Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

**Лабораторная работа №2**

Выполнили: студенты IV курса

ИВТ, гр. ИП-713

Михеев Н.А.

Трусов К.В.

Устенко Д.А.

Проверил: старший преподаватель

кафедры ПМиК

Грязнов Н. Г.

Новосибирск, 2020 г.

1. Напишите сценарий для вывода имен и комиссионных первых двух продавцов, работающих не в Лондоне.

DECLARE

    vname VARCHAR2(10);

    vcomm NUMBER(7,2);

    CURSOR cur1 IS SELECT sname, comm FROM sal WHERE city <> 'London';

BEGIN

    DBMS\_OUTPUT.enable;

    OPEN cur1;

    IF NOT cur1%ISOPEN THEN OPEN CUR1; END IF;

    LOOP

        FETCH cur1 INTO vname, vcomm;

        IF cur1%ROWCOUNT > 2 THEN exit; END IF;

        IF cur1%NOTFOUND THEN exit; END IF;

        DBMS\_OUTPUT.PUT\_LINE('Name = '||vname||', Commission = '||vcomm);

    END LOOP;

END;

В цикле необходимы два условия. Первое условие совершает выход при достижении третьего результата (записи). Второе условие совершает выход, если курсор ничего не прочёл. Второе условие гарантирует правильную работу курсора, потому что первое условие не гарантирует выход из цикла, а лишь проверяет счётчик.

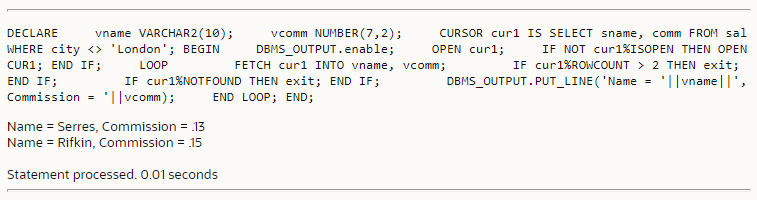


Рис. 1 – Вывод первых двух продавцов с комиссионными, работающих не в Лондоне

1. Напишите сценарий, в котором при помощи курсора выбираются все заказы после 4-го числа, и эта дата задается в качестве параметра. Параметр курсора определите типа даты, а не строки.

DECLARE

    CURSOR c\_ord(p\_Date DATE) IS SELECT onum, amt, odate FROM ord WHERE odate >= p\_Date;

BEGIN

    DBMS\_OUTPUT.enable;

    FOR v\_ord IN c\_ord('01/04/2010') LOOP

        DBMS\_OUTPUT.PUT\_LINE('Order number = '||v\_ord.onum||', amount = '||v\_ord.amt||', date = '||v\_ord.odate);

    END LOOP;

END;

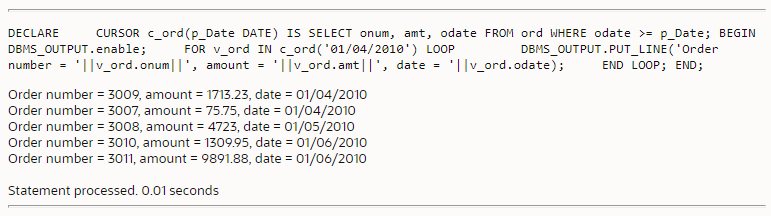


Рис. 2 – Вывод всех заказов от 4 января и позднее (дата в формате mm/dd/yy).

1. Выполните приведенный выше запрос с функцией DECODE. Напишите и выполните запрос для вывода в виде матрицы суммарных стоимостей заказов за каждое число для каждого продавца (для каждого числа – отдельный столбец, для каждого продавца - отдельная строка).

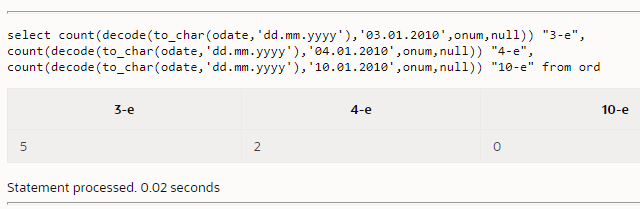


Рис. 3 – Выполненный запрос

SELECT

    DISTINCT ord.snum,

    sal.sname,

    SUM(DECODE(ord.odate, '01/03/2010', ord.amt, 0)) "3rd Jan",

    SUM(DECODE(ord.odate, '01/04/2010', ord.amt, 0)) "4rd Jan",

    SUM(DECODE(ord.odate, '01/05/2010', ord.amt, 0)) "5rd Jan",

    SUM(DECODE(ord.odate, '01/06/2010', ord.amt, 0)) "6rd Jan"

FROM ord, sal WHERE ord.snum = sal.snum GROUP BY ord.snum, sal.sname ORDER BY ord.snum;

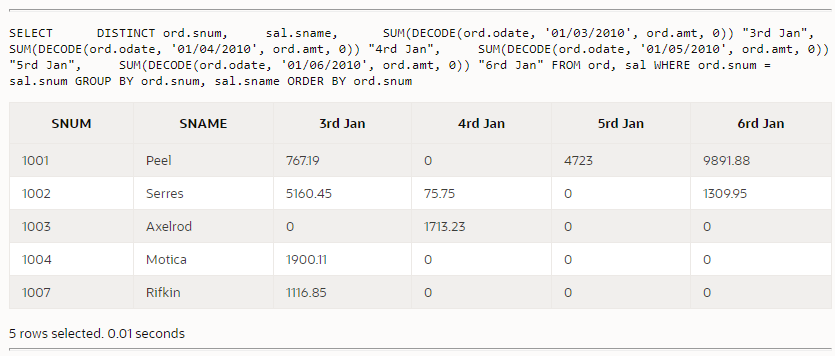


Рис. 4 – Суммирование стоимостей заказов по датам и продавцам.

1. Используя SQL Developer, создайте свою последовательность и свою таблицу. Вставьте в таблицу три строки, причем в командах вставки для занесения значений в одно из числовых полей таблицы используйте созданную последовательность.

DROP SEQUENCE myseq;

DROP TABLE test\_table;

CREATE SEQUENCE myseq;

create table test\_table(

      num number(4),

      name varchar2(10) NOT NULL

);

INSERT INTO test\_table

    VALUES (myseq.NEXTVAL, 'first rec');

INSERT INTO test\_table

    VALUES (myseq.NEXTVAL, 'second rec');

INSERT INTO test\_table

    VALUES (myseq.NEXTVAL, 'third rec');

COMMIT;

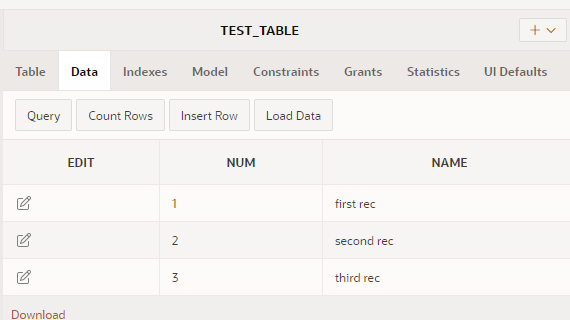


Рис. 5 – Создание новой таблицы.

1. Составьте сценарий, в котором создайте новую таблицу для занесения имен продавцов и минимальных стоимостей их заказов. В сценарии измените параметры последовательности из Задания 4 так, чтобы она формировала четные числа, начиная с 5000. Используя курсор, заполните созданную таблицу, причем, при вставке строк для формирования уникальных значений идентификаторов примените свою последовательность. Сценарий должен заканчиваться выводом данных из заполненной таблицы. Выполните сценарий.

DROP SEQUENCE myseq;

DROP TABLE lab2\_task5\_table;

CREATE SEQUENCE myseq

  MINVALUE 5000

  MAXVALUE 99999999999

  START WITH 500

  INCREMENT BY 2

  CACHE 20;

create table lab2\_task5\_table(

      num number(6),

      selname varchar2(10) NOT NULL,

      min\_amt number(7,3)

);

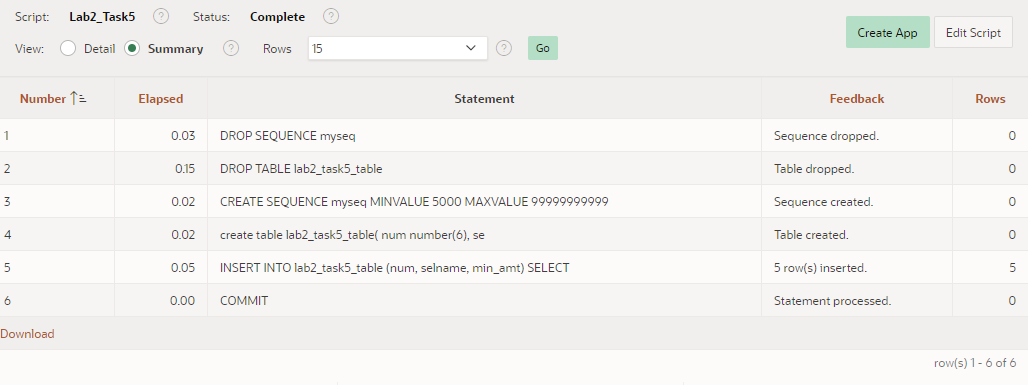
INSERT INTO lab2\_task5\_table (num, selname, min\_amt)

SELECT myseq.NEXTVAL, xxx, zzz

FROM (SELECT sal.sname xxx, MIN(ord.amt) zzz FROM ord, sal WHERE ord.snum = sal.snum GROUP BY ord.snum, sal.sname);

COMMIT;

Для данного задания была создана последовательность, начинающаяся с 5000. Инкремент равный 2 обеспечивает чётность. Возможность задания последовательности при группировке (GROUP BY) недоступна, поэтому сначала происходит задание последовательности, а только потом таблица заполняется данными из других таблиц.

  
Рис. 6 – Итог работы сценария.

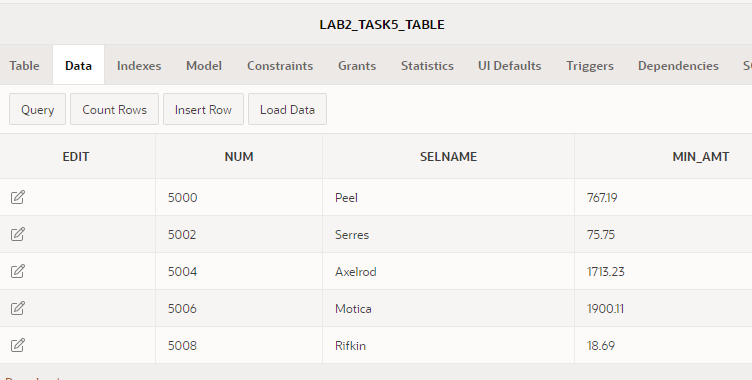


Рис. 7 – Вид таблицы.