1.1. SELECT Staff_Name AS STAFF NAME,Design_code AS DESIGNATION CODE from Staff_Master
WHERE HireDate < 01-JAN-2003 AND Staff_sal BETWEEN 12000 AND 25000;

1.2. SELECT Staff_code,Staff_Name,Dept_code from Staff_Master WHERE
ROUND((MONTHS_BETWEEN(SYSTEM,HireDate)/12) >= 18 ORDER BY HireDate;

1.3. SELECT Staff_code,Staff_Name,Staff_dob,Staf_Address,Staff_sal FROM Staff-Master WHERE
Mgr_code IS NULL;

1.4. SELECT Book_Code,Book_Name,Book_pub_year,Book_pub_author
WHERE Book_pub_year BETWEEN 2001 AND 2004 OR Book_Name LIKE '%&%';

1.5. SELECT Staff_Name FROM Staff_Master WHERE Staff_Name LIKE '%$_%' ESCAPE= '$';

2.1

1. SELECT Staff_Name,$||LPAD(Staff_sal,15,'') FROM Staff_Master;

2. SELECT Student_Name,TO_CHAR(Student_dob,'MONTH,DD YYY') from Student_Master
WHERE (TO_CHAR(Student_dob,'day') IN ('saturday','sunday');

3. SELECT Staff_name, ROUND(MONTHS_BETWEEN(SYSDATE,HireDate))AS Months Worked FROM
Staff_Master
ORDER BY HireDate;

4. SELECT Staff_Name from Staff_Master WHERE EXTRACT(DATE FROM HireDate) < 16 AND
EXTRACT(MONTH FROM HireDate) = 12 ;

5. SELECT Staff_Name,HireDate,Staff_Sal
CASE
WHEN Stff_Sal >=50000 THEN 'A',
WHEN Staff_Sal >= 25000 AND Staff_Sal <5000 THEN 'B',
WHEN Staff_Sal >= 10000 AND Staff_Sal <25000 THEN 'C',
WHEN Staff_Sal 10000 THEN 'D'
END AS GRADE
FROM Staff_master;
6. SELECT Staff_Name,HireDate,TO_CHAR(HireDate,'DAY') AS DAY FROM Staff_Master
ORDER BY next_day(HireDate,'Monday')-HireDate;

7. SELECT INSTR('mississippi','i',1,3) from Dual;

8. SELECT TO_CHAR(NEXT_DAY(
LAST_DAY(SYSDATE) - INTERVAL '7' DAY,

```
        'FRIDAY'
      ),'ddthsp "of"  Month,YYYY') AS PAY_DATE FROM DUAL;


          // SELECT TO_CHAR(NEXT_DAY(SYSDATE,'FRIDAY'),'DD MONTH ,YYYY')
          AS DAY FROM DUAL WHERE NEXT_DAY(SYSDATE,'FRIDAY')<LAST_DAY(SYSDATE) ;
//

9. SELECT  Student_Code,Student-name,Dept_Code
   DECODE(Dept_Code,10,'Electricals',20,'Electronics','Others')
  FROM Student_Master;
```

Lab 2.2

```
1. SELECT ROUND(max(Staff_sal)) AS Maximum,ROUND(Min(Staff_Sal)) AS
Minimum,ROUND(Avg(Staff_sal)) AS Average
     FROM Staff_master;

2.  SELECT Deptno,COUNT(deptno) AS Total_Number_of_Managers
      FROM EMP WHERE JOB IN('MANAGER') GROUP BY Deptno,job;

3.  SELECT Deptno,sum(sal) FROM EMP
   WHERE JOB NOT IN('MANAGER')
       GROUP BY Deptno --HAVING sum(sal) > 20000-- ORDER BY DeptNo;
```

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.

==> create view accsvw10 AS SELECT * FROM Employee with READ ONLY;
// ==> REVOKE INSERT,UPDATE,DELETE ON accsvw10 from user1;
// ==> insert into accsvw10(eid,ename,sal,job)
values('SQL*PLUS','user1','INSERT','DISABLED';)

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;