# 综合案例

## 需求1: 编写 PL/SQL，用水吨数 12 吨，业主类型为 1，计算阶梯水费。

### 目标

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 输入 | | | |  | 结果 | | |
| 业主类型 | 价格 | 下限 | 上限 |  | 应付金额 | | |
| 1 | 3.45 | 5 | 10 |  | 38.4 | | |
| 1 | 2.45 | 0 | 5 |  |  |  |  |
| 1 | 4.45 | 10 |  |  |  |  |  |
| 2 | 3.87 | 0 | 5 |  |  |  |  |
| 2 | 5.87 | 10 |  |  |  |  |  |
| 2 | 4.87 | 5 | 10 |  |  |  |  |
| 3 | 4.36 | 0 | 5 |  |  |  |  |
| 3 | 5.36 | 5 | 10 |  |  |  |  |
| 3 | 6.36 | 10 |  |  |  |  |  |

### 分析

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分析1 用水吨数 12 吨，业主类型为 1，计算阶梯水费。 | | | | | | | | | | | | | | | | | | | | |
|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 输入 | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 业主类型 | 价格 | 下限 | 上限 | 1 过滤 业主类型=1 2 按照 下限排序 | 业主类型 | 价格 | 下限 | 上限 | 如何计算阶梯水费? 1 声明总金额变量 2 遍历价格集合 2.1 如果 上限为null 或 总吨数<上限,   吨数 = 总吨数 - 当前行的下限  2.2 否则 吨数 = 上限 - 下限 | 业主类型 | 价格 | 下限 | 上限 | 吨数 | 阶段水费 | 应付总水费 |
|  |  |  |  | 1 | 3.45 | 5 | 10 | 1 | 2.45 | 0 | 5 | 1 | 2.45 | 0 | 5 | 上限 - 下限 = 5 | 12.25 | 38.4 |
|  |  |  |  | 1 | 2.45 | 0 | 5 | 1 | 3.45 | 5 | 10 | 1 | 3.45 | 5 | 10 | 上限 - 下限 = 5 | 17.25 |
|  |  |  |  | 1 | 4.45 | 10 |  | 1 | 4.45 | 10 |  | 1 | 4.45 | 10 |  | 总吨数 - 下限 = 2 | 8.9 |
|  |  |  |  | 2 | 3.87 | 0 | 5 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 2 | 5.87 | 10 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 2 | 4.87 | 5 | 10 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 3 | 4.36 | 0 | 5 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 3 | 5.36 | 5 | 10 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 3 | 6.36 | 10 |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分析2 用水吨数 9 吨，业主类型为 1，计算阶梯水费。 | | | | | | | | | | | | | | | | | | | | |
|
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 业主类型 | 价格 | 下限 | 上限 | 1 过滤 业主类型=1 2 按照 下限排序 | 业主类型 | 价格 | 下限 | 上限 | 如何计算阶梯水费? 1 声明总金额变量 2 遍历价格集合 2.1 如果 上限为null 或 总吨数<上限,   吨数 = 总吨数 - 当前行的下限  2.2 否则 吨数 = 上限 - 下限 | 业主类型 | 价格 | 下限 | 上限 | 吨数 | 阶段水费 | 应付总水费 |
|  |  |  |  | 1 | 3.45 | 5 | 10 | 1 | 2.45 | 0 | 5 | 1 | 2.45 | 0 | 5 | 上限 - 下限 = 5 | 12.25 | 26.05 |
|  |  |  |  | 1 | 2.45 | 0 | 5 | 1 | 3.45 | 5 | 10 | 1 | 3.45 | 5 | 10 | 总吨数 - 下限 = 4 | 13.8 |
|  |  |  |  | 1 | 4.45 | 10 |  | 1 | 4.45 | 10 |  | 1 | 4.45 | 10 |  |  |  |
|  |  |  |  | 2 | 3.87 | 0 | 5 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 2 | 5.87 | 10 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 2 | 4.87 | 5 | 10 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 3 | 4.36 | 0 | 5 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 3 | 5.36 | 5 | 10 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 3 | 6.36 | 10 |  |  |  |  |  |  |  |  |  |  |  |  |

### 实现

路径

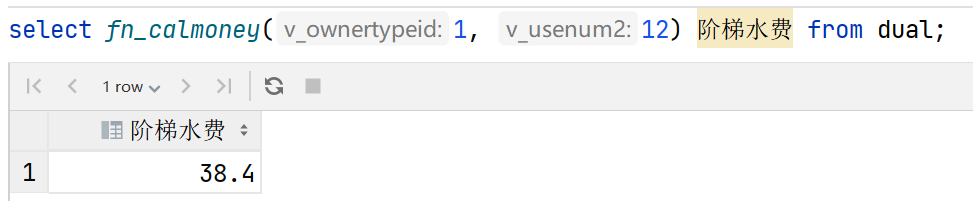
|  |
| --- |
| -- todo 需求1: 编写 PL/SQL，用水吨数 12 吨，业主类型为 1，计算阶梯水费。 -- todo 1 声明变量: v\_ownertypeid 业主类型ID, v\_usenum2 总吨数, v\_money 总金额, cur\_pricetable 带业主类型参数的价格游标, v\_pricetable 价格表行变量 -- todo 2 赋值 业主类型ID=1, 总吨数=12, 总金额初始化为 0 -- todo 3 遍历 带业主类型参数的价格游标集合 -- todo 3.1 如果 上限为null 或 总吨数<当前行上限 -- todo 3.1.1 总金额 累加 (总吨数 - 下限) \* 价格 -- todo 3.1.2 退出循环 exit -- todo 3.2 否则 -- todo 3.2.1 总金额 累加 (上限 - 下限) \* 价格 |

实现

|  |
| --- |
| declare  -- todo 需求1: 编写 PL/SQL，用水吨数 12 吨，业主类型为 1，计算阶梯水费。  -- todo 1 声明变量: v\_ownertypeid 业主类型ID, v\_usenum2 总吨数, v\_money 总金额, cur\_pricetable 带业主类型参数的价格游标, v\_pricetable 价格表行变量  v\_ownertypeid number;  v\_usenum2 number(10, 2);  v\_money number(10, 2);  cursor cur\_pricetable(v\_type number) is  select \* from t\_pricetable  where ownertypeid=v\_type  order by minnum asc;  v\_pricetable t\_pricetable%rowtype; begin  -- todo 2 赋值 业主类型ID=1, 总吨数=12, 总金额初始化为 0  v\_ownertypeid := 1;  v\_usenum2 := 12;  v\_money := 0;   -- todo 3 遍历 带业主类型参数的价格游标集合  for v\_pricetable in cur\_pricetable(v\_ownertypeid)  loop  if v\_pricetable.MAXNUM is null or v\_usenum2<v\_pricetable.MAXNUM then  -- todo 3.1 如果 上限为null 或 总吨数<当前行上限  -- todo 3.1.1 总金额 累加 (总吨数 - 下限) \* 价格  v\_money := v\_money + (v\_usenum2 - v\_pricetable.MINNUM) \* v\_pricetable.PRICE;  -- todo 3.1.2 退出循环 exit  exit;  else  -- todo 3.2 否则  -- todo 3.2.1 总金额 累加 (上限 - 下限) \* 价格  v\_money := v\_money + (v\_pricetable.MAXNUM - v\_pricetable.MINNUM) \* v\_pricetable.PRICE;  end if;  end loop;   DBMS\_OUTPUT.PUT\_LINE('阶梯水费为: ' || v\_money); end; |

## 需求2: 存储函数综合案例：创建计算阶梯水费的函数，参数为业主类型、吨数。

### 目标



### 分析

明确函数需要哪些参数?

答: 业主类型 和 总吨数

明确函数返回的结果?

答: 阶梯水费

|  |
| --- |
| -- todo 需求2: 存储函数综合案例：创建计算阶梯水费的函数，参数为业主类型、吨数。 -- todo 0 创建函数 fn\_calmoney 指定参数(业主类型, 吨数) 和 返回结果 -- todo 1 声明变量: v\_money 总金额, cur\_pricetable 带业主类型参数的价格游标, v\_pricetable 价格表行变量 -- todo 2 赋值 总金额初始化为 0 -- todo 3 遍历 带业主类型参数的价格游标集合 -- todo 3.1 如果 上限为null 或 总吨数<当前行上限 -- todo 3.1.1 总金额 累加 (总吨数 - 下限) \* 价格 -- todo 3.1.2 退出循环 exit -- todo 3.2 否则 -- todo 3.2.1 总金额 累加 (上限 - 下限) \* 价格 -- todo 4 返回结果 总金额  -- todo 测试此函数  select fn\_calmoney(1,12) from dual; |

### 实现

|  |
| --- |
| -- todo 需求2: 存储函数综合案例：创建计算阶梯水费的函数，参数为业主类型、吨数。 -- todo 0 创建函数 fn\_calmoney 指定参数(业主类型, 吨数) 和 返回结果 create or replace function fn\_calmoney(v\_ownertypeid number, v\_usenum2 number) return number is  -- todo 1 声明变量: v\_money 总金额, cur\_pricetable 带业主类型参数的价格游标, v\_pricetable 价格表行变量  v\_money number(10, 2);  cursor cur\_pricetable(v\_type number) is  select \* from t\_pricetable  where ownertypeid=v\_type  order by minnum asc;  v\_pricetable t\_pricetable%rowtype; begin -- todo 2 赋值 业主类型ID=1, 总吨数=12, 总金额初始化为 0  v\_money := 0;   -- todo 3 遍历 带业主类型参数的价格游标集合  for v\_pricetable in cur\_pricetable(v\_ownertypeid)  loop  if v\_pricetable.MAXNUM is null or v\_usenum2<v\_pricetable.MAXNUM then  -- todo 3.1 如果 上限为null 或 总吨数<当前行上限  -- todo 3.1.1 总金额 累加 (总吨数 - 下限) \* 价格  v\_money := v\_money + (v\_usenum2 - v\_pricetable.MINNUM) \* v\_pricetable.PRICE;  -- todo 3.1.2 退出循环 exit  exit;  else  -- todo 3.2 否则  -- todo 3.2.1 总金额 累加 (上限 - 下限) \* 价格  v\_money := v\_money + (v\_pricetable.MAXNUM - v\_pricetable.MINNUM) \* v\_pricetable.PRICE;  end if;  end loop;   return v\_money; end;  select fn\_calmoney(1, 12) 阶梯水费 from dual; |

## 存储过程综合案例。

### 目标

需求：增加业主信息时，同时在账务表（account）增加一条记录，年份与月份 为当前日期的年月，初始值（num0）为 0,其它字段信息（区域）与 t\_owners表一致

### 分析

难点分析：

1. 如何取得年和月
2. 如何取得区域 ID 到 address 表查询

用 to\_char()函数

参数中没有直接提供区域 ID，我们可以通过 addressid

### 实现

|  |
| --- |
| -- todo 需求：增加业主信息时，同时在账务表（account）增加一条记录，年份与月份 为当前日期的年月，初始值（num0）为 0,其它字段信息（区域）与 t\_owners表一致 -- todo 1 创建存储过程 pro\_owners\_add -- todo 2 初始化变量 v\_areaid, v\_year, v\_month -- todo 3 设置变量的值 v\_year, v\_month -- todo 4 通过select into 设置 v\_areaid的值 -- todo 5 通过序列获取 owneruuid的值 -- todo 6 向 业主表插入数据 -- todo 7 向 账务表插入数据 -- todo 8 提交 -- todo 9 异常处理 当出现 NO\_DATA\_FOUND 异常, 回滚撤销 |

|  |
| --- |
| create sequence seq\_owners increment by 10 start with 100;  -- todo 需求：增加业主信息时，同时在账务表（account）增加一条记录，年份与月份 为当前日期的年月，初始值（num0）为 0,其它字段信息（区域）与 t\_owners表一致 -- todo 1 创建存储过程 pro\_owners\_add create or replace procedure pro\_owners\_add(  v\_name varchar2,  v\_addressid number,  v\_housenumber varchar2,  v\_watermeter varchar2,  v\_ownertypeid number  , v\_owneruuid out number ) as  -- todo 2 初始化变量 v\_areaid, v\_year, v\_month  v\_areaid number;  v\_year char(4);  v\_month char(2); begin  -- todo 3 设置变量的值 v\_year, v\_month  v\_year := to\_char(sysdate, 'yyyy');  v\_month := to\_char(sysdate, 'mm');   -- todo 4 通过select into 设置 v\_areaid的值  select areaid into v\_areaid from t\_address t where id=v\_addressid;   -- todo 5 通过序列获取 owneruuid的值  select seq\_owners.nextval into v\_owneruuid from dual;   -- todo 6 向 业主表插入数据  insert into t\_owners (id, name, addressid, housenumber, watermeter, adddate, ownertypeid)  values (v\_owneruuid, v\_name, v\_addressid, v\_housenumber, v\_watermeter, sysdate, v\_ownertypeid);   -- todo 7 向 账务表插入数据  insert into t\_account (id, owneruuid, ownertype, areaid, year, month, num0)  values (seq\_account.nextval, v\_owneruuid, v\_ownertypeid, v\_areaid, v\_year, v\_month, 0);   -- todo 8 提交  commit;   -- todo 9 异常处理 当出现 no\_data\_found 异常, 回滚撤销  exception  when no\_data\_found then v\_owneruuid := -1;  rollback; end; |

|  |
| --- |
| declare  v\_owneruuid varchar2(32); begin  pro\_owners\_add('铁扇公主', 1, '火焰山', '1006', 2, v\_owneruuid);   DBMS\_OUTPUT.PUT\_LINE('添加成功! owneruuid = ' || v\_owneruuid); end; |