

Рабочая тетрадь №1

*База данных: **Oracle***

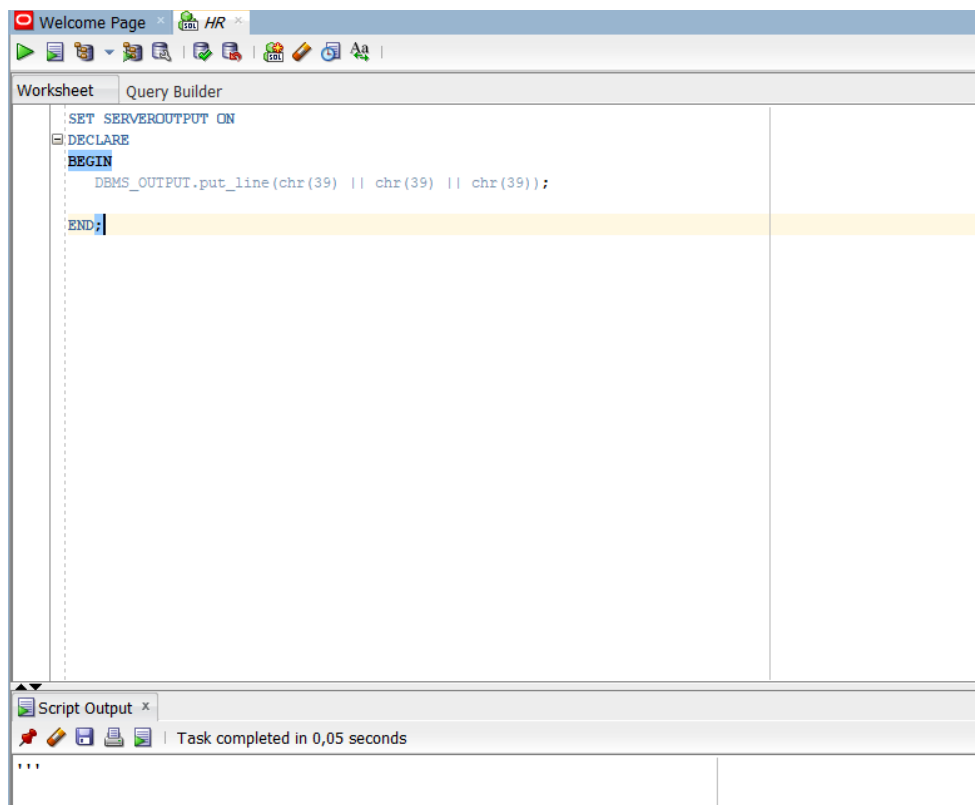
Рабочая тетрадь №1

Основные требования:

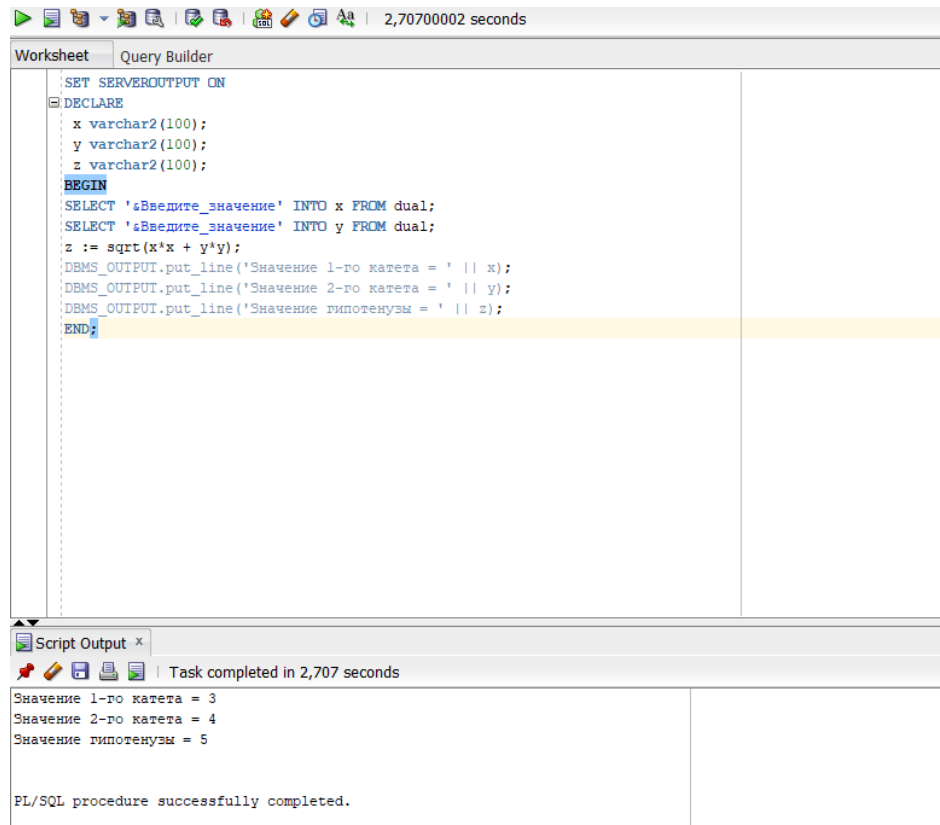
- *Выполнение команд производится в терминале или в графическом редакторе*
- *Результат работы команды предоставляется в виде скриншота*
- *Оформление ведется в удобном текстовом редакторе*

- Рабочие тетради должны предоставляться на проверку в формате .pdf

Рабочая тетрадь №1



Рабочая тетрадь №1



The screenshot shows a PL/SQL development environment. The top toolbar indicates a runtime of 2,707,000,002 seconds. The main editor window, titled 'Worksheet' and 'Query Builder', contains the following PL/SQL code:

```
SET SERVEROUTPUT ON
DECLARE
  x varchar2(100);
  y varchar2(100);
  z varchar2(100);
BEGIN
  SELECT '%Введите_значение' INTO x FROM dual;
  SELECT '%Введите_значение' INTO y FROM dual;
  z := sqrt(x*x + y*y);
  DBMS_OUTPUT.put_line('Значение 1-го катета = ' || x);
  DBMS_OUTPUT.put_line('Значение 2-го катета = ' || y);
  DBMS_OUTPUT.put_line('Значение гипотенузы = ' || z);
END;
```

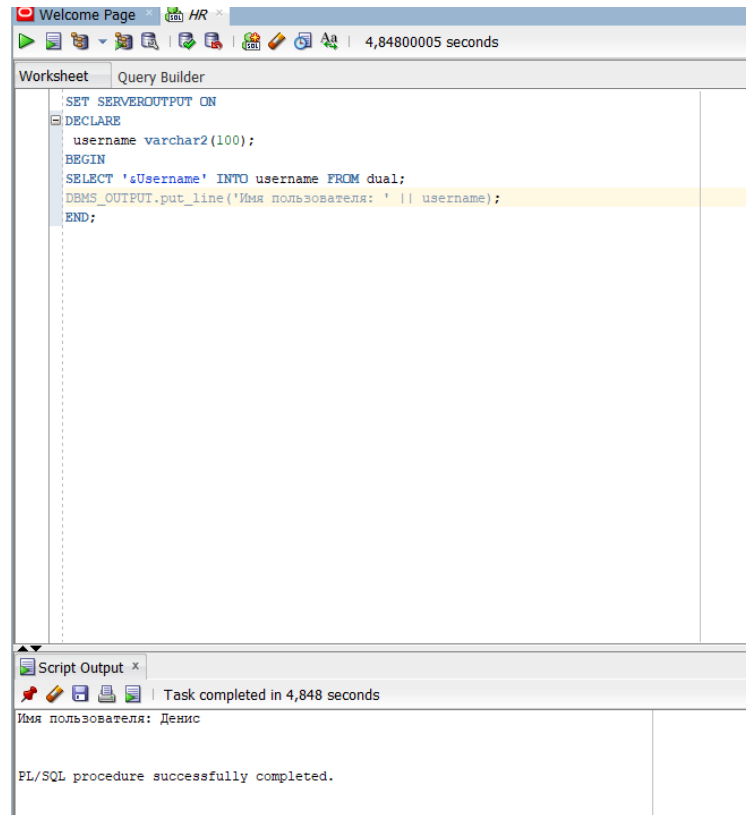
The bottom panel, titled 'Script Output', shows the results of the execution:

```
Значение 1-го катета = 3
Значение 2-го катета = 4
Значение гипотенузы = 5

PL/SQL procedure successfully completed.
```

The output panel also indicates that the task was completed in 2,707 seconds.

Рабочая тетрадь №1



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running, saving, and other database operations. The main window is titled 'Worksheet' and contains the following PL/SQL code:

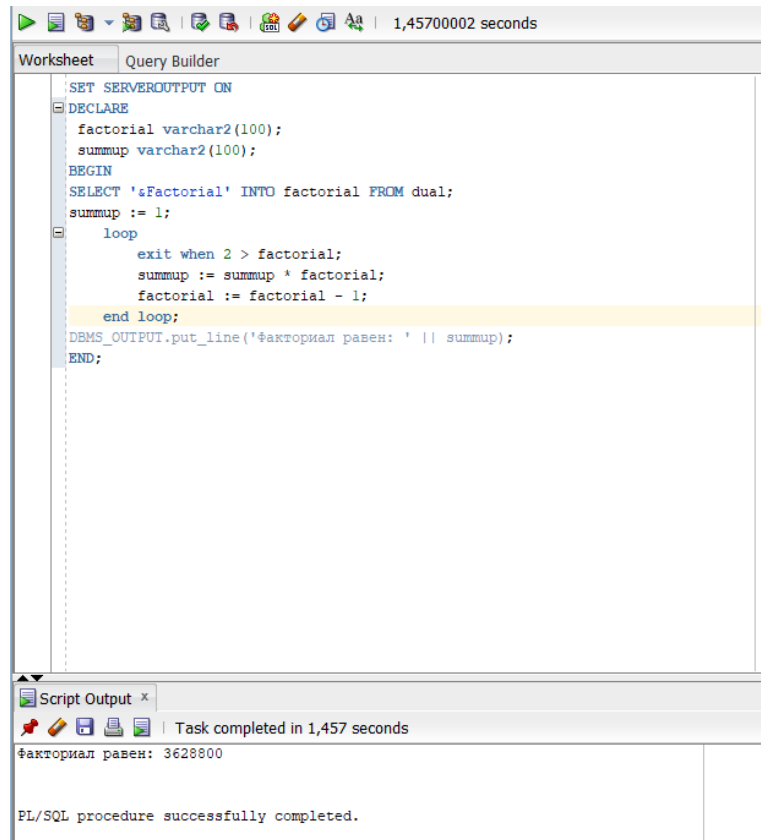
```
SET SERVEROUTPUT ON
DECLARE
  username varchar2(100);
BEGIN
  SELECT '&Username' INTO username FROM dual;
  DBMS_OUTPUT.put_line('Имя пользователя: ' || username);
END;
```

The bottom pane, titled 'Script Output', shows the execution results:

```
Task completed in 4,848 seconds
Имя пользователя: Денис

PL/SQL procedure successfully completed.
```

Рабочая тетрадь №1



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

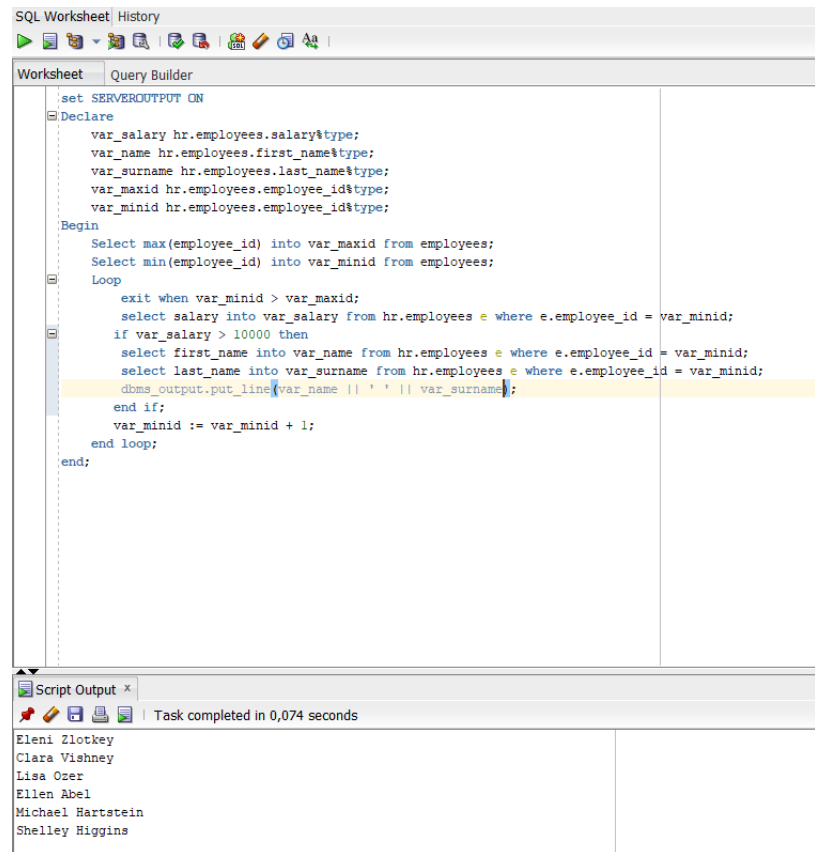
```
SET SERVEROUTPUT ON
DECLARE
    factorial varchar2(100);
    summap varchar2(100);
BEGIN
    SELECT 'sFactorial' INTO factorial FROM dual;
    summap := 1;
    loop
        exit when 2 > factorial;
        summap := summap * factorial;
        factorial := factorial - 1;
    end loop;
    DBMS_OUTPUT.put_line('факториал равен: ' || summap);
END;
```

Below the editor is a 'Script Output' window showing the execution results:

```
Task completed in 1,457 seconds
факториал равен: 3628800

PL/SQL procedure successfully completed.
```

Рабочая тетрадь №1



The screenshot shows a SQL Worksheet interface with a 'Worksheet' tab and a 'Query Builder' tab. The 'Worksheet' tab contains a PL/SQL script. The script declares variables for salary, first name, last name, and employee ID, then finds the maximum and minimum employee IDs. It enters a loop that starts from the minimum ID and iterates until it exceeds the maximum ID. In each iteration, it selects the salary, first name, and last name of the employee with the current ID and outputs them using dbms_output.put_line. The script ends with an end loop and end statements. Below the script, the 'Script Output' tab shows the results of the execution, listing the names of the employees: Eleni Zlotkey, Clara Vishney, Lisa Ozer, Ellen Abel, Michael Hartstein, and Shelley Higgins. The output also indicates that the task was completed in 0,074 seconds.

```
SQL Worksheet History
set SERVEROUTPUT ON
Declare
var_salary hr.employees.salary%type;
var_name hr.employees.first_name%type;
var_surname hr.employees.last_name%type;
var_maxid hr.employees.employee_id%type;
var_minid hr.employees.employee_id%type;
Begin
Select max(employee_id) into var_maxid from employees;
Select min(employee_id) into var_minid from employees;
Loop
exit when var_minid > var_maxid;
select salary into var_salary from hr.employees e where e.employee_id = var_minid;
if var_salary > 10000 then
select first_name into var_name from hr.employees e where e.employee_id = var_minid;
select last_name into var_surname from hr.employees e where e.employee_id = var_minid;
dbms_output.put_line(var_name || ' ' || var_surname);
end if;
var_minid := var_minid + 1;
end loop;
end;
```

Script Output x
Task completed in 0,074 seconds

Eleni Zlotkey
Clara Vishney
Lisa Ozer
Ellen Abel
Michael Hartstein
Shelley Higgins

Рабочая тетрадь №1

```
Worksheet | Query Builder
set SERVEROUTPUT ON
Declare
    var_depindex hr.employees.department_id%type;
    var_selectindex hr.employees.department_id%type;
    var_index hr.employees.department_id%type;
    var_sumsalary hr.employees.salary%type;
    var_sum hr.employees.salary%type;
    var_salary hr.employees.salary%type;
    var_maxid hr.employees.employee_id%type;
    var_minid hr.employees.employee_id%type;
    var_counter hr.employees.employee_id%type;
Begin
    Select max(employee_id) into var_maxid from employees;
    Select min(employee_id) into var_minid from employees;
    var_counter := var_minid;
    var_sum := 0;
    var_sumsalary := 0;
    var_selectindex := 0;
    loop
        exit when var_counter > var_maxid;
        select department_id into var_depindex from employees e where e.employee_id = var_minid;
        if var_depindex != var_selectindex then
            var_sum := 0;
        Loop
            exit when var_minid > var_maxid;
            select department_id into var_index from employees e where e.employee_id = var_minid;
            if var_index = var_depindex then
                select salary into var_salary from hr.employees e where e.employee_id = var_minid;
                var_sum := var_sum + var_salary;
            end if;
            var_minid := var_minid + 1;
        end loop;
    end if;
    var_counter := var_counter + 1;
    var_selectindex := var_index;
    var_sumsalary := var_sumsalary + var_salary;
    var_sum := var_sum + var_salary;
    var_minid := var_minid + 1;
end loop;
```

Worksheet Query Builder

```
var_maxid hr.employees.employee_id%type;  
var_minid hr.employees.employee_id%type;  
var_counter hr.employees.employee_id%type;  
  
Begin  
Select max(employee_id) into var_maxid from employees;  
Select min(employee_id) into var_minid from employees;  
var_counter := var_minid;  
var_sum := 0;  
var_sumsalary := 0;  
var_selectindex := 0;  
loop  
exit when var_counter > var_maxid;  
select department_id into var_depindex from employees e where e.employee_id = var_minid;  
if var_depindex != var_selectindex then  
var_sum := 0;  
Loop  
exit when var_minid > var_maxid;  
select department_id into var_index from employees e where e.employee_id = var_minid;  
if var_index = var_depindex then  
select salary into var_salary from hr.employees e where e.employee_id = var_minid;  
var_sum := var_sum + var_salary;  
end if;  
var_minid := var_minid + 1;  
end loop;  
end if;  
var_counter := var_counter + 1;  
var_minid := var_counter;  
if var_sum > var_sumsalary then  
var_sumsalary := var_sum;  
var_selectindex := var_depindex;  
end if;  
end loop;  
dms_output.put_line('Самый дорогостоящий отдел по получаемой зарплате сотрудников, с ID : ' || var_selectindex);  
end;
```

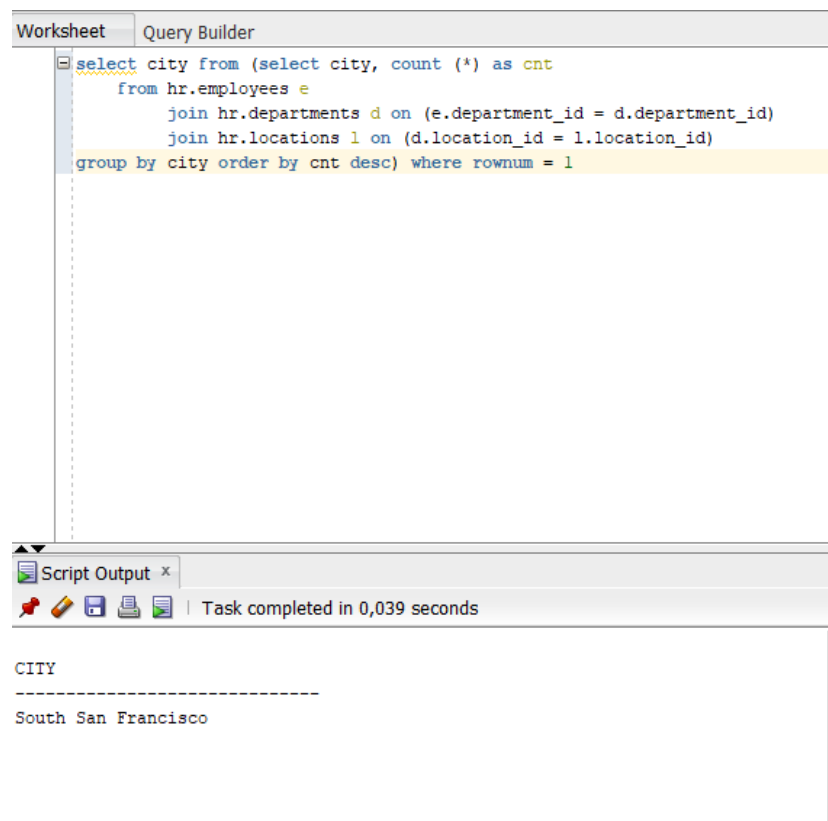
Script Output x

Task completed in 0,141 seconds

Самый дорогостоящий отдел по получаемой зарплате сотрудников, с ID : 80

PL/SQL procedure successfully completed.

Рабочая тетрадь №1



The screenshot shows a database query tool interface. At the top, there are two tabs: "Worksheet" and "Query Builder". The "Query Builder" tab is active, displaying a SQL query. The query is as follows:

```
select city from (select city, count (*) as cnt
  from hr.employees e
    join hr.departments d on (e.department_id = d.department_id)
    join hr.locations l on (d.location_id = l.location_id)
 group by city order by cnt desc) where rownum = 1
```

Below the query editor, there is a "Script Output" window. It shows the results of the query execution. The output is as follows:

```
CITY
-----
South San Francisco
```

The "Script Output" window also displays a status message: "Task completed in 0,039 seconds".

Рабочая тетрадь №1

```
Worksheet | Query Builder
create table student
(
  id number(3),
  Name varchar2(25),
  Surname varchar2(25),
  Otchestvo varchar2(25),
  Student_group varchar2(15),
  Year number(4),
  Studak varchar2(20),
  CONSTRAINT promo_studak_u UNIQUE (Studak)
);
insert into student(id,Name, Surname, Otchestvo, Student_group, Year, Studak)
values(100, 'David', 'Chenkaleev', 'Borisovich', 'IVBO-04-21', 2021, '21I5678');
insert into student(id,Name, Surname, Otchestvo, Student_group, Year, Studak)
values(101,'Ivan', 'Stoka', 'Pavlovich', 'IVBO-07-21', 2021, '21I1228');
insert into student(id,Name, Surname, Otchestvo, Student_group, Year, Studak)
values(102,'Kate', 'Skorohodova', 'Alexandrovna', 'IVBO-07-21', 2021, '21I3344');
insert into student(id,Name, Surname, Otchestvo, Student_group, Year, Studak)
values(103,'Valentine', 'Yakushev', 'Petrovich', 'IVBO-07-20', 2020, '20I1717');
```

Script Output x

Task completed in 0,088 seconds

Table STUDENT created.

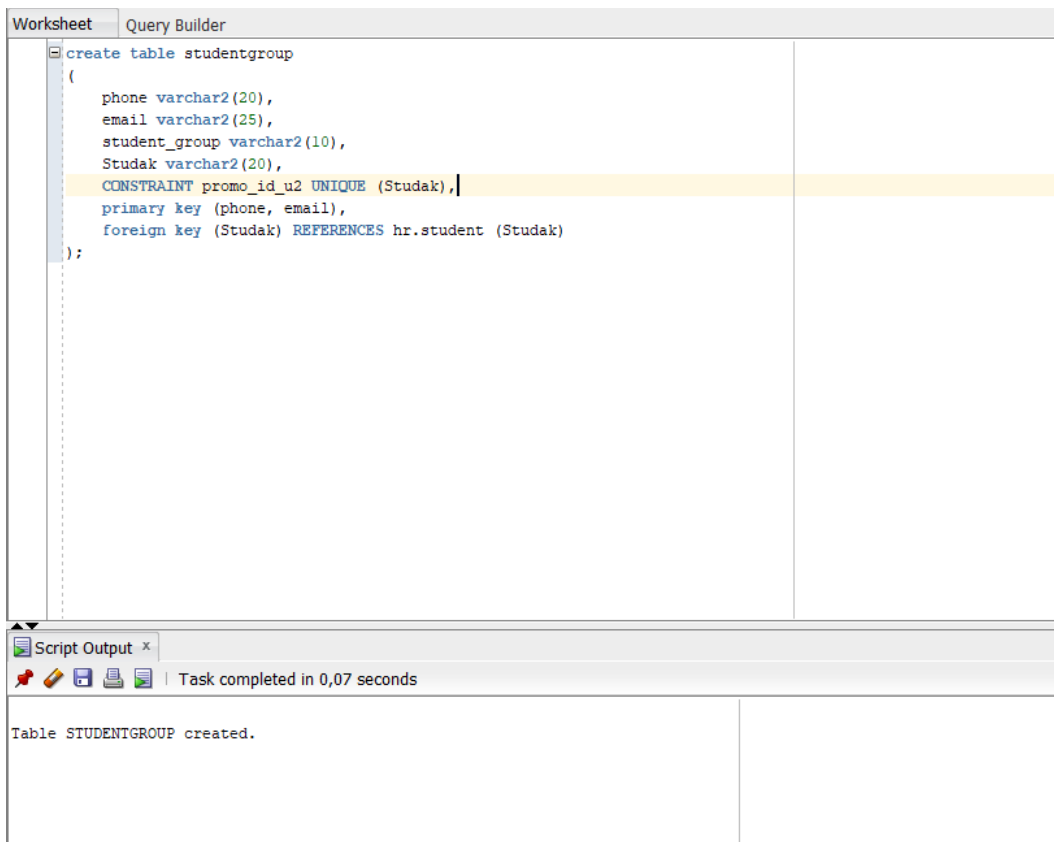
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Рабочая тетрадь №1



The screenshot shows a database query editor with two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a SQL script to create a table named 'studentgroup'. The script includes columns for 'phone', 'email', 'student_group', and 'Studak', along with a primary key and a foreign key constraint. The 'Script Output' tab at the bottom shows the successful execution of the script, indicating that the table 'STUDENTGROUP' has been created.

```
create table studentgroup
(
  phone varchar2(20),
  email varchar2(25),
  student_group varchar2(10),
  Studak varchar2(20),
  CONSTRAINT promo_id_u2 UNIQUE (Studak),
  primary key (phone, email),
  foreign key (Studak) REFERENCES hr.student (Studak)
);
```

Script Output x
Task completed in 0,07 seconds
Table STUDENTGROUP created.