Name: Shaikh Inamul Hasan

Roll No: 100

LAB1: Practical Assignment 2

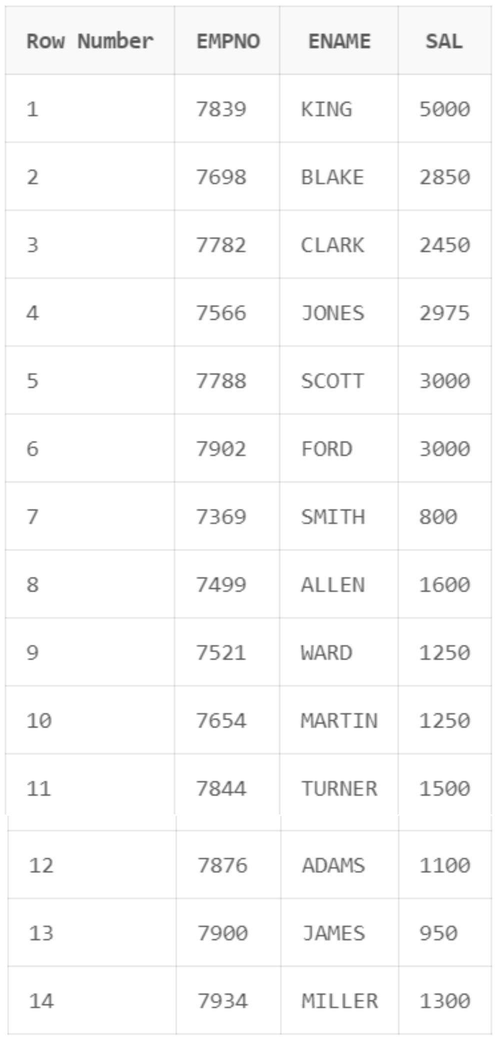
Topic: Analytical functions
(Rank(), Dense_rank(), ROW_NUMBER(),Top N query)

1. Display empno, ename, sal from emp table and give numbers to each row.

Code:

```
SELECT ROWNUM AS "Row Number",
       EMPNO,
       ENAME,
       SAL
FROM SCOTT."EMP";
```

Output:



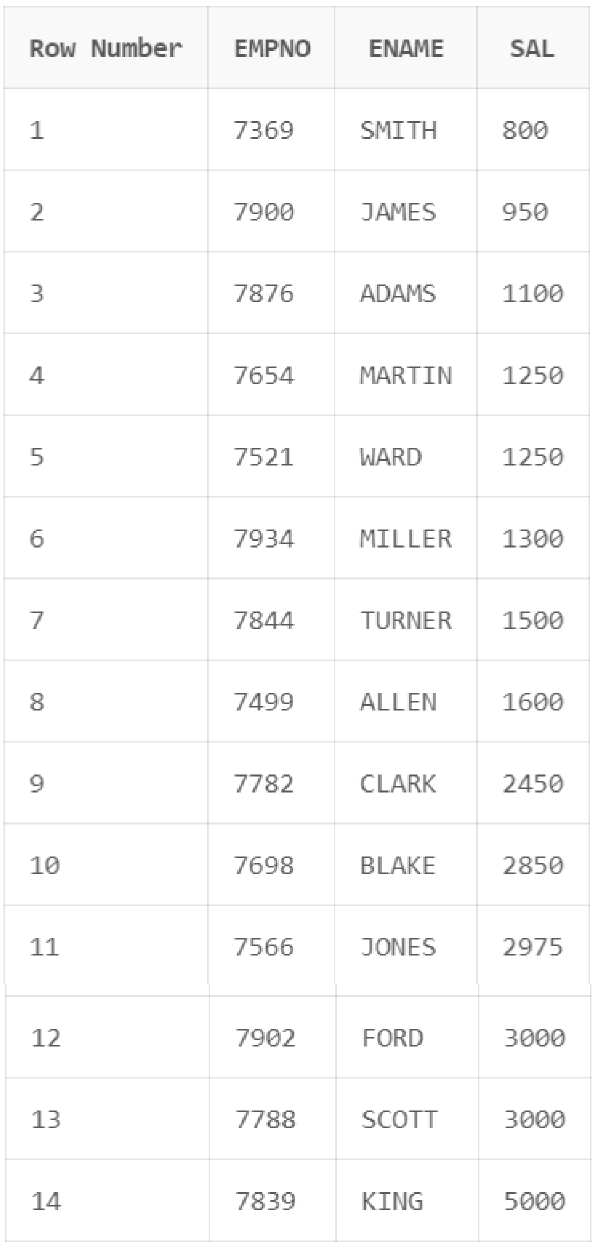
Row Number	EMPNO	ENAME	SAL
1	7839	KING	5000
2	7698	BLAKE	2850
3	7782	CLARK	2450
4	7566	JONES	2975
5	7788	SCOTT	3000
6	7902	FORD	3000
7	7369	SMITH	800
8	7499	ALLEN	1600
9	7521	WARD	1250
10	7654	MARTIN	1250
11	7844	TURNER	1500
12	7876	ADAMS	1100
13	7900	JAMES	950
14	7934	MILLER	1300

2. Display empno, ename, sal and give numbers to each row in ascending order of salary.

<u>Code:</u>

```
SELECT ROW_NUMBER() OVER (ORDER BY SAL ASC) AS "Row Number",
    EMPNO,
    ENAME,
    SAL
FROM SCOTT."EMP";
```

<u>Output:</u>



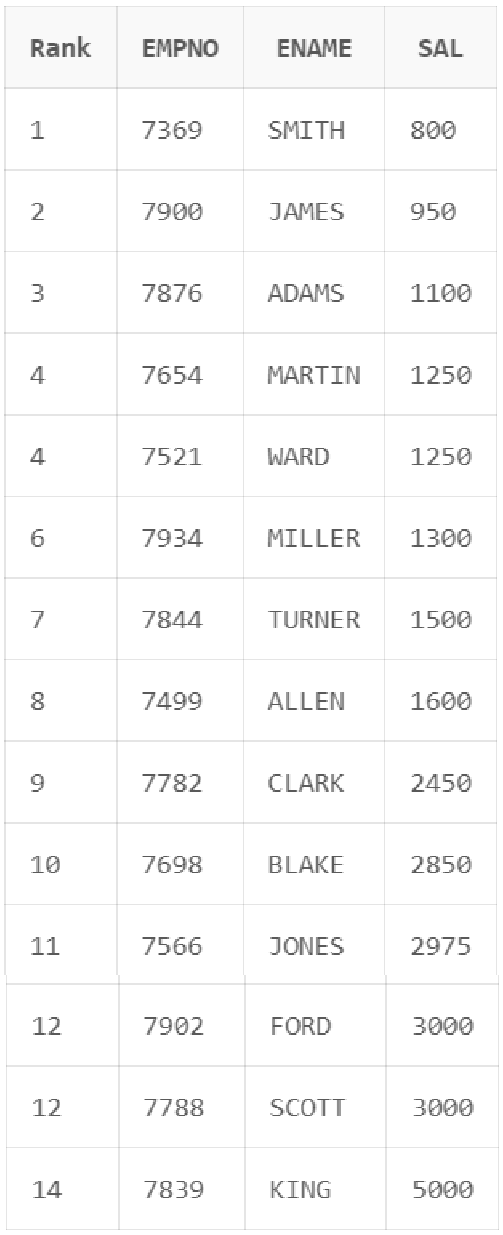
Row Number	EMPNO	ENAME	SAL
1	7369	SMITH	800
2	7900	JAMES	950
3	7876	ADAMS	1100
4	7654	MARTIN	1250
5	7521	WARD	1250
6	7934	MILLER	1300
7	7844	TURNER	1500
8	7499	ALLEN	1600
9	7782	CLARK	2450
10	7698	BLAKE	2850
11	7566	JONES	2975
12	7902	FORD	3000
13	7788	SCOTT	3000
14	7839	KING	5000

3. Assign the ranks to employees in ascending order of salary.

Code:

```
SELECT RANK() OVER (ORDER BY SAL ASC) AS "Rank",
       EMPNO,
       ENAME,
       SAL
FROM SCOTT."EMP";
```

Output:



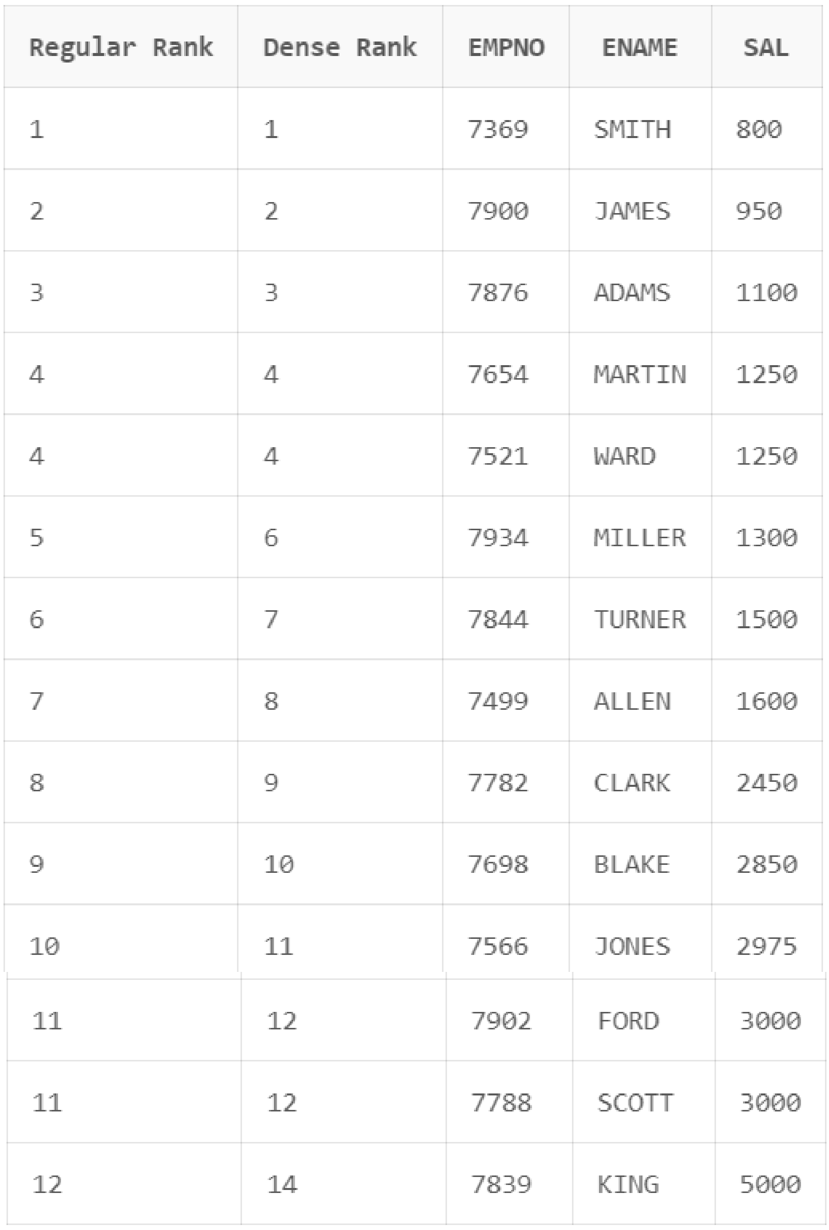
Rank	EMPNO	ENAME	SAL
1	7369	SMITH	800
2	7900	JAMES	950
3	7876	ADAMS	1100
4	7654	MARTIN	1250
4	7521	WARD	1250
6	7934	MILLER	1300
7	7844	TURNER	1500
8	7499	ALLEN	1600
9	7782	CLARK	2450
10	7698	BLAKE	2850
11	7566	JONES	2975
12	7902	FORD	3000
12	7788	SCOTT	3000
14	7839	KING	5000

4. Assign the ranks to employees in ascending order of salary using dense rank and point out the difference.

Code:

```
SELECT DENSE_RANK() OVER (ORDER BY SAL ASC) AS "Regular Rank",
       RANK() OVER (ORDER BY SAL ASC) AS "Dense Rank",
       EMPNO,
       ENAME,
       SAL
FROM SCOTT."EMP";
```

Output:



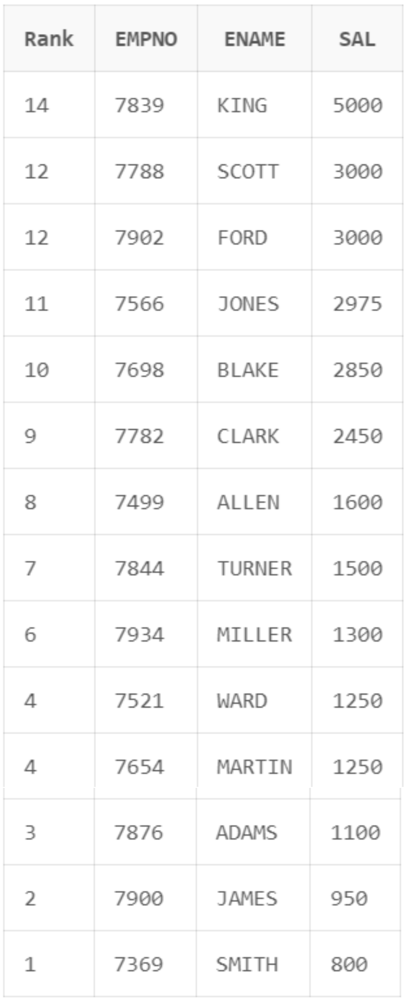
Regular Rank	Dense Rank	EMPNO	ENAME	SAL
1	1	7369	SMITH	800
2	2	7900	JAMES	950
3	3	7876	ADAMS	1100
4	4	7654	MARTIN	1250
4	4	7521	WARD	1250
5	6	7934	MILLER	1300
6	7	7844	TURNER	1500
7	8	7499	ALLEN	1600
8	9	7782	CLARK	2450
9	10	7698	BLAKE	2850
10	11	7566	JONES	2975
11	12	7902	FORD	3000
11	12	7788	SCOTT	3000
12	14	7839	KING	5000

5. Assign the ranks to employees in ascending order of salary but display records in descending order of salary.

Code:

```
WITH RankedEmployees AS (
   SELECT RANK() OVER (ORDER BY SAL ASC) AS "Rank",
        EMPNO,
        ENAME,
        SAL
   FROM SCOTT."EMP"
)

SELECT "Rank",
      EMPNO,
      ENAME,
      SAL
FROM RankedEmployees
ORDER BY SAL DESC;
```

Output:



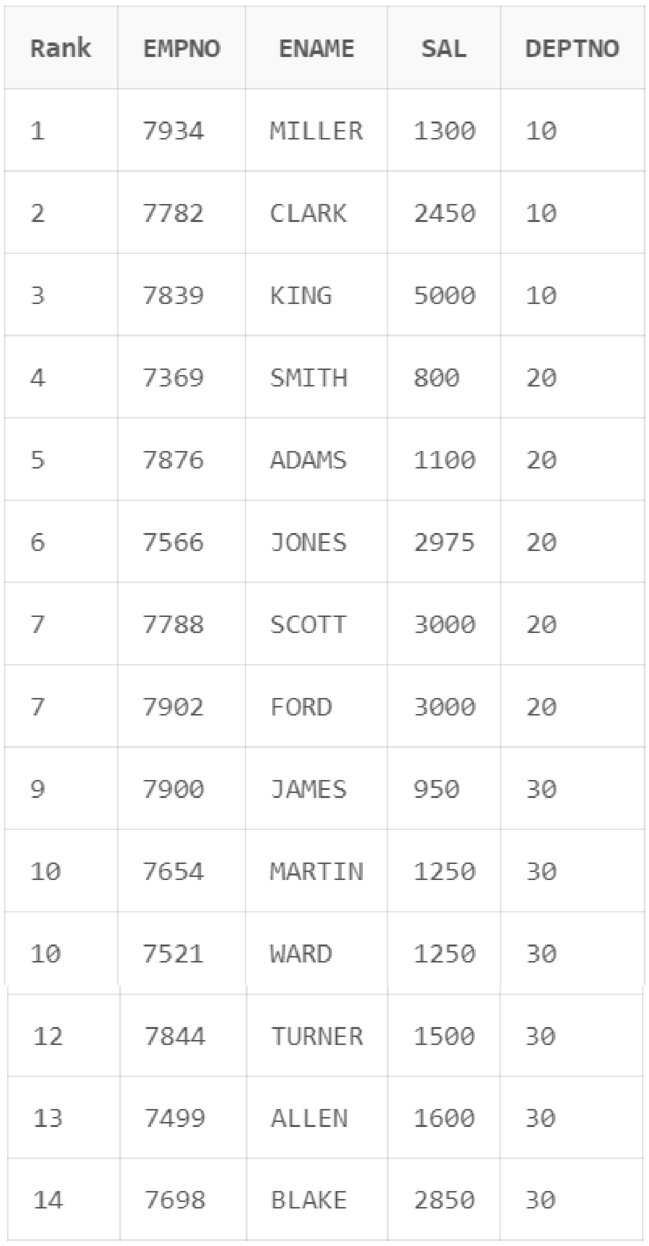
Rank	EMPNO	ENAME	SAL
14	7839	KING	5000
12	7788	SCOTT	3000
12	7902	FORD	3000
11	7566	JONES	2975
10	7698	BLAKE	2850
9	7782	CLARK	2450
8	7499	ALLEN	1600
7	7844	TURNER	1500
6	7934	MILLER	1300
4	7521	WARD	1250
4	7654	MARTIN	1250
3	7876	ADAMS	1100
2	7900	JAMES	950
1	7369	SMITH	800

6. Assign the rank to emp table rows in the ascending order of department and salary.

Code:

```
SELECT RANK() OVER (ORDER BY DEPTNO ASC, SAL ASC) AS "Rank",
       EMPNO,
       ENAME,
       SAL,
       DEPTNO
FROM SCOTT."EMP";
```

Output:



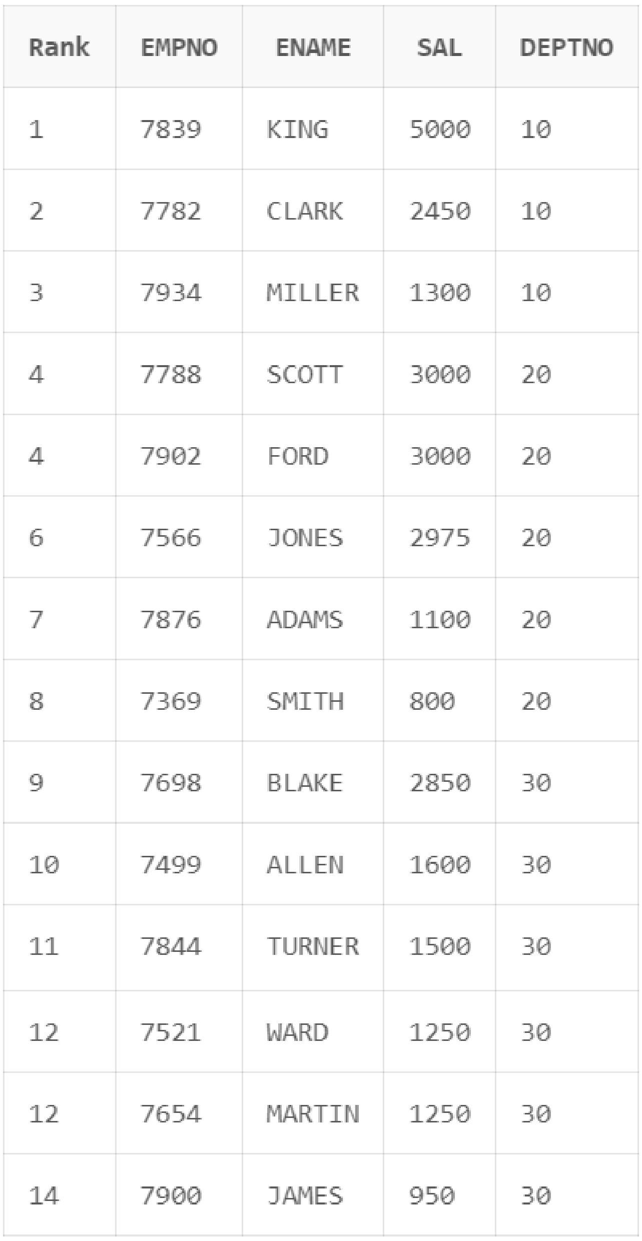
Rank	EMPNO	ENAME	SAL	DEPTNO
1	7934	MILLER	1300	10
2	7782	CLARK	2450	10
3	7839	KING	5000	10
4	7369	SMITH	800	20
5	7876	ADAMS	1100	20
6	7566	JONES	2975	20
7	7788	SCOTT	3000	20
7	7902	FORD	3000	20
9	7900	JAMES	950	30
10	7654	MARTIN	1250	30
10	7521	WARD	1250	30
12	7844	TURNER	1500	30
13	7499	ALLEN	1600	30
14	7698	BLAKE	2850	30

7. Assign the rank to emp table rows in the ascending order of department and descending order of salary.

Code:

```
SELECT RANK() OVER (ORDER BY DEPTNO ASC, SAL DESC) AS "Rank",
       EMPNO,
       ENAME,
       SAL,
       DEPTNO
FROM SCOTT."EMP";
```

Output:



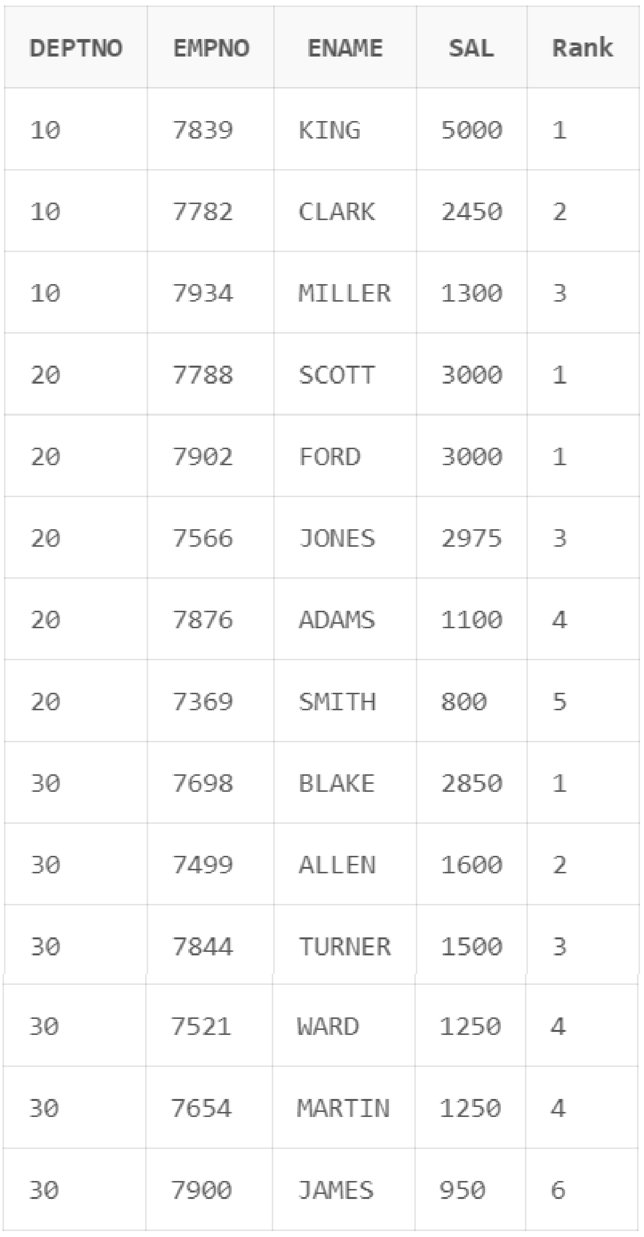
Rank	EMPNO	ENAME	SAL	DEPTNO
1	7839	KING	5000	10
2	7782	CLARK	2450	10
3	7934	MILLER	1300	10
4	7788	SCOTT	3000	20
4	7902	FORD	3000	20
6	7566	JONES	2975	20
7	7876	ADAMS	1100	20
8	7369	SMITH	800	20
9	7698	BLAKE	2850	30
10	7499	ALLEN	1600	30
11	7844	TURNER	1500	30
12	7521	WARD	1250	30
12	7654	MARTIN	1250	30
14	7900	JAMES	950	30

8. Calculate the ranks of employee for each department according to sal.

Code:

```
SELECT DEPTNO,
      EMPNO,
      ENAME,
      SAL,
      RANK() OVER (PARTITION BY DEPTNO ORDER BY SAL DESC) AS
"Rank"
FROM SCOTT."EMP";
```

Output:



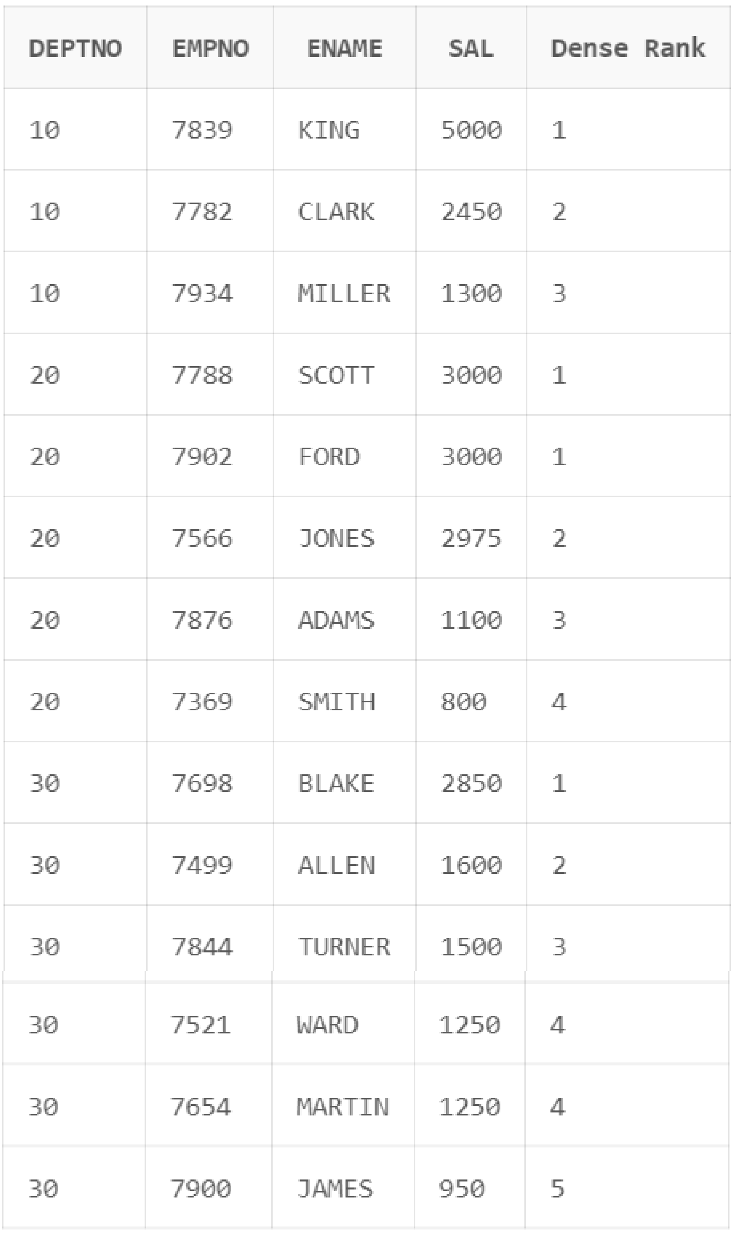
DEPTNO	EMPNO	ENAME	SAL	Rank
10	7839	KING	5000	1
10	7782	CLARK	2450	2
10	7934	MILLER	1300	3
20	7788	SCOTT	3000	1
20	7902	FORD	3000	1
20	7566	JONES	2975	3
20	7876	ADAMS	1100	4
20	7369	SMITH	800	5
30	7698	BLAKE	2850	1
30	7499	ALLEN	1600	2
30	7844	TURNER	1500	3
30	7521	WARD	1250	4
30	7654	MARTIN	1250	4
30	7900	JAMES	950	6

9. For the above query use dense_rank() and point out the difference.

<u>Code:</u>

```
SELECT DEPTNO,
      EMPNO,
      ENAME,
      SAL,
      DENSE_RANK() OVER (PARTITION BY DEPTNO ORDER BY SAL DESC)
AS "Dense Rank"
FROM SCOTT."EMP";
```

<u>Output:</u>



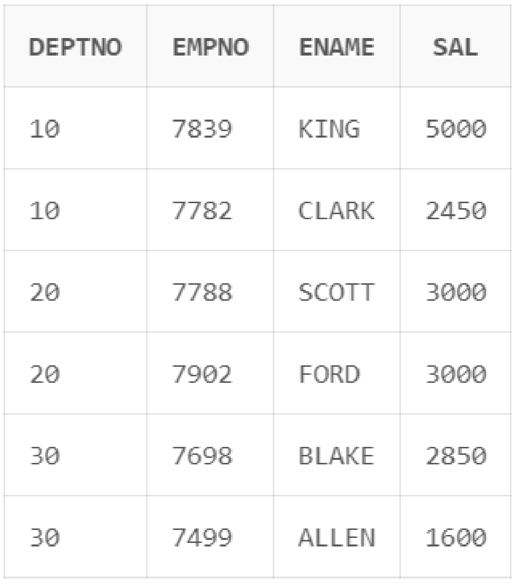
DEPTNO	EMPNO	ENAME	SAL	Dense Rank
10	7839	KING	5000	1
10	7782	CLARK	2450	2
10	7934	MILLER	1300	3
20	7788	SCOTT	3000	1
20	7902	FORD	3000	1
20	7566	JONES	2975	2
20	7876	ADAMS	1100	3
20	7369	SMITH	800	4
30	7698	BLAKE	2850	1
30	7499	ALLEN	1600	2
30	7844	TURNER	1500	3
30	7521	WARD	1250	4
30	7654	MARTIN	1250	4
30	7900	JAMES	950	5

10. Calculate the ranks of employee for each department & display only top 2 high salaried employees for each of them.

Code:

```
WITH RankedEmployees AS (
   SELECT DEPTNO,
          EMPNO,
          ENAME,
          SAL,
          RANK() OVER (PARTITION BY DEPTNO ORDER BY SAL DESC) AS
"Rank"
   FROM SCOTT."EMP"
)

SELECT DEPTNO,
       EMPNO,
       ENAME,
       SAL
FROM RankedEmployees
WHERE "Rank" <= 2;
```

Output:



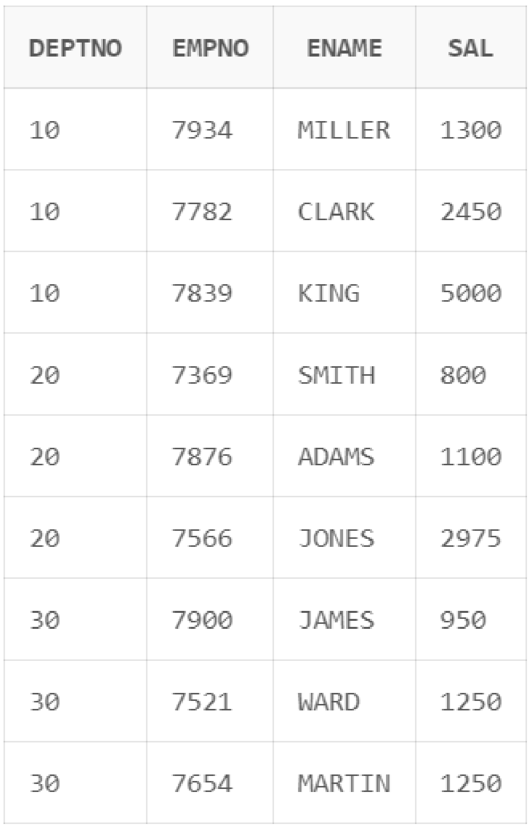
DEPTNO	EMPNO	ENAME	SAL
10	7839	KING	5000
10	7782	CLARK	2450
20	7788	SCOTT	3000
20	7902	FORD	3000
30	7698	BLAKE	2850
30	7499	ALLEN	1600

11. Find out top 3 low salaried employees for each department.

Code:

```
WITH RankedEmployees AS (
   SELECT DEPTNO,
          EMPNO,
          ENAME,
          SAL,
          RANK() OVER (PARTITION BY DEPTNO ORDER BY SAL ASC) AS
"Rank"
    FROM SCOTT."EMP"
)

SELECT DEPTNO,
       EMPNO,
       ENAME,
       SAL
FROM RankedEmployees
WHERE "Rank" <= 3;
```

<u>Output:</u>



DEPTNO	EMPNO	ENAME	SAL
10	7934	MILLER	1300
10	7782	CLARK	2450
10	7839	KING	5000
20	7369	SMITH	800
20	7876	ADAMS	1100
20	7566	JONES	2975
30	7900	JAMES	950
30	7521	WARD	1250
30	7654	MARTIN	1250

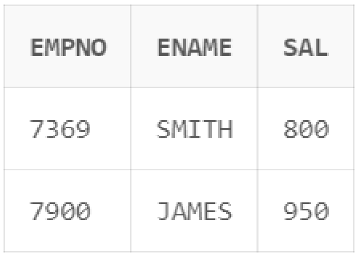
12. Find out top 2 low salaried employees.

<u>Code:</u>

```
WITH RankedEmployees AS (
    SELECT EMPNO,
          ENAME,
          SAL,
          RANK() OVER (ORDER BY SAL ASC) AS "Rank"
    FROM SCOTT."EMP"
)
```

```
SELECT EMPNO,
       ENAME,
       SAL
FROM RankedEmployees
WHERE "Rank" <= 2;
```

Output:



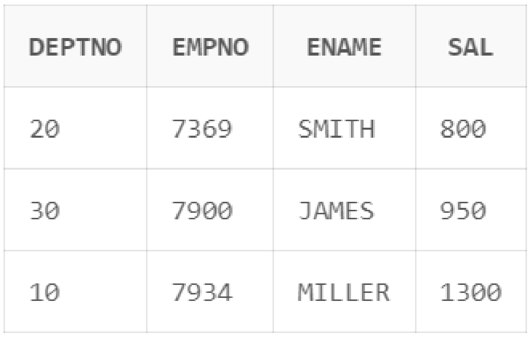
EMPNO	ENAME	SAL
7369	SMITH	800
7900	JAMES	950

13. Find information of employee who is having lowest sal in each department.

Code:

```
SELECT E1.DEPTNO, E1.EMPNO, E1.ENAME, E1.SAL
FROM SCOTT."EMP" E1
INNER JOIN (
   SELECT DEPTNO, MIN(SAL) AS MIN_SAL
   FROM SCOTT."EMP"
   GROUP BY DEPTNO
) E2 ON E1.DEPTNO = E2.DEPTNO AND E1.SAL = E2.MIN_SAL;
```

Output:



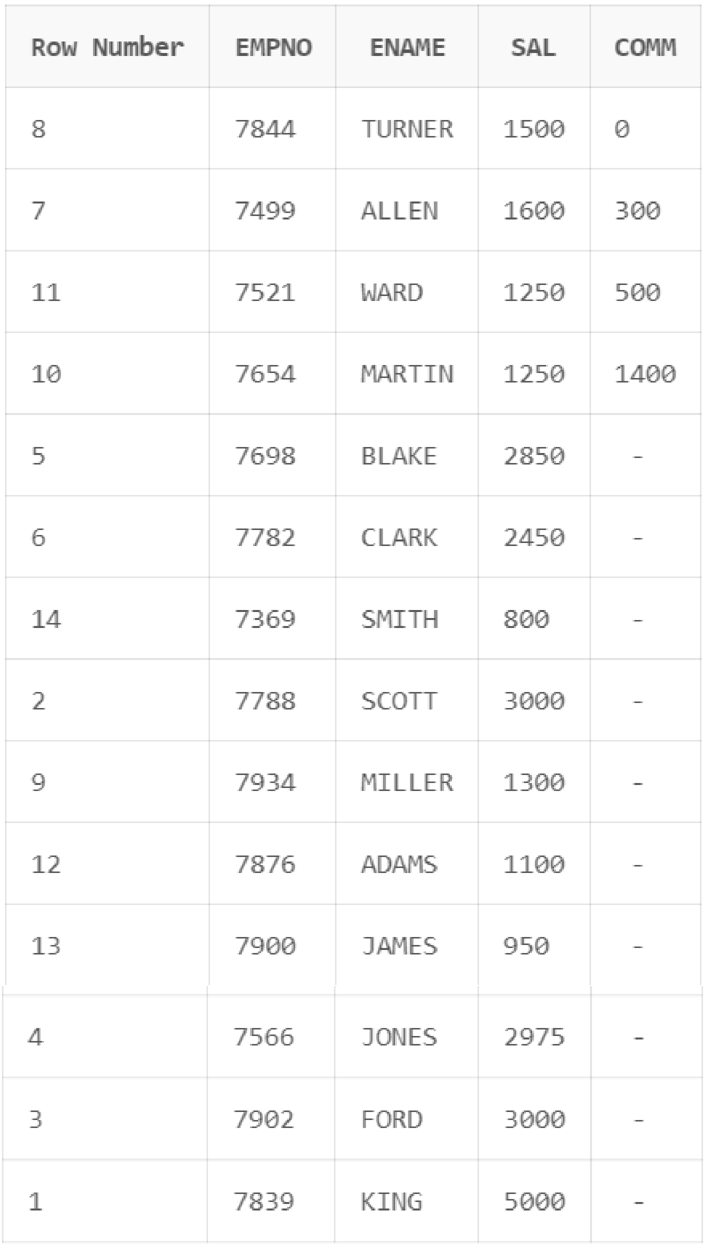
DEPTNO	EMPNO	ENAME	SAL
20	7369	SMITH	800
30	7900	JAMES	950
10	7934	MILLER	1300

14. Assign row numbers in desc order of salary. Display the records in asc order of commission and null values of commission should come last.

Code:

```
SELECT
   ROW_NUMBER() OVER (ORDER BY SAL DESC) AS "Row Number",
   EMPNO,
   ENAME,
   SAL,
   COMM
FROM
   SCOTT."EMP"
ORDER BY
   CASE WHEN COMM IS NULL THEN 2 ELSE 1 END, COMM ASC;
```

Output:



Row Number	EMPNO	ENAME	SAL	COMM
8	7844	TURNER	1500	0
7	7499	ALLEN	1600	300
11	7521	WARD	1250	500
10	7654	MARTIN	1250	1400
5	7698	BLAKE	2850	-
6	7782	CLARK	2450	-
14	7369	SMITH	800	-
2	7788	SCOTT	3000	-
9	7934	MILLER	1300	-
12	7876	ADAMS	1100	-
13	7900	JAMES	950	-
4	7566	JONES	2975	-
3	7902	FORD	3000	-
1	7839	KING	5000	-

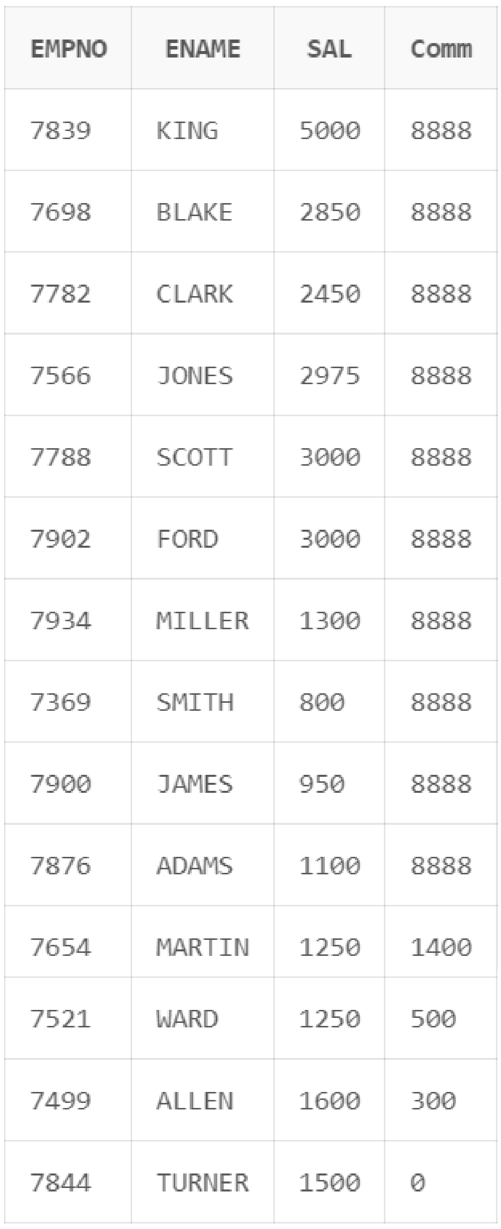
15. Display empno, ename, sal, comm., in desc order of comm. And replace all null values of comm. by 8888.

Code:

```
SELECT
```

```
    EMPNO,
    ENAME,
    SAL,
    COALESCE(COMM, 8888) AS "Comm"
FROM
    SCOTT."EMP"
ORDER BY
    "Comm" DESC;
```

Output:



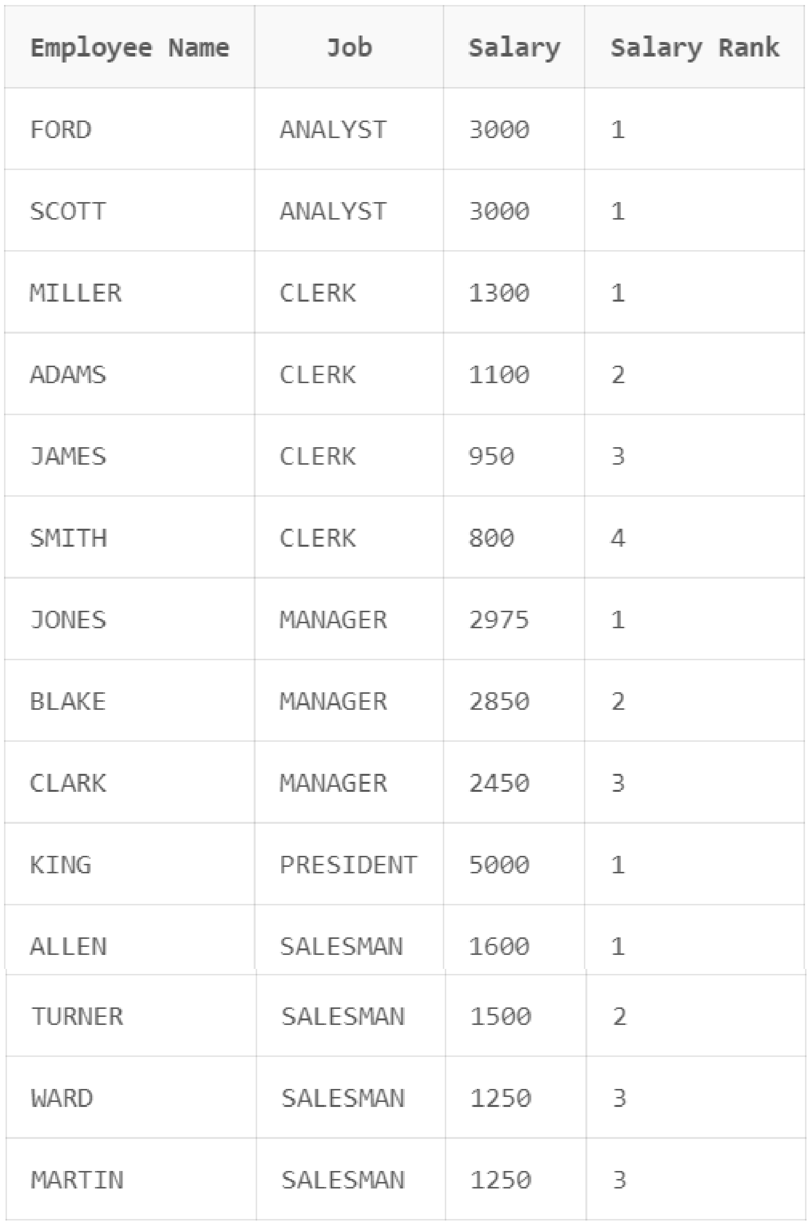
EMPNO	ENAME	SAL	Comm
7839	KING	5000	8888
7698	BLAKE	2850	8888
7782	CLARK	2450	8888
7566	JONES	2975	8888
7788	SCOTT	3000	8888
7902	FORD	3000	8888
7934	MILLER	1300	8888
7369	SMITH	800	8888
7900	JAMES	950	8888
7876	ADAMS	1100	8888
7654	MARTIN	1250	1400
7521	WARD	1250	500
7499	ALLEN	1600	300
7844	TURNER	1500	0

16. Display empname, job & sal. Give ranking to sal job wise.

Code:

```
SELECT
   ENAME AS "Employee Name",
   JOB AS "Job",
   SAL AS "Salary",
   RANK() OVER (PARTITION BY JOB ORDER BY SAL DESC) AS "Salary Rank"
FROM
   SCOTT."EMP"
ORDER BY
   JOB, SAL DESC;
```

Output:



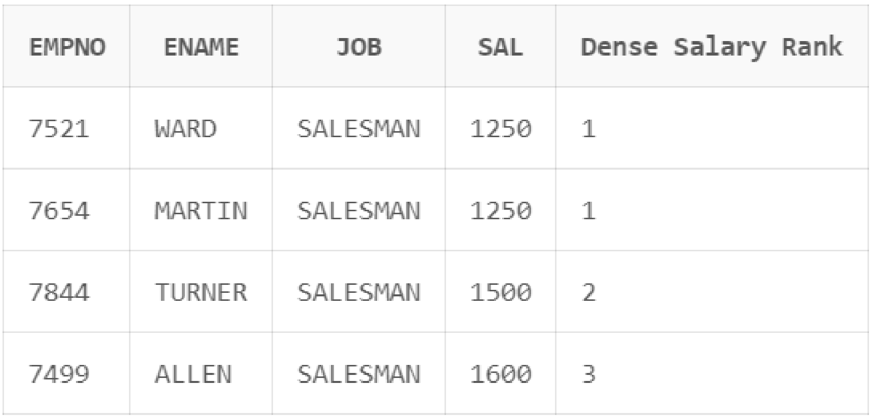
Employee Name	Job	Salary	Salary Rank
FORD	ANALYST	3000	1
SCOTT	ANALYST	3000	1
MILLER	CLERK	1300	1
ADAMS	CLERK	1100	2
JAMES	CLERK	950	3
SMITH	CLERK	800	4
JONES	MANAGER	2975	1
BLAKE	MANAGER	2850	2
CLARK	MANAGER	2450	3
KING	PRESIDENT	5000	1
ALLEN	SALESMAN	1600	1
TURNER	SALESMAN	1500	2
WARD	SALESMAN	1250	3
MARTIN	SALESMAN	1250	3

17. Display the details of all salesman using dense rank on ascending order of salary.

<u>Code:</u>

```
SELECT
   EMPNO,
   ENAME,
   JOB,
   SAL,
   DENSE_RANK() OVER (ORDER BY SAL ASC) AS "Dense Salary Rank"
FROM
   SCOTT."EMP"
WHERE
   JOB = 'SALESMAN'
ORDER BY
   "Dense Salary Rank";
```

<u>Output:</u>



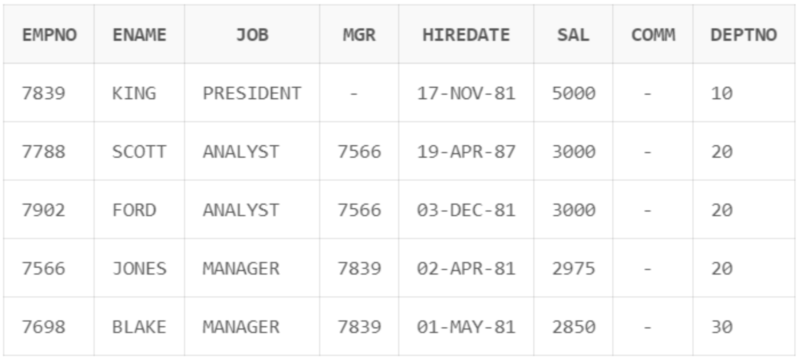
EMPNO	ENAME	JOB	SAL	Dense Salary Rank
7521	WARD	SALESMAN	1250	1
7654	MARTIN	SALESMAN	1250	1
7844	TURNER	SALESMAN	1500	2
7499	ALLEN	SALESMAN	1600	3

18. Display first 5 records of employee in descending order of salary.

<u>Code:</u>

```
SELECT *
FROM SCOTT."EMP"
ORDER BY SAL DESC
FETCH FIRST 5 ROWS ONLY;
```

<u>Output:</u>



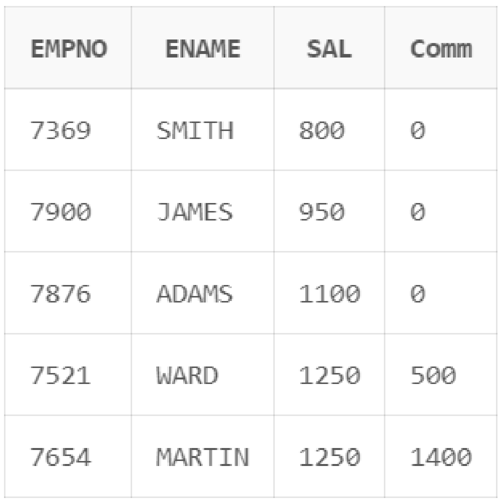
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7788	SCOTT	ANALYST	7566	19-APR-87	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30

19. Display first 5 records of employee in ascending order of salary, replace null values of comm. by zero.

Code:

```
SELECT
   EMPNO,
   ENAME,
   SAL,
   COALESCE(COMM, 0) AS "Comm"
FROM
   SCOTT."EMP"
ORDER BY
   SAL ASC
FETCH FIRST 5 ROWS ONLY;
```

Output:



EMPNO	ENAME	SAL	Comm
7369	SMITH	800	0
7900	JAMES	950	0
7876	ADAMS	1100	0
7521	WARD	1250	500
7654	MARTIN	1250	1400

20. Create weather table with fields month, year and avgtemp. Values could be:

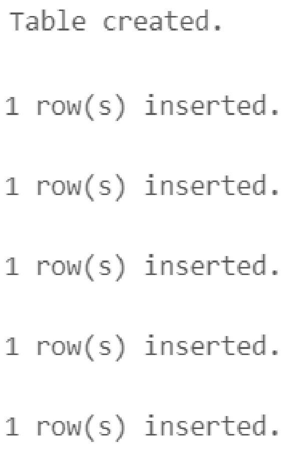
Month	Year	Avgtemp
1	2012	14.5
2	2012	34.5

Put atleast 12 records for 4 different years.

Code:

```
-- Create the weather table
CREATE TABLE weather (
   month INT,
   year INT,
   avgtemp DECIMAL(5, 1)
);

-- Insert sample data for year 2012
INSERT INTO weather (month, year, avgtemp) VALUES (1, 2012, 14.5);
INSERT INTO weather (month, year, avgtemp) VALUES (2, 2012, 34.5);
INSERT INTO weather (month, year, avgtemp) VALUES (3, 2012, 22.0);
INSERT INTO weather (month, year, avgtemp) VALUES (4, 2012, 19.8);
INSERT INTO weather (month, year, avgtemp) VALUES (5, 2012, 26.2);
INSERT INTO weather (month, year, avgtemp) VALUES (6, 2012, 31.5);
INSERT INTO weather (month, year, avgtemp) VALUES (7, 2012, 35.7);
INSERT INTO weather (month, year, avgtemp) VALUES (8, 2012, 33.4);
INSERT INTO weather (month, year, avgtemp) VALUES (9, 2012, 27.8);
INSERT INTO weather (month, year, avgtemp) VALUES (10, 2012, 21.3);
INSERT INTO weather (month, year, avgtemp) VALUES (11, 2012, 15.7);
INSERT INTO weather (month, year, avgtemp) VALUES (12, 2012, 10.1);

-- Insert sample data for year 2013
INSERT INTO weather (month, year, avgtemp) VALUES (1, 2013, 16.2);
INSERT INTO weather (month, year, avgtemp) VALUES (2, 2013, 35.2);
-- Continue inserting data for 2013...

-- Insert sample data for year 2014
INSERT INTO weather (month, year, avgtemp) VALUES (1, 2014, 15.8);
INSERT INTO weather (month, year, avgtemp) VALUES (2, 2014, 34.0);
-- Continue inserting data for 2014...

-- Insert sample data for year 2015
INSERT INTO weather (month, year, avgtemp) VALUES (1, 2015, 16.5);
INSERT INTO weather (month, year, avgtemp) VALUES (2, 2015, 35.6);
```

Output:



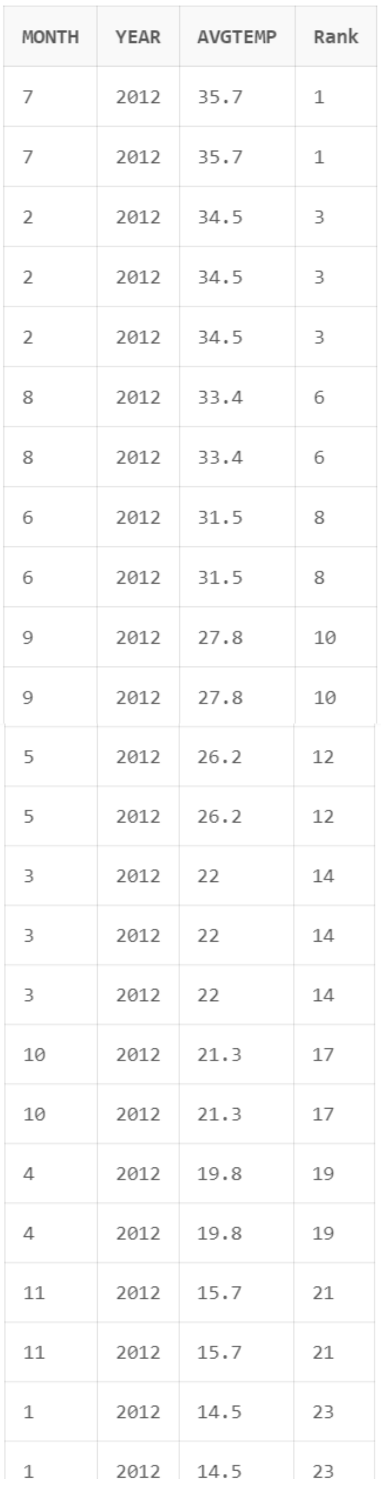
```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

a) Use rank function to display the information of weather in order of hottest to coolest month year to year.

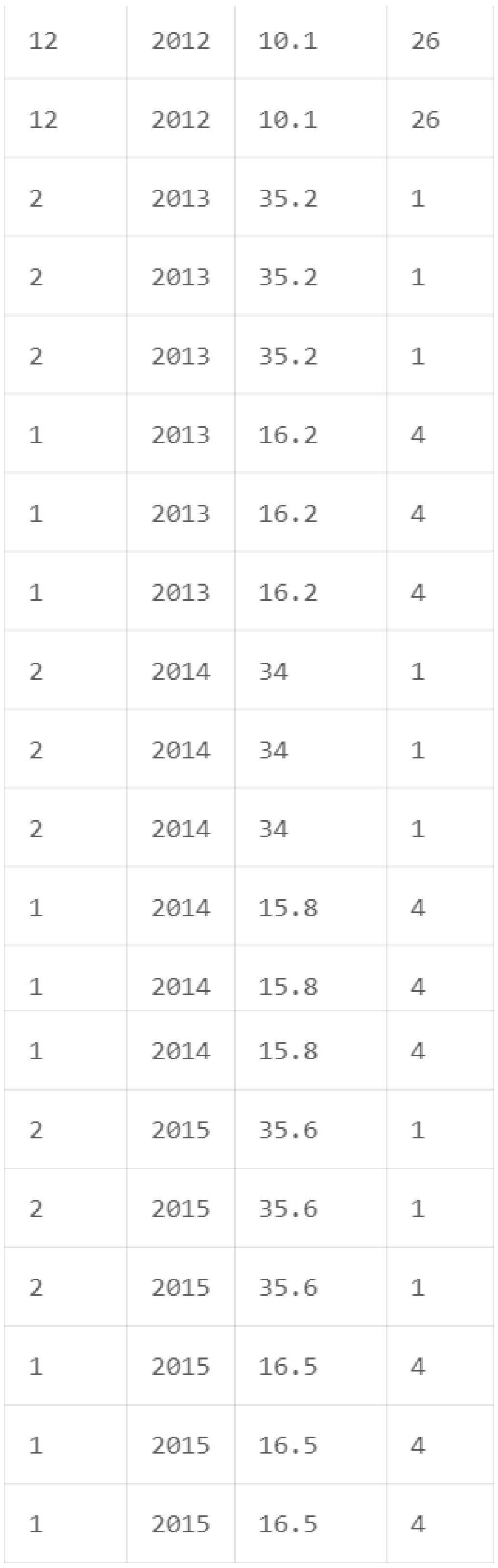
Code:

```
SELECT
   month,
   year,
   avgtemp,
   RANK() OVER (PARTITION BY year ORDER BY avgtemp DESC) AS "Rank"
FROM
   weather
ORDER BY
   year, "Rank";
```

<u>Output:</u>



MONTH	YEAR	AVGTEMP	Rank
7	2012	35.7	1
7	2012	35.7	1
2	2012	34.5	3
2	2012	34.5	3
2	2012	34.5	3
8	2012	33.4	6
8	2012	33.4	6
6	2012	31.5	8
6	2012	31.5	8
9	2012	27.8	10
9	2012	27.8	10
5	2012	26.2	12
5	2012	26.2	12
3	2012	22	14
3	2012	22	14
3	2012	22	14
10	2012	21.3	17
10	2012	21.3	17
4	2012	19.8	19
4	2012	19.8	19
11	2012	15.7	21
11	2012	15.7	21
1	2012	14.5	23
1	2012	14.5	23



12	2012	10.1	26
12	2012	10.1	26
2	2013	35.2	1
2	2013	35.2	1
2	2013	35.2	1
1	2013	16.2	4
1	2013	16.2	4
1	2013	16.2	4
2	2014	34	1
2	2014	34	1
2	2014	34	1
1	2014	15.8	4
1	2014	15.8	4
1	2014	15.8	4
2	2015	35.6	1
2	2015	35.6	1
2	2015	35.6	1
1	2015	16.5	4
1	2015	16.5	4
1	2015	16.5	4

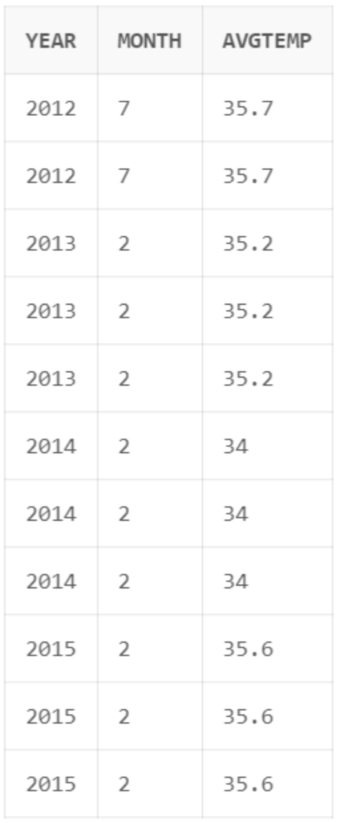
b) Find the hottest month of every year.

<u>Code:</u>

SELECT

```
    year,
    month,
    avgtemp
FROM (
    SELECT
        year,
        month,
        avgtemp,
        RANK() OVER (PARTITION BY year ORDER BY avgtemp DESC) AS "Rank"
    FROM
        weather
) RankedWeather
WHERE
    "Rank" = 1;
```

<u>Output:</u>



YEAR	MONTH	AVGTEMP
2012	7	35.7
2012	7	35.7
2013	2	35.2
2013	2	35.2
2013	2	35.2
2014	2	34
2014	2	34
2014	2	34
2015	2	35.6
2015	2	35.6
2015	2	35.6

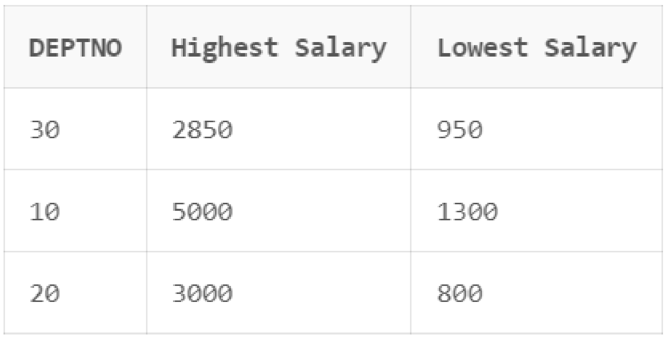
Topic: Analytical functions
(keep…First, keep… Last, Lead(), Lag())

1. Write a query for finding highest and lowest salary of each department.

Code:

```
SELECT
   DEPTNO,
   MAX(SAL) AS "Highest Salary",
   MIN(SAL) AS "Lowest Salary"
FROM
   SCOTT."EMP"
GROUP BY
   DEPTNO;
```

Output:



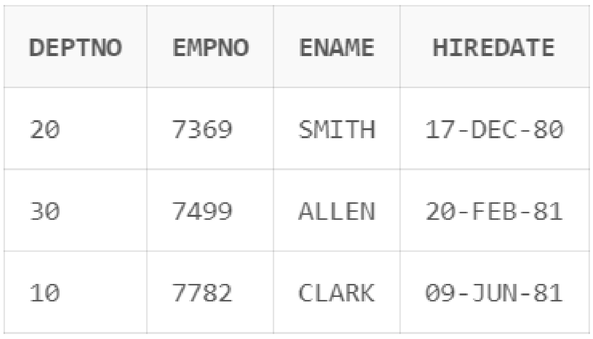
DEPTNO	Highest Salary	Lowest Salary
30	2850	950
10	5000	1300
20	3000	800

2. Write a query to find information of employees who were hired first in each department.

Code:

```
SELECT
   E.DEPTNO,
   E.EMPNO,
   E.ENAME,
   E.HIREDATE
FROM
   SCOTT."EMP" E
WHERE
   (E.DEPTNO, E.HIREDATE) IN (
      SELECT DEPTNO, MIN(HIREDATE)
      FROM SCOTT."EMP" E2
      WHERE E2.DEPTNO = E.DEPTNO
      GROUP BY DEPTNO
   );
```

Output:



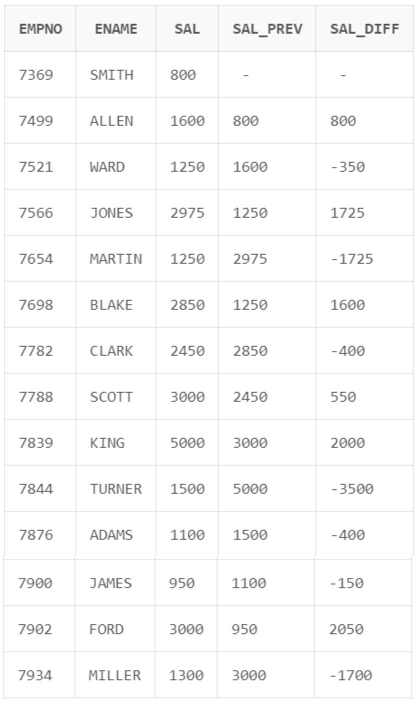
DEPTNO	EMPNO	ENAME	HIREDATE
20	7369	SMITH	17-DEC-80
30	7499	ALLEN	20-FEB-81
10	7782	CLARK	09-JUN-81

3. Write a query which returns the salary from previous row; give the column name as sal_prev. calculate the difference between sal of current row and that of previous row.

Code:

```
SELECT
   EMPNO,
   ENAME,
   SAL,
   LAG(SAL) OVER (ORDER BY EMPNO) AS sal_prev,
   SAL - LAG(SAL) OVER (ORDER BY EMPNO) AS sal_diff
FROM
   SCOTT."EMP"
ORDER BY
   EMPNO;
```

Output:



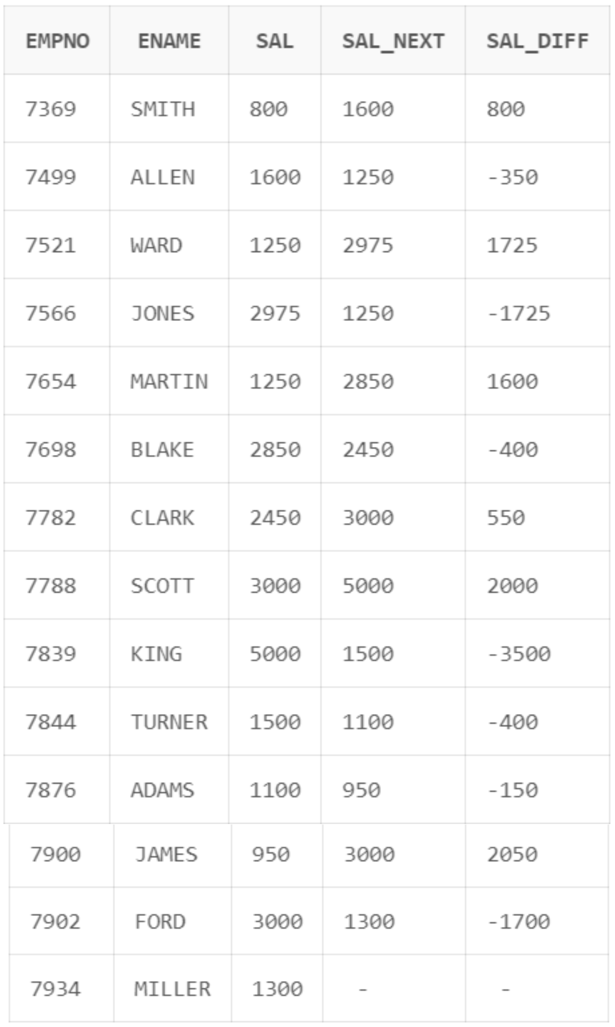
EMPNO	ENAME	SAL	SAL_PREV	SAL_DIFF
7369	SMITH	800	-	-
7499	ALLEN	1600	800	800
7521	WARD	1250	1600	-350
7566	JONES	2975	1250	1725
7654	MARTIN	1250	2975	-1725
7698	BLAKE	2850	1250	1600
7782	CLARK	2450	2850	-400
7788	SCOTT	3000	2450	550
7839	KING	5000	3000	2000
7844	TURNER	1500	5000	-3500
7876	ADAMS	1100	1500	-400
7900	JAMES	950	1100	-150
7902	FORD	3000	950	2050
7934	MILLER	1300	3000	-1700

4. Write a query which returns a salary from next row, name it as sal_next and calculate the diff between the sal of current and following row.

Code:

```
SELECT
   EMPNO,
   ENAME,
   SAL,
   LEAD(SAL) OVER (ORDER BY EMPNO) AS sal_next,
   LEAD(SAL) OVER (ORDER BY EMPNO) - SAL AS sal_diff
FROM
   SCOTT."EMP"
ORDER BY
   EMPNO;
```

Output:



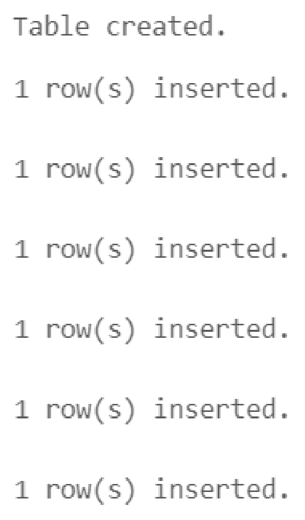
EMPNO	ENAME	SAL	SAL_NEXT	SAL_DIFF
7369	SMITH	800	1600	800
7499	ALLEN	1600	1250	-350
7521	WARD	1250	2975	1725
7566	JONES	2975	1250	-1725
7654	MARTIN	1250	2850	1600
7698	BLAKE	2850	2450	-400
7782	CLARK	2450	3000	550
7788	SCOTT	3000	5000	2000
7839	KING	5000	1500	-3500
7844	TURNER	1500	1100	-400
7876	ADAMS	1100	950	-150
7900	JAMES	950	3000	2050
7902	FORD	3000	1300	-1700
7934	MILLER	1300	-	-

5. Create a table with fields pro_id, order date, quantity. Insert at least 6 records into it.

Code:

```
CREATE TABLE orders (
   pro_id INT,
   order_date DATE,
   quantity INT
);

INSERT INTO orders (pro_id, order_date, quantity) VALUES (101, TO_DATE('2023-09-01',
'YYYY-MM-DD'), 10);
INSERT INTO orders (pro_id, order_date, quantity) VALUES (102, TO_DATE('2023-09-02',
'YYYY-MM-DD'), 15);
INSERT INTO orders (pro_id, order_date, quantity) VALUES (103, TO_DATE('2023-09-03',
'YYYY-MM-DD'), 8);
INSERT INTO orders (pro_id, order_date, quantity) VALUES (104, TO_DATE('2023-09-04',
'YYYY-MM-DD'), 12);
INSERT INTO orders (pro_id, order_date, quantity) VALUES (105, TO_DATE('2023-09-05',
'YYYY-MM-DD'), 20);
INSERT INTO orders (pro_id, order_date, quantity) VALUES (106, TO_DATE('2023-09-06',
'YYYY-MM-DD'), 6);
```

Output:



```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

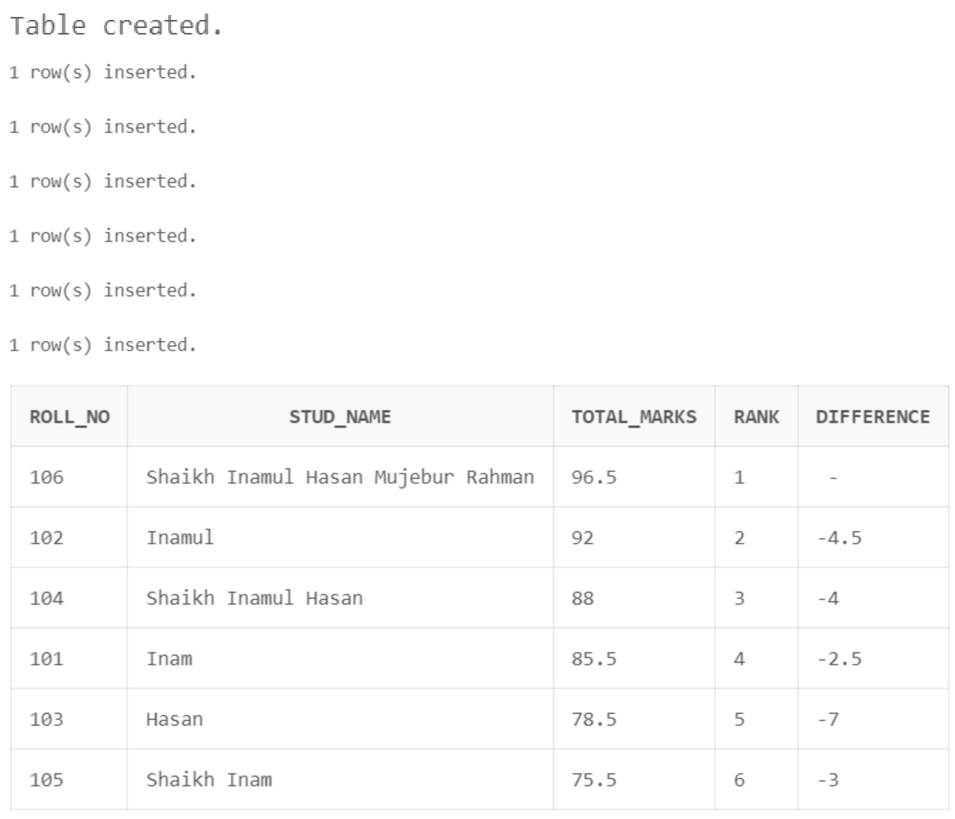
6. Create a table student with roll no, stud_name, total_marks scored in last semester. Insert at least 6 records in it. Find out difference between different rank holders in class.

Code:

```
CREATE TABLE student (
   roll_no INT,
   stud_name VARCHAR(50),
   total_marks DECIMAL(5, 2)
);
```

```
INSERT INTO student (roll_no, stud_name, total_marks) VALUES (101, 'Inam', 85.5);
INSERT INTO student (roll_no, stud_name, total_marks) VALUES (102, 'Inamul', 92.0);
INSERT INTO student (roll_no, stud_name, total_marks) VALUES (103, 'Hasan', 78.5);
INSERT INTO student (roll_no, stud_name, total_marks) VALUES (104, 'Shaikh Inamul Hasan', 88.0);
INSERT INTO student (roll_no, stud_name, total_marks) VALUES (105, 'Shaikh Inam', 75.5);
INSERT INTO student (roll_no, stud_name, total_marks) VALUES (106, 'Shaikh Inamul Hasan
Mujebur Rahman', 96.5);

SELECT
   roll_no,
   stud_name,
   total_marks,
   RANK() OVER (ORDER BY total_marks DESC) AS rank,
   total_marks - LAG(total_marks) OVER (ORDER BY total_marks DESC) AS difference
FROM
   student
ORDER BY
   rank;
```

Output:



```
Table created.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

ROLL_NO	STUD_NAME	TOTAL_MARKS	RANK	DIFFERENCE
106	Shaikh Inamul Hasan Mujebur Rahman	96.5	1	-
102	Inamul	92	2	-4.5
104	Shaikh Inamul Hasan	88	3	-4
101	Inam	85.5	4	-2.5
103	Hasan	78.5	5	-7
105	Shaikh Inam	75.5	6	-3