＃包括< stdio.h中>

＃包括< string.h中>

＃包括< stdlib.h中>

{

int c\_custkey; //顾客编号

char c\_mkgsegment [ 20 ]; //对应的某个市场部门

}顾客; //顾客结构体

类型定义 结构

{

int o\_orderkey; //订单号

int o\_custkey; //顾客编号

char o\_orderdate [ 10 ]; //订货日期

}命令; //订单

类型定义 结构

{

int l\_orderkey; //订单号

double l\_extendedprice; //额外价格

char l\_shipdate [ 10 ]; //发货日期

} lineitem; //商品信息

类型定义 结构

{

int l\_orderkey; //订单号

char o\_orderdate [ 10 ]; //订货日期

double l\_extendedprice; //额外价格

} select\_result;

customer \* read\_customer\_txt（）//读取customer。txt内容

{

文件 \* fp;

客户\* a = NULL ;

a =（客户\*）malloc（101 \* sizeof（客户））;

int i = 0 ;

字符 b;

fp = fopen（“ customer.txt ”，“ r ”）;

如果（NULL == fp）

{

printf（“无法打开customer.txt！”）;

返回 NULL ;

}

而（！feof（fp））

{

fscanf（fp，“ ％d％c％s ”，＆a [i] 。c\_custkey，＆b，＆a [i] 。c\_mkgsegment）;

i ++;

}

fclose（fp）;

返回一个;

}

orders \* read\_orders\_txt（）//读取orders.txt内容

{

int i = 0 ;

订单\* a = NULL ;

a =（订单\*）malloc（4001 \* sizeof（订单））;

字符 b，c;

长 长 D组;

文件 \* fp;

fp = fopen（“ orders.txt ”，“ r ”）;

如果（fp == NULL）

{

printf（“无法打开orders.txt！”）;

返回 NULL ;

}

而（！feof（fp））

{

fscanf（fp，“ ％d％c％lld％c％s ”，＆a [i] 。o\_orderkey，＆b，＆d，＆c，＆a [i] 。o\_orderdate）;

a [i]。o\_custkey = d％100 ;

i ++;

}

fclose（fp）;

返回一个;

}

lineitem \* read\_lineitem\_txt（）//读取lineitem.txt内容

{

文件 \* fp;

lineitem \* l = NULL ;

l =（lineitem \*）malloc（1001 \* sizeof（lineitem））;

int i = 0 ;

字符 b，c;

fp = fopen（“ lineitem.txt ”，“ r ”）;

如果（fp == NULL）

{

printf（“无法打开lineitem.txt！”）;

返回 NULL ;

}

而（！feof（fp））

{

fscanf（fp，“ ％d％c％lf％c％s ”，＆l [i] 。l\_orderkey，＆c，＆l [i] 。l\_extendedprice，＆b，＆l [i] 。l\_shipdate）;

i ++;

}

fclose（fp）;

返回 l;

}

select\_result \* 选择（客户\* cus，订单\* ord，lineitem \*项目，char \* order\_date，char \* ship\_date，char \* mktsegment）//进行选择

{

int i，j，k; //循环

int sindex = 0 ; // slct\_res的索引

双倍 TMP; //用于排序

select\_result \* slct\_res = NULL ;

slct\_res =（select\_result \*）malloc（1001 \* sizeof（select\_result））;

//创建slct\_res

对于（i = 0 ; i < 100 ; i ++）{

if（strcmp（cus [i] 。c\_mkgsegment，mktsegment）== 0）{

对于（j = 0 ; j < 4000 ; j ++）{

如果（CUS [i]中。条件是c\_custkey == ORD [J]。o\_custkey && 的strcmp（ORD [J]。o\_orderdate，order\_date的）< 0）{

对于（k = 0 ; k < 1000 ; k ++）{

if（ord [j] 。o\_orderkey == item [k] 。l\_orderkey && strcmp（item [k] 。l\_shipdate，ship\_date）> 0）{

slct\_res [sindex]。l\_extendedprice =项目[k]。l\_extendedprice ;

slct\_res [sindex]。l\_orderkey = item [k]。l\_orderkey ;

strcpy（slct\_res [sindex] 。o\_orderdate，ord [j] 。o\_orderdate）;

sindex ++;

}

}

}

}

}

}

//现在，sindex是slct\_res的长度

//删除重复元素

对于（i = 0 ; i <sindex- 1 ; i ++）{

对于（j = i + 1 ; j <sindex; j ++）{

if（slct\_res [i] 。l\_orderkey == slct\_res [j] 。l\_orderkey && strcmp（slct\_res [i] 。o\_orderdate，slct\_res [j] 。o\_orderdate）== 0）{

slct\_res [i]。l\_extendedprice = slct\_res [i]。l\_extendedprice + slct\_res [j]。l\_extendedprice ;

slct\_res [j]。l\_extendedprice = 0 ;

}

}

}

//基于l\_extendedprice的Bubble\_Sort

对于（i = 0 ; i <sindex- 1 ; i ++）{

对于（j = 0 ; j <sindex- 1 ; j ++）{

如果（slct\_res [J]。l\_extendedprice <slct\_res [J + 1 ]。l\_extendedprice）{

tmp = slct\_res [j + 1 ]。l\_extendedprice ;

slct\_res [j + 1 ]。l\_extendedprice = slct\_res [j]。l\_extendedprice ;

slct\_res [j]。l\_extendedprice = tmp;

}

}

}

返回 slct\_res;

}

int change\_argv\_to\_number（char s []）//将命令行里读入的数字字符串转换为整形数字

{

int i = 0 ;

int number = 0 ;

而（s [i]！= ' \ 0 '）

{

如果（i == 0）

数=（s [i] -48）;

其他

数字=数字\* 10 +（s [i] -48）;

i ++;

}

返回编号；

}

int main（int argc，char \* argv []）// argc表示输入内容的总个数，argv []内部保存着输入的内容

{

int i，j;

INT NUM;

int limit = 3 ;

字符 order\_date [ 15 ];

char ship\_date [ 10 ];

char mktsegment [ 20 ];

select\_result \*结果= NULL ;

客户\* cus = NULL ; //指向客户表的指针

订单\* ord = NULL ; //指向订单表的指针

lineitem \* item = NULL；//指向产品表的指针

cus = read\_customer\_txt（）; //读取customer.txt的内容，引入客户表

ord = read\_orders\_txt（）; //读取orders.txt的内容，，导入订单表

item = read\_lineitem\_txt（）; //读取lineitem.txt的内容，导入产品表

num = change\_argv\_to\_number（argv [ 4 ]）; //总计算的次数

对于（i = 0 ; num> 0 ; num-，i = i + 4）

{

strcpy（mktsegment，argv [ 5 + i]）;

strcpy（ship\_date，argv [ 7 + i]）;

strcpy（order\_date，argv [ 6 + i]）;

限制= change\_argv\_to\_number（argv [ 8 + i]）;

//printf（“％d mktsegment：％s order\_date：％s ship\_date：％s limit：％d \ n”，num，mktsegment，order\_date，ship\_date，limit）;

结果= 选择（cus，ord，item，order\_date，ship\_date，mktsegment）;

printf（“ l\_orderkey | o\_orderdate | revenue \ n ”）;

对于（j = 0 ; j <limit; j ++）

{

如果（结果[j] 。l\_extendedprice == 0）

printf（“ null | null | null \ n ”）;

其他

的printf（“ ％-10D | ％-11s | ％-20.2lf \ n ”，结果[J]。L\_ORDERKEY，结果[J]。o\_orderdate，结果[J]。l\_extendedprice）;

}

免费（结果）；

结果= NULL ;

}

返回 0 ;

}