```
group by cube(Dept,Desig)
order by Dept,Desig;
```

Sample Output:

```
 DEPT  DESIG      TOTAL
---------- -------------------- ----------
10 Asst Manager   2
10 Manager  3
10  5
20 Asst Manager   3
20 Manager  2
20  5
30 Asst Manager   1
30 Manager  2
30  3
   Asst Manager   6
   Manager  7

      DEPT DESIG      TOTAL
---------- -------------------- ----------
13
```

When to Use CUBE?

(a) Use CUBE in any situation requiring cross-tabular reports.

(b) CUBE is especially valuable in queries that use columns from multiple dimensions rather than columns representing different levels of a single dimension. For instance, a commonly requested cross-tabulation might need subtotals for all the combinations of month/state/product.

4.4 Date Operations

DATE is an Oracle datatype, just as VARCHAR2 and NUMBER are, and it has its own unique properties. The DATE datatype is stored in a special internal Oracle format that includes not just the month, day, and year, but also the hour, minute, and second. TIMESTAMP datatype stores fractional seconds.

```
 select SysDate from DUAL;

 SYSDATE
---------
```

```
28-FEB-08


select Systimestamp from dual;
SYSTIMESTAMP
---------------------------------------------------------------------------
27-JAN-15 11:29:35.765787 +06:00
```

Fundamental Operations: (Chapter 10 for details)

1. **TO_DATE and TO_CHAR Formatting**

   TO_CHAR

```
  select max(hire_date)MX,
   TO_CHAR(max(hire_date),'DD-MM-YYYY') DateConv
    from employees;

    Sample Output:

    MX    DATECONV
------------------ ----------
21-APR-08    21-04-2008
```

   Month Part:

```
 Month Formats:
Format  Result
Month    August
Mon      Aug
```

   Day Part:

   The day of the month is produced by the DD in the format. A suffix of th on DD tells Oracle to use ordinal suffixes, such as TH, RD, and ND with the number. In this instance, the suffixes are also case sensitive, but their case is set by the DD, not the th:

```
 Format Result
DDth or DDTH 11TH
Ddth or DdTH 11Th
Ddth or ddTH 11th
```

For part of one hour: `'HH:MI:SS'`

2. **Adding and subtracting Month** You can fast forward or go back in term of number of Months. Use `ADD_MONTHS`

   Example:

```
  select hire_date DT, ADD_MONTHS(hire_date,47)Date_After47Moth
from employees
where department_id=50;
```

3. **Difference Between Two Dates** Date1 - Date1 gives the result in number of days.

   Example:

```
  select max(hire_date)MX
    from employees;


select min(hire_date)MN
    from employees;

select (max(hire_date)-min(hire_date))/365 Year
    from employees;
```

4. **Date inside where clause**

```
  --bETWEEN CLAUSE--

select hire_date
from employees
where hire_date between '01-JAN-06' AND '01-FEB-09';


--MORE SECURE WAY TO IT--

select hire_date
from employees
where hire_date between TO_DATE('01-JAN-06','DD-MON-YY') AND TO_DATE('01-FEB-09'
```

EXTRACT Function: You can use the EXTRACT function in place of the TO_CHAR function when you are selecting portions of date valuessuch as just the month or day from a date. The EXTRACT functions syntax is:

```
 EXTRACT
( { { YEAR
| MONTH
| DAY
| HOUR
| MINUTE
| SECOND
}
| { TIMEZONE_HOUR
| TIMEZONE_MINUTE
}
| { TIMEZONE_REGION
| TIMEZONE_ABBR
}
}
FROM { datetime_value_expression | interval_value_expression }
)
```

Example:

```
 select First_Name, EXTRACT( Month from hire_date) M
from employees;
```