**SQL**

**Lab 1**

**Question 1:**

SELECT Staff\_Name,Staff\_sal,Dept\_Code

FROM Staff\_Masters

WHERE Dept\_Code IN (20,30,40);

**Question 2:**

SELECT Staff\_Code AS emp\_no, Staff\_Name AS emp\_name, Design\_Code AS dcode,

Dept\_Code AS dept\_code, Staff\_dob AS emp\_dob, Hiredate AS doj, Mgr\_code AS m\_code,

Staff\_sal AS emp\_sal, Staff\_address AS emp\_addr

FROM Staff\_Masters;

**Question 3:**

SELECT student\_code, subject1, subject2, subject3, subject1+subject2+subject3 AS total\_marks

FROM Student\_Marks;

**Question 4:**

SELECT \*

FROM Staff\_Masters

WHERE Design\_Code IN

(SELECT Design\_Code

FROM Designation\_Masters

WHERE Design\_Name IN ('Professor','Lecturer'));

**Question 5**:

SELECT Staff\_Code, Staff\_Name, Dept\_Code

FROM Staff\_Masters

Where ((sysdate - hiredate)/365)>18;

**Question 6:**

SELECT s.Staff\_Name,d.Design\_Name

FROM Staff\_Masters s

NATURAL JOIN Designation\_Masters d

WHERE hiredate < '1-Jan-03';

**Question 7:**

SELECT s.Staff\_Name, d.Design\_Name, (s.Staff\_sal\*12)\*10

FROM Staff\_Masters s

NATURAL JOIN Designation\_Masters d

WHERE s.Dept\_Code IN (10,30);

**Question 8:**

SELECT s.Staff\_Name,round((sysdate-s.hiredate)/365,0) AS Experience

FROM Staff\_Masters s

NATURAL JOIN Designation\_Masters d

WHERE D.Design\_Name='Lecturer';

**Question 9:**

SELECT Staff\_Name || ', ' || Dept\_Code AS Student\_Info

FROM Staff\_Masters;

**Question 10:**

SELECT Staff\_Name,Staff\_sal

FROM Staff\_Masters

WHERE Staff\_sal BETWEEN 12000 AND 25000

ORDER BY Staff\_sal,Staff\_Name;

**Question 11:**

SELECT Staff\_Name

FROM Staff\_Masters

WHERE Mgr\_code = '';

**Question 12:**

SELECT Student\_Name,Dept\_Code,Student\_dob

FROM Student\_Masters

WHERE Student\_dob BETWEEN '01-Jan-1981' AND '31-Mar-1983'

ORDER BY Student\_dob;

**Question 13:**

SELECT SUM(staff\_sal),dept\_code   
FROM staff\_masters   
WHERE mgr\_code IS NULL   
HAVING sum(staff\_sal) > 20000   
GROUP BY dept\_code;

**Question 14:**

SELECT Book\_code

FROM Book\_transactions

WHERE Book\_actual\_return\_date > SYSDATE AND

Book\_expected\_return\_date = next\_day(SYSDATE -7, 'monday');

**Question 15**:

SELECT Student\_Name,Dept\_Code,NVL2(Dept\_Code, 'Have Departments', 'No Departments') AS

Departments

FROM Student\_Masters;

**Question 16**:

SELECT Staff\_Name, Staff\_sal, LPAD('x', Staff\_sal/1000,'x')

FROM Staff\_Masters;

**Lab 2**

**Question 1:**

SELECT Student\_Name, TO\_CHAR(Student\_Dob, 'MONTH, DD YYYY')

FROM Student\_Masters

WHERE TO\_CHAR(Student\_Dob, 'D') IN (6,0);

**Question 2:**

SELECT Staff\_Name, ROUND((SYSDATE - HIREDATE)/30) AS Months\_Worked

FROM Staff\_Masters

ORDER BY ROUND((SYSDATE - HIREDATE)/30);

**Question 3:**

SELECT \*

FROM Staff\_Masters

WHERE Staff\_Name LIKE ('A%') AND Staff\_Name LIKE ('%S');

**QUESTION 4**:

SELECT s.Staff\_Name , d.Design\_Name

FROM Staff\_Masters s

NATURAL JOIN Designation\_Masters d

WHERE (Staff\_Name LIKE ('\_N') OR Staff\_Name LIKE ('\_\_N'))

AND (Staff\_Name LIKE ('%N') OR Staff\_Name LIKE ('%S'));

**Question 5**

SELECT Staff\_Name,

LPAD(Staff\_Sal,15,'$') Staff\_Sal

FROM Staff\_Masters;

**Question 6**

SELECT Staff\_Name FROM staff\_Masters

WHERE Staff\_Name LIKE '%[\_]%';

**Question 7**

SELECT \* FROM staff\_masters

WHERE TO\_CHAR(hiredate,'mm')=12;

**Question 8**

SELECT staff\_name, staff\_sal,

case when staff\_sal = 50000 then 'A'

when staff\_sal between 25000 and 49999 then 'B'

when staff\_sal between 10000 and 24999 then 'C'

else 'D'

end as grade

From staff\_masters;

**Question 9**

SELECT staff\_name, hiredate, TO\_CHAR (hiredate, 'DAY') AS day

FROM staff\_masters

ORDER BY day;

**Question 10**

SELECT LPAD ('', length(staff\_name),'')

FROM staff\_masters;

**Question 11**

SELECT SUBSTR(Staff\_Name,1,1)|| LPAD(SUBSTR(Staff\_Name,length(staff\_name),1),

length(Staff\_Name)-1,’\*’) AS Names

FROM staff\_Masters;

**Question 12**

SELECT \*

FROM staff\_masters e

WHERE TO\_NUMBER(TO\_CHAR(e.hiredate, 'DD')) = 15;

**Question 13**

SELECT Staff\_Name,Hiredate,TO\_CHAR(Hiredate,'DAY')AS DAY

FROM staff\_Masters

ORDER BY Hiredate,DAY ASC;

**Question 14**

SELECT INSTR('Mississippi','i',1,3)

FROM dual;

**Question 15**

SELECT TO\_CHAR (NEXT\_DAY (

LAST\_DAY (SYSDATE) - INTERVAL '7' DAY,

'FRIDAY'), ‘DDSPTH MONTH, YYYY’) AS PAY\_DAY

FROM DUAL;

**Question 16**

SELECT

MAX (STAFF\_SAL) AS Maximum,

MIN (STAFF\_SAL) AS Minimum,

SUM (STAFF\_SAL) AS Total,

ROUND (AVG (STAFF\_SAL), 0) AS Average

FROM staff\_masters;

**Question 17**

SELECT staff\_masters.dept\_code, department\_masters.dept\_name,

MAX(staff\_sal) AS Maximum,

MIN(staff\_sal) As Minimum,

SUM(staff\_sal) AS Total,

ROUND(AVG(staff\_sal),0) AS Average

FROM staff\_masters ,department\_masters

WHERE Department\_masters.dept\_code = staff\_masters.dept\_code

GROUP by staff\_masters.dept\_code,department\_masters.dept\_name;

**Question 18**

SELECT DISTINCT (E1.dept\_code), E1.dept\_name,count(E2.staff\_code) as EMPLOYEE\_COUNT

FROM department\_masters E1,Staff\_Masters E2

WHERE E1.dept\_code=E2.dept\_code

GROUP BY E1.dept\_name,E1.dept\_code;

**Question 19**

SELECT COUNT(DISTINCT MGR\_CODE) Total\_Number\_of\_Managers

FROM STAFF\_MASTERS;

**Question 20**

SELECT e1.MGR\_CODE,e2.STAFF\_NAME,min(e1.STAFF\_SAL)

FROM STAFF\_MASTERS e1 , STAFF\_MASTERS e2

WHERE e1.MGR\_CODE = e2.STAFF\_CODE

GROUP BY e1.MGR\_CODE, e2.STAFF\_NAME

HAVING MIN (e1.STAFF\_SAL) > 10000

ORDER BY MIN (e1.STAFF\_SAL) DESC;

**Lab 3**

**Question 1**

SELECT sm.Staff\_Name,sm.Dept\_code,dm.Dept\_Name,sm.Staff\_sal

FROM Staff\_Masters sm,Department\_Masters dm

WHERE sm.Dept\_code=dm.Dept\_code

AND sm.Staff\_sal>20000;

**Question 2**

SELECT a.Staff\_Name, a.Dept\_Code, b.Dept\_name

FROM Staff\_Masters a, Department\_Masters b

WHERE b.Dept\_Code = a.Dept\_Code

AND a.Dept\_Code != 20

AND a.Staff\_Name LIKE '%A%';

**Question 3**

SELECT a.Staff\_Code AS Staff#,a.Staff\_Name AS Staff,Department\_Masters.Dept\_Name,b.Staff\_Code Mgr#,b.Staff\_Name Manager

FROM Staff\_Masters a JOIN Staff\_Masters b ON a.Mgr\_code=b.Staff\_Code

JOIN Department\_Masters ON Department\_Masters.Dept\_Code=a.Dept\_Code;

**Question 4**

SELECT sm.Student\_Code, sm.Student\_name, d.Dept\_name, s.Subject1, s.Subject2, s.Subject3

FROM Student\_Masters sm,

Student\_Marks s,

Department\_Masters d

WHERE sm.Student\_Code=s.Student\_Code

AND sm.Dept\_Code=d.Dept\_code

AND s.Subject1>60

AND s.Subject2>60

AND s.Subject3>60

AND d.Dept\_Code IN (10, 20);

**Question 5**

SELECT s.student\_code,

s.student\_name,

b.book\_code,b.book\_name

from book\_transactions t

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

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

where book\_expected\_return\_date=sysdate;

**Question 6**

SELECT

bt.book\_issue\_date,

bt.book\_code,

bt.staff\_code,

bm.book\_name,

sm.staff\_name,

ds.design\_name,

dp.dept\_name

FROM book\_transactions bt, book\_masters bm,staff\_masters sm, designation\_masters ds, department\_masters dp

WHERE

bm.book\_code=bt.book\_code and

sm.staff\_code=bt.staff\_code and

sm.design\_code=ds.design\_code and

sm.dept\_code=dp.dept\_code and

bt.book\_issue\_date > sysdate-30;

**Question 7**

SELECT s.staff\_code, s.staff\_name, des.design\_name, dep.dept\_name

FROM staff\_masters s, designation\_masters des, department\_masters dep

WHERE s.design\_code = des.design\_code

AND

des.design\_name != 'HOD';

**Question 8**

SELECT S.STUDENT\_CODE,S.STUDENT\_NAME,D.DEPT\_NAME,M.SUBJECT1+M.SUBJECT2 + M.SUBJECT3

AS TOTAL\_MARKS

FROM STUDENT\_MASTERS S,DEPARTMENT\_MASTERS D,STUDENT\_MARKS M

WHERE S.DEPT\_CODE = D.DEPT\_CODE

AND S.STUDENT\_CODE =M.STUDENT\_CODE

ORDER BY D.DEPT\_NAME,M.SUBJECT1+M.SUBJECT2 +M.SUBJECT3;

**Question 9**

SELECT S.staff\_code,   
S.staff\_name,desg.design\_name,dep.dept\_name,   
b.book\_code,b.book\_name,b.Book\_pub\_author ,((t.book\_actual\_return\_date - t.book\_expected\_return\_date)\*5) as fine   
FROM staff\_masters s   
join department\_masters dep on   
dep.dept\_code = s.dept\_code   
join designation\_masters desg on   
desg.design\_code = s.design\_code   
join book\_transactions t on   
t.staff\_code = s.staff\_code   
join book\_masters b on   
t.book\_code = b.book\_code   
WHERE ((t.book\_expected\_return\_date - t.book\_actual\_return\_date) < 0)

**Question 10**

SELECT STAFF\_CODE,STAFF\_NAME

FROM STAFF\_MASTERS

WHERE STAFF\_SAL<(SELECT AVG(STAFF\_SAL)

FROM STAFF\_MASTERS);

**Question 11**

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

SELECT STAFF\_NAME,STAFF\_CODE

FROM STAFF\_MASTERS

WHERE STAFF\_CODE NOT IN

(SELECT MGR\_CODE FROM STAFF\_MASTERS);

**Question 12**

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(1)>1 )

order by Book\_pub\_author, Book\_name

**Question 13**

SELECT Staff\_Name,Dept\_Name, Staff\_Code   
FROM Book\_Transactions NATURAL JOIN Staff\_Masters NATURAL JOIN Department\_Masters   
GROUP BY Staff\_Code, Staff\_Name, Dept\_Name   
HAVING COUNT(\*) > 1;

**Question 14**

SELECT S.STUDENT\_CODE,S.STUDENT\_NAME,D.DEPT\_NAME,M.SUBJECT1,M.SUBJECT2,M.SUBJECT3 ,M.SUBJECT1+M.SUBJECT2+M.SUBJECT3

AS TOTAL

FROM STUDENT\_MASTERS S

JOIN DEPARTMENT\_MASTERS D

ON S.DEPT\_CODE=D.DEPT\_CODE

JOIN STUDENT\_MARKS M

ON S.STUDENT\_CODE=M.STUDENT\_CODE

WHERE ROWNUM<=10

ORDER BY SUBJECT1+SUBJECT2+SUBJECT3 DESC;

**Question 15**

SELECT S1.Staff\_Code, S1.Staff\_sal, d.Dept\_Name   
FROM staff\_Masters S1 JOIN Department\_Masters d   
ON (S1.Dept\_Code = d.Dept\_Code)   
WHERE Staff\_sal < (SELECT AVG(Staff\_sal) FROM staff\_Masters S2 GROUP BY Dept\_Code HAVING S1.Dept\_Code = S2.Dept\_Code);

**Question 16**

SELECT STAFF\_NAME, COALESCE (DEPT\_NAME,'')

AS DEPARTMENT

FROM STAFF\_MASTERS S

JOIN DEPARTMENT\_MASTERS D

ON S.DEPT\_CODE=D.DEPT\_CODE

GROUP BY ROLLUP(DEPT\_NAME,STAFF\_NAME);

**Question 17**

SELECT s.Student\_Name, s.Student\_Code   
FROM Student\_Masters s   
JOIN Student\_Marks m1

ON s.Student\_Code = m1.Student\_Code   
JOIN Department\_Masters d ON

s.Dept\_Code = d.Dept\_Code   
WHERE

m1.subject1 = (SELECT MAX(subject1) FROM Student\_Marks WHERE d.Dept\_Name = 'Computer Science')

AND   
 m1.subject2 = (SELECT MAX(subject2) FROM Student\_Marks WHERE d.Dept\_Name = 'Computer Science') AND

m1.subject3 = (SELECT MAX(subject3) FROM Student\_Marks WHERE d.Dept\_Name = 'Computer Science');

**Question 18**

select student\_code, student\_name, dept\_name   
from student\_masters natural join department\_masters   
where dept\_code= (SELECT   
     dept\_code FROM   
  student\_masters   
GROUP BY   
    DEPT\_CODE   
HAVING   
    COUNT(\*) >= ALL(SELECT COUNT(\*)   
FROM student\_masters   
GROUP BY  dept\_code))

**Question 19**

SELECT s.staff\_code,

s.staff\_name,

dep.dept\_name,

desg.design\_name

FROM (select max(Hiredate) AS recent from staff\_Masters) a

JOIN staff\_Masters s

ON s.Hiredate= a.recent

JOIN Department\_Masters dep

ON dep.dept\_code = s.dept\_code

JOIN Designation\_Masters desg

ON desg.design\_code = s.design\_code;

**Question 20**

SELECT t1.staff\_name AS MANAGER

count(\*) AS Team\_strength

FROM staff\_masters t1

JOIN staff\_masters t2

ON t1.staff\_code = t2.mgr\_code

GROUP BY t1.staff\_name;

**3.2: Set Operators**

**1**

SELECT \*

FROM previous\_products

UNION

SELECT \*

FROM current\_products;

**2.**

SELECT \*

FROM previous\_products

UNION ALL

SELECT \*

FROM current\_products;

**3.**

SELECT \*

FROM previous\_products

INTERSECT

SELECT \*

FROM current\_products;

**4.**

SELECT \*

FROM previous\_products

MINUS

SELECT \*

FROM current\_products;

**Lab 4 Database Objects**

**Question 1.**

CREATE TABLE customer (

customerid NUMBER(5),

customername NUMBER(10),

address1 VARCHAR2(30),

address2 VARCHAR2(30)

)

**Question 2.**

ALTER TABLE customer

MODIFY customername VARCHAR2(30) NOT NULL

/

**Question 3.**

ALTER TABLE customer

ADD (gender VARCHAR2(1),

age NUMBER(3),

phoneno NUMBER(10)

)

/

**Question 4.**

INSERT INTO customer

VALUES (1000, 'Allen', '#115 Chicago', '#115 Chicago', 'M', 25, 7878776)

/

INSERT INTO customer

VALUES (1001, 'George', '#116 France', '#116 France', 'M', 25, 434524)

/

INSERT INTO customer

VALUES (1002, 'Becker', '#114 New York', '#114 New York', 'M', 45, 431525)

/

**Question 5.**

ALTER TABLE customer

MODIFY customerid

CONSTRAINT custid\_prim PRIMARY KEY

/

**Question 6.**

INSERT INTO customer

VALUES (1002, 'John', '#114 Chicago', '#114 Chicago', 'M', 45, 439525)

/

INSERT INTO customer

\*

ERROR at line 1:

ORA-00001: unique constraint (LAB08TRG20.CUSTID\_PRIM) violated

**Question 7.**

ALTER TABLE customer

DISABLE CONSTRAINT custid\_prim

/

INSERT INTO customer

VALUES (1002, 'Becker', '#114 New York', '#114 New york' , 'M', 45, 431525)

/

INSERT INTO customer

VALUES (1003, 'Nanapatekar', '#115 India', '#115 India' , 'M', 45, 431525)

/

**Question 8.**

ALTER TABLE customer

ENABLE CONSTRAINT custid\_prim

/

ALTER TABLE customer

\*

ERROR at line 1:

ORA-02437: cannot validate (LAB08TRG20.CUSTID\_PRIM) - primary key violated

**Question 9.**

ALTER TABLE customer

DROP CONSTRAINT custid\_prim

/

INSERT INTO customer

VALUES (1002, 'Becker', '#114 New York', '#114 New York' , 'M', 45, 431525, 15000.50)

/

INSERT INTO customer

\*

ERROR at line 1:

ORA-00913: too many values

INSERT INTO customer

VALUES (1003, 'Nanapatekar', '#115 India', '#115 India' ,'M', 45, 431525, 20000.50)

/

INSERT INTO customer

\*

ERROR at line 1:

ORA-00913: too many values

**Question 10.**

TRUNCATE TABLE customer

/

**Question 11.**

ALTER TABLE customer

ADD e\_mail VARCHAR2(20)

/

**Question 12.**

ALTER TABLE customer

DROP COLUMN e\_mail

**Question 13.**

ALTER TABLE customer

ADD emailid VARCHAR2(40)

/

**Question 14.**

ALTER TABLE customer

SET UNUSED COLUMN emailid

/

**Question 15.**

ALTER TABLE customer

DROP UNUSED COLUMN

/

**Question 16.**

COMMENT ON TABLE customer

IS 'Customer Details'

/

**Question 17.**

SELECT comments

FROM user\_tab\_comments

WHERE table\_name = 'CUSTOMER'

/

**Question 18.**

COMMENT ON COLUMN customer.phoneno

IS 'Personal Contact no'

/

**Question 19.**

SELECT comments

FROM user\_col\_comments

WHERE table\_name = 'CUSTOMER'

AND column\_name = 'PHONENO'

/

**Question 20.**

CREATE TABLE suppliers

AS (

SELECT customerid AS suppid,

customername AS sname,

address1 AS addr1,

address2 AS addr2,

phoneno AS contactno

FROM customer

);

**Question 21**

DROP TABLE suppliers;

CREATE TABLE CustomerMaster(CustomerId NUMBER(5) CONSTRAINT CustId\_PK PRIMARY KEY,

CustomerName VARCHAR2(30) NOT NULL,

Addressl VARCHAR2(30) NOT NULL,

Address2 VARCHAR2(30),

Gender VARCHAR2(1),

Age NUMBER(3),

PhoneNo NUMBER(10));

**Question 22**

CREATE table AccountsMaster(   
Customerld NUMBER(5),   
AccountNumber NUMBER(10,2) CONSTRAINT Acc\_PK PRIMARY KEY,   
AccountType CHAR(3),   
LedgerBalance NUMBER(10,2) NOT NULL);

CREATE SEQUENCE sequence\_1   
start with 100   
increment by 1   
cycle;

INSERT INTO accountsmaster   
VALUES (453,sequence\_1.nextval,'SBI',60000);

INSERT INTO accountsmaster   
VALUES (376,sequence\_1.nextval,'IND',90000);

INSERT INTO accountsmaster   
VALUES (987,sequence\_1.nextval,'SBI',50000);

INSERT INTO accountsmaster   
VALUES (754,sequence\_1.nextval,'ICI',30000);

INSERT INTO accountsmaster   
VALUES (543,sequence\_1.nextval,'ICI',30000);

**Question 23**

ALTER TABLE accountmaster   
ADD FOREIGN KEy (customerid)   
REFERENCES customermaster (customerid);

**Question 24**

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);

**Question 25**

ALTER TABLE accountSmaster   
ADD CONSTRAINT acc\_check CHECK (ACCOUNTTYPE= 'NRI' OR ACCOUNTTYPE='IND');

**Question 26**

INSERT INTO accountsmaster   
VALUES (453,109,'SBI',60000);

INSERT INTO accountsmaster   
VALUES (376,101,'IND',90000);

INSERT INTO accountsmaster   
VALUES (987,541,'SBI',50000);

INSERT INTO accountsmaster   
VALUES (754,254,'ICI',30000);

INSERT INTO accountsmaster   
VALUES (543,105,'ICI',30000);

**Question 27**

ALTER TABLE accountmaster   
ADD CONSTRAINT Balance\_Check CHECK (ledgerbalance>5000);

**Question 28**  
  
ALTER TABLE accountmaster   
ADD FOREIGN KEy (customerid)   
REFERENCES customermaster (customerid)   
ON DELETE CASCADE;

**Question 29**

CREATE table AccountDetails AS(Select \* from AccountsMaster);

**Question 30**

ALTER TABLE AccountDetails RENAME TO BackUpTable;

**Question 31**.

CREATE VIEW Acc\_view

AS

SELECT CustomerId CustomerCode , CustomerName AccountHolderName, AccountNumber AccountNumber, AccountType Type, LedgerBalance Balance

FROM customermaster NATURAL JOIN accountsmaster;

**Question** **32.**

CREATE VIEW vAccs\_Dlts

AS

SELECT \* FROM Accountsmaster

WHERE (accounttype = 'IND'and ledgerbalance > 10000);

**Question 33.**

CREATE VIEW accsvw10

AS

SELECT \* FROM AccountsMaster

WITH READ ONLY CONSTRAINT accsvw\_RD ;

**Question 34**.

SELECT d.dept\_name   
FROM Department\_Masters d, (   
SELECT MAX(staff\_sal) AS salary,dept\_code   
FROM Staff\_Masters   
GROUP BY dept\_code   
ORDER BY salary desc   
) s WHERE d.dept\_code = s.dept\_code and rownum = 1 ;

**Question 35.**

SELECT d.dept\_name,s.staff\_sal,d.dept\_code

FROM Department\_Masters d,

(SELECT \* FROM (SELECT staff\_sal,dept\_code,RANK() OVER (PARTITION BY dept\_code ORDER BY staff\_sal DESC) AS myrank

FROM staff\_masters)

WHERE myrank <=2) s

WHERE d.dept\_code = s.dept\_code

**Question 36.**

 CREATE SEQUENCE Seq\_Dept   
 START WITH 40   
 INCREMENT BY 10   
 MINVALUE 40   
 MAXVALUE 200   
 NOCYCLE;

 INSERT INTO dept   
 (deptno) VALUES (Seq\_Dept.nextval);

**Question 37.**

 INSERT INTO dept   
 (deptno) VALUES (Seq\_Dept.nextval);

 INSERT INTO dept   
 (deptno) VALUES (Seq\_Dept.nextval);

 INSERT INTO dept   
 (deptno) VALUES (Seq\_Dept.nextval);

**Question 38.**

ALTER SEQUENCE Seq\_Dept   
 INCREMENT BY 5;

**Question 39.**

DROP SEQUENCE Seq\_Dept;

**Question 40.** CREATE INDEX No\_Name ON dept (deptno,dname);

**Question 41.**

CREATE INDEX no\_name on emp(empno);

select \* from emp;

**Question 42.**

CREATE PUBLIC SYNONYM  synEmp

   FOR Emp;

**Question 43.**

select \* from synemp;

**Lab 5**

**Question 1**

CREATE TABLE employee

AS(

SELECT \*

FROM emp

WHERE 3=4);

**Question 2**

INSERT INTO employee

(empno,ename,sal,deptno)

SELECT empno,ename,sal,deptno

FROM emp;

**Question 3**

UPDATE employee

SET job =

(SELECT job

FROM emp

WHERE empno=7788),

deptno =

(SELECT deptno

FROM emp

WHERE empno=7788)

WHERE empno=7698;

**Question 4**

DELETE \*

FROM department\_masters

WHERE dept\_name='sales'

**Question 5**

UPDATE employee

SET deptno =

(SELECT deptno

FROM emp

WHERE empno=7698)

WHERE empno=7788;

**Question 6**

INSERT INTO employee

VALUES (&empno,&ename,&job,&mgr,&hiredate,&sal,&comm,&deptno);

**Lab 6**

**Question 1**

INSERT INTO Customer VALUES(&customer\_id, &customer\_name, &customer\_address, &customer\_accno, &gender, &age, &mobile\_no, &price);

Enter value for customer\_id: 6000

Enter value for customer\_name: 'John'

Enter value for customer\_address: '#115 Chicago'

Enter value for customer\_accno: 123456787

Enter value for gender: 'M'

Enter value for age: 25

Enter value for mobile\_no: 7878776

Enter value for price: 10000

old 1: insert into customer values(&customer\_id, &customer\_name, &customer\_address, &customer\_accno, &gender, &age, &mobile\_no, &price)

new 1: insert into customer values(6000, 'John', '#115 Chicago', 123456787, 'M', 25, 7878776, 10000)

1 row created.

**Question 2**

savepoint SP1;

**Question 3**

INSERT INTO Customer VALUES(&customer\_id, &customer\_name, &customer\_address, &customer\_accno, &gender, &age, &mobile\_no, &price);

**Question 4**

rollback to SP1;

commit;