B Yaswanth 22BRS1269 vit Chennai

Sql queries (08/06/2024)

```
CREATE DATABASE ORG1;
SHOW DATABASES;
USE ORG1;
CREATE TABLE Worker1 (
 WORKER_ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
 FIRST_NAME CHAR(25),
 LAST_NAME CHAR(25),
 SALARY INT,
 JOINING_DATE DATETIME,
 DEPARTMENT CHAR(25)
);

INSERT INTO Worker1
(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 Bonus2 (
WORKER_REF_ID INT,
BONUS_AMOUNT INT,
BONUS_DATE DATETIME,
FOREIGN KEY (WORKER_REF_ID)
REFERENCES Worker1(WORKER_ID)
 ON DELETE CASCADE
);
INSERT INTO Bonus2
(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 Title3 (
WORKER_REF_ID INT,
WORKER_TITLE CHAR(25),
AFFECTED_FROM DATETIME,
```

**FOREIGN KEY (WORKER_REF_ID)**

**REFERENCES Worker1(WORKER_ID)**

**ON DELETE CASCADE**

**);**

**INSERT INTO Title3**

**(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');**

**Q 1 :**

**SELECT FIRST_NAME AS WORKER_NAME FROM Worker1;**

```
71 •  SELECT FIRST_NAME AS WORKER_NAME FROM Worker1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_NAME |
|---|
| Monika |
| Niharika |
| Vishal |
| Amitabh |
| Vivek |

Worker1 2

**Q-2. Write an SQL query to fetch "FIRST_NAME" from Worker table in upper case.**

```
73 •    SELECT UPPER(FIRST_NAME) FROM Worker1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

| UPPER(FIRST_NAME) |
|---|
| MONIKA |
| NIHARIKA |
| VISHAL |
| AMITABH |
| VIVEK |

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

```
75 •    SELECT DISTINCT DEPARTMENT
76      FROM Worker1;
77
78
79
80
81
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| DEPARTMENT |
|---|
| HR |
| Admin |
| Account |

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

```
78 •    SELECT SUBSTRING(FIRST_NAME, 1, 3) FROM Worker1;
79
80
81
82
83
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| SUBSTRING(FIRST_NAME, 1, 3) |
|---|
| Ami |
| Viv |
| Vip |
| Sat |
| Gee |

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

```
80 •    SELECT INSTR(FIRST_NAME, 'a') FROM Worker1 WHERE FIRST_NAME = 'Amitabh';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| INSTR(FIRST_NAME, 'a') |
|---|
| 1 |

**Q-6. Write an SQL query to print the FIRST_NAME from Worker table after removing white spaces from the right side.**

```
82    SELECT RTRIM(FIRST_NAME) AS FIRST_NAME
83    FROM Worker1;
84
85
86
```

Result Grid | Filter Rows: | Export: | Wrap Cell

| FIRST_NAME |
|---|
| Monika |
| Niharika |
| Vishal |
| Amitabh |
| Vivek |
| Vipul |

Result 7

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

```
85    SELECT LTRIM(DEPARTMENT) AS DEPARTMENT FROM Worker1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| DEPARTMENT |
|---|
| HR |
| Admin |
| HR |
| Admin |
| Admin |
| Account |

Result 8

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

```
87 •  SELECT DEPARTMENT, LENGTH(DEPARTMENT) FROM Worker1 GROUP BY DEPARTMENT;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| DEPARTMENT | LENGTH(DEPARTMENT) |
|---|---|
| HR | 2 |
| Admin | 5 |
| Account | 7 |

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

```
89 •  SELECT REPLACE(FIRST_NAME, 'a', 'A') AS FIRST_NAME FROM Worker1;
90
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| FIRST_NAME |
|---|
| MonikA |
| NihArikA |
| VishAl |
| AmitAbh |
| Vivek |
| Vipul |

Result 10

**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.**

```
91 •  SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS COMPLETE_NAME
92    FROM Worker1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| COMPLETE_NAME |
|---|
| Monika Arora |
| Niharika Verma |
| Vishal Singhal |
| Amitabh Singh |
| Vivek Bhati |
| Vipul Diwan |

Result 11

**Q-11. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending.**

```
94 •  SELECT *
95    FROM Worker1
96    ORDER BY FIRST_NAME ASC;
97
98
99
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

**Q-12. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending and DEPARTMENT Descending.**

```
98 •  SELECT *
99    FROM Worker1
100   ORDER BY FIRST_NAME ASC, DEPARTMENT DESC;
101
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |

Worker1 13 ×

**Q-13. Write an SQL query to print details for Workers with the first name as "Vipul" and "Satish" from Worker table.**

```
102 •   SELECT *
103     FROM Worker1
104     WHERE FIRST_NAME IN ('Vipul', 'Satish');
105
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| NULL | NULL | NULL | NULL | NULL | NULL |

**Q-14. Write an SQL query to print details of workers excluding first names, "Vipul" and "Satish" from Worker table.**

```
106 •   SELECT *
107     FROM Worker1
108     WHERE FIRST_NAME NOT IN ('Vipul', 'Satish');
109
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

**Q-15. Write an SQL query to print details of Workers with DEPARTMENT name as "Admin".**

```
110 •   SELECT *
111     FROM Worker1
112     WHERE DEPARTMENT = 'Admin';
113
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

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

```
114 •   SELECT *
115     FROM Worker1
116     WHERE FIRST_NAME LIKE '%a%';
117
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

**Q-17. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'a'.**

```
118 •  SELECT *
119    FROM Worker1
120    WHERE FIRST_NAME LIKE '%a';
121
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell C

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

**Q-18. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'h' and contains six alphabets.**

```
122 •  SELECT *
123    FROM Worker1
124    WHERE FIRST_NAME LIKE '_____h';
125
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| NULL | NULL | NULL | NULL | NULL | NULL |

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

```
126 •   SELECT *
127     FROM Worker1
128     WHERE SALARY BETWEEN 100000 AND 500000;
129
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| NULL | NULL | NULL | NULL | NULL | NULL |

**Q-20. Write an SQL query to print details of the Workers who have joined in Feb 2014.**

```
130 •   SELECT *
131     FROM Worker1
132     WHERE YEAR(JOINING_DATE) = 2014 AND MONTH(JOINING_DATE) = 2;
133
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

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

```
134 •   SELECT COUNT(*) AS employee_count
135     FROM Worker1
136     WHERE DEPARTMENT = 'Admin';
137
```

Result Grid | Filter Rows: | Export:

| employee_count |
|---|
| 4 |

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

```
138 •   SELECT FIRST_NAME, LAST_NAME
139     FROM Worker1
140     WHERE SALARY BETWEEN 50000 AND 100000;
141
```

Result Grid | Filter Rows: | Export: | Wrap

| FIRST_NAME | LAST_NAME |
|---|---|
| Monika | Arora |
| Niharika | Verma |
| Satish | Kumar |
| Geetika | Chauhan |

**Q-23. Write an SQL query to fetch the no. of workers for each department in the descending order.**

```sql
SELECT DEPARTMENT, COUNT(*) AS worker_count
FROM Worker1
GROUP BY DEPARTMENT
ORDER BY worker_count DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| DEPARTMENT | worker_count |
|---|---|
| Admin | 4 |
| HR | 2 |
| Account | 2 |

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

```sql
SELECT *
FROM Worker1
WHERE WORKER_ID IN (
    SELECT WORKER_REF_ID
    FROM Title3
    WHERE WORKER_TITLE = 'Manager'
);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

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

```
155    SELECT FIRST_NAME, LAST_NAME, COUNT(*)
156    FROM Worker1
157    GROUP BY FIRST_NAME, LAST_NAME
158    HAVING COUNT(*) > 1;
159
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| | FIRST_NAME | LAST_NAME | COUNT(*) |
|---|---|---|---|

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

```
160  WITH NumberedRows AS (
161      SELECT *, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum
162      FROM Worker1
163  )
164  SELECT *
165  FROM NumberedRows
166  WHERE RowNum % 2 != 0
167  ORDER BY RAND();
168
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | RowNum |
|---|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR | 1 |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR | 3 |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin | 5 |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account | 7 |

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

```
169  WITH NumberedRows AS (
170      SELECT *, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) AS RowNum
171      FROM Worker1
172  )
173  SELECT *
174  FROM NumberedRows
175  WHERE RowNum % 2 = 0;
176
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | RowNum |
|---|---|---|---|---|---|---|
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin | 2 |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin | 4 |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account | 6 |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin | 8 |

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

```
177  CREATE TABLE ClonedWorkerTable AS
178  SELECT *
179  FROM Worker1;
180
```

59 21:15:06 CREATE TABLE ClonedWorkerTable AS SELEC... 8 rov

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

```
181  SELECT *
182  FROM Worker1
183  INNER JOIN Bonus2 ON Worker1.WORKER_ID = Bonus2.WORKER_REF_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | WORKER_REF_ID | BONUS_AMO |
|---|---|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR | 1 | 5000 |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin | 2 | 3000 |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR | 3 | 4000 |
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR | 1 | 4500 |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin | 2 | 3500 |

Result Grid | Form Editor

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

```
185 •   SELECT Worker1.*
186     FROM Worker1
187     LEFT JOIN Bonus2 ON Worker1.WORKER_ID = Bonus2.WORKER_REF_ID
188     WHERE Bonus2.WORKER_REF_ID IS NULL;
189
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

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

```
190 •   SELECT NOW() AS CurrentDateTime;
191
```

Result Grid | Filter Rows: | Export:

| CurrentDateTime |
|---|
| 2024-06-08 21:18:25 |

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

```
192 •   SELECT *
193     FROM Worker1
194     LIMIT 4;
195
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

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

```
196 •   SELECT DISTINCT Salary
197     FROM Worker1
198     ORDER BY Salary DESC
199     LIMIT 4, 1;
200
```

Result Grid | Filter Rows:

| | Salary |
|---|---|
| ▶ | 90000 |

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

```
202 •   SELECT Salary
203     FROM (
204         SELECT Salary, ROW_NUMBER() OVER (ORDER BY Salary DESC) AS RowNum
205         FROM Worker1
206     ) AS Subquery
207     WHERE RowNum = 5;
208
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| | Salary |
|---|---|
| ▶ | 100000 |

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

```
209 •   SELECT Salary, GROUP_CONCAT(FIRST_NAME, ' ', LAST_NAME) AS Employees
210     FROM Worker1
211     GROUP BY Salary
212     HAVING COUNT(*) > 1;
213
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| | Salary | Employees |
|---|---|---|
| ▶ | 500000 | Amitabh Singh,Vivek Bhati |

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

```
214 •   SELECT MAX(Salary) AS SecondHighestSalary
215     FROM Worker1
216     WHERE Salary < (SELECT MAX(Salary) FROM Worker1);
217
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| SecondHighestSalary |
|---|
| 300000 |

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

```
218 •   SELECT *
219     FROM Worker1
220
221     UNION ALL
222
223     SELECT *
224     FROM Worker1
225     LIMIT 1;
226
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |

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

```
227 •   SELECT *
228     FROM Worker1
229     INNER JOIN Bonus2 ON Worker1.WORKER_ID = Bonus2.WORKER_REF_ID;
230
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | WORKER_REF_ID | BONUS_AMC |
|---|---|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR | 1 | 5000 |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin | 2 | 3000 |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR | 3 | 4000 |
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR | 1 | 4500 |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin | 2 | 3500 |

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

```
SET @half_count = (SELECT CEIL(COUNT(*) * 0.5) FROM Worker1);
```

```
76  21:32:01  SET @half_count = (SELECT CEIL(COUNT(*) * 0...
```

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

```
235 •   SELECT DEPARTMENT
236     FROM Worker1
237     GROUP BY DEPARTMENT
238     HAVING COUNT(*) < 5;
239
240
```

Result Grid | Filter Rows:

| DEPARTMENT |
|---|
| HR |
| Admin |
| Account |

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

```
241 •  SELECT DEPARTMENT, COUNT(*) AS "Number of People"
242    FROM Worker1
243    GROUP BY DEPARTMENT;
244
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| DEPARTMENT | Number of People |
|---|---|
| HR | 2 |
| Admin | 4 |
| Account | 2 |

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

```
245 •  SELECT *
246    FROM Worker1
247    ORDER BY WORKER_ID DESC
248    LIMIT 1;
249
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |
| NULL | NULL | NULL | NULL | NULL | NULL |

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

```
250 •  SELECT *
251    FROM Worker1
252    WHERE WORKER_ID = (SELECT MIN(WORKER_ID) FROM Worker1);
253
```

Result Grid | Filter Rows: | Edit: | Export/Import:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 1 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| NULL | NULL | NULL | NULL | NULL | NULL |

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

```
255    SELECT *
256    FROM (
257        SELECT *, ROW_NUMBER() OVER (ORDER BY WORKER_ID DESC) AS row_num
258        FROM Worker1
259    ) AS ordered_workers
260    WHERE row_num BETWEEN 1 AND 5;
261
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | row_num |
|---|---|---|---|---|---|---|
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin | 1 |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account | 2 |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account | 3 |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin | 4 |
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin | 5 |

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

```
262    WITH MaxSalaries AS (
263        SELECT DEPARTMENT, MAX(SALARY) AS max_salary
264        FROM Worker1
265        GROUP BY DEPARTMENT
266    )
267    SELECT w.FIRST_NAME, w.LAST_NAME, w.DEPARTMENT
268    FROM Worker1 w
269    JOIN MaxSalaries m ON w.DEPARTMENT = m.DEPARTMENT AND w.SALARY = m.max_salary;
270
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| FIRST_NAME | LAST_NAME | DEPARTMENT |
|---|---|---|
| Vishal | Singhal | HR |
| Amitabh | Singh | Admin |
| Vivek | Bhati | Admin |
| Vipul | Diwan | Account |

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

```
271 •  SELECT *
272    FROM (
273        SELECT *,
274               ROW_NUMBER() OVER (ORDER BY SALARY DESC) AS row_num
275        FROM Worker1
276    ) AS ranked_workers
277    WHERE row_num <= 3;
278
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | row_num |
|---|---|---|---|---|---|---|
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin | 1 |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin | 2 |
| 3 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR | 3 |

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

```
279 •  SELECT *
280    FROM (
281        SELECT *,
282               ROW_NUMBER() OVER (ORDER BY SALARY) AS row_num
283        FROM Worker1
284    ) AS ranked_workers
285    WHERE row_num <= 3;
286
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | row_num |
|---|---|---|---|---|---|---|
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account | 1 |
| 2 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin | 2 |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin | 3 |

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

```
288    SELECT *
289    FROM (
290        SELECT *,
291               ROW_NUMBER() OVER (ORDER BY SALARY DESC) AS row_num
292        FROM Worker1
293    ) AS ranked_workers
294    WHERE row_num = 2;
295
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT | row_num |
|---|---|---|---|---|---|---|
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin | 2 |

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

```
296    SELECT DEPARTMENT, SUM(SALARY) AS total_salary_paid
297    FROM Worker1
298    GROUP BY DEPARTMENT;
299
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| DEPARTMENT | total_salary_paid |
|---|---|
| HR | 400000 |
| Admin | 1170000 |
| Account | 275000 |

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

```sql
SELECT FIRST_NAME, LAST_NAME
FROM Worker1
WHERE SALARY = (
    SELECT MAX(SALARY)
    FROM Worker1
);
```

Result Grid | Filter Rows: | Export:

| FIRST_NAME | LAST_NAME |
|---|---|
| Amitabh | Singh |
| Vivek | Bhati |