Диалект: Oracle (PL/SQL только в задаче №1)

Запросы для задач 1 и 2 - ниже

ЗАДАЧА 3

```
CREATE TABLE dict_item_prices AS
   SELECT item_id,item_name,item_price,
   created_dttm as valid_from_dt,
   CASE  --as valid_to_dt
        WHEN LEAD(created_dttm,1) OVER (PARTITION BY item_id
ORDER BY created_dttm) IS NULL THEN TIMESTAMP'9999-12-31 12:00:00'
        ELSE LEAD(created_dttm,1) OVER (PARTITION BY item_id ORDER
BY created_dttm)-1 END as valid_to_dt
   FROM item_prices;
```

ЗАДАЧА 4

```
CREATE TABLE customer_aggr AS
   SELECT customer_id,amount_spent_1m, top_item_1m FROM(
      SELECT DISTINCT customer_id,
      FIRST_VALUE(item_name)
        OVER (PARTITION BY customer_id ORDER BY sumprice DESC) as
top_item_1m
      FROM(
          SELECT  customer_id,item_name , sum(item_number*item_price) as
sumprice
          FROM transaction_details
          JOIN dict_item_prices USING(item_id)
          WHERE ((transaction_dttm BETWEEN valid_from_dt AND
valid_to_dt)
                AND (transaction_dttm>=SYSDATE-30))
          GROUP BY customer_id,item_name ORDER BY customer_id))
  JOIN
   (SELECT customer_id, SUM(item_number*item_price) as amount_spent_1m
      FROM transaction_details
        JOIN dict_item_prices USING(item_id)
        WHERE ((transaction_dttm BETWEEN valid_from_dt AND valid_to_dt)
          AND (transaction_dttm>=SYSDATE-30))
        GROUP BY customer_id ORDER BY customer_id) temp
   USING(customer_id);
```

ЗАДАЧА 5

```
CREATE TABLE results AS
   SELECT data as dt,
   counter,
   CASE
   WHEN CONCAT(        /* объединяет округленное значение процента и знак % */
      ROUND(
         (counter/(LAG(counter) OVER (ORDER BY data)))*100-100,1),'%') = '%' THEN NULL
   ELSE CONCAT(
      ROUND(
         (counter/(LAG(counter) OVER (ORDER BY data)))*100-100,1),'%') END as prcnt_growth

   FROM(
      SELECT count(*) as counter, to_char(created_at, 'yyyy-mm') as data
      FROM posts GROUP BY  to_char(created_at, 'yyyy-mm') );
```

ЗАДАЧА 2

```
SELECT t.abonent, a.region_id, maxd as dttm FROM
(SELECT abonent, region_id, dttm FROM test ORDER BY dttm) a
JOIN(
   SELECT abonent, MAX(dttm) as maxd
   FROM test GROUP BY
      SUBSTR(dttm,1,10),
      abonent) t
ON ((a.dttm = t.maxd) and (a.abonent = t.abonent))
ORDER BY dttm
```

Задача 1

Процедура

```
DECLARE
     a NUMBER;

PROCEDURE topN(n IN NUMBER) IS
    v_name VARCHAR2(20);
BEGIN
    EXECUTE IMMEDIATE 'drop table answer';
    EXECUTE IMMEDIATE
        'CREATE TABLE answer(id NUMBER GENERATED ALWAYS AS IDENTITY,
        name VARCHAR2(20))';
    INSERT INTO answer(name)
        SELECT name FROM(
            SELECT name FROM test ORDER BY age,name);

    for i in 1..n LOOP
        SELECT name INTO v_name FROM answer WHERE id=i;
        dbms_output.put_line(v_name);
    END LOOP;
END topN;

BEGIN
    dbms_output.put_line('привет');
    topN(3);
END;
```

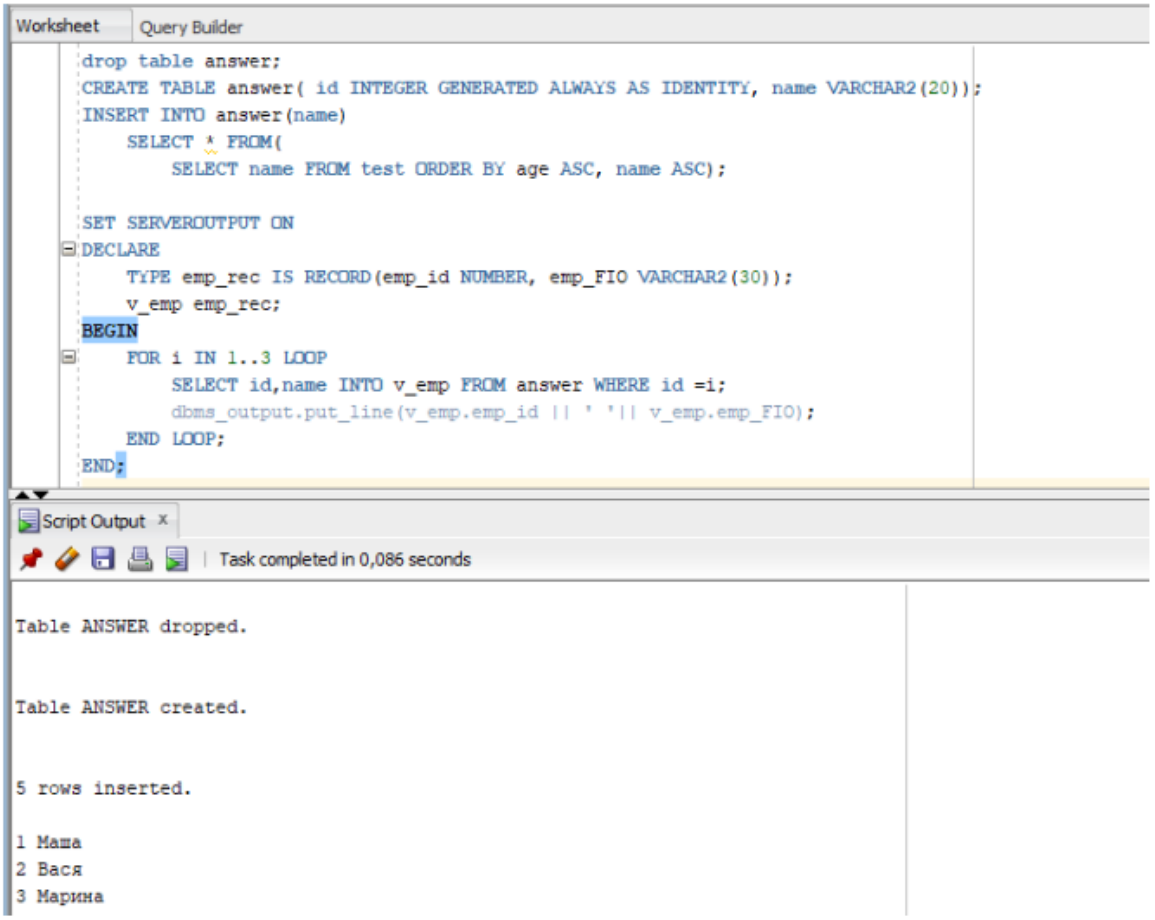
Script Output

Task completed in 0,115 seconds

```
привет
Маша
Вася
Марина


PL/SQL procedure successfully completed.
```

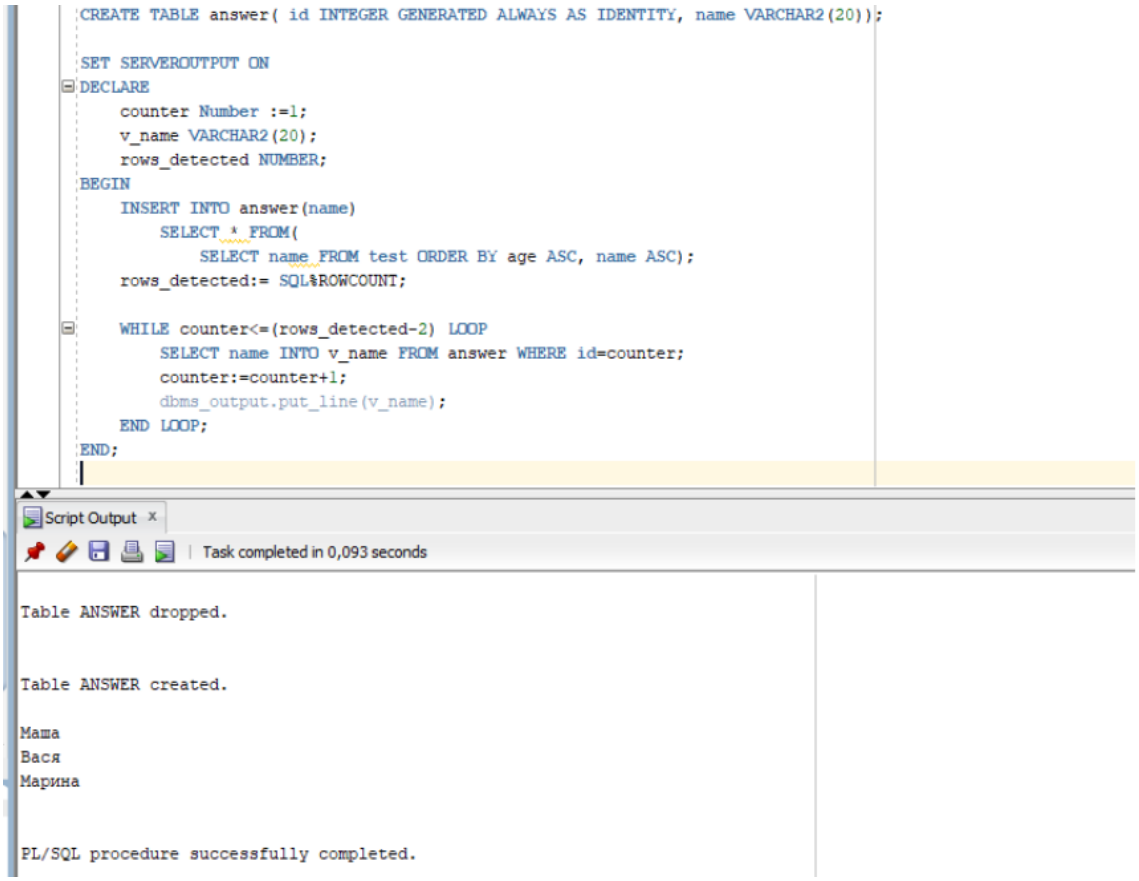
Конструкция




```
DROP TABLE answer;
CREATE TABLE answer (id INTEGER GENERATED ALWAYS AS IDENTITY, name
VARCHAR2(20));
INSERT INTO answer(name)
   SELECT * FROM (
      SELECT name FROM test ORDER BY age ASC, name ASC);

SET SERVEROUTPUT ON
DECLARE
   TYPE emp_rec IS RECORD(emp_id NUMBER, emp_FIO VARCHAR2(30));
   v_emp emp_rec;
BEGIN
   FOR i IN 1..3 LOOP
      SELECT id, name INTO v_emp FROM answer WHERE id=i;
      dbms_output.put_line(v_emp.emp_id || ' ' || v_emp.emp_FIO);
   END LOOP;
END;
```

Использование неявного курсора



```
CREATE TABLE answer( id INTEGER GENERATED ALWAYS AS IDENTITY, name VARCHAR2(20));

SET SERVEROUTPUT ON
DECLARE
    counter Number :=1;
    v_name VARCHAR2(20);
    rows_detected NUMBER;
BEGIN
    INSERT INTO answer(name)
        SELECT * FROM(
            SELECT name FROM test ORDER BY age ASC, name ASC);
    rows_detected:= SQL%ROWCOUNT;

    WHILE counter<=(rows_detected-2) LOOP
        SELECT name INTO v_name FROM answer WHERE id=counter;
        counter:=counter+1;
        dbms_output.put_line(v_name);
    END LOOP;
END;
```

```
Script Output
Task completed in 0,093 seconds

Table ANSWER dropped.


Table ANSWER created.

Маша
Вася
Марина


PL/SQL procedure successfully completed.
```

```
DROP TABLE answer;
CREATE TABLE answer (id INTEGER GENERATED ALWAYS AS IDENTITY, name
VARCHAR2(20));

SET SERVEROUTPUT ON
DECLARE
   counter NUMBER :=1;
   v_name VARCHAR2(20);
   rows_detected NUMBER;
BEGIN
   INSERT INTO answer(name)
      SELECT * FROM (
         SELECT name FROM test ORDER BY age ASC, name ASC);
   rows_detected:=SQL%ROWCOUNT;

   WHILE counter <=(rows_detected-2) LOOP
      SELECT name INTO v_name FROM answer WHERE id=counter;
      counter:=counter+1;
      dbms_output.put_line(v_name);
   END LOOP;
END;
```

Ввод пользователем места в топе

```
drop table answer;
CREATE TABLE answer( id INTEGER GENERATED ALWAYS AS IDENTITY, name VARCHAR2(20));
INSERT INTO answer(name)
    SELECT * FROM(
        SELECT name FROM test ORDER BY age ASC, name ASC);

SET SERVEROUTPUT ON
ACCEPT f PROMPT 'Введите место первого сотрудника: ' /*1*/
ACCEPT s PROMPT 'Введите место второго сотрудника: '/*2*/
ACCEPT t PROMPT 'Введите место третьего сотрудника: '/*3*/
DECLARE
    a NUMBER :=&f;
    b NUMBER :=&s;
    c NUMBER :=&t;
    v_f VARCHAR2(30);
    v_s VARCHAR2(30);
    v_t VARCHAR2(30);
BEGIN
    SELECT name INTO v_f FROM answer WHERE id =a;
    dbms_output.put_line(v_f);
    SELECT name INTO v_s FROM answer WHERE id =b;
    dbms_output.put_line(v_s);
    SELECT name INTO v_t FROM answer WHERE id =c;
    dbms_output.put_line(v_t);
END;
```

Script Output

Task completed in 2,313 seconds

```
Table ANSWER dropped.


Table ANSWER created.


5 rows inserted.

old:DECLARE
    a NUMBER :=&f;
    b NUMBER :=&s;
    c NUMBER :=&t;
    v_f VARCHAR2(30);
    v_s VARCHAR2(30);
    v_t VARCHAR2(30);
BEGIN
    SELECT name INTO v_f FROM answer WHERE id =a;
    dbms_output.put_line(v_f);
    SELECT name INTO v_s FROM answer WHERE id =b;
    dbms_output.put_line(v_s);
    SELECT name INTO v_t FROM answer WHERE id =c;
    dbms_output.put_line(v_t);
END;
new:DECLARE
    a NUMBER :=1;
    b NUMBER :=2;
    c NUMBER :=3;
    v_f VARCHAR2(30);
    v_s VARCHAR2(30);
    v_t VARCHAR2(30);
BEGIN
    SELECT name INTO v_f FROM answer WHERE id =a;
    dbms_output.put_line(v_f);
    SELECT name INTO v_s FROM answer WHERE id =b;
    dbms_output.put_line(v_s);
    SELECT name INTO v_t FROM answer WHERE id =c;
    dbms_output.put_line(v_t);
END;
Маша
Вася
Марина
```

```
CREATE TABLE answer (id INTEGER GENERATED ALWAYS AS
IDENTITY, name VARCHAR2(20));
INSERT into answer(name)
   SELECT * FROM(
      SELECT name FROM test ORDER BY age ASC, name ASC);

SET SERVEROUTPUT ON
ACCEPT f PROMPT 'Введите место первого сотрудника: '; /*1*/
ACCEPT s PROMPT 'Введите место первого сотрудника: '; /*2*/
ACCEPT t PROMPT 'Введите место первого сотрудника: '; /*3*/
DECLARE
   a NUMBER  :=&f;
   b NUMBER  :=&s;
   c NUMBER  :=&t;
   v_f VARCHAR2(30);
   v_s VARCHAR2(30);
   v_t VARCHAR2(30);
BEGIN
   SELECT name INTO v_f FROM answer WHERE id=a;
   dbms_output.put_line(v_f);
   SELECT name INTO v_s FROM answer WHERE id=b;
   dbms_output.put_line(v_s);
   SELECT name INTO v_t FROM answer WHERE id=c;
   dbms_output.put_line(v_t);
END;
```

Стандартные способы

```
SELECT name FROM test ORDER BY age, name
FETCH NEXT 3 ROWS ONLY;
```

Script Output

Task completed in 0,034 seconds

```
NAME
--------------------
Маша
Вася
Марина
```

Worksheet Query Builder

```
SELECT name FROM test ORDER BY age ASC, name ASC LIMIT 3;
```

Script Output

Task completed in 0,034 seconds

```
NAME
--------------------
Маша
Вася
Марина
```

```
SELECT * FROM(
SELECT name FROM test ORDER BY age ASC, name ASC) WHERE ROWNUM<=3;
```

Script Output

Task completed in 0,029 seconds

```
NAME
--------------------
Маша
Вася
Марина
```