-----------------------------Assignment-2------------------------------------------

1. Write an SQL query to fetch “FIRST\_NAME” from Worker table

using the alias name as <WORKER\_NAME>.

Ans. select first\_name worker\_name

-> from worker;

2. Write an SQL query to fetch “FIRST\_NAME” from Worker table in

upper case.

Ans. select upper(first\_name)

-> from worker;

3. Write an SQL query to fetch unique values of DEPARTMENT from

Worker table.

Ans. select distinct department

-> from worker;

4. Write an SQL query to print the first three characters

of FIRST\_NAME from Worker table.

Ans. select substring(first\_name, 1,3)

-> from worker;

5. Write an SQL query to find the position of the alphabet (‘a’) in the

first name column ‘Amitabh’ from Worker table.

Ans. select position('a' in first\_name)

-> from worker

-> where first\_name='Amitabh';

6. Write an SQL query to print the FIRST\_NAME , departmentname

from Worker table separated by white space.

Ans. select concat(first\_name," ", department)

-> from worker;

7. Write an SQL query to print the DEPARTMENT from Worker table

after removing white spaces from the left side.

Ans. select ltrim(department)

-> from worker;

8. Write an SQL query that fetches the unique values of DEPARTMENT

from Worker table and prints its length.

Ans. select distinct(department),length(department)

-> from worker;

9. Write an SQL query to print the FIRST\_NAME from Worker table

after replacing ‘a’ with ‘A’.

Ans. select replace(first\_name, 'a', 'A') new

-> from worker;

10. Write an SQL query to print the FIRST\_NAME and LAST\_NAME

from Worker table into a single column COMPLETE\_NAME. A space

char should separate them.

Ans. select concat(first\_name," ",last\_name) complete\_name

-> from worker;

11. Write an SQL query to print all Worker details from the Worker

table order by FIRST\_NAME Ascending.

Ans. select \*

-> from worker

-> order by first\_name ;

12. Write an SQL query to print all Worker details from the Worker

table order by FIRST\_NAME Ascending and DEPARTMENT

Descending.

Ans. select \*

-> from worker

-> order by first\_name asc, department desc;

13. Write an SQL query to print details for Workers with the first

name as “Vipul” and “Satish” from Worker table.

Ans. select \*

-> from worker

-> where first\_name in ('vipul','satish');

14. Write an SQL query to print details of workers excluding first

names, “Vipul” and “Satish” from Worker table.

Ans. select \*

-> from worker

-> where first\_name not in ('vipul','satish');

15. Write an SQL query to print details of Workers with

DEPARTMENT name as “Admin”.

Ans. select \*

-> from worker

-> where department ='admin';

16. Write an SQL query to print details of the Workers whose

FIRST\_NAME contains ‘a’.

Ans. select \*

-> from worker

-> where first\_name regexp '^.\*a.\*$';

17. Write an SQL query to print details of the Workers whose

FIRST\_NAME ends with ‘a’.

Ans. select \*

-> from worker

-> where first\_name regexp 'a$';

18. Write an SQL query to print details of the Workers whose

FIRST\_NAME ends with ‘h’ and contains six alphabets.

Ans. select \*

-> from worker

-> where first\_name regexp 'h$' and length(first\_name) = 6;

19. Write an SQL query to print details of the Workers whose

SALARY lies between 100000 and 500000.

Ans. select\*

-> from worker

-> where salary between 100000 and 500000;

20. Write an SQL query to print details of the Workers who have

joined in Feb’2014.

Ans. select \*

-> from worker

-> where month(joining\_date) = 2 and year(joining\_date) = 2014;

21. Write an SQL query to fetch the count of employees working in

the department ‘Admin’.

Ans. select department, count(department)

-> from worker

-> where department = 'admin';

22. Write an SQL query to fetch worker names with salaries >= 50000

and <= 100000.

Ans. select \*

-> from worker

-> where salary between 50000 and 100000;

23. Write an SQL query to fetch the no. of workers for each

department in the descending order.

Ans. select count(department),department

-> from worker

-> group by department

-> order by count(first\_name) desc;

24. Write an SQL query to print details of the Workers who are also Managers.

Ans. SELECT \*

FROM

emp

WHERE

empno IN (SELECT mgr FROM emp);

1. Write an SQL query to show only odd rows from a table.

Ans. select \*

from (select \*, row\_number() over() as rn from worker)subtable

where rn % 2 <> 0;

27. Write an SQL query to show only even rows from a table.

Ans select \*

-> from (select \*, row\_number() over() as rn from worker) subtable

-> where rn % 2 = 0;

28. Write an SQL query to clone a new table from another table.

Ans create table emp1

-> as

-> select \* from emp;

1. Write an SQL query to fetch intersecting records of two tables.

Ans.Select \*

From emp

Where sal > 2000

Intersect

Select \*

From emp

Where job = ‘manager’

1. Write an SQL query to show records from one table that anothertable does not have.

Ans. select \*

-> from worker w

-> where not exists (select worker\_ref\_id

-> from title t

-> where w.worker\_id = t.worker\_ref\_id);

1. Write an SQL query to show the current date and time.

Ans select date\_format(curdate(),'%Y-%M-%D %hh:%mm:%ss');

1. Write an SQL query to show the top n (say 10) records of a table.

Ans select \* from worker limit 10;

1. Write an SQL query to determine the nth (say n=5) highest salary from a table.

Ans. Select \*

From worker

Order by salary desc

Limit 4, 1;

1. Write an SQL query to determine the 5th highest salary without using TOP or limit method.

Ans. select \*

-> from (select \*, dense\_rank() over(order by salary) as rn from worker) as Ranked

-> where rn =5 ;

1. Write an SQL query to fetch the list of employees with the same salary.

Ans.select w.\*

-> from worker w

-> where exists (select 1

-> from worker w1

-> where w1.salary = w.salary

-> and w1.worker\_id != w.worker\_id);

1. Write an SQL query to show the second highest salary from a table.

Ans. select \*

-> from (select \*,

-> dense\_rank() over(order by salary desc) as rk from worker) as subtable

-> where rk =2;

1. Write an SQL query to show one row twice in results from a table.

Ans. Select \*

-> from worker

-> union all

-> select \*

-> from worker;

1. Write an SQL query to fetch intersecting records of two tables.

Ans. Select \*

-> from worker

-> where department ='HR'

-> intersect

-> select \*

-> from worker

-> where salary > 8000;

39. Write an SQL query to fetch the first 50% records from a table.

40. Write an SQL query to fetch the departments that have less than five people in it.

Ans. select department , count(\*)

-> from worker

-> group by department

-> having count(\*) < 5;

41. Write an SQL query to show all departments along with the

number of people in there.

Ans. select department , count(\*)

-> from worker

-> group by department

-> ;

1. Write an SQL query to show the last record from a table.

Ans. from (select \*, row\_number() over() as rn from worker) as subtable

-> where rn = (select count(\*) from worker);

43. Write an SQL query to fetch the first row of a table.

Ans. Select \* from wroker limit 1;

44. Write an SQL query to fetch the last five records from a table.

Ans. select \*

-> from (select \*, row\_number() over() as rn from worker) as subtable

-> where rn between (select count(\*)-4 from worker) and (select count(\*) from worker);

1. Write an SQL query to print the name of employees having the highest salary in each department.

Ans. select \*

-> from (select \* , dense\_rank() over (partition by department order by salary desc) as rk

-> from worker)as subtable

-> where rk = 1;

1. Write an SQL query to fetch three max salaries from a table.

Ans. select \*

-> from (select \* , dense\_rank() over (order by salary desc) as rk

-> from worker)as subtable

-> where rk <= 3;

1. Write an SQL query to fetch three min salaries from a table.

Ans. select \*

-> from (select \* , dense\_rank() over (order by salary asc) as rk

-> from worker)as subtable

-> where rk <= 3;

1. Write an SQL query to fetch nth max salaries from a table.

Ans. select \*

-> from (select \* , dense\_rank() over (order by salary desc) as rk

-> from worker)as subtable

-> where rk = n;

1. Write an SQL query to fetch departments along with the total salaries paid for each of them.

Ans. select department, sum(salary)

-> from worker

-> group by department;

50. Write an SQL query to fetch the names of workers who earn the highest salary.

from (select \* , dense\_rank() over (order by salary asc) as rk

-> from worker)as subtable

-> where rk = 1;