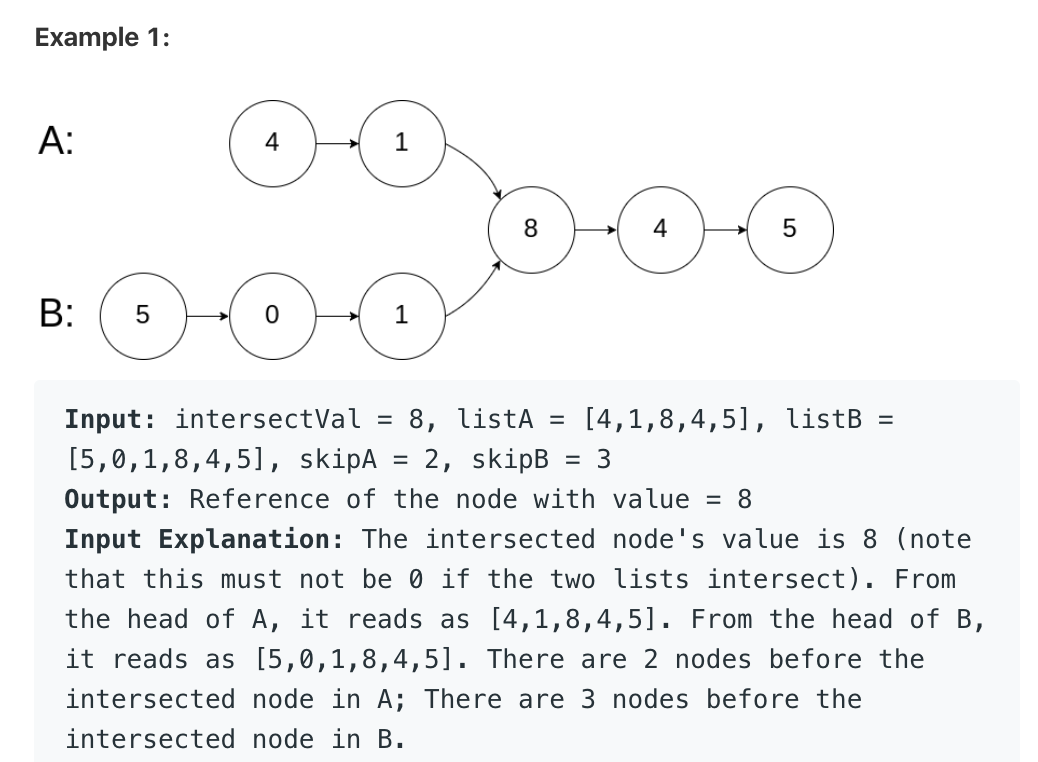
#### 前言：

**今天的题目都有点不大会，特别是学习了数据库的知识，具体内容都在每道题中分别有讲解**

题目一.

#### LeetCode [160. Intersection of Two Linked Lists](https://leetcode-cn.com/problems/intersection-of-two-linked-lists/)



这题hashSet的方法就不说了，主要介绍一下双指针，因为两个指针的长度可能不一样，而我们想让指针同时停在相交的那个点，这该如何解？？

两个指针的移动速度一样，所以我们要确保两个指针移动的距离一样，就能让两个指针同时到达终点，而相交部分是一样长的，也就是能同时到达相交点。移动相同的距离就是两每个指针都走完这两条链表

1.指针 pA 指向 A 链表，指针 pB 指向 B 链表，依次往后遍历

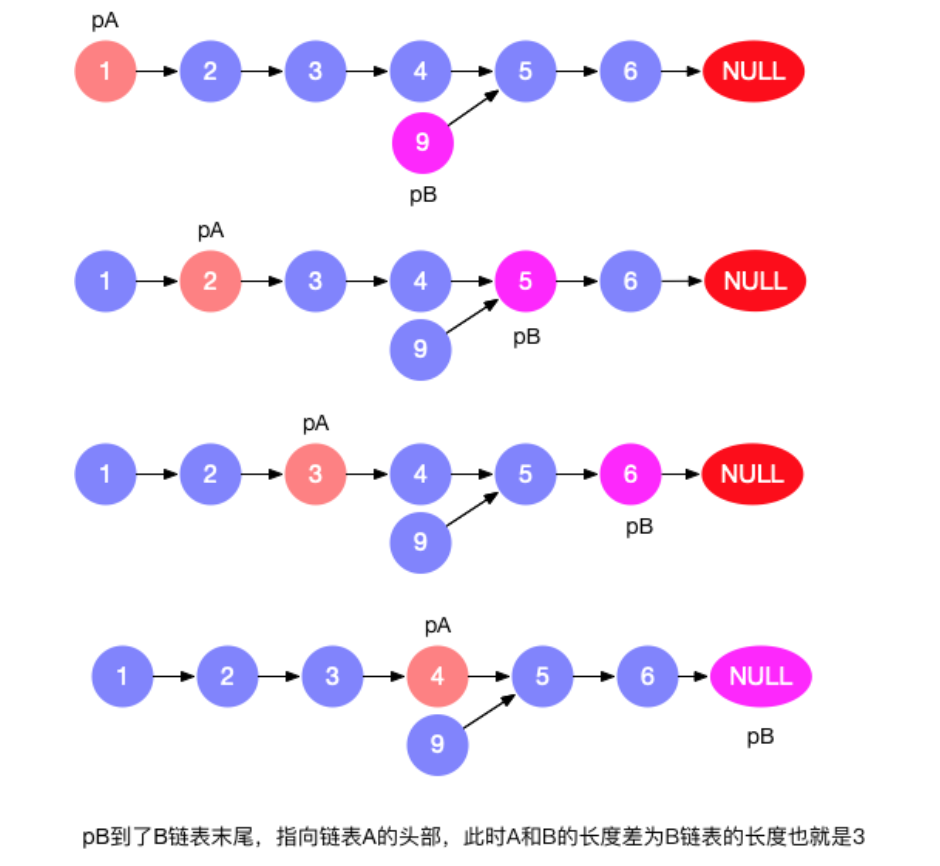
2.如果 pA 到了末尾，则 pA = headB 继续遍历

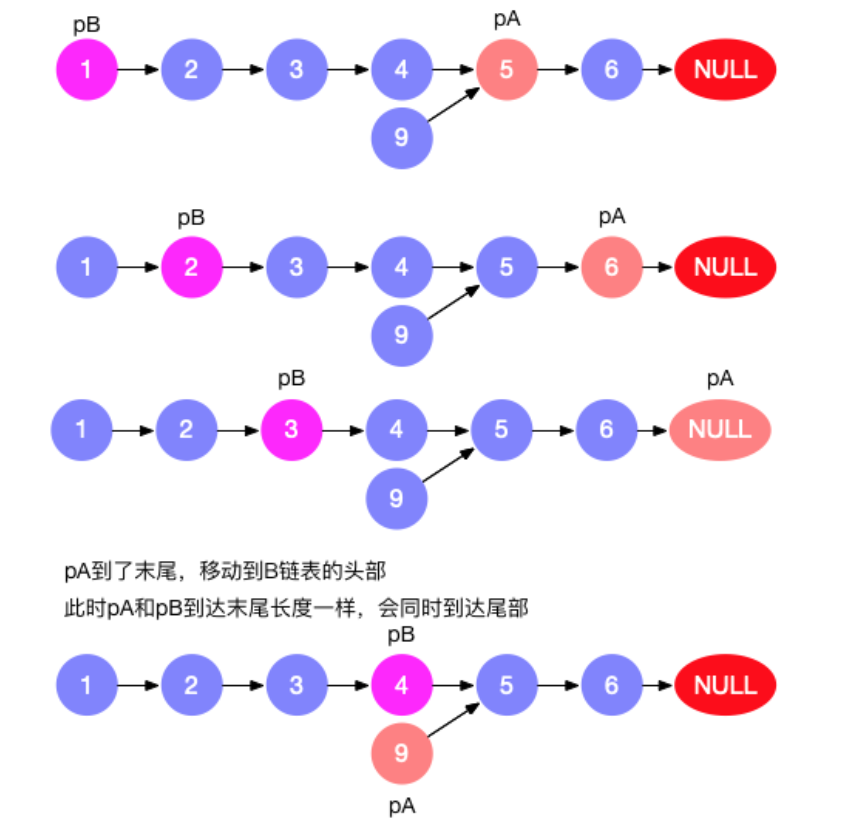
3.如果 pB 到了末尾，则 pB = headA 继续遍历

4.比较长的链表指针指向较短链表head时，长度差就消除了

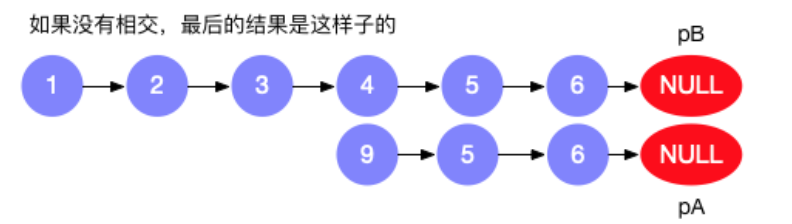
5.如此，只需要将最短链表遍历两次即可找到位置

设A链表长度n,B链表长度m，公共长度是k，当两个链表都走了n+m-k时就相遇了，





如果两个链表没有相交，经过n+m次移动后也会同时到达空指针



源代码如下：

public ListNode getIntersectionNode(ListNode headA, ListNode headB) {

        //题解里学到的双指针方法

        if(headA==null||headB==null)return null;

        ListNode p1=headA,p2=headB;

        while(p1!=p2)

        {

            //System.out.println(p1.val+" "+p2.val);

            p1= p1==null?headB:p1.next;

            p2= p2==null?headA:p2.next;

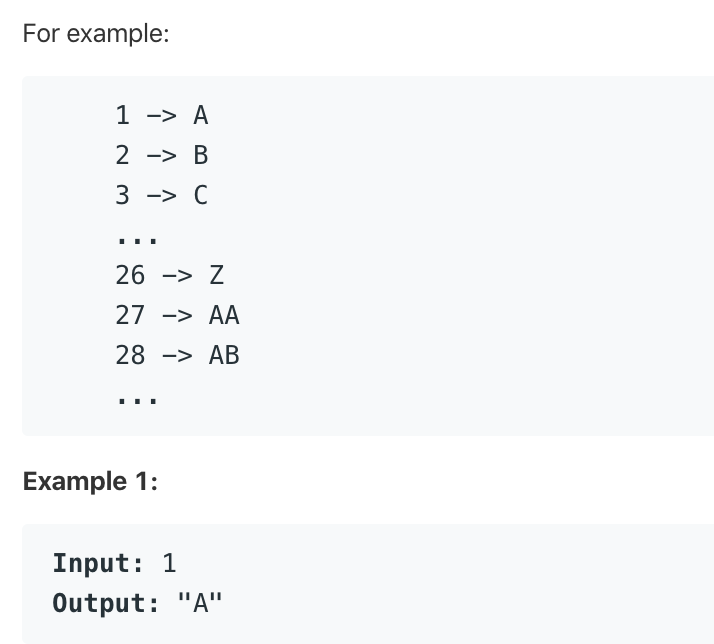
        }

        return p1;

    }

题目二.

#### LeetCode[168. Excel Sheet Column Title](https://leetcode-cn.com/problems/excel-sheet-column-title/)



这道题其实是进制问题，26进制，做法类似于16进制。

28=26^1 \*1+26^0 \*2 所有对应A和B，所以每次求后面位数对n%26就能得到，然后在通过n/26 将得到28/26=26^0 \*1 也就是指数位减一，将最后一位去除。

N=.....26^2 \*x3+26^1\*x2+ 26^0 \*x1

但是，这里有个问题，一般的进制问题，如16进制是从0--15，而本题好似1--26，所以如果x1是26，那在n/26以后N=.....26^1 \*x3+26^0\*x2+1 并不能把最后一位去除，所以单独判断一下，如果n%26==0则该元素位‘Z',并且N-1，好让（x1-1）/26==0，也就是去除最后一位。

题目三.

#### LeetCode [172. Factorial Trailing Zeroes](https://leetcode-cn.com/problems/factorial-trailing-zeroes/)

求阶乘后面0的个数。一开始傻逼了虽然直到不能全不求完在统计，直到会超过int类型，然后就每次乘完都统计一下，可是这样也很容易就超过int类型，差不多到20多一些就超了，但是题目中输入的是整型，会非常大。

思考一下：0的个数就是能被10整除的，也就是2和5对应的组数，而很明显2出现的次数肯定比5多，所以只要统计n的阶乘中5出现的次数！！很简单的想法就是每5次出现一个5，例如n=5，则出现一次5；n=19，则出现n/5=3次5。但是当n=5\*5时，要多加一次，也就是每25次要多加一个5，例如n=25,则出现n/5+n/25=6次5.同理每5\*5\*5次也要多加一次5.

cnt=N/5+N/5^2+N/5^3+....+N/5^log5(N)

但是这样分母因为是指数增长，大概在5^15次会超过int类型，所以每次都将N=N/5;迭代

源代码如下：

    public int trailingZeroes(int n) {

        int cnt=0;

        while(n!=0)

        {

            cnt+=n/5;

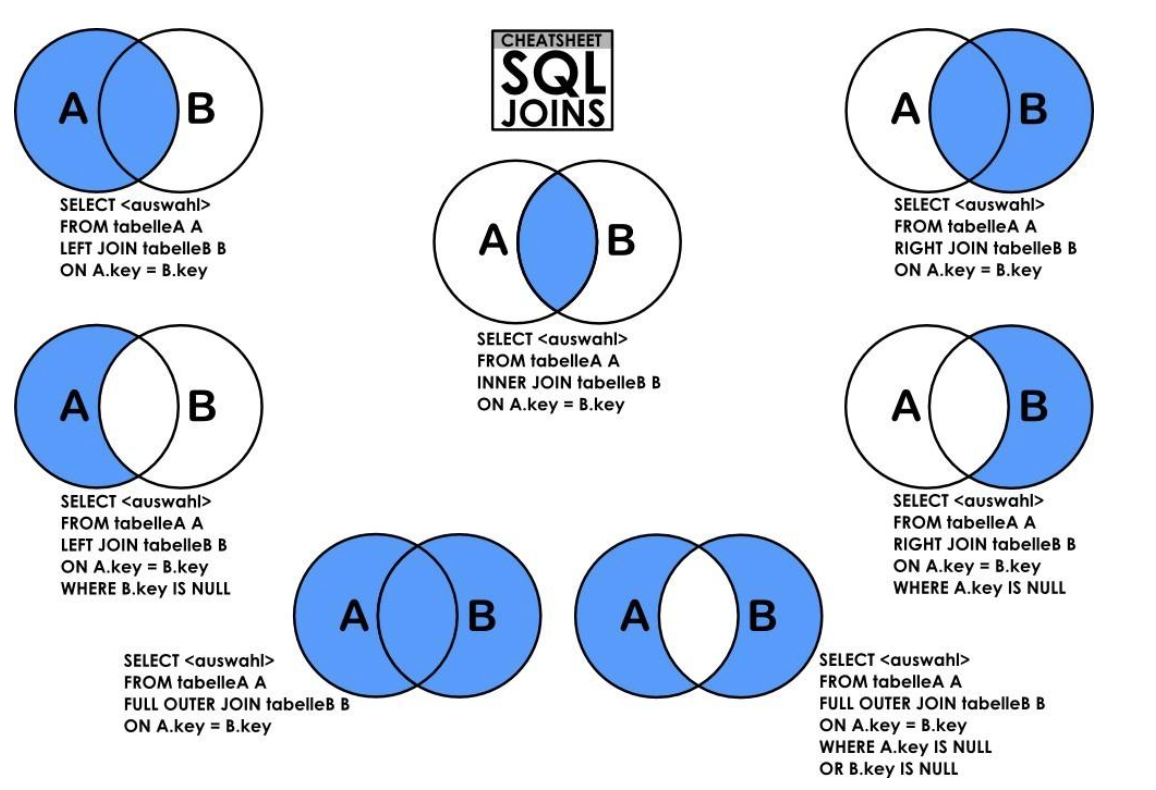
            n=n/5;

        }

        return cnt;

    }

#### LeetCode[175. Combine Two Tables](https://leetcode-cn.com/problems/combine-two-tables/)



1. 使用JOIN连接多表以后要用on限定条件，不能直接使用where。
2. 笛卡尔积CROSS JOIN是将左边表的每一项数据都和右边表的每一项数据连接，会产生m\*n条数据。和全连接FULL OUTER JOIN不一样

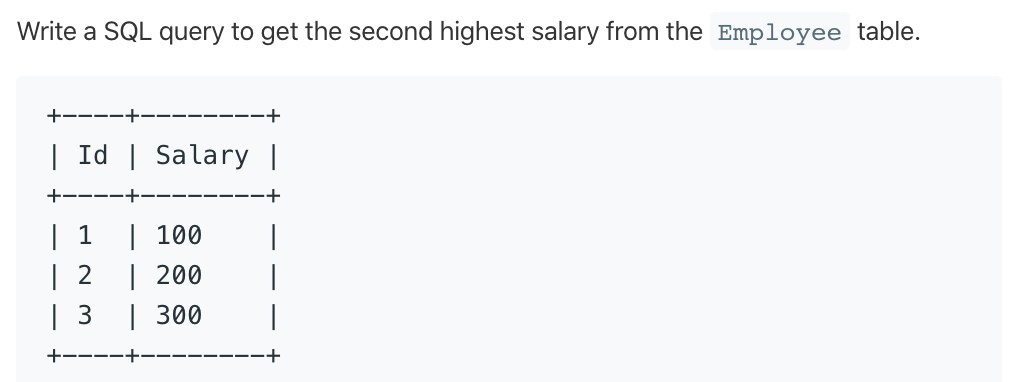
源代码如下：

select FirstName,LastName,City,State

from Person LEFT JOIN Address on Person.PersonId=Address.PersonId

题目四.

#### LeetCode[176. Second Highest Salary](https://leetcode-cn.com/problems/second-highest-salary/)



这题要学习指定项去重，排序，判断查询结构是否为null等内容。

* **distinct**字段必须出现在查询语句的开始，后面跟表中待去重的目标字段。如果要查询user表中不重复的名字，用如下mySQL语句

**select distinct name from user;**

**也可以用distinct结合count来统计表中不重复人的个数**

**select count(distinct name) from user;**

**如果distinct后跟了好几个字段，那就是要求去除这些字段都重复的值。**

* **order by 可以对表中的指定字段进行排序，也可以多字段排序(每个排序条件用逗号分开)如：**

**select \* from table order by id desc,name asc;（desc：降序，asc：升序**

* limit，offset：可以用来限定输出哪几行的数据，例如：

1.select \* from student limit 9,4

2.slect \* from student limit 4 offset 9

两条语句都表示返回返回第9，10，11，12行四条数据

* IF() IFNULL() NULLIF() ISNULL()函数的用法

1.SELECT IF(TRUE,'A','B'); 三个表达式，如果第一个为true，第输出第二个表达式，否则第三个

2.SELECT IFNULL(‘A','B'); 两个表达式 如果第一个表达式为null，则输出第二个表达式，否则输出第一个

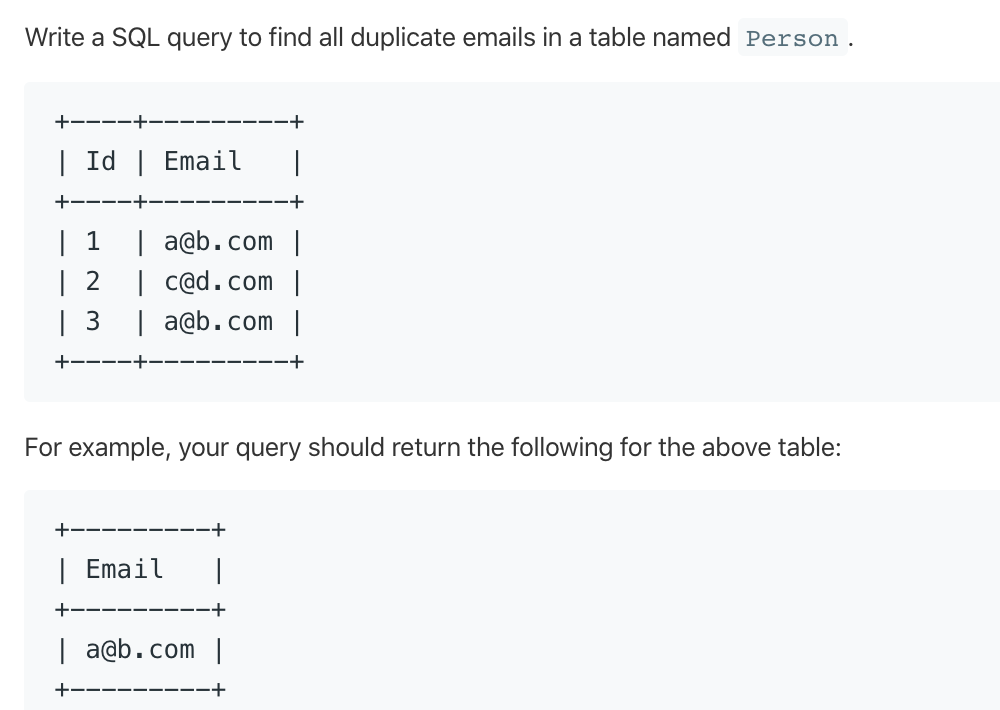
3.SELECT NULLIF('A','A'); 两个表达式 如果表达式相等，就返回第一个表达式，如果不想等就返回null

4.SELECT ISNULL(NULL); 一个表达式，如果为null就返回1，否则换回0

注意：这里的表达式返回值不能超过一行，否则无法判断，换句话说，如果‘A’里又是一个查询语句，那么返回值不能超过1.

题目五.

#### LeetCode[182. Duplicate Emails](https://leetcode-cn.com/problems/duplicate-emails/)



* 这题需要用到group by知识点，用于对指定类别进行分组，例如：

select 类别, sum(数量) as 数量之和

from A

group by 类别

* 和聚合函数的使用,因为select指定的字段必须是“分组”，其他字段只能包含在聚合函数中

1. sum，max，min，avg，count//统计记录数

* having和where的区别：前者是在分组过后过滤数据，后者则是在分组之前将不符合条件的去掉，where条件不能包含聚合函数

源代码如下：

1,先用group by对Email分组获得Email的个数的临时表，然后对这个临时表使用查询语句，判断个数大于1的为duplict email。注意在中间的临时表写完后，要给他取别名（alias）

select Email

from

(

    select Email,count(Email) as num

    from Person

    group by Email

) as t1

where num>1

2直接食用having更简便，分组完后直接判断。

select Email

from Person

group by Email

having count(Email)>1