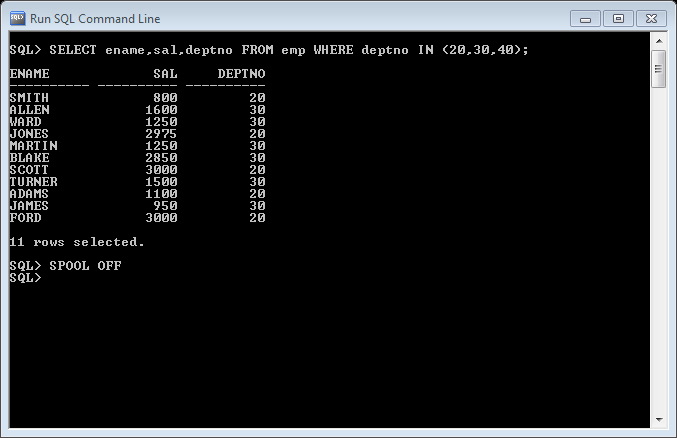
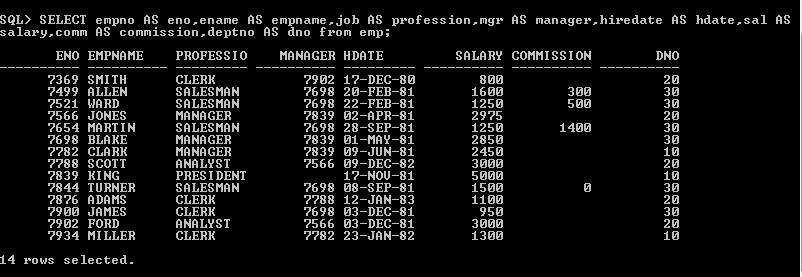
**1.1: Data Query Language**

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

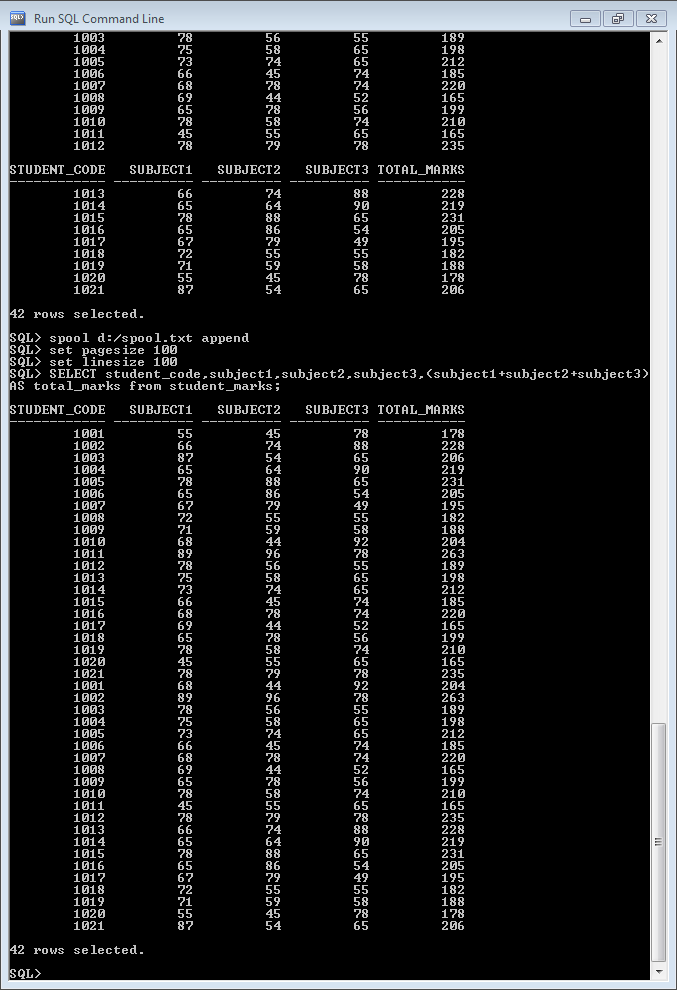
SELECT ename,sal,deptno FROM emp WHERE deptno IN (20,30,40);

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



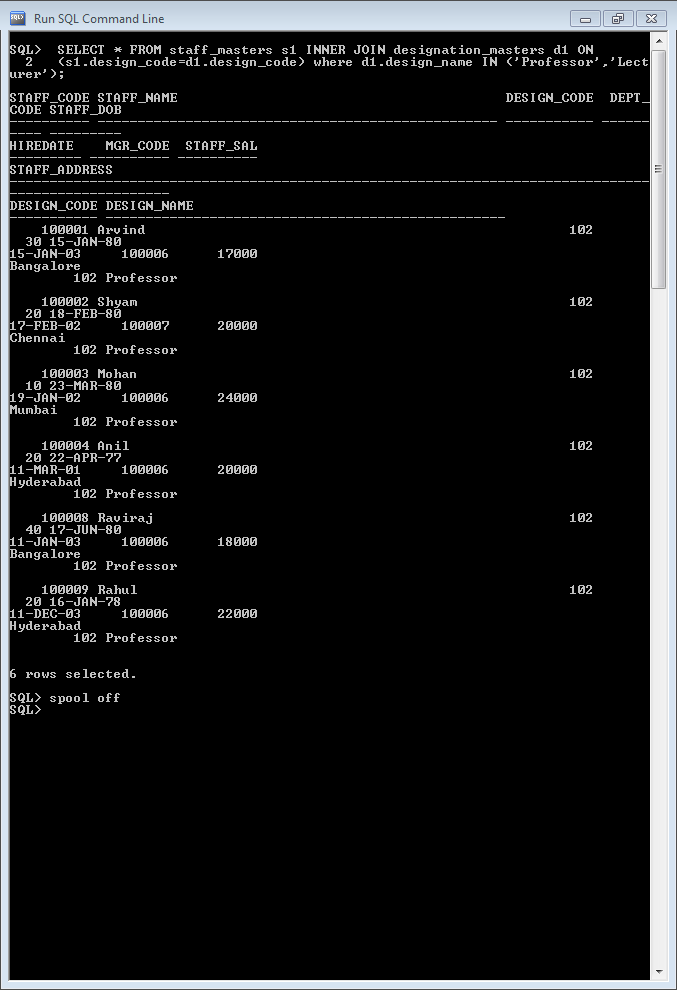
SELECT empno AS eno,ename AS empname,job AS profession,mgr AS manager,hiredate AS hdate,sal AS salary,comm AS commission,deptno AS dno from emp;

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

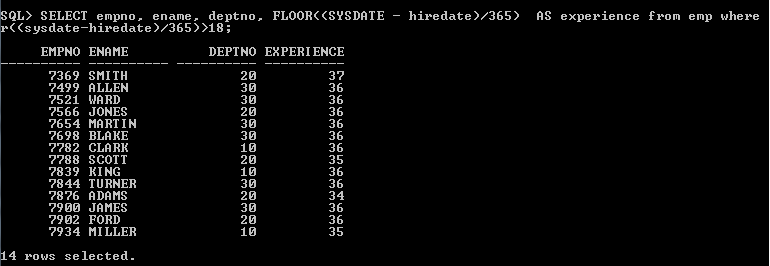


SELECT student\_code,subject1,subject2,subject3,(subject1+subject2+subject3)AS total\_marks from student\_marks;

4. List the details of the staff whose designations are either PROFESSOR or LECTURER.

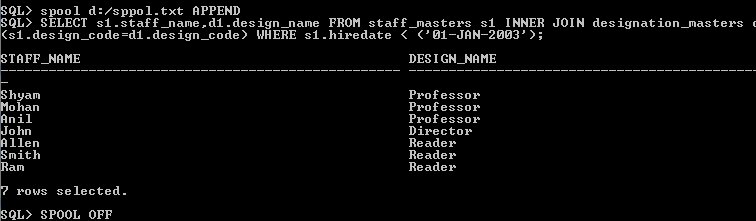


SELECT \* FROM staff\_masters s1 INNER JOIN designation\_masters d1 ON (s1.design\_code=d1.design\_code) where d1.design\_name IN ('Professor','Lecturer');

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

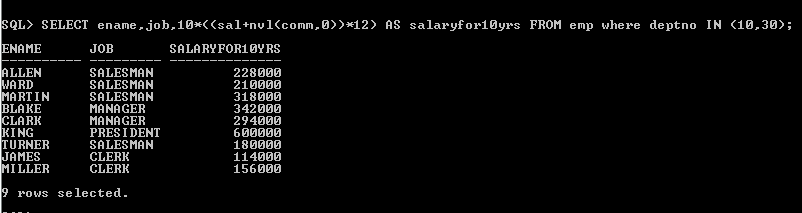
SELECT empno, ename, deptno, FLOOR((SYSDATE - hiredate)/365) AS experience from emp where floor((sysdate-hiredate)/365)>18;

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



SELECT s1.staff\_name,d1.design\_name FROM staff\_masters s1 INNER JOIN designation\_masters d1 ON (s1.design\_code=d1.design\_code) WHERE s1.hiredate < ('01-JAN-2003');

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

Working in departments 10 and 30.

SELECT ename,job,10\*((sal+nvl(comm,0))\*12) AS salaryfor10yrs FROM emp where deptno IN (10,30);

8. List the name and experience (in years) of employees who are working as

LECTURER.



SELECT staff\_name,FLOOR((sysdate-hiredate)/365) AS experience from staff\_masters s1 INNER JOIN designation\_masters d1 ON (s1.design\_code=d1.design\_code) WHERE d1.design\_name='Lecturer';

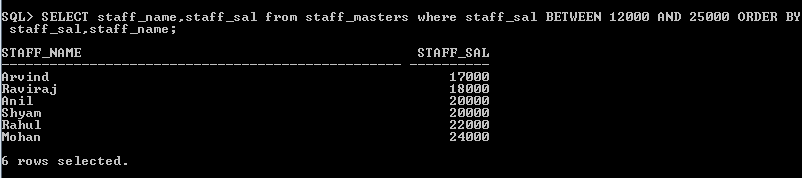
9. Display name concatenated with dept code separated by comma and space. Name

the column as ‘Student Info’.

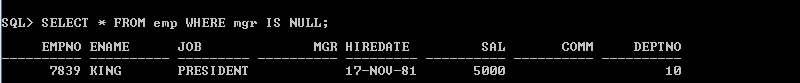


SELECT CONCAT(CONCAT(student\_name,', '),dept\_code) AS student\_info FROM student\_masters;

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



SELECT staff\_name,staff\_sal from staff\_masters where staff\_sal BETWEEN 12000 AND 25000 ORDER BY staff\_sal,staff\_name;

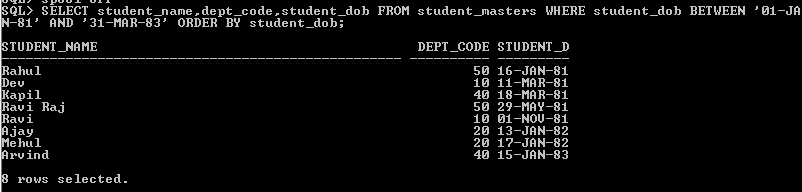
11. Display employees who do not have manager. 

SELECT \* FROM emp WHERE mgr IS NULL;

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).



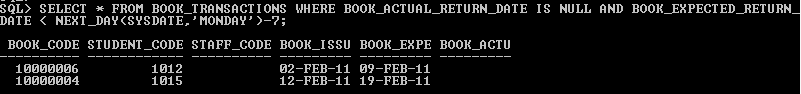
SELECT student\_name,dept\_code,student\_dob FROM student\_masters WHERE student\_dob BETWEEN '01-JAN-81' AND '31-MAR-83' ORDER BY student\_dob;

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



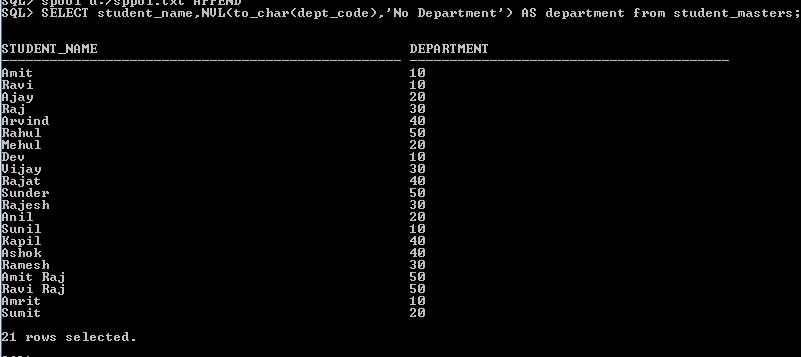
SELECT dept\_code, SUM(staff\_sal) FROM staff\_masters NATURAL JOIN designation\_masters where UPPER(design\_name) != 'MANAGER' GROUP BY dept\_code HAVING SUM(staff\_sal)>20000;

14. Display the details of books that have not been returned and expected return date was last Monday.



 SELECT \* FROM BOOK\_TRANSACTIONS WHERE BOOK\_ACTUAL\_RETURN\_DATE IS NULL AND BOOK\_EXPECTED\_RETURN\_DATE < NEXT\_DAY(SYSDATE,'MONDAY')-7;

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)



SELECT student\_name,NVL(to\_char(dept\_code),'No Department') AS department from student\_masters;

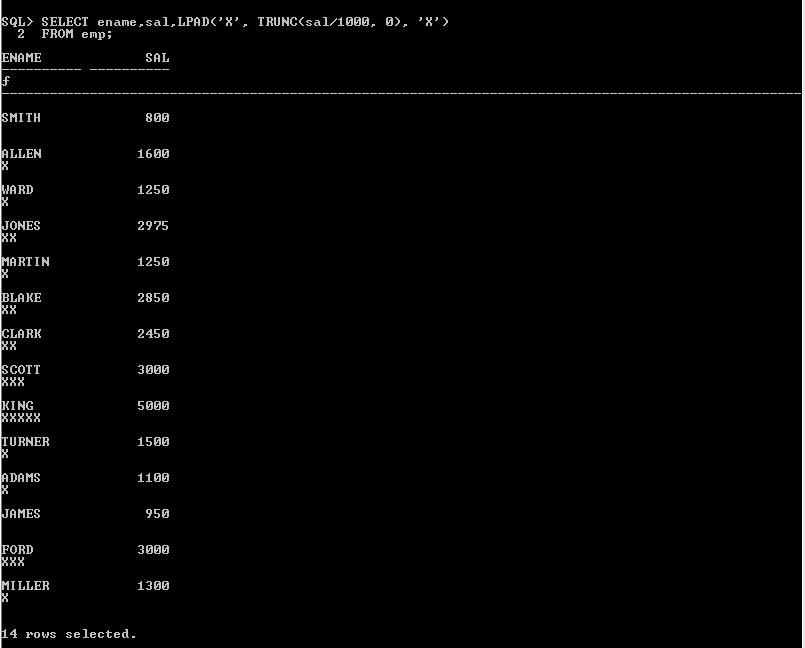
16. Display the name and salary of the staff. Salary should be represented as X. Each X

represents a 1000 in salary.

Sample Output

JOHN 10000 XXXXXXXXXX

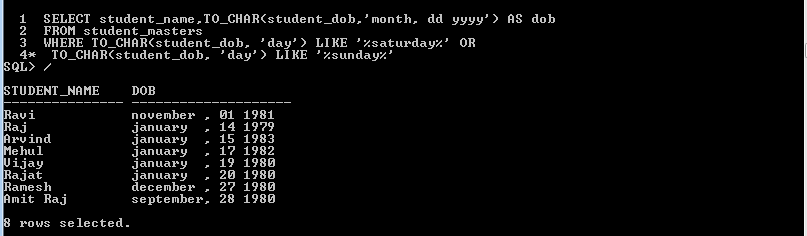
ALLEN 12000 XXXXXXXXXXXX



SELECT ename,sal,LPAD('X', TRUNC(sal/1000, 0), 'X') FROM emp;

**2.1 : Single Row Functions:**

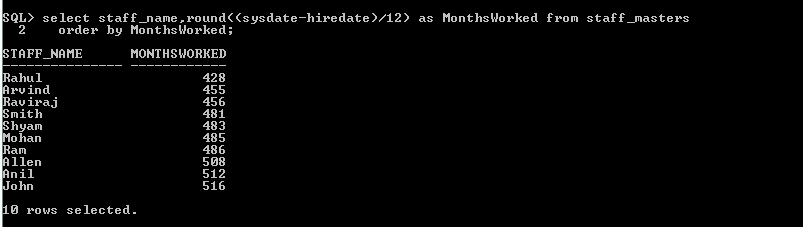
1. Display name and date of birth of students where date of birth must be displayed in the format similar to “January, 12 1981” for those who were born on Saturday or Sunday.



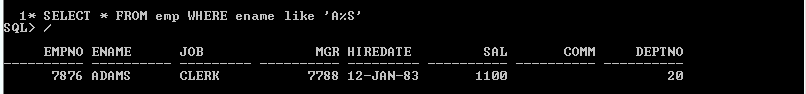
SELECT student\_name,TO\_CHAR(student\_dob,'month, dd yyyy') AS dob FROM student\_masters

WHERE TO\_CHAR(student\_dob, 'day') LIKE '%saturday%' OR TO\_CHAR(student\_dob, 'day') LIKE '%sunday%';

2. Display each staff name and number of months they worked for the organization. Label the column as ‘Months Worked’. Order your result by number of months employed. Round the number of months to closest whole number.



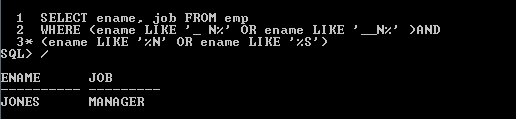
select staff\_name,round((sysdate-hiredate)/12) as MonthsWorked from staff\_masters order by MonthsWorked;

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

with ‘S’.

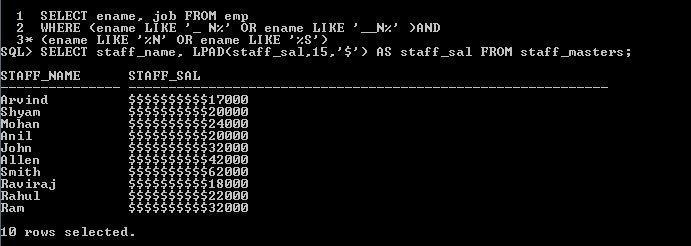
SELECT \* FROM emp WHERE ename like 'A%S';

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’.



SELECT ename, job FROM emp WHERE (ename LIKE '\_ N%' OR ename LIKE '\_\_N%' )AND (ename LIKE '%N' OR ename LIKE '%S')

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 ‘$’.



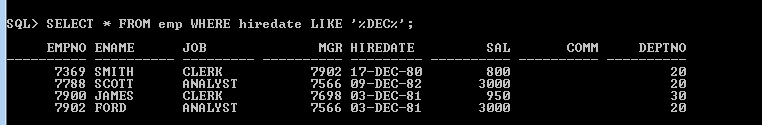
SELECT staff\_name, LPAD(staff\_sal,15,’$’) AS staff\_sal FROM staff\_masters;

6. List the names of the Employees having ‘\_’ character in their name.



SELECT ename FROM emp WHERE ename LIKE '%\\_%' ESCAPE '\';

7. List the details of the employees who have joined in December (irrespective of the year).



SELECT \* FROM emp WHERE hiredate LIKE '%DEC%'

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

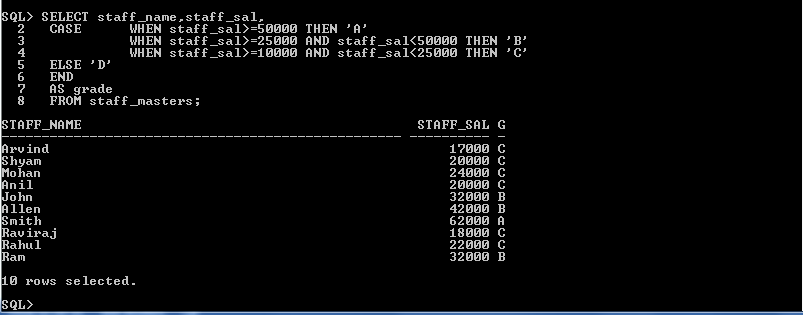
Salary Grade

Salary >=50000 A

Salary >= 25000 < 50000 B

Salary>=10000 < 25000 C

OTHERS D



SELECT staff\_name, staff\_sal,

CASE WHEN staff\_sal>=50000 THEN ‘A’

WHEN staff\_sal>=25000 AND staff\_sal<50000 THEN ‘B’

WHEN staff\_sal>=10000 AND staff\_sal<25000 THEN ‘C’

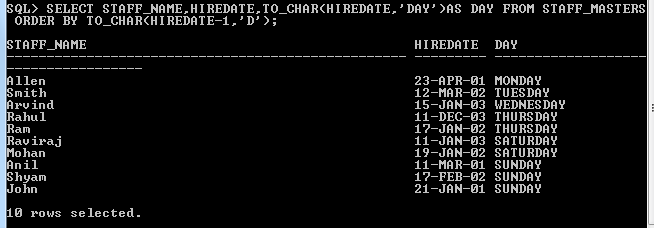
ELSE ‘D’

END

AS grade

FROM staff\_masters;

9. Display the Staff Name, Hire date and day of the week on which staff was hired. Label the column as DAY. Order the result by the day of the week starting with Monday.



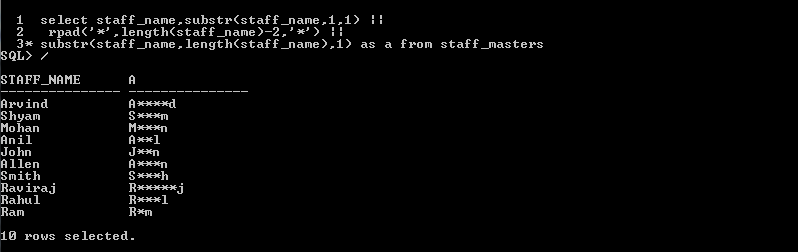
SELECT staff\_name, hiredate, TO\_CHAR(hiredate, ‘day’) AS day FROM staff\_masters ORDER BY TO\_CHAR(hirtedate - 1,’D’);

10. Show staff names with the respective numbers of asterisk from Staff table.



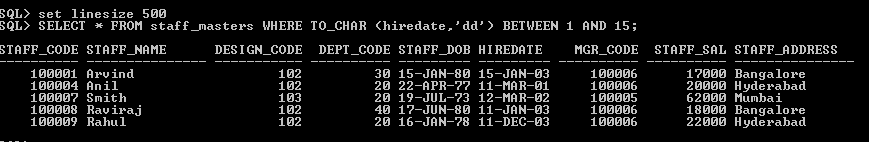
SELECT LPAD(‘\*’,LENGTH(staff\_name),’\*’) AS staff\_name FROM staff\_masters;

11. Show staff names with the respective numbers of asterisk from Staff table except first and last characters. For example: KING will be replaced with K\*\*G.



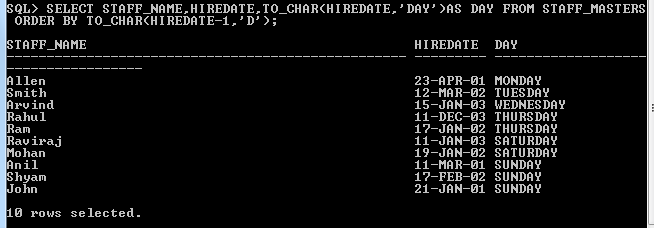
select staff\_name,substr(staff\_name,1,1) ||rpad('\*',length(staff\_name)-2,'\*') ||substr(staff\_name,length(staff\_name),1) as a from staff\_masters;

12. Show all staffs who were hired in the first half of the month.



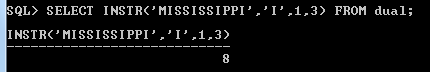
SELECT \* FROM staff\_masters WHERE TO\_CHAR (hiredate,'dd') BETWEEN 1 AND 15

13. Display the staff name, hire date and day of the week on which the staff joined. Order the results by the day of the week starting with Monday.



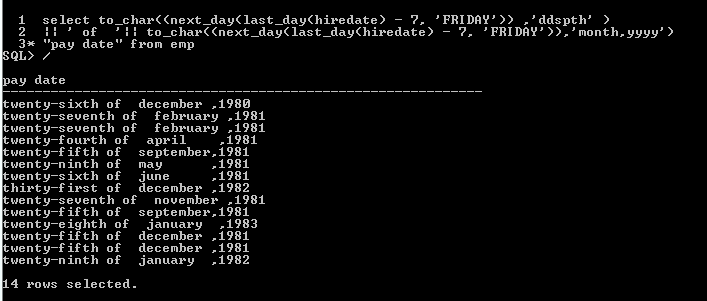
SELECT staff\_name, hiredate, TO\_CHAR(hiredate, ‘day’) FROM staff\_masters ORDER BY TO\_CHAR(hirtedate – 1,’D’);

14. Write a query to find the position of third occurrence of ‘i’ in the given word ‘Mississippi’.



SELECT INSTR(‘MISSISSIPPI’,’I’,2,3) FROM dual;

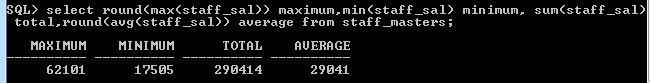
15. Write a query to find the pay date for the month. Pay date is the last Friday of the month. Display the date in the format “Twenty Eighth of January, 2002”. Label the heading as PAY DATE.



select to\_char((next\_day(last\_day(hiredate) - 7, 'FRIDAY')) ,'ddspth' ) || ' of  ' || to\_char((next\_day(last\_day(hiredate) - 7, 'FRIDAY')) ,'month,yyyy' ) "pay date" from emp

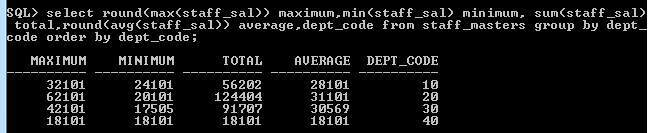
* 1. **GROUP FUNCTIONS**

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.



SELECT ROUND(MAX(staff\_sal)) AS maximum, ROUND(MIN(staff\_sal)) AS minimum, ROUND(SUM(staff\_sal)) AS total, ROUND(AVG(staff\_sal)) AS average FROM staff\_masters;

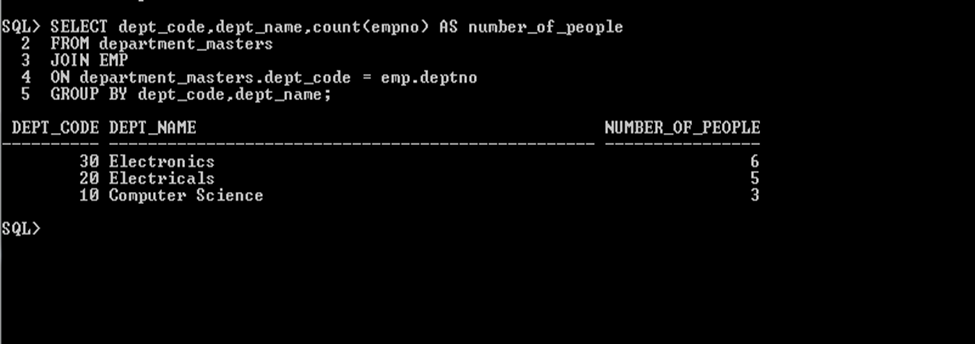
17. Edit the above query and display the same for each Department Name.



SELECT ROUND(MAX(staff\_sal)) AS maximum, ROUND(MIN(staff\_sal)) AS minimum, ROUND(SUM(staff\_sal)) AS total, ROUND(AVG(staff\_sal)) AS average FROM staff\_masters group by dept\_code order by dept\_code

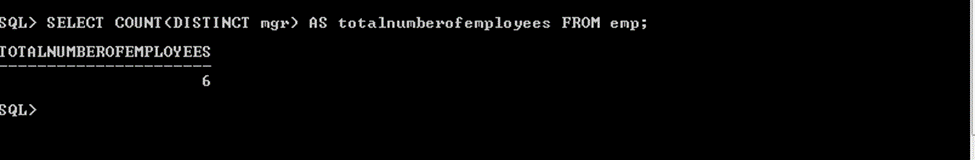
;

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



SELECT dept\_code, dept\_name, count(empno) AS number\_of\_people FROM department\_masters JOIN emp ON department\_masters.dept\_code = emp.deptno GROUP BY dept\_code, dept\_name;

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

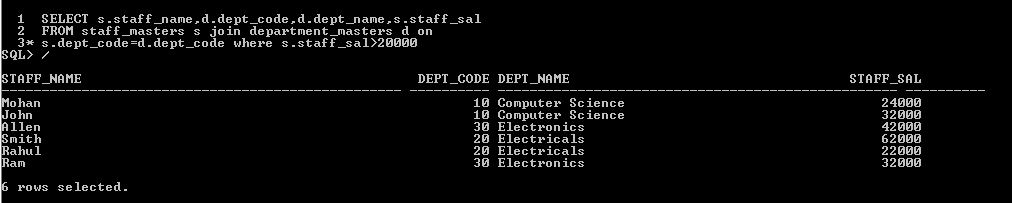


SELECT COUNT( DISTINCT mgr) AS total numberofmanagers FROM emp;

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.

**4.1 : Joins and Subqueries**

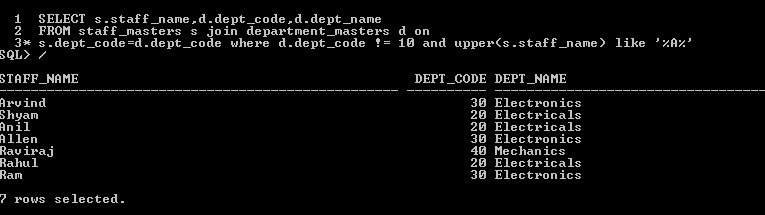
1. Write a query which displays Staff Name, Department Code, Department Name, and Salary for all staff who earns more than 20000.



SELECT s.staff\_name,d.dept\_code,d.dept\_name,s.staff\_sal FROM staff\_masters s join department\_masters d on

s.dept\_code=d.dept\_code where s.staff\_sal>20000

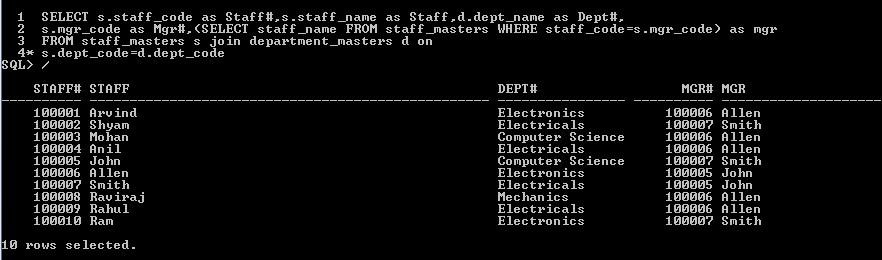
2. Write a query to display Staff Name, Department Code, and Department Name for all staff who do not work in Department code 10 and have ‘A’ in their name.



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 upper(s.staff\_name) like '%A%'

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



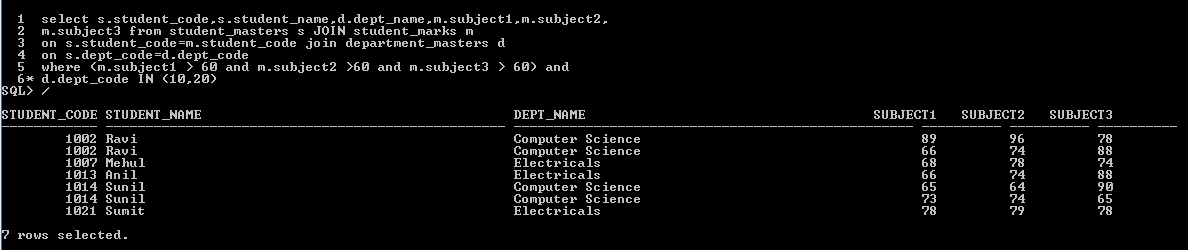
SELECT s.staff\_code AS Staff#,s.staff\_name AS Staff,d.dept\_name AS Dept#,

s.mgr\_code AS Mgr#,(SELECT staff\_name FROM staff\_masters WHERE staff\_code=s.mgr\_code) AS mgr

FROM staff\_masters s JOIN department\_masters d ON

s.dept\_code=d.dept\_code

4. 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.



select s.student\_code,s.student\_name,d.dept\_name,m.subject1,m.subject2,

m.subject3 from student\_masters s JOIN student\_marks m

on s.student\_code=m.student\_code join department\_masters d

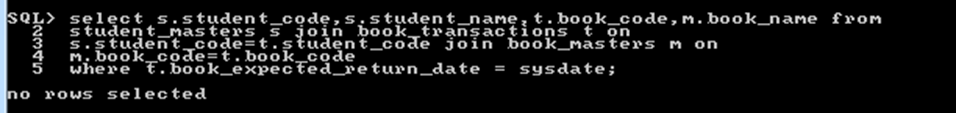
on s.dept\_code=d.dept\_code

where (m.subject1 > 60 and m.subject2 >60 and m.subject3 > 60) and

d.dept\_code IN (10,20)

/

5. Create a query that will display Student Code, Student Name, Book Code, and Book Name for all students whose expected book return date is today.



select s.student\_code,s.student\_name,t.book\_code,m.book\_name from

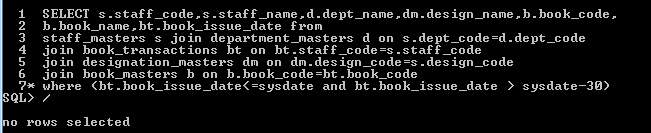
student\_masters s join book\_transactions t on

s.student\_code=t.student\_code join book\_masters m on

m.book\_code=t.book\_code

where t.book\_expected\_return\_date = sysdate;

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.



SELECT s.staff\_code,s.staff\_name,d.dept\_name,dm.design\_name,b.book\_code,

b.book\_name,bt.book\_issue\_date from

staff\_masters s join department\_masters d on s.dept\_code=d.dept\_code

join book\_transactions bt on bt.staff\_code=s.staff\_code

join designation\_masters dm on dm.design\_code=s.design\_code

join book\_masters b on b.book\_code=bt.book\_code

where (bt.book\_issue\_date<=sysdate and bt.book\_issue\_date > sysdate-30)

7. Generate a report which contains the following information.

Staff Code Staff Name Designation Department Name Department Head

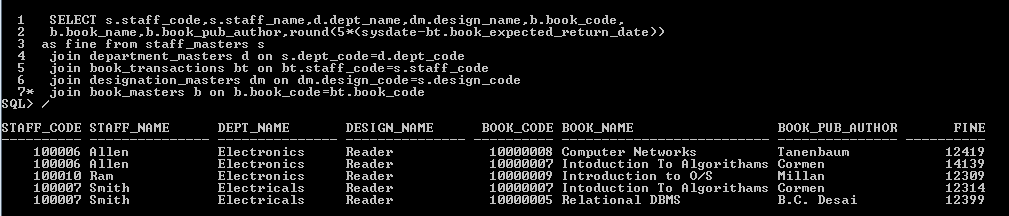
For all staff excluding HOD (List should not contain the details of Department head).

8. Generate a report which contains the following information Student Code Student Name Department Name Total Marks HOD Name Sort the output on Department Name and Total Marks.

9. Generate a report which contains the following information. Staff Code, Staff Name, Designation, Department, Book Code, Book Name,

Author, Fine For the staff who have not return the book. Fine will be calculated as Rs. 5 per day.

Fine = 5 \* (No. of days = Current Date – Expected return date).



SELECT s.staff\_code,s.staff\_name,d.dept\_name,dm.design\_name,b.book\_code,

b.book\_name,b.book\_pub\_author,round(5\*(sysdate-bt.book\_expected\_return\_date))

as fine from staff\_masters s

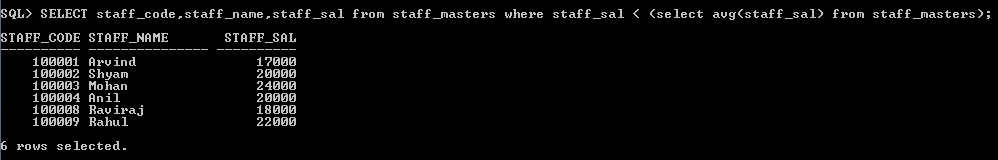
join department\_masters d on s.dept\_code=d.dept\_code

join book\_transactions bt on bt.staff\_code=s.staff\_code

join designation\_masters dm on dm.design\_code=s.design\_code

join book\_masters b on b.book\_code=bt.book\_code

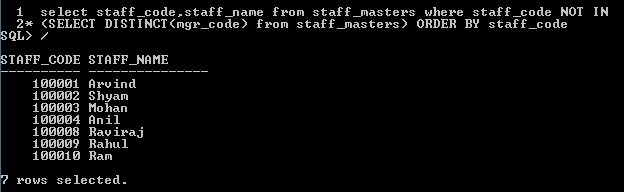
10. List Staff Code, Staff Name, and Salary for those who are getting less than the average salary of organization.



SELECT staff\_code,staff\_name,staff\_sal from staff\_masters where staff\_sal < (select avg(staff\_sal) from staff\_masters)

/

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



select staff\_code,staff\_name from staff\_masters where staff\_code NOT IN

(SELECT DISTINCT(mgr\_code) from staff\_masters) ORDER BY staff\_code

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

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

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

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

16. 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.

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

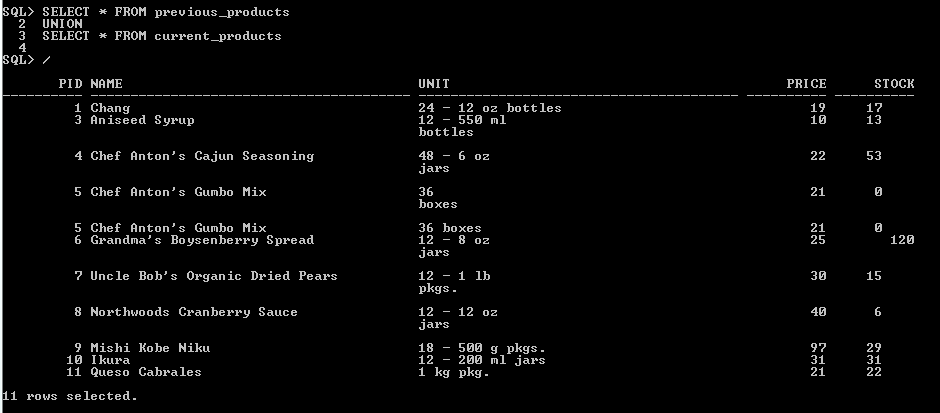
18. Display the Student Code, Student Name, and Department Name for that department in which there are maximum number of student are studying.

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.

**5.1 : Set Operators**

1. Get the details of all products irrespective of the fact whether they are in previous set or current set.

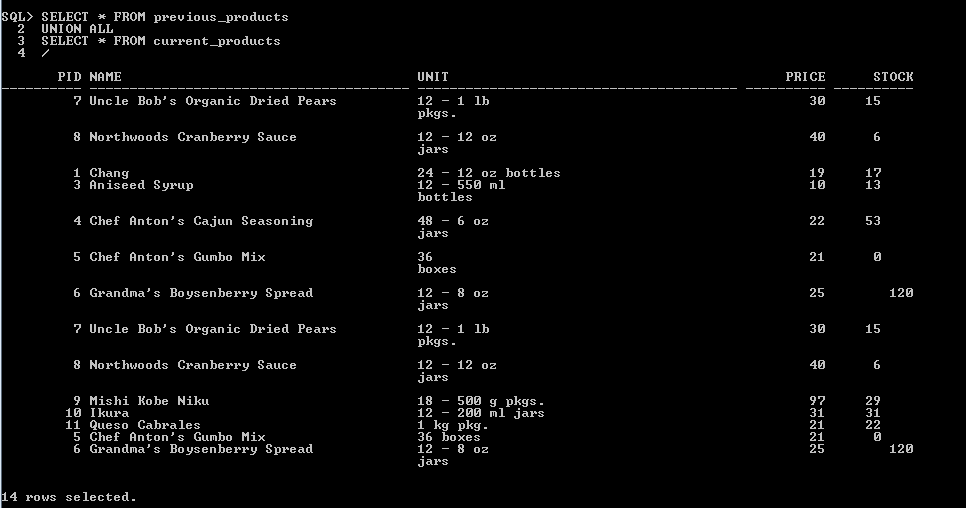


SELECT \* FROM previous\_products

UNION

SELECT \* FROM current\_products

2. Get the details of all products along with the repetition of those that were present both in the previous and current sets.

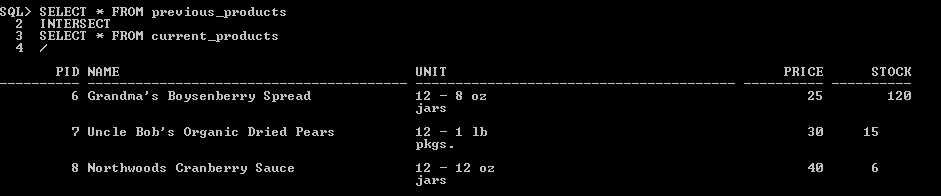


SELECT \* FROM previous\_products

UNION ALL

SELECT \* FROM current\_products

3. Get the details of only those products which were present in the previous set and are still continuing.



SELECT \* FROM previous\_products

INTERSECT

SELECT \* FROM current\_products

4. Get the details of all obsolete products (no longer continued).



SELECT \* FROM previous\_products

MINUS

SELECT \* FROM current\_products

**6.1 4.1: Database Objects**

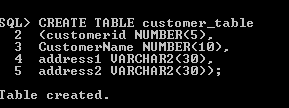
1. Create the Customer table with the following columns.

Customerid Number(5)

CustomerName Number(10)

Address1 Varchar2(30)

Address2 Varchar2(30)



CREATE TABLE customer\_table

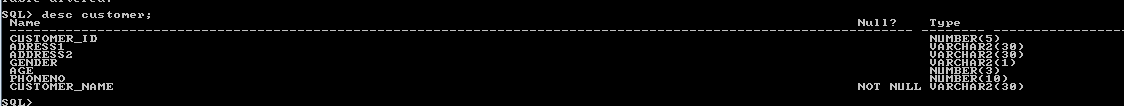
(customerid NUMBER(5),

CustomerName NUMBER(10),

address1 VARCHAR2(30),

address2 VARCHAR2(30));

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



ALTER TABLE customer\_table MODIFY customername VARCHAR2(30) NOT NULL;

3. Add the following Columns to the Customer table.

Gender Varchar2(1)

Age Number(3)

PhoneNo Number(10)



ALTER TABLE customer\_table ADD (gender VARCHAR2(1), age NUMBER(3),

phoneno NUMBER(10))

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

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

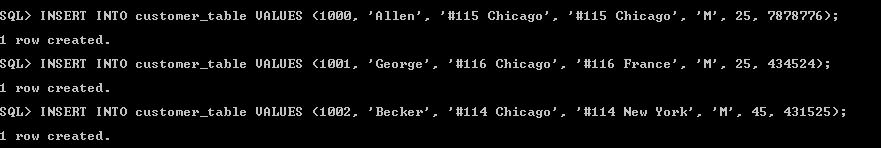
7878776’)

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



INSERT INTO customer\_table VALUES (1000, 'Allen', '#115 Chicago', '#115 Chicago', 'M', 25, 7878776)

INSERT INTO customer\_table VALUES (1001, 'George', '#116 Chicago', '#116 France', 'M', 25, 434524)

INSERT INTO customer\_table VALUES (1002, 'Becker', '#114 Chicago', '#114 New York', 'M', 45, 431525)

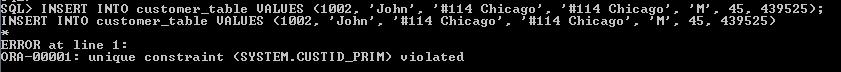
5. Add the Primary key constraint for Customerld with the name Custld\_Prim



ALTER TABLE customer\_table ADD CONSTRAINT custid\_prim PRIMARY KEY(customerid);

6. 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

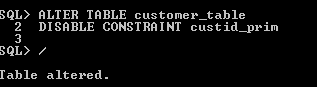


INSERT INTO customer\_table VALUES (1002, 'John', '#114 Chicago', '#114 Chicago', 'M', 45, 439525)

7. 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



ALTER TABLE customer\_table

DISABLE CONSTRAINT custid\_prim



INSERT INTO customer\_table VALUES (1002, 'Becker', '#114 New York', '#114 New York', 'M', 45, 431525);

INSERT INTO customer\_table VALUES (1003, 'Nanapatekar', '#114 India', '#114 New York', 'M', 45, 431525);

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



ALTER TABLE customer\_table ENABLE CONSTRAINT custid\_prim;

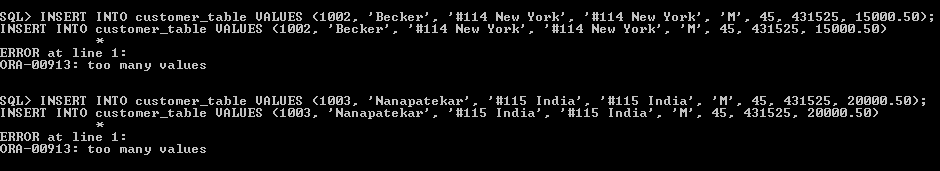
9. Drop the constraint Custld\_Prim on CustomerId and insert the following Data. Alter Customer table, drop constraint Custid\_Prim.

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

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



ALTER TABLE customer\_table DROP CONSTRAINT custid\_prim;



INSERT INTO customer\_table VALUES (1002, 'Becker', '#114 New York', '#114 New York', 'M', 45, 431525, 15000.50);

INSERT INTO customer\_table VALUES (1003, 'Nanapatekar', '#115 India', '#115 India', 'M', 45, 431525, 20000.50);

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



TRUNCATE TABLE customer\_table;

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



alter table customer\_table add(e\_mail varchar(30))

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



ALTER TABLE customer\_table DROP COLUMN E\_mail;

13. Add a new column EmailId to Customer table.



ALTER TABLE customer\_table ADD EmailId VARCHAR2(30);

14. Mark EmailId column as unused before dropping it.



ALTER TABLE customer\_table SET UNUSED (EmailId);

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



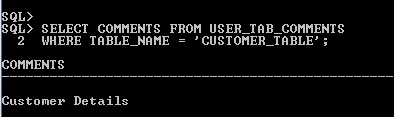
ALTER TABLE customer\_table DROP UNUSED columns;

16. Define the COMMENT ‘Customers Details’ for Customer table.



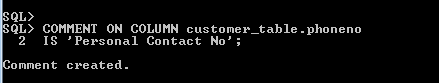
COMMENT ON TABLE customer\_table IS 'Customer Details';

17. Use Data Dictionary USER\_TAB\_COMMENTS to view the created comment.



SELECT COMMENTS FROM USER\_TAB\_COMMENTS  
WHERE TABLE\_NAME = 'CUSTOMER\_TABLE';

18. Define the COMMENT ‘Personal Contact no’ for the phoneno column of the Customer table.



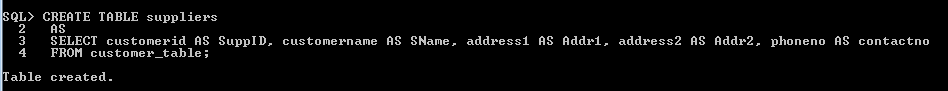
COMMENT ON COLUMN customer\_table.phoneno  
IS 'Personal Contact No';

19. Use Data Dictionary USER\_COL\_COMMENTS to view the created comment.



SELECT COMMENTS FROM USER\_COL\_COMMENTS  
WHERE TABLE\_NAME = 'CUSTOMER\_TABLE';

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.



CREATE TABLE suppliers

AS

SELECT customerid AS SuppID, customername AS SName, address1 AS Addr1, address2 AS Addr2, phoneno AS contactno

FROM customer\_table;

21. Drop the a/bove 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

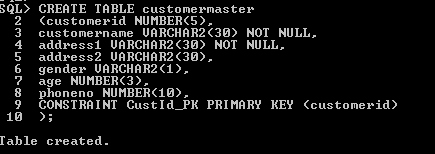
Address2 Varchar2(30)  
Gender Varchar2(l)

Age Number(3)

PhoneNo Number(10)



DROP table suppliers;



CREATE TABLE customermaster

(customerid NUMBER(5),

customername VARCHAR2(30) NOT NULL,

address1 VARCHAR2(30) NOT NULL,

address2 VARCHAR2(30),

gender VARCHAR2(1),

age NUMBER(3),

phoneno NUMBER(10),

CONSTRAINT CustId\_PK PRIMARY KEY (customerid)

);

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

generate Account number

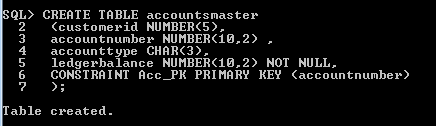
Customerld Number(5)

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

Acc\_PK)

AccountType Char(3)

LedgerBalance Number(10,2) Not Null



CREATE TABLE accountsmaster

(customerid NUMBER(5),

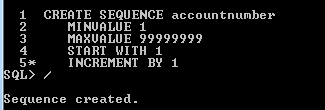
accountnumber NUMBER(10,2) ,

accounttype CHAR(3),

ledgerbalance NUMBER(10,2) NOT NULL,

CONSTRAINT Acc\_PK PRIMARY KEY (accountnumber)

);



CREATE SEQUENCE accountnumber

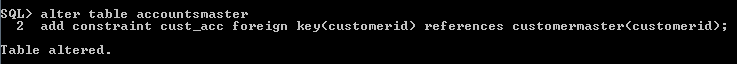
MINVALUE 1

MAXVALUE 99999999

START WITH 1

INCREMENT BY 1

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



alter table accountsmaster

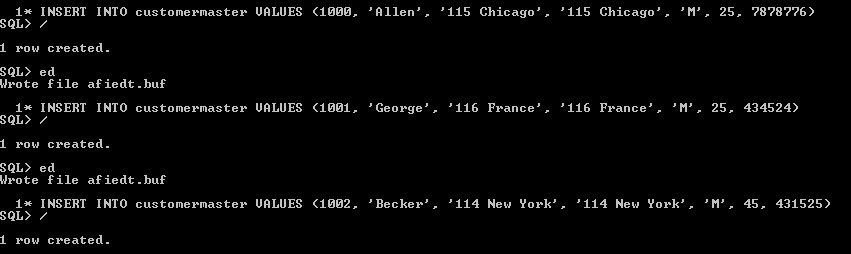
add constraint cust\_acc foreign key(customerid) references customermaster(customerid);

24. Insert the following rows to the CustomerMaster 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



INSERT INTO customermaster VALUES (1000, 'Allen', '115 Chicago', '115 Chicago', 'M', 25, 7878776);

INSERT INTO customermaster VALUES (1001, 'George', '116 France', '116 France', 'M', 25, 434524);

INSERT INTO customermaster VALUES (1002, 'Becker', '114 New York', '114 New York', 'M', 45, 431525);

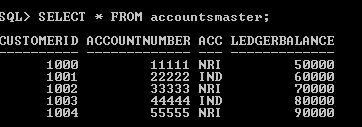
25. Modify the AccountMaster table with the Check constraint to ensure AccountType

should be either NRI or IND.



alter table accountsmaster modify (accounttype check(accounttype in('NRI','IND')));

26. Insert 5 rows into the AccountsMaster table:



insert into accountsmaster values(1000,11111,'NRI',50000);

insert into accountsmaster values(1001,22222,'IND',60000);

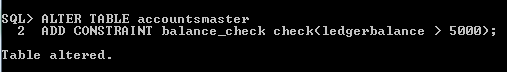
insert into accountsmaster values(1002,33333,'NRI',70000);

insert into accountsmaster values(1003,44444,'IND',80000);

insert into accountsmaster values(1004,55555,'NRI',90000);

27. Modify the AccountsMaster table keeping a Check constraint with the name

Balance\_Check for the Minimum Balance which should be greater than 5000.



ALTER TABLE accountsmaster

ADD CONSTRAINT balance\_check check(ledgerbalance > 5000);

28. Modify the AccountsMaster table such that if Customer is deleted from Customer table then all his details should be deleted from AccountsMaster table.

29. Create Backup copy for the AccountsMaster table with the name ‘AccountDetails’.



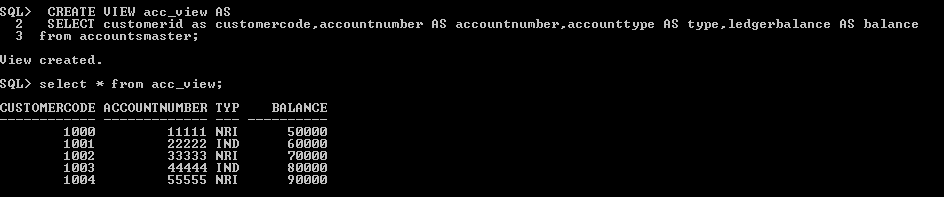
CREATE TABLE accountdetails AS (SELECT \* FROM accountsmaster);

**30.** Change the name of the AccountDetails table to ‘BackUpTable’ table.



ALTER TABLE accountdetails RENAME TO BackUpTable;

31. Create a view ‘Acc\_view’ with columns Customerld, CustomerName, AccountNumber, AccountType, and LedgerBalance from AccountsMaster. In the view Acc\_view, the column names should be CustomerCode, AccountHolderName, AccountNumber, Type, and Balance for the respective columns from AccountsMaster table.

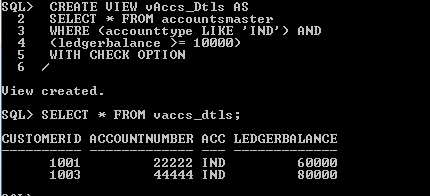


CREATE VIEW acc\_view AS

SELECT customerid as customercode,accountnumber AS accountnumber,accounttype AS type,ledgerbalance AS balance

from accountsmaster;

32. Create a view on AccountsMaster table with name vAccs\_Dtls. This view should list all customers whose AccountType is ‘IND’ and their balance amount should not be less than 10000. Using this view any DML operation should not violate the view conditions



CREATE VIEW vAccs\_Dtls AS

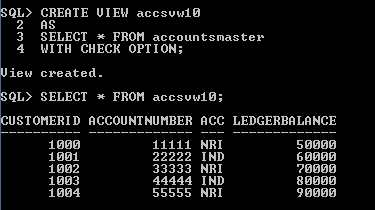
SELECT \* FROM accountsmaster

WHERE (accounttype LIKE 'IND') AND

(ledgerbalance >= 10000)

WITH CHECK OPTION

33. Create a view accsvw10 which will not allow DML statement against it.



CREATE VIEW accsvw10

AS

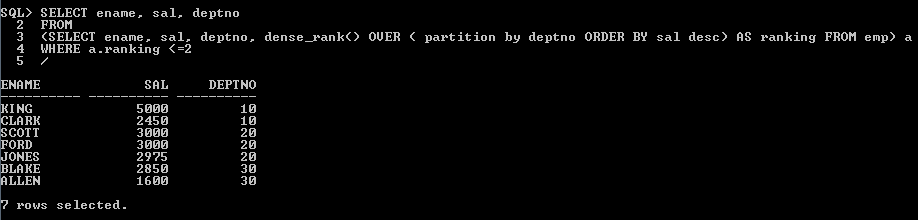
SELECT \* FROM accountsmaster

WITH CHECK OPTION;

34. Display the department from Staff table which has the highest salary by using Inline

View.

35.List the top two highest earning employees in each department.



SELECT ename, sal, deptno

FROM

(SELECT ename, sal, deptno, dense\_rank() OVER ( partition by deptno ORDER BY sal desc) AS ranking FROM emp) a

WHERE a.ranking <=2