Module 02: - OLAP with Oracle: Analytical Queries

Practical No.2

Aim: - Implementation of Analytical queries like ROLL_UP, CUBE, RANK, DENSE_RANK, LEAD, LAG, FIRST and LAST.

Objective: - To learn analytical queries.

Theory:

Analytical functions: - Analytic functions compute an aggregate value based on a group of rows. They differ from aggregate functions in that they return multiple rows for each group.

Syntax: -

analytic_function([arguments])OVER(analytic_clause
)

The analytic_clause breaks down into the following optional elements.

[query_partition_clause][order_by_clause]

Implementation: -

ROLL_UP: - The ROLLUP is an extension of the GROUP BY clause. The ROLLUP calculates multiple levels of subtotals across a group of columns (or dimensions) along with the grand total. the ROLLUP extension produces group subtotals from right to left and a grand total. If "n" is the number of columns listed in the ROLLUP, there will be n+1 levels of subtotals.

Step 1: Create table employee with fields emp_no, dep_no, emp_name, dob, salary, comm, job.

Query: -

```
CREATE TABLE employee8 (
emp_no NUMBER(10),
dep_no NUMBER(10),
emp_name VARCHAR(25),
dob DATE,
salary NUMBER(10),
comm NUMBER(10),
job VARCHAR(25)
);
```

Output: -

```
SQL> CREATE TABLE employee6
2 (
3 emp_no NUMBER(10),
4 dep_no NUMBER(10),
5 emp_name VARCHAR(25),
6 dob DATE,
7 salary NUMBER(10),
8 comm NUMBER(10),
9 job VARCHAR(25)
10 );
Table created.
SQL>
```

Step 2: - Insert data into the above table

Query: -

insert into employee8 VALUES(101,10, 'TEJAS', TO DATE('12/01/82', 'DD/ MM/YYYY'),22000,1000,'CLERK'); insert into employee8 VALUES(102,10,'SACHIN',TO DATE('12/02/83','D D/MM/YYYY'),52000,2000,'CLERK'); insert into employee8 VALUES(103,10,'MANISH',TO_DATE('12/03/84','D D/MM/YYYY'),25000,5000,'CLERK'); insert into employee8 VALUES(104,20,'VANCHITA',TO DATE('12/04/82' ,'DD/MM/YYYY'),35000,5000,'MANAGER'); insert into employee8 VALUES(105,20,'SWARUPA',TO_DATE('12/05/86', 'DD/MM/YYYY'),45000,6000,'MANAGER'); insert into employee8 VALUES(106,20,'ASHWINI',TO DATE('12/06/87',' DD/MM/YYYY'),26000,2000,'MANAGER'); insert into employee8 VALUES(107,10,'SUDIP',TO_DATE('12/07/88','DD/ MM/YYYY'),55000,6000,'MANAGER'); insert into employee8 VALUES(108,20,'AKSHAY',TO DATE('12/08/89',' DD/MM/YYYY'),35000,6000,'CLERK');

Output: -



Step 3: - Display the data from employee7

Query: -

select * from employee8;

Output: -

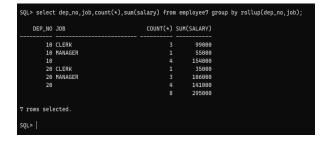
EMI			EMP_NAME			COMM
эов						
CLERK	101		TEJAS		2 22000	1000
					3 52000	2000
			MANISH		4 25000	5000
JOB EM	_NO	DEP_NO	EMP_NAME		SALARY	COMM
Јов						
HANAGE	1.04	20	VANCHITA	12-APR-8	2 35000	5000
HANAGE	105		SWARUPA		6 45000	6000
HANAGEI	106		ASHWINI		7 26000	2000
	_NO	DEP_NO	EMP_NAME			сомм
JOB						
MANAGEI	107				8 55000	6000
					9 35000	6000

Q. Display dep_no, job, job count, sum of salary group them up using a roll up function in the order of dep_no, job.

Query: -

select dep_no,job,count(*),sum(salary) from
employee7 group by rollup(dep_no,job);

Output: -



Name: Arif Shaikh

MCAL13 Advanced Database Management System Lab

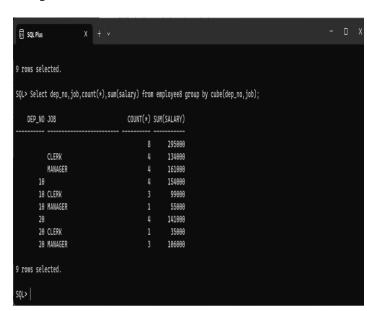
CUBE: - In addition to the subtotals generated by the ROLLUP extension, the CUBE extension will generate subtotals for all combinations of the dimensions specified. If "n" is the number of columns listed in the CUBE, there will be 2ⁿ subtotal combinations.

Q. Display dep_no,job,job count, sum of salary group them up using a cube function in the order of dep_no,job.

Query: -

Select dep_no,job,count(*),sum(salary) from employee8 group by cube(dep_no,job);

Output: -

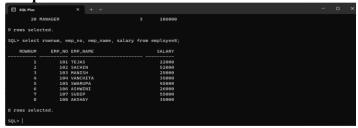


Q. Display emp_no, emp_name and salary from employee table and give numbers to each row.

Query: -

select rownum, emp_no, emp_name, salary from employee8;

Output: -



FYMCA Sem – I RollNo:39

Q. Display emp_no,emp_name and salary from employee table and give numbers to each row.

Query:-

Select rownum,emp_no,emp_name,salary from employee8;

Output:-

SQL> Select	rownum,emp	o_no,emp_name,salary	from employee8;
ROWNUM	EMP_NO	EMP_NAME	SALARY
1	101	TEJAS	22000
2	102	SACHIN	52000
3	103	MANISH	25000
4	104	VANCHITA	35000
5	105	SWARUPA	45000
6	106	ASHWINI	26000
7	107	SUDIP	55000
8	108	AKSHAY	35000
8 rows sele	cted.		

Query:-

select rownum,emp_no,emp_name,salary from
employee8 order by salary;

Output: -

SQL> selec	t rownum,em	o_no,emp_name,salary	from employee8 o	order by salary ;
ROWNUM	EMP_NO	EMP_NAME	SALAR	(
1	101	TEJAS	22000	9
3	103	MANISH	25006	9
6	106	ASHWINI	26006	3
8	108	AKSHAY	35006	9
4	104	VANCHITA	35006	9
5	105	SWARUPA	45000	3
2	102	SACHIN	52000	9
7	107	SUDIP	55000	9
8 rows sel	ected.			

Q. Display emp_no, emp_name and descending order of salary from employee table and give numbers to each row.

Query:-

select rownum,emp_no,emp_name,salary from employee8 order by salary desc;

Output:-

SQL> select		o_no,emp_name,salary	from employee8 order	by salary desc;
ROWNUM	EMP_NO	EMP_NAME	SALARY	
7	107	SUDIP	55000	
2	102	SACHIN	52000	
5	105	SWARUPA	45000	
8	108	AKSHAY	35000	
4	104	VANCHITA	35000	
6	106	ASHWINI	26000	
3	103	MANISH	25000	
1	101	TEJAS	22000	
8 rows selec	ted.			

Name: Arif Shaikh

ROW_NUMBER():- It is an analytical function and unlike NTILE this function assigns a unique sequential number to each row of the result set.

Q. Use Row_number() analytical function to give numbering according to salary.

Query:-

select row_number() over(order by salary),emp_no,emp_name,salary from employee order by salary desc;

Output:-

SQL> select row_number() over(order	by salary),emp_no,emp	_name,salary from employee8 order by	salary desc;
ROW_NUMBER()OVER(ORDERBYSALARY)	EMP_NO EMP_NAME	SALARY	
8	107 SUDIP	55000	
7	102 SACHIN	52000	
6	105 SWARUPA	45000	
4	108 AKSHAY	35000	
5	104 VANCHITA	35000	
3	106 ASHWINI	26000	
2	103 MANISH	25000	
1	101 TEJAS	22000	
8 rows selected.			

RANK():- The RANK() function is an analytic function that calculates the rank of a value in a set of values. The RANK() function returns the same rank for the rows with the same values.

Q. Display eno, ename and salary from employee table and rank them according to ascending order of salary.

Query:-

select emp_no,emp_name,salary,rank() over(order by salary) from employee8 order by salary;

Output:-

	12 1 12 1 21 0	over(order by salam SALARY RANK()OVER(OF	ry) from employee8 order by salary; RDERBYSALARY)
101	TEJAS	22000	1
103	MANISH	25000	2
106	ASHWINI	26000	
108	AKSHAY	35000	4
104	VANCHITA	35000	4
105	SWARUPA	45000	6
102	SACHIN	52000	7
107	SUDIP	55000	8
8 rows sele	ected.		

DENSE_RANK():- The DENSE_RANK() function is an analytic function that calculates the rank of a value in a set of values. The DENSE_RANK() function returns the same rank for the rows with the same values. DENSE_RANK() does not have anygap in rankings

Q. Display eno, name, salary from employee table and rank them according to ascending order of salary using dense_rank()

Query:-

select emp_no, emp_name,dense_rank() over(order by salary) from employee8 order by salary;

Output:-

```
      SQL> select emp_no, emp_name,dense_rank() over(order by salary) from employee8 order by salary;

      EMP_NO EMP_NAME
      DENSE_RANK()OVER(ORDERBYSALARY)

      101 TEJAS
      1

      103 MANISH
      2

      106 ASHMINI
      3

      108 AKSHAY
      4

      104 VANCHITA
      4

      105 SWARUPA
      5

      107 SUDIP
      7

      3 rows selected.
```

Q. Display three highest salaried person.

Query:-

select * from (select dense_rank() over(order by salary
desc) top,emp_name,emp_no,salary from
employee8)where top<=3;</pre>

Output:-

SQL> select * from (select de	nse_rank() over(order	by salary	desc)top,emp_	name,emp_n	o,salary	from	employee8)wher	e top	ζ=	3;
TOP EMP_NAME	EMP_NO	SALARY								
1 SUDIP	107	55000								
2 SACHIN	102	52000								
3 SWARUPA	105	45000								

Q. Partition by department_number

Query:-

select emp_no,dep_no,salary,comm,rank()
over(partition by dep_no order by salary) as Rank
from employee8;

Name: Arif Shaikh

Output:-

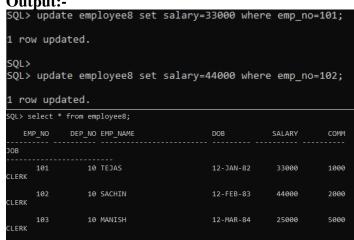
SQL> select	emp_no,dep_r	no,salary,co	mm,rank()	over(partition	dep_no	order	by	salary)	Rank	from	employee8;
EMP_NO	DEP_NO	SALARY	COMM	RANK							
101	10	22000	1000								
103	10	25000	5000								
102	10	52000	2000								
107	10	55000	6000								
106	20	26000	2000								
108	20	35000	6000								
104	20	35000	5000								
105	20	45000	6000								
8 rows sele	cted.										

Q. Update employee salary

Query:-

update employee8 set salary=33000 where emp_no=101; update employee8 set salary=44000 where emp_no=102;

Output:-



Q. Inserting null value in commission

Query:-

insert into employee8 values(101,10,'TEJAS',TO_DATE('12/01/82','DD/M M/YYYY'),22000,null,'CLERK');

Q. Replace null values of commission by 1000

Query:-

Selectemp_no,emp_name,salary,nvl(comm,1000)new _comm from employee8 order by comm desc;

Output:-

EMP_NO EMP_NAME	SALARY	NEW_COMM		
101 TEJAS	22000	1000		
105 SWARUPA	45000	6000		
108 AKSHAY	35000	6000		
107 SUDIP	55000	6000		
104 VANCHITA	35000	5000		
103 MANISH	25000	5000		
106 ASHWINI	26000	2000		
102 SACHIN	44000	2000		
101 TEJAS	33000	1000		

Q. Display details of employee and give ranking only for employee in dept no 10

Query:-

select dense_rank()over(partition by dep_no order by salary)Rank,dep_no,emp_name,salary from employee8 where dep_no=10;

Output:-

SQL> select (dense_rank(over(partition by)	dep_no order by	salary)Rank,dep_no,emp_name,salary from employee8 where dep_no=10;
RANK	DEP_NO EI	MP_NAME	SALAR	4
	10 T	EJAS	22000	9
2	10 M	ANISH	25000	9
	10 T	EJAS	33000	9
4	10 S	ACHIN	44000	9
5	10 SI	JDIP	55000	9

Q. Display name, job & salary and Rank the salary job wise

Query:-

select dense_rank()over(partition by job order by salary)Rank,job,emp_name,salary from employee8 order by job;

Output:-

SQL> :	select	dense_rank()over(partiti	on by job order by	salary)Rank,job,emp_name,salary	from employee8	order by	job;
	RANK	ЈОВ	EMP_NAME	SALARY			
		CLERK	TEJAS	22000			
		CLERK	MANISH	25000			
		CLERK	TEJAS	33000			
		CLERK	AKSHAY	35000			
		CLERK	SACHIN	44000			
		MANAGER	ASHWINI	26000			
		MANAGER	VANCHITA	35000			
		MANAGER	SWARUPA	45000			
		MANAGER	SUDIP	55000			
9 row	s sele	ected.					

Q. Display information of employee & rank them for employee working as manager

Query:-

select dense_rank()over(partition by job order by salary)Rank,job,emp_name,salary from employee8 where job='MANAGER';

Output:-

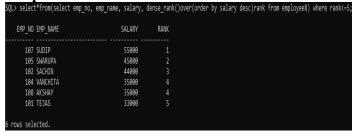
SQL> select	t dense_rank()over(partiti	on by job order by salary)Rank	job,emp_name,salary fr,	om employee8 where	job='MANAGER';
RANK	JOB	EMP_NAME	SALARY		
1	MANAGER	ASHWINI	26000		
_	MANAGER	VANCHITA	35000		
3	MANAGER	SWARUPA	45000		
4	MANAGER	SUDIP	55000		
	•				

Q. Display first five records of employee in descending order of salary

Query:-

select*from(select emp_no, emp_name, salary,
dense_rank()over(order by salary desc)rank from
employee8) where rank<=5;</pre>

Output:-



LEAD() and **LAG():-** The LEAD function is used to access data from SUBSEQUENT rows along with data from the current row. The LAG function is used to access data from PREVIOUS rows along with data from the current row

Syntax:- LEAD(expression,[offset],[default]) over ([query_partition_clause] order_by_clause)

Q. Display Employee details using Lead() analytical function.

Query:-

select emp_no,dob,lead(dob,1)over(order by dob)as
"next" from employee8;

Output:-

Q. Display Employee details using Lag() analytical function.

Query:-

select emp_no,dob,lag(dob,1)over(order by dob)as "previous" from employee8;

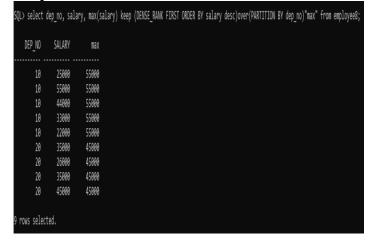
Output:-

FIRST and LAST: FIRST is an analytic function as the name suggests is used to provide the value of the first row in an ordered set of rows.LAST is also an analytical function which is used to get the value of the last row in an ordered set of rows.

Query:-

select dep_no, salary, max(salary) keep (DENSE_RANK FIRST ORDER BY salary desc)over(PARTITION BY dep_no)"max" from employee8;

Output:-

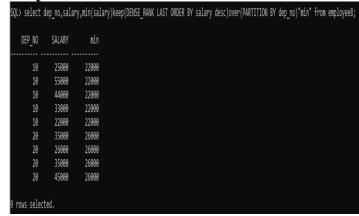


MCAL13 Advanced Database Management System Lab

Query:-

select dep_no,salary,min(salary)keep(DENSE_RANK LAST ORDER BY salary desc)over(PARTITION BY dep_no)"min" from employee8;

Output:-



Windowing Functions: -

The ROWS PRECEDING and ROWS FOLLOWING functions are used to specify rows before and after the current row in a window function:

- **ROWS PRECEDING**: Specifies a fixed number of rows before the current row
- **ROWS FOLLOWING**: Specifies a fixed number of rows after the current row
- Q. Display empno,name,salary,sum of salary dept wise and display the salary of current and previous 2 records.

Output:

109 ankita

111 sheela

11 rows selected.

SQL> select empno,ename,sal,deptno,sum(sal) over (partition by deptno order by d eptno rows 2 PRECEDING>total from emp1 order by deptno; EMPNO ENAME SAL DEPTNO TOTAL 35000 35000 105 kalpita 104 suman 95000 130000 1 103 kajal 65000 195000 102 julie 55000 1 215000 101 john 25000 145000 108 neha 87000 87000 107 sonal 65000 152000 25000 177000 106 darshana 20000 110 priya 20000

25000

45000

70000

Q. Display empno,name,salary,sum of 3 earlier rows and 1 next row dept wise.

Output:

\$200 select empno,ename,sal,deptno,sum(sal) over (partition by deptno order by deptno rows BETWEEN 3 PRECEDING AND 1 FOLLOWING)total from emp1 order by deptno;

EMPNO	ENAME	SAL	DEPTNO	TOTAL
105 104 103 102 101 108 107	kalpita suman kajal julie john neha sonal darshana	35000 95000 65000 55000 25000 87000 65000	1 1 1 1 1 1 2 2 2	13000 195000 250000 275000 240000 152000 177000
110 109	priya ankita	20000 25000	3	45000 70000
111	sheela	25000	3	70000

11 rows selected.

Q. Display empno, ename, deptno, sal and sum of salary for 1 earlier row and 1 next row.

Output:

SQL> select empno,ename,sal,deptno,sum(sal) over (partition by deptno order by deptno rows BETWEEN 1 PRECEDING AND 1 FOLLOWING)total from emp1 order by deptno;

EMPNO	ENAME	SAL	DEPTNO	TOTAL
104 103 102 101 108 107 106 110	kalpita suman kajal julie john neha sonal darshana priya ankita sheela	35000 95000 65000 55000 25000 87000 65000 25000 25000	1 1 1 1 1 2 2 2 3 3	130000 195000 215000 145000 80000 152000 177000 90000 45000 70000

11 rows selected.

Q.Display empno, ename, deptno, sal and sum of salary for 1 following and 3 following dept wise.

Output:

SQL/ select empno,ename,sal,deptno,sum(sal/ over (partition by deptno order by d eptno rows BETWEEN 1 FOLLOWING AND 3 FOLLOWING/total from emp1 order by deptno;

-	DETHEEN I TODAY			•
EMPNO	ENAME	SAL	DEPTNO	TOTAL
	kalpita	35000	1	215000
104	suman	95000	1	145000
103	kajal	65000	1	80000
102	julie	55000	1	25000
101	.john	25000	1	
	neha	87000	2	90000
107	sonal	65000	2 2 2 3 3 3	25000
106	darshana	25000	2	
110	priya	20000	3	50000
	ankita	25000	3	25000
111	sheela	25000	3	
11 rows se	lected.			

Q. Display empno,ename,deptno,sal and sum of salary for 3 preceding and 1 preceding row dept wise.

Output:

SQL) select empno,ename,sal,deptno,sum(sal) over (partition by deptno order by d eptno rows BETWEEN 3 PRECEDING AND 1 PRECEDING)total from emp1 order by deptno;

EMPNO	ENAME	SAL	DEPTNO	TOTAL
105	kalpita	35000	1	
104		95000	ī	35000
103	kajal	65000	ī	130000
102	julie	55000	1	195000
101	john	25000	1	215000
108	neha	87000	2	
107	sonal	65000	2	87000
	darshana	25000	2	152000
110		20000	3	
	ankita	25000		20000
111	sheela	25000	3	45000
11 rows se	lected.			

Q.Display empno, ename, deptno, sal and sum of salary for all preceding and current row dept wise.

Output:

 $\mbox{SQL}\mbox{$\rangle$ select empno,ename,sal,deptno,sum(sal) over (partition by deptno order by deptno rows BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)total from emp1 order by deptno;$

EMPNO	ENAME	SAL	DEPTNO	TOTAL
104 103 102 101 108 107 106 110	kalpita suman kajal julie john neha sonal darshana priya ankita sheela	35000 95000 65000 55000 25000 87000 65000 25000 25000 25000	1 1 1 1 1 2 2 2 2 3 3 3	35000 130000 195000 250000 275000 87000 152000 177000 20000 45000

11 rows selected.

Q. Write a query to find out information of employee who were hired first in every department.

Output:

SQL> select deptno,ename,sal,min(hire) keep(dense_rank FIRST order by hire) over (partition by deptno) from emp1;

DEPTNO ENAME	SAL MIN(HIRE)KEEP(DENSE_
1 kalpita 1 suman 1 kajal 1 julie 1 john 2 neha 2 sonal 2 darshana 3 priya 3 ankita 3 sheela	35000 01-aug-2014 95000 01-aug-2014 65000 01-aug-2014 55000 01-aug-2014 25000 01-aug-2014 87000 01-aug-2014 65000 01-aug-2014 25000 01-aug-2014 25000 01-aug-2014 25000 01-aug-2014 25000 01-aug-2014

Q. write a query which returns salary from previous row. Give colname as sal_previous.

Output:

SQL> select empno, ename, job, sal, lag(sal, 1,0) over (order by sal) as sal_prev fro

EMPNO	ENAME	JOB	SAL	SAL_PREV
110	priya	programmer	20000	0
111	sheela	programmer	25000	20000
101	.john	manager	25000	25000
109	ankita	manager	25000	25000
106	darshana	designer	25000	25000
105	kalpita	manager	35000	25000
102	julie	designer	55000	35000
103	kajal	manager	65000	55000
	sonal	manager	65000	65000
108	neha	programmer	87000	65000
104	suman	designer	95000	87000

Q. calculate difference between salary of current row with previous row.

Output:

Name: Arif Shaikh

SQL' select empno.ename,sal,lag(sal,1,0) over (order by sal) as sal_prev,sal-lag (sal,1,0) over (order by sal) as diff from emp1;

EMPNO	ENAME	SAL	SAL_PREU	DIF
110	priya	20000	0	2000
	sheela	25000	20000	500
101	.john	25000	25000	
109	ankita	25000	25000	
106	darshana	25000	25000	
105	kalpita	35000	25000	1000
102	julie	55000	35000	2000
103	kajal	65000	55000	1000
107	sonal	65000	65000	
108	neha	87000	65000	2200
104	suman	95000	87000	800

110 priya 101 john 106 darshana 111 sheela 109 ankita 105 kalpita 102 julie 107 sonal 103 kajal 108 neha 104 suman 25000 25000 20000 25000 25000 25000 35000 25000 65000 65000 25000 25000 35000 55000 65000

11 rows selected.

Q. write a query which returns the sal from next row. Give colname as sal next.

Output:

SQL> select empno,ename,job,sal,lead(sal,1,0) over (order by sal) as sal_next fr om emp1 order by sal;

EMPNO	ENAME	JOB	SAL	SAL_NEXT
	priya	programmer	20000	25000
-	sheela	programmer	25000	25000
	john	manager	25000	25000
109	ankita	manager	25000	25000
106	darshana	designer	25000	35000
105	kalpita	manager	35000	55000
102	julie	designer	55000	65000
103	kajal	manager	65000	65000
107	sonal	manager	65000	87000
108	neha	programmer	87000	95000
104	suman	designer	95000	0

Q. Calculate the difference between current sal and its following salary.

Output:

SQL) select empno,ename,sal,lag(sal,1,0) over (order by sal) as sal_prev,sal-lead(sal,1,0) over (order by sal) as diff from emp1;

EMPNO	ENAME	SAL	SAL_PREV	DIFF
	priya	20000	0	-5000
	sheela	25000	20000	0
	john	25000	25000	0
109	ankita	25000	25000	0
106	darshana	25000	25000	-10000
105	kalpita	35000	25000	-20000
102	julie	55000	35000	-10000
	kajal	65000	55000	0
107	sonal	65000	65000	-22000
	neha	87000	65000	-8000
104	suman	95000	87000	95000
11 rows se	lected.			

Q. Write a query which returns the difference of salary of current and previous row department wise.

Output:

SQL) select empno,ename,sal,lag(sal,1,0) over (partition by deptno order by sal) as sal_prev,sal-lag(sal,1,0) over (order by sal) as diff from emp1; EMPNO ENAME SAL_PREU DIFF

5000

10000 20000

10000 22000

MCAL13 Advanced Database Management System Lab

Q. create table sales

Query: -

create table sales (time number(4), Region varchar2(7) check(region in('central','east','west')), department varchar2(10) check(department in('pensales','booksales')), profit number(5));

Insert 10 records in sales table.

insert into sales values(2000,'central','pensales',5000); insert into sales values(2001,'central','pensales',6000); insert into sales values(2002,'east','booksales',5500); insert into sales values(2000,'west','pensales',5000); insert into sales values(2001,'east','pensales',5000); insert into sales values(2001,'east','pensales',5500); insert into sales values(2002,'west','booksales',6000); insert into sales values(2000,'central','pensales',7000); insert into sales values(2000,'central','pensales',2000); insert into sales values(2000,'east','pensales',2000); insert into sales values(2000,'east','pensales',2000); insert into sales values(2000,'east','pensales',2000); insert into sales values(2000,'east','pensales',2000); insert into sales values(2000,'east','pensales',12000);

Display data of sales.

Output:

SQL>	select	* from	sales;	
	TIME	REGION	DEPARTMENT	PROFIT
	2000	central	pensales	5000
	2001	central		6000
	2002	east	booksales	5500
	2000	west	pensales	5000
	2003	central	booksales	5000
	2001	east	pensales	5500
	2002	west	booksales	6000
	2000	central	pensales	7000
	2002	central	booksales	2000
	2001	west	booksales	11000
	2000	west	booksales	11000
	TIME	REGION	DEPARTMENT	PROFIT
	2000	east	pensales	2000
	2000	east	booksales	2000
	2000	east	pensales	12000
14 r	ows sel	lected.		

Q. Display the time ,region, department and sum of profit from the sales table and group them in the order of time, region and department.

Output:

SQL> SELECT TIME, REGION, DEPARTMENT, SUM(PROFIT) AS PROFIT FROM SALES GROUP BY TIME, REGION, DEPARTMENT; TIME REGION DEPARTMENT 2003 central booksales 2001 east 2000 west hooksales 2002 west booksales 2002 central booksales 2000 2001 west booksales 2000 central pensales 12000 2001 central pensales 2000 west pensales 5000 2000 east pensales TIME REGION DEPARTMENT 2002 east booksales 12 rows selected.

Q. Display the time ,region, department and sum of profit from the sales table and group them up using a roll up function in the order of time, region and department.

Output:

SQL> select time, region, department, sum (profit) as PROFIT from sales group by ROLLUP(time, region, department);

TIME REGION DEPARTMENT PROFIT

2000 west pensales 5000
2000 west booksales 11000
2000 central pensales 12000
2000 central pensales 12000
2000 central pensales 5500
2001 east pensales 5500
2001 east booksales 5500
2001 west booksales 11000
2001 central pensales 6000
2001 central pensales 6000
2001 central 6000
2001 central 6000
2002 central 6000
2003 central 5000
2004 central 5000
2005 central 5000

Q. Perform a partial rollup on the sales table by grouping the time first and the region and department together.

Output:

 SQL> select time, region, department, sum(profit) as PROFIT from sales GROUP BY time, ROLLUP(region, department);

 TIME REGION DEPARTMENT
 PROFIT

 2000 east
 2000

 2000 west
 2000

 2000 west
 5000

 2000 west
 16000

 2000 central pensales
 12000

 2000 central pensales
 12000

 2000 central pensales
 12000

 2001 east pensales
 5500

 2001 east booksales
 11000

 2001 west booksales
 11000

 2001 uest booksales
 11000

 2001 central pensales
 6000

 2001 central
 2500

 2002 cest booksales
 5500

 2003 cest booksales
 5500

 2004 west booksales
 5500

 2002 cest booksales
 5500

 2002 cest booksales
 5500

 2002 cest booksales
 2000

 2002 central booksales
 2000

 2002 central booksales
 2000

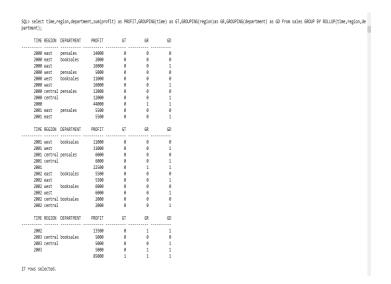
 2003 central booksales
 5000

 2003 central booksales
 5000

 2003 central book

Q. Using a grouping clause display time, region and department sum of profit and rollup by time, region and department.

Output:



Q. Display the time, region and department from sales table and group them and cube only for time, region and department.

Output:

SQL> select time,nvl(region,0) REGION,nvl(department,0)DEPT,sum(profit) as PROFIT from sales GROUP BY CUBE(time,region,department);

TIME	REGION	DEPT	PROFIT
	0	0 pensales	85000 42500
	-		42500
	east		27000
		-	19500
		booksales	7500
	west		33000
	west	pensales	5000
	west	booksales	28000
	central	0	25000
	central	pensales	18000
	REGION		PROFIT
		booksales	7000
		0 DOOK201E2	44000
			31000
		booksales	13000
	east		16000
2000	east	pensales	14000
2000	east	booksales	2000
2000	west	0	16000
		pensales	5000
		booksales	11000
2000	central	0	12000
	REGION		PROFIT
		pensales 0	12000 22500
		pensales	11500
	А		11000
		0	5500
		pensales	5500
		0	11000
2001	west	booksales	11000
2001	central	0	6000
2001	central	pensales	6000
200	12 0	0	13500
TTA	IE DECTA	N DEDT	DDOETT
111	ie keulu	N DEPT	PROFIT
	12 0	booksales	13500
200	2 east	0	5500
200	2 east	booksales	5500
	2 west	0	6000
		•	
200	12 west	booksales	6000
200	2 centr	al 0	2000
200	2 centr	al booksales	2000
200	- centi	at nonypates	2000

44 rows selected.

2003 0 0

2003 central 0

2003 central booksales

booksales

2003 0

5000

5000

5000

Q. Display time, region and department from sales table and group them by time and cube only for region and department.

Output:

SQL> select time, region, department, sum(profit) as PROFIT from sales GROUP BY time, CUBE(region, department);

-			
	TIME REGIO	ON DEPARTMENT	PROFIT
	2000		44000
		pensales	
	2000	booksales	
	2000 east		16000
	2000 east	pensales	14000
		booksales	
	2000 west		16000
	2000 west	pensales	5000
	2000 west	booksales	11000
	2000 centr	ral	12000
	2000 centr	ral pensales	12000
	TIME REGIO	ON DEPARTMENT	PROFIT
	2001		22500
		pensales	
	2001	booksales	
	2001 east		5500
		pensales	
	2001 west		11000
		booksales	
	2001 centr		6000
		ral pensales	
	2002		13500
	2002	booksales	13500
	TTUE DE		2225
	IIME KEGIO	ON DEPARTMENT	PROFIT
	2002		FF00
	2002 east		5500
		booksales	
	2002 west		6000
		booksales	
	2002 centr		2000
		ral booksales	2000
	2003		5000
		booksales	
	2003 centr	ral ral booksales	5000 5000

Q. Display time, region and department from sales table and apply grouping sets on them.

Output:

9 rows selected.

SQL> SELECT REGION, DEPARTMENT, TIME, SUM(PROFIT) AS PROFIT FROM SALES GROUP BY grouping sets(region, department, time);

REGION	DEPARTMENT	TIME	PROFIT
west			33000
central			25000
east			27000
	booksales		42500
	pensales		42500
		2001	22500
		2000	44000
		2003	5000
		2002	13500

9 rows selected.