<https://blog.csdn.net/adayan_2015/article/details/81080586>

## MySQL

select subject\_id,student\_id,score

from score sc1

where (select count(1) from score sc2 where sc2.subject\_id=sc1.subject\_id and sc2.score >= sc1.score) <=3

order by subject\_id ,score desc;

<https://blog.csdn.net/qq_36770189/article/details/99674117?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase>

## 利用hiveSQL实现：

select \* from

(select name, subject,

rank() over(partition by subject order by score desc) t

from score) tt

where tt.t<4;

利用窗口函数对按各学科分区，并降序排列，利用rank（）函数规避有相同成绩名次的不一的情况学生。最后where语句取出前三名。

rank（）排序相同时会重复，总数不变

dense\_rank()排序相同时会重复，总数会减少

row\_rank()会顺序计算

<https://www.cnblogs.com/anyhow/p/3641681.html>

## ORACLE分科目统计每科前三名的学生的语句

有个成绩表 score（student\_no,Subject\_no,Score）分别为学号，课程号，成绩。我想用语句查询出每科的前三名学生的学号，请各位高手教教小弟

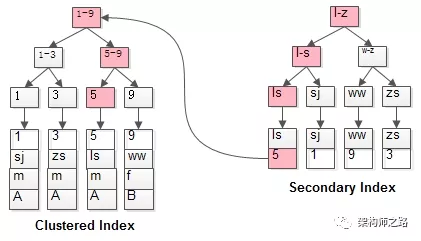
select \* from (

select t.\*,row\_number() over(partition by t.subject\_no order by t.score desc) pm from score t) where pm<4;

<https://www.cnblogs.com/myseries/p/11265849.html>

## 回表

**是如何执行的呢？**



如**粉红色**路径，需要扫码两遍索引树：

（1）先通过普通索引定位到主键值id=5；

（2）在通过聚集索引定位到行记录；

这就是所谓的**回表查询**，先定位主键值，再定位行记录，它的性能较扫一遍索引树更低。

<https://www.cnblogs.com/mr-wuxiansheng/p/11188733.html>

## Mysql中HAVING的相关使用方法

having字句可以让我们筛选分组之后的各种数据，where字句在聚合前先筛选记录，也就是说作用在group by和having字句前。

而having子句在聚合后对组记录进行筛选。我的理解就是真实表中没有此数据，这些数据是通过一些函数产生的。

<https://www.cnblogs.com/tufujie/p/9413852.html>

## MySQL Explain详解

概要描述：

id:选择标识符

select\_type:表示查询的类型。

table:输出结果集的表

partitions:匹配的分区

type:表示表的连接类型

possible\_keys:表示查询时，可能使用的索引

key:表示实际使用的索引

key\_len:索引字段的长度

ref:列与索引的比较

rows:扫描出的行数(估算的行数)

filtered:按表条件过滤的行百分比

Extra:执行情况的描述和说明

总结：

• EXPLAIN不会告诉你关于触发器、存储过程的信息或用户自定义函数对查询的影响情况

• EXPLAIN不考虑各种Cache

• EXPLAIN不能显示MySQL在执行查询时所作的优化工作

• 部分统计信息是估算的，并非精确值

• EXPALIN只能解释SELECT操作，其他操作要重写为SELECT后查看执行计划。