Дан pl/sql-тип: **type** tNumTbl **is table of number index by pls\_integer;**

1. Написать функцию с одним входящим параметром, с типом tNumTbl, возвращающую минимальное значение из непустых элементов этой таблицы.

**Function get\_min(pVal** tNumTbl**) return number;**

1. Написать функцию с двумя параметрами (оба с типом tNumTbl), возвращающую таблицу (tNumTbl) уникальных чисел, содержащихся в двух входных параметрах (включая пустое значение, если таковое встретилось).

**Function get\_unique(pVal1** tNumTbl, **pVal2** tNumTbl**) return** tNumTbl**;**

Дана таблица операций

|  |  |  |
| --- | --- | --- |
| ID  Ид операции | NAME  Название операции | TYPE  Тип: основная (MAIN), дочерняя (CHILD) |
| 1 | Первая | MAIN |
| 2 | Вторая | MAIN |
| 3 | Простая | CHILD |
| 4 | Незначительная | CHILD |
| 5 | Несущественная | CHILD |
| 6 | Легкая | CHILD |
| 7 | Подчиненная | CHILD |

И таблица приоритета операций

|  |  |  |  |
| --- | --- | --- | --- |
| OPER\_ID  Ид операции | MAIN\_POSITION  Приоритет основной операции | CHILD\_POSITION  Приоритет дочерней операции внутри основной | OPER\_POSITION  Сквозной приоритет в общем списке операций |
| 7 | 1 | -2 | 1 |
| 3 | 1 | -1 | 2 |
| **1** | **1** | **0** | **3** |
| 4 | 1 | 1 | 4 |
| **2** | **2** | **0** | **5** |
| 5 | 2 | 1 | 6 |
| 6 | 2 | 2 | 7 |

1. Написать функцию возвращающую курсор с наименованиями основных операций, у которых количество связанных дочерних операций с приоритетом выше, чем у основной, равно количеству связанных дочерних операций с приоритетом ниже, чем у основной, отсортированный в порядке приоритета полученных операций.

**Function get\_oper\_list return sys\_refcursor;**

1. Разработать процедуру для изменения приоритета указанной операции с двумя параметрами: ID операции и тип изменения приоритета (повысить или понизить).

**Procedure Change\_Position(pID number, pInc boolean);**

**При увеличении приоритета любой операции, ее сквозной приоритет уменьшается на 1, а при уменьшении приоритета, увеличивается на 1.**

При изменении приоритета основной операции, она меняется местами с вышестоящей или нижестоящей основной операцией (если таковая имеется). При этом, все привязанные к ней дочерние операции «переезжают» вместе с ней.

При изменении приоритета дочерней операции, она меняется местами с соседней дочерней, принадлежащей этой же основной. Если следом по приоритету нет такой дочерней операции, то есть там находится соседняя основная или ее дочерняя, то изменяемая дочерняя операция должна перепривязаться к этой соседней основной операции. Если нет операции, с которой можно поменяться местами, то ничего не должно происходить. Это относится как для основной, так и для дочерней операции.

**Например**, при повышении приоритета у операции с ID=2 должно получиться:

|  |  |  |  |
| --- | --- | --- | --- |
| OPER\_ID  Ид операции | MAIN\_POSITION  Приоритет основной операции | CHILD\_POSITION  Приоритет дочерней операции внутри основной | OPER\_POSITION  Сквозной приоритет в общем списке операций |
| **2** | **2** | **0** | **1** |
| 5 | 2 | 1 | 2 |
| 6 | 2 | 2 | 3 |
| 7 | 1 | -2 | 4 |
| 3 | 1 | -1 | 5 |
| **1** | **1** | **0** | **6** |
| 4 | 1 | 1 | 7 |

А при понижении (от исходного состояния примера) приоритета у операции с ID=4 должно получиться:

|  |  |  |  |
| --- | --- | --- | --- |
| OPER\_ID  Ид операции | MAIN\_POSITION  Приоритет основной операции | CHILD\_POSITION  Приоритет дочерней операции внутри основной | OPER\_POSITION  Сквозной приоритет в общем списке операций |
| 7 | 1 | -2 | 1 |
| 3 | 1 | -1 | 2 |
| **1** | **1** | **0** | **3** |
| **2** | **2** | **0** | **4** |
| 4 | 2 | 1 | 5 |
| 5 | 2 | 2 | 6 |
| 6 | 2 | 3 | 7 |