Atitit unique索引的使用和问题解决

目录

[1.1. 多列所以union unique 1](#_Toc11147)

[2. 在MySQL的一张innodb引擎的表上添加一个唯一索引(uk\_id)，但是表中该字段有大量重复数据 1](#_Toc8250)

[2.1. alter ignore table 添加 bef 5.7.4 1](#_Toc13051)

[2.2. 从MySQL 5.7.4开始，新表+ INSERT IGNORE 2](#_Toc23997)

[3. 添加流程总结 3](#_Toc21530)

[4. Other 4](#_Toc28167)

[4.1. 可以使用 INSERT INTO football\_score\_t SELECT \* from tmp\_data; 判断问题如果插入不进去数据 4](#_Toc18016)

## 多列所以union unique

# 在MySQL的一张innodb引擎的表上添加一个唯一索引(uk\_id)，但是表中该字段有大量重复数据

## alter ignore table 添加 bef 5.7.4

DataFlow范式 2015-02-24 17:14:08 4142 收藏

展开

场景：

在MySQL的一张innodb引擎的表上添加一个唯一索引(uk\_id)，但是表中该字段有大量重复数据。

方法：

 alterignore add unique index语法

行为类似于insert ignore，即遇到冲突的unique数据则直接抛弃而不报错。对于加唯一索引的情况来说就是建一张空表，然后加上唯一索引，将老数据用insert ignore语法插入到新表中，遇到冲突则抛弃数据。

————————————————

版权声明：本该查询将失败，并显示错误1062（重复键）。

但是如果你用 IGNORE

-- (only works before MySQL 5.7.4)ALTER IGNORE TABLE mytable ADD UNIQUE INDEX myindex (A, B, C, D);

重复项将被删除。但是文档没有指定要保留的行：

IGNORE是标准SQL的MySQL扩展。它控制ALTER TABLE新表中唯一键上是否有重复项或启用严格模式时是否出现警告的工作方式。如果IGNORE未指定，则复制副本将中止并在发生重复键错误时回滚。如果IGNORE指定，则仅一行使用唯一键重复的行。其他冲突的行将被删除。不正确的值将被截断为最接近的匹配可接受值。

## 从MySQL 5.7.4开始，**新表+ INSERT IGNORE**

ALTER TABLE的IGNORE子句被删除，

新表+ INSERT IGNORE

尽管此方法看起来非常相似，但在内部它的语义却大不相同：

CREATE TABLE users\_new LIKE users;

ALTER TABLE users ADD UNIQUE INDEX (emailaddress);

INSERT IGNORE INTO users\_new SELECT \* FROM users;

DROP TABLE users;

RENAME TABLE users\_new TO users;

通过首先创建一个表，MySQL服务器将不必管理外键关系中的行。使用基于行的复制还将行重新发送到从属服务器，因此上面提到的问题（3）不会起作用。

使用它会产生错误。

（[ALTER TABLE语法](http://dev.mysql.com/doc/refman/5.7/en/alter-table.html)）

如果您的版本是5.7.4或更高版本-您可以：

* 将数据复制到临时表中（从技术上讲，它不需要是临时表）。
* 截断原始表。
* 创建唯一索引。
* 并将数据复制回INSERT IGNORE（仍然可用）。
* 并将数据复制回INSERT IGNORE（仍然可用）。

CREATE TABLE tmp\_data SELECT \* FROM mytable;

TRUNCATE TABLE mytable;

ALTER TABLE mytable ADD UNIQUE INDEX myindex (A, B, C, D);

INSERT IGNORE INTO mytable SELECT \* from tmp\_data;

DROP TABLE tmp\_data;

SET FOREIGN\_KEY\_CHECKS = 0;

INSERT IGNORE INTO football\_score\_t SELECT \* from tmp\_data;

INSERT INTO football\_score\_t SELECT \* from tmp\_data where team\_id in (select id from football\_team\_t);

# 添加流程总结

select \* from football\_tlive\_t

CREATE TABLE tmp\_tlive SELECT \* FROM football\_tlive\_t;

select \* from tmp\_tlive

TRUNCATE TABLE football\_tlive\_t;

ALTER TABLE `football\_tlive\_t`

ADD UNIQUE INDEX `直播防重索引` (`match\_id`, `position`, `type`, `time`) ;

SET FOREIGN\_KEY\_CHECKS = 0;

INSERT IGNORE INTO football\_tlive\_t SELECT \* from tmp\_tlive

select \* from football\_tlive\_t

# Other

## 可以使用 INSERT INTO football\_score\_t SELECT \* from tmp\_data; 判断问题如果插入不进去数据