



链滴

MySQL 「 Every derived table must have its own alias」 1248 错误修复法

作者: [HiJiangChuan](#)

原文链接: <https://ld246.com/article/1638508001730>

来源网站: [链滴](#)

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



MySQL 「 Every derived table must have its own alias 」 错误修复方法

卡拉云 kalacloud.com

本文首发: 《[MySQL 「 Every derived table must have its own alias 」 1248 错误修复法](#)》

在写带有子查询或者在查询时产生临时表的查询时, 可能会出现这个错误:

ERROR 1248 (42000): Every derived table must have its own alias

意思是「每一个派生出来的表都必须给它命名一个自己的别名」

我们看个例子:

假设有一张「顾客购买记录」的表 - `kalacloud_purchases` 记录了顾客在商店购物的信息。我们要写查询, 查看哪些客户在多个商店买过东西:

```
SELECT DISTINCT customer_id, SUM(1)
FROM ( SELECT DISTINCT customer_id, store_id FROM kalacloud_purchases )
GROUP BY customer_id HAVING 1 < SUM(1);
```

```
卡拉云 kalacloud.com 低代码开发工具

MariaDB [kalacloud_demo]> SELECT DISTINCT customer_id, SUM(1)
  → FROM ( SELECT DISTINCT customer_id, store_id FROM kalacloud_purchases )
  → GROUP BY customer_id HAVING 1 < SUM(1);
ERROR 1248 (42000): Every derived table must have its own alias
MariaDB [kalacloud_demo]>
```

运行后, 可以看到出现 1248 错误: `Every derived table must have its own alias`

在这段报错代码中:

FROM (SELECT DISTINCT customer_id, store_id FROM kalacloud_purchases)

这段命令会先查 `kalacloud_purchases` 表，然后生成一张新的临时表，如果这个临时表没有命名，会导致 1248 错误。我们只需要加上「as 临时表别名」即可修复错误

```
卡拉云 kalacloud.com 低代码开发工具

MariaDB [kalacloud_demo]> SELECT DISTINCT customer_id, SUM(1)
  → FROM ( SELECT DISTINCT customer_id, store_id FROM kalacloud_purchases) AS customer
  → GROUP BY customer_id HAVING 1 < SUM(1);
```

加上「AS customer」别名，这样就解决了这个问题。

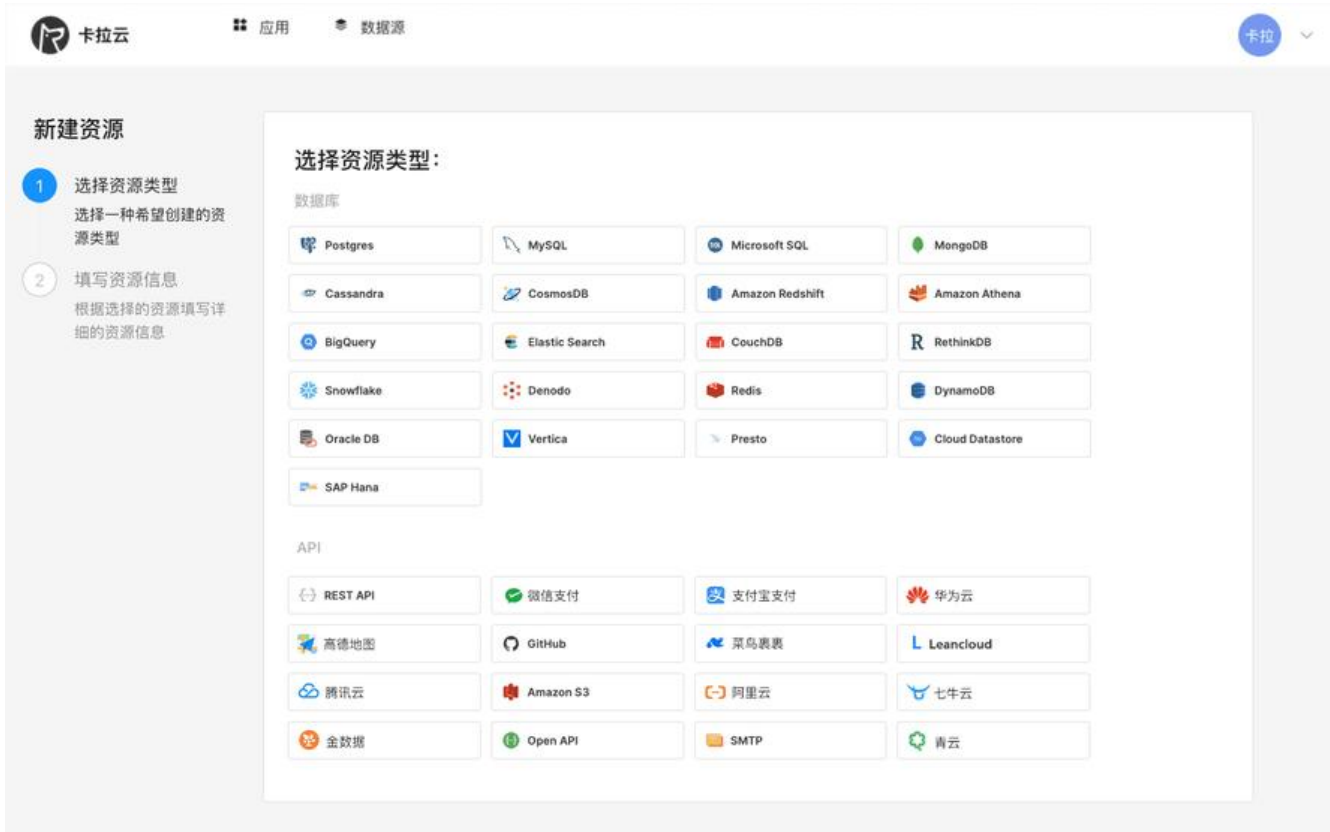
使用「卡拉云」搭建数据看板

我们可以在命令行中直接执行 SQL 对需要的数据进行查询，不过有时候一些运营数据，分析数据需周期性的反复查询，总是复制粘贴代码非常麻烦。有时我们还需要和各部门协同，将查询好的数据分给合作组。这种场景下，你可以试试卡拉云。

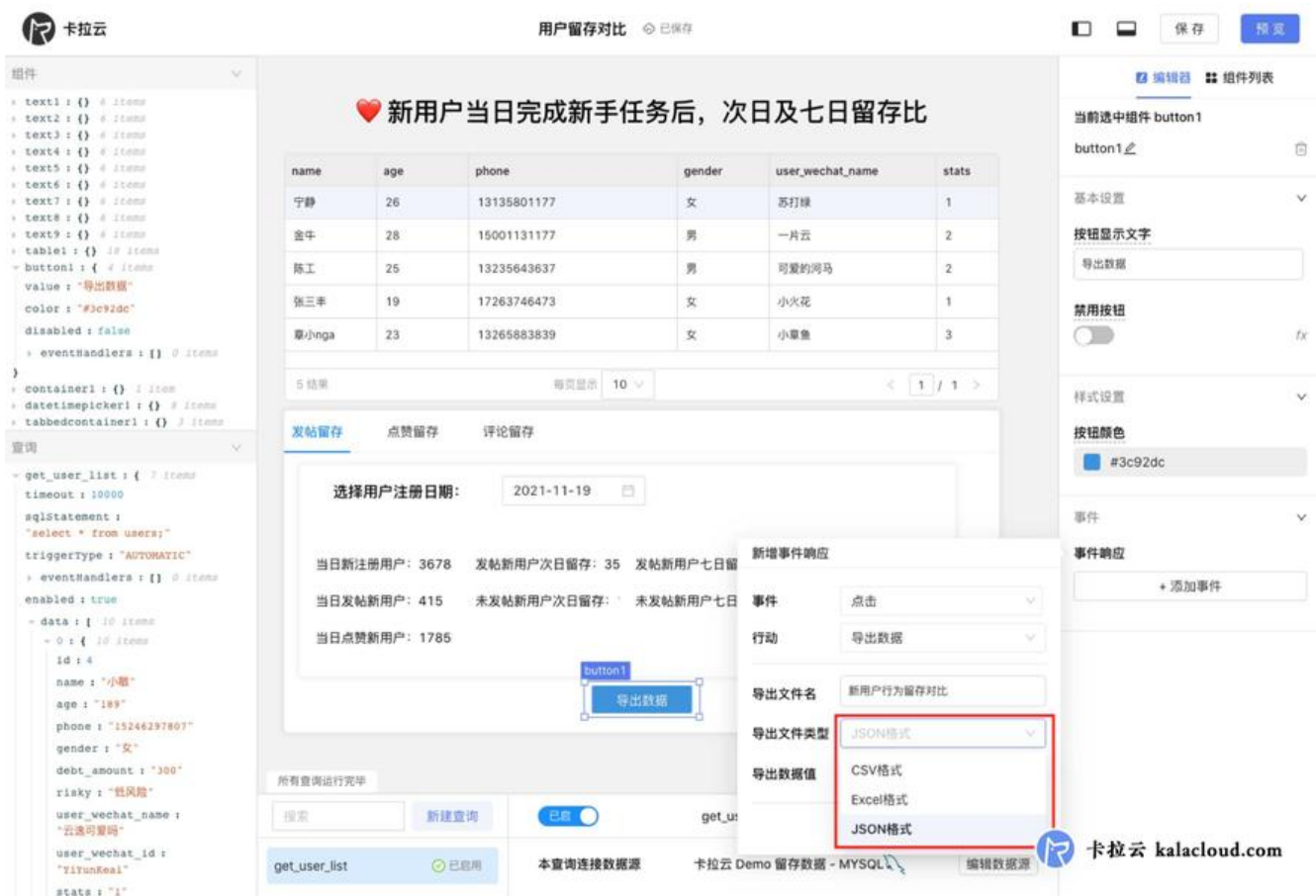
卡拉云可以快速搭建数据看板和数据后台，只要你会写 SQL，完全不用懂前端也可在 5 分钟内搭建套根据自己需求设计的数据看板，还可以做成数据查询系统，一键分享给协作组的同学使用，再也不担心自己沦为肉查询机。



卡拉云不仅可以像命令行一样灵活，还可根据自己的工作流，定制开发。无需繁琐的前端开发，只需简单拖拽，即可快速搭建企业内部工具。数月的开发工作量，使用卡拉云后可缩减至数天。



卡拉云可快速接入的常见数据库及 API，可根据公司 workflows 需求，轻松搭建数据看板，并且可分享给内的小伙伴共享数据，仅需拖拽一键生成前端代码，简单一行代码即可映射数据到指定组件中。



卡拉云可直接添加导出按钮，导出适用于各类分析软件的数据格式，方便快捷。

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。

The screenshot shows the '优惠券发放核销' (Coupon Issuance and Redemption) backend interface. At the top, there is a table listing coupons with columns: coupon_id, code, 优惠券名称 (Coupon Name), 优惠券类型 (Coupon Type), 核销状态 (Redemption Status), 金额 (Amount), 满 (Full), 减 (Discount), 用户类型 (User Type), 发券人 (Issuer), 发券时间 (Issue Time), 过期时间 (Expiration Time), and 发券 (Issue). Below the table is a configuration area for coupon rules, including fields for 优惠券名称 (Coupon Name), 优惠券类型 (Coupon Type), 现金券金额 (Cash Coupon Amount), 满减券, 满 (Full), 减 (Discount), 固定期, 有效期至 (Valid Until), 领券后, 生效天数 (Effective Days), 用户类型 (User Type), 发券人 (Issuer), and 发券原因 (Issue Reason). A red box highlights this configuration area, and a red arrow points to the '生成「优惠券」' (Generate 'Coupons') button. Below the configuration area is a section for SQL queries and triggers, with a red box highlighting the SQL query:

```
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}}');
```

 and a trigger definition:

```
= 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');
```

立即开通卡拉云，搭建属于自己的数据后台工具吧。

有关 MySQL 教程，可继续拓展学习：

- [如何远程连接 MySQL 数据库，阿里云允许外网连接教程](#)
- [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](#)
- [MySQL 触发器的创建、使用、查看、删除教程-六种情况实战](#)
- [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](#)