

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**
КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Институт цифры

ОТЧЕТ
О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ 6

«Базовые понятия PL/SQL. Циклы.»

студента 1 курса, группы ПИМ-231

Давыдова Игоря Павловича

Направление 09.04.03 – «Прикладная информатика»

Руководитель:
К.т.н, доц. Завозкин С. Ю.

Работа защищена
« _____ »
“ _____ ” _____ 2023г.

Кемерово 2023

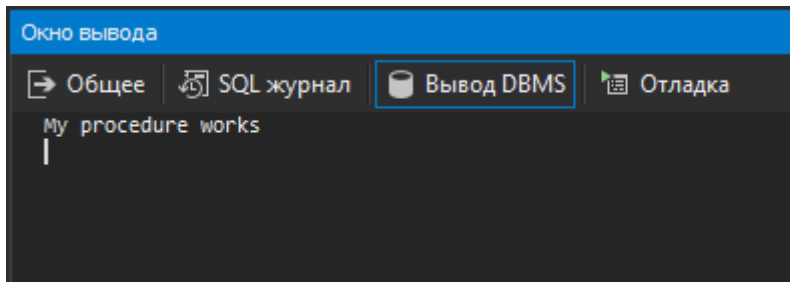
Задание

1. Создайте процедуру MY_PROCEDURE для вывода на экран фразы "My procedure works".
2. Создайте процедуру для включения нового отдела в таблицу S_DEPT. Входные данные – информация про новый отдел.
3. Создайте процедуру, обновляющую номер региона для заданного отдела. Входные данные – номер отдела.
4. Создайте процедуру для удаления отдела по заданному номеру.
5. Создайте процедуру, принимающую два числа через переменные. Первое нужно разделить на второе и к результату прибавить второе число. Результат должен быть записан в переменную PL/SQL и выведен на экран.
6. Создайте функцию, вычисляющую общее вознаграждение за год. Функции должны передаваться годовая зарплата и процент премиальных., Премии необходимо преобразовать из целого числа в десятичное. Если зарплата не определена, то функция должна выдавать значение «ноль». Если премия не определена, функция должна выдавать только зарплату.
7. (2 балла) Создайте процедуру SET_COMM, которая устанавливает процент комиссионных для служащего в зависимости от общего объема продаж:
 - Для подготовки к упражнению отмените действие ограничения для столбца COMMISSION_PCT в таблице S_EMP (если оно есть). *ALTER TABLE s_emp DROP CONSTRAINT s_emp_comission_pct_ck;*
 - Создайте параметр, чтобы можно было ввести номер служащего.
 - Подсчитайте сумму всех заказов, размещенных этим служащим.
 - Если эта сумма меньше 10000 установите процент комиссионных «10».
 - Если сумма находится в диапазоне от 10000 до 1000000, установите процент комиссионных «15».
 - Если в таблице S_ORD нет заказов, размещенных данным служащим, установите процент комиссионных «0».
8. (2 балла) Создайте процедуру CUST_UPDATE с циклом для обработки кредитного рейтинга всех заказчиков по регионам. Не фиксируйте изменения:
 - Если номер региона четный, установите кредитный рейтинг "Excellent", даже если он уже был таким; в противном случае установите кредитный рейтинг "Good".
 - Когда строки будут обновлены, определите количество обновленных строк.

Ход работы

1. Создайте процедуру MY_PROCEDURE для вывода на экран фразы "My procedure works".

```
CREATE OR REPLACE PROCEDURE MY_PROCEDURE AS  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('My procedure works');  
END;  
  
/  
  
BEGIN  
    MY_PROCEDURE;  
END;  
  
/
```



2. Создайте процедуру для включения нового отдела в таблицу S_DEPT. Входные данные – информация про новый отдел.

```
CREATE OR REPLACE PROCEDURE INSERT_DEPARTMENT(  
    p_id IN NUMBER,  
    p_name IN VARCHAR2,  
    p_region_id IN NUMBER  
) AS  
BEGIN  
    INSERT INTO S_DEPT (ID, NAME, REGION_ID)  
    VALUES (p_id, p_name, p_region_id);  
    COMMIT;  
END;  
  
/  
  
BEGIN
```

```
INSERT_DEPARTMENT(70, 'New', 3);
```

```
END;
```

```
----- Выполнение начато: Соединение: SQL.sql -----  
DML выполнен успешно [0,015с]  
Запрос открыт за 0,049с [0,049с выполнение, < 0,001с выборка]  
0 строк выбрано  
  
----- Готово: SQL.sql -----  
SQL1.sql: Запрос открыт за 0,028с [0,014с выполнение, 0,014с выборка]
```

ID NUMBER(7)	NAME VARCHAR2(25)	REGION_ID NUMBER(7)
10	Finance	1
31	Sales	1
32	Sales	2
33	Sales	3
34	Sales	4
35	Sales	5
41	Operations	1
42	Operations	2
43	Operations	3
44	Operations	4
45	Operations	5
50	Administration	1
11	Finance	2
61	Accounting	2
54	Education	1
70	New	3

3. Создайте процедуру, обновляющую номер региона для заданного отдела.
Входные данные – номер отдела.

```
CREATE OR REPLACE PROCEDURE  
UPDATE_REGION_FOR_DEPARTMENT(  
    p_department_id IN NUMBER,  
    p_new_region_id IN NUMBER  
) AS  
BEGIN  
    UPDATE S_DEPT  
    SET REGION_ID = p_new_region_id  
    WHERE ID = p_department_id;  
    COMMIT;  
END;
```

/

BEGIN

UPDATE_REGION_FOR_DEPARTMENT(10, 3);

END;

```
----- Выполнение начато: Соединение: SQL.sql -----
DML выполнен успешно [0,131с]
Запрос открыт за 0,145с [0,145с выполнение, < 0,001с выборка]
0 строк выбрано

----- Готово: SQL.sql -----
SQL1.sql: Запрос открыт за 0,028с [0,014с выполнение, 0,014с выборка]
```

ID NUMBER(7)	NAME VARCHAR2(25)	REGION_ID NUMBER(7)
10	Finance	3
31	Sales	1
32	Sales	2
33	Sales	3
34	Sales	4
35	Sales	5
41	Operations	1
42	Operations	2
43	Operations	3
44	Operations	4
45	Operations	5
50	Administration	1
11	Finance	2
61	Accounting	2
54	Education	1
70	New	3

4. Создайте процедуру для удаления отдела по заданному номеру.

CREATE OR REPLACE PROCEDURE DELETE_DEPARTMENT(

p_department_id IN NUMBER

) AS

BEGIN

DELETE FROM S_DEPT WHERE ID = p_department_id;

COMMIT;

END;

/

BEGIN

DELETE_DEPARTMENT(70);

END;

```
----- Выполнение начато: Соединение: SQL.sql -----  
DML выполнен успешно [0,026с]  
Запрос открыт за 0,102с [0,102с выполнение, < 0,001с выборка]  
0 строк выбрано  
  
----- Готово: SQL.sql -----  
SQL1.sql: Запрос открыт за 0,028с [0,014с выполнение, 0,014с выборка]
```

ID NUMBER(7)	NAME VARCHAR2(25)	REGION_ID NUMBER(7)
10	Finance	3
31	Sales	1
32	Sales	2
33	Sales	3
34	Sales	4
35	Sales	5
41	Operations	1
42	Operations	2
43	Operations	3
44	Operations	4
45	Operations	5
50	Administration	1
11	Finance	2
61	Accounting	2
54	Education	1

5. Создайте процедуру, принимающую два числа через переменные. Первое нужно разделить на второе и к результату прибавить второе число. Результат должен быть записан в переменную PL/SQL и выведен на экран.

```
CREATE OR REPLACE PROCEDURE CALCULATION(
```

```
    p_operand1 IN NUMBER,
```

```
    p_operand2 IN NUMBER,
```

```
    p_result OUT NUMBER
```

```
) AS
```

```
BEGIN
```

```
    p_result := (p_operand1 / p_operand2) + p_operand2;
```

```
    DBMS_OUTPUT.PUT_LINE('Result: ' || p_result);
```

```
END;
```

```
/
```

```
DECLARE
```

```
    v_result NUMBER;
```

```

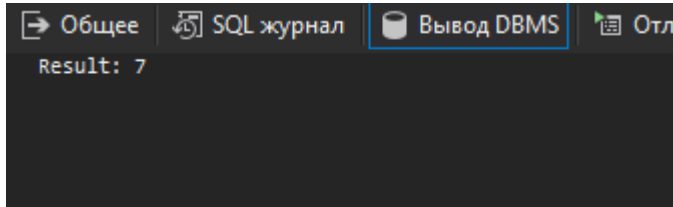
BEGIN

    CALCULATION(10, 2, v_result);

END;

/

```



6. Создайте функцию, вычисляющую общее вознаграждение за год. Функции должны передаваться годовая зарплата и процент премиальных., Премии необходимо преобразовать из целого числа в десятичное. Если зарплата не определена, то функция должна выдавать значение «ноль». Если премия не определена, функция должна выдавать только зарплату.

```

CREATE          OR          REPLACE          PROCEDURE
CALCULATE_COMPENSATION_S_EMP(
    p_emp_id IN NUMBER,
    p_total_compensation OUT NUMBER
) AS
    v_salary NUMBER;
    v_bonus_percent NUMBER;
BEGIN
    SELECT SALARY, COMMISSION_PCT
    INTO v_salary, v_bonus_percent
    FROM S_EMP
    WHERE ID = p_emp_id;
    IF v_salary IS NOT NULL THEN
        p_total_compensation := (v_salary * 12) + (v_salary * v_bonus_percent /
100);
    ELSE
        p_total_compensation := 0;
    END IF;
END;

```

```

/
DECLARE
    v_employee_id NUMBER := 11;
    v_total_compensation NUMBER;
BEGIN
    CALCULATE_COMPENSATION_S_EMP(v_employee_id,
v_total_compensation);
    DBMS_OUTPUT.PUT_LINE('Общее вознаграждение для сотрудника ' ||
v_employee_id || ': ' || v_total_compensation);
END;
/

```

Я подумал, что необходимо рассчитать общее вознаграждение для какого-то сотрудника именно из таблицы S_EMP и соответственно записал в переменные для зарплаты и процента премии результаты SQL-запроса.

Выбрал сотрудника с номером 11:

Общее вознаграждение для сотрудника 11: 16940

8	Biri	Ben	bbiri	07.04. 1990 00:00:00	(null)	2	Warehouse Manager	43	1100	(null)	
9	Catchpole	Antoinette	acatchpo	09.02. 1992 00:00:00	(null)	2	Warehouse Manager	44	1300	0	
10	Havel	Marta	mhavel	27.02. 1991 00:00:00	(null)	2	Warehouse Manager	45	1307	(null)	
11	Magee	Colin	cmagee	14.05. 1990 00:00:00	(null)	3	Sales Representative	31	1400	10	*****
13	Sedeghi	Yasmin	ysedeghi	18.02. 1991 00:00:00	(null)	3	Sales Representative	33	1515	10	*****
14	Nguyen	Mai	mnguyen	22.01. 1992 00:00:00	(null)	3	Sales Representative	34	1525	10	*****

И еще один вариант выполнения этого задания:

```

CREATE OR REPLACE FUNCTION
CALCULATE_TOTAL_COMPENSATION(
    p_salary IN NUMBER,
    p_bonus_percent IN NUMBER
) RETURN NUMBER IS
    v_bonus DECIMAL(6, 2);
BEGIN
    IF p_salary IS NULL THEN
        RETURN 0;
    ELSIF p_bonus_percent IS NULL THEN
        RETURN p_salary;

```

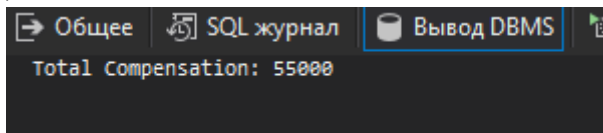


```

ELSE
    v_bonus := p_salary * (p_bonus_percent / 100);
    RETURN p_salary + v_bonus;
END IF;
END;
/

DECLARE
    v_salary NUMBER := 50000; -- Зарплата
    v_bonus_percent NUMBER := 10; -- Процент премиальных
    v_total_compensation NUMBER;
BEGIN
    v_total_compensation :=
    CALCULATE_TOTAL_COMPENSATION(v_salary, v_bonus_percent);
    DBMS_OUTPUT.PUT_LINE('Total Compensation: ' ||
    v_total_compensation);
END;
/

```



7. (2 балла) Создайте процедуру SET_COMM, которая устанавливает процент комиссионных для служащего в зависимости от общего объема продаж:

- Для подготовки к упражнению отмените действие ограничения для столбца COMMISSION_PCT в таблице S_EMP (если оно есть). *ALTER TABLE s_emp DROP CONSTRAINT s_emp_comission_pct_ck;*
- Создайте параметр, чтобы можно было ввести номер служащего.
- Подсчитайте сумму всех заказов, размещенным этим служащим.
- Если эта сумма меньше 10000 установите процент комиссионных «10».
- Если сумма находится в диапазоне от 10000 до 1000000, установите процент комиссионных «15».
- Если в таблице S_ORD нет заказов, размещенных данным служащим, установите процент комиссионных «0».

```

CREATE OR REPLACE PROCEDURE SET_COMM(
    p_employee_id IN NUMBER
) AS
    v_total_sales NUMBER;
BEGIN
    SELECT SUM(TOTAL) INTO v_total_sales
    FROM S_ORD
    WHERE SALES_REP_ID = p_employee_id;

    IF v_total_sales < 10000 THEN
        UPDATE S_EMP
        SET COMMISSION_PCT = 10
        WHERE ID = p_employee_id;
    ELSIF v_total_sales BETWEEN 10000 AND 1000000 THEN
        UPDATE S_EMP
        SET COMMISSION_PCT = 15
        WHERE ID = p_employee_id;
    ELSE
        UPDATE S_EMP
        SET COMMISSION_PCT = 0
        WHERE ID = p_employee_id;
    END IF;

    COMMIT;
END;
/
BEGIN
    SET_COMM(16);
END;

```

```

----- Выполнение начато: Соединение: SQL.sql -----
DML выполнен успешно [0,015с]
Запрос открыт за 0,047с [0,047с выполнение, < 0,001с выборка]
0 строк выбрано

----- Готово: SQL.sql -----
SQL1.sql: Запрос открыт за 0,050с [0,035с выполнение, 0,015с выборка]

```

11	Magee	Colin	cmagee	14.05. 1990 00:00:00	(null)	3	Sales Representative	31	1400	0	*****
13	Sedeghi	Yasmin	ysedeghi	18.02. 1991 00:00:00	(null)	3	Sales Representative	33	1515	10	*****
14	Nguyen	Mai	minguyen	22.01. 1992 00:00:00	(null)	3	Sales Representative	34	1525	10	*****
16	Maduro	Elena	emaduro	07.02. 1992 00:00:00	(null)	6	Stock Clerk	41	1400	0	
17	Smith	George	gsmith	08.03. 1990 00:00:00	(null)	6	Stock Clerk	34	940	(null)	

8. (2 балла) Создайте процедуру CUST_UPDATE с циклом для обработки кредитного рейтинга всех заказчиков по регионам. Не фиксируйте изменения:

- Если номер региона четный, установите кредитный рейтинг “Excellent”, даже если он уже был таким; в противном случае установите кредитный рейтинг “Good”.
- Когда строки будут обновлены, определите количество обновленных строк.

```
CREATE OR REPLACE PROCEDURE CUST_UPDATE AS
```

```
  v_credit_rating VARCHAR2(20);
```

```
BEGIN
```

```
  BEGIN
```

```
    FOR region_rec IN (SELECT DISTINCT REGION_ID FROM
S_CUSTOMER) LOOP
```

```
      -- Определение кредитного рейтинга в зависимости от номера региона
```

```
      BEGIN
```

```
        IF MOD(region_rec.REGION_ID, 2) = 0 THEN
```

```
          v_credit_rating := 'Excellent';
```

```
        ELSE
```

```
          v_credit_rating := 'Good';
```

```
        END IF;
```

```
      END;
```

```
DECLARE
```

```
  v_updated_rows NUMBER;
```

```

BEGIN

UPDATE S_CUSTOMER

SET CREDIT_RATING = v_credit_rating

WHERE REGION_ID = region_rec.REGION_ID;

-- Определение количества обновленных строк
v_updated_rows := SQL%ROWCOUNT;

DBMS_OUTPUT.PUT_LINE('Количество обновленных строк для
региона ' || region_rec.REGION_ID || ': ' || v_updated_rows);

END;

END LOOP;

EXCEPTION

WHEN OTHERS THEN

-- Ошибка, откатываем изменения

ROLLBACK;

DBMS_OUTPUT.PUT_LINE('Произошла ошибка: ' || SQLERRM);

END;

END;

/

BEGIN

CUST_UPDATE();

END;

```

```

Количество обновленных строк для региона 1: 4
Количество обновленных строк для региона 2: 2
Количество обновленных строк для региона 4: 3
Количество обновленных строк для региона 5: 4
Количество обновленных строк для региона 3: 2

```

Таблица S_CUSTOMER до использования процедуры:

ID NUMBER(7)	NAME VARCHAR2(50)	PHONE VARCHAR2(25)	COMPANY_ADDRESS VARCHAR2(400)	CITY VARCHAR2(30)	STATE VARCHAR2(20)	COUNTRY VARCHAR2(30)	ZIP_CODE VARCHAR2(75)	CREDIT_RATING VARCHAR2(9)	SALES_REP_ID NUMBER(7)	REGION_ID NUMBER(7)	COMMENT VARCHAR2(200)
201	Unisports	55-2066101	72 Via Bahia	Sao Paolo	(null)	Brazil	(null)	GOOD	11		2 Customer
202	Simms Athletics	8 1-20101	6741 Takashi Blvd.	Osaka	(null)	Japan	(null)	EXCELLENT		14	4 Customer
203	Delhi Sports	91-10351	11368 Chanakya	New Delhi	(null)	India	(null)	GOOD		14	4 Customer
204	Womansport	1-206-104-0103	281 King Street	Seattle	(null)	USA	98 101	EXCELLENT	11		1 (null)
205	Kams Sporting Goods	852-3692888	15 Henessey Road	Hong Kong	(null)	(null)	(null)	GOOD	21		4 (null)
206	Sportique	33-2257201	172 Rue de Rivoli	Cannes	(null)	France	(null)	EXCELLENT	11		5 Customer
207	Sweet Rock Sports	234-6036201	6 Saint Antoine	Lagos	(null)	Nigeria	(null)	GOOD	(null)		3 (null)
208	Muench Sports	49-527454	435 Gruenstrasse	Stuttgart	(null)	Germany	(null)	EXCELLENT		17	5 Customer
209	Beisbol Si!	809-352689	792 Playa Del Mar	San Pedro de Macoris	(null)	Dominican Republic	(null)	GOOD	11		1 (null)
210	Futbol Sonora	52-404562	3 Via Saguario	Nogales	(null)	Mexico	(null)	EXCELLENT	21		2 Customer
211	Kuhns Sports	42-111292	7 Modrany	Prague	(null)	Czechoslovakia	(null)	GOOD	(null)		5 (null)
212	Hamada Sport	20-1209211	57A Corniche	Alexandria	(null)	Egypt	(null)	EXCELLENT		13	3 Customer
213	Big Johns Sports Emporium	1-415-555-6281	4783 18th Street	San Francisco	(null)	USA	94117	GOOD	11		1 Customer
214	Ojbvay Retail	1-716-555-7171	415 Main Street	Buffalo	(null)	USA	14202	EXCELLENT	11		1 (null)
215	Sporta Russia	7-3892456	6000 Yekatomina	Saint Petersburg	(null)	Russia	(null)	GOOD	(null)		5 This custo

Таблица S_CUSTOMER после использования процедуры:

ID NUMBER(7)	NAME VARCHAR2(50)	PHONE VARCHAR2(25)	COMPANY_ADDRESS VARCHAR2(400)	CITY VARCHAR2(30)	STATE VARCHAR2(20)	COUNTRY VARCHAR2(30)	ZIP_CODE VARCHAR2(75)	CREDIT_RATING VARCHAR2(9)	SALES_REP_ID NUMBER(7)	REGION_ID NUMBER(7)	COMMENT VARCHAR2(200)
201	Unisports	55-2066101	72 Via Bahia	Sao Paolo	(null)	Brazil	(null)	Excellent	11		2 Custom
202	Simms Athletics	8 1-20101	6741 Takashi Blvd.	Osaka	(null)	Japan	(null)	Excellent		14	4 Custom
203	Delhi Sports	91-10351	11368 Chanakya	New Delhi	(null)	India	(null)	Excellent		14	4 Custom
204	Womansport	1-206-104-0103	281 King Street	Seattle	(null)	USA	98 101	Good	11		1 (null)
205	Kams Sporting Goods	852-3692888	15 Henessey Road	Hong Kong	(null)	(null)	(null)	Excellent	21		4 (null)
206	Sportique	33-2257201	172 Rue de Rivoli	Cannes	(null)	France	(null)	Good	11		5 Custom
207	Sweet Rock Sports	234-6036201	6 Saint Antoine	Lagos	(null)	Nigeria	(null)	Good	(null)		3 (null)
208	Muench Sports	49-527454	435 Gruenstrasse	Stuttgart	(null)	Germany	(null)	Good		17	5 Custom
209	Beisbol Si!	809-352689	792 Playa Del Mar	San Pedro de Macoris	(null)	Dominican Republic	(null)	Good	11		1 (null)
210	Futbol Sonora	52-404562	3 Via Saguario	Nogales	(null)	Mexico	(null)	Excellent	21		2 Custom
211	Kuhns Sports	42-111292	7 Modrany	Prague	(null)	Czechoslovakia	(null)	Good	(null)		5 (null)
212	Hamada Sport	20-1209211	57A Corniche	Alexandria	(null)	Egypt	(null)	Good		13	3 Custom
213	Big Johns Sports Emporium	1-415-555-6281	4783 18th Street	San Francisco	(null)	USA	94117	Good	11		1 Custom
214	Ojbvay Retail	1-716-555-7171	415 Main Street	Buffalo	(null)	USA	14202	Good	11		1 (null)
215	Sporta Russia	7-3892456	6000 Yekatomina	Saint Petersburg	(null)	Russia	(null)	Good	(null)		5 This cus

Заключение

В ходе данной лабораторной работы, я изучил и практиковал использование PL/SQL - мощного инструмента Oracle Database для создания хранимых процедур и функций. В процессе выполнения заданий, я освоил различные аспекты PL/SQL, включая создание процедур и функций, работу с параметрами, циклами и работу с данными в таблицах.