PRIME INTUIT

SQL - Functions

SWAROOP N C , 28th April of 2022

What's Ahead ?

- Functions
- Types
  - User Defined
  - Pre Defined
    - Group
    - Character
    - Numeric
    - Date
    - Special

Functions – it is a re-usable program that returns a value.

There are 2 types,

- Pre – defined
- User defined

Pre – defined

- → GROUP functions
- → CHARACTER functions
- → NUMERIC functions
- → DATE functions
- → SPECIAL functions

These are used both in SQL and PL/SQL. PL – Procedural Language (it‟s a extension to SQL, can contain IF statements, loops, exceptions, OOPs, etc ..)

User – defined
Used only in PL/SQL and we will not study it here.

We have already learnt about GROUP functions.

Now, let us study the various CHARACTER functions.

CHARACTER functions

a) Upper

b) Lower

c) Length

Script Output
Query Result
SQL

```sql
select upper('oracle') from dual;
```

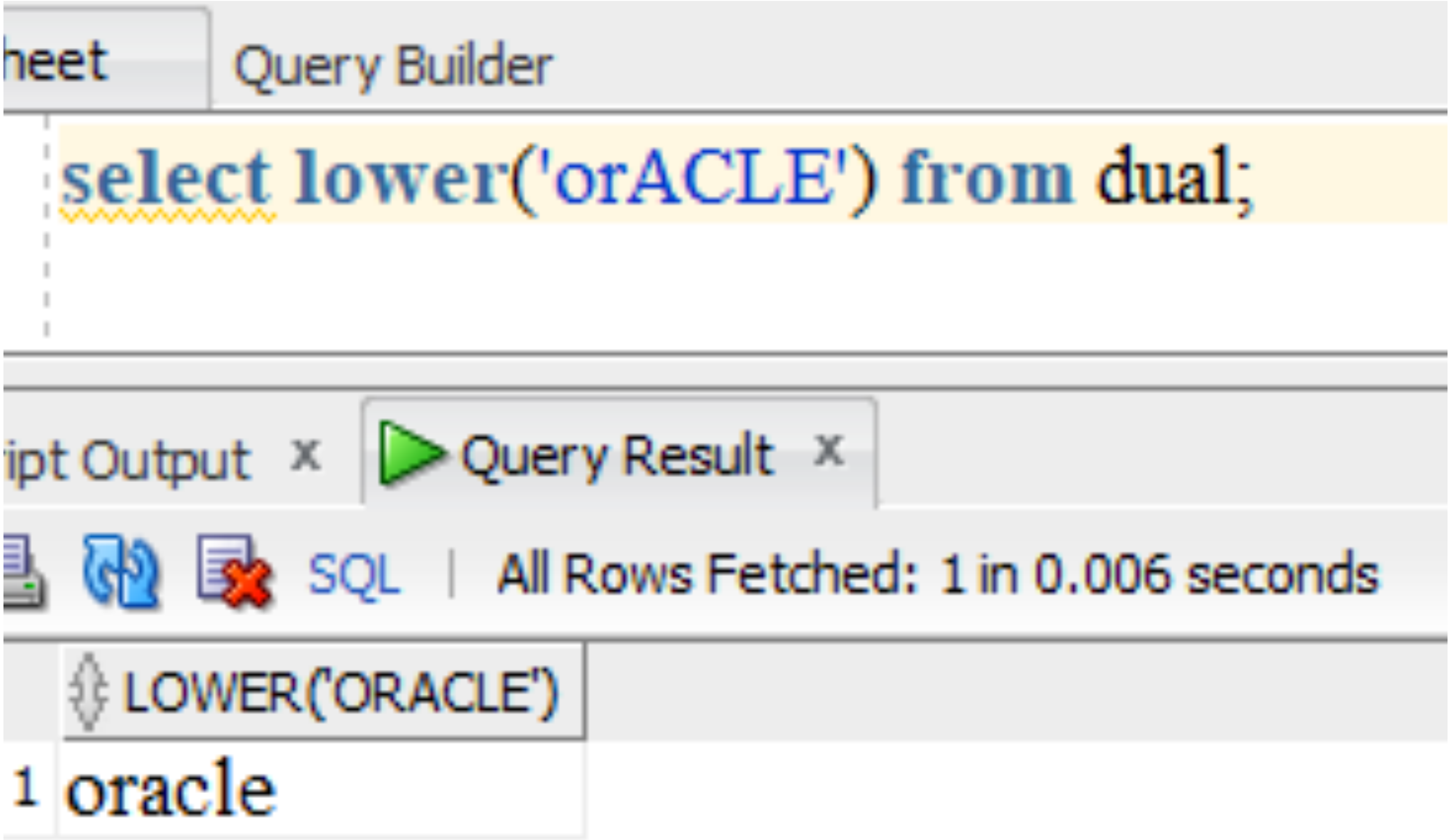
```
All Rows Fetched: 1 in 0.031 seconds

   UPPER('ORACLE')
1  ORACLE
```

In the 1st query, we see something called as dual.

Dual – is a dummy table which is used for performing some independent operations which will not depend on any of the existing tables.

For ex,

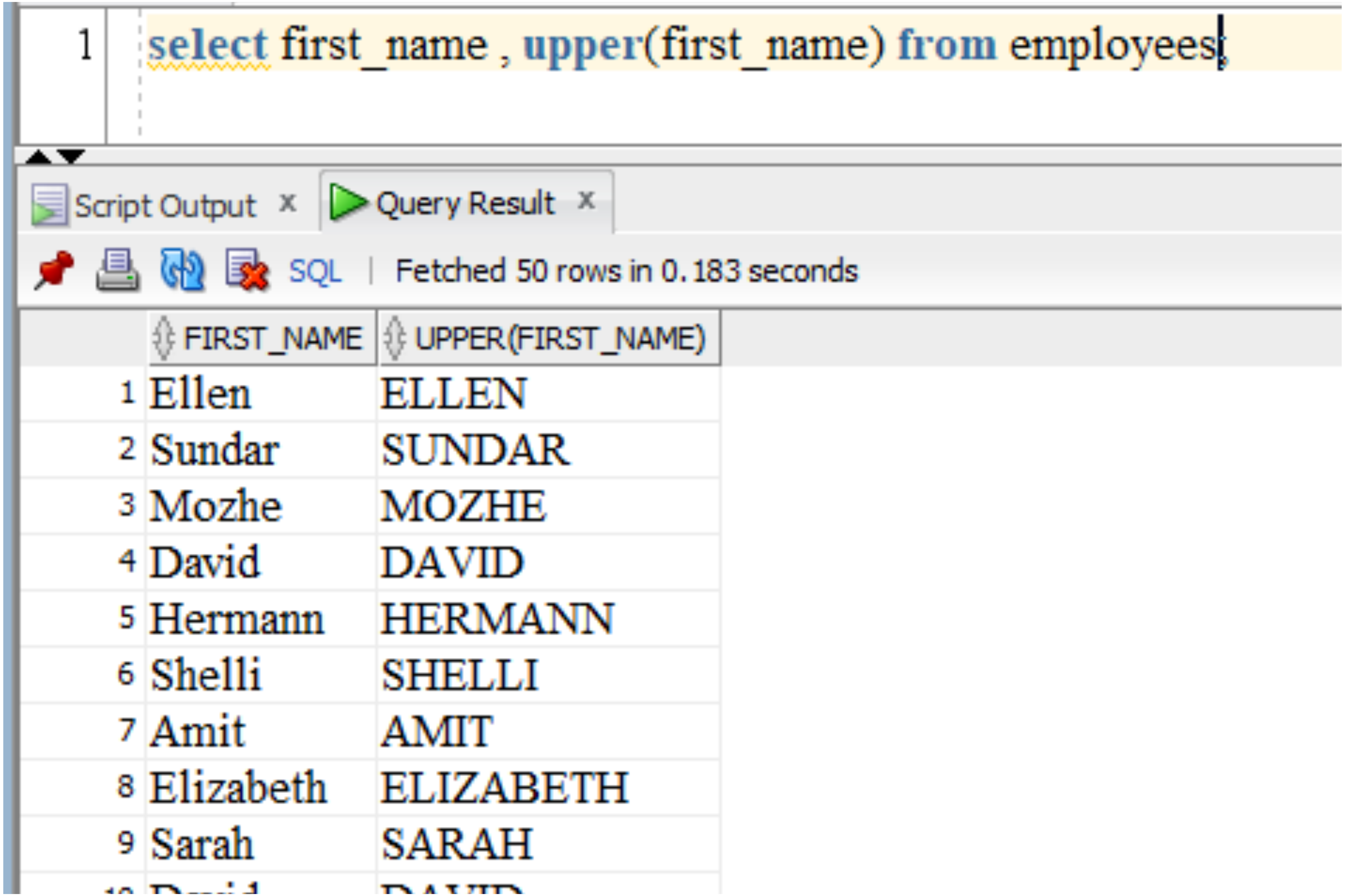
We use dual – when the data is not present in any of the existing tables. Then we use dual.



```sql
select lower('orACLE') from dual;
```

All Rows Fetched: 1 in 0.006 seconds

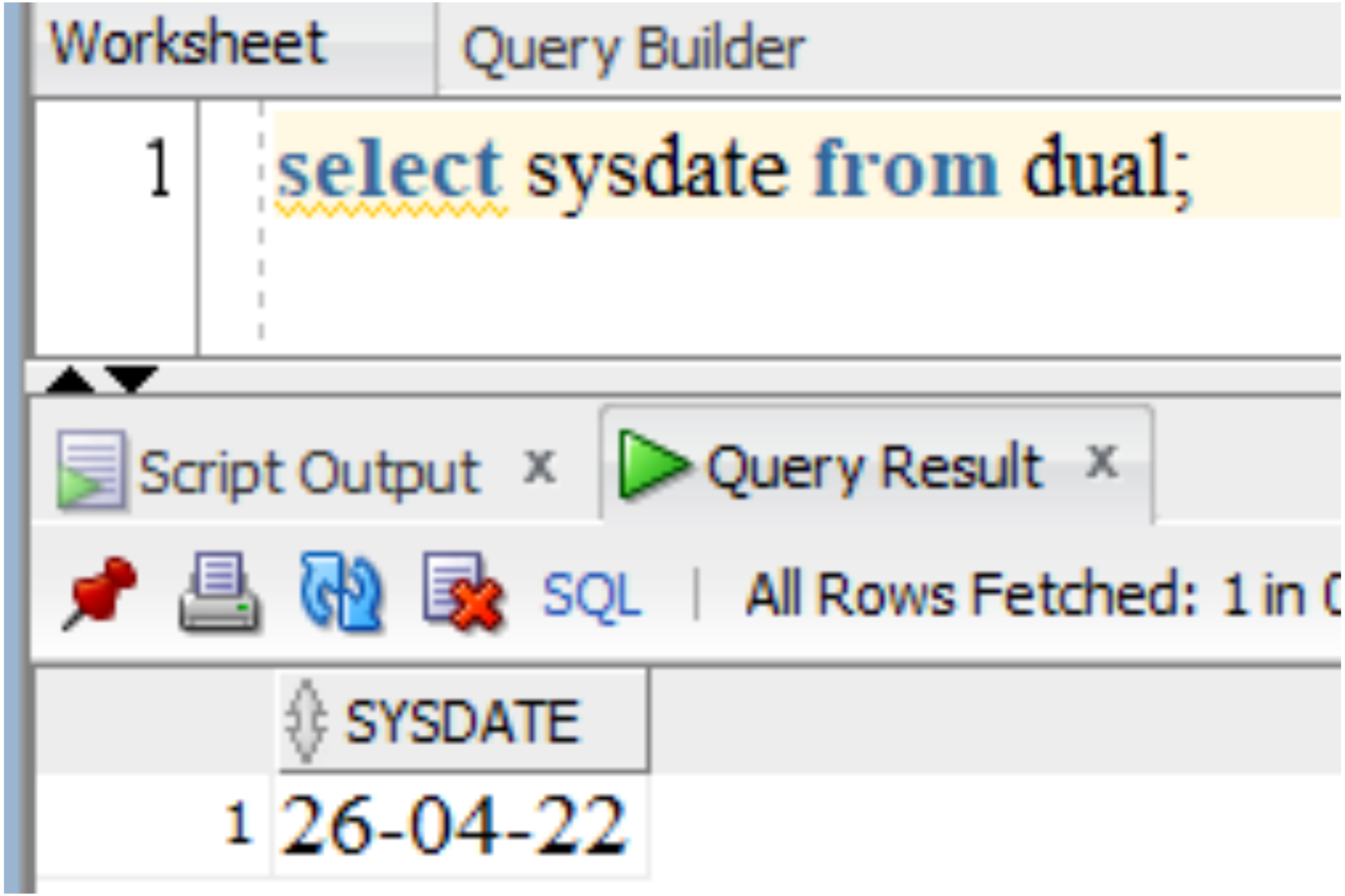
```
  LOWER('ORACLE')
1 oracle
```



```
select first_name , upper(first_name) from employees
```

Fetched 50 rows in 0.183 seconds

	FIRST_NAME	UPPER(FIRST_NAME)
1	Ellen	ELLEN
2	Sundar	SUNDAR
3	Mozhe	MOZHE
4	David	DAVID
5	Hermann	HERMANN
6	Shelli	SHELLI
7	Amit	AMIT
8	Elizabeth	ELIZABETH
9	Sarah	SARAH
10	David	DAVID

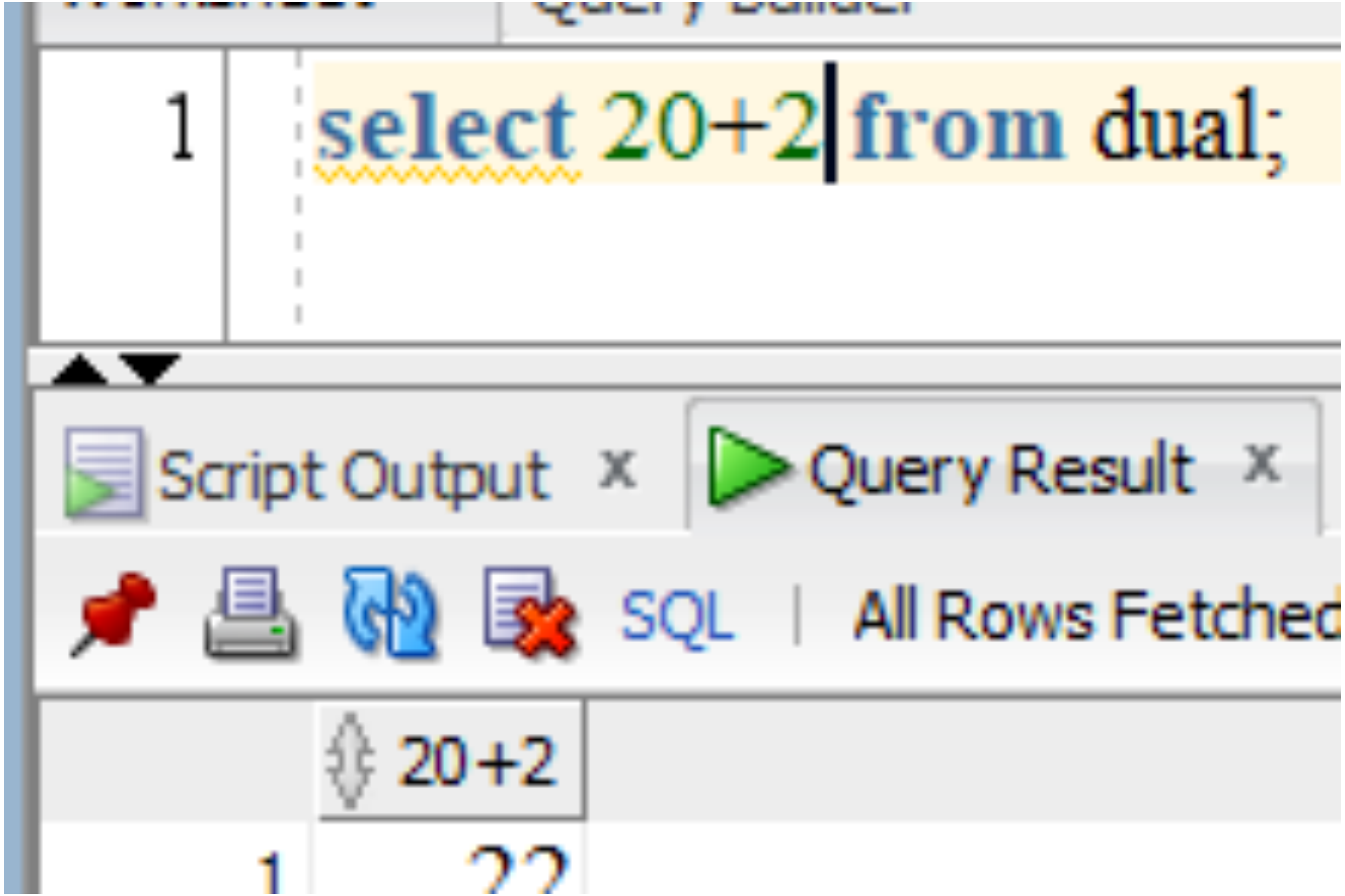


```
select sysdate from dual;
```

All Rows Fetched: 1 in

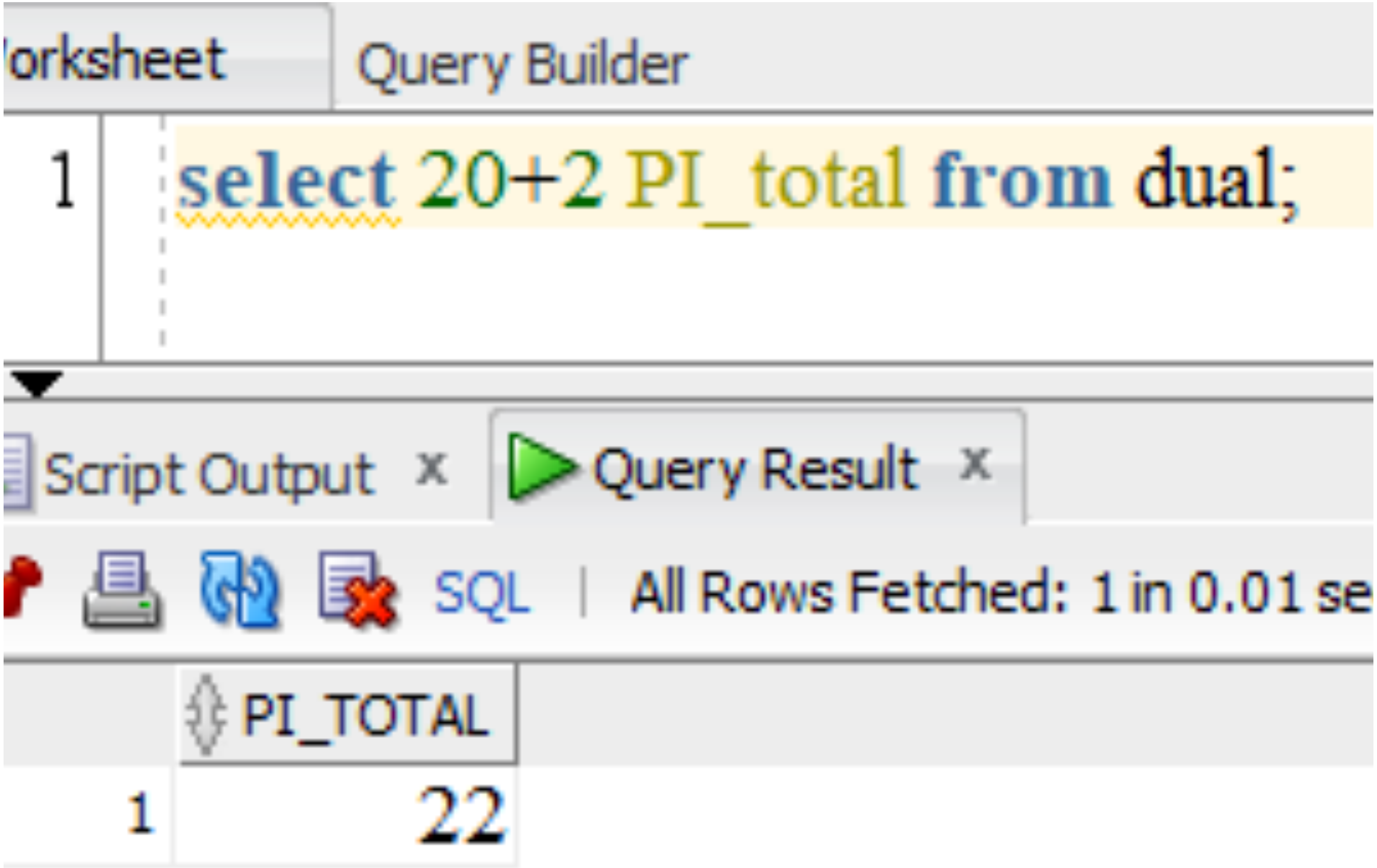
```
    SYSDATE
1   26-04-22
```

This gives the system date.



```
select 20+2 from dual;
```

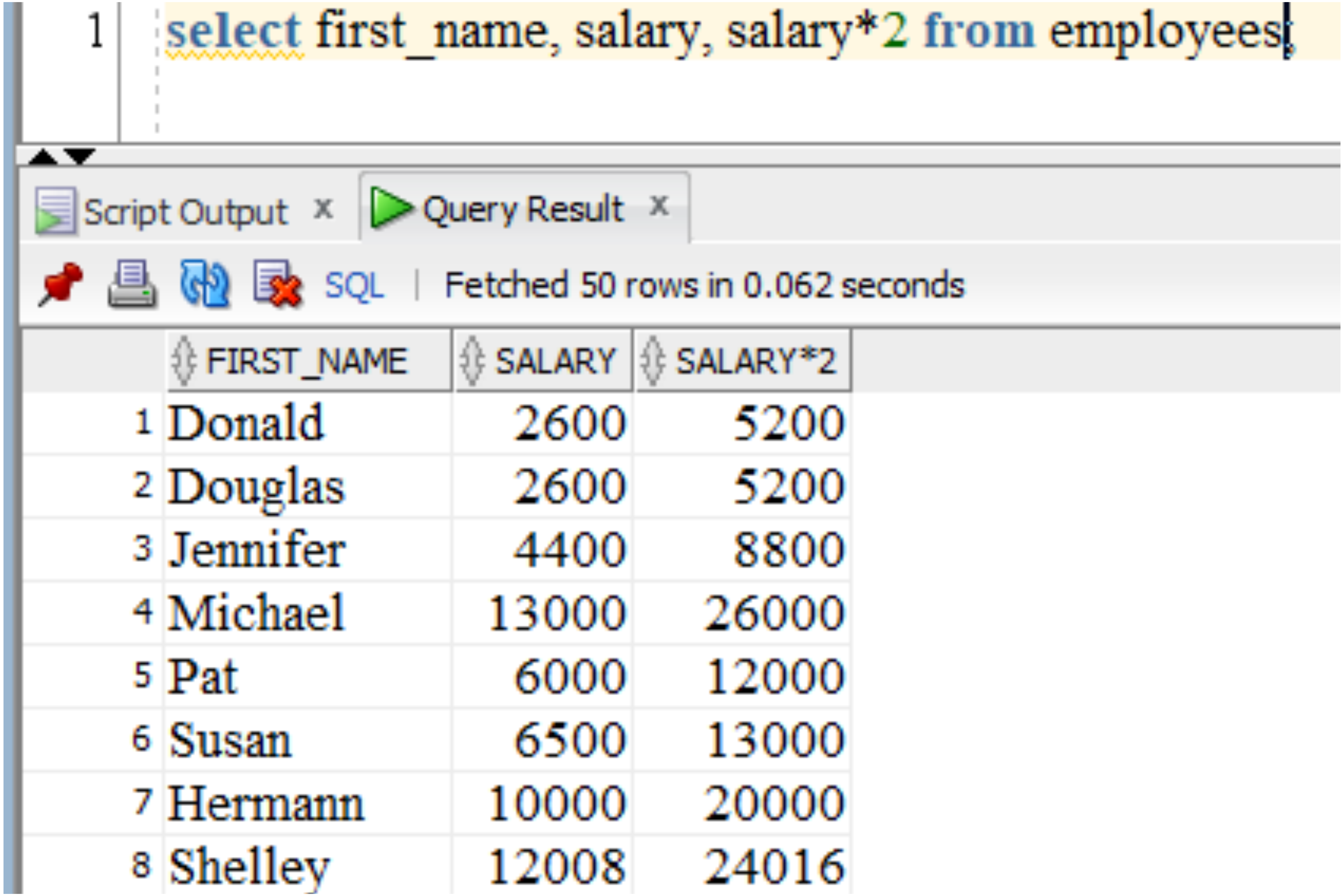
```
     20+2
1      22
```



```
select 20+2 PI_total from dual;
```

All Rows Fetched: 1 in 0.01 se

```
   PI_TOTAL
1        22
```

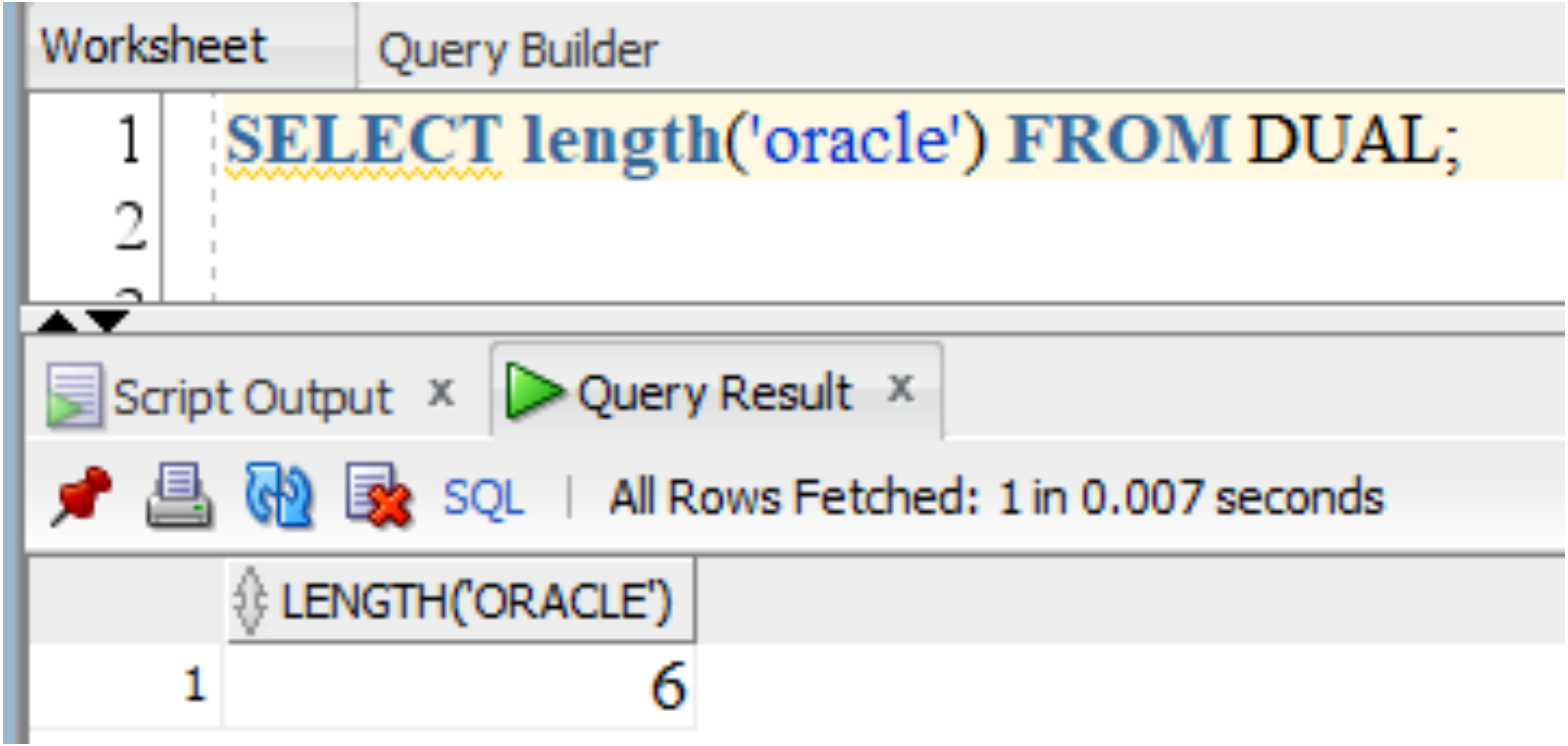


```sql
select first_name, salary, salary*2 from employees;
```

Fetched 50 rows in 0.062 seconds

	FIRST_NAME	SALARY	SALARY*2
1	Donald	2600	5200
2	Douglas	2600	5200
3	Jennifer	4400	8800
4	Michael	13000	26000
5	Pat	6000	12000
6	Susan	6500	13000
7	Hermann	10000	20000
8	Shelley	12008	24016

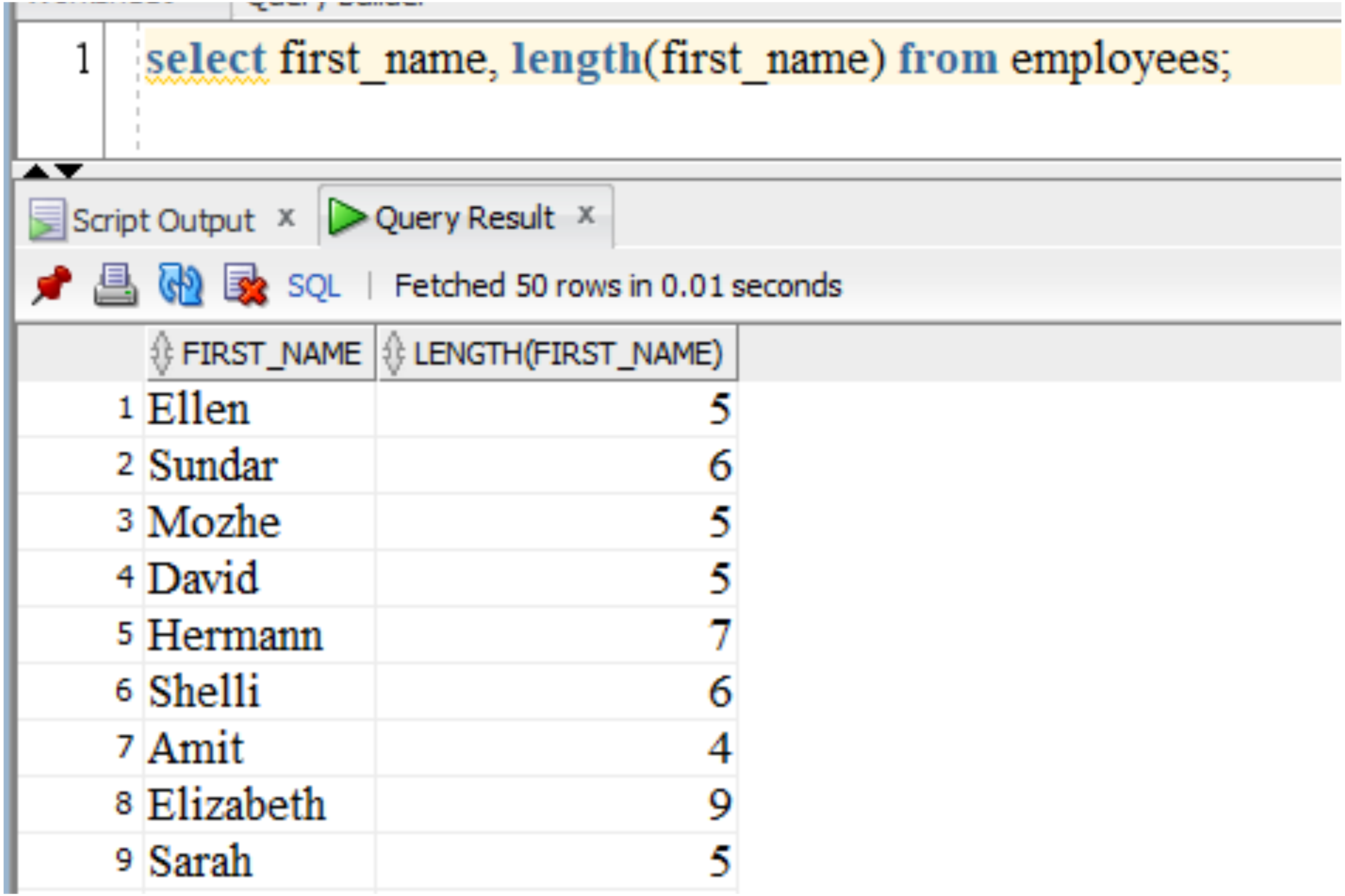
Length– it returns the length of a given string.



```
SELECT length('oracle') FROM DUAL;
```

All Rows Fetched: 1 in 0.007 seconds

	LENGTH('ORACLE')
1	6



```
select first_name, length(first_name) from employees;
```

Fetched 50 rows in 0.01 seconds

	FIRST_NAME	LENGTH(FIRST_NAME)
1	Ellen	5
2	Sundar	6
3	Mozhe	5
4	David	5
5	Hermann	7
6	Shelli	6
7	Amit	4
8	Elizabeth	9
9	Sarah	5

Display all the employees whose name & job is having exactly 5 characters

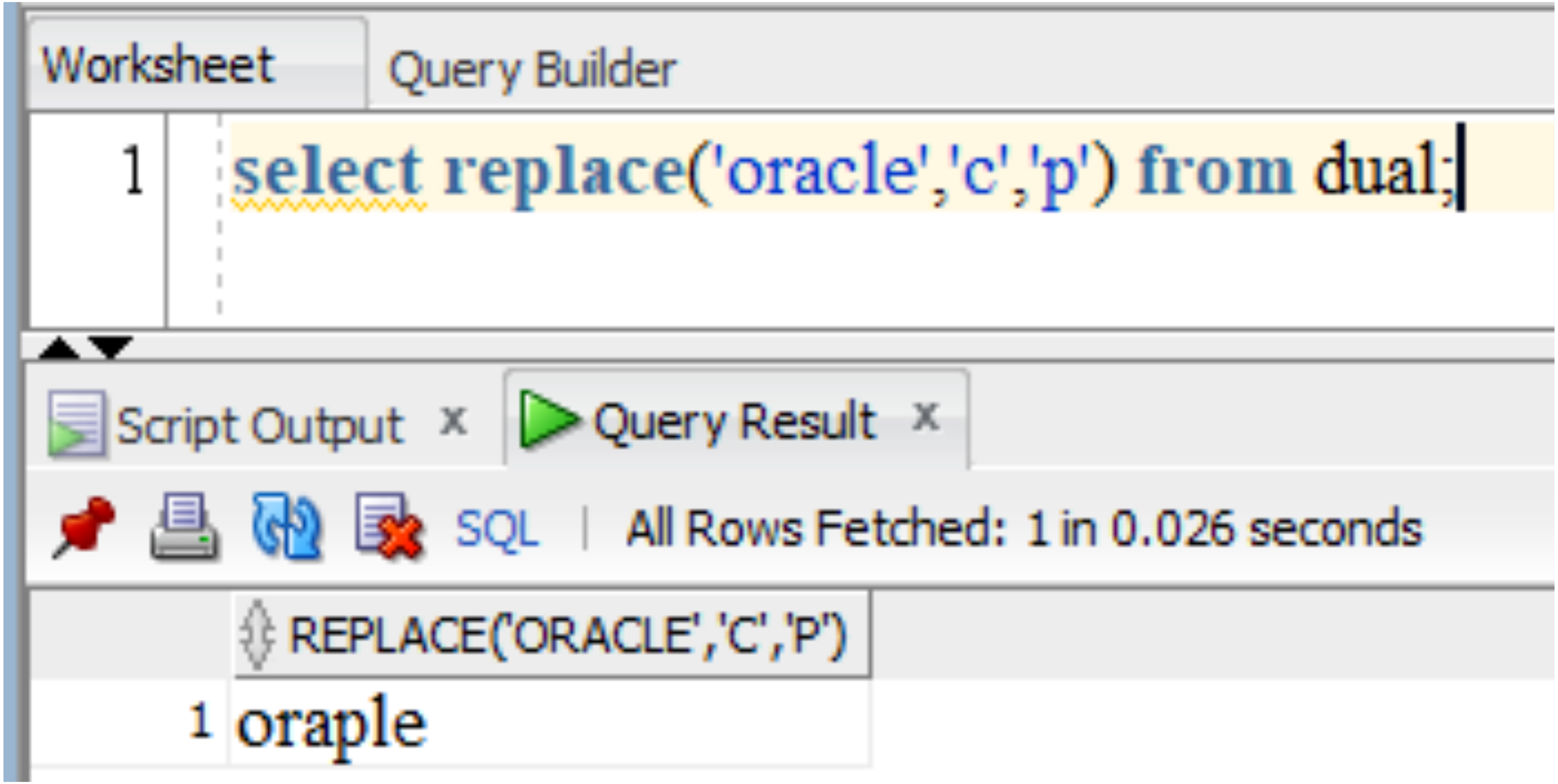
Worksheet
Query Builder
Script Output
Query Result
SQL

```sql
select * from employees where length(first_name)=5 and length(last_name)=5;
```

All Rows Fetched: 5 in 0.011 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	104	Bruce	Ernst	BERNST	590.423.4568	21-05-07	IT_PROG	6000	(null)	103	60
2	125	Julia	Nayer	JNAYER	650.124.1214	16-07-05	ST_CLERK	3200	(null)	120	50
3	188	Kelly	Chung	KCHUNG	650.505.1876	14-06-05	SH_CLERK	3800	(null)	122	50
4	195	Vance	Jones	VJONES	650.501.4876	17-03-07	SH_CLERK	2800	(null)	123	50
5	196	Alana	Walsh	AWALSH	650.507.9811	24-04-06	SH_CLERK	3100	(null)	124	50

REPLACE It replaces the old value with a new value in the given string.

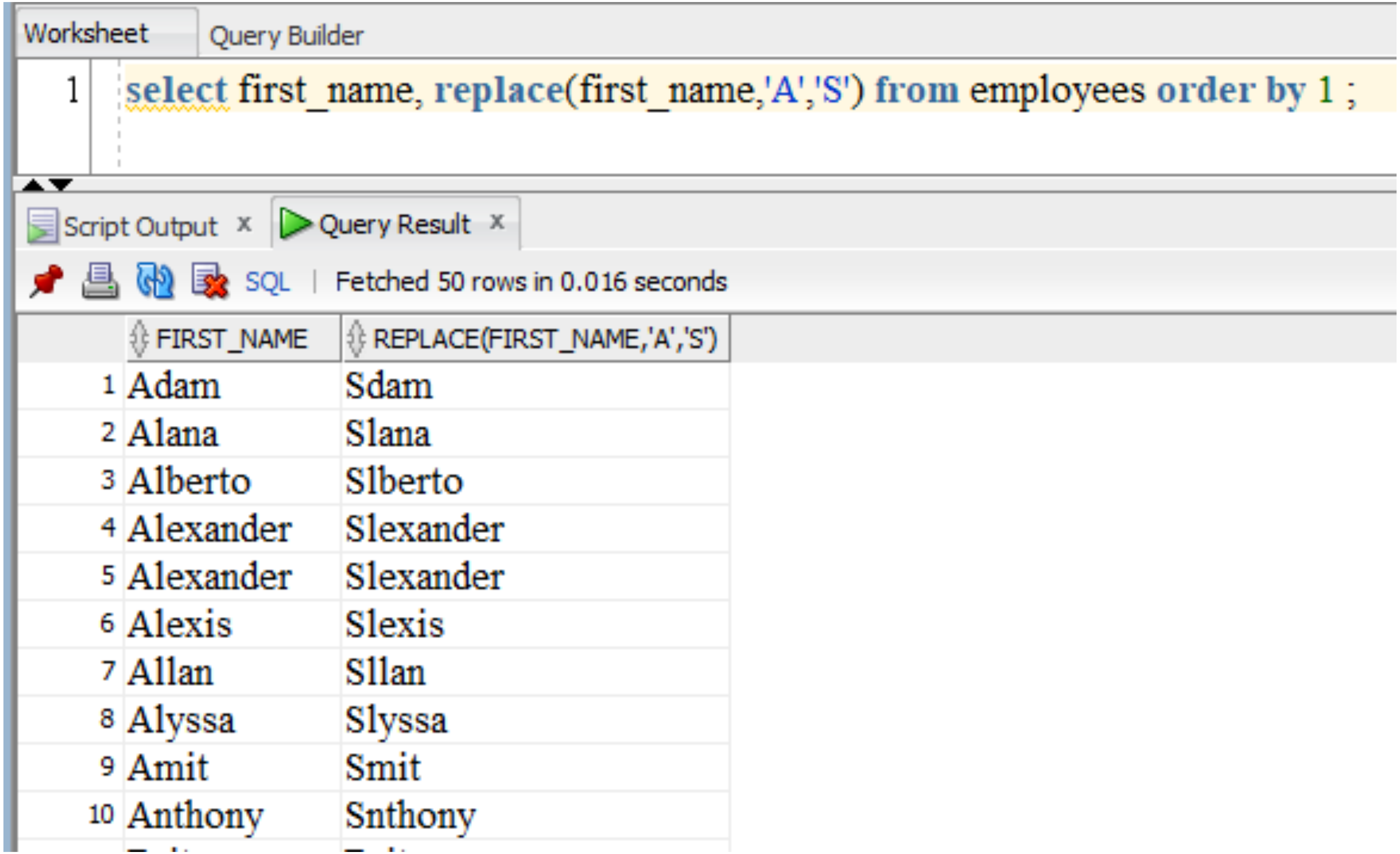


```
select replace('oracle','c','p') from dual;
```

All Rows Fetched: 1 in 0.026 seconds

	REPLACE('ORACLE','C','P')
1	oraple

Here, a–is the old value c to be replaced with p–which is the new value.

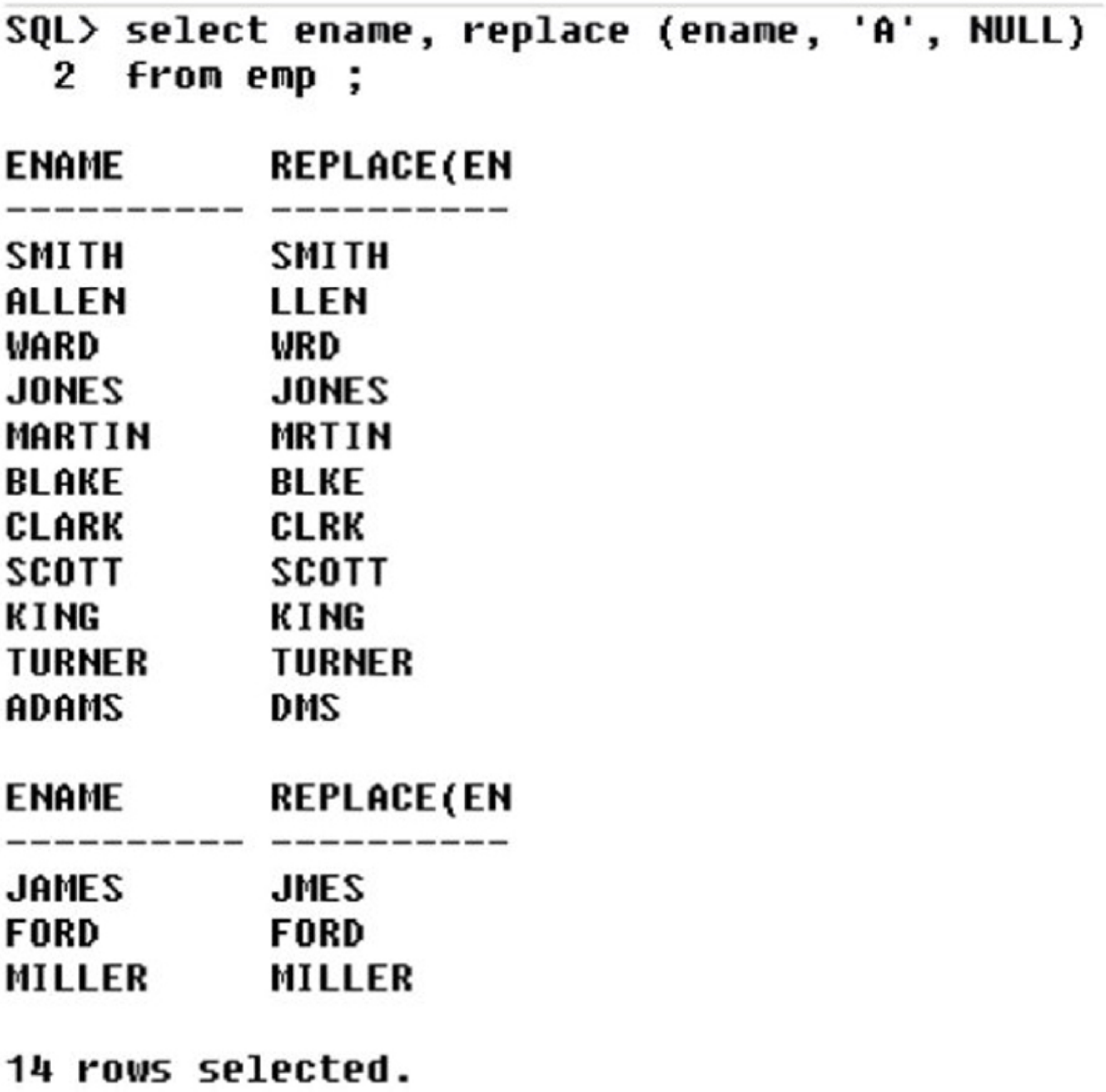


```
select first_name, replace(first_name,'A','S') from employees order by 1 ;
```

Fetched 50 rows in 0.016 seconds

	FIRST_NAME	REPLACE(FIRST_NAME,'A','S')
1	Adam	Sdam
2	Alana	Slana
3	Alberto	Slberto
4	Alexander	Slexander
5	Alexander	Slexander
6	Alexis	Slexis
7	Allan	Sllan
8	Alyssa	Slyssa
9	Amit	Smit
10	Anthony	Snthony

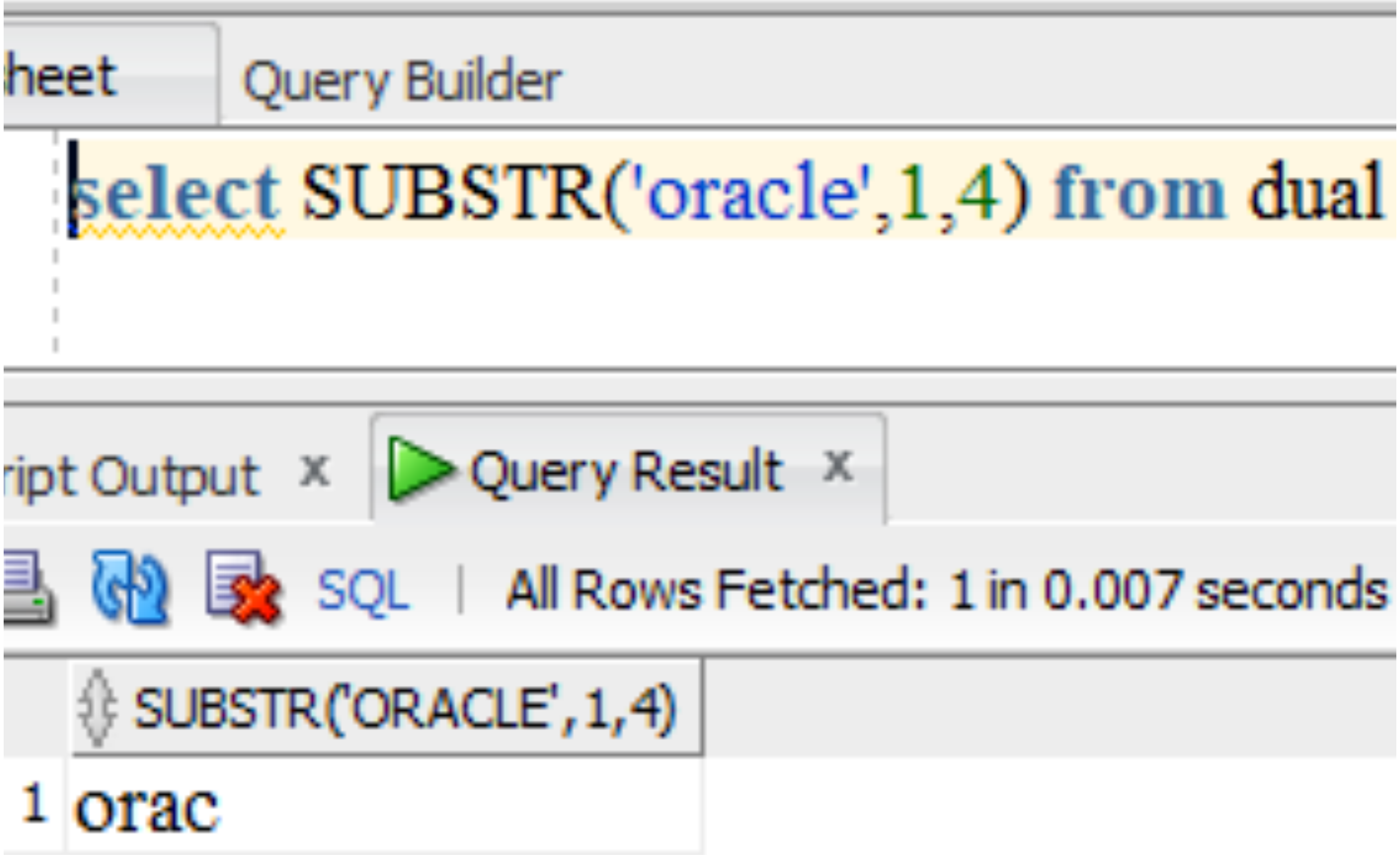
This query replaces all the names which has A in it with S



```
SQL> select ename, replace (ename, 'A', NULL)
  2  from emp ;

ENAME      REPLACE(EN
---------- ----------
SMITH      SMITH
ALLEN      LLEN
WARD       WRD
JONES      JONES
MARTIN     MRTIN
BLAKE      BLKE
CLARK      CLRK
SCOTT      SCOTT
KING       KING
TURNER     TURNER
ADAMS      DMS

ENAME      REPLACE(EN
---------- ----------
JAMES      JMES
FORD       FORD
MILLER     MILLER

14 rows selected.
```

SUBSTR This is called substring.

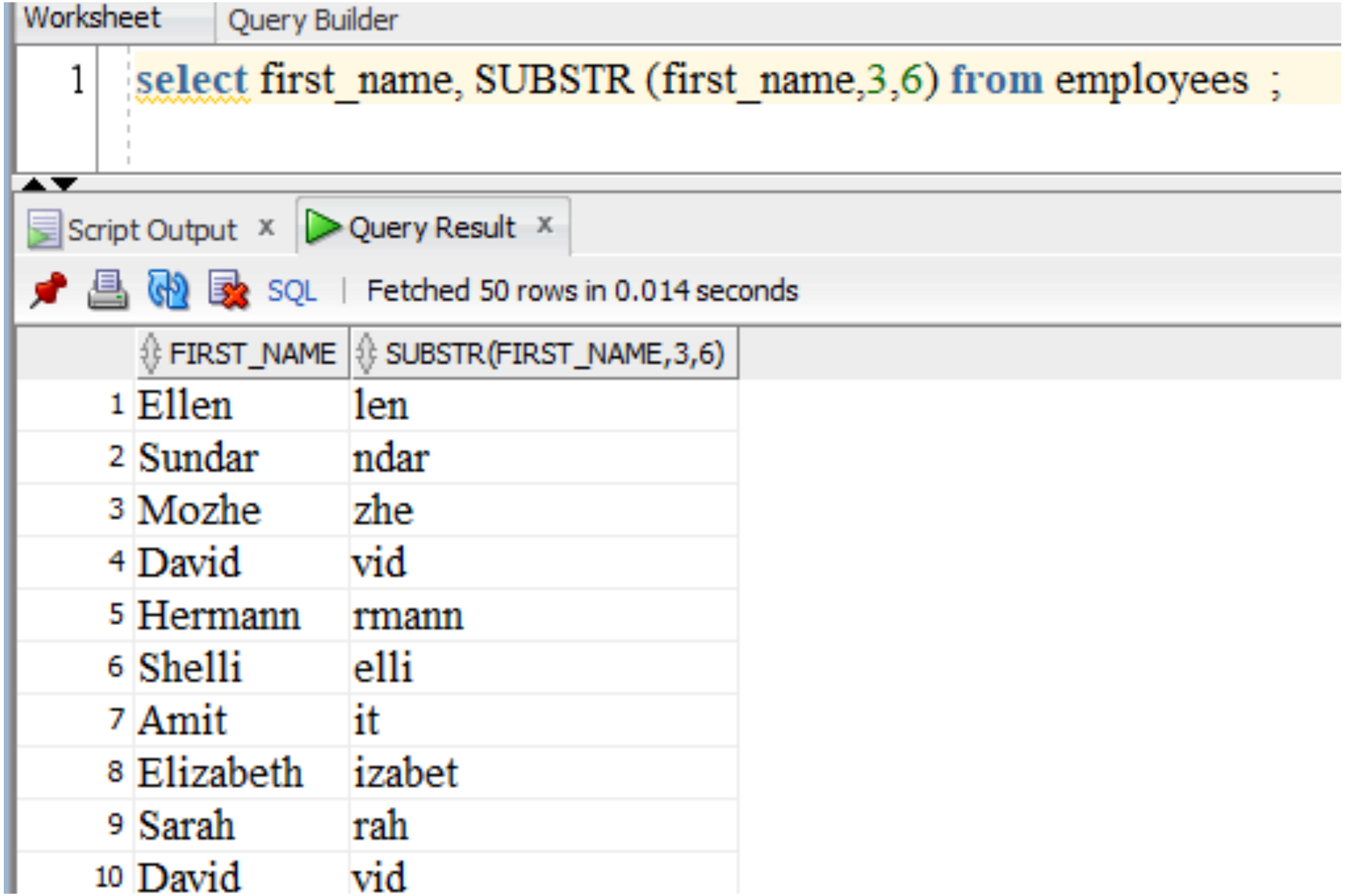
It extracts „n" characters from x(th) position of a given string.



```
select SUBSTR('oracle',1,4) from dual
```

All Rows Fetched: 1 in 0.007 seconds

```
  SUBSTR('ORACLE',1,4)
1 orac
```

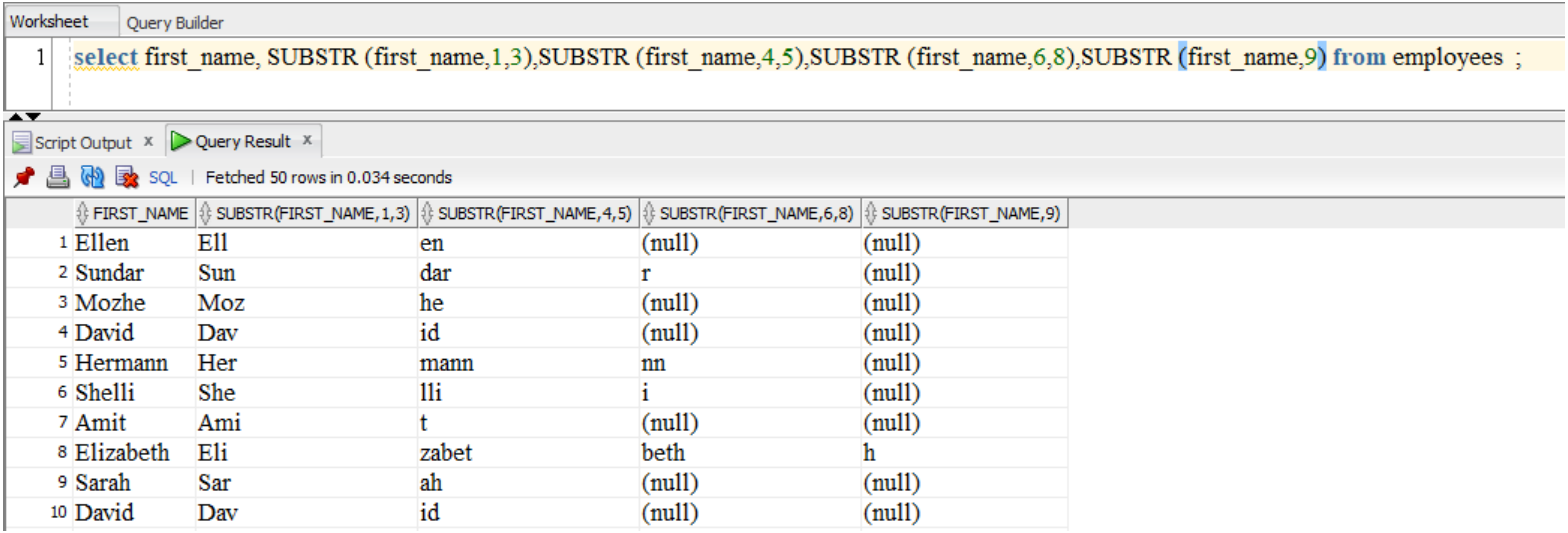


```sql
select first_name, SUBSTR (first_name,3,6) from employees ;
```

Fetched 50 rows in 0.014 seconds

	FIRST_NAME	SUBSTR(FIRST_NAME,3,6)
1	Ellen	len
2	Sundar	ndar
3	Mozhe	zhe
4	David	vid
5	Hermann	rmann
6	Shelli	elli
7	Amit	it
8	Elizabeth	izabet
9	Sarah	rah
10	David	vid

Here , (first name , 3,6“)– means from extract from 3rd position , 6th characters.



```sql
select first_name, SUBSTR (first_name,1,3),SUBSTR (first_name,4,5),SUBSTR (first_name,6,8),SUBSTR (first_name,9) from employees  ;
```

Fetched 50 rows in 0.034 seconds

	FIRST_NAME	SUBSTR(FIRST_NAME,1,3)	SUBSTR(FIRST_NAME,4,5)	SUBSTR(FIRST_NAME,6,8)	SUBSTR(FIRST_NAME,9)
1	Ellen	Ell	en	(null)	(null)
2	Sundar	Sun	dar	r	(null)
3	Mozhe	Moz	he	(null)	(null)
4	David	Dav	id	(null)	(null)
5	Hermann	Her	mann	nn	(null)
6	Shelli	She	lli	i	(null)
7	Amit	Ami	t	(null)	(null)
8	Elizabeth	Eli	zabet	beth	h
9	Sarah	Sar	ah	(null)	(null)
10	David	Dav	id	(null)	(null)

Worksheet
Query Builder
Script Output
Query Result
SQL

```sql
select first_name, SUBSTR (first_name,1,3) "1-3" ,SUBSTR (first_name,4,5) "4-5" ,SUBSTR (first_name,6,8) "6-8" ,SUBSTR (first_name,9) "9-n"from employees  ;
```

Fetched 50 rows in 0.022 seconds

	FIRST_NAME	1-3	4-5	6-8	9-n
1	Ellen	Ell	en	(null)	(null)
2	Sundar	Sun	dar	r	(null)
3	Mozhe	Moz	he	(null)	(null)
4	David	Dav	id	(null)	(null)
5	Hermann	Her	mann	nn	(null)
6	Shelli	She	lli	i	(null)
7	Amit	Ami	t	(null)	(null)
8	Elizabeth	Eli	zabet	beth	h
9	Sarah	Sar	ah	(null)	(null)
10	David	Dav	id	(null)	(null)

Display the employees whose jobid starts with „AD"

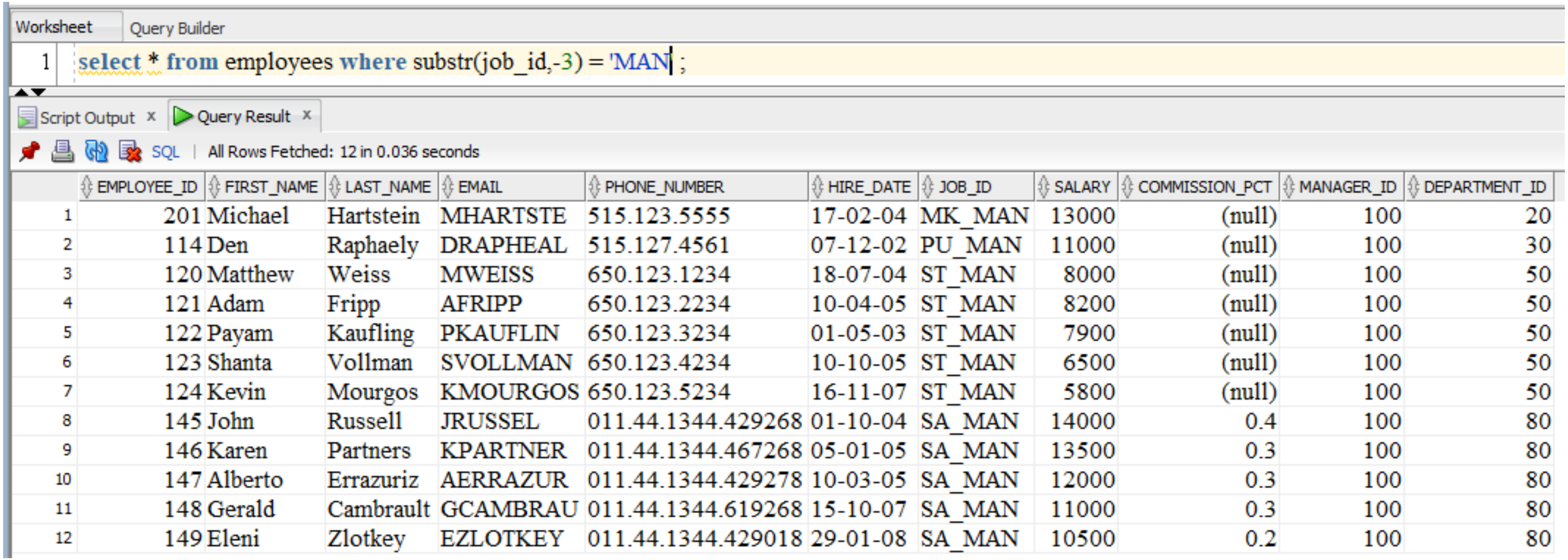
Worksheet
Query Builder
Script Output
Query Result
SQL

```
select * from employees where substr(job_id,1,2) = 'AD' ;
```

All Rows Fetched: 4 in 0.03 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-09-03	AD_ASST	4400	(null)	101	10
2	100	Steven	King	SKING	515.123.4567	17-06-03	AD_PRES	24000	(null)	(null)	90
3	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-09-05	AD_VP	17000	(null)	100	90
4	102	Lex	De Haan	LDEHAAN	515.123.4569	13-01-01	AD_VP	17000	(null)	100	90

Display the employees whose jobid ends with „man“



```
select * from employees where substr(job_id,-3) = 'MAN' ;
```

All Rows Fetched: 12 in 0.036 seconds

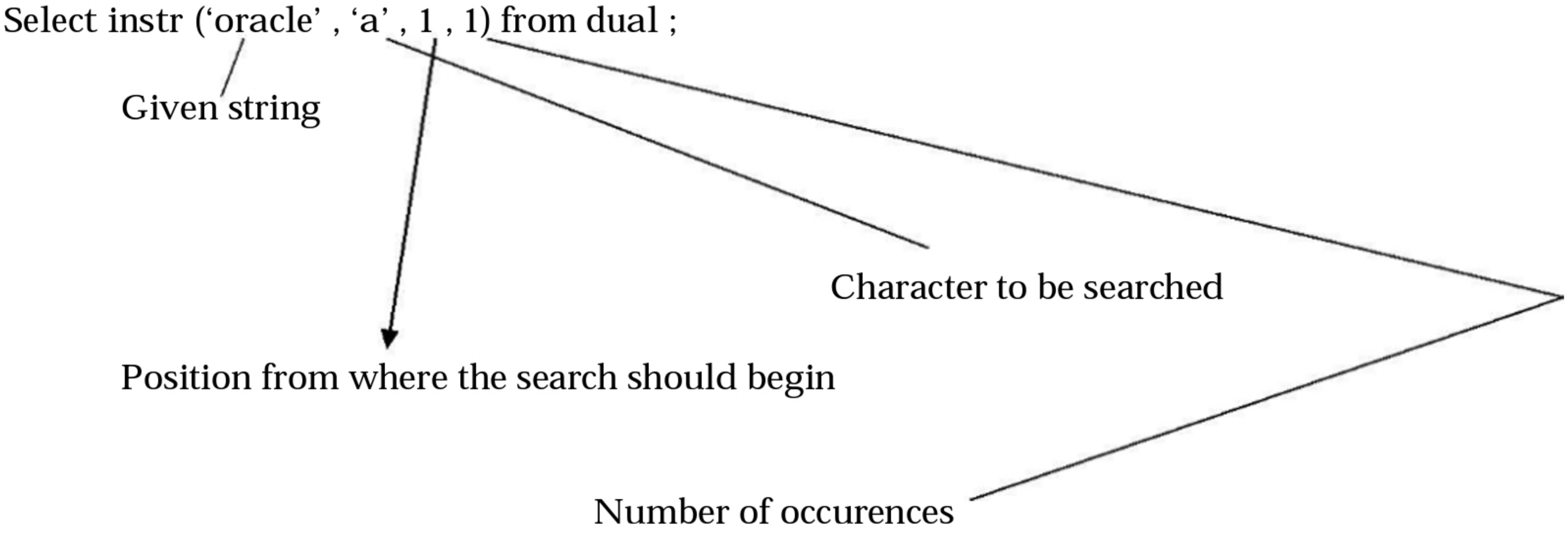
	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-02-04	MK_MAN	13000	(null)	100	20
2	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-12-02	PU_MAN	11000	(null)	100	30
3	120	Matthew	Weiss	MWEISS	650.123.1234	18-07-04	ST_MAN	8000	(null)	100	50
4	121	Adam	Fripp	AFRIPP	650.123.2234	10-04-05	ST_MAN	8200	(null)	100	50
5	122	Payam	Kaufling	PKAUFLIN	650.123.3234	01-05-03	ST_MAN	7900	(null)	100	50
6	123	Shanta	Vollman	SVOLLMAN	650.123.4234	10-10-05	ST_MAN	6500	(null)	100	50
7	124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-11-07	ST_MAN	5800	(null)	100	50
8	145	John	Russell	JRUSSEL	011.44.1344.429268	01-10-04	SA_MAN	14000	0.4	100	80
9	146	Karen	Partners	KPARTNER	011.44.1344.467268	05-01-05	SA_MAN	13500	0.3	100	80
10	147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-03-05	SA_MAN	12000	0.3	100	80
11	148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	15-10-07	SA_MAN	11000	0.3	100	80
12	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-01-08	SA_MAN	10500	0.2	100	80

INSTR

This is also called as instring.

It returns position of a given character in a given string.

For ex,



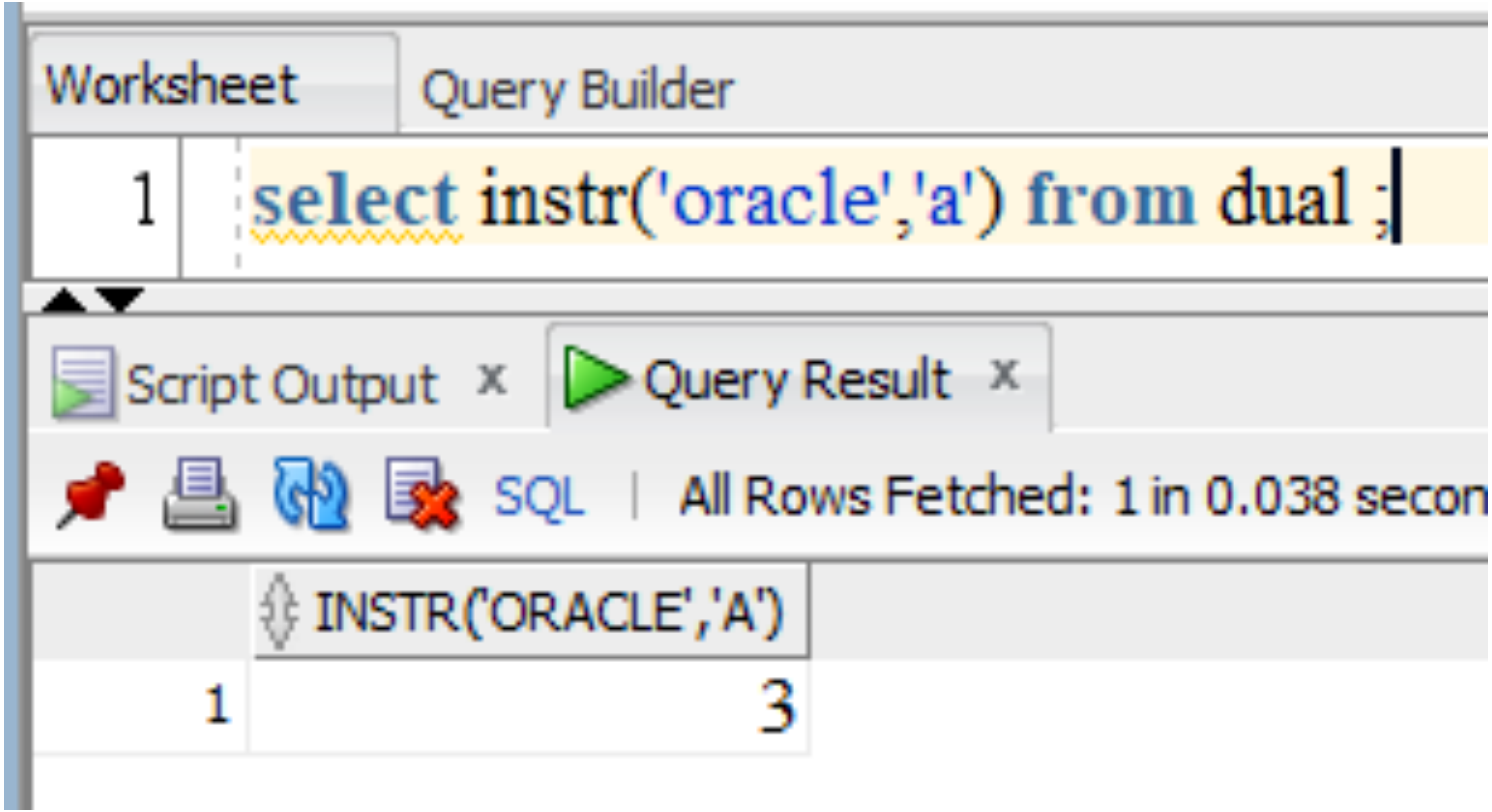
Select instr (‘oracle’ , ‘a’ , 1 , 1) from dual ;

Given string

Character to be searched

Position from where the search should begin

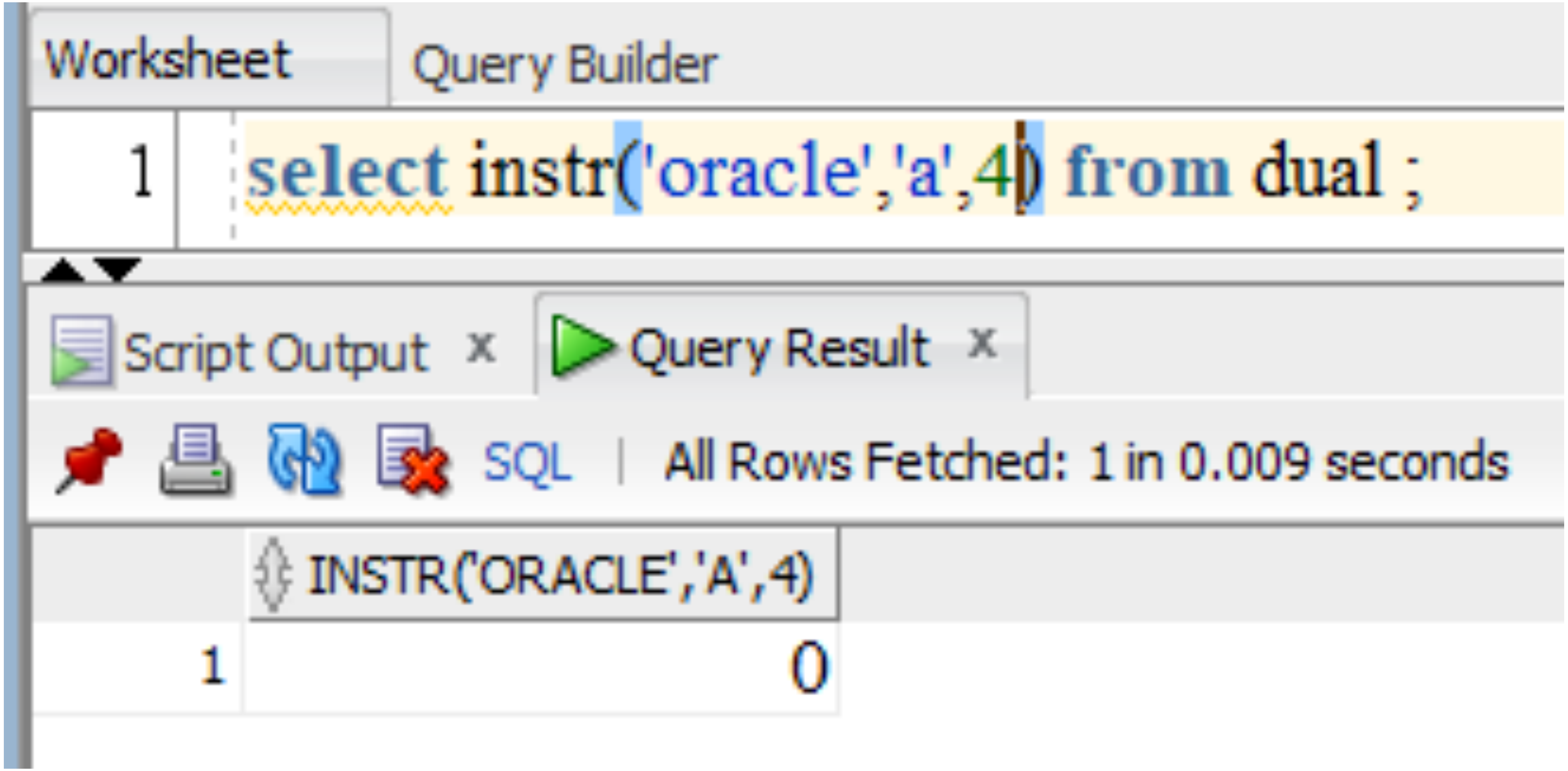
Number of occurences



```
1   select instr('oracle','a') from dual ;
```

All Rows Fetched: 1 in 0.038 secon

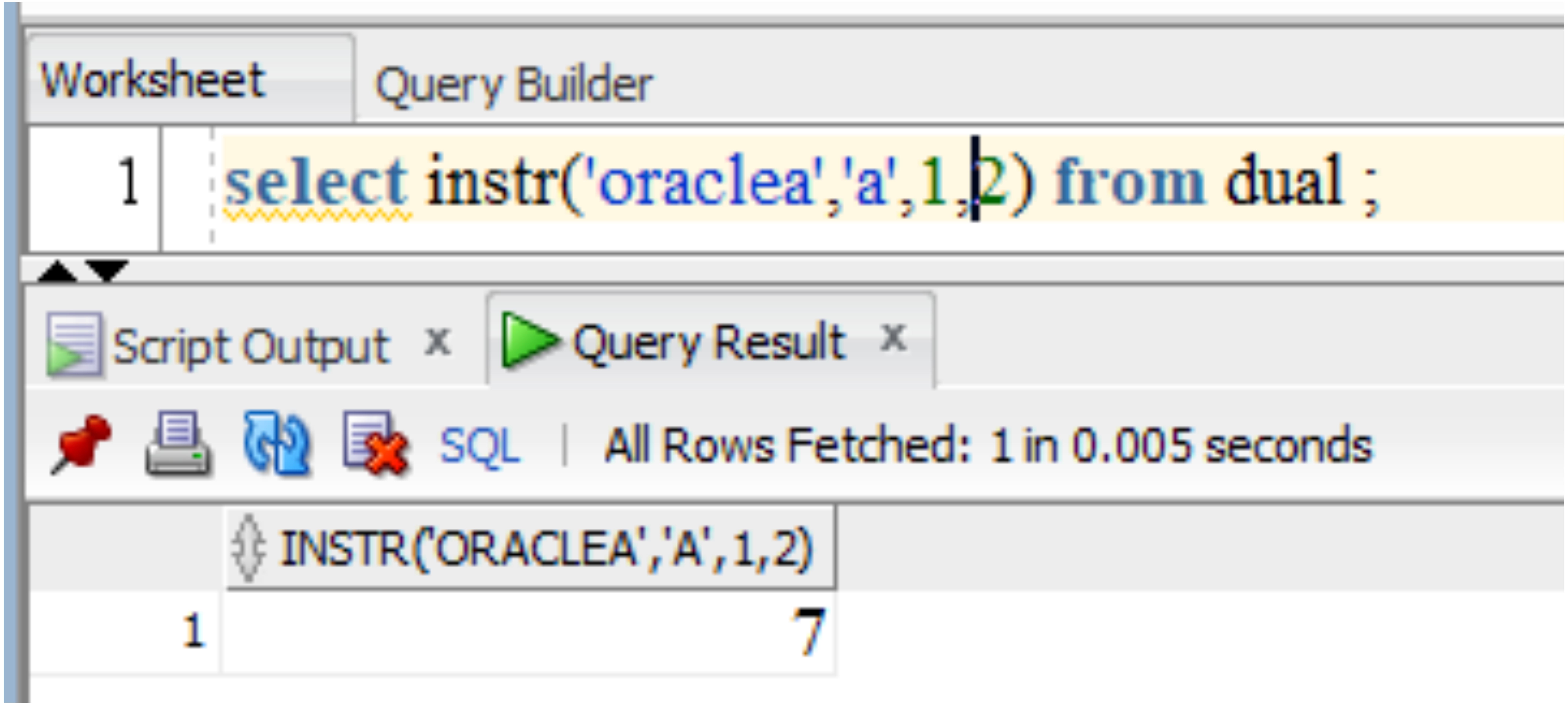
	INSTR('ORACLE','A')
1	3



```
select instr('oracle','a',4) from dual ;
```

All Rows Fetched: 1 in 0.009 seconds

	INSTR('ORACLE','A',4)
1	0

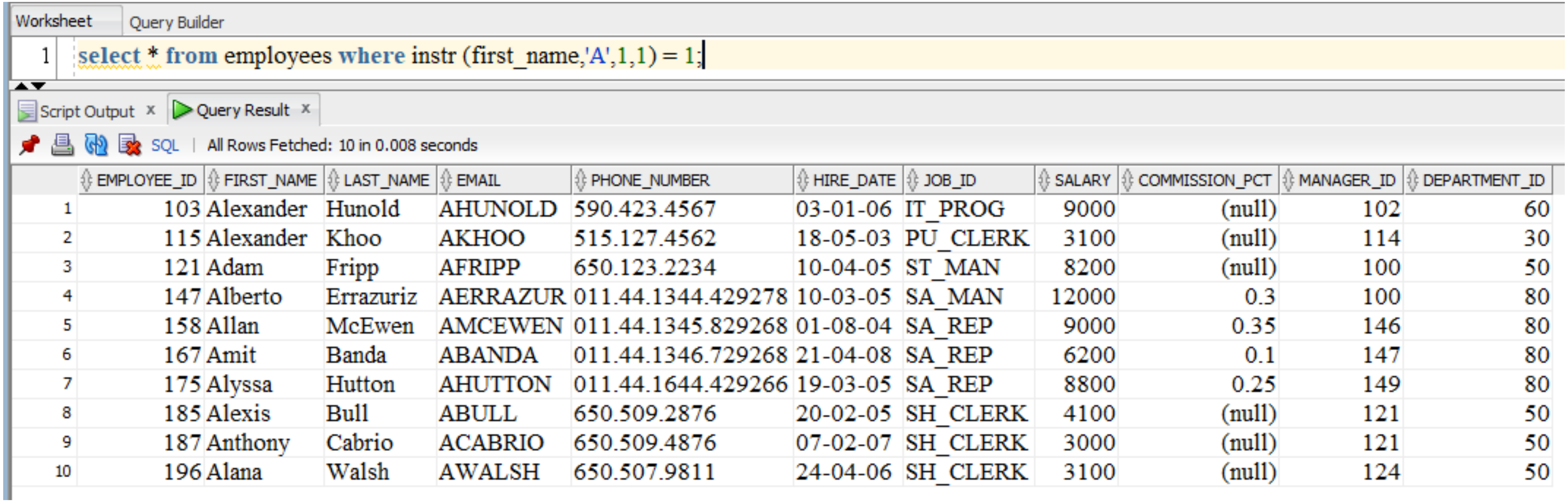


```
select instr('oraclea','a',1,2) from dual ;
```

All Rows Fetched: 1 in 0.005 seconds

	INSTR('ORACLEA','A',1,2)
1	7

Display all the employees whose name is having one „A“ in first position

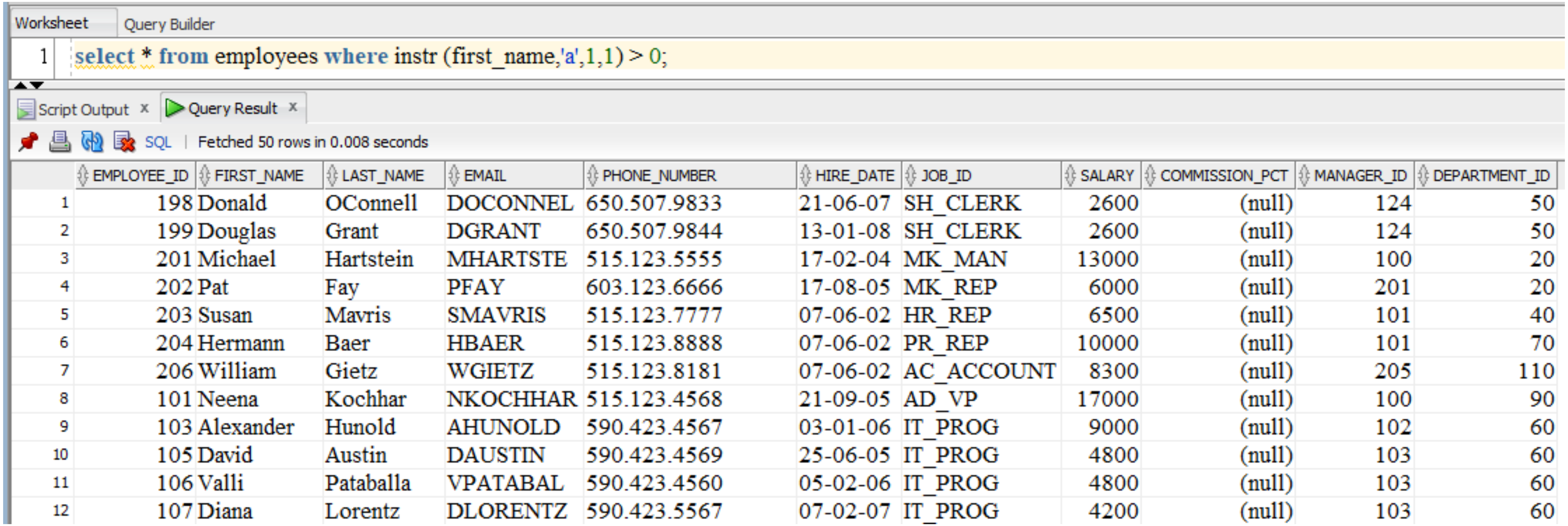


```
select * from employees where instr (first_name,'A',1,1) = 1;
```

All Rows Fetched: 10 in 0.008 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-01-06	IT_PROG	9000	(null)	102	60
2	115	Alexander	Khoo	AKHOO	515.127.4562	18-05-03	PU_CLERK	3100	(null)	114	30
3	121	Adam	Fripp	AFRIPP	650.123.2234	10-04-05	ST_MAN	8200	(null)	100	50
4	147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-03-05	SA_MAN	12000	0.3	100	80
5	158	Allan	McEwen	AMCEWEN	011.44.1345.829268	01-08-04	SA_REP	9000	0.35	146	80
6	167	Amit	Banda	ABANDA	011.44.1346.729268	21-04-08	SA_REP	6200	0.1	147	80
7	175	Alyssa	Hutton	AHUTTON	011.44.1644.429266	19-03-05	SA_REP	8800	0.25	149	80
8	185	Alexis	Bull	ABULL	650.509.2876	20-02-05	SH_CLERK	4100	(null)	121	50
9	187	Anthony	Cabrio	ACABRIO	650.509.4876	07-02-07	SH_CLERK	3000	(null)	121	50
10	196	Alana	Walsh	AWALSH	650.507.9811	24-04-06	SH_CLERK	3100	(null)	124	50

Display all the employees whose name is having „a‟ in any position

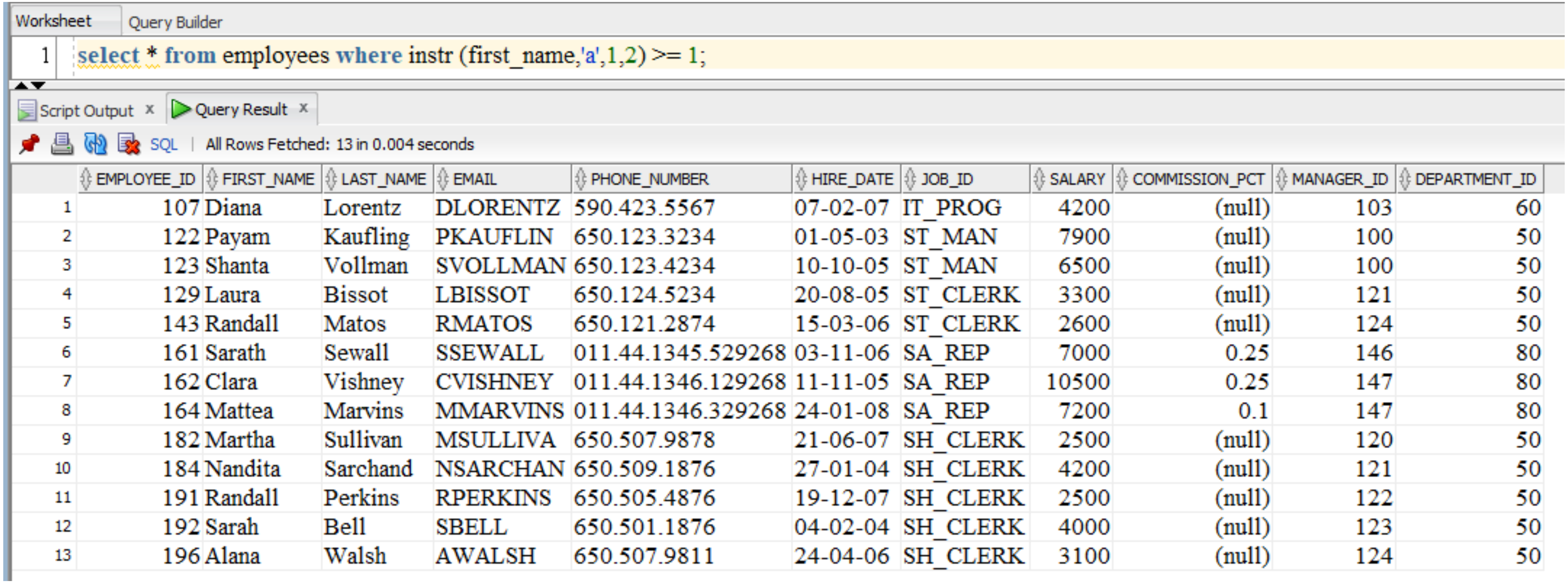


```
select * from employees where instr (first_name,'a',1,1) > 0;
```

Fetched 50 rows in 0.008 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	198	Donald	OConnell	DOCONNEL	650.507.9833	21-06-07	SH_CLERK	2600	(null)	124	50
2	199	Douglas	Grant	DGRANT	650.507.9844	13-01-08	SH_CLERK	2600	(null)	124	50
3	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-02-04	MK_MAN	13000	(null)	100	20
4	202	Pat	Fay	PFAY	603.123.6666	17-08-05	MK_REP	6000	(null)	201	20
5	203	Susan	Mavris	SMAVRIS	515.123.7777	07-06-02	HR_REP	6500	(null)	101	40
6	204	Hermann	Baer	HBAER	515.123.8888	07-06-02	PR_REP	10000	(null)	101	70
7	206	William	Gietz	WGIETZ	515.123.8181	07-06-02	AC_ACCOUNT	8300	(null)	205	110
8	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-09-05	AD_VP	17000	(null)	100	90
9	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-01-06	IT_PROG	9000	(null)	102	60
10	105	David	Austin	DAUSTIN	590.423.4569	25-06-05	IT_PROG	4800	(null)	103	60
11	106	Valli	Pataballa	VPATABAL	590.423.4560	05-02-06	IT_PROG	4800	(null)	103	60
12	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-02-07	IT_PROG	4200	(null)	103	60

Display all the employees whose name is having at-least two „a‟ and 2nd a from 1st position onwards

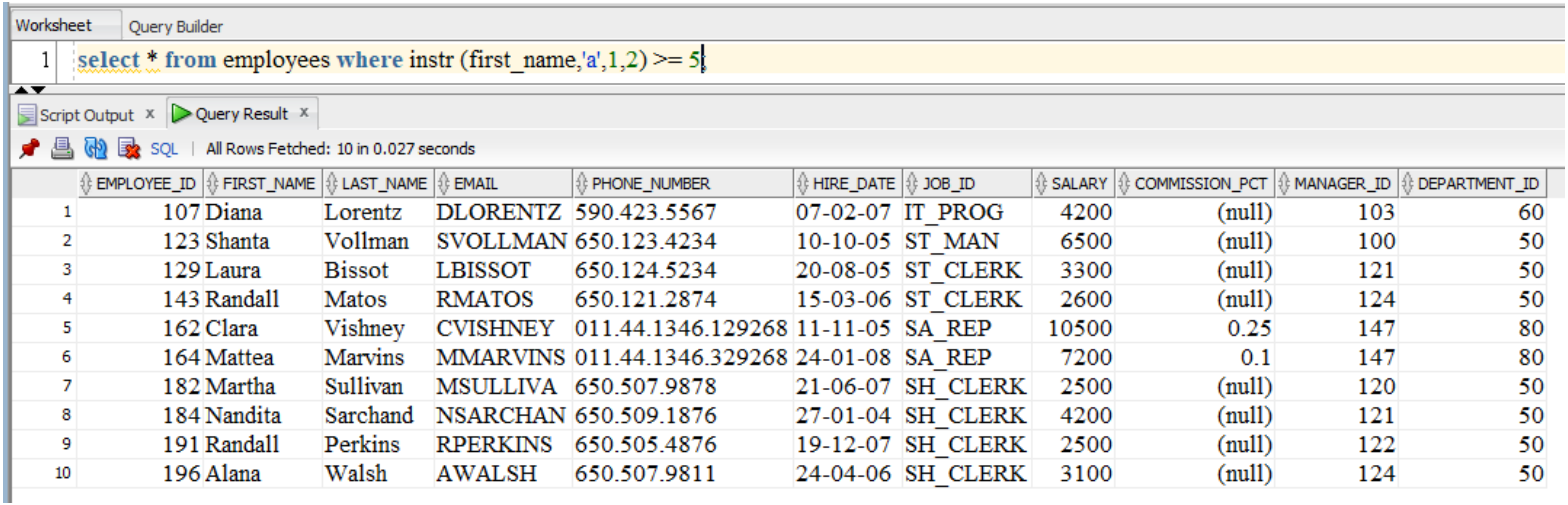


```
select * from employees where instr (first_name,'a',1,2) >= 1;
```

All Rows Fetched: 13 in 0.004 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-02-07	IT_PROG	4200	(null)	103	60
2	122	Payam	Kaufling	PKAUFLIN	650.123.3234	01-05-03	ST_MAN	7900	(null)	100	50
3	123	Shanta	Vollman	SVOLLMAN	650.123.4234	10-10-05	ST_MAN	6500	(null)	100	50
4	129	Laura	Bissot	LBISSOT	650.124.5234	20-08-05	ST_CLERK	3300	(null)	121	50
5	143	Randall	Matos	RMATOS	650.121.2874	15-03-06	ST_CLERK	2600	(null)	124	50
6	161	Sarath	Sewall	SSEWALL	011.44.1345.529268	03-11-06	SA_REP	7000	0.25	146	80
7	162	Clara	Vishney	CVISHNEY	011.44.1346.129268	11-11-05	SA_REP	10500	0.25	147	80
8	164	Mattea	Marvins	MMARVINS	011.44.1346.329268	24-01-08	SA_REP	7200	0.1	147	80
9	182	Martha	Sullivan	MSULLIVA	650.507.9878	21-06-07	SH_CLERK	2500	(null)	120	50
10	184	Nandita	Sarchand	NSARCHAN	650.509.1876	27-01-04	SH_CLERK	4200	(null)	121	50
11	191	Randall	Perkins	RPERKINS	650.505.4876	19-12-07	SH_CLERK	2500	(null)	122	50
12	192	Sarah	Bell	SBELL	650.501.1876	04-02-04	SH_CLERK	4000	(null)	123	50
13	196	Alana	Walsh	AWALSH	650.507.9811	24-04-06	SH_CLERK	3100	(null)	124	50

Display all the employees whose name is having at-least two „a‟ and 2nd a from 5th position onwards

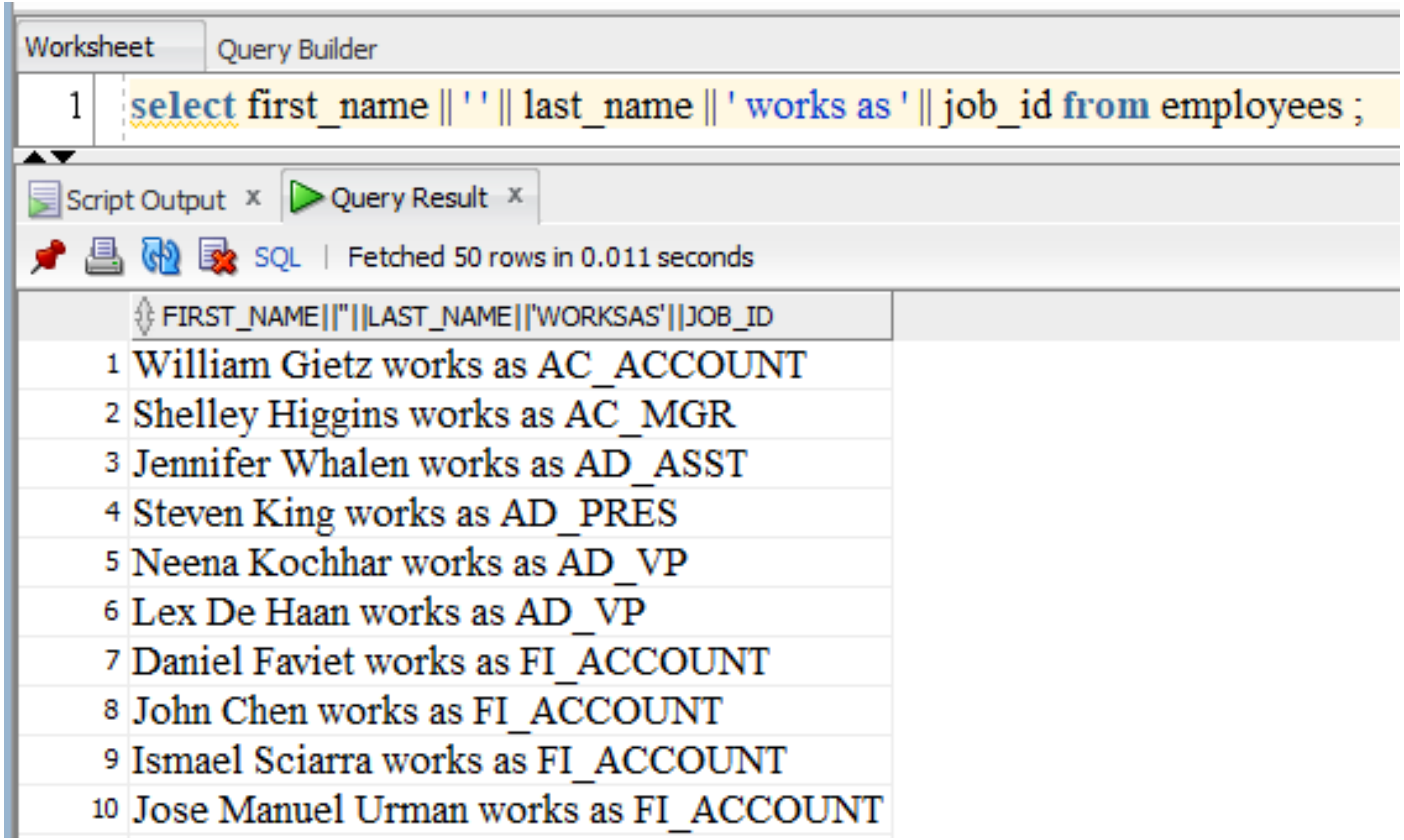


```
select * from employees where instr (first_name,'a',1,2) >= 5
```

All Rows Fetched: 10 in 0.027 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-02-07	IT_PROG	4200	(null)	103	60
2	123	Shanta	Vollman	SVOLLMAN	650.123.4234	10-10-05	ST_MAN	6500	(null)	100	50
3	129	Laura	Bissot	LBISSOT	650.124.5234	20-08-05	ST_CLERK	3300	(null)	121	50
4	143	Randall	Matos	RMATOS	650.121.2874	15-03-06	ST_CLERK	2600	(null)	124	50
5	162	Clara	Vishney	CVISHNEY	011.44.1346.129268	11-11-05	SA_REP	10500	0.25	147	80
6	164	Mattea	Marvins	MMARVINS	011.44.1346.329268	24-01-08	SA_REP	7200	0.1	147	80
7	182	Martha	Sullivan	MSULLIVA	650.507.9878	21-06-07	SH_CLERK	2500	(null)	120	50
8	184	Nandita	Sarchand	NSARCHAN	650.509.1876	27-01-04	SH_CLERK	4200	(null)	121	50
9	191	Randall	Perkins	RPERKINS	650.505.4876	19-12-07	SH_CLERK	2500	(null)	122	50
10	196	Alana	Walsh	AWALSH	650.507.9811	24-04-06	SH_CLERK	3100	(null)	124	50

CONCAT It concatenates any two values or columns. It is represented by -||

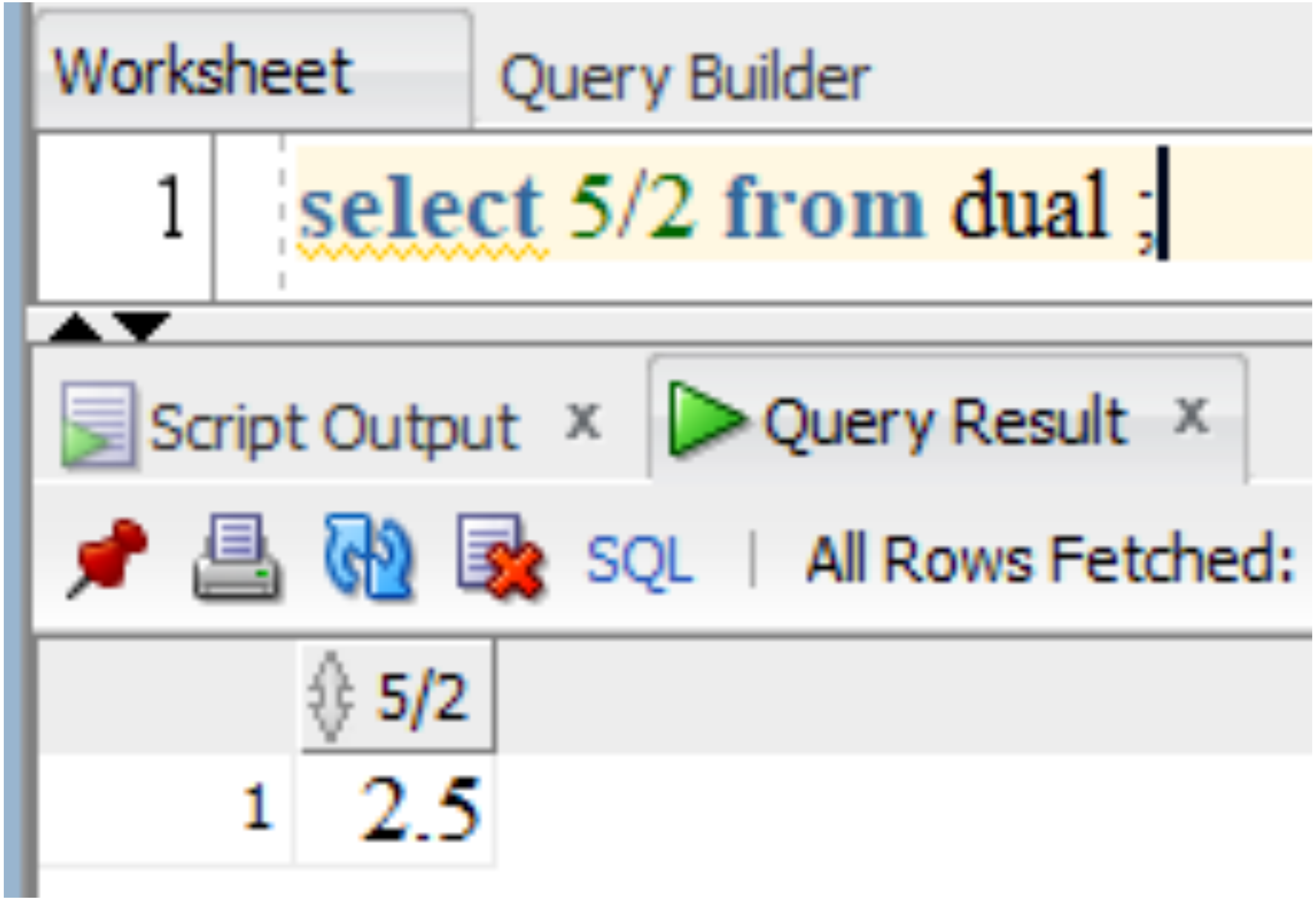


```sql
select first_name || ' ' || last_name || ' works as ' || job_id from employees ;
```

Fetched 50 rows in 0.011 seconds

	FIRST_NAME||''||LAST_NAME||'WORKSAS'||JOB_ID
1	William Gietz works as AC_ACCOUNT
2	Shelley Higgins works as AC_MGR
3	Jennifer Whalen works as AD_ASST
4	Steven King works as AD_PRES
5	Neena Kochhar works as AD_VP
6	Lex De Haan works as AD_VP
7	Daniel Faviet works as FI_ACCOUNT
8	John Chen works as FI_ACCOUNT
9	Ismael Sciarra works as FI_ACCOUNT
10	Jose Manuel Urman works as FI_ACCOUNT

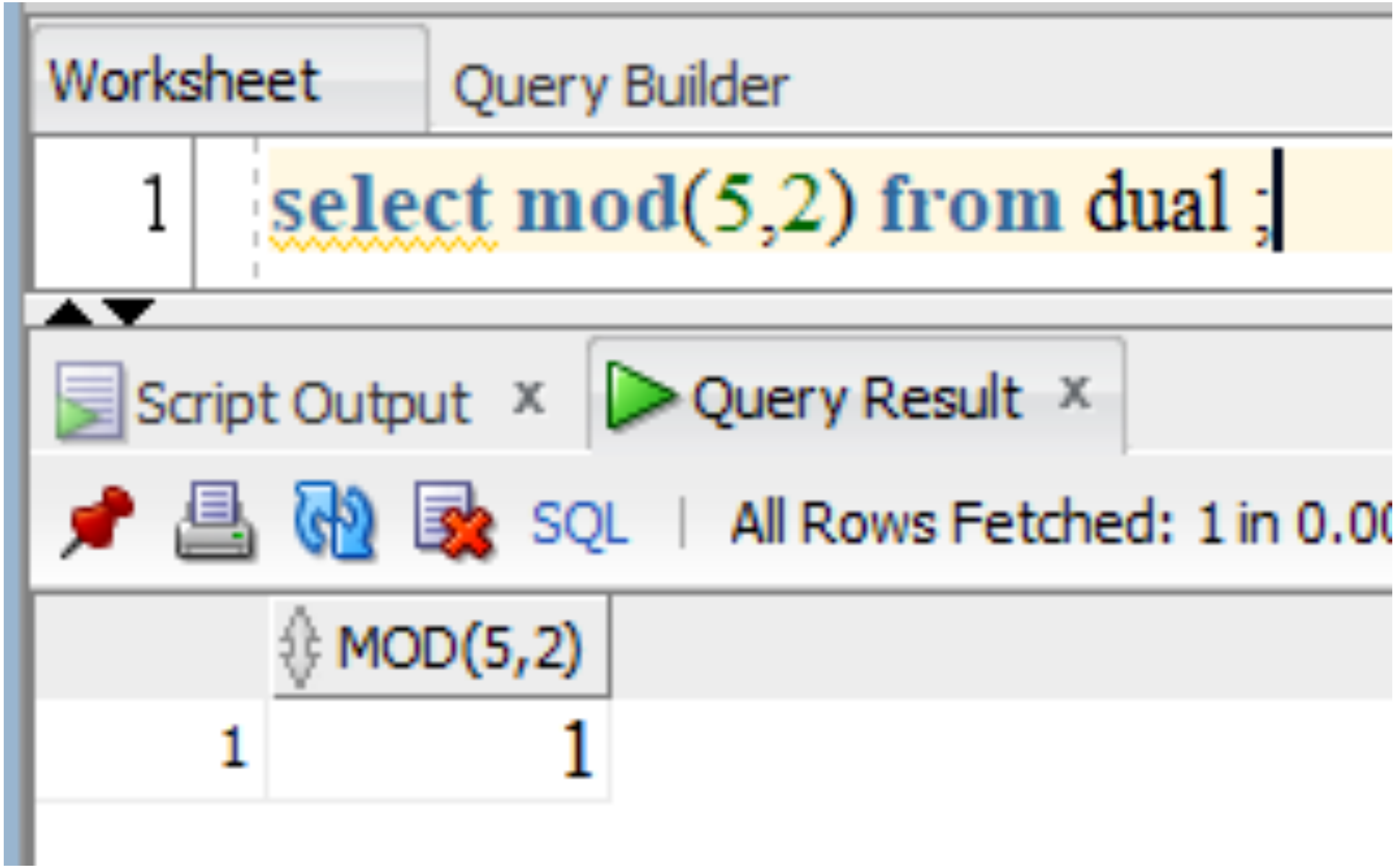
NUMERIC FUNCTIONS



```
select 5/2 from dual ;
```

```
   5/2
1  2.5
```

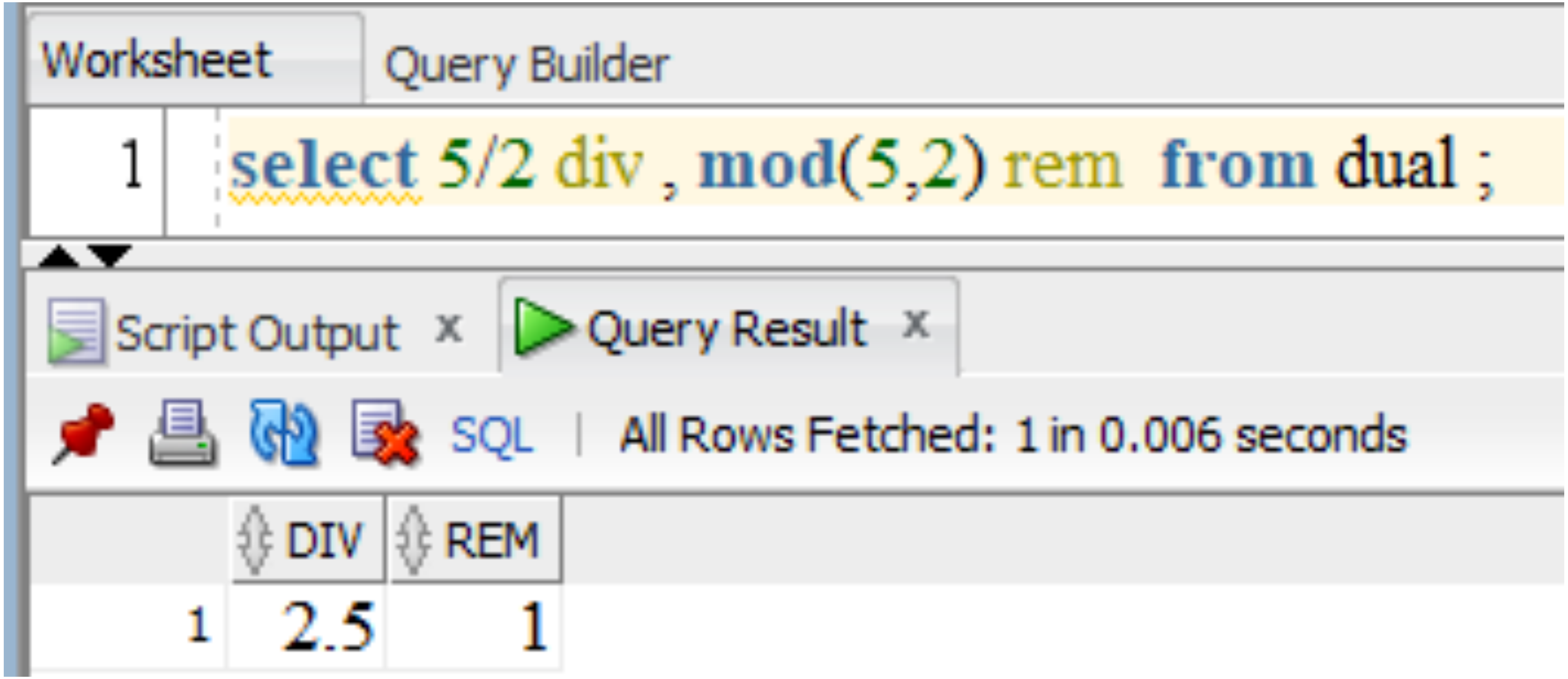
1) Mod:-it returns the remainder when 1 number is divided by the other.



```
select mod(5,2) from dual ;
```

All Rows Fetched: 1 in 0.00

	MOD(5,2)
1	1



```
select 5/2 div , mod(5,2) rem  from dual ;
```

All Rows Fetched: 1 in 0.006 seconds

	DIV	REM
1	2.5	1

Display the employees earning even numbered salaries.

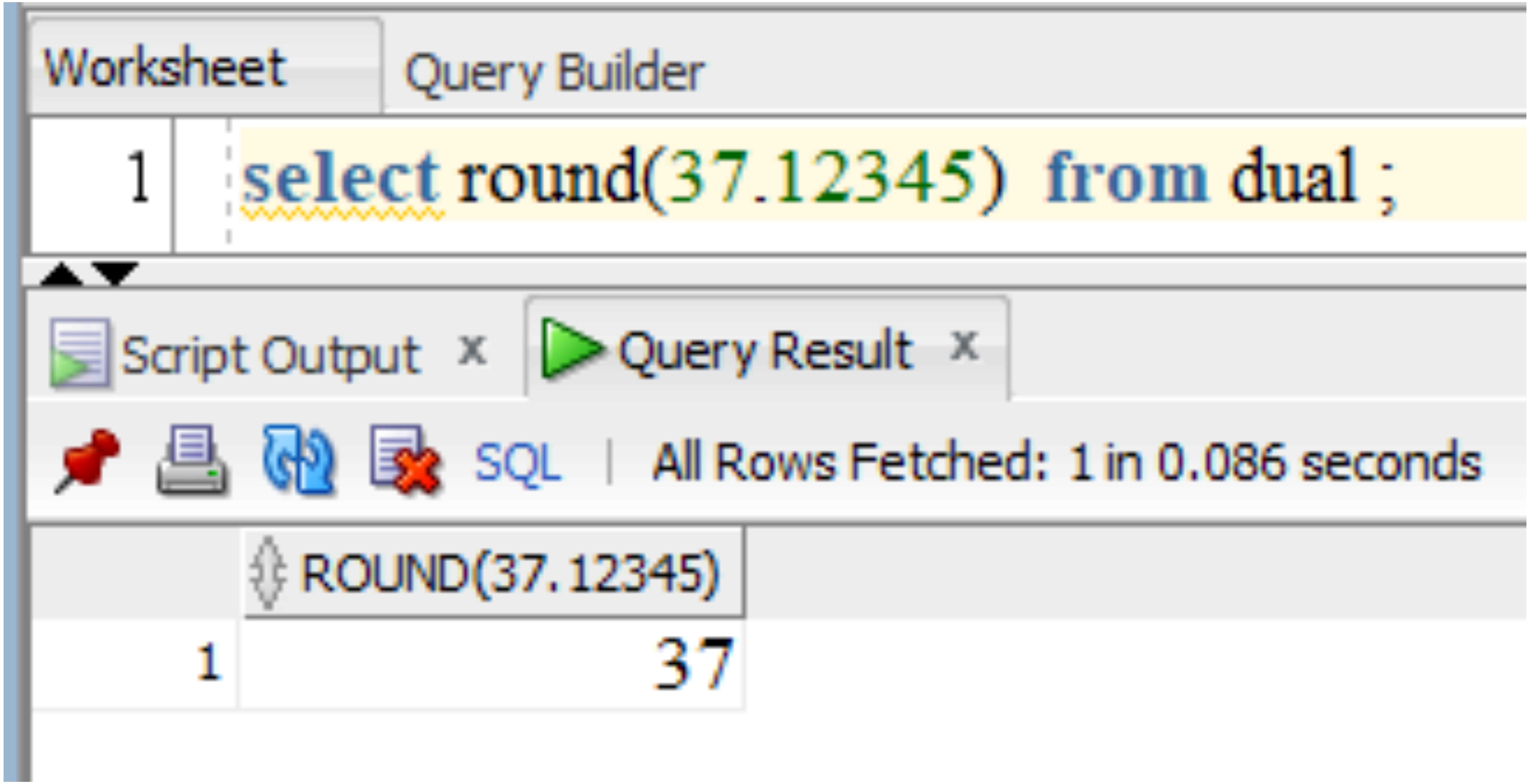
Worksheet
Query Builder
Script Output
Query Result
SQL

```sql
select *  from employees where mod(salary,2)=0 ;
```

All Rows Fetched: 107 in 0.057 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	198	Donald	OConnell	DOCONNEL	650.507.9833	21-06-07	SH_CLERK	2600	(null)	124	50
2	199	Douglas	Grant	DGRANT	650.507.9844	13-01-08	SH_CLERK	2600	(null)	124	50
3	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-09-03	AD_ASST	4400	(null)	101	10
4	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-02-04	MK_MAN	13000	(null)	100	20
5	202	Pat	Fay	PFAY	603.123.6666	17-08-05	MK_REP	6000	(null)	201	20
6	203	Susan	Mavris	SMAVRIS	515.123.7777	07-06-02	HR_REP	6500	(null)	101	40
7	204	Hermann	Baer	HBAER	515.123.8888	07-06-02	PR_REP	10000	(null)	101	70
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-06-02	AC_MGR	12008	(null)	101	110
9	206	William	Gietz	WGIETZ	515.123.8181	07-06-02	AC_ACCOUNT	8300	(null)	205	110
10	100	Steven	King	SKING	515.123.4567	17-06-03	AD_PRES	24000	(null)	(null)	90

Round It rounds off a given number to the nearest decimal place.

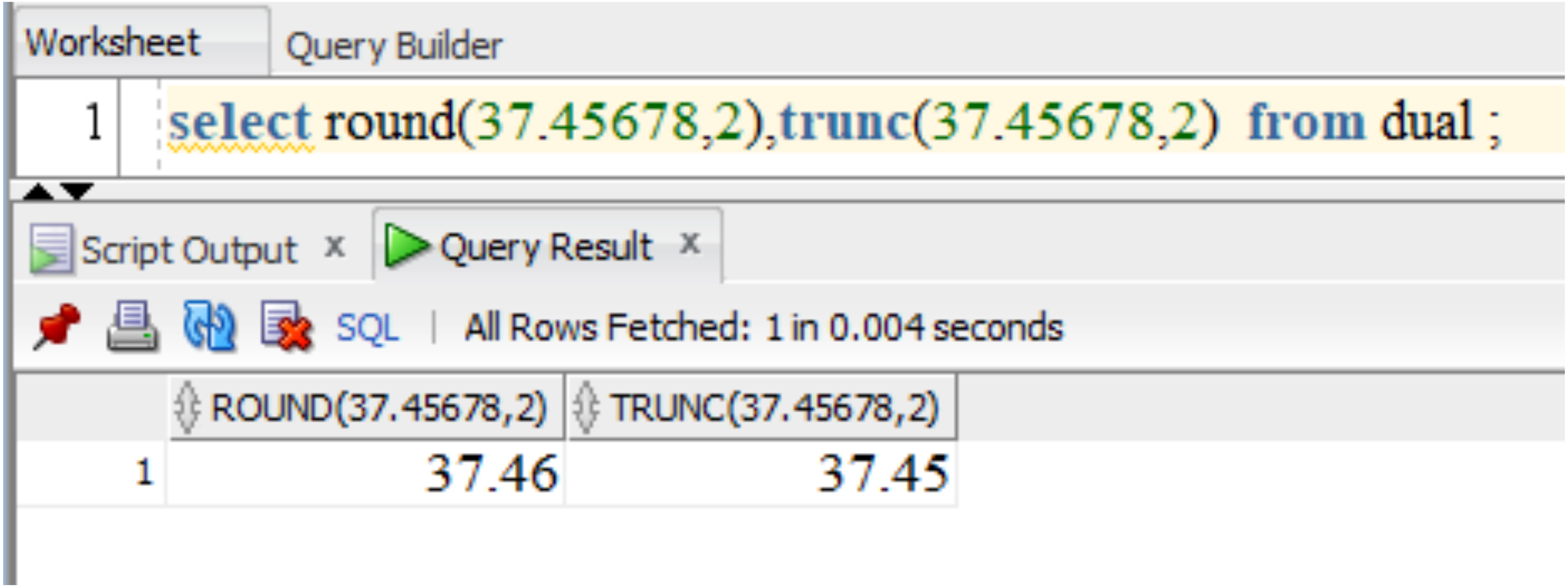


```
select round(37.12345)  from dual ;
```

All Rows Fetched: 1 in 0.086 seconds

	ROUND(37.12345)
1	37

Trunc It truncates the given number to the given decimal place. Truncate does not do any rounding.

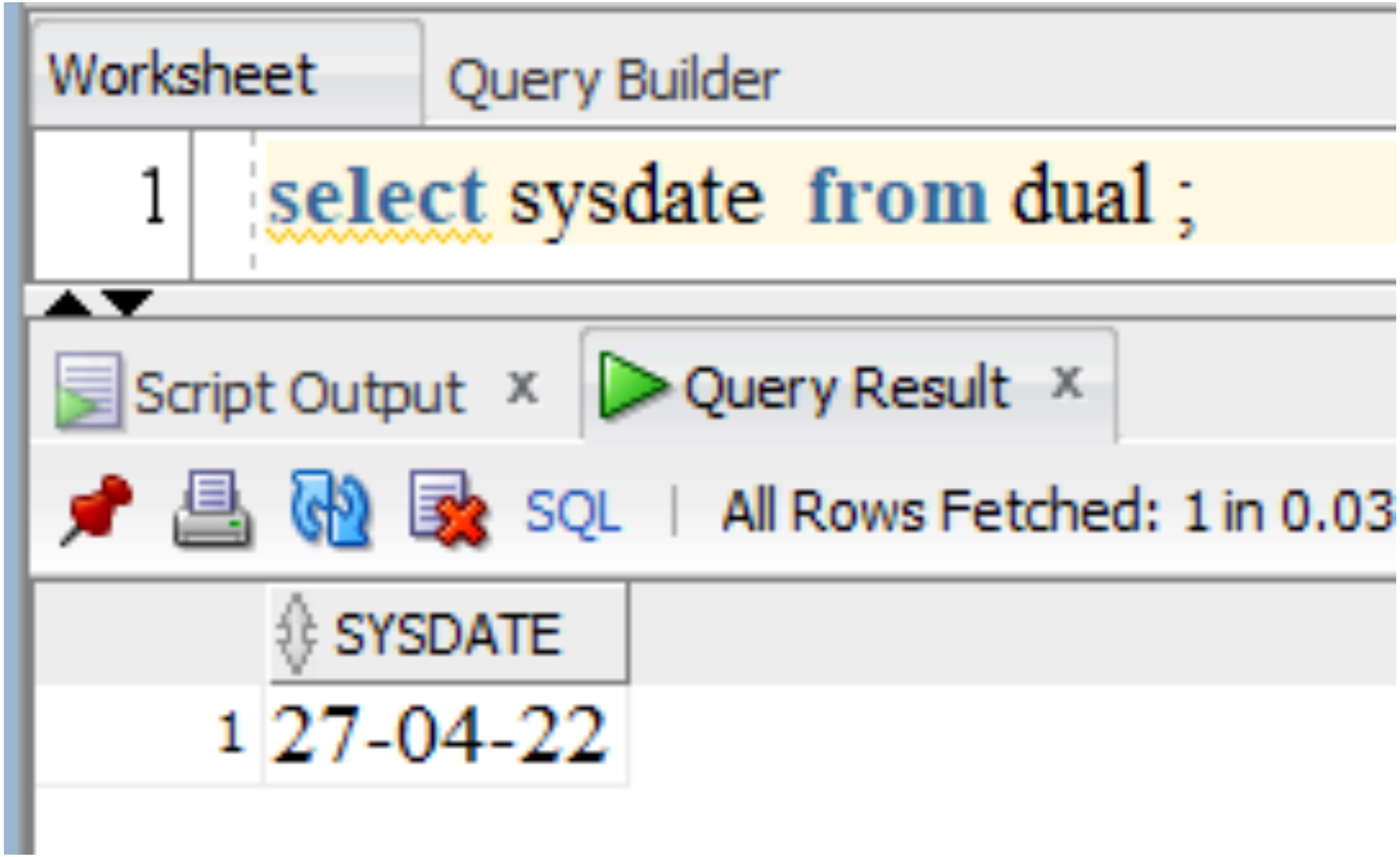


```
select round(37.45678,2),trunc(37.45678,2)  from dual ;
```

All Rows Fetched: 1 in 0.004 seconds

	ROUND(37.45678,2)	TRUNC(37.45678,2)
1	37.46	37.45

DATE FUNCTIONS 1) Sysdate Stands for System date. It returns both date & time, but by default – only date is displayed. The default format is, dd – mon – yy

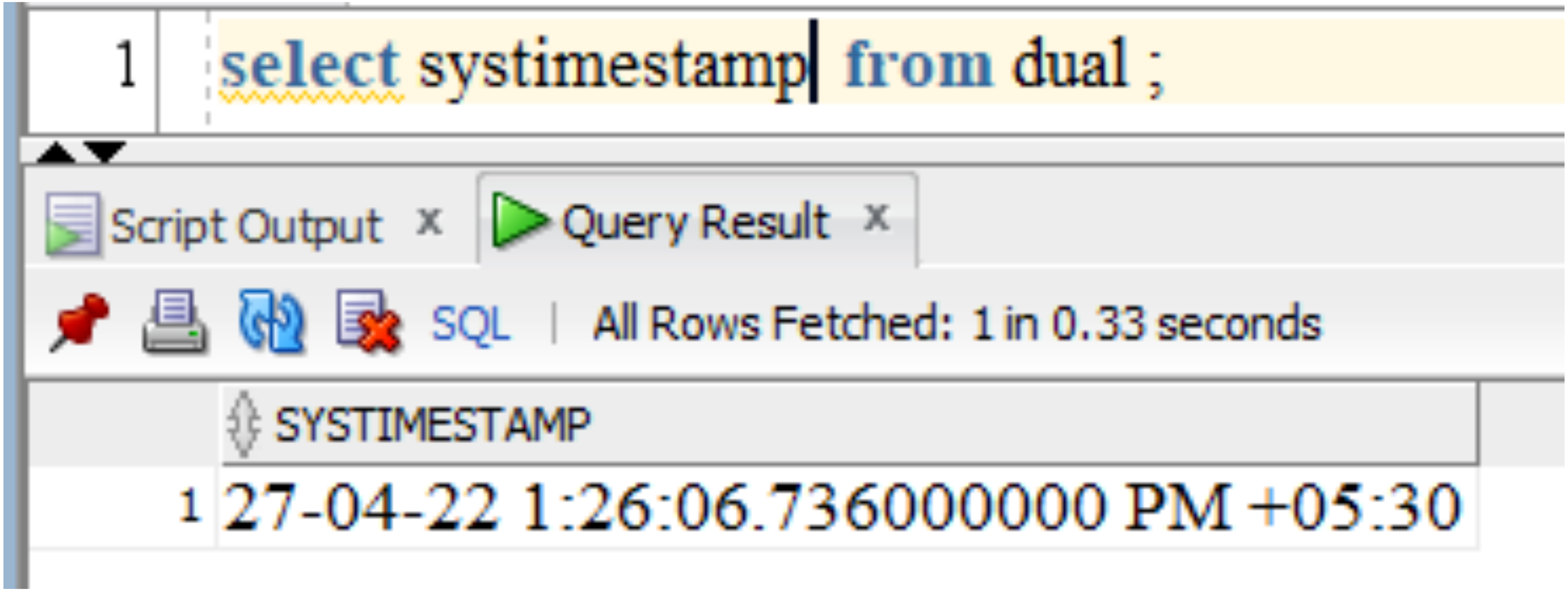


```
select sysdate  from dual ;
```

All Rows Fetched: 1 in 0.03

```
  SYSDATE
1 27-04-22
```

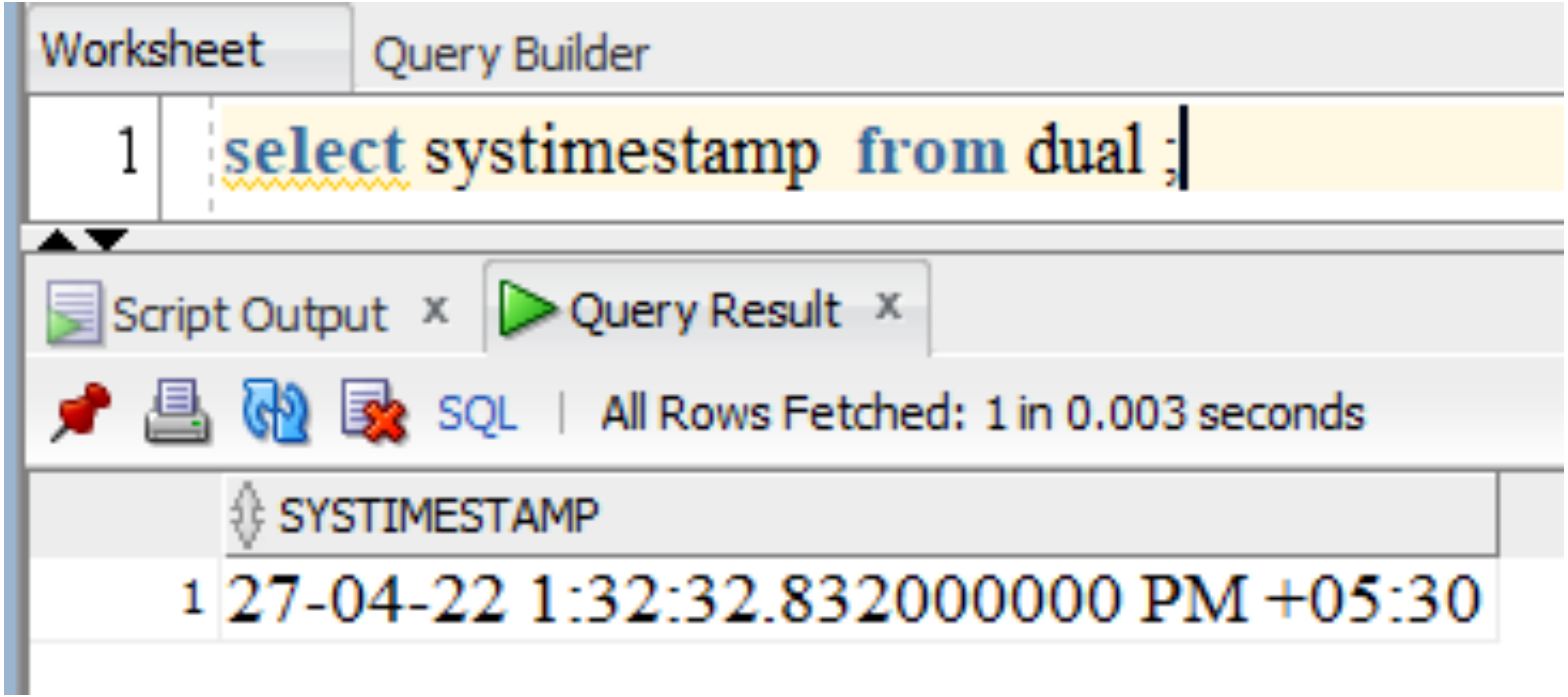
2) Systimestamp Introduced from Oracle 9i Returns date, time and timezone.



```sql
select systimestamp from dual ;
```

All Rows Fetched: 1 in 0.33 seconds

```
    SYSTIMESTAMP
  1 27-04-22 1:26:06.736000000 PM +05:30
```



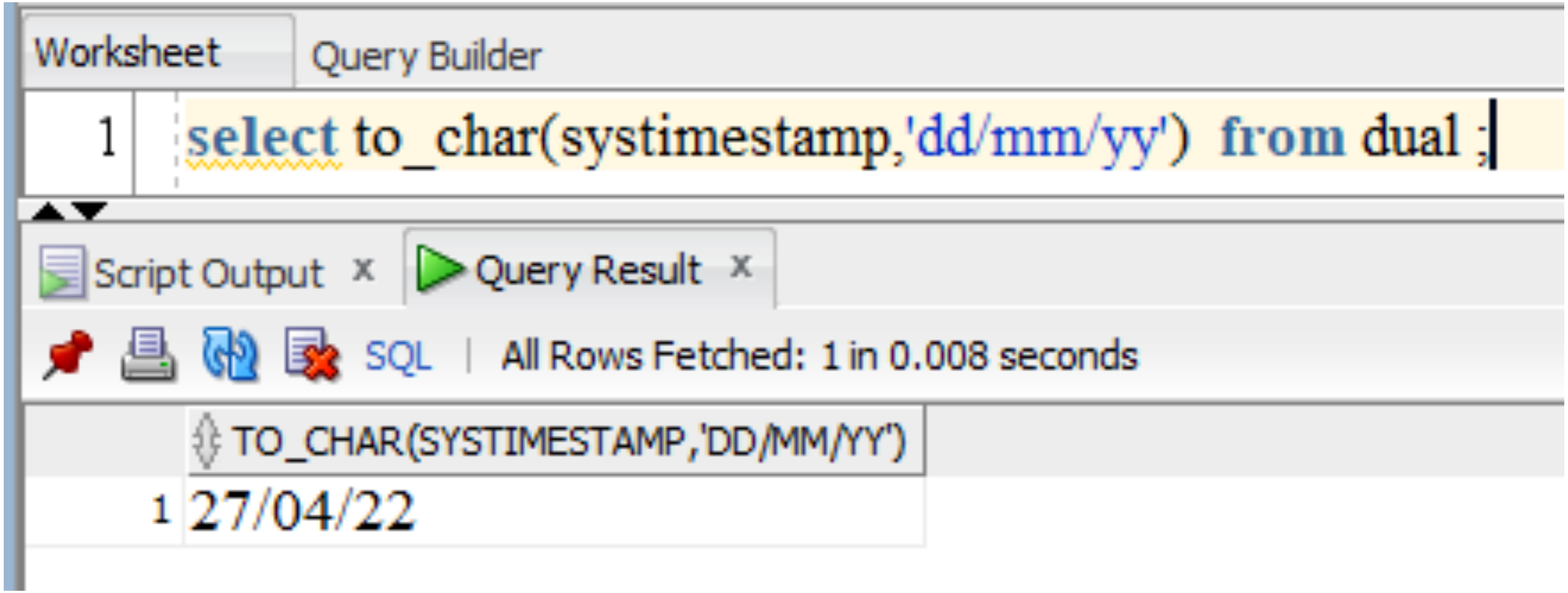
```
select systimestamp  from dual ;
```

All Rows Fetched: 1 in 0.003 seconds

```
  SYSTIMESTAMP
1 27-04-22 1:32:32.832000000 PM +05:30
```

In interview –if they ask you –"which function contains fractions of a second" OR "how to see the system time" –then answer is "SYSTIMESTAMP".

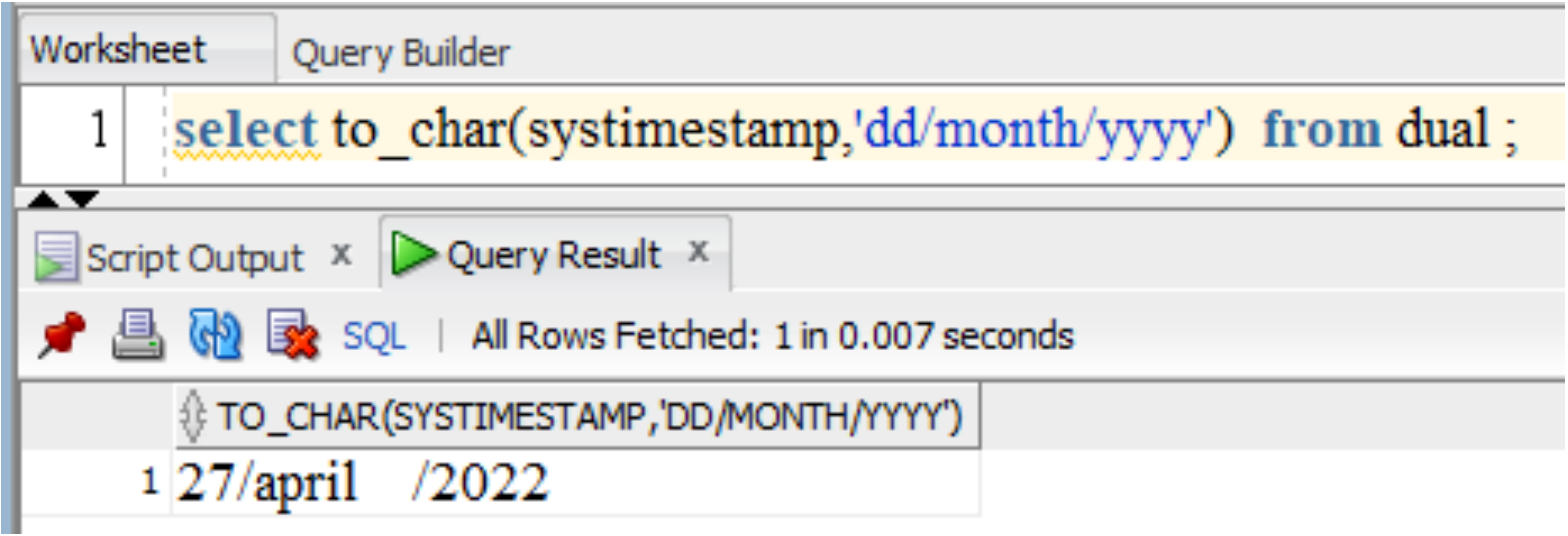
SPECIAL FUNCTIONS 1) TO –CHAR Used for displaying the date in different formats.



```
select to_char(systimestamp,'dd/mm/yy')  from dual ;
```

All Rows Fetched: 1 in 0.008 seconds

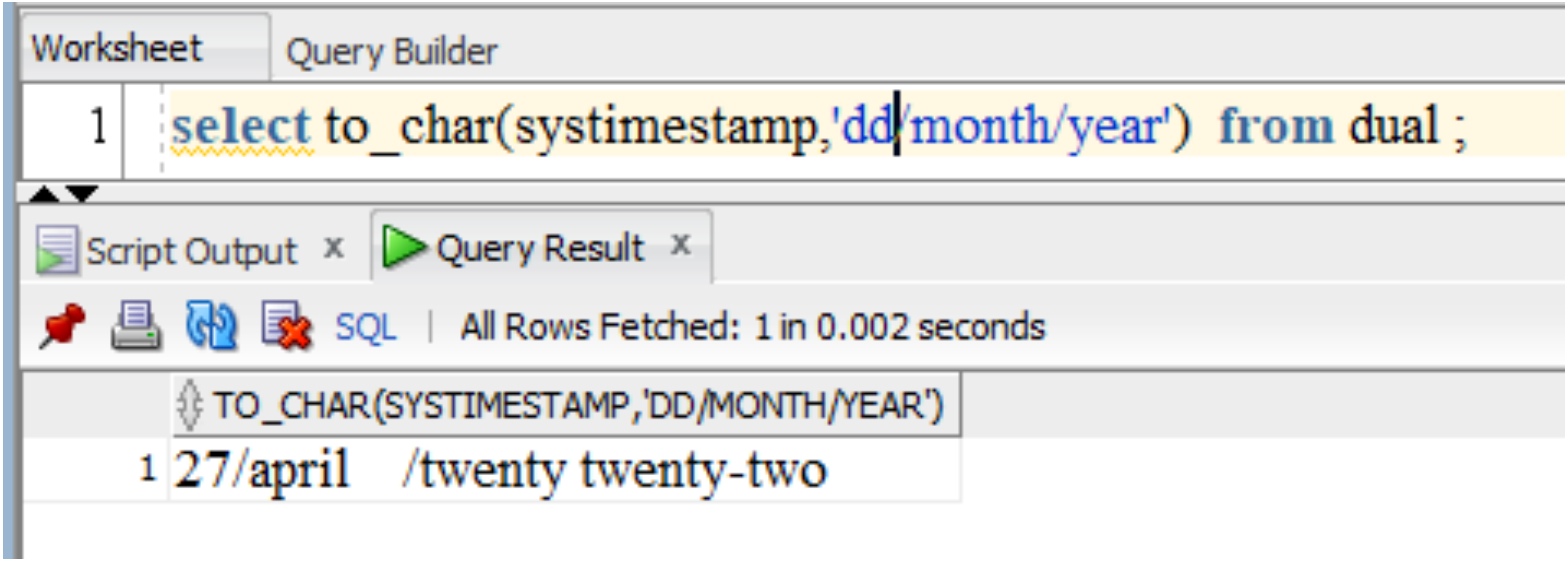
```
  TO_CHAR(SYSTIMESTAMP,'DD/MM/YY')
1 27/04/22
```



```
select to_char(systimestamp,'dd/month/yyyy')  from dual ;
```

All Rows Fetched: 1 in 0.007 seconds

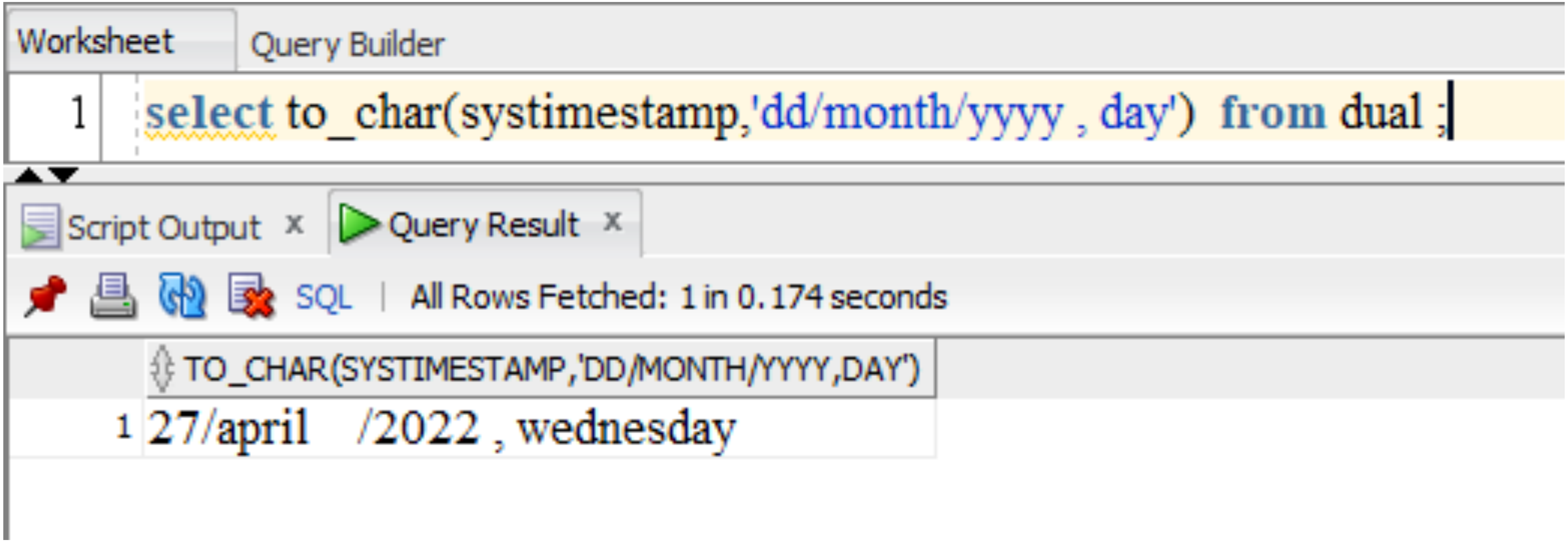
```
  TO_CHAR(SYSTIMESTAMP,'DD/MONTH/YYYY')
1 27/april    /2022
```



```sql
select to_char(systimestamp,'dd/month/year')  from dual ;
```

All Rows Fetched: 1 in 0.002 seconds

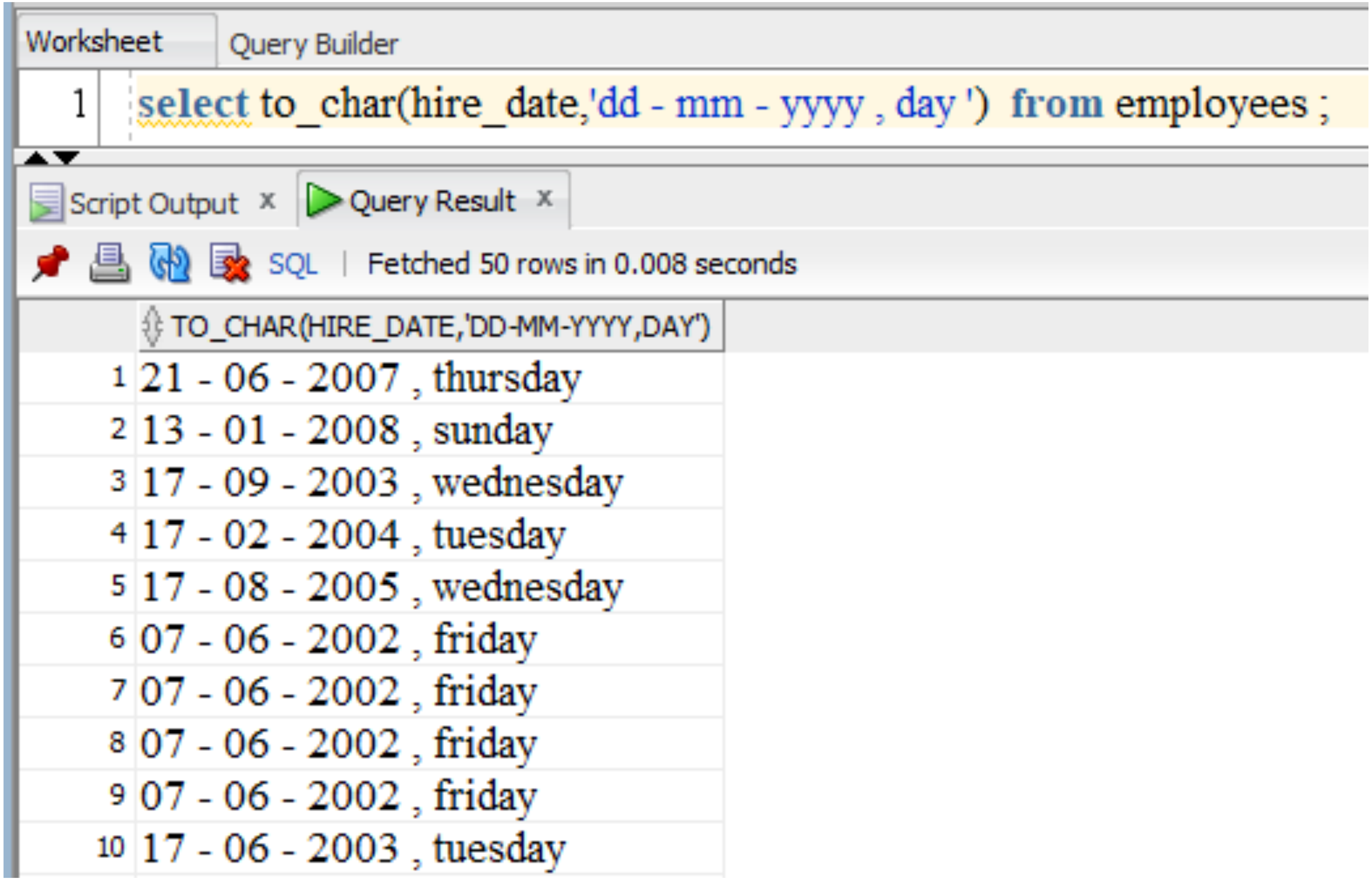
```
  TO_CHAR(SYSTIMESTAMP,'DD/MONTH/YEAR')
1 27/april    /twenty twenty-two
```



```
select to_char(systimestamp,'dd/month/yyyy , day')  from dual ;
```

All Rows Fetched: 1 in 0.174 seconds

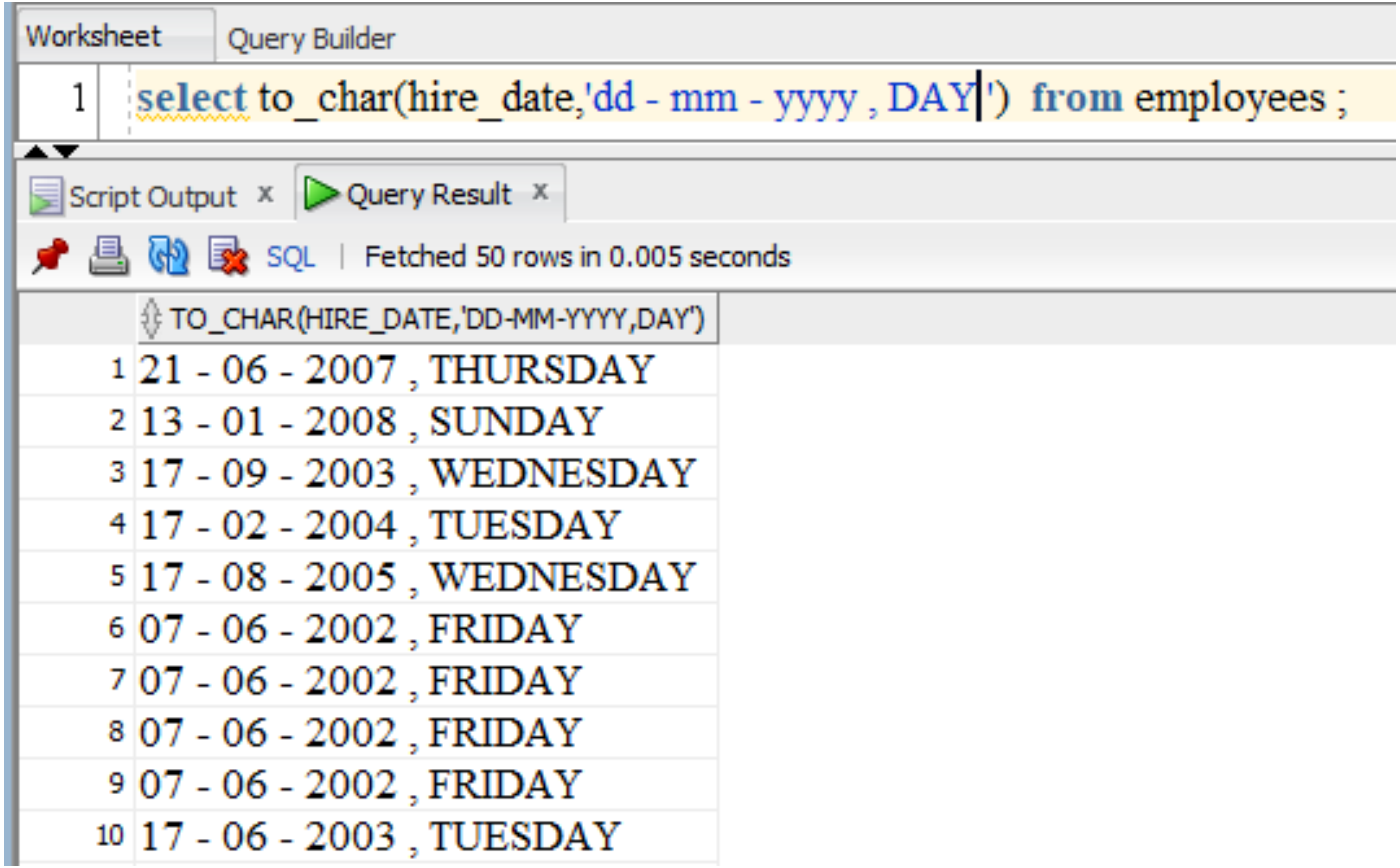
```
  TO_CHAR(SYSTIMESTAMP,'DD/MONTH/YYYY,DAY')
1 27/april    /2022 , wednesday
```



```
select to_char(hire_date,'dd - mm - yyyy , day ')  from employees ;
```

Fetched 50 rows in 0.008 seconds

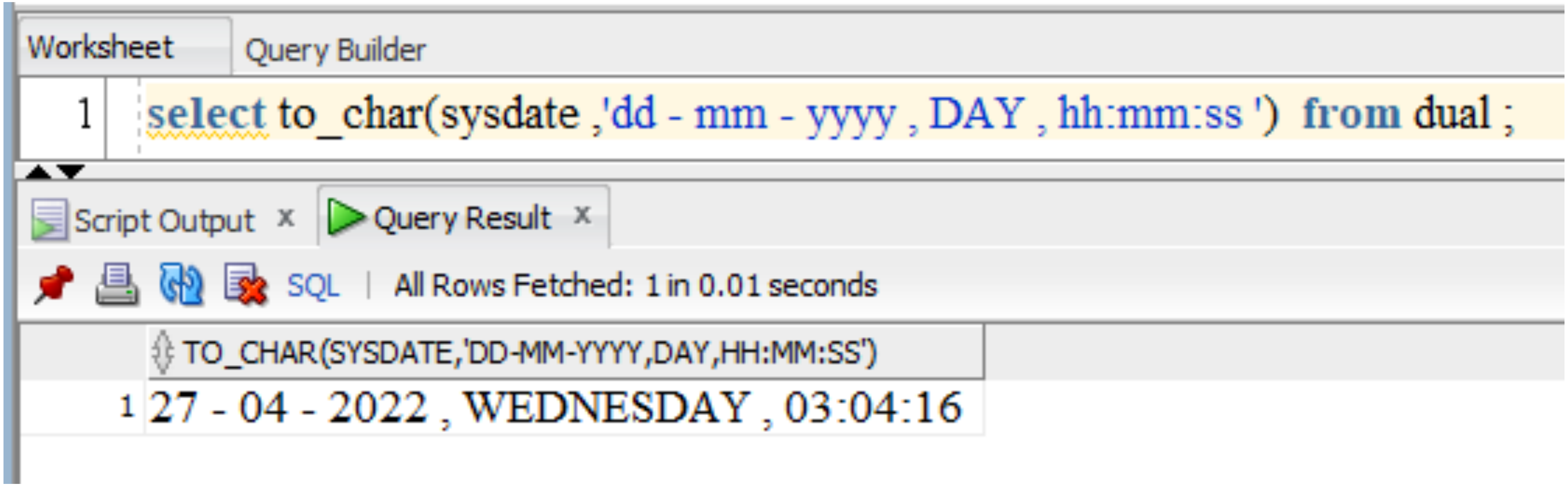
```
   TO_CHAR(HIRE_DATE,'DD-MM-YYYY,DAY')
 1 21 - 06 - 2007 , thursday
 2 13 - 01 - 2008 , sunday
 3 17 - 09 - 2003 , wednesday
 4 17 - 02 - 2004 , tuesday
 5 17 - 08 - 2005 , wednesday
 6 07 - 06 - 2002 , friday
 7 07 - 06 - 2002 , friday
 8 07 - 06 - 2002 , friday
 9 07 - 06 - 2002 , friday
10 17 - 06 - 2003 , tuesday
```



```sql
select to_char(hire_date,'dd - mm - yyyy , DAY')  from employees ;
```

Fetched 50 rows in 0.005 seconds

	TO_CHAR(HIRE_DATE,'DD-MM-YYYY,DAY')
1	21 - 06 - 2007 , THURSDAY
2	13 - 01 - 2008 , SUNDAY
3	17 - 09 - 2003 , WEDNESDAY
4	17 - 02 - 2004 , TUESDAY
5	17 - 08 - 2005 , WEDNESDAY
6	07 - 06 - 2002 , FRIDAY
7	07 - 06 - 2002 , FRIDAY
8	07 - 06 - 2002 , FRIDAY
9	07 - 06 - 2002 , FRIDAY
10	17 - 06 - 2003 , TUESDAY

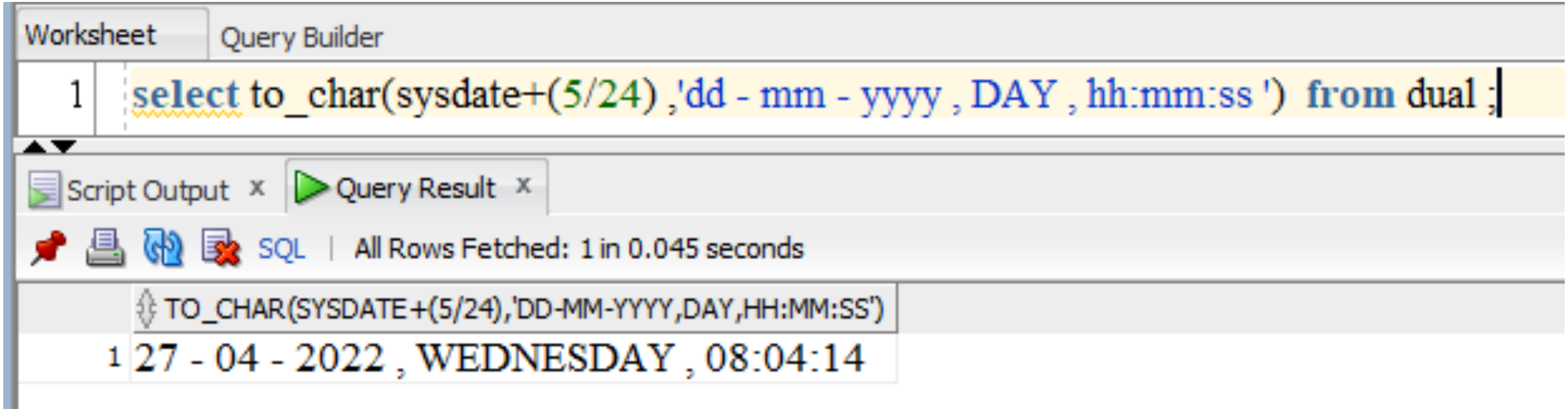


```
select to_char(sysdate ,'dd - mm - yyyy , DAY , hh:mm:ss ')  from dual ;
```

All Rows Fetched: 1 in 0.01 seconds

	TO_CHAR(SYSDATE,'DD-MM-YYYY,DAY,HH:MM:SS')
1	27 - 04 - 2022 , WEDNESDAY , 03:04:16

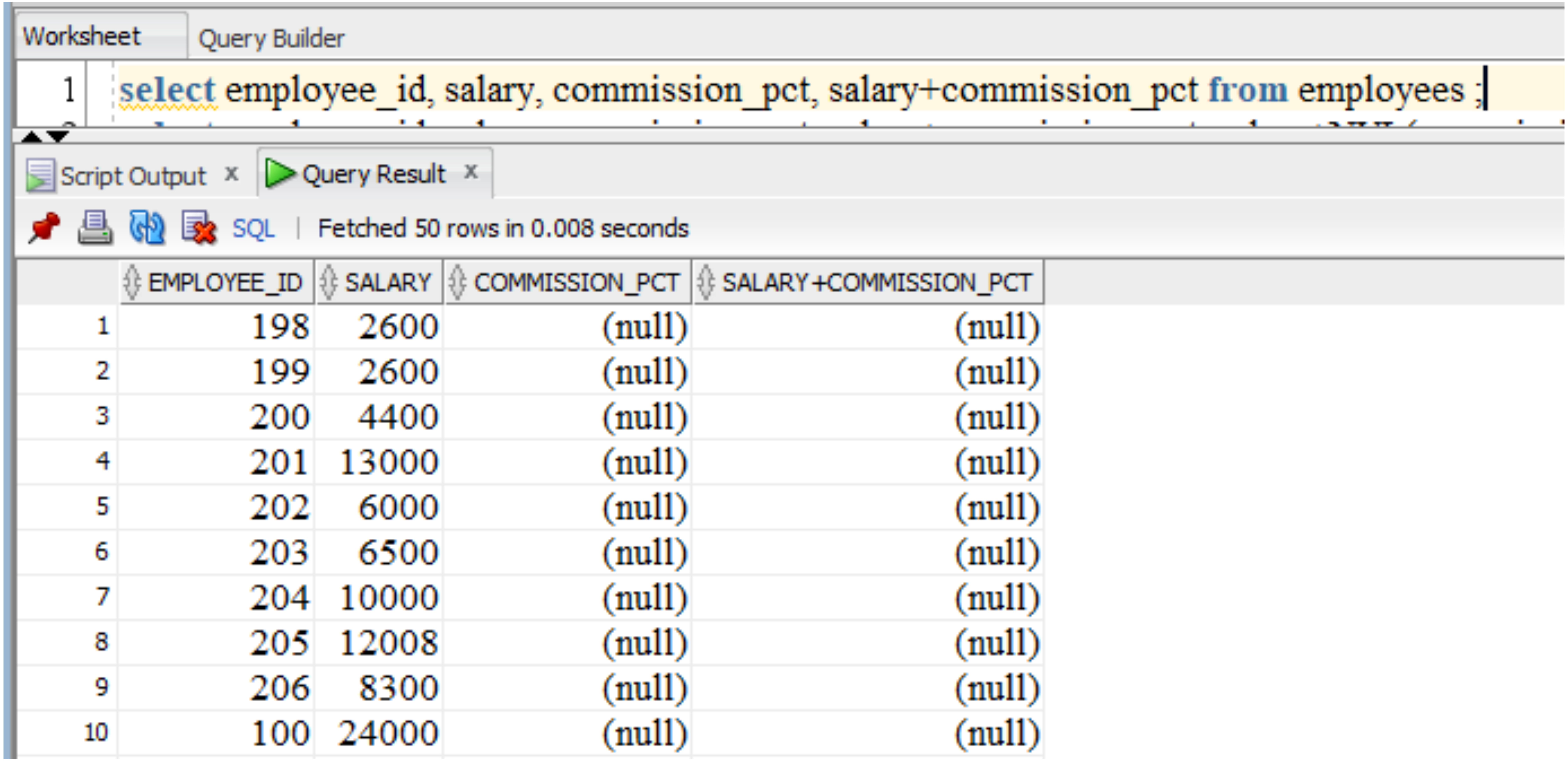
Now, let us see how to add 5 hrs to the existing time,



```
select to_char(sysdate+(5/24) ,'dd - mm - yyyy , DAY , hh:mm:ss ')  from dual ;
```

All Rows Fetched: 1 in 0.045 seconds

```
  TO_CHAR(SYSDATE+(5/24),'DD-MM-YYYY,DAY,HH:MM:SS')
1 27 - 04 - 2022 , WEDNESDAY , 08:04:14
```

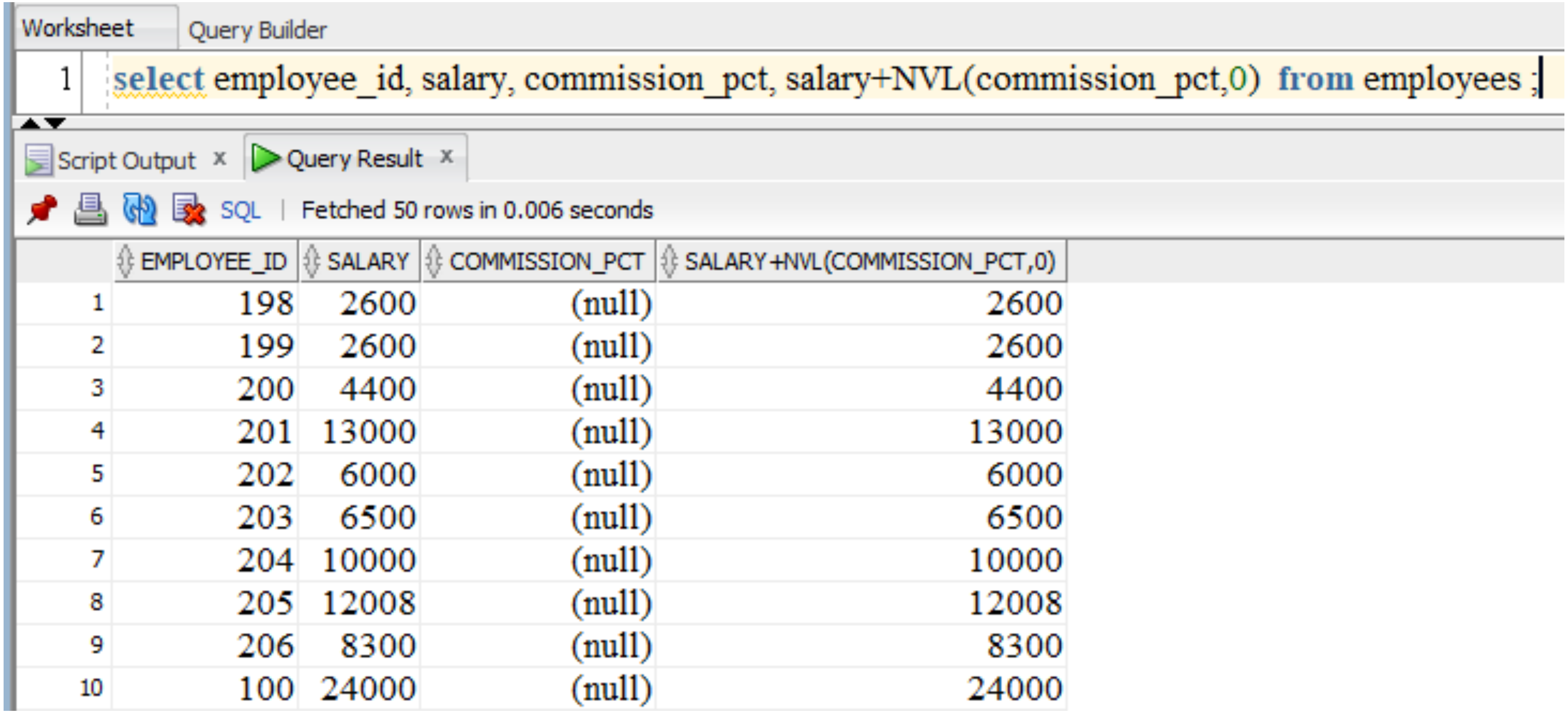


```
select employee_id, salary, commission_pct, salary+commission_pct from employees ;
```

Fetched 50 rows in 0.008 seconds

	EMPLOYEE_ID	SALARY	COMMISSION_PCT	SALARY+COMMISSION_PCT
1	198	2600	(null)	(null)
2	199	2600	(null)	(null)
3	200	4400	(null)	(null)
4	201	13000	(null)	(null)
5	202	6000	(null)	(null)
6	203	6500	(null)	(null)
7	204	10000	(null)	(null)
8	205	12008	(null)	(null)
9	206	8300	(null)	(null)
10	100	24000	(null)	(null)

NVL It substitutes a value for a null.

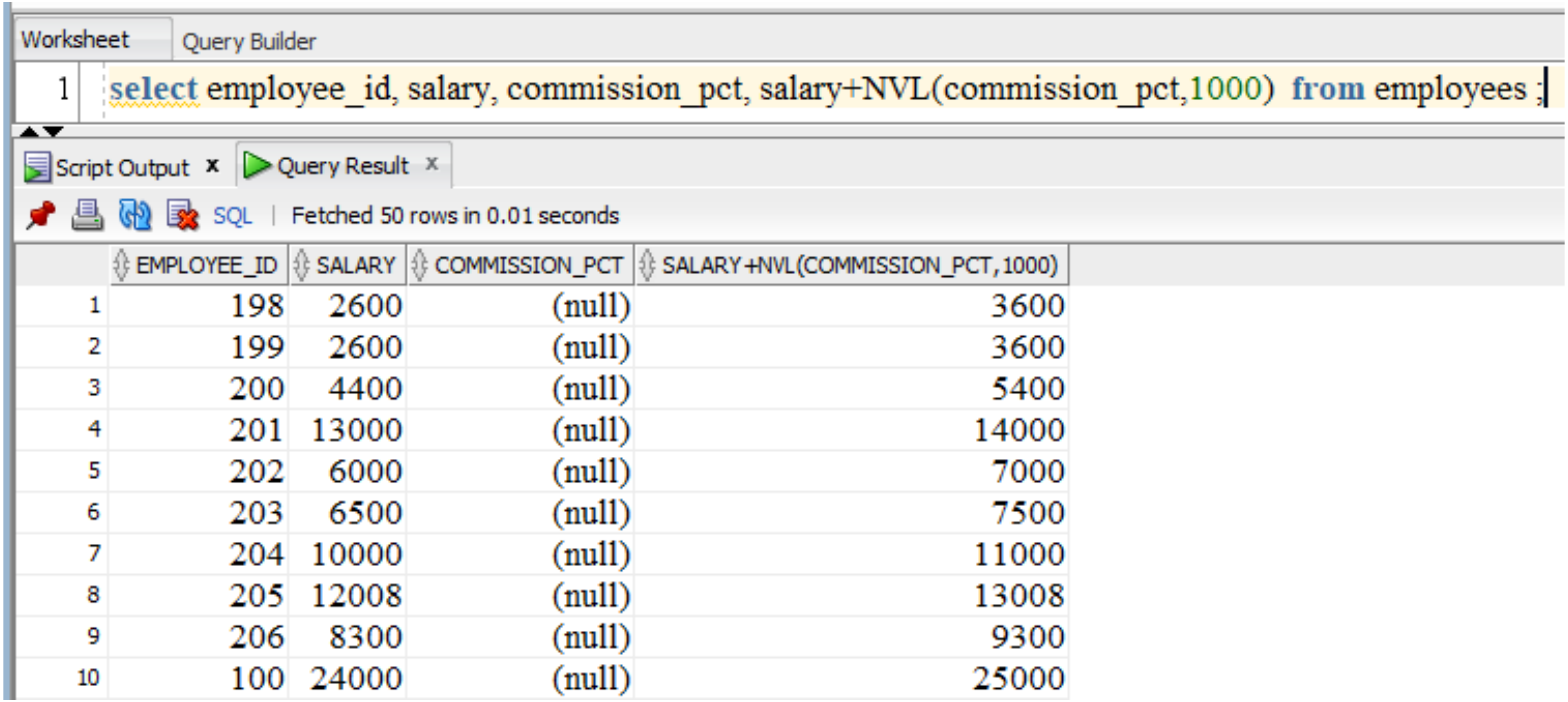


```
select employee_id, salary, commission_pct, salary+NVL(commission_pct,0)  from employees ;
```

Fetched 50 rows in 0.006 seconds

	EMPLOYEE_ID	SALARY	COMMISSION_PCT	SALARY+NVL(COMMISSION_PCT,0)
1	198	2600	(null)	2600
2	199	2600	(null)	2600
3	200	4400	(null)	4400
4	201	13000	(null)	13000
5	202	6000	(null)	6000
6	203	6500	(null)	6500
7	204	10000	(null)	10000
8	205	12008	(null)	12008
9	206	8300	(null)	8300
10	100	24000	(null)	24000

The above query means – if the employee has commission, then add sal + comm. To get total salary else add 0 to the sal and display total salary.

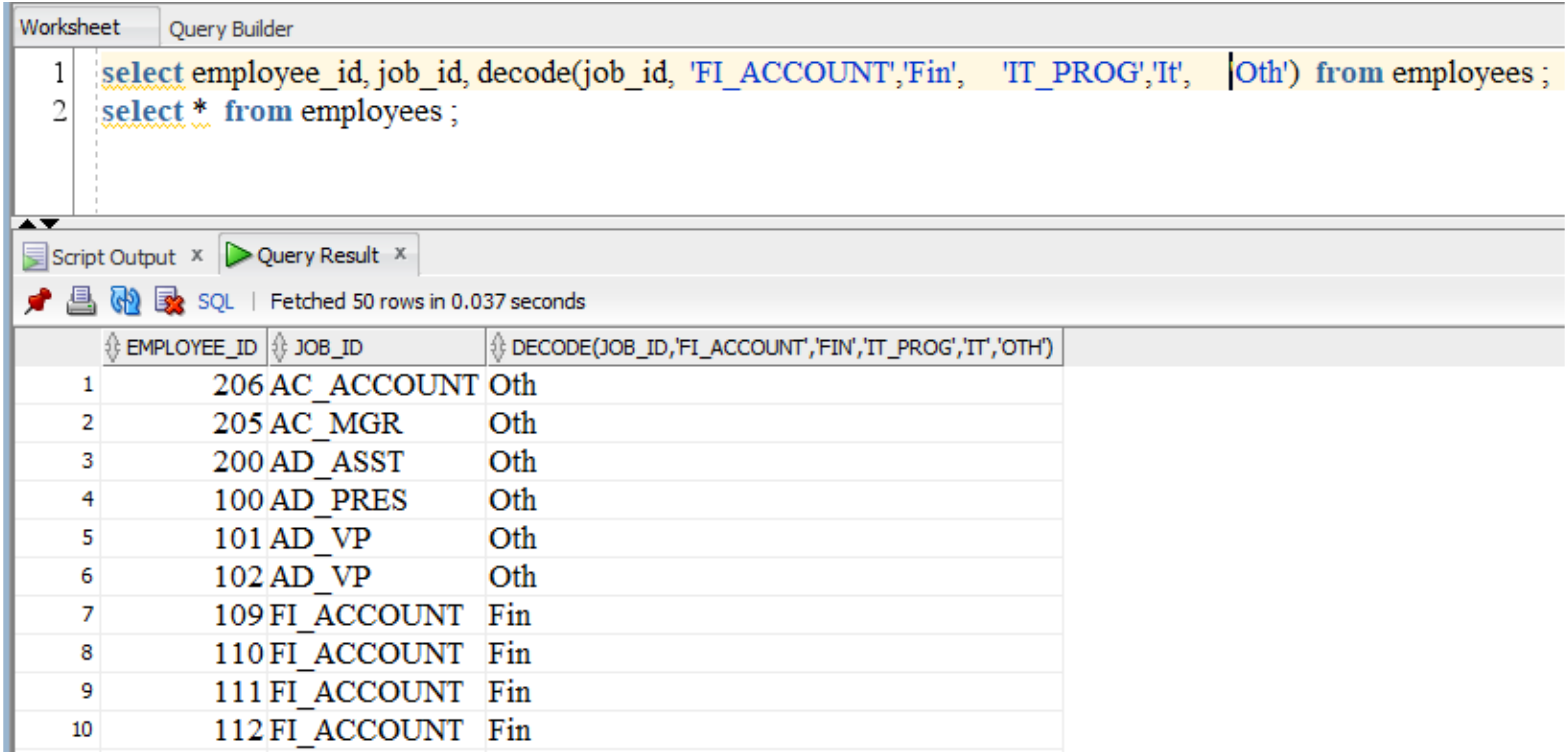


```
select employee_id, salary, commission_pct, salary+NVL(commission_pct,1000)  from employees ;
```

Fetched 50 rows in 0.01 seconds

	EMPLOYEE_ID	SALARY	COMMISSION_PCT	SALARY+NVL(COMMISSION_PCT,1000)
1	198	2600	(null)	3600
2	199	2600	(null)	3600
3	200	4400	(null)	5400
4	201	13000	(null)	14000
5	202	6000	(null)	7000
6	203	6500	(null)	7500
7	204	10000	(null)	11000
8	205	12008	(null)	13008
9	206	8300	(null)	9300
10	100	24000	(null)	25000

DECODE It works like „if– then– else“ statement.

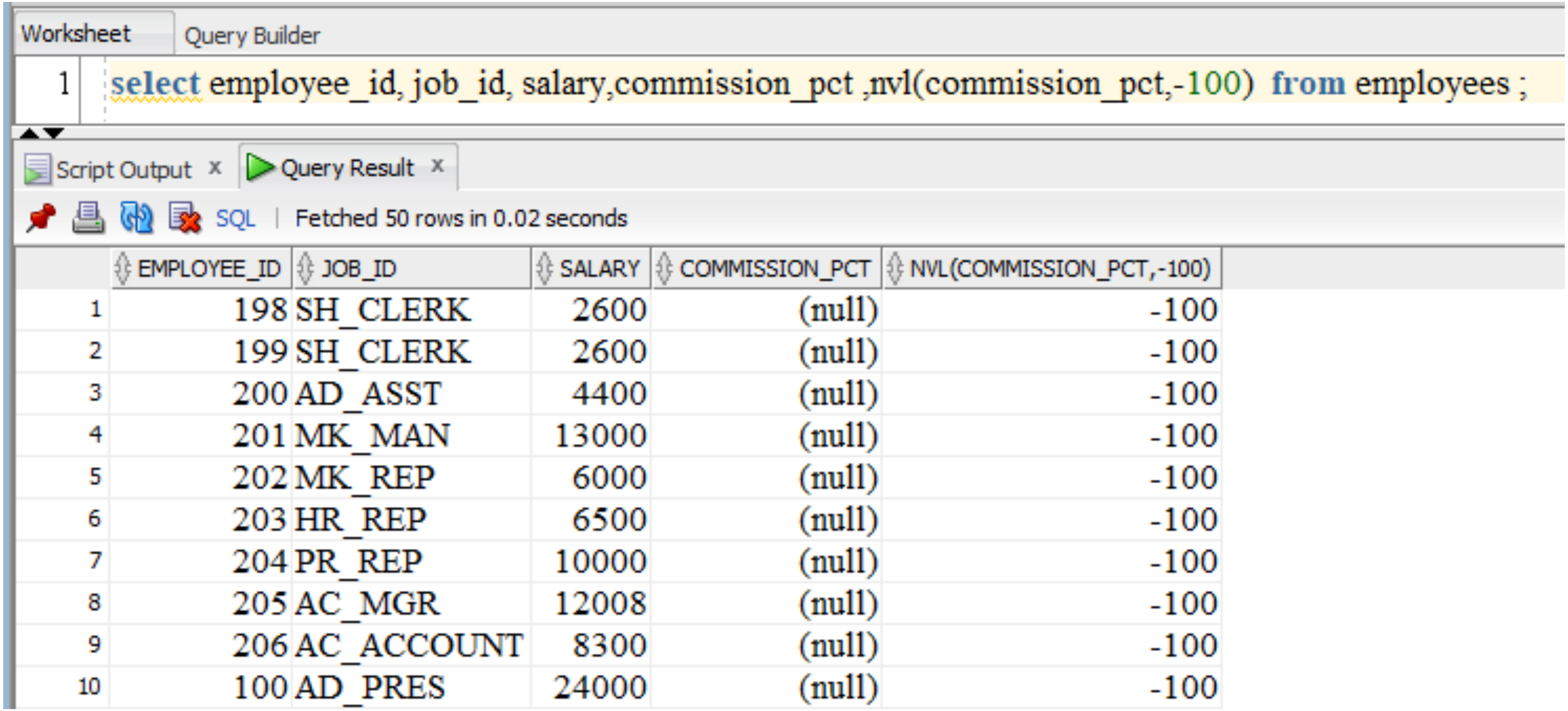


```
select employee_id, job_id, decode(job_id, 'FI_ACCOUNT','Fin',    'IT_PROG','It',    'Oth')  from employees ;
select *  from employees ;
```

Fetched 50 rows in 0.037 seconds

	EMPLOYEE_ID	JOB_ID	DECODE(JOB_ID,'FI_ACCOUNT','FIN','IT_PROG','IT','OTH')
1	206	AC_ACCOUNT	Oth
2	205	AC_MGR	Oth
3	200	AD_ASST	Oth
4	100	AD_PRES	Oth
5	101	AD_VP	Oth
6	102	AD_VP	Oth
7	109	FI_ACCOUNT	Fin
8	110	FI_ACCOUNT	Fin
9	111	FI_ACCOUNT	Fin
10	112	FI_ACCOUNT	Fin

Display employee name, job, salary and commission. If the commission is NULL, then display -100

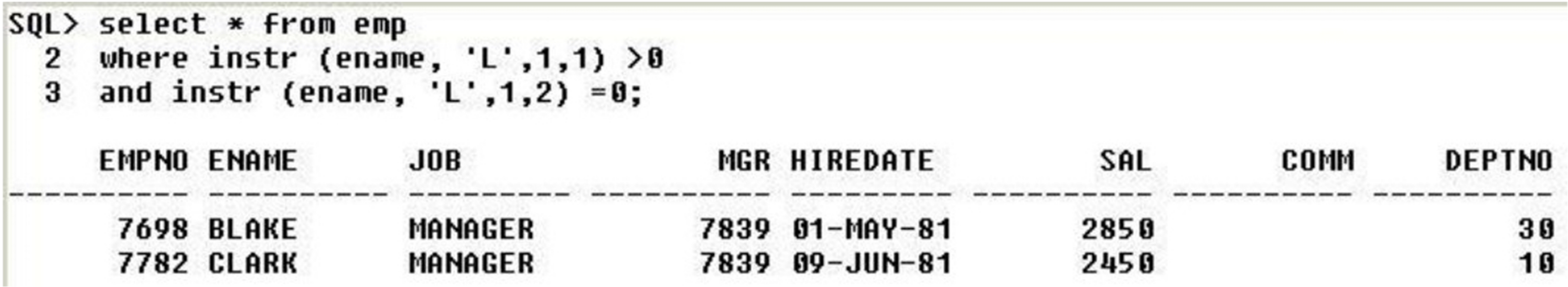


```
select employee_id, job_id, salary,commission_pct ,nvl(commission_pct,-100)  from employees ;
```

Fetched 50 rows in 0.02 seconds

	EMPLOYEE_ID	JOB_ID	SALARY	COMMISSION_PCT	NVL(COMMISSION_PCT,-100)
1	198	SH_CLERK	2600	(null)	-100
2	199	SH_CLERK	2600	(null)	-100
3	200	AD_ASST	4400	(null)	-100
4	201	MK_MAN	13000	(null)	-100
5	202	MK_REP	6000	(null)	-100
6	203	HR_REP	6500	(null)	-100
7	204	PR_REP	10000	(null)	-100
8	205	AC_MGR	12008	(null)	-100
9	206	AC_ACCOUNT	8300	(null)	-100
10	100	AD_PRES	24000	(null)	-100

Display all employees whose name is having exactly 1 „L“ in it



```
SQL> select * from emp
  2  where instr (ename, 'L',1,1) >0
  3  and instr (ename, 'L',1,2) =0;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
```

To Summarise

- Functions
- Types
  - User Defined
  - Pre Defined
    - Group
    - Character
    - Numeric
    - Date
    - Special

Thank You