**6.17SIG组会问题回复：**

1. ustore行存表是否不支持reindex concurrently，不支持需增加相应的报错提示，如在reindex index和reindex table中报ERROR，在reindex database中报WARNING

通过测试分析，ustore行存表目前不支持create index concurrently。需在reindex concurrently语句中增加相应操作，避免对该模式下的表执行在线重建功能。

在checkTableForReindexConcurrently函数中对ustore表报错ERROR



在ReindexDatabase中增加相应函数，在concurrently情况下跳过ustore表



自测结果显示：



2.列存表在reindex database中报ERROR会影响效率，列存表产生的行存表在特定的namespace当中，想办法识别并修改reindex database的报错机制，同时也需要在reindex index、reindex table中对产生的行存表进行相应的报错处理

通过测试分析，创建列存表后，会在模式cstore创建两个relkind=r的表，分别名为pg\_delta\_xxx和pg\_cudesc\_xxx的表。因此，在reindex concurrently语句中增加操作，避免对该模式下的表执行在线重建功能。

在checkTableForReindexConcurrently函数中对列存表和其相关表报错ERROR



在ReindexDatabase中增加相应函数，在concurrently情况下跳过列存表和其相关表



自测结果显示：



3.由于reindex concurrently不支持在事务中执行，需要采用白盒测试的方式，即在代码中增加断点或者sleep，在不同阶段并发执行DML语句，测试是否存在问题，特别是执行速度特别快的阶段

4.目前reindex concurrently只是对普通的int、text、timestamp类型进行测试，对索引的类型也是默认类型，需测试不同索引类型、不同数据类型下执行在线重建是否存在问题

将3和4的问题进行合并回复。采用白盒测试，在reindex concurrently的6个阶段事务中增加睡眠点pg\_usleep，对不同数据类型、不同索引类型执行reindex concurrently语句，在每个睡眠点对数据执行增删改查操作，根据执行时间和返回值判断是否存在影响。为了便于测试，每次测试的测试表中只包含10000条数据。由于reindex database concurrently和reindex table concurrently对索引的重建方式与reindex index concurrently相同，因此，测试以reindex index concurrently语句为主。

共进行了93种测试，包含int、double、number、char、varcher、text、blob、timestamp、point、int4range、tsvector、bit数据类型类型，涉及到btree、hash、gin、gist等行存表索引，同时对不同分区表（range、hash、interval、list）也进行相应的测试。相应的测试记录见附件。