Program No:	**3**
Roll No :	**1534**
Title of Program :	**Implement of Analytical queries**
Objective :	- rollup, cube, rank, dense, rank, lead, lag, first, last

THE MET LEAGUE OF COLLEGES
MET
AS SHARP AS YOU CAN GET
Bhujbal Knowledge City

SOURCE CODE:

```
CREATE TABLE employee (
   emp_no NUMBER(10),
   dep_no NUMBER(10),
   emp_name VARCHAR(25),
   salary NUMBER(10),
   comm NUMBER(10),
   job VARCHAR(25),
   bdate DATE
);

INSERT INTO employee VALUES (101, 10, 'sanket', 22000, 10000, 'assistent', TO_DATE('1990-01-15',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (102, 10, 'josh', 22000, 2000, 'assistent', TO_DATE('1991-02-20',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (103, 20, 'hendry', 22000, 1000, 'CLERK', TO_DATE('1992-03-25',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (104, 70, 'sanket cha papa', 22000, 1000, 'manager', TO_DATE('1989-04-
30', 'YYYY-MM-DD'));
INSERT INTO employee VALUES (105, 10, 'abhay', 22000, 1000, 'assistent', TO_DATE('1993-05-05',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (106, 50, 'james', 22000, 5000, 'CLERK', TO_DATE('1988-06-10',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (107, 10, 'bond', 22000, 1000, 'CLERK', TO_DATE('1990-07-15',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (108, 10, 'hardik', 22000, 3000, 'assistent', TO_DATE('1994-08-20',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (109, 70, 'rohit', 22000, 1000, 'manager', TO_DATE('1987-09-25',
'YYYY-MM-DD'));
INSERT INTO employee VALUES (110, 90, 'rishab', 22000, 7000, 'CLERK', TO_DATE('1995-10-30',
'YYYY-MM-DD'));
```

OUTPUT:

```
SQL> INSERT INTO employee VALUES (107, 10, 'bond', 22000, 1000, 'CLERK', TO_DATE('1990-07-15', 'YYYY-MM-DD'));

1 row created.

SQL> INSERT INTO employee VALUES (108, 10, 'hardik', 22000, 3000, 'assistent', TO_DATE('1994-08-20', 'YYYY-MM-DD'));

1 row created.

SQL> INSERT INTO employee VALUES (109, 70, 'rohit', 22000, 1000, 'manager', TO_DATE('1987-09-25', 'YYYY-MM-DD'));

1 row created.

SQL> INSERT INTO employee VALUES (110, 90, 'rishab', 22000, 7000, 'CLERK', TO_DATE('1995-10-30', 'YYYY-MM-DD'));

1 row created.

SQL>
```

```
  9  );

Table created.
```

LEAD:

SOURCE CODE:
SELECT emp_no, bdate, LEAD(bdate, 1) OVER (ORDER BY bdate) AS "next"FROM employee;

OUTPUT:

```
SQL> SELECT emp_no, bdate, LEAD(bdate, 1) OVER (ORDER BY bdate) AS "next"FROM employee;

    EMP_NO BDATE     next
---------- --------- ---------
       109 25-SEP-87 10-JUN-88
       106 10-JUN-88 30-APR-89
       104 30-APR-89 15-JAN-90
       101 15-JAN-90 15-JUL-90
       107 15-JUL-90 20-FEB-91
       102 20-FEB-91 25-MAR-92
       103 25-MAR-92 05-MAY-93
       105 05-MAY-93 20-AUG-94
       108 20-AUG-94 30-OCT-95
       110 30-OCT-95

10 rows selected.

SQL>
```

LAG

SOURCE CODE: SELECT emp_no,bdate, LAG(bdate,1) OVER (order by bdate) as "Previous" FROM Employee ;

OUTPUT:

```
SQL> SELECT emp_no,bdate, LAG(bdate,1) OVER (order by bdate ) as "Previous" FROM Employee ;

    EMP_NO BDATE     Previous
---------- --------- ---------
       109 25-SEP-87
       106 10-JUN-88 25-SEP-87
       104 30-APR-89 10-JUN-88
       101 15-JAN-90 30-APR-89
       107 15-JUL-90 15-JAN-90
       102 20-FEB-91 15-JUL-90
       103 25-MAR-92 20-FEB-91
       105 05-MAY-93 25-MAR-92
       108 20-AUG-94 05-MAY-93
       110 30-OCT-95 20-AUG-94

10 rows selected.
```

FIRST
SOURCE CODE:
SELECT dep_no, salary, MAX(salary) KEEP (DENSE_RANK FIRST ORDER BY salary DESC) OVER (PARTITION BY dep_no) AS "max" FROM employee;

OUTPUT:

```
SQL> SELECT     dep_no, salary,  MAX(salary) KEEP (DENSE_RANK FIRST ORDER BY salary DESC) OVER (PARTITION BY dep_no) AS "max" FROM employee;

    DEP_NO     SALARY        max
---------- ---------- ----------
        10      22000      22000
        10      22000      22000
        10      22000      22000
        10      22000      22000
        10      22000      22000
        20      22000      22000
        50      22000      22000
        70      22000      22000
        70      22000      22000
        90      22000      22000

10 rows selected.

SQL>
```

LAST:

SOURCE CODE : SELECT dep_no, salary, MIN(salary) KEEP (DENSE_RANK LAST ORDER BY salary DESC) OVER (PARTITION BY dep_no) AS "last" FROM employee;

OUTPUT:

```
SQL> SELECT dep_no, salary,  MIN(salary) KEEP (DENSE_RANK LAST ORDER BY salary DESC) OVER (PARTITION BY dep_no) AS "last" FROM employee;

    DEP_NO     SALARY       last
---------- ---------- ----------
        10      22000      22000
        10      22000      22000
        10      22000      22000
        10      22000      22000
        10      22000      22000
        20      22000      22000
        50      22000      22000
        70      22000      22000
        70      22000      22000
        90      22000      22000

10 rows selected.

SQL>
```