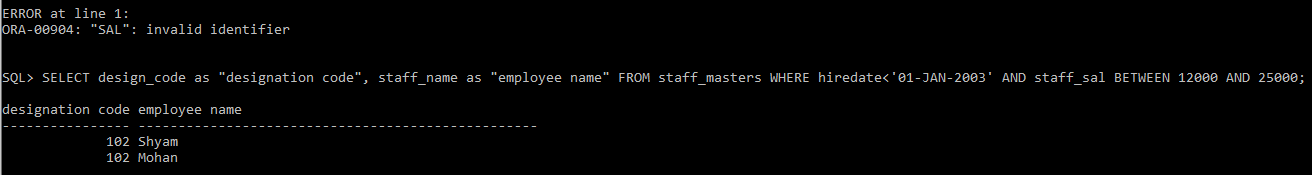
**1.1**: Data Query Language

1. List the Name and Designation code of the staff who have joined before Jan 2003 and whose salary range is between 12000 and 25000. Display the columns with user defined Column headers. Hint: Use As clause along with other operators

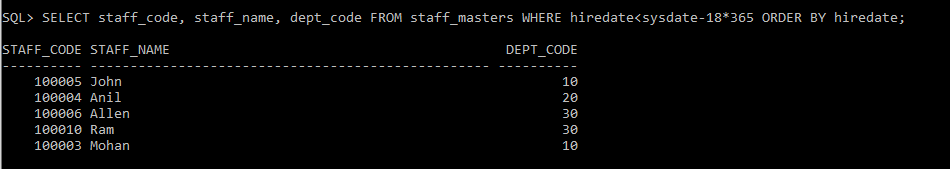
Query:

SELECT design\_code as "designation code", staff\_name as "employee name" FROM staff\_masters WHERE hiredate<'01-JAN-2003' AND sal BETWEEN 12000 AND 25000; 

2. List the staff code, name, and department number of the staff who have experience of 18 or more years and sort them based on their experience.

Query:

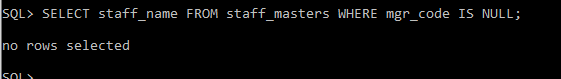
SELECT staff\_code, staff\_name, dept\_code FROM staff\_masters WHERE hiredate<sysdate-18\*365 ORDER BY hiredate;



3. Display the staff details who do not have manager. Hint: Use is null

Query:

SELECT staff\_name FROM staff\_masters WHERE mgr\_code IS NULL;



4. Display the Book details that were published during the period of 2001 to 2004. Also display book details with Book name having the character ‘&’ anywhere.

Query:

SELECT \* FROM book\_masters WHERE book\_pub\_year BETWEEN 2001 AND 2004 AND book\_name LIKE '%&%';

C:\Users\ADM-IG-HWDLAB2D\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.4.png

5. List the names of the staff having ‘\_’ character in their name

Query:

select staff\_name from staff\_masters WHERE staff\_name LIKE '%\\_%' ESCAPE'\';

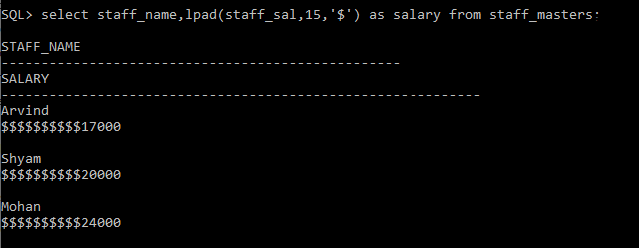


**2.1**: Single Row Functions:

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

Query:

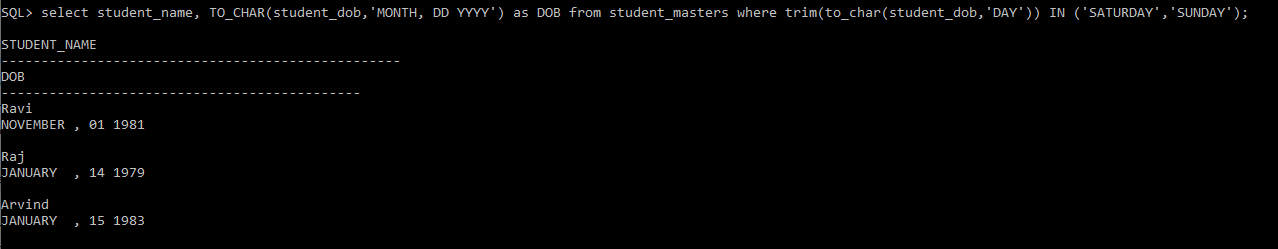
select staff\_name,lpad(staff\_sal,15,'$') as salary from staff\_masters;



2. 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.

Query:

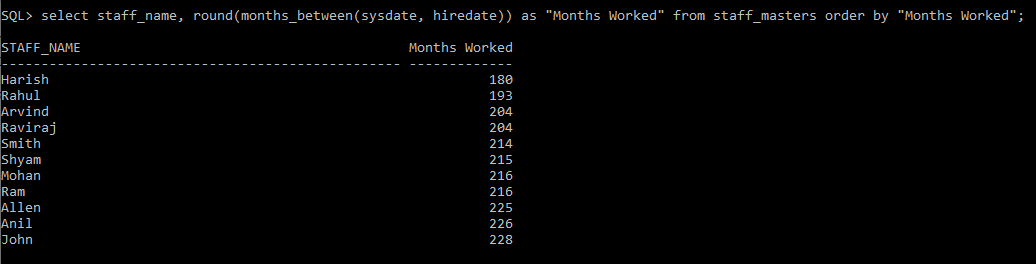
select student\_name, TO\_CHAR(student\_dob,'MONTH, DD YYYY') as DOB from student\_masters where trim(to\_char(student\_dob,'DAY')) IN ('SATURDAY','SUNDAY');



3. 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. Also Round the number of months to closest whole number.

Query:

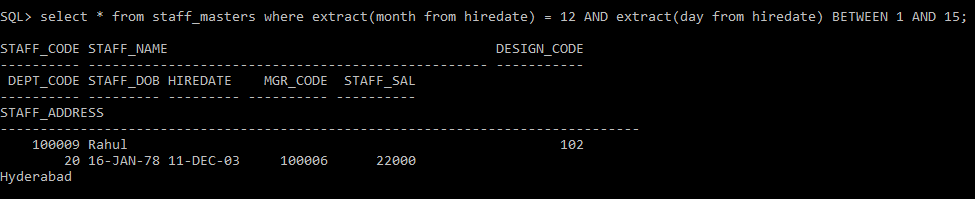
select staff\_name, round(months\_between(sysdate, hiredate)) as "Months Worked" from staff\_masters order by "Months Worked";



4. List the details of the staff who have joined in first half of December month (irrespective of the year).

Query:

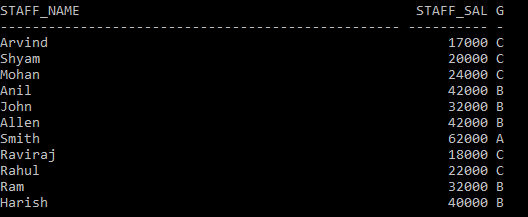
select \* from staff\_masters where extract(month from hiredate) = 12 AND extract(day from hiredate) BETWEEN 1 AND 15;



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

Query:

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;

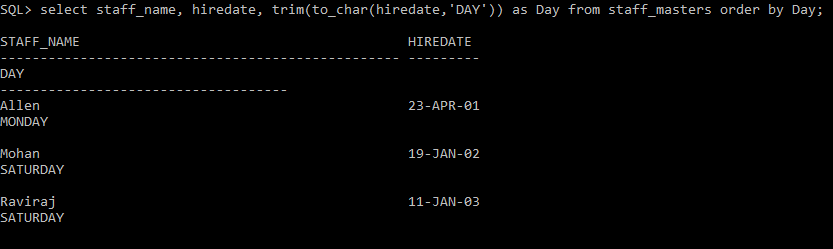


Salary Grade Salary >=50000 A Salary >= 25000 < 50000 B Salary>=10000 < 25000 C OTHERS D

6. 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. Hint :Use to\_char with hiredate and formats ‘DY’ and ’D’

Query:

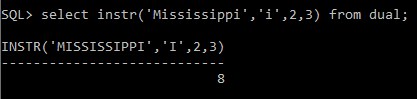
select staff\_name, hiredate, trim(to\_char(hiredate,'DAY')) as Day from staff\_masters order by Day;



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

Query:

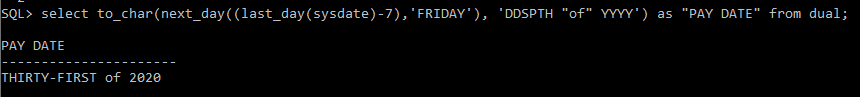
select instr('Mississippi','i',2,3) from dual;



8. 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. Hint: use to\_char, next\_day and last\_day functions

Query:

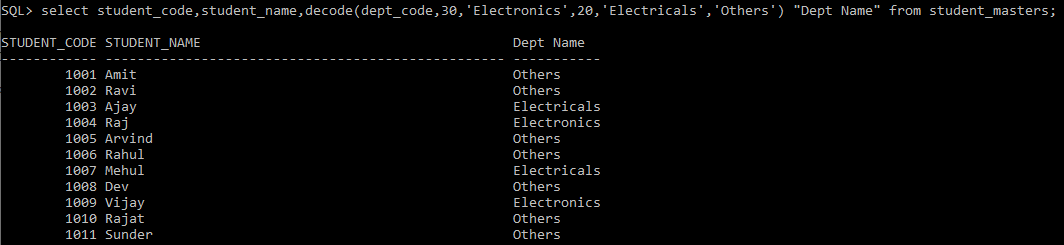
select to\_char(next\_day((last\_day(sysdate)-7),'FRIDAY'), 'DDSPTH "of" YYYY') as "PAY DATE" from dual;



9. Display Student code, Name and Dept Name. Display “Electricals” if dept code = 20, “Electronics” if Dept code =30 and “Others” for all other Dept codes in the Dept Name column. Hint : Use Decode

Query:

select student\_code,student\_name,decode(dept\_code,30,'Electronics',20,'Electricals','Others') "Dept Name" from student\_masters;

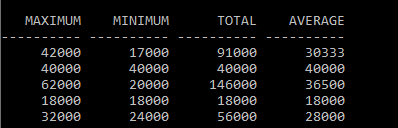


**2.2**: Group Functions:

1. Display the Highest, Lowest, Total & Average salary of all staff. Label the columns Maximum, Minimum, Total and Average respectively for each Department code. Also round the result to the nearest whole number.

Query:

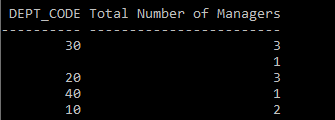
select round(max(staff\_sal)) Maximum,round(min(staff\_sal)) Minimum, round(sum(staff\_sal)) Total,round(avg(staff\_sal)) Average from staff\_masters group by dept\_code;



2. Display Department code and number of managers working in that department. Label the column as ‘Total Number of Managers’ for each department.

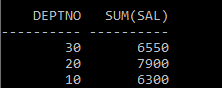
Query:

select dept\_code,count(distinct(mgr\_code)) as "Total Number of Managers" from staff\_masters group by dept\_code;



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

Query:  
select deptno, sum(sal) from emp where job!='MANAGER' group by deptno having sum(sal)>5000;

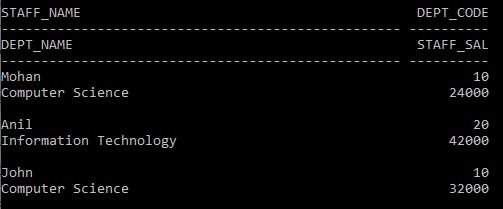


**3.1**: Joins and Subqueries

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

Query:

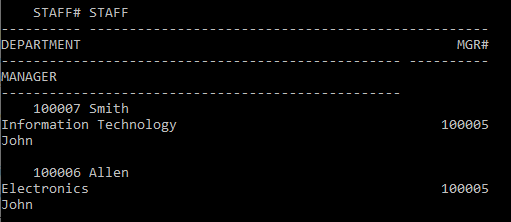
select s.staff\_name,s.dept\_code,d.dept\_name,s.staff\_sal from staff\_masters s,department\_masters d where s.dept\_code=d.dept\_code and s.staff\_sal>20000;



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

Query:

select s.staff\_code Staff#, s.staff\_name Staff, d.dept\_name Department, s1.staff\_code Mgr#, s1.staff\_name Manager from staff\_masters S,department\_masters D, staff\_masters S1 where s.dept\_code = d.dept\_code and s.mgr\_code = s1.staff\_code;



3. 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.

Query:

select s.student\_code, s.student\_name, t.book\_code,b.book\_name from book\_transactions t, student\_masters s, book\_masters b where t.student\_code = s.student\_code and t.book\_code = b.book\_code and t.book\_expected\_return\_date=sysdate;



4. Create a query that will display Staff Code, Staff Name, Department Name, Designation name, Book Code, Book Name, and Issue Date for only those staff who have taken any book in last 30 days. . If required, make changes to the table to create such a scenario.

Query:

select t.staff\_code, s.staff\_name, d.dept\_name, d1.design\_name, b.book\_name, t.book\_issue\_date from book\_transactions t, staff\_masters s, department\_masters d, designation\_masters d1, book\_masters b where t.staff\_code = s.staff\_code and s.design\_code=d1.design\_code and s.dept\_code = d.dept\_code and t.book\_code = b.book\_code and t.book\_issue\_date BETWEEN sysdate and sysdate-30;



5. Generate a report which contains the following information.

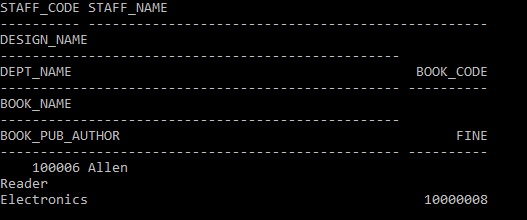
Staff Code, Staff Name, Designation Name, Department, Book Code, Book Name,

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

Fine = 5 \* (No. of days = Current Date – Expected return date). Include records in the table to suit this problem statement

Query:

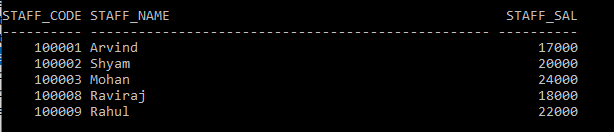
select t.staff\_code, s.staff\_name, d1.design\_name, d.dept\_name, t.book\_code, b.book\_name, b.book\_pub\_author, case when t.book\_actual\_return\_date is null then (5\*(sysdate-to\_date(t.book\_expected\_return\_date,'dd/mm/yyyy'))) when t.book\_actual\_return\_date>t.book\_expected\_return\_date then (5\*(to\_date(t.book\_actual\_return\_date,'dd/mm/yyyy')-to\_date(t.book\_expected\_return\_date,'dd/mm/yyyy'))) end Fine from book\_transactions t, staff\_masters s, designation\_masters d1, department\_masters d, book\_masters b where t.staff\_code = s.staff\_code and t.book\_code = b.book\_code and s.design\_code = d1.design\_code and s.dept\_code = d.dept\_code;



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

Query:

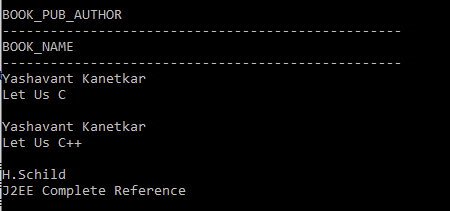
select staff\_code,staff\_name,staff\_sal from staff\_masters where staff\_sal<(select avg(staff\_sal) from staff\_masters);



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

Query:

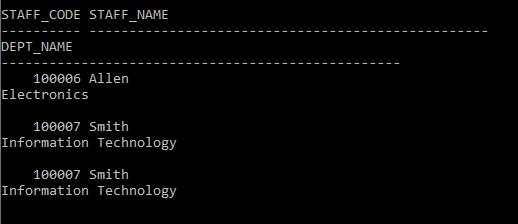
select book\_pub\_author, book\_name from book\_masters where book\_pub\_author in (select book\_pub\_author from book\_masters group by book\_pub\_author having count(book\_code)>1);



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

Query:

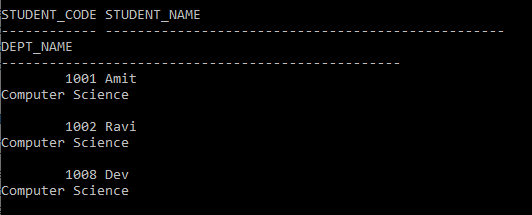
select b.staff\_code, s.staff\_name,d.dept\_name from staff\_masters s, book\_transactions b, department\_masters d where b.staff\_code = s.staff\_code and s.dept\_code = d.dept\_code and b.staff\_code in (select staff\_code from book\_transactions group by staff\_code having count(book\_code)>1);



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

Query:

select s.student\_code, s.student\_name,d.dept\_name from student\_masters s, department\_masters d where s.dept\_code=d.dept\_code and s.dept\_code in (select dept\_code from (select dept\_code, count(student\_code) nstuds from student\_masters group by dept\_code) where nstuds=(select max(count(student\_code)) from student\_masters group by dept\_code));



10. Display Staff Code, Staff Name, Department Name, and Designation name for those who have joined in last 3 months.

Query:

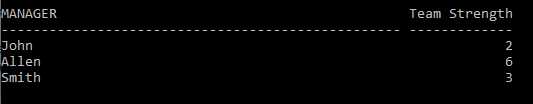
select s.staff\_code, s.staff\_name, d.dept\_name,d1.design\_name from staff\_masters s, department\_masters d, designation\_masters d1 where s.dept\_code=d.dept\_code and s.design\_code = d1.design\_code and s.hiredate between sysdate and sysdate-90;



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

Query:

select s.staff\_name Manager,count(m.staff\_code) "Team Strength" from staff\_masters s, staff\_masters m where m.mgr\_code=s.staff\_code group by s.staff\_name;



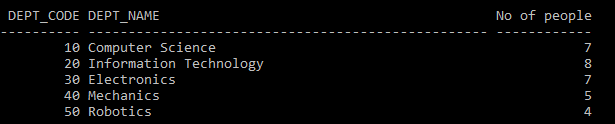
12. Display the details of books that have not been returned and expected return date was last Monday. Book name should be displayed in proper case.. Hint: You can change /add records so that the expected return date suits this problem statement.

Query:

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

Query:

select dept\_code, dept\_name, CNT "No of people" from (select dept\_code, sum(C) "CNT" from (select dept\_code, count(\*) "C" from student\_masters group by dept\_code union all select dept\_code, count(\*) "C" from staff\_masters group by dept\_code) group by dept\_code) natural join department\_masters;



**4.1**: Database Objects

1. Create the Customer table with the following columns.

CustomerId int(5)

Cust\_Name varchar(20)

Address1 Varchar(30)

Address2 Varchar(30)

Query:

create table dummy\_customer( customerid number(5), cust\_name varchar2(20), address1 varchar2(30), address2 varchar2(30));

2. Modify the Customer table Cust\_Name column of datatype with Varchar2(30), rename the column to CustomerName and it should not accept Nulls.

Query:

a. alter table dummy\_customer modify(cust\_name varchar(30) NOT NULL);

b. alter table dummy\_customer rename column cust\_name to customername;

3. a) Add the following Columns to the Customer table.

Gender Varchar(1)

Age int(3)

PhoneNo int(10)

b) Rename the Customer table to Cust\_Table

Query:

a. alter table dummy\_customer add (gender varchar(1), age number(3), phoneno number(10));

b. alter table dummy\_customer rename to cust\_table;

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:

1001, George, #116 France, #116 France, M, 25, 434524

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

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

Query:

alter table cust\_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

Query:

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

Query:

alter table cust\_table disable constraint custid\_prim;

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

Query:

alter table cust\_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

Query:

alter table cust\_table drop constraint custid\_prim;

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

Query:

truncate table cust\_table;

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

Query:

alter table cust\_table add (email varchar2(30));

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

Query:

alter table cust\_table drop column email;

13. 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.

Query:

create table suppliers (suppid,sname,addr1,contactno) as select custid,name,addr,contact from customer;

14. Drop the above table and recreate the following table with the name CustomerMaster.

CustomerId int(5) Primary key(Name of constraint is CustId\_PK)

CustomerName Varchar(30) Not Null

Addressl Varchar(30) Not Null

Address2 Varchar(30)

Gender Varchar(l)

Age int(3)

PhoneNo int(10)

Query:

a. drop table suppliers;

b. create table CustomerMaster(customerid number(5) constraint custid\_pk PRIMARY KEY, customername varchar2(30) NOT NULL, address1 varchar2(30) not null, address2 varchar(30), gender varchar2(1), age number(3), phoneno number(10));

15. Create the AccountsMaster table with the following Columns. Use auto generate to generate Account number

Customerld int(5)

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

AccountType Char(3)

LedgerBalance int(10,2) Not Null

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

Query:

alter table accountsMaster add constraint cust\_acc FOREIGN KEY (customerid) references customermaster(customerid);

17. 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

Query:

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);

18. Modify the AccountMaster table with the Check constraint to ensure AccountType should be either NRI or IND.

Query:

alter table accountsmaster modify(accounttype varchar2(3) CHECK(accounttype IN('NRI','IND')));

19. Modify the AccountsMaster table keeping a Check constraint with the name Balance\_Check for the Minimum Balance which should be greater than 5000.

Query:

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

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

Query:

alter table accountsmaster drop constraint cust\_acc;

alter table accountsmaster add constraint cust\_acc FOREIGN KEY(customerid) references customermaster(customerid) on delete cascade;

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

Query:

create table accountdetails as select \* from accountsmaster;

22. 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.

Query:

create view acc\_view (customercode, accountnumber, type, balance) as select \* from accountsmaster;

23. 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.

Hint: Use the With Check Option constraint.

Query:

create view vaccs\_dtls (customer, accounttype, balance) as select customerid, accounttype, ledgerbalance from accountsmaster where accounttype='IND' and ledgerbalance>10000;

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

Query:

25. Insert three sample rows by using the above auto generate in Department\_Masters table.

Query:

26. Get information on the index No\_Name from the Data Dictionary.

Query:

create no\_name on emp(empno);

27. Create synonym synEmp for the EMP table.

Query:

create synonym synemp for emp;

28. Get Information on synonym synEmp from the Data Dictionary.

Query:

select \* from user\_synonyms where synonym\_name='SYNEMP';

29. Note: Perform this after creating the Employee Table mentioned in the next Lab assignment. Create Index on HireDate column and give the name as idx\_emp\_hiredate for this object.Data Manipulation Language

Query:

create index idx\_emp\_hiredate on emp(hiredate);

**5.1**: Data Manipulation Language

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

Query:

create table employee as select \* from emp where 1=3;

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

Query:

insert into employee(empno,ename,sal,deptno) select empno,ename,sal,deptno from emp;

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.

Query:

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

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

Query:

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.

Query:

update employee set deptno=(select deptno from employee where deptno=7788) where deptno=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

Query:

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (1000,Allen, Clerk,1001,12-jan-01, 3000, 2,10);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (1001,George, analyst, null, 08 Sep 92, 5000,0, 10);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (1002, Becker, Manager, 1000, 4 Nov 92, 2800,4, 20);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (1003, 'Bill', Clerk, 1002, 4 Nov 92,3000, 0, 20);

**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.

Query:

INSERT INTO customer VALUES(

&custid,

'&name',

'&addr',

'&confirmaddr',

'&gender',

&age,

&contact,

&sal);

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

Query:

SAVEPOINT SP1;

3. Insert the below row in the Customer table.

6003, John, #114 Chicago, #114 Chicago, M, 45, 439525, 19000.60

Query:

INSERT INTO CUSTOMER VALUES(6003,'John','#114 Chicago','#114 Chicago','M',45,439525,19000);

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.

Query:

ROLLBACK TO SP1;