

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»

Лабораторная работа №3

Выполнили: студенты IV курса
ИВТ, гр. ИП-911
Давыдов И.П.
Шумихин Д.В.

Проверила: Дьячкова И.С.

Новосибирск, 2023 г.

1. Создайте процедуру, которая изменяет указанный номер отдела в таблице DEP (создайте сами) на другой. Текущий номер отдела и новый номер должны задаваться как входные параметры процедуры.

```
DROP TABLE dep;

CREATE TABLE dep(
    dnum number(3),
    dname varchar2(20) NOT NULL
);

INSERT INTO dep
VALUES (10, 'Moscow');
INSERT INTO dep
VALUES (20, 'Novosibirsk');
INSERT INTO dep
VALUES (30, 'Kirov');
INSERT INTO dep
VALUES (40, 'Saint Petersburg');
INSERT INTO dep
VALUES (50, 'Omsk');
INSERT INTO dep
VALUES (60, 'Tomsk');
INSERT INTO dep
VALUES (70, 'Kazan');

COMMIT;

CREATE OR REPLACE PROCEDURE change_dep_num
(dep_num_old IN dep.dnum%TYPE,
 dep_num_new IN dep.dnum%TYPE)
AS
BEGIN
    UPDATE dep SET dnum = dep_num_new WHERE dnum = dep_num_old;
END change_dep_num;
/

BEGIN
    change_dep_num(20, 54);
END;
```

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.13	DROP TABLE dep	Table dropped.	0
2	0.05	CREATE TABLE dep(dnum number(3), dname varchar2(20)	Table created.	0
3	0.04	INSERT INTO dep VALUES (10, 'Moscow')	1 row(s) inserted.	1
4	0.00	INSERT INTO dep VALUES (20, 'Novosibirsk')	1 row(s) inserted.	1
5	0.00	INSERT INTO dep VALUES (30, 'Kirov')	1 row(s) inserted.	1
6	0.00	INSERT INTO dep VALUES (40, 'Saint Petersburg')	1 row(s) inserted.	1
7	0.00	INSERT INTO dep VALUES (50, 'Omsk')	1 row(s) inserted.	1
8	0.00	INSERT INTO dep VALUES (60, 'Tomsk')	1 row(s) inserted.	1
9	0.00	INSERT INTO dep VALUES (70, 'Kazan')	1 row(s) inserted.	1
10	0.00	COMMIT	Statement processed.	0
11	0.05	CREATE OR REPLACE PROCEDURE change_dep_num (dep_num_old	Procedure created.	0
12	0.01	BEGIN change_dep_num(20, 54); END;	Statement processed.	1

Рис.1.1 – Всё успешно отработало








DEP		
Query	Count Rows	Insert Row
	Load Data	
EDIT	DNUM	DNAME
	10	Moscow
	<u>54</u>	Novosibirsk
	30	Kirov
	40	Saint Petersburg
	50	Omsk
	60	Tomsk
	70	Kazan

Рис.1.2 – Заменили номер отдела 20 на 54

- Разработайте функцию, которая принимает в качестве параметра значение поля АМТ заказа. Функция должна возвращать строку в формате NNN рублей NN копеек. Слова «рубли» и «копейки» должны склоняться по правилам русского языка. Используйте функцию в команде SELECT следующим образом:

SELECT onum, имя_функции(AMT) FROM ord;

```
CREATE OR REPLACE FUNCTION show_amount
(AMT IN number) RETURN VARCHAR2 IS result VARCHAR2(50);
kop int;
rub int;
last_digit int;
last_digit_kop int;
BEGIN
kop := TO_NUMBER(REPLACE(TO_CHAR(mod(AMT, 1), '0.99'), '.'));
rub := floor(AMT);
last_digit := mod(rub, 10);
last_digit_kop := mod(kop, 10);
IF (last_digit > 4 OR last_digit = 0) then
result := TO_CHAR(rub, '9999') || ' Рублей ';

ELSIF (last_digit = 1) then
result := TO_CHAR(rub, '9999') || ' Рубль ';

ELSIF (last_digit < 5 and last_digit > 1) then
result := TO_CHAR(rub, '9999') || ' Рубля ';
END IF;
IF (last_digit_kop > 4 OR last_digit_kop = 0) then
result := result|| TO_CHAR(kop, '99') || ' копеек ';

ELSIF (last_digit_kop = 1) then
result := result|| TO_CHAR(kop, '99') || ' копейка ';

ELSIF (last_digit_kop < 5 AND last_digit_kop > 1) then
result := result|| kop || ' копейки ';
END IF;
result := result || ' ';
RETURN result;
```

```
END;
/
```

```
SELECT onum, show_amount(AMT) FROM ord;
```

SELECT onum, show_amount(AMT) FROM ord	
ONUM	SHOW_AMOUNT(AMT)
3001	18 Рублей 69 копеек
3003	767 Рублей 19 копеек
3002	1900 Рублей
3005	5160 Рублей 45 копеек
3006	1098 Рублей 16 копеек
3009	1713 Рубля 23 копейки
3007	75 Рублей 75 копеек
3008	4723 Рубля 0 копеек
3010	1309 Рублей 95 копеек
3011	9891 Рубль 88 копеек

Рис.2 – Вернули строку в формате NNN рублей NN копеек.

3. Создайте свой пакет по аналогии с Pack_1. В теле пакета в каждой процедуре определите курсор с параметром для указания города. Аналогичный параметр задайте для обеих процедур. В процедурах вместо команды SELECT используйте курсор для получения количества продавцов (покупателей) в городе, указанном в параметре. Название города укажите при вызове процедуры.

```
CREATE OR REPLACE PACKAGE Pack_1 IS
    PROCEDURE Sal_city_cnt (cityname IN varchar2);
    PROCEDURE Cust_city_cnt (cityname IN varchar2);
END Pack_1;

/

CREATE OR REPLACE PACKAGE BODY Pack_1 IS
    cnt NUMBER(3);
    PROCEDURE Sal_city_cnt (cityname IN varchar2) IS
        CURSOR cursor1 IS SELECT COUNT(sname) FROM sal WHERE city=cityname;
    BEGIN
        OPEN cursor1;
        FETCH cursor1 INTO cnt;
        DBMS_OUTPUT.PUT_LINE('Number of sellers in '||cityname||' = '||cnt);
    END Sal_city_cnt;

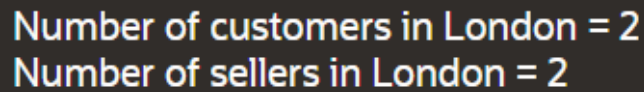
    PROCEDURE Cust_city_cnt (cityname IN varchar2) IS
        CURSOR cursor2 IS SELECT COUNT(cname) FROM cust WHERE city=cityname;
    BEGIN
        OPEN cursor2;
        FETCH cursor2 INTO cnt;
        DBMS_OUTPUT.PUT_LINE('Number of customers in '||cityname||' = '||cnt);
    END Cust_city_cnt;
END Pack_1;

/
```

```

BEGIN
    Pack_1.Cust_city_cnt('London');
    Pack_1.Sal_city_cnt('London');
END;

```



```

Number of customers in London = 2
Number of sellers in London = 2

```

Рис. 3 – Вывод количество продавцов и покупателей в указанном городе

4. Создайте два пакета, один из которых является библиотечным и содержит две функции. Во втором пакете выполняются вызовы этих функций. Библиотечные функции должны иметь параметры для задания фамилий членов бригады, причем первая функция должна возвращать самую длинную из фамилий, а вторая - самую короткую. Для определения длины строки используйте стандартную функцию LENGTH().

```

CREATE OR REPLACE PACKAGE Pack_1 IS
    PROCEDURE thelongestsurname (sn1 IN varchar2, sn2 IN varchar2, sn3 IN varchar2);
    PROCEDURE theshortestsurname (sn1 IN varchar2, sn2 IN varchar2, sn3 IN
varchar2);
END Pack_1;

/
CREATE OR REPLACE PACKAGE BODY Pack_1 IS

    PROCEDURE thelongestsurname (sn1 IN varchar2, sn2 IN varchar2, sn3 IN varchar2)
IS
    BEGIN
        IF LENGTH(sn1) >= LENGTH(sn2) THEN
            IF LENGTH(sn1) >= LENGTH(sn3) THEN
                DBMS_OUTPUT.PUT_LINE('The longest surname = '||sn1);
            ELSE
                DBMS_OUTPUT.PUT_LINE('The longest surname = '||sn3);
            END IF;
        ELSE
            IF LENGTH(sn2) >= LENGTH(sn3) THEN
                DBMS_OUTPUT.PUT_LINE('The longest surname = '||sn2);
            ELSE
                DBMS_OUTPUT.PUT_LINE('The longest surname = '||sn3);
            END IF;
        END IF;
    END thelongestsurname;

    PROCEDURE theshortestsurname (sn1 IN varchar2, sn2 IN varchar2, sn3 IN varchar2)
IS
    BEGIN
        IF LENGTH(sn1) <= LENGTH(sn2) THEN
            IF LENGTH(sn1) <= LENGTH(sn3) THEN
                DBMS_OUTPUT.PUT_LINE('The shortest surname = '||sn1);
            ELSE
                DBMS_OUTPUT.PUT_LINE('The shortest surname = '||sn3);
            END IF;
        ELSE

```

```

        IF LENGTH(sn2) <= LENGTH(sn3) THEN
            DBMS_OUTPUT.PUT_LINE('The shortest surname = '||sn2);
        ELSE
            DBMS_OUTPUT.PUT_LINE('The shortest surname = '||sn3);
        END IF;
    END IF;
END theshortestsurname;
END Pack_1;
/

CREATE OR REPLACE PACKAGE Pack_2 IS
    PROCEDURE xyz(sn1 IN varchar2, sn2 IN varchar2, sn3 IN varchar2);
END Pack_2;

/
CREATE OR REPLACE PACKAGE BODY Pack_2 IS
    PROCEDURE xyz (sn1 IN varchar2, sn2 IN varchar2, sn3 IN varchar2) IS
    BEGIN
        Pack_1.thelongestsurname (sn1, sn2, sn3);
        Pack_1.theshortestsurname (sn1, sn2, sn3);
    END xyz;
END Pack_2;
/

BEGIN
    Pack_2.xyz('Shumihin', 'Davydov', 'Lee');
END;

```

```

BEGIN      Pack_2.xyz('Shumihin', 'Davydov', 'Lee'); END;

The longest surname = Shumihin
The shortest surname = Lee

```

Рис. 4 – Вывод самой длинной и короткой фамилии

Дополнительные задания

br13

1. Напишите сценарий, в котором создается таблица Нагрузка. В таблице должны быть столбцы: идентификатор строки, номер курса, вид контроля (экзамен или зачет) и число сданных дисциплин. Таблицу заполняйте при помощи команд Insert. Для формирования значений идентификаторов строк используйте последовательность.

```
DROP SEQUENCE myseq;
DROP TABLE workload;

CREATE SEQUENCE myseq;
CREATE TABLE workload(
    num number(4),
    kurs_num number(2),
    type_control varchar2(30),
    number_disciplines number(2)
);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 1, 'зачёт', 3);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 2, 'зачёт', 10);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 1, 'экзамен', 4);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 2, 'экзамен', 3);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 3, 'зачёт', 5);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 4, 'экзамен', 6);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 4, 'зачёт', 2);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 3, 'экзамен', 8);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 2, 'экзамен', 6);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 3, 'зачёт', 5);

INSERT INTO workload
VALUES (myseq.NEXTVAL, 1, 'экзамен', 4);

COMMIT;

/
```

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.01	DROP SEQUENCE myseq	Sequence dropped.	0
2	0.13	DROP TABLE workload	Table dropped.	0
3	0.04	CREATE SEQUENCE myseq	Sequence created.	0
4	0.02	CREATE TABLE workload(num number(4), kurs_num numbe	Table created.	0
5	0.02	INSERT INTO workload VALUES (myseq.NEXTVAL, 1, 'зачёт',	1 row(s) inserted.	1
6	0.00	INSERT INTO workload VALUES (myseq.NEXTVAL, 2, 'зачёт',	1 row(s) inserted.	1
7	0.00	INSERT INTO workload VALUES (myseq.NEXTVAL, 1, 'экзамен'	1 row(s) inserted.	1
8	0.01	INSERT INTO workload VALUES (myseq.NEXTVAL, 2, 'экзамен'	1 row(s) inserted.	1
9	0.00	INSERT INTO workload VALUES (myseq.NEXTVAL, 3, 'зачёт',	1 row(s) inserted.	1
10	0.00	INSERT INTO workload VALUES (myseq.NEXTVAL, 4, 'экзамен'	1 row(s) inserted.	1
11	0.01	INSERT INTO workload VALUES (myseq.NEXTVAL, 4, 'зачёт',	1 row(s) inserted.	1
12	0.00	INSERT INTO workload VALUES (myseq.NEXTVAL, 3, 'экзамен'	1 row(s) inserted.	1
13	0.00	INSERT INTO workload VALUES (myseq.NEXTVAL, 2, 'экзамен'	1 row(s) inserted.	1
14	0.00	INSERT INTO workload VALUES (myseq.NEXTVAL, 3, 'зачёт',	1 row(s) inserted.	1
15	0.01	INSERT INTO workload VALUES (myseq.NEXTVAL, 1, 'экзамен'	1 row(s) inserted.	1
16	0.00	COMMIT	Statement processed.	0

Рис.1.1 – Всё успешно отработало












WORKLOAD												
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
Query		Count Rows		Insert Row		Load Data						
EDIT		NUM		KURS_NUM			TYPE_CONTROL				NUMBER_DISCIPLINES	
		1		1			ЗАЧЕТ				3	
		2		2			ЗАЧЕТ				10	
		3		1			ЭКЗАМЕН				4	
		4		2			ЭКЗАМЕН				3	
		5		3			ЗАЧЕТ				5	
		6		4			ЭКЗАМЕН				6	
		7		4			ЗАЧЕТ				2	
		8		3			ЭКЗАМЕН				8	
		9		2			ЭКЗАМЕН				6	
		10		3			ЗАЧЕТ				5	
		11		1			ЭКЗАМЕН				4	

Рис. 1.2 – Созданная и успешно заполненная таблица WORKLOAD

2. Создайте в базе таблицу-матрицу.
Создайте процедуру, которая заполняет эту таблицу-матрицу,
используя данные из таблицы Нагрузка. В строках таблицы должны
находиться виды контроля, в столбцах - номера курсов, на
пересечении строк и столбцов - число сданных предметов.
Создайте пакет и поместите в него процедуру.

```
create or replace PACKAGE Pack_dop2 IS
    PROCEDURE fill_table;
END Pack_dop2;

/

create or replace PACKAGE BODY Pack_dop2 IS
    tc varchar2(30);
    a1 number(2);
    a2 number(2);
    a3 number(2);
    a4 number(2);
    CURSOR curs1 IS
    SELECT
        type_control,
        SUM(decode(kurs_num, 1, number_disciplines, null)),
        SUM(decode(kurs_num, 2, number_disciplines, null)),
        SUM(decode(kurs_num, 3, number_disciplines, null)),
        SUM(decode(kurs_num, 4, number_disciplines, null))
    FROM workload GROUP BY type_control;
    PROCEDURE fill_table IS
    BEGIN
        DBMS_OUTPUT.enable;
        OPEN curs1;
        WHILE curs1%ROWCOUNT < 2 LOOP
            FETCH curs1 INTO tc, a1, a2, a3, a4;
            INSERT INTO dop32 VALUES (tc, a1, a2, a3, a4);
            COMMIT;
        END LOOP;
    END fill_table;
END Pack_dop2;

/

drop table dop32;
create table dop32(
    type_controls varchar2(30),
    first_kurs number(2),
    second_kurs number(2),
    third_kurs number(2),
    four_kurs number(2)
);

BEGIN
    Pack_dop2.fill_table;
END;
```

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.04	create or replace PACKAGE Pack_dop2 IS PROCEDURE fill_1	Package created.	0
2	0.08	create or replace PACKAGE BODY Pack_dop2 IS tc varchar2(Package Body created.	0
3	0.10	drop table dop32	Table dropped.	0
4	0.05	create table dop32(type_controls varchar2(30), firs	Table created.	0
5	0.16	BEGIN Pack_dop2.fill_table; END;	Statement processed.	1

Рис.2.1 – Всё успешно отработало



DOP32					
Table	Data	Indexes	Model	Constraints	Grants
Statistics	UI Defaults	Triggers	Dependencies	SQL	REST
Sample Queries	Query	Count Rows	Insert Row	Load Data	
EDIT	TYPE_CONTROLS	FIRST_KURS	SECOND_KURS	THIRD_KURS	FOUR_KURS
	зачёт	3	10	10	2
	ЭКЗАМЕН	8	9	8	6

Рис. 2.2 – Созданная и успешно заполненная таблица-матрица dop32