**SEED ORGANISATION DATABASE**

**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, '21-02-20 09.00.00', 'HR'),**

**(002, 'Niharika', 'Verma', 80000, '21-06-11 09.00.00', 'Admin'),**

**(003, 'Vishal', 'Singhal', 300000, '21-02-20 09.00.00', 'HR'),**

**(004, 'Amitabh', 'Singh', 500000, '21-02-20 09.00.00', 'Admin'),**

**(005, 'Vivek', 'Bhati', 500000, '21-06-11 09.00.00', 'Admin'),**

**(006, 'Vipul', 'Diwan', 200000, '21-06-11 09.00.00', 'Account'),**

**(007, 'Satish', 'Kumar', 75000, '21-01-20 09.00.00', 'Account'),**

**(008, 'Geetika', 'Chauhan', 90000, '21-04-11 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, '23-02-20'),**

**(002, 3000, '23-06-11'),**

**(003, 4000, '23-02-20'),**

**(001, 4500, '23-02-20'),**

**(002, 3500, '23-06-11');**

**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', '2023-02-20 00:00:00'),**

**(002, 'Executive', '2023-06-11 00:00:00'),**

**(008, 'Executive', '2023-06-11 00:00:00'),**

**(005, 'Manager', '2023-06-11 00:00:00'),**

**(004, 'Asst. Manager', '2023-06-11 00:00:00'),**

**(007, 'Executive', '2023-06-11 00:00:00'),**

**(006, 'Lead', '2023-06-11 00:00:00'),**

**(003, 'Lead', '2023-06-11 00:00:00');**

**SEED ORGANISATION SQL QUESTIONS**

-- 1) Write an SQL query to fetch “FIRST\_NAME” from the Worker table using the alias name <WORKER\_NAME>.

-- 2) Write an SQL query to fetch “FIRST\_NAME” from the Worker table in upper case.

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

-- 4) Write an SQL query to print the first three characters of FIRST\_NAME from the Worker table.

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

-- 6) Write an SQL query to print the FIRST\_NAME from the Worker table after removing white spaces from the right side.

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

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

-- 9) Write an SQL query to print the FIRST\_NAME from the Worker table after replacing ‘a’ with ‘A’.

-- 10) Write an SQL query to print the FIRST\_NAME and LAST\_NAME from the Worker table into a single column COMPLETE\_NAME. A space char should separate them.

-- 11) Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending.

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

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

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

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

-- 16) Write an SQL query to print details of the Workers whose FIRST\_NAME contains ‘a’

-- 17) Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘a’.

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

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

-- 20) Write an SQL query to print details of the Workers who joined in Feb 2021.

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

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

-- 23) Write an SQL query to fetch the number of workers for each department in descending order.

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

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

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

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

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

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

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

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

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

**let’s take a look at some more advanced SQL query questions. These questions will require us to use more complex SQL syntax and concepts, such as nested queries, joins, unions, and intersects.**

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

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

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

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

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

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

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

**Practicing SQL query interview questions is a great way to improve your understanding of the language and become more proficient in using it. However, in addition to improving your technical skills, practicing SQL query questions can also help you advance your career. Many employers are looking for candidates who have strong SQL skills, so being able to demonstrate your proficiency in the language can give you a competitive edge.**

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

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

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

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

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

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

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

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

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

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

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