**DBMS SQL Assignment**

1.Retrieve the details (Name, Salary and dept code) of the employees who are working in department 20, 30 and 40.

**select dept\_name, salary, dept\_code from Department\_Masters where dept\_code in (10,20,40);**

**DEPT\_NAME SALARY DEPT\_CODE**

**-------------------------------------------------- ---------- ---------**

**Computer Science 20000 10**

**Electricals 25000 20**

**Mechanics 30000 40**

1. List the details of the employees with user defined Column headers.

**desc Department\_Masters;**

**Name Null? Type**

**----------------------------------------------------- -------- ------------------------------------**

**DEPT\_CODE NOT NULL NUMBER(2)**

**DEPT\_NAME VARCHAR2(50)**

**SALARY NUMBER**

3. Display the code, subjects and total marks for every student. Total Marks will be calculated as Subject1+Subject2+Subject3. (Refer Student\_Marks table)

**select Student\_Code,Maths,Physics,History,(Maths+Physics+History) as Total\_Marks from Student\_Marks;**

**STUDENT\_CODE MATHS PHYSICS HISTORY TOTAL\_MARKS**

**------------ ---------- ---------- ---------- -----------**

**1001 55 45 78 178**

**1002 66 74 88 228**

**1003 87 54 65 206**

**1004 65 64 90 219**

**1005 78 88 65 231**

**1006 65 86 54 205**

**1007 67 79 49 195**

**1008 72 55 55 182**

**1009 71 59 58 188**

**1010 68 44 92 204**

**1011 89 96 78 263**

**STUDENT\_CODE MATHS PHYSICS HISTORY TOTAL\_MARKS**

**------------ ---------- ---------- ---------- -----------**

**1012 78 56 55 189**

**1013 75 58 65 198**

**1014 73 74 65 212**

**1015 66 45 74 185**

**1016 68 78 74 220**

**1017 69 44 52 165**

**1018 65 78 56 199**

**1019 78 58 74 210**

**1020 45 55 65 165**

**1021 78 79 78 235**

**1001 68 44 92 204**

**STUDENT\_CODE MATHS PHYSICS HISTORY TOTAL\_MARKS**

**------------ ---------- ---------- ---------- -----------**

**1002 89 96 78 263**

**1003 78 56 55 189**

**1004 75 58 65 198**

**1005 73 74 65 212**

**1006 66 45 74 185**

**1007 68 78 74 220**

**1008 69 44 52 165**

**1009 65 78 56 199**

**1010 78 58 74 210**

**1011 45 55 65 165**

**1012 78 79 78 235**

**STUDENT\_CODE MATHS PHYSICS HISTORY TOTAL\_MARKS**

**------------ ---------- ---------- ---------- -----------**

**1013 66 74 88 228**

**1014 65 64 90 219**

**1015 78 88 65 231**

**1016 65 86 54 205**

**1017 67 79 49 195**

**1018 72 55 55 182**

**1019 71 59 58 188**

**1020 55 45 78 178**

1. **87 54 65 206**

5. List the code, name, and department number of the employees who have experience of more than 18 years.

**select Staff\_Code,Staff\_Name,Dept\_Code from staff\_Masters where(sysdate-Hiredate)> (18\*365);**

**STAFF\_CODE STAFF\_NAME DEPT\_CODE**

**---------- -------------------------------------------------- ---------**

**100004 Anil 20**

**100005 John 10**

**100006 Allen 30**

6. List the name and Designations of the staff who have joined before Jan 2003.

**select staff\_name , design\_name from staff\_masters , designation\_masters where hiredate > to\_date ('01-Jan-2003','DD\_MM\_YYYY');**

**STAFF\_NAME DESIGN\_NAME**

**--------------- --------------------------------------------------**

**Arvind HOD**

**Arvind Professor**

**Arvind Reader**

**Arvind Sr.Lecturer**

**Arvind Lecturer**

**Arvind Director**

**Raviraj HOD**

**Raviraj Professor**

**Raviraj Reader**

**Raviraj Sr.Lecturer**

**Raviraj Lecturer**

**STAFF\_NAME DESIGN\_NAME**

**--------------- --------------------------------------------------**

**Raviraj Director**

**Rahul HOD**

**Rahul Professor**

**Rahul Reader**

**Rahul Sr.Lecturer**

**Rahul Lecturer**

**Rahul Director**

7. List the name, designation, and income for 10 years of the employees who are working in departments 10 and 30.

**select sm.staff\_name, dm.design\_name, staff\_sal \*10 from staff\_masters sm, designation\_masters dm where sm.design\_code = dm.design\_code and dept\_code in (10,30);**

**STAFF\_NAME DESIGN\_NAME STAFF\_SAL\*10**

**--------------- -------------------------------------------------- ------------**

**Mohan Professor 240000**

**Arvind Professor 170000**

**Ram Reader 320000**

**Allen Reader 420000**

**John Director 320000**

9. Display name concatenated with dept code separated by comma and space. Name the column as ‘Student Info’.

**select Student\_Name ||','|| Dept\_Code Student\_Info from Student\_Masters;**

**STUDENT\_INFO**

**-------------------------------------------------------------------------------------------**

**Amit,10**

**Ravi,10**

**Ajay,20**

**Raj,30**

**Arvind,40**

**Rahul,50**

**Mehul,20**

**Dev,10**

**Vijay,30**

**Rajat,40**

**Sunder,50**

**STUDENT\_INFO**

**-------------------------------------------------------------------------------------------**

**Rajesh,30**

**Anil,20**

**Sunil,10**

**Kapil,40**

**Ashok,40**

**Ramesh,30**

**Amit Raj,50**

**Ravi Raj,50**

**Amrit,10**

**Sumit,20**

10. List the Name and Salary of the staff who are earning between 12000 and 25000. Sort them based on their salaries and name.

**SQL> select staff\_name, staff\_sal from staff\_masters where staff\_sal between 12000 and 25000 order by staff\_sal, staff\_name;**

**STAFF\_NAME STAFF\_SAL**

**--------------- ----------**

**Arvind 17000**

**Raviraj 18000**

**Anil 20000**

**Rahul 22000**

**Mohan 24000**

11. Display employees who do not have manager.

**SQL> select staff\_name from staff\_masters where mgr\_code is null;**

**STAFF\_NAME**

**---------------**

**Arvind**

**Anil**

**Smith**

**Ram**

12. Write a query which will display name, department code and date of birth of all students who were born between January 1, 1981 and March 31, 1983. Sort it based on date of birth (ascending).

**SQL> select student\_name, dept\_code, student\_dob from student\_masters where student\_dob between to\_date ('01/01/1981','DD/MM/YYYY') and to\_date ('31/03/1983','DD/MM/YYYY') order by student\_dob;**

**STUDENT\_NAME DEPT\_CODE STUDENT\_D**

**-------------------------------------------------- --------- ---------**

**Rahul 50 16-JAN-81**

**Dev 10 11-MAR-81**

**Kapil 40 18-MAR-81**

**Ravi Raj 50 29-MAY-81**

**Ravi 10 01-NOV-81**

**Ajay 20 13-JAN-82**

**Mehul 20 17-JAN-82**

**Arvind 40 15-JAN-83**

13. Get the Department number, and sum of Salary of all non managers where the sum is greater than 20000.

**SQL> select dept\_code, sum(staff\_sal) from staff\_masters where staff\_code not in (select distinct mgr\_code from staff\_masters ) and staff\_sal > 20000 group by dept\_code;**

**DEPT\_CODE SUM(STAFF\_SAL)**

**--------- --------------**

**30 32000**

**20 22000**

15. Display the name and department code of students. If student does not belong to any department, display “No Department”. Label the column as “Department”. (Hint: Use NVL function)

**SQL> select student\_name ,NVL (to\_char (dept\_code),'No department') department from student\_masters;**

**STUDENT\_NAME DEPARTMENT**

**-------------------------------------------------- ----------------------------------------**

**Amit 10**

**Ravi 10**

**Ajay 20**

**Raj 30**

**Arvind 40**

**Rahul 50**

**Mehul 20**

**Dev 10**

**Vijay 30**

**Rajat 40**

**Sunder 50**

**STUDENT\_NAME DEPARTMENT**

**-------------------------------------------------- ----------------------------------------**

**Rajesh 30**

**Anil 20**

**Sunil 10**

**Kapil 40**

**Ashok 40**

**Ramesh 30**

**Amit Raj 50**

**Ravi Raj 50**

**Amrit 10**

**Sumit 20**

16. Display the name and salary of the staff. Salary should be represented as X. Each X represents a 1000 in salary. Sample Output

JOHN 10000 XXXXXXXXXX ALLEN 12000 XXXXXXXXXXXX

**SQL> select staff\_name ,lpad('x', floor(staff\_sal/10000),'x') as salary from staff\_masters;**

**STAFF\_NAME SALARY**

**---------- ---------------**

**Arvind x**

**Mohan xx**

**Anil xx**

**John xxx**

**Allen xxxx**

**Smith xxxxxx**

**Raviraj x**

**Rahul xx**

**Ram xxx**

**^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^**

1. List the details of the employees, whose names start with ‘A’ and end with ‘S’.

**SQL> select staff\_code , staff\_name , dept\_code , hiredate mgr\_code , staff\_sal from staff\_masters where staff\_name like 'a%' and staff\_name like '%s';**

**STAFF\_CODE STAFF\_NAME DEPT\_CODE MGR\_CODE STAFF\_SAL**

**---------- --------------- --------- --------- ----------**

**100005 adams 10 21-JAN-01 32000**

**100007 amos 20 12-MAR-02 62000**

**100010 adonis 30 17-JAN-02 32000**

4. List the name and job of the employees whose names should contain N as the second or third character, and ending with either ‘N’ or ‘S’.

**SQL> select staff\_name from staff\_masters where (staff\_name like '\_n%' or staff\_name like '\_\_n%' ) and (staff\_name like '%n' or staff\_name like '%s');**

**STAFF\_NAME**

**---------------**

**sanikas**

**mn staff\_name format a10;**

**SQL> column salary format a15 ;**

**SQL> select staff\_name ,lpad('x', floor(staff\_sal/10000),'x') as salary from staff\_masters;**

5. Create a query which will display Staff Name, Salary of each staff. Format the salary to be 15 character long and left padded with ‘$’.

**SQL> select staff\_name, lpad(staff\_sal,15,'$') from staff\_masters;**

**STAFF\_NAME LPAD(STAFF\_SAL,15,'$')**

**--------------- ------------------------------------------------------------**

**Arvind $$$$$$$$$$17000**

**Mohan $$$$$$$$$$24000**

**Anil $$$$$$$$$$20000**

**John $$$$$$$$$$32000**

**Allen $$$$$$$$$$42000**

**Smith $$$$$$$$$$62000**

**Raviraj $$$$$$$$$$18000**

**Sanika $$$$$$$$$$22000**

**Satya $$$$$$$$$$32000**

8. Write a query that displays Staff Name, Salary, and Grade of all staff. Grade depends on the following table.

**Salary Grade**

Salary >=50000 A

Salary >= 25000 < 50000 B

Salary>=10000 < 25000 C

OTHERS D

**SQL> update staff\_masters set Grade='A' where staff\_sal >= 50000;**

**1 row updated.**

**SQL> update staff\_masters set Grade='B' where staff\_sal >= 25000 and staff\_sal <= 50000;**

**3 rows updated.**

**SQL> update staff\_masters set Grade='C' where staff\_sal >= 10000 and staff\_sal <= 25000;**

**5 rows updated.**

**SQL> update staff\_masters set Grade='D' where staff\_sal <= 10000;**

**0 rows updated.**

**SQL> select staff\_name, staff\_sal ,grade from staff\_masters;**

**STAFF\_NAME STAFF\_SAL GRADE**

**--------------- ---------- ----------**

**Arvind 17000 C**

**Mohan 24000 C**

**Anil 20000 C**

**John 32000 B**

**Allen 42000 B**

**Smith 62000 A**

**Raviraj 18000 C**

**Sanika 22000 C**

**Satya 32000 B**

16. Display the Highest, Lowest, Total & Average salary of all staff. Label the columns Maximum, Minimum, Total and Average respectively. Round the result to nearest whole number.

**SQL> select min(staff\_sal) as minimum\_sal, max(staff\_sal) as maximum\_sal, sum(staff\_sal) as total\_sal,avg(staff\_sal) as average\_sal from staff\_masters;**

**MINIMUM\_SAL MAXIMUM\_SAL TOTAL\_SAL AVERAGE\_SAL**

**----------- ----------- ---------- -----------**

**17000 62000 269000 29888.8889**

17. Edit the above query and display the same for each Department Name.

**SQL> select dept\_code, min(staff\_sal) as minimum\_sal, max(staff\_sal) as maximum\_sal, sum(staff\_sal) as total\_sal,avg(staff\_sal) as average\_sal from staff\_masters where staff\_sal is not null group by dept\_code;**

**DEPT\_CODE MINIMUM\_SAL MAXIMUM\_SAL TOTAL\_SAL AVERAGE\_SAL**

**--------- ----------- ----------- ---------- -----------**

**30 17000 42000 91000 30333.3333**

**20 20000 62000 104000 34666.6667**

**40 18000 18000 18000 18000**

**10 24000 32000 56000 28000**

18. Write a query to display number of people in each Department. Output should display Department Code, Department Name and Number of People.

**SQL> select dept\_code, count(\*) as no\_of\_people from staff\_masters where dept\_code is not null group by dept\_code;**

**DEPT\_CODE NO\_OF\_PEOPLE**

**--------- ------------**

**30 3**

**20 3**

**40 1**

**10 2**

19. Determine the number of managers without listing them. Label the column as ‘Total Number of Managers’.

**SQL> select count(designation) as total\_no\_of\_managers from staff\_masters where designation='Manager' group by designation;**

**TOTAL\_NO\_OF\_MANAGERS**

**--------------------**

**3**

20. Display Manager Code, Manager Name and salary of lowest paid staff in that manager’s team. Exclude any group where minimum salary is less than 10000. Order the result on descending order of salary

**SQL> select staff\_code, staff\_name, staff\_sal from staff\_masters where staff\_sal >=10000 and designation ='Manager' order by staff\_sal desc;**

**STAFF\_CODE STAFF\_NAM STAFF\_SAL**

**---------- --------- ----------**

**100006 Allen 42000**

**100010 Satya 32000**

**100001 Arvind 17000**

**Lab 3. JOINS AND SUBQUERIES**

1. Write a query which displays Staff Name, Department Code, Department Name, and Salary for all staff who earns more than 20000.

**SQL> select s.staff\_name, d.dept\_code, d.dept\_name from staff\_masters s join department\_masters d on s.dept\_code = d.dept\_code where staff\_sal > 20000;**

1. Write a query to display Staff Name, Department Code, and Department Name for all staff who do not work in Department code 10 and have ‘A’ in their name.

**SQL> select s.staff\_name, d.dept\_code, d.dept\_name from staff\_masters s join department\_masters d on s.dept\_code = d.dept\_code where d.dept\_code <> 10 and staff\_name like '%a%';**

1. Display Staff Code, Staff Name, Department Name, and his manager’s number and name. Label the columns Staff#, Staff, Mgr#, Manager.

**SQL> select s.staff\_name as Staff, s.staff\_code as Staff#, d.dept\_name as manager, s.mgr\_code as mgr# from staff\_masters s join department\_masters d on s.dept\_code = d.dept\_code;**

1. Create a query that will display Student Code, Student Name, Department Name, Subjec1, Subject2, and Subject3 for all students who are getting 60 and above in each subject from department 10 and 20.

**SQL> select s.student\_code, s.student\_name, d.dept\_name, m.maths, m.physics, m.history from student\_masters s, department\_masters d, student\_marks m where s.dept\_code=d.dept\_code and s.student\_code=m.student\_code and m.maths>=60 and m.physics>=60 and m.history>=60 and d.dept\_code in (10,20);**

6.Create a query that will display Staff Code, Staff Name, Department Name, Designation, Book Code, Book Name, and Issue Date. For only those staff who have taken any book in last 30 days

**SQL> select s.staff\_code, s.staff\_name,b.dept\_name, a.design\_name, d.book\_code, d.book\_name, t.book\_issue\_date from designation\_masters a, department\_masters b, staff\_masters s, book\_masters d, book\_transactions t where a.design\_code=s.design\_code and b.dept\_code=s.dept\_code and d.book\_code=t.book\_code and book\_issue\_date>=sysdate-30;**

7. Generate a report which contains the following information. Staff Code, Staff Name, Designation ,Department Name, Department Head For all staff excluding HOD (List should not contain the details of Department head).

**SQL> select a.staff\_code, a.staff\_name, b.design\_name, c.dept\_name from staff\_masters a, designation\_masters b, department\_masters c where a.dept\_code=c.dept\_code and a.design\_code=b.design\_code and b.design\_name not in 'hod';**

1. List Staff Code, Staff Name, and Salary for those who are getting less than the average salary of organization.

**SQL> select staff\_code, staff\_name, staff\_sal from staff\_masters where staff\_sal < (select avg(staff\_sal) from staff\_masters);**

1. List the Staff Code, Staff Name who are not Manager.

**SQL> select staff\_code, staff\_name from staff\_masters where mgr\_code not in (select staff\_code from staff\_masters);**

1. Display Author Name, Book Name for those authors who wrote more than one book.

**SQL> select distinct Book\_pub\_author, Book\_name from book\_masters where book\_pub\_author in (select book\_pub\_author from book\_masters group by book\_pub\_author having count(1)>1) order by 1,2;**

1. Display Staff Code, Staff Name, and Department Name for those who have taken more than one book.

**SQL> select distinct s.staff\_code, s.staff\_name, d.dept\_name, m.book\_code from department\_masters d, staff\_masters s, book\_transactions b, book\_masters m where s.dept\_code=d.dept\_code and b.book\_code=m.book\_code and s.staff\_code=b.staff\_code and m.book\_code in (select m.book\_code from department\_masters d, staff\_masters s, book\_transactions b, book\_masters m where s.dept\_code=d.dept\_code and b.book\_code=m.book\_code and s.staff\_code=b.staff\_code group by b.book\_code having count(b.book\_code)>1);**

14. Display top ten students for a specified department. Details are: Student Code, Student Name, Department Name, Subject1, Subject2, Subject3, Total.

**SQL> select s.student\_code, s.student\_name, d.dept\_name, (m.maths+m.physics+m.history) as total from student\_masters s, department\_masters d, student\_marks m where s.dept\_code=d.dept\_code and m.student\_code=s.student\_code and rownum<11 order by 1 desc;**

15. Display the Staff Name, Department Name, and Salary for those staff who are getting less than average salary in their own department

**SQL> (select s.staff\_name, d.dept\_name, s.staff\_sal from staff\_masters s, department\_masters d, (select avg(s.staff\_sal) avg\_sal, d.dept\_code from staff\_masters s, department\_masters d where s.dept\_code=d.dept\_code group by d.dept\_code) tab1 where s.dept\_code=d.dept\_code and d.dept\_code=tab1.dept\_code and s.staff\_sal<tab1.avg\_sal);**

1. Create a query that will display the Staff Name, Department Name, and all the staff that work in the same department as a given staff. Give the column as appropriate label.

**SQL> select s.staff\_name, d.dept\_name from staff\_masters s, department\_masters d where s.dept\_code=d.dept\_code group by s.staff\_name,d.dept\_name;**

1. List the Student Code, Student Name for that student who got highest marks in all three subjects in Computer Science department for current year.

18. Display the Student Code, Student Name, and Department Name for that department in which there are maximum number of student are studying.

19. Display Staff Code, Staff Name, Department Name, and Designation for those who have joined most recently.

20. Display the Manager Name and the total strength of his/her team.

**Lab 4. Database Objects**

1. Create the Customer table with the following columns.

Customerid Number(5)

CustomerName Number(10)

Address1 Varchar2(30)

Address2 Varchar2(30)

**SQL> create table customer\_table(**

**2 customer\_id number(5),**

**3 customer\_name number(10),**

**4 address1 varchar2(30),**

**5 address2 varchar2(30));**

**Table created.**

1. Modify the Customer table CustomerName column of datatype with Varchar2(30). CustomerName should not accept Nulls.

**SQL> alter table customer\_table MODIFY customer\_name varchar2(30);**

**SQL> alter table customer\_table modify customer\_name varchar2(30) not null;**

1. Add the following Columns to the Customer table.

Gender Varchar2(1)

Age Number(3)

PhoneNo Number(10)

**SQL> alter table customer\_table ADD gender varchar2(1);**

**SQL> alter table customer\_table ADD age number(3);**

**SQL> alter table customer\_table ADD PhoneNo number(10);**

1. Insert rows with the following data in to the Customer table.

Insert into customer values: (1000, ‘Allen’, ‘#115 Chicago’, ‘#115 Chicago’, ‘M’, ‘25, 7878776’)

In similar manner, add the below records to the Customer table:

1000, Allen, #115 Chicago, #115 Chicago, M, 25, 7878776 1001,

George, #116 France, #116 France, M, 25, 434524 1002,

Becker, #114 New York, #114 New York, M, 45, 431525

**SQL> insert into customer\_table (customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (1000, 'Allen', '#115 Chicago', '#115 Chicago', 'M', 25, 7878776);**

**SQL> insert into customer\_table (customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (1001, 'George', '#116 France', '#116 France', 'M', 25, 434524);**

**SQL> insert into customer\_table (customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (1002, 'Becker', '#114 New York', '#114 New York', 'M', 45, 431525 );**

5. Add the Primary key constraint for Customerld with the name Custld\_Prim.

**SQL> alter table customer\_table add constraint Custid\_Prim primary key(customer\_id);**

1. Insert the row given below in the Customer table and see the message generated by the Oracle server.

1002, John, #114 Chicago, #114 Chicago, M, 45, 439525

**SQL> insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1002, 'John',45, 'M', ' #114 Chicago', '#114 Chicago', 439525 );**

**insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1002, 'John',45, 'M', ' #114 Chicago', '#114 Chicago', 439525 )**

**\***

**ERROR at line 1:**

**ORA-00001: unique constraint (SYSTEM.CUSTID\_PRIM) violated**

1. Disable the constraint on CustomerId, and insert the following data:

1002, Becker, #114 New York, #114 New york , M, 45, 431525

1003, Nanapatekar, #115 India, #115 India , M, 45, 431525

**SQL> alter table customer\_table disable constraint custid\_prim;**

**SQL> insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1002, 'Becker',45, 'M', ' #114 New York', '#114 New York',431525);**

**SQL> insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1003, 'Nanapatekar',45, 'M', ' #115 India', '#115 India',431525);**

1. Enable the constraint on CustomerId of the Customer table, and see the message generated by the Oracle server.

**SQL> alter table customer\_table add constraint Custid\_Prim primary key(customer\_id);**

**alter table customer\_table add constraint Custid\_Prim primary key(customer\_id)**

**\***

**ERROR at line 1:**

**ORA-02260: table can have only one primary key**

1. Drop the constraint Custld\_Prim on CustomerId and insert the following Data. Alter Customer table, drop constraint Custid\_Prim.

1002, Becker, #114 New York, #114 New york , M, 45, 431525, 15000.50

1003, Nanapatekar, #115 India, #115 India , M, 45, 431525, 20000.50

**SQL> alter table customer\_table drop constraint custid\_prim;**

**SQL> insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1002, 'Becker',45, 'M', ' #114 New York', '#114 New York',431525, 15000.50);**

**insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1002, 'Becker',45, 'M', ' #114 New York', '#114 New York',431525, 15000.50)**

**\***

**ERROR at line 1:**

**ORA-00913: too many values**

**SQL> insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1003, 'Nanapatekar',45, 'M', ' #115 India', '#115 India',431525, 20000.50);**

**insert into customer\_table (customer\_id, customer\_name, age, gender, address1, address2, phoneno) values(1003, 'Nanapatekar',45, 'M', ' #115 India', '#115 India',431525, 20000.50)**

**\***

**ERROR at line 1:**

**ORA-00913: too many values**

10. Delete all the existing rows from Customer table, and let the structure remain itself using TRUNCATE statement.

**SQL> truncate table customer\_table;**

**Table truncated.**

1. In the Customer table, add a column E\_mail.

**SQL> alter table customer\_table add E\_mail varchar2(10);**

**Table altered**.

1. Drop the E\_mail column from Customer table.

**SQL> alter table customer\_table drop column e\_mail;**

**Table altered.**

1. Add a new column EmailId to Customer table.

**SQL> alter table customer\_table add emailid varchar2(20);**

**Table altered.**

1. Mark EmailId column as unused before dropping it.

**SQL> alter table customer\_table set unused column emailid;**

**Table altered.**

1. Drop the unused EmailId column from the Customer table.

**SQL> alter table customer\_table drop unused column;**

**Table altered.**

20. Create the Suppliers table based on the structure of the Customer table. Include only the CustomerId, CustomerName, Address1, Address2, and phoneno columns. Name the columns in the new table as SuppID, SName, Addr1, Addr2, and Contactno respectively

**SQL> create table Suppliers(**

**2 suppid number,**

**3 sname varchar2(10),**

**4 addr1 varchar2(10),**

**5 addr2 varchar2(10),**

**6 contactno number);**

21. Drop the above table and recreate the following table with the name CustomerMaster. CustomerId Number(5) Primary key(Name of constraint is CustId\_PK)

CustomerName Varchar2(30) Not Null

Addressl Varchar2(30) Not Null

Address2 Varchar2(30)

Gender Varchar2(l)

Age Number(3)

PhoneNo Number(10)

**SQL> create table customermasters(**

**2 customerid number(5),**

**3 customername varchar2(30) not null,**

**4 address1 varchar2(30) not null,**

**5 address2 varchar2(30),**

**6 gender varchar2(1),**

**7 age number(3),**

**8 phoneno number(10),**

**9 constraint custid\_pk primary key(customerid));**

**Table created.**

22. Create the AccountsMaster table with the following Columns. Use sequence to generate Account number

Customerld Number(5)

AccountNumber Number(10,2) Primary key(Name of constraint is Acc\_PK)

AccountType Char(3)

LedgerBalance Number(10,2) Not Null

**SQL> create table accountsmaster(**

**2 customerid number(5),**

**3 accountnumber number(10,2),**

**4 accounttype char(3),**

**5 ledgerbalance number(10,2) not null,**

**6 constraint acc\_pk primary key (customerid));**

**Table created.**

23. Relate AccountsMaster table and CustomerMaster table through Customerld column with the constraint name Cust\_acc.

SQL> alter table accountsmaster add constraint balance\_check check(ledgerbalance>5000);

**5.1: Data Manipulation Language**

**1. Create Employee table with same structure as EMP table.**

SQL> CREATE TABLE emp (

2 empno NUMBER(4) CONSTRAINT pk\_emp PRIMARY KEY,

3 ename VARCHAR2(10),

4 job VARCHAR2(9),

5 mgr NUMBER(4),

6 hiredate DATE,

7 sal NUMBER(7,2),

8 comm NUMBER(7,2),

9 deptno NUMBER(2)

10 );

Table created.

SQL> INSERT INTO emp VALUES (7369,'SMITH','CLERK',7902,to\_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);

1 row created.

SQL> INSERT INTO emp VALUES (7499,'ALLEN','SALESMAN',7698,to\_date('20-2-1981','dd-mm-yyyy'),1600,300,30);

1 row created.

SQL> INSERT INTO emp VALUES (7521,'WARD','SALESMAN',7698,to\_date('22-2-1981','dd-mm-yyyy'),1250,500,30);

1 row created.

SQL> INSERT INTO emp VALUES (7566,'JONES','MANAGER',7839,to\_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);

1 row created.

SQL> INSERT INTO emp VALUES (7654,'MARTIN','SALESMAN',7698,to\_date('28-9-1981','dd-mm-yyyy'),1250,1400,30);

1 row created.

SQL> INSERT INTO emp VALUES (7698,'BLAKE','MANAGER',7839,to\_date('1-5-1981','dd-mm-yyyy'),2850,NULL,30);

1 row created.

SQL> INSERT INTO emp VALUES (7782,'CLARK','MANAGER',7839,to\_date('9-6-1981','dd-mm-yyyy'),2450,NULL,10);

1 row created.

SQL> INSERT INTO emp VALUES (7788,'SCOTT','ANALYST',7566,to\_date('13-JUL-87','dd-mm-rr')-85,3000,NULL,20);

1 row created.

SQL> INSERT INTO emp VALUES (7839,'KING','PRESIDENT',NULL,to\_date('17-11-1981','dd-mm-yyyy'),5000,NULL,10);

1 row created.

SQL> INSERT INTO emp VALUES (7844,'TURNER','SALESMAN',7698,to\_date('8-9-1981','dd-mm-yyyy'),1500,0,30);

1 row created.

SQL> INSERT INTO emp VALUES (7876,'ADAMS','CLERK',7788,to\_date('13-JUL-87', 'dd-mm-rr')-51,1100,NULL,20);

1 row created.

SQL> INSERT INTO emp VALUES (7900,'JAMES','CLERK',7698,to\_date('3-12-1981','dd-mm-yyyy'),950,NULL,30);

1 row created.

SQL> INSERT INTO emp VALUES (7902,'FORD','ANALYST',7566,to\_date('3-12-1981','dd-mm-yyyy'),3000,NULL,20);

1 row created.

SQL> INSERT INTO emp VALUES (7934,'MILLER','CLERK',7782,to\_date('23-1-1982','dd-mm-yyyy'),1300,NULL,10);

1 row created.

SQL> create table Employee as select \*from emp where 1=3;

Table created.

**2 . Write a query to populate Employee table using EMP table’s empno, ename, sal, deptno columns.**

SQL> insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno);

Enter value for empno: 7369

Enter value for ename: 'smith'

Enter value for sal: 800

Enter value for deptno: 20

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7369, 'smith', 800, 20)

1 row created.

SQL> /

Enter value for empno: 7499

Enter value for ename: 'allen'

Enter value for sal: 1600

Enter value for deptno: 30

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7499, 'allen', 1600, 30)

1 row created.

SQL> /

Enter value for empno: 7521

Enter value for ename: 'ward'

Enter value for sal: 1250

Enter value for deptno: 30

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7521, 'ward', 1250, 30)

1 row created.

SQL> /

Enter value for empno: 7566

Enter value for ename: 'jones'

Enter value for sal: 2975

Enter value for deptno: 20

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7566, 'jones', 2975, 20)

1 row created.

SQL> /

Enter value for empno: 7654

Enter value for ename: 'martin'

Enter value for sal: 1250

Enter value for deptno: 30

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7654, 'martin', 1250, 30)

1 row created.

SQL> /

Enter value for empno: 7698

Enter value for ename: 'blake'

Enter value for sal: 2850

Enter value for deptno: 30

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7698, 'blake', 2850, 30)

1 row created.

SQL> /

Enter value for empno: 7782

Enter value for ename: 'clark'

Enter value for sal: 2550

Enter value for deptno: 10

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7782, 'clark', 2550, 10)

1 row created.

SQL> /

Enter value for empno: 7788

Enter value for ename: 'scott'

Enter value for sal: 3000

Enter value for deptno: 20

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7788, 'scott', 3000, 20)

1 row created.

SQL> /

Enter value for empno: 7839

Enter value for ename: 'king'

Enter value for sal: 5000

Enter value for deptno: 10

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7839, 'king', 5000, 10)

1 row created.

SQL> /

Enter value for empno: 7844

Enter value for ename: 'turner'

Enter value for sal: 1500

Enter value for deptno: 30

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7844, 'turner', 1500, 30)

1 row created.

SQL> /

Enter value for empno: 7876

Enter value for ename: 'adams'

Enter value for sal: 1100

Enter value for deptno: 20

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7876, 'adams', 1100, 20)

1 row created.

SQL> /

Enter value for empno: 7900

Enter value for ename: 'james'

Enter value for sal: 950

Enter value for deptno: 30

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7900, 'james', 950, 30)

1 row created.

SQL> /

Enter value for empno: 7902

Enter value for ename: 'ford'

Enter value for sal: 3000

Enter value for deptno: 20

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7902, 'ford', 3000, 20)

1 row created.

SQL> /

Enter value for empno: 7934

Enter value for ename: 'miller'

Enter value for sal: 1300

Enter value for deptno: 10

old 1: insert into Employee(empno, ename, sal, deptno) values (&empno, &ename, &sal, &deptno)

new 1: insert into Employee(empno, ename, sal, deptno) values (7934, 'miller', 1300, 10)

1 row created.

SQL> select \*from Employee;

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

---------- ---------- --------- ---------- --------- ---------- ---------- ----------

7369 smith 800 20

7499 allen 1600 30

7521 ward 1250 30

7566 jones 2975 20

7654 martin 1250 30

7698 blake 2850 30

7782 clark 2550 10

7788 scott 3000 20

7839 king 5000 10

7844 turner 1500 30

7876 adams 1100 20

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

---------- ---------- --------- ---------- --------- ---------- ---------- ----------

7900 james 950 30

7902 ford 3000 20

7934 miller 1300 10

14 rows selected.

**3 Write a query to change the job and deptno of employee whose empno is 7698 to the job and deptno of employee having empno 7788.**

SQL> update Employee set job=(select job from employee where empno=7788) ,deptno=(select deptno from employee where empno=7788) where empno=7698;

1 row updated.

**4 Delete the details of department whose department name is ‘SALES’.**

SQL> delete from department\_masters where dept\_name='sales';

**5 Write a query to change the deptno of employee with empno 7788 to that of employee having empno 7698**

SQL> update employee set deptno=(select deptno from employee where empno=7788) where empno=7698;

**6 Insert the following rows to the Employee table through parameter substitution.**

** 1000,Allen, Clerk,1001,12-jan-01, 3000, 2,10**

** 1001,George, analyst, null, 08 Sep 92, 5000,0, 10**

** 1002, Becker, Manager, 1000, 4 Nov 92, 2800,4, 20**

** 1003, 'Bill', Clerk, 1002, 4 Nov 92,3000, 0, 20**

SQL> insert into employee values (&empno, &empname, &job, &mgr, &hiredate, &sal, &comm, &deptno);

Enter value for empno: 1000

Enter value for empname: 'allen'

Enter value for job: 'clerk'

Enter value for mgr: 1001

Enter value for hiredate: '12-jan-01'

Enter value for sal: 3000

Enter value for comm: 2

Enter value for deptno: 10

old 1: insert into employee values (&empno, &empname, &job, &mgr, &hiredate, &sal, &comm, &deptno)

new 1: insert into employee values (1000, 'allen', 'clerk', 1001, '12-jan-01', 3000, 2, 10)

1 row created.

SQL> /

Enter value for empno: 1001

Enter value for empname: 'george'

Enter value for job: 'analyst'

Enter value for mgr: null

Enter value for hiredate: '08-sep-92'

Enter value for sal: 5000

Enter value for comm: 0

Enter value for deptno: 10

old 1: insert into employee values (&empno, &empname, &job, &mgr, &hiredate, &sal, &comm, &deptno)

new 1: insert into employee values (1001, 'george', 'analyst', null, '08-sep-92', 5000, 0, 10)

1 row created.

SQL> /

Enter value for empno: 1002

Enter value for empname: 'becker'

Enter value for job: 'manager'

Enter value for mgr: 1000

Enter value for hiredate: '04-nov-92'

Enter value for sal: 2800

Enter value for comm: 4

Enter value for deptno: 20

old 1: insert into employee values (&empno, &empname, &job, &mgr, &hiredate, &sal, &comm, &deptno)

new 1: insert into employee values (1002, 'becker', 'manager', 1000, '04-nov-92', 2800, 4, 20)

1 row created.

SQL> /

Enter value for empno: 1003

Enter value for empname: 'bill'

Enter value for job: 'clerk'

Enter value for mgr: 1002

Enter value for hiredate: '04-nov-92'

Enter value for sal: 3000

Enter value for comm: 0

Enter value for deptno: 20

old 1: insert into employee values (&empno, &empname, &job, &mgr, &hiredate, &sal, &comm, &deptno)

new 1: insert into employee values (1003, 'bill', 'clerk', 1002, '04-nov-92', 3000, 0, 20)

1 row created.

**6.1: Transaction Control Language Statements**

**1. Insert rows with the following data into the Customer table.**

**6000, John, #115 Chicago, #115 Chicago, M, 25, 7878776, 10000**

** 6001, Jack, #116 France, #116 France, M, 25, 434524, 20000**

** 6002, James, #114 New York, #114 New York, M, 45, 431525, 15000.50**

**Use parameter substitution.**

SQL> insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (&customer\_id, &customer\_name, &address1, &address2, &gender, &age, &phoneno);

Enter value for customer\_id: 600

Enter value for customer\_name: 'john'

Enter value for address1: '#115 chicago'

Enter value for address2: '#115 chicago'

Enter value for gender: 'm'

Enter value for age: 25

Enter value for phoneno: 7878776

old 1: insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (&customer\_id, &customer\_name, &address1, &address2, &gender, &age, &phoneno)

new 1: insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (600, 'john', '#115 chicago', '#115 chicago', 'm', 25, 7878776)

1 row created.

SQL> /

Enter value for customer\_id: 6001

Enter value for customer\_name: 'jack'

Enter value for address1: '#116 france'

Enter value for address2: '#116 france'

Enter value for gender: 'm'

Enter value for age: 25

Enter value for phoneno: 434524

old 1: insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (&customer\_id, &customer\_name, &address1, &address2, &gender, &age, &phoneno)

new 1: insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (6001, 'jack', '#116 france', '#116 france', 'm', 25, 434524)

1 row created.

SQL> /

Enter value for customer\_id: 6002

Enter value for customer\_name: 'james'

Enter value for address1: '#114 new york'

Enter value for address2: '#114 new york'

Enter value for gender: 'm'

Enter value for age: 45

Enter value for phoneno: 431525

old 1: insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (&customer\_id, &customer\_name, &address1, &address2, &gender, &age, &phoneno)

new 1: insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values (6002, 'james', '#114 new york', '#114 new york', 'm', 45, 431525)

1 row created.

SQL> select \*from customer\_table;

1. **Create a Savepoint named ‘SP1’ after third record in the Customer table .**

SQL> savepoint sp1;

Savepoint created.

**3. Insert the below row in the Customer table. 6003, John, #114 Chicago, #114 Chicago, M, 45, 439525, 19000.60**

SQL> insert into customer\_table(customer\_id, customer\_name, address1, address2, gender, age, phoneno) values(6003, 'john', '#114 chicago', '#114 chicago', 'm', 45, 439525);

1 row created.

**4. Execute rollback statement in such a way that whatever manipulations done before Savepoint sp1 are permanently implemented, and the ones after Savepoint SP1 are not stored as a part of the Customer table.**

SQL> rollback to sp1;

Rollback complete.

SQL>

**PLSQL ASSIGNMENT**

2.1

Identify the problems(if any) in the below declarations:

DECLARE

V\_Sample1 NUMBER(2);

V\_Sample2 CONSTANT NUMBER(2) ;

V\_Sample3 NUMBER(2) NOT NULL ;

V\_Sample4 NUMBER(2) := 50;

V\_Sample5 NUMBER(2) DEFAULT 25;

**ERROR at line 6:**

**ORA-06550: line 6, column 31:**

**PLS-00103: Encountered the symbol "end-of-file" when expecting one of the**

**following:**

**begin function pragma procedure subtype type <an identifier>**

**<a double-quoted delimited-identifier> current cursor delete**

**exists prior**

2.2 The following PL/SQL block is incomplete. Modify the block to achieve requirements as stated in the comments in the block.

DECLARE --outer block var\_num1 NUMBER := 5;

BEGIN

DECLARE --inner block

var\_num1 NUMBER := 10;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Value for var\_num1:' ||var\_num1); --Can outer block variable (var\_num1) be printed here.If Yes,Print the same.

END; --Can inner block variable(var\_num1) be printed here.If Yes,Print the same.

END;

**SQL> declare**

**2 num1 number:=5;**

**3 begin**

**4 declare**

**5 num1 number:=10;**

**6 begin**

**7 dbms\_output.put\_line ('Value for num1 :' || num1);**

**8 end;**

**9 dbms\_output.put\_line ('Value for num1 :' || num1);**

**10 end;**

**11 /**

**Value for num1 :10**

**Value for num1 :5**

**PL/SQL procedure successfully completed.**

2.4. Write a PL/SQL block to increase the salary by 30 % or 5000 whichever minimum for a given Department\_Code.

**SQL> declare**

**2 cursor c1 is select student\_year,maths,physics,history from student\_marks where student\_code ='&student\_code';**

**3 total number;**

**4 student\_year varchar2(10);**

**5 maths number;**

**6 physics number;**

**7 history number;**

**8 per number;**

**9 grade varchar2(2);**

**10 begin**

**11 open c1 ;**

**12 fetch c1 into student\_year,maths,physics,history;**

**13 total:= (maths+physics+history);**

**14 per:=((total)/300)\*100;**

**15 if per >= 80 then**

**16 dbms\_output.put\_line(student\_year||' '||maths||' ' ||physics||' ' ||history||' '||total||' '||per||' '||'the grade is' || ' '|| 'A');**

**17 elsif per < 80 and per >= 70 then**

**18 dbms\_output.put\_line(student\_year||' '||maths||' ' ||physics||' ' ||history||' '||total||' '||per||' '||'the grade is' ||''|| 'B');**

**19 elsif per < 70 and per >= 60 then**

**20 dbms\_output.put\_line(student\_year||' '||maths||' ' ||physics||' ' ||history||' '||total||' '||per||' '||'the grade is' ||''|| 'C');**

**21 else**

**22 dbms\_output.put\_line(student\_year||' '||maths||' ' ||physics||' ' ||history||' '||total||' '||per||' '||'the grade is'||''|| 'D');**

**23 --exception**

**24 --when invalid\_number then**

**25 --dbms\_output.put\_line('enter proper student id');**

**26 end if;**

**27 --dbms\_output.put\_line(maths);**

**28 close c1;**

**29 end;**

**30 /**

**Enter value for student\_code: 1013**

**old 2: cursor c1 is select student\_year,maths,physics,history from student\_marks where student\_code ='&student\_code';**

**new 2: cursor c1 is select student\_year,maths,physics,history from student\_marks where student\_code ='1013';**

**2010 75 58 65 198 66 the grade isC**

**PL/SQL procedure successfully completed.**

**3.4**

**SQL>**

**SQL> BEGIN**

**2 DECLARE**

**3 fname emp.ename%TYPE;**

**4 BEGIN**

**5 SELECT ename INTO fname**

**6 FROM emp**

**7 WHERE 1=2;**

**8 DBMS\_OUTPUT.PUT\_LINE('This statement will print');**

**9 EXCEPTION**

**10 WHEN OTHERS THEN**

**11 DBMS\_OUTPUT.PUT\_LINE('Some inner block error');**

**12 END;**

**13 EXCEPTION**

**14 WHEN NO\_DATA\_FOUND THEN**

**15 DBMS\_OUTPUT.PUT\_LINE('No data found in fname');**

**16 WHEN OTHERS THEN**

**17 DBMS\_OUTPUT.PUT\_LINE('Some outer block error');**

**18 END;**

**19 /**

**Some inner block error**

**PL/SQL procedure successfully completed.**

**SQL>**

3.6: Write a PL/SQL program to check for the commission for an employee no 7369. If no commission exists, then display the error message. Use Exceptions.

**SQL> declare**

**2 name emp.ename%type;**

**3 salary emp.salary%type;**

**4 commision emp.comm%type;**

**5 my\_exp exception;**

**6 begin**

**7 select ename,salary,nvl(comm,0) into name,salary,commision from emp where empno=7369;**

**8 if commision = 0 then**

**9 raise my\_exp;**

**10 else**

**11 dbms\_output.put\_line(name || ',' ||salary || ',' || commision);**

**12 end if;**

**13**

**14 exception**

**15 when my\_exp then**

**16 dbms\_output.put\_line('commision is not found');**

**17 end;**

**18 /**

**commision is not found**

**PL/SQL procedure successfully completed.**

**SQL>**

3.7: Write a PL/SQL block to drop any user defined table.

**SQL> create or replace procedure s\_table(input in varchar2,input2 in varchar2)**

**2 is**

**3 begin**

**4 execute immediate 'drop'||input||' '||input2;**

**5 dbms\_output.put\_line('the row deleted');**

**6 end;**

**7 /**

**Procedure created.**

4.1 Write a PL/SQL block to find the maximum salary of the staff in the given department. Note: Department code should be passed as parameter to the cursor.

**SQL> create or replace procedure s\_table(input in number)**

**2 is**

**3 name emp.ename%type;**

**4 salary emp.salary%type;**

**5 cursor cur1 is select ename,max(salary) from emp where deptno=input;**

**6 begin**

**7 open cur1;**

**8 loop**

**9 fetch cur1 into name,salary;**

**10 exit when cur1%notfound;**

**11 dbms\_output.put\_line(name||' '||salary);**

**12 end loop;**

**13 close cur1;**

**14 end;**

**15 /**

**Procedure created.**

4.2. Write a function to compute age. The function should accept a date and return age in years.

**SQL> declare**

**2 name varchar2(20) :='&name';**

**3 dob date :='&dob';**

**4 age number;**

**5 begin**

**6 select floor(months\_between(sysdate,dob)/12) into age from dual;**

**7 dbms\_output.put\_line(name ||' '|| age);**

**8 end;**

**9 /**

**Enter value for name: sanika**

**old 2: name varchar2(20) :='&name';**

**new 2: name varchar2(20) :='sanika';**

**Enter value for dob: 04-sep-97**

**old 3: dob date :='&dob';**

**new 3: dob date :='04-sep-97';**

**sanika 22**

**PL/SQL procedure successfully completed.**