**How will you be doing this?**

Enhance and maintain the continuous integration system. Make sure it is fast, stable and super easy to use.

Manage the configuration of Jenkins job.

Build dev tool-chain, to make engineer [keep](/jump/super-jump/word?word=keep" \t "_blank) away the bored repeat work. For example, IDE plugin, code checker, git hook, hipchat integration, wallboard, etc.

Build new feature of the monitoring system.

**What will you bring to the team?**

Bachelor or Master of Computer science or related major.

Be familiar with at least one programming language. Python or Java is preferred.

Have knowledge about how the web application works.

You MUST own works out of your course homework. It can be anything like game, scraper, tool, data model, or from other internship experience.

Know how to use linux for daily work.

Ever used version control system. Know Git is a plus.

It's a plus if you know Jenkins or Docker.

python面试

1、python中常用的数据结构有哪些？请简要介绍一下。

2、简要描述python中单引号、双引号、三引号的区别。

3、如何在一个function里设置一个全局的变量。

4、python里面如何拷贝一个对象？（赋值、浅拷贝、深拷贝的区别）

5、如果custname字符串的编码格式为uft-8,如何将custname的内容转化为gb18030的字符串？

6、请写出一段python代码实现删除list中的重复元素。

7、这两个参数是什么意思？args和 kwargs。

mysql 窗口函数：

1. 序号函数 ROW\_NUMBER()、RANK()、DENSE\_RANK()

显示分区中的当前行号 应用：查询每个学生的分数最高的前三门课程(比较的是score)

**ROW\_NUMBER() OVER (PARTITION BY stu\_id Order by score)**

PARTITION BY子句：窗口按照哪个字段进行分组。

Order by 按哪个字段排序

SELECT \* FROM(

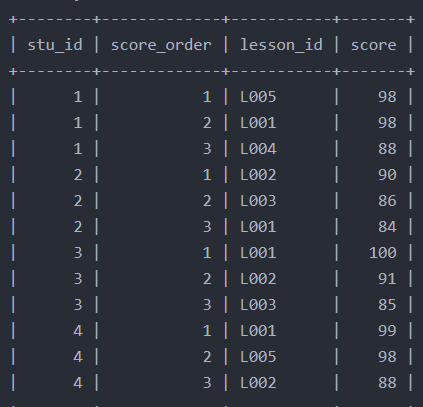
SELECT stu\_id,

ROW\_NUMBER() OVER (PARTITION BY stu\_id ORDER BY score DESC) AS score\_order,

lesson\_id, score

FROM t\_score) t

where score\_order <= 3;



对于stu\_id=1的同学，有两门课程的成绩均为98，序号随机排了1和2。但很多情况下二者应该是并列第一，则他的成绩为88的这门课的序号可能是第2名，也可能为第3名。

这时候，ROW\_NUMBER()就不能满足需求，需要RANK()和DENSE\_RANK()出场，它们和ROW\_NUMBER()非常类似，只是在出现重复值时处理逻辑有所不同。

ROW\_NUMBER()：顺序排序——1、2、3  
RANK()：并列排序，跳过重复序号——1、1、3  
DENSE\_RANK()：并列排序，不跳过重复序号——1、1、2

SELECT \*

FROM(

SELECT

ROW\_NUMBER() OVER (PARTITION BY stu\_id ORDER BY score DESC) AS score\_order1,

RANK() OVER (PARTITION BY stu\_id ORDER BY score DESC) AS score\_order2,

DENSE\_RANK() OVER (PARTITION BY stu\_id ORDER BY score DESC) AS score\_order3,

stu\_id, lesson\_id, score

FROM t\_score) t

WHERE stu\_id =1 AND score\_order1<=3 AND score\_order2<=3 AND score\_order3<=3;

分布函数 PERCENT\_RANK()、CUME\_DIST()

PERCENT\_RANK() 每行按照公式(rank - 1) / (rows - 1) rank为RANK()产生的序号，rows为当前窗口的记录总行数

给窗口指定别名： WINDOW w AS (PARTITION BY stu\_id ORDER BY score)

SELECT

RANK() OVER w AS rk,

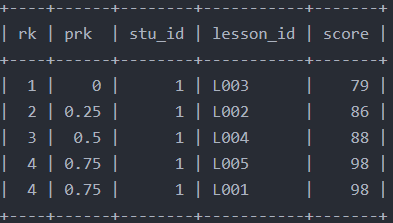
PERCENT\_RANK() OVER w AS prk,

stu\_id, lesson\_id,score

FROM t\_score

where stu\_id = 1

WINDOW w AS (PARTITION BY stu\_id ORDER BY score);



CUME\_DIST()

用途：分组内<=当前rank值的行数/分组内总行数

查询小于等于当前成绩的比例

cd1：无分区，总行数8 cd2：按照lesson\_id分成两组，行数各为4

SELECT stu\_id, lesson\_id, score,

CUME\_DIST() OVER (ORDER BY score) AS cd1,

CUME\_DIST() OVER (PARTITION lesson\_id ORDER BY score) AS cd2,

FROM t\_score

WHERE lesson\_id IN (‘L001’,‘L002’);

