_id), COUNT(*) AS visits
→ FROM kalalcloud_website_log
→ GROUP BY page_url ORDER BY COUNT(*) DESC;
```

page_url	MAX(user_id)	visits
/index.html	3	3
/news.php	2	2
/store_offers.php	3	2
/faq.html	1	1

```
MariaDB [kalacloud_demo]> SELECT page_url, GROUP_CONCAT(user_id), COUNT(*) AS visits
→ FROM kalalcloud_website_log
→ GROUP BY page_url ORDER BY COUNT(*) DESC;
```

page_url	GROUP_CONCAT(user_id)	visits
/index.html	1,2,3	3
/news.php	1,2	2
/store_offers.php	3,2	2
/faq.html	1	1

```
MariaDB [kalacloud_demo]>
```

MySQL 还提供了 `ANY_VALUE()` 函数，来解决这类问题：

```
卡拉云 kalacloud.com 低代码开发工具  卡拉云 kalacloud.com

MariaDB [kalacloud_demo]> SELECT page_url, ANY_VALUE(user_id), COUNT(*) AS visits
  → FROM kalacloud_website_log
  → GROUP BY page_url ORDER BY COUNT(*) DESC;

+-----+-----+-----+
| page_url          | ANY_VALUE(user_id) | visits |
+-----+-----+-----+
| /index.html      | 1                  | 3      |
| /news.php        | 1                  | 2      |
| /store_offers.php | 3                  | 2      |
| /faq.html        | 1                  | 1      |
+-----+-----+-----+

MariaDB [kalacloud_demo]>
```

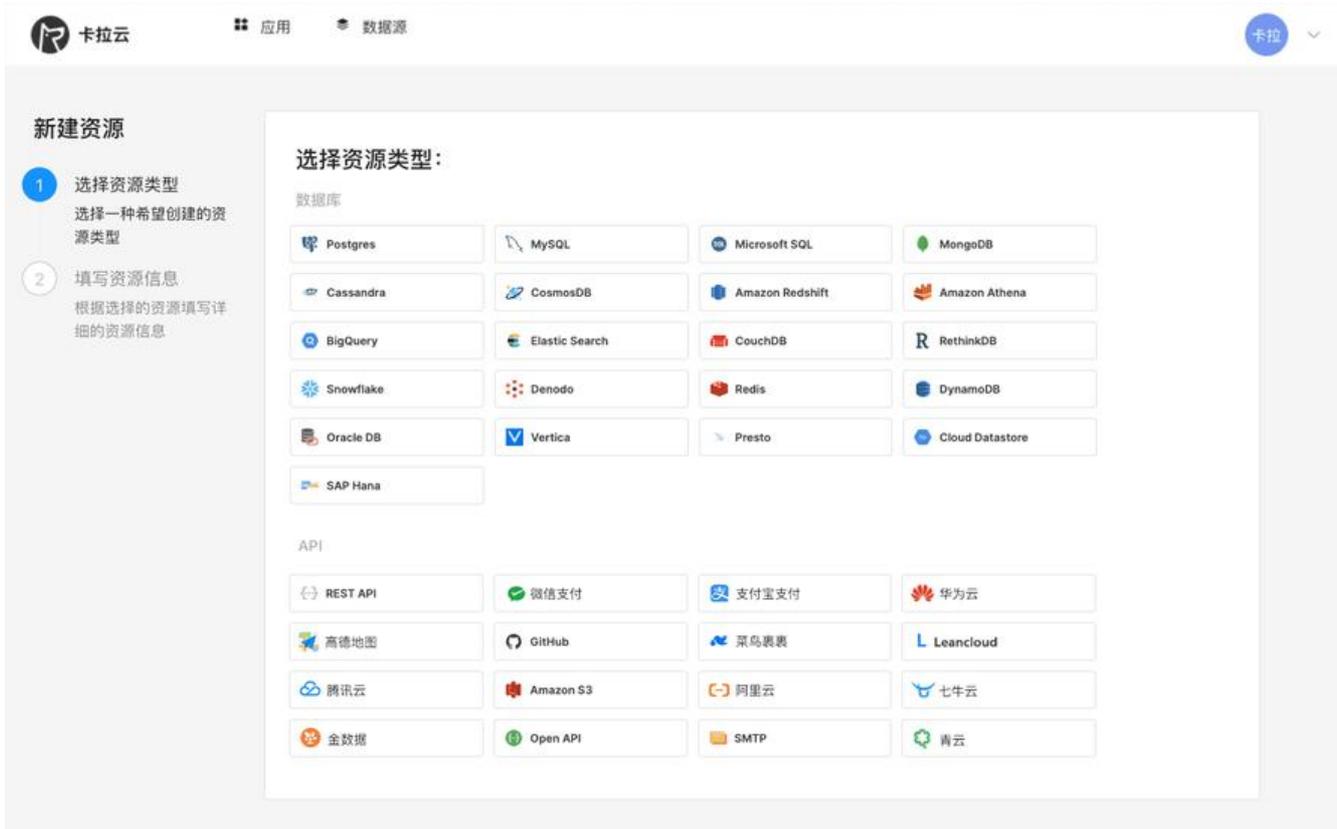
扩展阅读: [MySQL 分组查询实战 如何查询每组第一、最后、单多字段查询、分组筛选、最大小值排序](#)

总结和推荐

在代码严谨、清晰的道路上走多远都不为过，我个人推荐使用第一种解决方案，趁着这一次机会，把代码中潜藏的隐患一网打尽。之后遵循严谨的代码逻辑来写代码。当然，有很多特殊情况，你也可以照方案二、三来临时处理。

接着推荐一下我开发的[卡拉云](#)，只要你会写 MySQL，就能使用卡拉云搭建自己的数据工具，比如，[数据看板](#)，[企业 CRM](#)、[ERP](#)，[权限管理后台](#)，[对账系统](#)等。

卡拉云是新一代低代码开发工具，免安装部署，可一键接入包括 MySQL 在内的常见数据库及 API。根据自己的工作流，定制开发。无需繁琐的前端开发，只需要简单拖拽，即可快速搭建企业内部工具。**月的开发工作量，使用卡拉云后可缩减至数天。**



卡拉云可一键接入常见的数据库及 API

卡拉云可根据公司 workflow 需求，轻松搭建数据看板或其他内部工具，并且可一键分享给组内的小伙伴。



下图为使用卡拉云在 5 分钟内搭建的「优惠券发放核销」后台，仅需要简单拖拽即可快速生成前端组，只要会写 SQL，便可搭建一套趁手的数据库工具。**立即使用卡拉云。**

The screenshot shows the '优惠券发放核销' (Coupon Issuance and Redemption) backend interface. At the top, there's a table listing various coupons with columns for coupon_id, code, name, type, status, amount, and more. Below the table is a configuration form for creating a coupon. The form is divided into three sections: '优惠券信息' (Coupon Information), '优惠券基本规则' (Basic Rules), and '发券人操作信息' (Issuer Operation Information). The '优惠券信息' section includes fields for name, type, amount, and validity. The '优惠券基本规则' section includes fields for validity date, expiration days, and user type. The '发券人操作信息' section includes fields for issuer and reason. A red box highlights the configuration form, and a red arrow points to the '生成优惠券' (Generate Coupon) button. Below the form is a SQL editor with a pre-written SQL statement for inserting a coupon into the 'users' table. The SQL statement is: `INSERT INTO users (coupon_name,type,amount,over,by,date,day,user_type,operator,reason) VALUES ('{{coupon_name.value}}','{{type.value}}','{{amount.value}}','{{over.value}}','{{by.value}}','{{date.value}}','{{day.value}}','{{user_type.value}}','{{operator.value}}','{{reason.value}}');` Below this is a specific example: `= INSERT INTO users (coupon_name,type,amount,over,by,date,day,user_type,operator,reason) VALUES ('老用户回馈券','1','100','','','2021-12-31','','','运营铁柱','合作厂商渠道销售');` A red box highlights the SQL editor, and a red arrow points to the '生成优惠券' (Generate Coupon) button.

有关 MySQL 教程，可继续拓展学习：

- [如何远程连接 MySQL 数据库，阿里云腾讯云外网连接教程](#)
- [MySQL 时间戳用什么类型 - MySQL 时间函数详解](#)
- [MySQL 中如何实现 BLOB 数据类型的存取，BLOB 有哪些应用场景？](#)
- [如何在 MySQL / MariaDB 中跳过多张表导出或指定多张表导出备份](#)
- [如何将 MySQL / MariaDB 的查询结果保存到文件](#)
- [MySQL 分组查询实战 如何查询每组最第一条、最后一条记录，单多字段查询、分组筛选、最大小排序](#)