

**NAME :-** Atharva Bhupesh Sankhe.

**UID :-** 2022300095.

**BRANCH :-** Comps -B. **BATCH:** B.

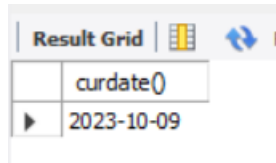
**EXPERIMENT 4:**

**SUBJECT :-** DBMS (DATABASE MANAGEMENT SYSTEM)

**TOPIC NO.1:** Perform the date operation on given date and write the output:

**1. curdate():**

select curdate();

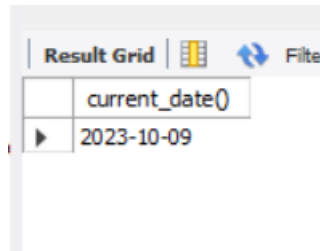


The screenshot shows a 'Result Grid' window with a single column header 'curdate()' and one data row containing the value '2023-10-09'.

curdate()
2023-10-09

**2. currentdate():**

select current\_date();

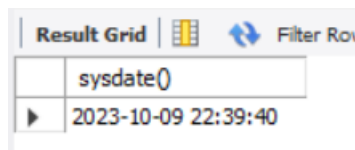


The screenshot shows a 'Result Grid' window with a single column header 'current\_date()' and one data row containing the value '2023-10-09'.

current_date()
2023-10-09

**3. sysdate():**

select sysdate();

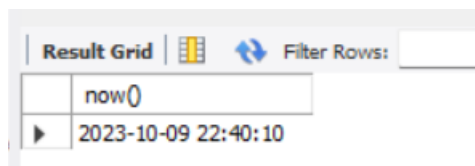


The screenshot shows a 'Result Grid' window with a single column header 'sysdate()' and one data row containing the value '2023-10-09 22:39:40'.

sysdate()
2023-10-09 22:39:40

**4. now():**

select now();

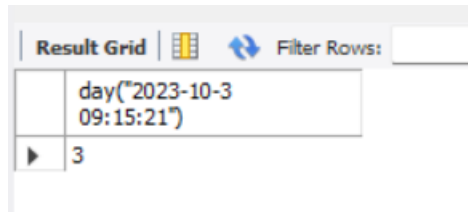


The screenshot shows a 'Result Grid' window with a single column header 'now()' and one data row containing the value '2023-10-09 22:40:10'.

now()
2023-10-09 22:40:10

**5. day:**

```
select day("2023-10-3 09:15:21");
```

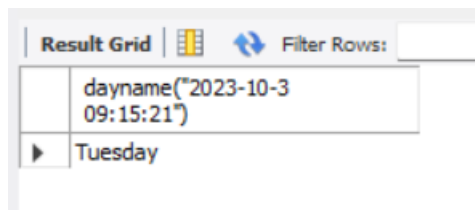


The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains one row with the expression 'day("2023-10-3 09:15:21")' and the result '3'.

	day("2023-10-3 09:15:21")
▶	3

**6. dayname:**

```
select dayname("2023-10-3 09:15:21");
```

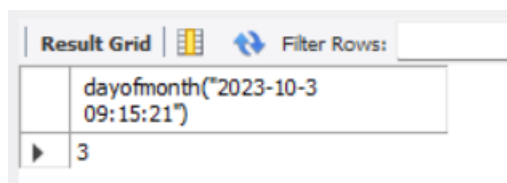


The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains one row with the expression 'dayname("2023-10-3 09:15:21")' and the result 'Tuesday'.

	dayname("2023-10-3 09:15:21")
▶	Tuesday

**7. dayofmonth:**

```
select dayofmonth("2023-10-3 09:15:21");
```

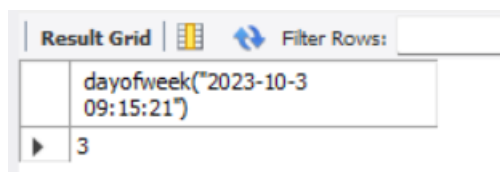


The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains one row with the expression 'dayofmonth("2023-10-3 09:15:21")' and the result '3'.

	dayofmonth("2023-10-3 09:15:21")
▶	3

**8. dayofweek:**

```
select dayofweek("2023-10-3 09:15:21");
```

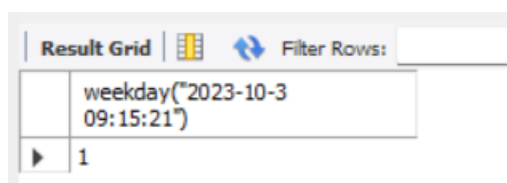


The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains one row with the expression 'dayofweek("2023-10-3 09:15:21")' and the result '3'.

	dayofweek("2023-10-3 09:15:21")
▶	3

**9. weekday:**

```
select weekday("2023-10-3 09:15:21");
```



The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid contains one row with the expression 'weekday("2023-10-3 09:15:21")' and the result '1'.

	weekday("2023-10-3 09:15:21")
▶	1

**10. year:**

```
select year("2023-10-3 09:15:21");
```

Result Grid		Filter Rows:
	weekday("2023-10-3 09:15:21")	
▶	1	

**11. extract function for date, extract the month,year,seconds,minutes,hour,day,week, month,quarter:**

```
select extract(month from("2023-10-3"))as date;
```

Result Grid	
	date
▶	10

```
select extract(year from("2023-10-3"))as date;
```

Result Grid	
	date
▶	2023

```
select extract(day from("2023-10-3"))as date;
```

Result Grid	
	date
▶	3

```
select dayofmonth("2023-10-3");
```

Result Grid		Filter Rows:
	dayofmonth("2023-10-3")	
▶	3	

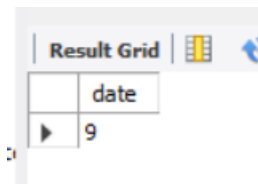
```
select extract(second from("2023-10-3 09:15:21"))as date;
```

Result Grid	
	date
▶	21

```
select extract(minute from("2023-10-3 09:15:21"))as date;
```

Result Grid	
	date
▶	15

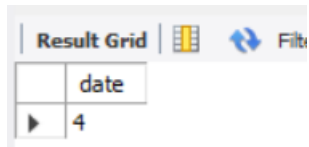
```
select extract(hour from("2023-10-3 09:15:21"))as date;
```



The screenshot shows a 'Result Grid' window with a table containing one column named 'date' and one row with the value '9'. The window has a toolbar with icons for grid, table, and refresh, and a 'Filter Rows' input field.

date
9

```
select extract(quarter from("2023-10-3 09:15:21"))as date;
```

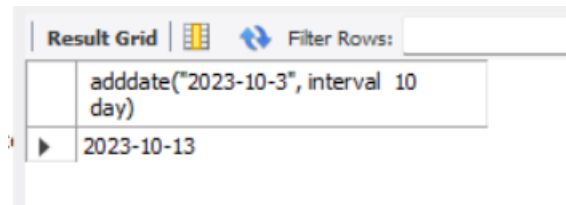


The screenshot shows a 'Result Grid' window with a table containing one column named 'date' and one row with the value '4'. The window has a toolbar with icons for grid, table, and refresh, and a 'Filter Rows' input field.

date
4

## 12. Use adddate and subdate function:

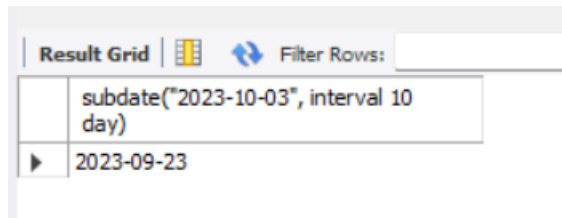
```
select adddate("2023-10-3", interval 10 day);
```



The screenshot shows a 'Result Grid' window with a table containing one column with the expression 'adddate("2023-10-3", interval 10 day)' and one row with the result '2023-10-13'. The window has a toolbar with icons for grid, table, and refresh, and a 'Filter Rows' input field.

adddate("2023-10-3", interval 10 day)
2023-10-13

```
select subdate("2023-10-03", interval 10 day);
```



The screenshot shows a 'Result Grid' window with a table containing one column with the expression 'subdate("2023-10-03", interval 10 day)' and one row with the result '2023-09-23'. The window has a toolbar with icons for grid, table, and refresh, and a 'Filter Rows' input field.

subdate("2023-10-03", interval 10 day)
2023-09-23

**TOPIC NO. 2: Use following arithmetic functions:****1. PI():**

```
select pi();
```

Result Grid	
	pi()
▶	3.141593

**2. Round():**

```
select round(2.5);
```

Result Grid	
	round(2.5)
▶	3

**3. CEIL():**

```
select ceil(2.5);
```

Result Grid	
	ceil(2.5)
▶	3

**4. Floor():**

```
select floor(2.5);
```

Result Grid	
	floor(2.5)
▶	2

**5. Pow():**

```
select pow(2,9);
```

Result Grid	
	pow(2,9)
▶	512

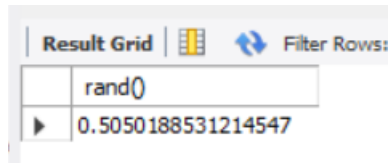
**6. SQRT():**

```
select sqrt(28);
```

Result Grid	
	sqrt(28)
▶	5.291502622129181

**7. RAND():**

select rand();

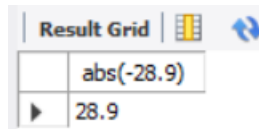


The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains one row with the expression 'rand()' and its corresponding value '0.5050188531214547'.

	rand()
▶	0.5050188531214547

**8. ABS():**

select abs(-28.9);

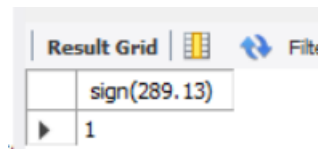


The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains one row with the expression 'abs(-28.9)' and its corresponding value '28.9'.

	abs(-28.9)
▶	28.9

**9. SIGN():**

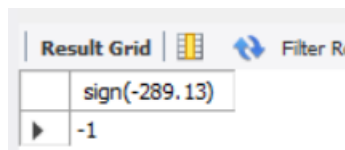
select sign(289.13);



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains one row with the expression 'sign(289.13)' and its corresponding value '1'.

	sign(289.13)
▶	1

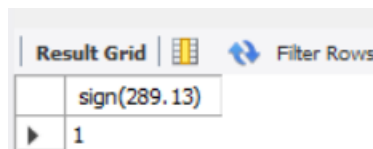
select sign(-289.13);



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains one row with the expression 'sign(-289.13)' and its corresponding value '-1'.

	sign(-289.13)
▶	-1

select sign(0);

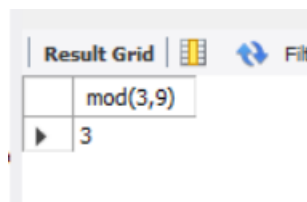


The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains one row with the expression 'sign(289.13)' and its corresponding value '1'.

	sign(289.13)
▶	1

**10. MOD():**

select mod(3,9);



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains one row with the expression 'mod(3,9)' and its corresponding value '3'.

	mod(3,9)
▶	3

select mod(3,9) as remainder;

Result Grid		Filter
	remainder	
▶	3	

**TOPIC NO 3:** Create EMP ( id, lname,fname,title,deptno , salary) table and insert 10 values in it and perform following queries on it.

- a. Write a SQL query and show the resulting table of salaries if all warehouse manager were given a Rs. 250 per month raise.

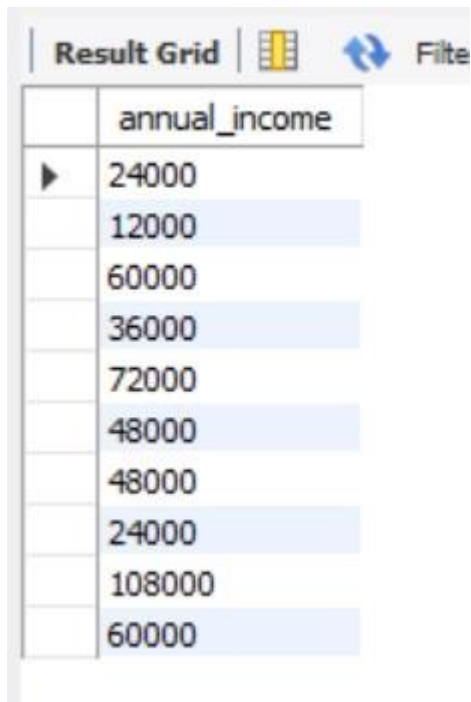
update EMP set salary = salary + 250 where title="Manager";

Result Grid		Filter
	salary	
▶	2250	
	1000	
	5250	
	3000	
	6250	
	4000	
	4250	
	2000	
	9250	
	5000	

### DBMS EXPERIMENT NO. 3

- b. Write a SQL query and show the resulting table to display the amount each warehouse manager currently makes in a year and how much each one would make in a year with the Rs.250 per month raise. name the new column appropriately.

update EMP set annual\_income= (salary)\*12;



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays a single column named 'annual\_income' with 12 rows of data. The values are: 24000, 12000, 60000, 36000, 72000, 48000, 48000, 24000, 108000, 60000, and two empty rows at the bottom. The interface includes icons for a film strip, a refresh button, and a 'Filter' label.

annual_income
24000
12000
60000
36000
72000
48000
48000
24000
108000
60000