

# DDL 设计

---

有哪些DDL语句？

虚拟表的创建

增、删、改虚拟表的字段

## 创建表

没有什么影响。因为是全新的表，没有任何数据和它相关。

## 增加字段

新增一个字段到表里面，这个字段是新的，还没有任何数据使用这个字段。

只需要考虑要不要为所有记录更新这个字段并设置值。一个新字段，就是能不能为空。

nullable是否可为空

- 如果可为空，什么都不需要做；
- 如果不可以为空，需要在现有虚拟表上遍历所有记录增加值，这个值只能是缺省值。
  - 引入default来设置缺省值。
    - 如果设置了unique=true，default的值一定要满足唯一性。
    - 如果使用引用，那么此字段是外键，还得考虑此缺省值是否存在被引用表的主键中

## 修改字段

- 修改name，没有问题，只是一个描述字符串而已。
- 修改meta
  - 改type，要遍历所有虚拟表的该字段数据，拿出来转类型成功后，写回去。操作成本太高了，因此，禁用
  - nullable
    - 从false到true，什么都不做
    - 从true到false，就需要对这个虚拟表的该字段遍历。使用schema\_id和field\_id在value表中查找value字段是否为null，或者缺失（因为新增字段时，可以为空，就什么都不做）。如果为null或者缺失，校验失败  
SELECT count(id) from value WHERE field\_id=2 and entity\_id=3;  
count为0，校验失败
  - unique
    - 从true到false，什么都不做
    - 从false到true，如何判断？  
使用field\_id在value表中查找value字段中count的变化  
SELECT count(id) from value WHERE field\_id=2;  
SELECT DISTINCT count(value) from value WHERE field\_id=2;  
如果两次count结果一样，就是无重复的。  
注意这种查询太慢了
- multi

- 从false到true，什么都不做
- 从true到false，需要计算每一个entity的entity\_id、field\_id的count  
SELECT count(id) from value WHERE field\_id=2 and entity\_id=1;  
count > 1 说明是多个值，检验失败
- reference
  - 清除引用信息，就是去掉约束，不用校验。而且value字段上的值原来谁就是合法的，所以继续保留
  - 增加引用，就要检验外键的值是否出现在了被引用表的主键上，只要有一个不在，检验失败。如果检验成功，在reference字段上填上外键信息。这些主键、外键值都存储在value表中，所以使用自关联。看下面的 参考1。如果通过自关联左联查出count大于0，就检验失败。
- option修改，只是数据处理的限制条件，我们这里的约定修改了，将影响以后的数据限制，已经入库的不检验不更新

### 参考1

-- 使用自连接

```
SELECT source.*, target.*
FROM `value` AS source LEFT JOIN `value` AS target
ON source.`value` = target.`value` AND target.field_id = 3
WHERE target.`value` is NULL AND source.field_id = 2
```

```
SELECT count(source.id) FROM
`value` AS source LEFT JOIN `value`
AS target ON source.`value` = target.`value`
AND target.field_id = 3
WHERE target.`value` is NULL AND source.field_id = 2
```

-- 使用左联，从外键所在的source表往主键所在的target表看，不过target表的数据在join时先要过滤一下 target.field\_id = 3，也就是field表中id=3的字段被引用了。这样，左联后，如果source表的ip比target表多，就违反了外键约束，代表source用到的ip不存在target表中。为了过滤出这个多出来的ip，使用了条件 target.`value` is NULL AND source.field\_id = 2。