CREATE DATABASE ORG;

SHOW DATABASES;

USE ORG;

-- command to drop the table

drop table employee\_info;

-- command to create a table

CREATE TABLE employee\_info

(

emp\_id INT PRIMARY KEY auto\_increment,

first\_name Varchar(30),

last\_name Varchar(30),

gender Varchar(30),

DOB date,

department Varchar(30),

salary bigint(20),

city Varchar(30)

);

-- command to insert data into table

insert into employee\_info values

(1,'AYUSH', 'YADUVANSHI', 'MALE', '2000-09-20','FINANCE',300000,'BANGALORE'),

(2,'ASHUTOSH', 'ROY', 'MALE', '2000-04-11','HR',300000,'KOLKATA'),

(3,'KANKSHIT', 'BHOJNE', 'MALE','1999-04-11','IT',300000,'BANGALORE'),

(4,'RITIK', 'BHATI', 'MALE', '2001-08-28','ADMIN',150000,'NEWDELHI'),

(5,'SHAZIA', 'SHEHZAD', 'FEMALE', '2000-07-07','HR',250000,'NEWDELHI'),

(6,'ZOYA', 'KHAN', 'FEMALE', '2000-01-01','ACCOUNT',150000,'NEWDELHI'),

(7,'AMIT', 'BAJYA', 'MALE', '2001-03-10','ADMIN',75000,'MUMBAI'),

(8,'SHEKHAR', 'RANA', 'MALE', '1999-09-03','ACCOUNT',100000,'MUMBAI'),

(9,'SNEHA', 'SHARMA', 'FEMALE', '2000-08-30','ADMIN',50000,'HYDERABAD'),

(10,'SHINAM', 'KAMBOJ', 'FEMALE', '2001-06-02','ADMIN',200000,'HYDERABAD'),

(11,'AMRIT', 'AULAKH', 'MALE', '2000-09-10','IT',250000,'KOLKATA'),

(12,'YOGESH', 'SHARMA', 'MALE', '1999-04-15','ACCOUNT',150000,'KOLKATA'),

(13,'AMRIT', 'KAUR', 'FEMALE', '2000-08-04','FINANCE',100000,'MUMBAI'),

(14,'ABHISHEK', 'ARYA', 'MALE','1999-08-03','ACCOUNT',75000,'BANGALORE'),

(15,'ROHIT', 'CHANDEL', 'MALE', '2001-03-22','IT',50000,'NEWDELHI'),

(16,'ROBIN', 'BRAR', 'FEMALE', '2001-11-10','ADMIN',75000,'MUMBAI'),

(17,'GEETANJALI', 'VERMA', 'FEMALE', '2001-04-17','FINANCE',100000,'BANGALORE'),

(18,'YUVRAJ', 'RANA', 'MALE', '2000-02-11','FINANCE',100000,'BANGALORE'),

(19,'SANDEEP', 'SINGH', 'MALE', '2000-01-15','HR',100000,'KOLKATA'),

(20,'SIMRANJEET', 'SINGH', 'MALE','1999-10-01','IT',50000,'BANGALORE'),

(21,'MANISH', 'KUMAR', 'MALE', '2001-011-25','IT',75000,'NEWDELHI'),

(22,'MONIKA', 'RAJPUT', 'FEMALE', '1999-06-30','HR',100000,'KOLKATA'),

(23,'PAYAL', 'DEEWAN', 'FEMALE', '2000-07-07','ACCOUNT',50000,'HYDERABAD'),

(24,'NANDAN', 'RAJ', 'MALE', '2000-02-12','FINANCE',200000,'NEWDELHI'),

(25,'RAHUL', 'BHAGAT', 'MALE', '1998-12-15','ADMIN',150000,'MUMBAI'),

(26,'MANISH', 'SINGH', 'MALE','1999-11-01','IT',150000,'BANGALORE'),

(27,'SAURABH', 'MEHTA', 'MALE', '1999-02-02','ACCOUNT',200000,'HYDERABAD'),

(28,'ZARA', 'KHAN', 'FEMALE', '2001-05-17','IT',100000,'MUMBAI'),

(29,'SONA', 'THAKUR', 'FEMALE', '2000-05-07','ADMIN',150000,'HYDERABAD'),

(30,'PETER', 'ANDERSON', 'MALE', '1999-10-19','IT',150000,'NEWDELHI');

select \* from employee\_info;

CREATE TABLE employee\_position

(

emp\_id INT (20),

emp\_position Varchar(30),

date\_of\_joining date,

bonus bigint(20)

);

insert into employee\_position values

(1, 'EXECUTIVE', '2020-01-07',50000),

(2, 'EXECUTIVE', '2020-01-08',45000),

(3, 'EXECUTIVE', '2020-01-010',55000),

(4, 'SUPERVISOR', '2020-02-07',25000),

(5, 'MANAGER', '2020-02-12',40000),

(6, 'LEAD', '2020-02-15',20000),

(7, 'ANALYST', '2020-06-13',20000),

(8, 'SDE', '2020-05-17',20000),

(9, 'SDE', '2020-08-15',20000),

(10, 'MANAGER', '2020-08-20',40000),

(11, 'MANAGER', '2020-10-06',40000),

(12, 'SUPERVISOR', '2020-11-02',25000),

(13, 'LEAD', '2020-11-25',20000),

(14, 'SDE', '2021-01-03',0),

(15, 'SDE', '2021-01-06',15000),

(16, 'LEAD', '2021-01-14',10000),

(17, 'LEAD', '2020-02-10',25000),

(18, 'ANALYST', '2021-04-13',0),

(19, 'ANALYST', '2021-06-13',20000),

(20, 'SUPERVISOR', '2021-07-08',10000),

(21, 'SDE', '2021-08-21',10000),

(22, 'SDE', '2021-08-27',15000),

(23, 'LEAD', '2021-10-10',25000),

(24, 'MANAGER', '2022-04-12',35000),

(24, 'ANALYST', '2022-06-19',15000),

(25, 'LEAD', '2022-07-08',0),

(26, 'SDE', '2022-010-11',0),

(27, 'MANAGER', '2023-01-11',30000),

(28, 'ANALYST', '2023-02-22',0),

(29, 'LEAD', '2023-03-24',20000),

(30, 'SDE', '2023-05-12',20000);

select \* from employee\_position;

-- Q1. write a query to delete the records whose bonus is zero.

delete from employee\_position where bonus = 0;

-- Q2. write a query to change column name from emp\_position to position.

alter table employee\_position rename column emp\_position to position;

-- Q3. Write an SQL query to fetch “FIRST\_NAME” from employee\_info table in lower case.

select LOWER(first\_name) from employee\_info;

-- Q4. Write an SQL query to replace first\_name zoya with hina.

select REPLACE(first\_name,'ZOYA','HINA') from employee\_info;

-- Q5. Write an SQL query to fetch unique DEPARTMENT from employee\_info.

SELECT distinct department from employee\_info;

-- Q6. write a query to replace city kolkata with ahmedabad

select CITY, replace(city,'kolkata','ahmedabad') from employee\_info;

-- Q7.. Write an SQL query to find the position of the alphabet (‘U’) in the first\_name column from employee\_info table.

select INSTR(first\_name, 'U') from employee\_info

where first\_name = 'Ayush';

-- Q8. Write an SQL query that fetches the unique values of DEPARTMENT from employee\_info table and prints its length.

select distinct department, LENGTH(department) from employee\_info;

-- Q9. Write an SQL query to print the first three characters of distinct CITY from employee\_info table.

select distinct(substring(city, 1, 3)) from employee\_info;

-- Q10. Write an SQL query to print the FIRST\_NAME from employee\_info table after replacing ‘A’ with ‘@’.

select REPLACE(first\_name, 'A', '@') from employee\_info;

-- Q11. Write an SQL query to print the FIRST\_NAME and LAST\_NAME from employee\_info table into a single column FULL\_NAME.

select CONCAT(first\_name, ' ', last\_name) AS FULL\_NAME from employee\_info;

-- Q12. Write an SQL query to print all employee details from the employee\_info table order by FIRST\_NAME Ascending.

select \* from employee\_info ORDER by first\_name;

-- Q13. Write an SQL query to print details for employee with the last name as “sharma”,"singh" and “khan” from employee\_info table.

select \* from employee\_info where last\_name IN ('sharma', 'Singh','khan') ORDER by first\_name;

-- Q14. Write an SQL query to print details of employee with DEPARTMENT name as “Admin\*”.

select \* from employee\_info where department = 'Admin';

-- Q15. Write an SQL query to fetch the departments that have less than 6 employee in it.

select department, count(department) as depCount from employee\_info

group by department having depCount < 6;

-- Q16. Write an SQL query to fetch the count of employees working in the distinct department.

select department, count(\*) from employee\_info group by department order by count(\*) desc;

-- Q17. Write an SQL query to fetch the count of employees working in the distinct city in descending order.

select city, count(\*) from employee\_info group by city order by count(\*) desc;

-- Q18.write a query to fetch the details of female employee whose city is bangalore.

select \* from employee\_info where city = 'bangalore' and gender = 'female';

-- Q19. Write an SQL query to print details of the employee whose FIRST\_NAME contains ‘O’.

select \* from employee\_info where first\_name LIKE '%O%';

-- Q20. Write an SQL query to print details of the employee whose FIRST\_NAME ends with ‘A’.

select \* from employee\_info where first\_name LIKE '%A';

-- Q21. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘A’ and contains 4 alphabets.

select \* from employee\_info where first\_name LIKE '\_\_\_A';

-- Q22. Write an SQL query to print details of the Workers whose last\_name contains 5 alphabets.

select \* from employee\_info where last\_name LIKE '\_\_\_\_\_';

-- Q23. Write an SQL query to fetch the count of female employees from employee\_info table.

select count(\*) as total\_female\_employee from employee\_info

where gender= 'female';

-- Q24. Write an SQL query to print details of the employee whose SALARY lies between 100000 and 200000.

select \* from employee\_info where salary > 100000 AND salary < 200000;

-- Q25. Write an SQL query to fetch employees full names with salaries >= 200000.

select concat(first\_name, ' ', last\_name) as full\_name, salary from employee\_info

where salary >= 200000 ;

-- Q26. Write an SQL query to print details of the employees who are Managers.

select e.\* from employee\_info as e inner join employee\_position as p

on e.emp\_id = p.emp\_id where p.emp\_position = 'Manager';

-- Q27. write a query to fetch third and fourth record from a table.

select \* from employee\_info order by emp\_id limit 2 offset 2 ;

-- Q28. Write an SQL query to show the first record from a table.

select \* from employee\_info where emp\_id = (select min(emp\_id) from employee\_info);

-- Q29. Write an SQL query to show the last record from a table.

select \* from employee\_info where emp\_id = (select max(emp\_id) from employee\_info);

-- Q30. Write an SQL query to fetch the last five records from a table.

(select \* from employee\_info order by emp\_id desc limit 5) order by emp\_id;

-- Q31. Write an SQL query to fetch the first 50% records from a table.

select \* from employee\_info where emp\_id <= ( select count(emp\_id)/2 from employee\_info);

-- Q32. Write an SQL query to show only odd records from an employee table.

select \* from employee\_info where MOD (emp\_id, 2) != 0;

-- Q33. Write an SQL query to show only even records from an employee table.

select \* from employee\_info where MOD (emp\_id, 2) = 0;

-- Q34. Write an SQL query to fetch alternate records from an employee\_info table.

SELECT \*

FROM (

SELECT \*,

ROW\_NUMBER() OVER (ORDER BY emp\_id) AS row\_num

FROM employee\_info

) AS numbered\_rows

WHERE row\_num % 2 = 1;

-- Q35. Write an SQL query to clone a new table from another table.

CREATE TABLE employee\_details LIKE employee\_info;

INSERT INTO employee\_details select \* from employee\_info;

-- Q36. Write an SQL query to print details of the employee who have joined in jan 2021.

select \* from employee\_position where YEAR(date\_of\_joining) = 2021 AND MONTH(date\_of\_joining) = 01;

-- Q37. Write an SQL query to print name of the employee whose birthday is in august.

select first\_name, last\_name from employee\_info where MONTH(DOB) = 08;

-- Q38. Write an SQL query to print details of the employee who had born after 2000.

select \* from employee\_info where YEAR(DOB) >= 2000;

-- Q39. Write an SQL query to show the current date.

select curdate();

-- Q40. Write an SQL query to show the current date and time.

select now();

-- Q41. Write an SQL query to show the top 3 salary from a table order by descending salary.

select distinct salary from employee\_info order by salary desc LIMIT 3;

-- Q42. Write an SQL query to show the second highest salary from a table using sub-query.

select max(salary) as second\_highest\_salary from employee\_info

where salary < (select max(salary) from employee\_info);

-- Q43. Write an SQL query to determine the 5th highest salary without using LIMIT keyword.

select distinct salary from employee\_info e1

where 5 = (select count(distinct(salary)) from employee\_info e2

where e2.salary >= e1.salary);

-- Q44. Write an SQL query to show one row twice in results from a table.

select \* from employee\_info

UNION ALL

select \* from employee\_info ORDER BY emp\_id;

-- Q45. Write an SQL query to list emp\_name, department who does not get bonus.

select e.first\_name,e.last\_name,e.department from employee\_info as e

inner join employee\_position as p

on e.emp\_id = p.emp\_id where bonus = 0;

-- Q46. Write an SQL query to fetch the names of employee who earn the highest salary.

select first\_name, salary from employee\_info where salary = (select max(Salary) from employee\_info);

-- Q47. Write an SQL query to fetch maximum salary given in each department.

select department, max(salary) from employee\_info

group by department;

-- Q48. Write an SQL query to print the department and name of employees having the highest salary in each department.

select e.first\_name, e.last\_name, e.department,

max(salary) over (partition by department order by salary desc) as sal\_rank

from employee\_info as e;

-- Q 49. fetch the first 2 employees from each department to join the company.

select \* from (

select \* ,

row\_number() over(partition by department order by emp\_id) as rn

from employee\_info ) as x

where x.rn <3;

-- Q50. Write an SQL query to fetch the details of top 2 employees who earn the highest salary in each department.

select \* from (

select \* ,

dense\_rank() over(partition by department order by salary desc) as sal\_rank

from employee\_info ) as x

where x.sal\_rank <3;

-- Q51. Write an SQL query to fetch details of employee who earn the highest salary from each city.

select \* from (

select \* ,

dense\_rank() over(partition by city order by salary desc) as sal\_rank

from employee\_info ) as x

where x.sal\_rank <2;