Задание 2. Применение аналитических функций ORACLE в запросах на выборку

Задание 2.1. В БД есть таблица **`tBalance`** проводок по счетам. Требуется получить баланс после совершения каждой проводки (к стартовой сумме надо прибавить или вычесть сумму первой проводки, к полученной сумме добавить или вычесть сумму второй и т.д. Другими словами, накапливать результат. В стандартном SQL задача не решается, т.к. надо каким-то образом хранить сумму, но это можно сделать аналитической функцией Oracle.

Используя тестовые данные, под своей учетной записью в APEX создать и заполнить таблицу **`tBalance.`** Написать запрос, выводящий состояние баланса после совершения каждой проводки. Использовать оконную функцию OVER и конструкцию PARTITION BY.

Тестовые данные:

```
create table tBalance (
 id    number,
 account varchar2(20),
 value  number);

insert into tBalance values (1,'01',100);
insert into tBalance values (2,'01',200);
insert into tBalance values (3,'01',-100);
insert into tBalance values (4,'01',200);
insert into tBalance values (5,'01',100);
insert into tBalance values (6,'01',-100);
insert into tBalance values (7,'01',100);

insert into tBalance values (8, '02',10);
insert into tBalance values (9, '02',20);
insert into tBalance values (10,'02',-10);
insert into tBalance values (11,'02',-20);
insert into tBalance values (12,'02',10);
insert into tBalance values (13,'02',-10);
insert into tBalance values (14,'02',10);
```

Задание 2.2. Получить список служащих, проранжированных по количеству совершенных рейсов. Использовать оконную функцию OVER и ранжирующую функцию DENSE_RANK.

Задание 2.3. Получить распределение возраста служащих по отрезкам 18 ÷ 29; 30 ÷ 39; 40 ÷49; 50 ÷ 59; 60 ÷ 70. Использовать конструкцию **`CASE WHEN age BETWEEN значение_1 AND значение_1 THEN 1 ELSE 0 END`**