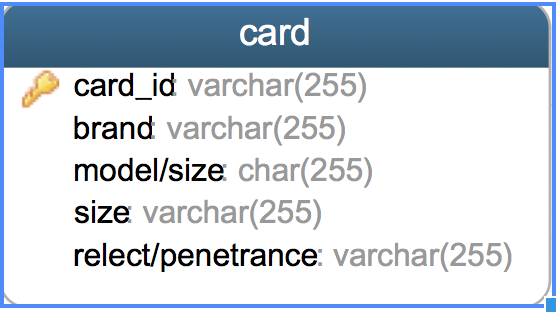
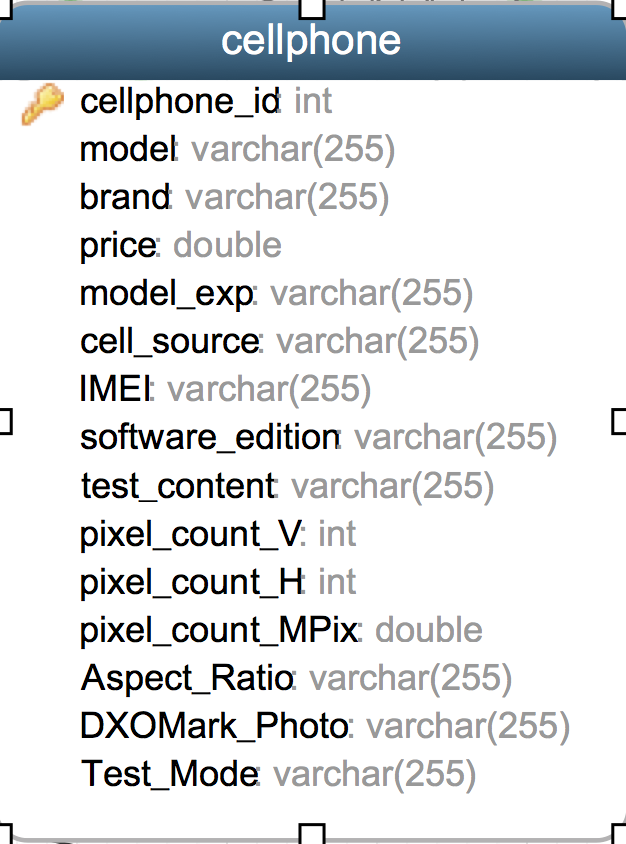
数据库工作总结

一、构建基础数据库

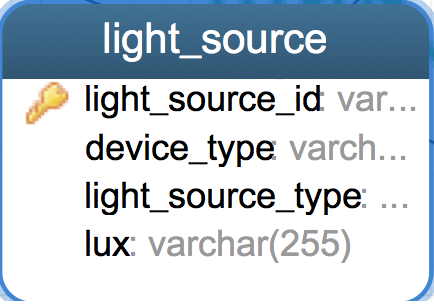
card表，记录所有图卡的基本信息



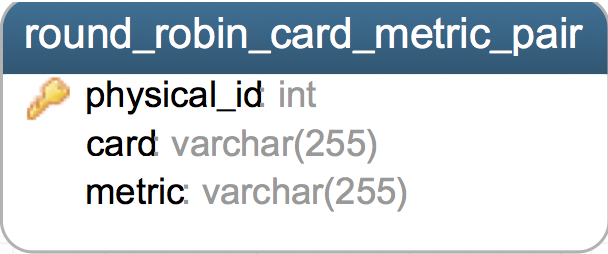
cellphone表，记录所有样机的基本信息



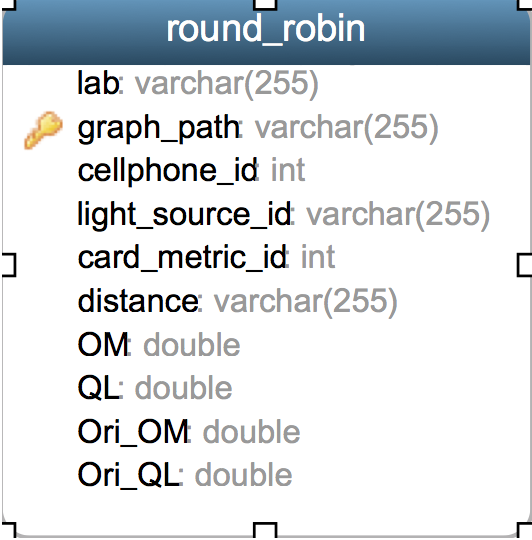
light\_source表，记录所有光源的基本信息



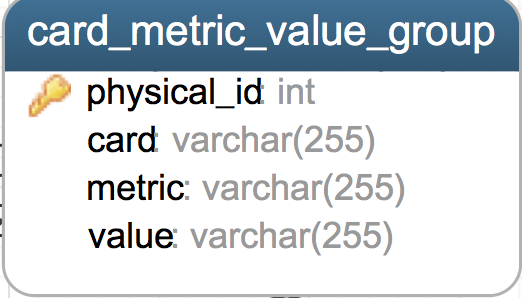
round\_robin\_card\_metric\_pair表，记录round\_robin项目中的所有图卡与metric的测试需求映射，拥有一个逻辑主键



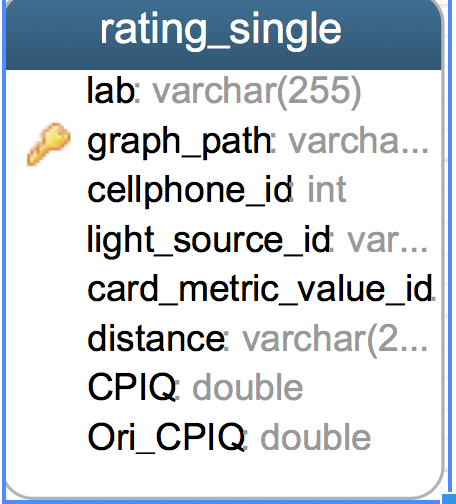
round\_robin表，记录round\_robin项目每一个测试测试结果，每一条测试结果包含card、cellphone、light source、lab、distance、metric、QM、QL、原始数据等信息。card、cellphone、light source、round\_robin\_card\_metric\_pair的主键作为其外键，方便关联查找



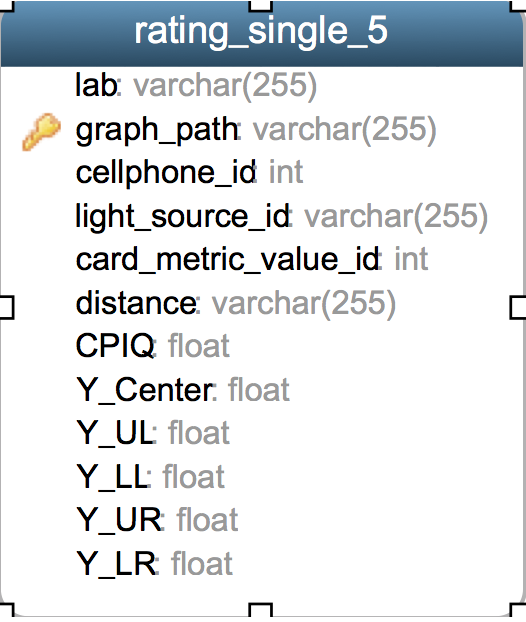
card\_metric\_value\_group表，记录rating项目中所有图卡的metric、value映射，拥有一个逻辑主键



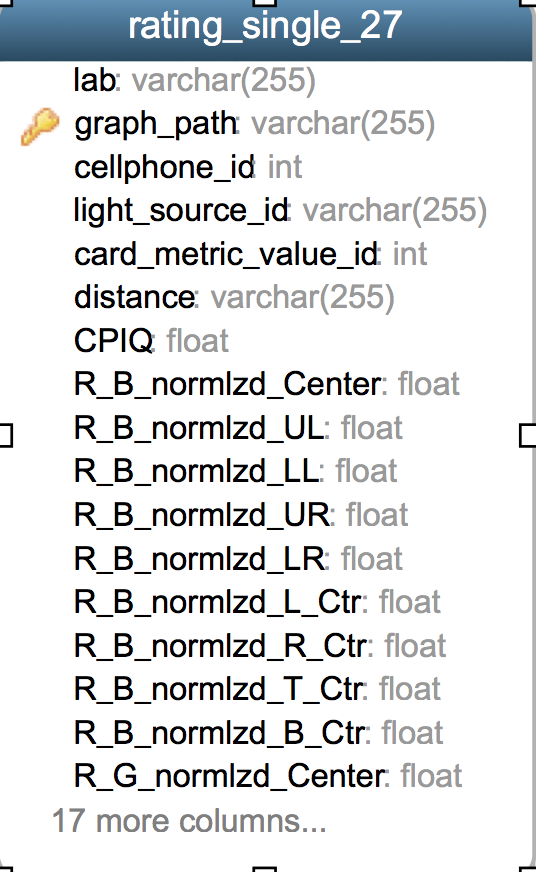
rating\_single表，记录rating项目里图卡为DXO SFR、DXO Dot、DXO texture的原始数据以及计算结果(CPIQ)



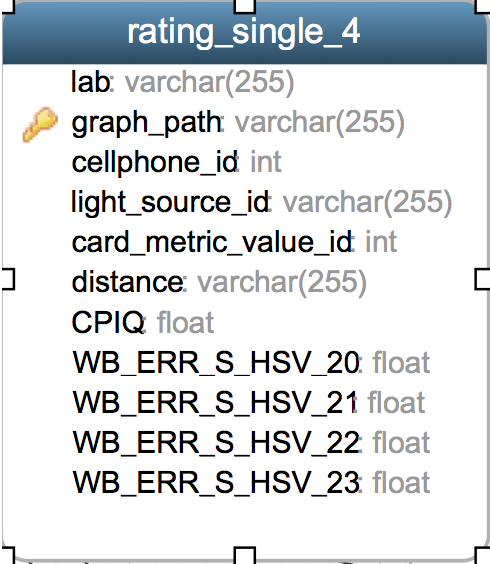
rating\_single\_5表，记录rating项目里图卡为TE255、grey，value为KA、KB、KC、KD均值的原始数据以及计算结果(CPIQ)



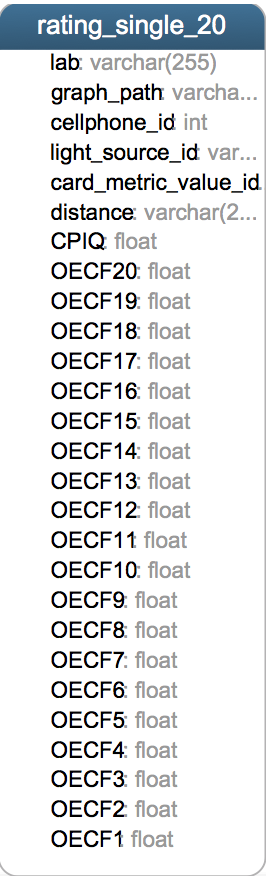
rating\_single\_27表，记录rating项目里图卡为TE255、grey，value为九点R/B R/G B/G MIN、九点R/B R/G B/G 平均值、九点R/B R/G B/G MAX、九点R/B R/G B/G 与1差值的绝对值的最大值的原始数据以及计算结果(CPIQ)

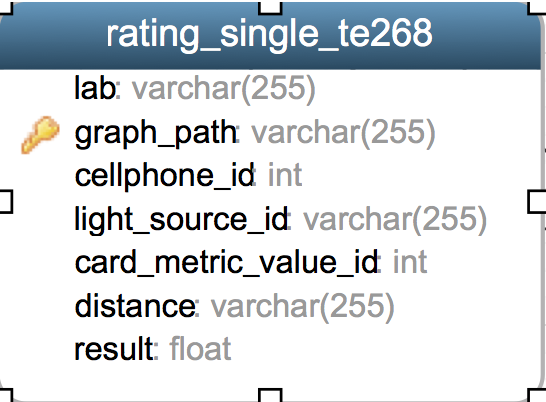


rating\_single\_4表，记录rating项目里图卡为Colorchecker，value为各色块对应的饱和度S值的最大值、各色块对应的饱和度S值的平均值的原始数据以及计算结果(CPIQ)



rating\_single\_te270表，记录rating项目里图卡为TE270，metric为Visual noise的原始数据以及计算结果(CPIQ)

   
rating\_single\_te268表，记录rating项目里图卡为TE268的所有计算结果(result)



二、代码逻辑解析

1 每个项目的入口都是scanFromRoot函数，该函数接收一个文件根目录作为参数。函数会遍历该根目录下的所有文件与文件夹，如果子项目是一个文件夹，则进入步骤2

2 scanfile函数将传入的文件夹名作为参数，遍历该目录下的所有文件。每一个子文件的文件名都会以”\_” 分割并且被保存起来，进入步骤3

3 所有子文件的文件名信息传送给controller函数，controller函数会建立与数据库的连接，并且将遍历所有子文件的文件名信息。根据文件名信息中的图卡名称信息，进入各子文件的处理方式，处理方式见步骤4。处理完毕之后，关闭与数据库的连接

4 每一种图卡的处理方式大致类似。首先会根据文件名信息去查询手机id，光源id，以及测试项目的逻辑主键。再根据需求执行txt/csv/xls的数据读取操作，读取操作的原理是先进入results文件夹下相应的结果文件，遍历结果文件的每一行，找到有关键字的行，记录目标数据。使用目标数据进行一定的运算从而得到结果。最后将所有信息写入一条sql插入语句之中，执行插入语句，完成数据的录入。

三、操作说明

服务器使用pyCharm打开两个.py文件，分别运行即可。如若存放测试图片以及处理结果的文件根目录改变，需要修改\_\_name\_\_函数里的rootpath变量。