SQL_Assignment_B

Worker

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Monika	Arora	100000	00:00.0	HR
2	Niharika	Verma	80000	00:00.0	Admin
3	Vishal	Singhal	300000	00:00.0	HR
4	Amitabh	Singh	500000	00:00.0	Admin
5	Vivek	Bhati	500000	00:00.0	Admin
6	Vipul	Diwan	200000	00:00.0	Account
7	Satish	Kumar	75000	00:00.0	Account
8	Geetika	Chauhan	90000	00:00.0	Admin

Bonus

WORKER_REF_ID	BONUS_AMOUNT	BONUS_DATE
1	5000	00:00.0
2	3000	00:00.0
3	4000	00:00.0
1	4500	00:00.0
2	3500	00:00.0

Title

WORKER_REF_ID	WORKER_TITLE	AFFECTED_FROM
1	Manager	00:00.0
2	Executive	00:00.0
8	Executive	00:00.0
5	Manager	00:00.0
4	Asst. Manager	00:00.0
7	Executive	00:00.0
6	Lead	00:00.0
3	Lead	00:00.0

```sql
/*

Creating Databse - SQL_Assignment_B

*/

CREATE DATABASE SQL_Assignment_B;
USE SQL_Assignment_B;


CREATE TABLE Worker (
       WORKER_ID INT NOT NULL identity(1,1),
       FIRST_NAME varchar(25),
       LAST_NAME varchar(25),
       SALARY INT,
       JOINING_DATE DATETIME,
       DEPARTMENT varchar(25),
       PRIMARY KEY (WORKER_ID)
);


INSERT INTO Worker
       (FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE, DEPARTMENT) VALUES
                     ('Monika', 'Arora', 100000, '02/14/20 09:00:00', 'HR'),
              ('Niharika', 'Verma', 80000, '06/14/11 09:00:00', 'Admin'),
              ('Vishal', 'Singhal', 300000, '02/14/20 09:00:00', 'HR'),
              ('Amitabh', 'Singh', 500000, '02/14/20 09:00:00', 'Admin'),
              ('Vivek', 'Bhati', 500000, '06/14/11 09:00:00', 'Admin'),
              ('Vipul', 'Diwan', 200000, '06/14/11 09:00:00', 'Account'),
              ('Satish', 'Kumar', 75000, '01/14/20 09:00:00', 'Account'),
              ('Geetika', 'Chauhan', 90000, '04/14/11 09:00:00', 'Admin');




CREATE TABLE Bonus (
       WORKER_REF_ID INT,
       BONUS_AMOUNT INT,
       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
              (1, 5000, '02/16/20'),
              (2, 3000, '06/16/11'),
              (3, 4000, '02/16/20'),
              (1, 4500, '02/16/20'),
              (2, 3500, '06/16/11');
```

```
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
 (1, 'Manager', '06/20/2016 00:00:00'),
 (2, 'Executive', '06/11/2016 00:00:00'),
 (8, 'Executive', '06/11/2016 00:00:00'),
 (5, 'Manager', '06/11/2016 00:00:00'),
 (4, 'Asst. Manager', '06/11/2016 00:00:00'),
 (7, 'Executive', '06/11/2016 00:00:00'),
 (6, 'Lead', '06/11/2016 00:00:00'),
 (3, 'Lead', '06/11/2016 00:00:00');


select * from worker
select * from bonus
select * from title
```

```sql
-- 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.
select DISTINCT DEPARTMENT
from worker;



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

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.
-- Select INSTR(FIRST_NAME, BINARY'a') from Worker where FIRST_NAME = 'Amitabh' -

select charindex( 'a', first_name)
from worker
where  FIRST_NAME = 'Amitabh'


SELECT collation_name
FROM sys.databases


SELECT COLUMN_NAME, COLLATION_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'worker' AND CHARACTER_SET_NAME IS NOT NULL


-- Modified wrt SQL SERVER
--- Q-5. Write an SQL query to find 2nd occurence of the position of the alphabet
(‘a’) in the first name column ‘Amitabh’ from Worker table.

select charindex( 'a', first_name, 0)
from worker
where  FIRST_NAME = 'Amitabh'

select charindex( 'a', first_name, CHARINDEX('a', first_name) + 1)
from worker
where  FIRST_NAME = 'Amitabh'
-- this will start after 1st occurence of 'a'
-- Q-6. Write an SQL query to print the FIRST_NAME from Worker table after removing
white spaces from the right side

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.

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.

select count(distinct DEPARTMENT)
from worker


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

select (first_name + ' ' + last_name) as complete_name
from worker


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.

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.


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.

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.

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”.

select *
from worker
where DEPARTMENT like 'admin%'



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


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’.

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.

select *
from worker
where FIRST_NAME like '%h' and len(first_name) = 6;


select *
from worker
where FIRST_NAME like '_____h';
```

```sql
-- Q-19. 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



-- Q-20. Write an SQL query to print details of the Workers who have joined in
Feb'2014.
--** Answer to be found
select *
from worker
where (year(JOINING_DATE) = 2014) and (month(JOINING_DATE) = 2);

select *
from worker
where (year(JOINING_DATE) = 2014);

select year(JOINING_DATE), month(JOINING_DATE)
from worker


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

select count(*)
from worker
where DEPARTMENT like '%admin%'


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

select concat(FIRST_NAME, ' ', LAST_NAME) as worker_name, salary
from worker
where SALARY between 50000 and 100000


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.

select count(department) as dept_count, department
from worker
group by department
order by dept_count desc
```

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

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 'title'.

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

-- Q-27. Write an SQL query to show only even rows from a table.
--- SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) = 0;



/*
SELECT *
FROM Worker
WHERE MOD(WORKER_ID, 2) = 1;
*/



SELECT *
FROM (
    SELECT *, Row_Number() OVER(ORDER BY WORKER_ID) AS RowNumber
            --Row_Number() starts with 1
    FROM worker
) t
WHERE t.RowNumber % 2 = 0 --Even
--WHERE t.RowNumber % 2 = 1 --Odd



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

-- 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 WorkerClone2 FROM Worker WHERE 1 = 0;


--- An alternate way to clone a table (for MySQL) without is:
-- CREATE TABLE WorkerClone LIKE Worker;

SELECT * from WorkerClone;
```

```sql
-- Q-29. Write an SQL query to fetch intersecting records of two tables WorkerClone
and Worker.

/*
select w.WORKER_ID, w.FIRST_NAME, w.LAST_NAME, w.JOINING_DATE , w.DEPARTMENT ,
t.WORKER_TITLE , t.AFFECTED_FROM
from worker w
inner join title t
on w.worker_id = t.WORKER_REF_ID
*/

(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.

SELECT * FROM Worker
MINUS
SELECT * FROM Title;


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

select GETDATE()


--- Q-32. Write an SQL query to show the top n (say 3) records of a table and sort by
salary.


select top 3*
from worker
order by salary


/* MYSQL
select *
from worker
order by salary
limit 3
*/




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


select *, ROW_NUMBER() over(order by salary desc) as rm
from worker


select distinct salary, ROW_NUMBER() over(order by salary desc) as rm
from worker
order by rm
```

```sql
select * from(
select *, ROW_NUMBER() over(order by salary desc) as rm
from worker) wt
where wt.rm = 5;


SELECT TOP 1*
FROM (
 SELECT DISTINCT TOP 5 Salary
 FROM Worker
 ORDER BY Salary DESC
 ) wt
ORDER BY Salary ASC;

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

--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.

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.

select * from(
select *,
 dense_rank() over(order by salary desc) rk
from worker ) tableName
where rk = 2


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

```sql
-- Notes Practice Below
---------------------------------
select *
from ( select *,  ROW_NUMBER () over(order by salary desc) as rm
from worker
) wk
where rm = 2


select top 1*
from worker


select * from worker order by salary desc

select top 1*
from
( select top 2* from worker order by salary desc)
worker


select *,
 rank() over(order by salary desc) rk
from worker


select *,
 dense_rank() over(order by salary desc) rk
from worker


select *,
 rank() over(order by department desc) rk
from worker


select *,
 rank() over(partition by department order by salary desc) rk
from worker

select *,
 dense_rank() over(order by department desc) rk
from worker
---------------------------------
```

```sql
-- Q-37. Write an SQL query to show one row twice in results from a table.
-- example, 'Amitabh' from first name shall appear twice.
-- example, row having 'hr' from department and salary more than 100000 shal apear
twice

/*
select * from worker where DEPARTMENT = 'hr'
union all
select * from worker where DEPARTMENT = 'hr'
*/


select * from worker where first_name = 'amitabh'
union all
select * from worker where first_name = 'amitabh'


select * from worker where DEPARTMENT = 'hr' and salary > 100000
union all
select * from worker where DEPARTMENT = 'hr' and salary > 100000



-- Q-38. Write an SQL query to fetch intersecting records of two tables (worker and
worker).

select * from worker
intersect
select * from worker


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

select top 50 percent*
from worker


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

select DEPARTMENT , count(department) as ['No of workers']
 from Worker
 group by DEPARTMENT
 having count(department) < 5


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

select DEPARTMENT , count(department) as ['No of workers']
 from Worker
 group by DEPARTMENT


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

select top 1 *
from worker order by WORKER_ID desc


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

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

select top 1 *
from worker

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.


select top 5 *
from worker order by WORKER_ID desc


-- my sql code
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.


select * from(
select *,
dense_rank() over(partition by department order by salary desc) rk
from worker) wk
where rk = 1


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


select top 3*
from worker
order by salary desc


-- my sql
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.

select top 3*
from worker
order by salary

-- my sql
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;
```

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

-- change value of n here to get the nth max
declare @n as int = 2

select * from (
select *, DENSE_RANK() over(order by salary desc) dk
from worker) wk
where dk = @n


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



select sum(salary), DEPARTMENT
from worker
group by DEPARTMENT


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

select CONCAT(FIRST_NAME, ' ', LAST_NAME) as [Worker Name]
from (
select *, RANK() over(order by salary desc) rk
from worker) wk
where rk = 1;
```