```
CREATE TABLE Worker (
        WORKER_ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
        FIRST_NAME CHAR(25),
        LAST_NAME CHAR(25),
        SALARY INT(15),
        JOINING_DATE DATETIME,
        DEPARTMENT CHAR(25)
);

INSERT INTO Worker
        (WORKER_ID, FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE, DEPARTMENT) VALUES
                (001, 'Monika', 'Arora', 100000, '14-02-20 09.00.00', 'HR'),
                (002, 'Niharika', 'Verma', 80000, '14-06-11 09.00.00', 'Admin'),
                (003, 'Vishal', 'Singhal', 300000, '14-02-20 09.00.00', 'HR'),
                (004, 'Amitabh', 'Singh', 500000, '14-02-20 09.00.00', 'Admin'),
                (005, 'Vivek', 'Bhati', 500000, '14-06-11 09.00.00', 'Admin'),
                (006, 'Vipul', 'Diwan', 200000, '14-06-11 09.00.00', 'Account'),
                (007, 'Satish', 'Kumar', 75000, '14-01-20 09.00.00', 'Account'),
                (008, 'Geetika', 'Chauhan', 90000, '14-04-11 09.00.00', 'Admin');



CREATE TABLE Bonus (
        WORKER_REF_ID INT,
        BONUS_AMOUNT INT(10),
        BONUS_DATE DATETIME,
        FOREIGN KEY (WORKER_REF_ID)
                REFERENCES Worker(WORKER_ID)
    ON DELETE CASCADE
);


INSERT INTO Bonus
        (WORKER_REF_ID, BONUS_AMOUNT, BONUS_DATE) VALUES
```

```
                (001, 5000, '16-02-20'),

                (002, 3000, '16-06-11'),

                (003, 4000, '16-02-20'),

                (001, 4500, '16-02-20'),

                (002, 3500, '16-06-11');
```

```
CREATE TABLE Title (

        WORKER_REF_ID INT,

        WORKER_TITLE CHAR(25),

        AFFECTED_FROM DATETIME,

        FOREIGN KEY (WORKER_REF_ID)

                REFERENCES Worker(WORKER_ID)

    ON DELETE CASCADE

);
```

```
INSERT INTO Title

        (WORKER_REF_ID, WORKER_TITLE, AFFECTED_FROM) VALUES

 (001, 'Manager', '2016-02-20 00:00:00'),

 (002, 'Executive', '2016-06-11 00:00:00'),

 (008, 'Executive', '2016-06-11 00:00:00'),

 (005, 'Manager', '2016-06-11 00:00:00'),

 (004, 'Asst. Manager', '2016-06-11 00:00:00'),

 (007, 'Executive', '2016-06-11 00:00:00'),

 (006, 'Lead', '2016-06-11 00:00:00'),

 (003, 'Lead', '2016-06-11 00:00:00');
```

Q-1. Write an SQL query to fetch “FIRST_NAME” from Worker table using the alias name as <WORKER_NAME>.

->Select FIRST_NAME AS WORKER_NAME from Worker;

Q-2. Write an SQL query to fetch “FIRST_NAME” from Worker table in upper case.

-> Select upper(FIRST_NAME) from Worker;

Q-3. Write an SQL query to fetch unique values of DEPARTMENT from Worker table.

Ans.

The required query is:

Select distinct DEPARTMENT from Worker;

Q-4. Write an SQL query to print the first three characters of FIRST_NAME from Worker table.

Ans.

The required query is:

Select substring(FIRST_NAME,1,3) from Worker;

Q-5. Write an SQL query to find the position of the alphabet (‘a’) in the first name column ‘Amitabh’ from Worker table.

Ans.

The required query is:

SELECT INSTR(FIRST_NAME , BINARY'a') FROM worker

WHERE FIRST_NAME='Amitabh';

SELECT POSITION(BINARY'a' IN FIRST_NAME) FROM worker

WHERE FIRST_NAME='Amitabh';

Q-6. Write an SQL query to print the FIRST_NAME from Worker table after removing white spaces from the right side.

Ans.

The required query is:

```
SELECT RTRIM(FIRST_NAME) FROM worker;
```

Q-7. Write an SQL query to print the DEPARTMENT from Worker table after removing white spaces from the left side.

Ans.

The required query is:

```
Select LTRIM(DEPARTMENT) from Worker;
```

Q-8. Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and prints its length.

Ans.

The required query is:

```
Select distinct length(DEPARTMENT) from Worker;
```

Q-9. Write an SQL query to print the FIRST_NAME from Worker table after replacing ‘a’ with ‘A’.

Ans.

The required query is:

```
Select REPLACE(FIRST_NAME,'a','A') from Worker;
```

Q-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.

The required query is:

```
Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from Worker;
```

Q-11. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending.

Ans.

The required query is:

```
Select * from Worker order by FIRST_NAME asc;
```

Q-12. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending and DEPARTMENT Descending.

Ans.

The required query is:

```
Select * from Worker order by FIRST_NAME asc,DEPARTMENT desc;
```

Q-13. Write an SQL query to print details for Workers with the first name as “Vipul” and “Satish” from Worker table.

Ans.

The required query is:

```
SELECT * FROM worker

WHERE FIRST_NAME='Vipul' OR FIRST_NAME='Satish';


Select * from Worker where FIRST_NAME in ('Vipul','Satish');
```

Q-14. Write an SQL query to print details of workers excluding first names, “Vipul” and “Satish” from Worker table.

Ans.

The required query is:

```
SELECT * FROM worker

WHERE FIRST_NAME NOT IN('Vipul','Satish');
```

Q-15. Write an SQL query to print details of Workers with DEPARTMENT name as “Admin”.

Ans.

The required query is:

```
SELECT * FROM worker
WHERE DEPARTMENT  LIKE 'Admin%';
```

department='admin'

```
SELECT * FROM worker
WHERE DEPARTMENT  IN('Admin');
```

Q-16. Write an SQL query to print details of the Workers whose FIRST_NAME contains ‘a’.

Ans.

The required query is:

```
Select * from Worker where FIRST_NAME like '%a%';
```

Q-17. Write an SQL query to print details of the Workers whose FIRST_NAME ends with ‘a’.

Ans.

The required query is:

```
Select * from Worker where FIRST_NAME like '%a';
```

Q-18. Write an SQL query to print details of the Workers whose FIRST_NAME ends with ‘h’ and contains six alphabets.

Ans.

The required query is:

```
Select * from Worker where FIRST_NAME like '_____h';
```

```
SELECT * FROM `worker` where FIRST_NAME LIKE '%h' and length(FIRST_NAME)=6;
```

Q-19. Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

Ans.

The required query is:

```
Select * from Worker where SALARY between 100000 and 500000;
```

Q-20. Write an SQL query to print details of the Workers who have joined in Feb’2014.

Ans.

The required query is:

```
Select * from Worker where year(JOINING_DATE) = 2014 and month(JOINING_DATE) = 2;
```

Q-21. Write an SQL query to fetch the count of employees working in the department ‘Admin’.

Ans.

The required query is:

```
SELECT COUNT(*) FROM worker WHERE DEPARTMENT = 'Admin';
```

Q-22. Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.

Ans.

```
SELECT * FROM worker
WHERE SALARY >=50000 AND SALARY <=100000;
```

The required query is:

```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) As Worker_Name, Salary
```

```
FROM worker
WHERE WORKER_ID IN
(SELECT WORKER_ID FROM worker
WHERE Salary BETWEEN 50000 AND 100000);
```

Q-23. Write an SQL query to fetch the no. of workers for each department in the descending order.

Ans.

The required query is:

```
SELECT DEPARTMENT,COUNT
(DEPARTMENT) AS number_of_worker FROM
worker
GROUP BY DEPARTMENT
ORDER by number_of_worker DESC;
```

```
SELECT DEPARTMENT, count(WORKER_ID) No_Of_Workers
FROM worker
GROUP BY DEPARTMENT
ORDER BY No_Of_Workers DESC;
```

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

Ans.

The required query is:

```
SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE
FROM Worker W
INNER JOIN Title T
ON W.WORKER_ID = T.WORKER_REF_ID
AND T.WORKER_TITLE in ('Manager');
```

Q-25. Write an SQL query to fetch duplicate records having matching data in some fields of a table.

Ans.

The required query is:

```
SELECT WORKER_TITLE, AFFECTED_FROM, COUNT(*)

FROM Title

GROUP BY WORKER_TITLE, AFFECTED_FROM

HAVING COUNT(*) > 1;
```

Q-26. Write an SQL query to show only odd rows from a table.

Ans.

The required query is:

```
SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) <> 0;
```

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

Ans.

The required query is:

```
SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) = 0;
```

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

Ans.

The general query to clone a table with data is:

```
create table worling_clone like worker;

insert into working_clone select * from worker;

create table working_clone select * from worker;
```

Q-29. Write an SQL query to fetch intersecting records of two tables.

Ans.

The required query is:

```
(SELECT * FROM Worker)

INTERSECT

(SELECT * FROM WorkerClone);
```

Q-30. Write an SQL query to show records from one table that another table does not have.

Ans.

The required query is:

```
SELECT * FROM Worker

MINUS

SELECT * FROM Title;
```

```
(SELECT * FROM worker) EXCEPT (SELECT * FROM worker_clone);
```

Q-31. Write an SQL query to show the current date and time.

Ans.

Following MySQL query returns the current date:

```
SELECT CURDATE();
```

Following MySQL query returns the current date and time:

```
SELECT NOW();
```

Following SQL Server query returns the current date and time:

```
SELECT getdate();
```

Following Oracle query returns the current date and time:

```
SELECT SYSDATE FROM DUAL;
```

Q-32. Write an SQL query to show the top n (say 10) records of a table.

Ans.

Following MySQL query will return the top n records using the LIMIT method:

```
SELECT * FROM Worker ORDER BY Salary DESC LIMIT 10;
```

Following SQL Server query will return the top n records using the TOP command:

```
SELECT TOP 10 * FROM Worker ORDER BY Salary DESC;
```

Following Oracle query will return the top n records with the help of ROWNUM:

```
SELECT * FROM (SELECT * FROM Worker ORDER BY Salary DESC)
WHERE ROWNUM <= 10;
```

Q-33. Write an SQL query to determine the nth (say n=5) highest salary from a table.

Ans.

The following MySQL query returns the nth highest salary:

```
SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT n-1,1;
```

The following SQL Server query returns the nth highest salary:

```
SELECT TOP 1 Salary
FROM (
 SELECT DISTINCT TOP n Salary
 FROM Worker
 ORDER BY Salary DESC
 )
ORDER BY Salary ASC;
```

Q-34. Write an SQL query to determine the 5th highest salary without using TOP or limit method.

Ans.

The following query is using the correlated subquery to return the 5th highest salary:

```
SELECT Salary
FROM Worker W1
WHERE 4 = (
 SELECT COUNT( DISTINCT ( W2.Salary ) )
 FROM Worker W2
 WHERE W2.Salary >= W1.Salary
 );
```

Use the following generic method to find nth highest salary without using TOP or limit.

```
SELECT Salary
FROM Worker W1
WHERE n-1 = (
 SELECT COUNT( DISTINCT ( W2.Salary ) )
 FROM Worker W2
 WHERE W2.Salary >= W1.Salary
 );
```

Q-35. Write an SQL query to fetch the list of employees with the same salary.

Ans.

The required query is:

```
Select distinct W.WORKER_ID, W.FIRST_NAME, W.Salary
from Worker W, Worker W1
where W.Salary = W1.Salary
```

```
SELECT w.* FROM worker as w,worker as w1
WHERE w.salary=w1.SALARY
and w.WORKER_ID != w1.WORKER_ID;
and W.WORKER_ID != W1.WORKER_ID;
```

Q-36. Write an SQL query to show the second highest salary from a table.

Ans.

third highest

```
SELECT * FROM worker
order by SALARY DESC LIMIT 2,1;
```

The required query is:

```
Select max(Salary) from Worker

where Salary not in (Select max(Salary) from Worker);
```

```
SELECT * FROM worker as w
WHERE (SELECT COUNT(DISTINCT SALARY)
    FROM worker AS w1
    WHERE w1.SALARY>=w.SALARY)=2;
```

Q-37. Write an SQL query to show one row twice in results from a table.

Ans.

The required query is:

```
select FIRST_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'

union all

select FIRST_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';
```

Q-38. Write an SQL query to fetch intersecting records of two tables.

Ans.

The required query is:

```
(SELECT * FROM Worker)

INTERSECT

(SELECT * FROM WorkerClone);
```

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

Ans.

The required query is:

```
SELECT *

FROM WORKER

WHERE WORKER_ID <= (SELECT count(WORKER_ID)/2 from Worker);
```

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

Ans.

The required query is:

```
SELECT DEPARTMENT, COUNT(WORKER_ID) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;
```

Q-41. Write an SQL query to show all departments along with the number of people in there.

Ans.

The following query returns the expected result:

```
SELECT DEPARTMENT, COUNT(DEPARTMENT) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT;
```

Q-42. Write an SQL query to show the last record from a table.

Ans.

The following query will return the last record from the Worker table:

```
Select * from Worker where WORKER_ID = (SELECT max(WORKER_ID) from Worker);
```

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

Ans.

The required query is:

```
Select * from Worker where WORKER_ID = (SELECT min(WORKER_ID) from Worker);
```

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

Ans.

The required query is:

```
SELECT * FROM Worker WHERE WORKER_ID <=5

UNION

SELECT * FROM (SELECT * FROM Worker W order by W.WORKER_ID DESC) AS W1 WHERE W1.WORKER_ID <=5;
```

```
(SELECT * FROM worker ORDER by WORKER_ID DESC
LIMIT 5) ORDER by WORKER_ID ASC;
```

```
SELECT * FROM worker
WHERE WORKER_ID>(SELECT COUNT(*) FROM worker)-5;
```

Q-45. Write an SQL query to print the name of employees having the highest salary in each department.

Ans.

The required query is:

```
SELECT t.DEPARTMENT,t.FIRST_NAME,t.SALARY

FROM (SELECT MAX(SALARY) as total_salary,DEPARTMENT FROM worker GROUP by DEPARTMENT) as TempNew

INNER JOIN worker t on t.DEPARTMENT=TempNew.DEPARTMENT

AND TempNew.total_salary=t.SALARY;
```

```
SELECT w.WORKER_ID,concat(w.FIRST_NAME,' ',w.LAST_NAME) as
Name,w.DEPARTMENT,w.SALARY
FROM(SELECT MAX(SALARY) as total_salary,DEPARTMENT FROM worker
GROUP by DEPARTMENT) AS new
INNER JOIN
worker AS w
on w.DEPARTMENT=new.DEPARTMENT
AND w.SALARY=new.total_salary;
```

Q-46. Write an SQL query to fetch three max salaries from a table.

Ans.

The required query is:

```
SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;
```

Q-47. Write an SQL query to fetch three min salaries from a table.

Ans.

The required query is:

```
SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary desc;
```

Q-48. Write an SQL query to fetch nth max salaries from a table.

Ans.

The required query is:

```
SELECT distinct Salary from worker a WHERE n >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;
```

Q-49. Write an SQL query to fetch departments along with the total salaries paid for each of them.

Ans.

The required query is:

```
 SELECT DEPARTMENT, sum(Salary) from worker group by DEPARTMENT;
```

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

Ans.

The required query is:

```
SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);
```