CREATE DATABASE test;

SHOW DATABASES;

USE test;

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)

);

f

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');

1. Write an SQL query to fetch unique values of DEPARTMENT from Worker table.

SELECT distinct department  
FROMf worker;

1. Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending and DEPARTMENT Descending

SELECT \*   
FROM worker   
ORDER BY first\_name , department DESC;

1. Write an SQL query to print details of the Workers whose FIRST\_NAME contains ‘a’

SELECT \*

FROM worker

WHERE first\_name

LIKE '%a%';

1. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘h’ and contains six alphabets

SELECT \*

FROM worker

WHERE first\_name LIKE '\_\_\_\_\_h';

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

SELECT \*

FROM worker

WHERE salary BETWEEN 100000 AND 500000;

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

SELECT \*

FROM Worker

WHERE JOINING\_DATE >= '2014-02-01' AND JOINING\_DATE < '2014-03-01';

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

SELECT COUNT(\*)

FROM worker

WHERE department='Admin';

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

SELECT first\_name, last\_name

FROM worker

WHERE salary>=50000 AND salary<=100000;

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

SELECT department, COUNT(\*) as workers\_count   
FROM worker

GROUP by department

ORDER BY worker\_count DESC;

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

SELECT \*

FROM worker, title

WHERE worker.worker\_id = title.worker\_ref\_id AND title.worker\_title = 'Manager';

1. Write an SQL query to determine the 2nd lowest salary without using TOP or limit method.

SELECT MIN(salary) as second\_lowest\_sal   
FROM worker

WHERE salary > (SELECT MIN(salary) FROM worker);

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

SELECT \*

FROM Worker W1

JOIN Worker W2 ON W1.SALARY = W2.SALARY AND W1.WORKER\_ID != W2.WORKER\_ID;

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

SELECT DISTINCT SALARY

FROM Worker

ORDER BY SALARY DESC

LIMIT 1 OFFSET 1;

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

SELECT \*

FROM worker

WHERE worker\_id=001

UNION ALL

SELECT \*

FROM worker

WHERE worker\_id=001;

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

SELECT \*

FROM worker

ORDER BY worker\_id

LIMIT (

SELECT FLOOR(COUNT(\*) / 2) FROM Worker

);

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

SELECT department

FROM worker

GROUP BY department

HAVING COUNT(\*)<3;

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

SELECT department, COUNT(\*)

FROM worker

GROUP BY department;

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

SELECT \*

FROM worker

ORDER BY worker\_id DESC

LIMIT 5;

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

SELECT first\_name, last\_name, department, salary

FROM worker w

WHERE salary = (

SELECT MAX(salary)

FROM worker

WHERE department = w.department

);

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

SELECT DISTINCT salary

FROM worker

ORDER BY salary DESC

LIMIT 3;

1. Write an SQL query to print the name of employees having the lowest salary in account and admin department

SELECT first\_name, last\_name, department, salary

FROM worker w

WHERE department IN ('account', 'admin')

AND salary = (

SELECT MIN(salary)

FROM worker

WHERE department = w.department

);