**Prepare Sample Data To Practice SQL Skill.**

**Sample Table – Worker**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WORKER\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** |
| 001 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 002 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 003 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 004 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 005 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 006 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 007 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| 008 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

**Sample Table – Bonus**

|  |  |  |
| --- | --- | --- |
| **WORKER\_REF\_ID** | **BONUS\_DATE** | **BONUS\_AMOUNT** |
| 1 | 2016-02-20 00:00:00 | 5000 |
| 2 | 2016-06-11 00:00:00 | 3000 |
| 3 | 2016-02-20 00:00:00 | 4000 |
| 1 | 2016-02-20 00:00:00 | 4500 |
| 2 | 2016-06-11 00:00:00 | 3500 |

**Sample Table – Title**

|  |  |  |
| --- | --- | --- |
| **WORKER\_REF\_ID** | **WORKER\_TITLE** | **AFFECTED\_FROM** |
| 1 | Manager | 2016-02-20 00:00:00 |
| 2 | Executive | 2016-06-11 00:00:00 |
| 8 | Executive | 2016-06-11 00:00:00 |
| 5 | Manager | 2016-06-11 00:00:00 |
| 4 | Asst. Manager | 2016-06-11 00:00:00 |
| 7 | Executive | 2016-06-11 00:00:00 |
| 6 | Lead | 2016-06-11 00:00:00 |
| 3 | Lead | 2016-06-11 00:00:00 |

**SQL Script to Seed Sample Data.**

CREATE DATABASE ORG;

SHOW DATABASES;

USE ORG;

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');

### 50 SQL Query Questions and Answers for Practice.

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

**Ans.**

select FIRST\_NAME AS WORKER\_NAME from Worker;

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

**Ans.**

Select upper(FIRST\_NAME) from Worker;

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

**Ans.**

Select distinct DEPARTMENT from Worker;

#### Q-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;

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

**Ans.**

Select INSTR(FIRST\_NAME, BINARY'a') from Worker where FIRST\_NAME = 'Amitabh';

**Notes.**

* The INSTR method is in case-sensitive by default.
* Using Binary operator will make INSTR work as the case-sensitive function.

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

**Ans.**

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 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%';

#### 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';

#### 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.**

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

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

**Ans.**

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(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.**

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.**

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.**

SELECT \* FROM Worker WHERE MOD (WORKER\_ID, 2) <> 0;

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

**Ans.**

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:

SELECT \* INTO WorkerClone FROM Worker;

The general way to clone a table without information is:

SELECT \* INTO WorkerClone FROM Worker WHERE 1 = 0;

An alternate way to clone a table (for MySQL) without is:

CREATE TABLE WorkerClone LIKE 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;

#### 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.**

Select distinct W.WORKER\_ID, W.FIRST\_NAME, W.Salary

from Worker W, Worker W1

where W.Salary = W1.Salary

and W.WORKER\_ID != W1.WORKER\_ID;

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

**Ans.**

Select max(Salary) from Worker

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

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

**Ans.**

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.**

(SELECT \* FROM Worker)

INTERSECT

(SELECT \* FROM WorkerClone);

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

**Ans.**

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.**

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.**

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.**

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.**

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.**

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;

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

**Ans.**

SELECT t.DEPARTMENT,t.FIRST\_NAME,t.Salary from(SELECT max(Salary) as TotalSalary,DEPARTMENT from Worker group by DEPARTMENT) as TempNew

Inner Join Worker t on TempNew.DEPARTMENT=t.DEPARTMENT

and TempNew.TotalSalary=t.Salary;

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

**Ans.**

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.**

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.**

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.**

 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.**

SELECT FIRST\_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);