1.1

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

SQL>SELECT STAFF\_NAME,DESIGN\_CODE FROM STAFFMASTER WHERE (HIREDATE <'01-JAN-2003') AND STAFF\_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

SQL>SELECT STAFF\_CODE,STAFF\_NAME,DEPT\_CODE FROM STAFFMASTER WHERE (MONTHS\_BETWEEN(SYSDATE,HIREDATE))>=216 ORDER BY HIREDATE DESC;

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

SQL>SELECT \* FROM STAFFMASTER 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.

SQL>SELECT \* FROM BOOK\_MASTER WHERE BOOK\_PUB\_YEAR BETWEEN 2001 AND 2004 AND BOOK\_NAME LIKE '%"&"%';

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

SQL>SELECT STAFF\_NAME FROM STAFFMASTER WHERE STAFF\_NAME LIKE '%\_%';

2.1

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 ‘$’.

SQL>SELECT STAFF\_NAME,'$'||STAFF\_SAL AS STAFF\_SALARY FROM STAFFMASTER;

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.

SQL> SELECT STUDENT\_NAME,TO\_CHAR(STUDENTDOB,'MONTH DD YYYY') AS STUDENT\_DOB FROM STUDENTMASTER WHERE TO\_CHAR(STUDENTDOB,'DAY') LIKE ('%SATURDAY%') OR TO\_CHAR(STUDENTDOB,'DAY') LIKE ('%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

SQL> SELECT STAFF\_NAME,ROUND(MONTHS\_BETWEEN(SYSDATE,HIREDATE)) AS MONTHS\_WORKED FROM STAFFMASTER ORDER BY MONTHS\_WORKED DESC;

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

SQL>SELECT \* FROM STAFFMASTER WHERE TO\_CHAR(HIREDATE,'DD') BETWEEN 1 AND 15 AND TO\_CHAR(HIREDATE,'MONTH') LIKE '%DECEMBER%' ;

5. 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>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 CASE

FROM STAFFMASTER;

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’

SQL>SELECT STAFF\_NAME,TO\_CHAR(HIREDATE,'DD MONTH YYYY') AS HIRE\_DATE,TO\_CHAR(HIREDATE,'DAY')AS DAY FROM STAFFMASTER ORDER BY TO\_CHAR(HIREDATE,'DAY') DESC;

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

SQL> 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

SQL>SELECT TO\_CHAR(NEXT\_DAY(SYSDATE,'TUESDAY'),'DD MONTH ,YYYY') AS DAY FROM DUAL WHERE NEXT\_DAY(SYSDATE,'TUESDAY')<LAST\_DAY(SYSDATE) ;

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

SQL> SELECT STUDENT\_CODE,STUDENT\_NAME,DECODE(DEPT\_CODE,20,'ELECTRICALS',30,'ELECTRONICS','OTHERS') DEPARTMENT\_NAME FROM STUDENTMASTER;

3.1

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.

SQL> SELECT S.STAFF\_NAME,

D.DEPT\_CODE,

D.DEPT\_NAME,

S.STAFF\_SAL

FROM STAFFMASTER S,

DEPARTMENT\_MASTER D

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

AND 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.

SQL> SELECT S.STAFF\_CODE AS STAFF# ,

S.STAFF\_NAME AS STAFF,

D.DEPT\_NAME,

S.MGR\_CODE AS MGR#

FROM STAFFMASTER S,

DEPARTMENT\_MASTER D

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

SQL> SELECT S.STUDENT\_CODE,S.STUDENT\_NAME,B.BOOK\_CODE,BB.BOOK\_NAME FROM STUDENTMASTER S,BOOK\_TRANSACTIONS B, BOOK\_MASTER BB WHERE S.STUDENT\_CODE=B.STUDENT\_CODE AND TO\_CHAR(B.BOOK\_EXPECTED\_RETURN\_DATE,'DD MM YYYY') LIKE TO\_CHAR(SYSDATE,'DD MM YYYY');

SQL>SELECT S.STUDENT\_CODE,

S.STUDENT\_NAME,

B.BOOK\_CODE,

BB.BOOK\_NAME

FROM STUDENTMASTER S,

BOOK\_TRANSACTIONS B,

BOOK\_MASTER BB

WHERE S.STUDENT\_CODE=B.STUDENT\_CODE

AND TO\_CHAR(B.BOOK\_EXPECTED\_RETURN\_DATE,'DD MM YYYY') LIKE TO\_CHAR(SYSDATE,'DD MM YYYY');

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. HH

SQL>SELECT S.STAFF\_CODE,S.STAFF\_NAME,D.DEPT\_NAME,F.DESIGN\_NAME,G.BOOK\_NAME,H.BOOK\_ISSUE\_DATE FROM STAFFMASTER S,DEPARTMENT\_MASTER D,DESIGNATION-MASTER F,BOOK\_MASTER F,BOOK\_ISSUE\_DATE H WHERE MONTHS\_BETWEEN(TO\_CHAR(H.BOOK\_ISSUE\_DATE,'MM'),TO\_CHAR(SYSDATE,'MM'))<1;

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

6. 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 STAFFMASTER WHERE STAFF\_SAL<(SELECT AVG(STAFF\_SAL) FROM STAFFMASTER);

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

SQL>SELECT AUTHOR,BOOK\_NAME FROM BOOK\_MASTER GROUP BY AUTHORNAME HAVING COUNT(AUTHOR)>1;

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

SQL>SELECT S.Staff\_Code,D.Staff\_Name,D.DEPT\_NAME FROM STAFFMASTER S,BOOK\_TRANSACTIONS D GROUP BY S.STAFF\_NAME HAVING COUNT(D.STAFF\_NAME)>1;

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

SQL> SELECT S.STUDENT\_CODE,S.STUDENT\_NAME,D.DEPT\_NAME FROM STAFFMASTER S,DEPARTMENT\_MASTER D GROUP BY S.DEPT\_CODE HAVING MAX(S.DEPT\_CODE);

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

SQL>SELECT S.Staff\_Code,S.Staff\_Name,D.DEPT\_NAME,F.DESIGN\_NAME FROM STAFFMASTER S, DEPARTMENT\_MASTER D,DESIGNATION\_MASTER F WHERE MONTHS\_BETWEEN(TO\_CHAR(HIREDATE,'MM') ,TO\_CHAR(SYSDATE,'MM'))<3;

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

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

13. 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,DEPT\_NAME,COUNT(S.STAFF\_NAME) AS NUMBEROFPEOPLE FROM STAFFMASTER S,DEPARTMENT\_MASTER D GROUP BY DEPT\_CODE;

4.1

4.1: Database Objects

1. Create the Customer table with the following columns.

CustomerId Number(5)

Cust\_Name varchar2(20)

Address1 Varchar2(30)

Address2 Varchar2(30)

==>create table 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.

==>Alter table customer rename column cust\_name to customername;

==>Alter table customer modify customername varchar2(30) Not Null;

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

Gender Varchar2(1)

Age Number(3)

PhoneNo Number(10)

==>Alter table customer add Gender varchar2(1);

==>Alter table customer add Age Number(3);

==>Alter table customer add phoneNo(10);

b) Rename the Customer table to Cust\_Table

==>Rename customer to cust\_table;

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

==>insert into cust\_table(&Customerid,'&cust\_Name','&Address1','&Address2','&Gender',&Age,&phoneNo);

==>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

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

==>Alter table cust\_table add constraints 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

==>

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 cust\_table drop PRIMARY KEY custid\_prim;

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

==>Alter table cust\_table add constraints Custid\_prim PRIMARY KEY (customerid);

==>

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 cust\_table drop PRIMARY KEY custid\_prim;

==>Insert into cust\_table(1002, Becker, #114 New York, #114 New york , M, 45,431525, 15000.50);

==>Insert into cust\_table(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 cust\_table;

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

==>Alter table add e\_mail varchar2(30);

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

==>Alter table cust\_table DROP e\_mail;

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.

==>create table Suppliers as select(customerid as suppid,customername as sname,adddress1 as addr1,address2 as addr2,phoneno as contactno) from cust\_table;

14. 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)

==>Drop table Suppliers;

==>create table customermaster(customerid(10) primary key(custid\_pk),customername varchar2(30),Address1 varchar2(30),Address2 varchar2(30),Gender varchar2(1),Age number(3),phoneno number(10));

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

Customerid 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 Accoutnmaster(customerid number(5),Accountnumber number(10) primary key(acno),accounttype char(3),ledgerbalance number(10) Not Null);

==>Create sequence seq\_ano

MINVALUE 101

MAXVALUE 10000

START WITH 101

INCREMENT BY 1

CACHE 101;

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

==>Alter table Accountmaster ADD constraint ass\_fk 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

==>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, 4345240;

==>Insert into customermaster values(1002, Becker, #114 New York, #114 New York, M, 45, 4315250;

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

==>alter table Accountmaster add constraint ck\_ac check(accountype='NRI' or accountype='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.

==>alter table Accountmaster add constraint Balance\_check(ledger balance > 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.

==>Delete from Accountmaster,customertable where customerid = 1001

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

==>Create table accountdetails as select \* from Accountmaster;

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.

==>CREATE VIEW Acc\_view AS SELECT(Customerid,Customername,Accountnumber,AccountType,ledgerBalance)

from AccountMaster;

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.

==>CREATE VIEW vAccs\_Dtls AS SELECT Accounttype,ledgerbalance from Accountmaster where accounttype = 'IND' and ledgerbalance < 10000;

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

==>

25. Create a Sequence with the name Seq\_Dept on Deptno column of Department\_Masters table. It should start from 40 and stop at 200. Increment parameter for the sequence Seq\_Dept should be in step of 10.

==>CREATE sequence SEQ\_DEPT minvalue 40 start with 40

increment by 10 MAX VALUE 200 cache 40;

26. Insert three sample rows by using the above sequence in Department\_Masters table.

==>create table departmentmaster(deptno number(50),Dname varchar2(25),location varchar2(25));

==>insert into departmentmaster values(seq\_dept.NEXTVAL,'MARKETING','NEW DELHI');

==>insert into departmentmaster values(seq\_dept.NEXTVAL,'SALES','chennai');

==>insert into departmentmaster values(seq\_dept.NEXTVAL,'RESEARCH','BOSTON');

27. Drop the Seq\_Dept sequence.

==>DROP sequence seq\_dept;

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

==>CREATE INDEX no\_name on emp(empno);

==>select \* from emp;

29. Create synonym synEmp for the EMP table.

==>create SYNONYM synemp for emp;

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

==>select \* from synemp;

31. 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.

==>CREATE INDEX IDX\_EMP\_HIREDATE on emp(HIREDATE);

32. Create a Sequence with the name Seq\_Emp on Empno column of Employee table. It should start from 1001. Try to set Minimum value for this sequence which is less than / greater than 1001, use the sequence to generate Empno while inserting records in Employee table and check the values generated.

==> ==>CREATE sequence SEQ\_EMP minvalue 1001 start with 1001

increment by 1 cache 1001;

5.1

5.1: Data Manipulation Language

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

SQL>Create table employee as select \* from emp where 1=3;

SQL>desc employee;

Name Null? Type

EMPNO NOT NULL NUMBER(4)

ENAME VARCHAR2(10)

JOB VARCHAR2(50)

MGR NUMBER(4)

HIREDATE DATE

SAL NUMBER(7,2)

COMM NUMBER(7,2)

DEPTNO NUMBER(2)

SQL>select \* from employee

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

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 2450 10

7788 SCOTT 3000 20

7839 KING 5000 10

7844 TURNER 1500 30

7876 ADAMS 1100 20

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 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’.

SQL> delete from employee where departmentname like '%sales%';

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

SQL>update table 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.

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

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

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

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

6.1

6.1: Transaction Control Language Statements

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

SQL>insert into customermaster (customerid,'customername','address1','address2','gender',age,'phoneno) values ( 6000, John, #115 Chicago, #115 Chicago, M, 25, 7878776, 10000 );

SQL>insert into customermaster (customerid,'customername','address1','address2','gender',age,'phoneno) values ( 6001, Jack, #116 France, #116 France, M, 25, 434524, 20000 );

SQL>insert into customermaster (customerid,'customername','address1','address2','gender',age,'phoneno) values ( 6002, James, #114 New York, #114 New York, M, 45, 431525, 15000.50);

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

SQL>insert into customermaster (customerid,'customername','address1','address2','gender',age,'phoneno) values ( 6000, John, #115 Chicago, #115 Chicago, M, 25, 7878776, 10000 );

SQL>insert into customermaster (customerid,'customername','address1','address2','gender',age,'phoneno) values ( 6001, Jack, #116 France, #116 France, M, 25, 434524, 20000 );

SQL>insert into customermaster (customerid,'customername','address1','address2','gender',age,'phoneno) values ( 6002, James, #114 New York, #114 New York, M, 45, 431525, 15000.50);

SQL> savepoint p1;

3. Insert the below row in the Customer table.

SQL>insert into customermaster (customerid,'customername','address1','address2','gender',age,'phoneno) values ( 6003, John, #114 Chicago, #114 Chicago, M, 45, 439525, 19000.60);

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 p1;