**ЛАБОРАТОРНАЯ РАБОТА №6**  
Программные пакеты PL/SQL.

**Выполнил:**

студент 3 курса 13 группы кафедры ТП.

Петров Андрей Александрович

# ЗАДАЧА РАБОТЫ:

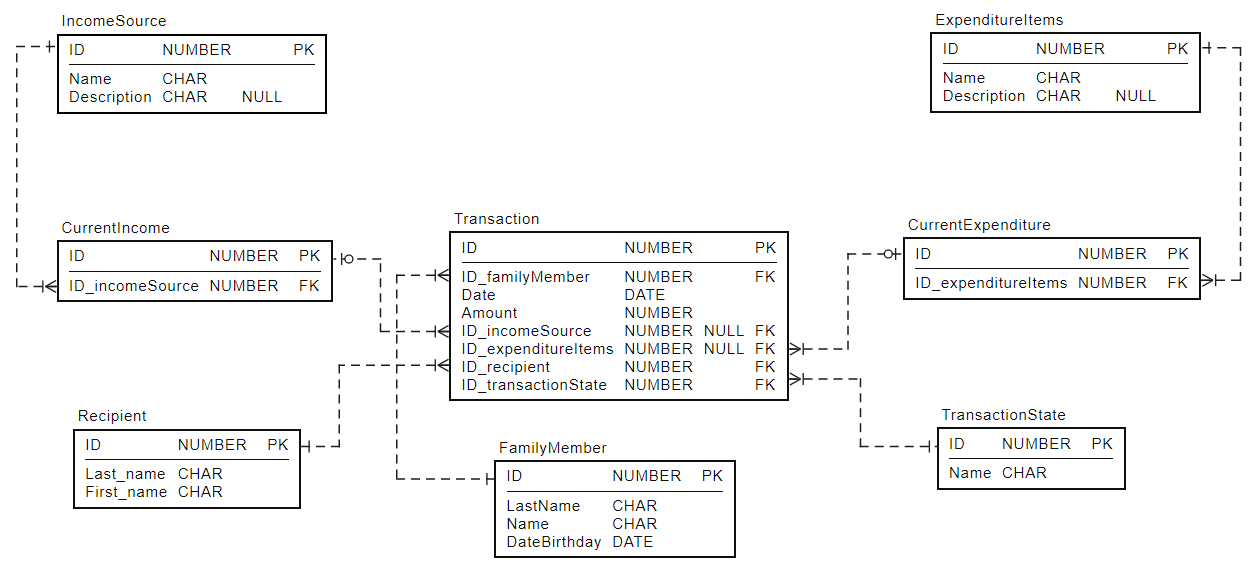
Создать программный пакет PL/SQL, содержащий следующие программные объекты.

1. Процедура вывода списка расходов за указанный период. Входные параметры – начало и конец периода. Выходной параметр – курсор результирующего набора: имя члена семьи, перечень расходных операций, общая сумма расхода члена семьи.

2. Функция вычисления остатка для указанного члена семьи. Входной параметр – имя члена семьи, выходной параметр – сумма остатка, которая определяется как разность суммы доходов и суммы расходов члена семьи.

3. Триггер для проверки корректности расходов несовершеннолетних членов семьи. Однократная сумма расходов несовершеннолетних членов семьи не должна превышать 50 руб.

# ХОД РАБОТЫ:



Cпецификация пакета:

CREATE OR REPLACE PACKAGE HOME\_FINANCE  
AS PROCEDURE *EXP\_BETWEEN* (START IN DATE, END IN DATE, RESULT IN OUT SYS\_REFCURSOR);  
FUNCTION *GET\_RESIDUE* (NAME IN VARCHAR2) RETURN NUMBER;  
END HOME\_FINANCE;  
/

1. Процедура вывода списка расходов за указанный период. Входные параметры – начало и конец периода. Выходной параметр – курсор результирующего набора: имя члена семьи, перечень расходных операций, общая сумма расхода члена семьи.

PROCEDURE EXP\_BETWEEN (START\_ IN DATE, END\_ IN DATE, RESULT IN OUT SYS\_REFCURSOR)  
IS  
BEGIN  
 OPEN RESULT FOR WITH NAME\_TYPE AS (  
 SELECT FM.*"ID"* AS FM\_ID,  
 FM.*"LastName"* || ' ' || FM.*"Name"* AS NAME,  
 EX.*"ID"* AS EXPENSE\_ID,  
 EX.*"Name"* FROM "FamilyMember" FM  
 CROSS JOIN "CurrentExpenditure" EX),  
 TRNS AS (SELECT *FAMILY\_MEMBER\_ID* AS FM\_ID,  
 *"ID\_expenditureItems"*,  
 *SUM*(*"Amount"*) AS AMOUNT  
 FROM "Transaction"  
 WHERE *"Date"* BETWEEN *START\_* AND *END\_* AND *"ID\_expenditureItems"* IS NOT NULL  
 GROUP BY (*"ID\_familyMember"*, *"ID\_expenditureItems"*)  
 )  
 SELECT NT."Name" AS NAME\_,  
 *NVL*(T.AMOUNT, 0) AS AMOUNT  
 FROM NAME\_TYPE NT  
 LEFT JOIN TRNS T  
 ON (T.FM\_ID = NT.FM\_ID AND  
 T.EXPENSE\_ID = NT.EXPENSE\_ID)  
 ORDER BY NAME\_;  
END EXP\_BETWEEN;

2. Функция вычисления остатка для указанного члена семьи. Входной параметр – имя члена семьи, выходной параметр – сумма остатка, которая определяется как разность суммы доходов и суммы расходов члена семьи.

FUNCTION GET\_RESIDUE(NAME IN VARCHAR2)  
 RETURN NUMBER  
IS  
 RES NUMBER(10,2);  
BEGIN  
 SELECT  
 *NVL*(*SUM*((CASE WHEN T.*"ID\_incomeSource"* IS NULL  
 THEN -T.*"Amount"* ELSE T.*"Amount"* END)), 0)  
 INTO RES  
 FROM "Transaction" T JOIN "FamilyMember" FM  
 ON (T.*"ID\_familyMember"* = FM.*"ID"*)  
 WHERE FM.*"LastName"* || ' ' || FM.*"Name"* = NAME;  
RETURN RES;  
END;  
  
END HOME\_FINANCE;  
/

3. Триггер для проверки корректности расходов несовершеннолетних членов семьи. Однократная сумма расходов несовершеннолетних членов семьи не должна превышать 50 руб.

CREATE OR REPLACE TRIGGER CHECK\_MINOR\_EXP  
BEFORE INSERT OR UPDATE ON "Transaction"  
FOR EACH ROW  
DECLARE  
 D DATE;  
BEGIN  
 IF ( ( :NEW."ID\_expenditureItems" IS NOT NULL ) AND ( :NEW."ID\_incomeSource" IS NULL ) )  
 AND (:NEW."Amount" > 50)  
 AND (*INSERTING* OR *UPDATING*('Date') OR *UPDATING*('Amount')  
 OR *UPDATING*('ID\_incomeSource') OR *UPDATING*('ID\_expenditureItems')  
 OR *UPDATING*('ID\_familyMember') )  
 THEN  
 SELECT *"DateBirthday"* INTO D  
 FROM "FamilyMember"  
 WHERE *"ID"* = :NEW."ID\_familyMember";  
  
 IF *FLOOR*(*MONTHS\_BETWEEN*(:NEW.*"Date"*, D)/12) < 18  
 THEN *RAISE\_APPLICATION\_ERROR*(-20011,'EXPENSES OF A MINOR SHOULD NOT  
 EXCEED 50 RUBLES.');  
 END IF;  
 END IF;  
END;  
/

Изображение выглядит как текст, снимок экрана, монитор, черный

Автоматически созданное описание