

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

ОТЧЕТ
О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ 7

«Курсоры.»

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. Используя курсор, создайте процедуру TOP_DOGS1 для определения самых высокооплачиваемых служащих:
 - Для этого упражнения создайте новую таблицу с данными о служащих и их заработной плате.
 - Включите параметр, чтобы пользователь мог ввести нужное количество самых высокооплачиваемых служащих (n).
 - Создайте цикл FOR с курсором для выборки из таблицы S_EMP фамилий и заработной платы n самых высокооплачиваемых служащих.
 - Сохраните фамилию и заработную плату в таблице TOP_DOGS.
 - Предполагается, что двух служащих с одинаковой заработной платой не существует.
 - Обработайте особые случаи: n=0 и n, превышающим количество служащих в таблице S_EMP.
 - После каждого теста удаляйте данные из таблицы TOP_DOGS.
2. Используя курсор с параметром, создайте процедуру для определения сотрудников, работающих больше заданного числа лет. Выведите этих сотрудников, регион, в котором они работают, количество сделанных ими заказов.
3. Создайте хранимую процедуру ADD_STARS, которая в новом столбце STARS проставляет для каждого служащего по призовой звёздочке за каждый процент заработанных комиссионных. Используйте курсор и цикл WHILE.
 - Для этого упражнения создайте в таблице S_EMP новый столбец для хранения звёздочек (*).
 - Определите процент комиссионных для каждого служащего, округлив его до ближайшего целого числа. Рассмотрите случай, когда служащий не получает комиссионных.
 - Добавляйте звёздочку в строку звёздочек за каждый процент комиссионных. Если, например, служащий получает 10 процентов комиссионных, символьная строка в столбце STARS должна содержать десять звёздочек.
 - Проставьте соответствующее количество звёздочек для каждого служащего в столбце STARS.
4. Напишите процедуру, выводящую без повторений список менеджеров с указанием рядом с каждым именем через запятую имён их подчиненных, работающих больше заданного числа лет.
5. Скопируйте процедуру TOP_DOGS1 из упражнения 1 и назовите новый вариант TOP_DOGS2. Измените процедуру TOP_DOGS2 с учётом случая, когда несколько служащих из упражнения 1 имеют одинаковую заработную плату.
 - Создайте новую таблицу с данными о служащих и их заработной плате. В таблице должен быть столбец COMMENTS.

- Если есть сотрудники, получающие такую же зарплату, как и n, то они тоже должны быть выведены в списке. Например, в качестве n выводится число 6, 7 или 8. В этом случае в выходных данных должны появиться фамилии Ngao, Dumas и Quick-To_See. Если же n равно 9, 10 или 11, должны появиться фамилии Nagayama, Magee и Maduro.
- Для каждой фамилии в столбце COMMENTS в списке должны быть перечислены все служащие с такой же заработной платой.
- Не забывайте полностью удалять данные из таблицы TOP_DOGS перед каждым запуском.

Ход работы

1. Используя курсор, создайте процедуру TOP_DOGS1 для определения самых высокооплачиваемых служащих:
 - Для этого упражнения создайте новую таблицу с данными о служащих и их заработной плате.
 - Включите параметр, чтобы пользователь мог ввести нужное количество самых высокооплачиваемых служащих (n).
 - Создайте цикл FOR с курсором для выборки из таблицы S_EMP фамилий и заработной платы n самых высокооплачиваемых служащих.
 - Сохраните фамилию и заработную плату в таблице TOP_DOGS.
 - Предполагается, что двух служащих с одинаковой заработной платой не существует.
 - Обработайте особые случаи: n=0 и n, превышающим количество служащих в таблице S_EMP.
 - После каждого теста удаляйте данные из таблицы TOP_DOGS.

```
CREATE OR REPLACE PROCEDURE TOP_DOGS1 (n IN NUMBER) AS
```

```
    CURSOR emp_cursor IS
```

```
        SELECT last_name, salary
```

```
        FROM S_EMP E
```

```
        WHERE salary IS NOT NULL
```

```
        ORDER BY salary DESC;
```

```
    v_count NUMBER;
```

```
BEGIN
```

```
    SELECT COUNT(*)
```

```
    INTO v_count
```

```
    FROM S_EMP;
```

```
    IF n = 0 THEN
```

```
        dbms_output.put_line('n = 0');
```

```
    ELSIF n > v_count THEN
```

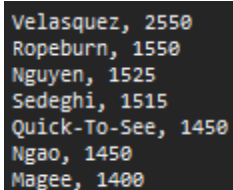
```
        dbms_output.put_line('N больше числа служащих');
```

```
    ELSE
```

```

FOR emp_record IN emp_cursor LOOP
    dbms_output.put_line(emp_record.last_name || ', ' || emp_record.salary);
    EXIT WHEN emp_cursor%ROWCOUNT >= n;
END LOOP;
END IF;
END;
/
BEGIN
    TOP_DOGS1(7);
END;

```



```

Velasquez, 2550
Ropeburn, 1550
Nguyen, 1525
Sedeghi, 1515
Quick-To-See, 1450
Ngao, 1450
Magee, 1400

```

- Используя курсор с параметром, создайте процедуру для определения сотрудников, работающих больше заданного числа лет. Выведите этих сотрудников, регион, в котором они работают, количество сделанных ими заказов.

```

CREATE OR REPLACE PROCEDURE GET_EMPLOYEES_BY_YEARS(
    p_experience_years IN NUMBER
) AS
    CURSOR employees_cursor(p_years NUMBER) IS
        SELECT      E.FIRST_NAME,      E.LAST_NAME,      D.NAME      AS
DEPARTMENT_NAME, R.NAME AS REGION_NAME, COUNT(O.ID) AS
ORDER_COUNT
        FROM S_EMP E
        JOIN S_DEPT D ON E.DEPT_ID = D.ID
        JOIN S_REGION R ON D.REGION_ID = R.ID
        LEFT JOIN S_ORD O ON E.ID = O.SALES_REP_ID
        WHERE MONTHS_BETWEEN(SYSDATE, E.START_DATE) / 12 >=

```

p_years

GROUP BY E.FIRST_NAME, E.LAST_NAME, D.NAME, R.NAME;

BEGIN

FOR employee_rec IN employees_cursor(p_experience_years) LOOP

DBMS_OUTPUT.PUT_LINE('Employee: ' || employee_rec.FIRST_NAME ||
' ' || employee_rec.LAST_NAME);

DBMS_OUTPUT.PUT_LINE('Department: ' || employee_rec.DEPARTMENT_NAME);

DBMS_OUTPUT.PUT_LINE('Region: ' || employee_rec.REGION_NAME);

DBMS_OUTPUT.PUT_LINE('Number of Orders: ' || employee_rec.ORDER_COUNT);

DBMS_OUTPUT.NEW_LINE;

END LOOP;

END;

/

BEGIN

GET_EMPLOYEES_BY_YEARS(32);

END;

```

Employee: Yasmin Sedeghi
Department: Sales
Region: Africa
Number of Orders: 1

Employee: Midori Nagayam
Department: Sales
Region: USA
Number of Orders: 0

Employee: Carmen Velasquez
Department: Sales
Region: S.America
Number of Orders: 0

Employee: LaDoris Ngao
Department: Education
Region: USA
Number of Orders: 0

Employee: Molly Uguhart
Department: Operations
Region: USA
Number of Orders: 0

Employee: Colin Magee
Department: Sales
Region: USA
Number of Orders: 6

Employee: George Smith
Department: Operations
Region: USA

```

3. Создайте хранимую процедуру ADD_STARS, которая в новом столбце STARS проставляет для каждого служащего по призовой звёздочке за каждый процент заработанных комиссионных. Используйте курсор и цикл WHILE.

- Для этого упражнения создайте в таблице S_EMP новый столбец для хранения звёздочек (*).
- Определите процент комиссионных для каждого служащего, округлив его до ближайшего целого числа. Рассмотрите случай, когда служащий не получает комиссионных.
- Добавляйте звёздочку в строку звёздочек за каждый процент комиссионных. Если, например, служащий получает 10 процентов комиссионных, символьная строка в столбце STARS должна содержать десять звёздочек.
- Проставьте соответствующее количество звёздочек для каждого служащего в столбце STARS.

CREATE OR REPLACE PROCEDURE ADD_STARS AS

CURSOR employees_cursor IS

SELECT ID, COMMISSION_PCT

FROM S_EMP;

```

v_stars VARCHAR2(2000);
v_percent NUMBER;
BEGIN
-- Новый столбец STARS
EXECUTE IMMEDIATE 'ALTER TABLE S_EMP ADD STARS
VARCHAR2(2000)';

-- Обход всех служащих
FOR employee_rec IN employees_cursor LOOP
    v_stars := NULL;

-- Проверка, есть ли коммиссионные
IF employee_rec.COMMISSION_PCT IS NOT NULL THEN
    -- Округление процента коммиссионных до ближайшего целого
    v_percent := ROUND(employee_rec.COMMISSION_PCT);

-- Добавление звездочек
WHILE v_percent > 0 LOOP
    v_stars := v_stars || '*';
    v_percent := v_percent - 1;
END LOOP;
END IF;

-- Обновление STARS для текущего служащего
EXECUTE IMMEDIATE 'UPDATE S_EMP SET STARS = :stars WHERE ID
= :employee_id'
    USING v_stars, employee_rec.ID;
END LOOP;
COMMIT;
END;

```


/

```
BEGIN  
  ADD_STARS();  
END;
```

	COMMISSION_PCT NUMBER(4, 2)	STARS VARCHAR2(2000)
50	0	(null)
00	0	(null)
50	0	(null)
50	0	(null)
00	0	(null)
50	0	(null)
00	0	(null)
00	0	(null)
07	0	(null)
00	20	*****
15	15	*****
25	10	*****
00	0	(null)
40	0	(null)

4. Напишите процедуру, выводящую без повторов список менеджеров с указанием рядом с каждым именем через запятую имён их подчиненных, работающих больше заданного числа лет.

```
CREATE OR REPLACE PROCEDURE MANAGERS_WITH_EMPLOYEES  
AS
```

```
  CURSOR manager_cursor IS
```

```
    SELECT  M.ID   AS  MANAGER_ID,  M.FIRST_NAME  ||  ' '  ||  
M.LAST_NAME AS MANAGER_NAME  
    FROM S_EMP M  
    WHERE M.MANAGER_ID IS NULL;
```

```
  v_manager_name VARCHAR2(100);
```

```
  v_veteran_employees VARCHAR2(1000);
```

```

BEGIN

FOR manager_rec IN manager_cursor LOOP

    v_veteran_employees := NULL; -- Обнуляем список подчиненных

    -- Ищем подчиненных данного менеджера
    FOR employee_rec IN (SELECT E.FIRST_NAME || ' ' || E.LAST_NAME
AS EMPLOYEE_NAME
                        FROM S_EMP E
                        WHERE E.MANAGER_ID = manager_rec.MANAGER_ID
                        AND MONTHS_BETWEEN(SYSDATE, E.START_DATE) /
12 >= 30) LOOP
        IF v_veteran_employees IS NULL THEN
            v_veteran_employees := employee_rec.EMPLOYEE_NAME;
        ELSE
            v_veteran_employees      :=      v_veteran_employees      ||      ' ' ||
employee_rec.EMPLOYEE_NAME;
        END IF;
    END LOOP;

    IF v_veteran_employees IS NOT NULL THEN
        v_manager_name := manager_rec.MANAGER_NAME;
        DBMS_OUTPUT.PUT_LINE('Manager: ' || v_manager_name);
        DBMS_OUTPUT.PUT_LINE('Veteran      Employees:      ' ||
v_veteran_employees);
        DBMS_OUTPUT.NEW_LINE;
    END IF;
END LOOP;

END;

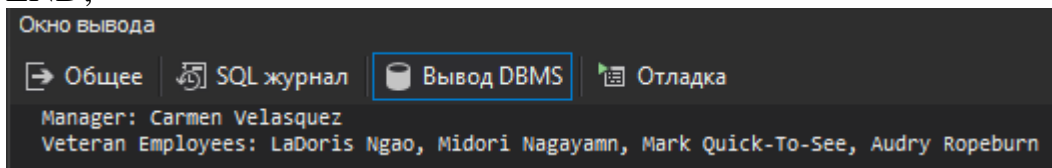
/

BEGIN

```

```
MANAGERS_WITH_EMPLOYEES();
```

```
END;
```



5. Скопируйте процедуру TOP_DOGS1 из упражнения 1 и назовите новый вариант TOP_DOGS2. Измените процедуру TOP_DOGS2 с учётом случая, когда несколько служащих из упражнения 1 имеют одинаковую заработную плату.
- Создайте новую таблицу с данными о служащих и их заработной плате. В таблице должен быть столбец COMMENTS.
 - Если есть сотрудники, получающие такую же зарплату, как и n, то они тоже должны быть выведены в списке. Например, в качестве n выводится число 6, 7 или 8. В этом случае в выходных данных должны появиться фамилии Ngao, Dumas и Quick-To-See. Если же n равно 9, 10 или 11, должны появиться фамилии Nagayama, Magee и Maduro.
 - Для каждой фамилии в столбце COMMENTS в списке должны быть перечислены все служащие с такой же заработной платой.
 - Не забывайте полностью удалять данные из таблицы TOP_DOGS перед каждым запуском.

```
CREATE OR REPLACE PROCEDURE TOP_DOGS2 (n IN NUMBER) AS
```

```
CURSOR emp_cursor IS
  SELECT last_name, salary, id
  FROM S_EMP E
  WHERE salary IS NOT NULL
  ORDER BY salary DESC;
v_count NUMBER;
```

```
CURSOR salary_cursor (v_salary NUMBER, v_emp_id NUMBER) IS
  SELECT last_name
  FROM S_EMP
  WHERE salary = v_salary AND id <> v_emp_id;
```

```
v_emps_salary VARCHAR2(2000);
v_salary s_emp.salary%TYPE;
```

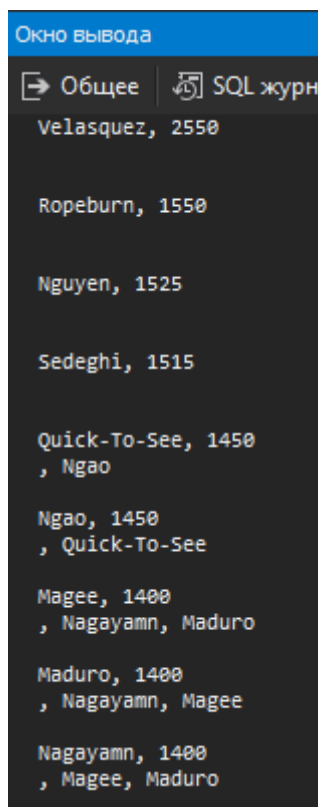
```
BEGIN
  SELECT COUNT(*)
  INTO v_count
  FROM S_EMP;
```

```

IF n = 0 THEN
    dbms_output.put_line('n = 0');
ELSIF n > v_count THEN
    dbms_output.put_line('N больше числа служащих');
ELSE
    FOR emp_record IN emp_cursor LOOP
        EXIT WHEN emp_cursor%ROWCOUNT > n AND v_salary <>
emp_record.salary;
        dbms_output.put_line(emp_record.last_name || ', ' || emp_record.salary);
        v_emps_salary := '';
        FOR salary_record IN salary_cursor(emp_record.salary, emp_record.id) LOOP
            v_emps_salary := v_emps_salary || ', ' || salary_record.last_name;
        END LOOP;
        dbms_output.put_line(v_emps_salary);
        dbms_output.put_line("");

        v_salary := emp_record.salary;
    END LOOP;
END IF;
END;
/
BEGIN
    TOP_DOGS2(7);
END;

```



Заключение

В ходе данной лабораторной работы, я изучил и практиковал использование курсоров в PL/SQL. В процессе выполнения заданий, я освоил различные аспекты в работе с курсорами, включая выборку данных с использованием курсоров и включение параметров в курсоры для более гибкого управления запросами. Также я научился использовать циклы FOR и WHILE для обработки данных в результатах запросов.

Важной частью работы стала работа с курсорами с параметрами, позволяющими задавать условия для выборки данных. Это позволило мне создать процедуры, которые могут анализировать и выводить информацию о сотрудниках и заказах, учитывая заданные параметры.

В целом, данная лабораторная работа позволила мне лучше понять работу с курсорами в PL/SQL и их применение для анализа и обработки данных в базе данных Oracle. Эти навыки могут быть полезными при разработке хранимых процедур и функций, а также при создании отчетов и аналитических запросов к базе данных.