# SQL – PROJECT

## COURSE – MASTERS IN DATA SCIENCE AND ANALYSTS WITH AI

SUBMITTED BY – MR. YATEEN PAWAR

BATCH – 02:00 PM TO 04:00 PM

SUBMISSION DATE – 27TH FEBRUARY, 2023

PROJECT GUIDE
MRS. AKANKSHA RANE

SUBMITTED TO

MRS. AKANKSHA RANE


## I.T.VEDANT EDUCATION PVT. LTD
## - VASHI,NAVI MUMBAI

## ACKNOWLEDGEMENT

I would like to express my deepest gratitude to the Institute for providing me with the opportunity to work on a project based on SQL. This project would not have been possible without the guidance, support, and encouragement of several individuals.

First and foremost, I would like to thank my project guide, Mrs. Akanksha Rane for providing me with invaluable insights and suggestions throughout the project. Your expertise in the field of SQL has been instrumental in shaping this project and I am grateful for your mentorship.

I would also like to thank all the faculty members of the Institute for their support and encouragement throughout this project. Their feedback and inputs have been critical in shaping the project and making it better.

Furthermore, I would like to extend my gratitude to my classmates who have provided me with their support and inputs during the course of this project. Their contributions and feedback have been instrumental in shaping this project.

Finally, I would like to thank my family and friends for their unwavering support and encouragement throughout this project. Their support has been critical in keeping me motivated and focused during the course of this project.

Once again, I express my sincere thanks to everyone who has contributed to the successful completion of this project.

**SQL – PROJECT 1**
**YATEEN PAWAR**

Below question are based on 2 tables: -

**Table 1 = WORKER**

| 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 |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Admin |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

**Table 2 = 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 Questions as below:-**

Q-1. Write an SQL query to fetch “FIRST_NAME” from Worker table using the alias name as <WORKER_NAME>.

**Syntax** = SELECT FIRST_NAME AS WORKER_NAME FROM WORKER;

| WORKER_NAME |
|---|
| Monika |
| Niharika |
| Vishal |
| Amitabh |
| Vivek |
| Vipul |
| Satish |
| Geetika |

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

**Syntax** = SELECT UPPER(FIRST_NAME) FROM WORKER;

| UPPER(FIRST_NAME) |
|---|
| MONIKA |
| NIHARIKA |
| VISHAL |
| AMITABH |
| VIVEK |
| VIPUL |
| SATISH |
| GEETIKA |

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

**Syntax =** SELECT DISTINCT(DEPARTMENT) FROM WORKER;

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

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

**Syntax =** SELECT SUBSTRING(FIRST_NAME,1,3) FROM WORKER;

| SUBSTRING(FIRST_NAME,1,3) |
|---|
| Mon |
| Nih |
| Vis |
| 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.

**Syntax =** SELECT INSTR(FIRST_NAME, BINARY 'a') FROM WORKER WHERE FIRST_NAME = 'Amitabh';

| INSTR(FIRST_NAME, BINARY 'a') |
|---|
| 5 |

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

**Syntax =** SELECT RTRIM(FIRST_NAME) FROM WORKER;

| RTRIM(FIRST_NAME) |
|---|
| Monika |
| Niharika |
| Vishal |
| Amitabh |
| Vivek |
| Vipul |
| Satish |
| Geetika |

Q-7. Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and prints its length.

**Syntax =** SELECT DISTINCT LENGTH(DEPARTMENT) FROM WORKER;

| LENGTH(DEPARTMENT) |
|---|
| 2 |
| 5 |
| 7 |

Q-8. Write an SQL query to print the FIRST_NAME from Worker table after replacing 'a' with 'A'.

**Syntax =** SELECT REPLACE(FIRST_NAME,'a','A') FROM WORKER;

| REPLACE(FIRST_NAME,'a','A') |
|---|
| MonikA |
| NihArikA |
| VishAl |
| AmitAbh |
| Vivek |
| Vipul |
| SAtish |
| GeetikA |

Q-9. 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.

**Syntax =** SELECT CONCAT(FIRST_NAME,' ',LAST_NAME) AS 'COMPLETE_NAME' FROM WORKER;

| COMPLETE_NAME |
|---|
| Monika Arora |
| Niharika Verma |
| Vishal Singhal |
| Amitabh Singh |
| Vivek Bhati |
| Vipul Diwan |
| Satish Kumar |
| Geetika Chauhan |

Q-10. Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME Ascending.

**Syntax =** SELECT * FROM WORKER ORDER BY FIRST_NAME ASC;

| WORKER_ID | FIRST_NAME ▲ 1 | 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 | Admin |
| 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 |

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

**Syntax =** SELECT * FROM WORKER ORDER BY FIRST_NAME ASC,DEPARTMENT DESC;

| WORKER_ID | FIRST_NAME ▲ 1 | 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 | Admin |
| 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 |

Q-12. Write an SQL query to print details for Workers with the first name as “Vipul” and “Satish” from Worker table.

**Syntax =** SELECT * FROM WORKER WHERE FIRST_NAME IN ('VIPUL','SATISH');

| 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 | Admin |

Q-13. Write an SQL query to print details of workers excluding first names, “Vipul” and “Satish” from Worker table.

**Syntax =** SELECT * FROM WORKER WHERE FIRST_NAME NOT IN ('VIPUL','SATISH');

| 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 |

Q-14. Write an SQL query to print details of Workers with DEPARTMENT name as “Admin”.

**Syntax =** SELECT * FROM WORKER WHERE DEPARTMENT = 'ADMIN';

| 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 |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Admin |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

Q-15. Write an SQL query to print details of the Workers whose FIRST_NAME contains ‘a’.

**Syntax =** SELECT * FROM WORKER WHERE FIRST_NAME LIKE "%a%";

| 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 | Admin |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

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

**Syntax =** SELECT * FROM WORKER WHERE FIRST_NAME LIKE "%a";

| 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 |

Q-17. Write an SQL query to print details of the Workers whose FIRST_NAME ends with ‘h’ and contains six alphabets.

**Syntax =** SELECT * FROM WORKER WHERE FIRST_NAME LIKE "_____h";

| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE | DEPARTMENT |
|---|---|---|---|---|---|
| 4 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |

Q-18. Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

**Syntax =** SELECT * FROM WORKER WHERE SALARY BETWEEN 100000 AND 500000;

| 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 |

Q-19. Write an SQL query to print details of the Workers who have joined in Feb’2014.

**Syntax =** SELECT * FROM WORKER WHERE YEAR(JOINING_DATE) = 2014 AND MONTH(JOINING_DATE) = 2;

| 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 |

Q-20. Write an SQL query to fetch the count of employees working in the department 'Admin'.

**Syntax =** SELECT COUNT(*) FROM WORKER WHERE DEPARTMENT = 'ADMIN';

| COUNT(*) |
|---|
| 5 |

Q-21. Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.

**Syntax =** SELECT CONCAT(FIRST_NAME,' ',LAST_NAME) AS WORKER_NAMES FROM WORKER WHERE SALARY>= 50000 and SALARY<= 100000;
**OR**
WHERE SALARY BETWEEN 50000 AND 100000;

| WORKER_NAMES |
|---|
| Monika Arora |
| Niharika Verma |
| Satish Kumar |
| Geetika Chauhan |

Q-22. Write an SQL query to fetch the no. of workers for each department in the descending order.

**Syntax =** SELECT COUNT(*) NO_OF_WORKER,DEPARTMENT FROM WORKER GROUP BY DEPARTMENT ORDER BY DEPARTMENT DESC;

| NO_OF_WORKER | DEPARTMENT |
|---|---|
| 2 | HR |
| 5 | Admin |
| 1 | Account |

Q-23. Write an SQL query to print details of the Workers who are also Managers.

**Syntax =** SELECT DISTINCT WORKER.FIRST_NAME,TITLE.WORKER_TITLE FROM WORKER INNER JOIN TITLE ON WORKER.WORKER_ID = TITLE.WORKER_REF_ID AND TITLE.WORKER_TITLE = 'MANAGER';

| FIRST_NAME | WORKER_TITLE |
|---|---|
| Monika | Manager |
| Vivek | Manager |

Q-24. Write an SQL query to fetch duplicate records having matching data in some fields from title table

**Syntax =** SELECT WORKER_TITLE, AFFECTED_FROM, COUNT(*) FROM Title GROUP BY WORKER_TITLE, AFFECTED_FROM HAVING COUNT(*) > 1;

| WORKER_TITLE | AFFECTED_FROM | COUNT(*) |
|---|---|---|
| Executive | 2016-06-11 00:00:00 | 3 |
| Lead | 2016-06-11 00:00:00 | 2 |

Q-25. Write an SQL query to show only odd rows from a table.

**Syntax =** SELECT * FROM WORKER WHERE MOD (WORKER_ID,2) <> 0;

| 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 |
| 5 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 7 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Admin |

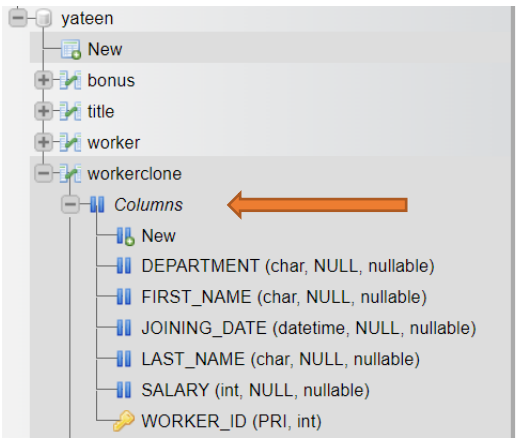
Q-26. Write an SQL query to show only even rows from a table.

**Syntax =** SELECT * FROM Worker WHERE MOD (WORKER_ID, 2) = 0;

| 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 |
| 6 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 8 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

Q-27. Write an SQL query to clone a new table from another table.

**Syntax =** CREATE TABLE WORKERCLONE LIKE WORKER;



Q-28. Write an SQL query to show the current date and time.

**Syntax =** SELECT NOW();

| NOW() |
| --- |
| 2023-02-23 14:44:09 |

Q-29. Write an SQL query to show the top n (say 10) records of a table.

**Syntax =** SELECT * FROM WORKER ORDER BY SALARY DESC LIMIT 10;

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

Q-30. Write an SQL query to determine the nth (say n=5) highest salary from a table.

**Syntax =** SELECT Salary AS 5TH_HIGHEST_SALARY FROM Worker ORDER BY Salary DESC LIMIT 5,4;

| 5TH_HIGHEST_SALARY |
| --- |
| 100000 |
| 90000 |
| 80000 |

Q-31. Write an SQL query to determine the 5th highest salary without using TOP or limit method.

**Syntax =** SELECT SALARY FROM WORKER W1 WHERE 4 = ( SELECT COUNT( DISTINCT ( W2.SALARY ) ) FROM WORKER W2 WHERE W2.SALARY >= W1.SALARY );

| Salary |
| --- |
| 100000 |

Q-32. Write an SQL query to fetch the list of employees with the same salary.

**Syntax =** 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;

| WORKER_ID | FIRST_NAME | Salary |
|---|---|---|
| 5 | Vivek | 500000 |
| 4 | Amitabh | 500000 |

Q-33. Write an SQL query to show the second highest salary from a table.

**Syntax =** SELECT MAX(SALARY) FROM WORKER WHERE SALARY NOT IN (SELECT MAX(SALARY) FROM WORKER);

| MAX(SALARY) |
|---|
| 300000 |

Q-34. Write an SQL query to show one row twice in results from a table.

**Syntax =** SELECT FIRST_NAME,LAST_NAME,DEPARTMENT FROM WORKER W WHERE W.DEPARTMENT = 'HR' UNION ALL SELECT FIRST_NAME,LAST_NAME,DEPARTMENT FROM WORKER W1 WHERE W1.DEPARTMENT = 'HR';

| FIRST_NAME | LAST_NAME | DEPARTMENT |
|---|---|---|
| Monika | Arora | HR |
| Vishal | Singhal | HR |
| Monika | Arora | HR |
| Vishal | Singhal | HR |

Q-35 Write an SQL query to fetch the first 50% records from a table.

**Syntax =** SELECT * FROM WORKER WHERE WORKER_ID <= (SELECT COUNT(WORKER_ID)/2 FROM WORKER);

| 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 |

Q-36. Write an SQL query to fetch the departments that have less than five people in it.

**Syntax =** SELECT DEPARTMENT,COUNT(WORKER_ID) AS NO_OF_WORKERS FROM WORKER GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID)<5;

| DEPARTMENT | NO_OF_WORKERS |
|---|---|
| Account | 1 |
| HR | 2 |

Q-37. Write an SQL query to show all departments along with the number of people in there.

**Syntax =** SELECT DEPARTMENT,COUNT(WORKER_ID) AS NO_OF_WORKERS FROM WORKER GROUP BY DEPARTMENT;

| DEPARTMENT | NO_OF_WORKERS |
|---|---|
| Account | 1 |
| Admin | 5 |
| HR | 2 |

Q-38. Write an SQL query to show the last record from a table.

**Syntax =** SELECT * FROM WORKER WHERE WORKER_ID = (SELECT MAX(WORKER_ID) FROM WORKER);

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

Q-39. Write an SQL query to fetch the last five records from a table.

**Syntax =** SELECT * FROM WORKER ORDER BY WORKER_ID DESC LIMIT 5;

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

Q-40. Write an SQL query to print the name of employees having the highest salary in each department.

**Syntax =** SELECT Y.DEPARTMENT,Y.FIRST_NAME,Y.SALARY FROM (SELECT MAX(SALARY) AS TOTAL_SALARY,DEPARTMENT FROM WORKER GROUP BY DEPARTMENT) AS P INNER JOIN WORKER Y ON P.DEPARTMENT=Y.DEPARTMENT AND P.TOTAL_SALARY=Y.SALARY;

| DEPARTMENT | FIRST_NAME | SALARY |
|---|---|---|
| HR | Vishal | 300000 |
| Admin | Amitabh | 500000 |
| Admin | Vivek | 500000 |
| Account | Vipul | 200000 |

Q-41. Write an SQL query to fetch three max salaries from a table.

**Syntax =** SELECT DISTINCT SALARY FROM WORKER Y WHERE 3 >= (SELECT COUNT(DISTINCT SALARY) FROM WORKER P WHERE Y.SALARY <= P.SALARY) ORDER BY Y.SALARY DESC;

| SALARY ▼ 1 |
|---|
| 500000 |
| 300000 |
| 200000 |

Q-42. Write an SQL query to fetch three min salaries from a table.

**Syntax =** SELECT DISTINCT SALARY FROM WORKER Y WHERE 3 >= (SELECT COUNT(DISTINCT SALARY) FROM WORKER P WHERE Y.SALARY >= P.SALARY) ORDER BY Y.SALARY ASC;

| SALARY ▲ 1 |
|---|
| 75000 |
| 80000 |
| 90000 |

Q-43. Write an SQL query to fetch departments along with the total salaries paid for each of them.

**Syntax =** SELECT DEPARTMENT, SUM(SALARY) FROM WORKER GROUP BY DEPARTMENT;

| DEPARTMENT | SUM(SALARY) |
|---|---|
| Account | 200000 |
| Admin | 1245000 |
| HR | 400000 |

Q-44. Write an SQL query to fetch the names of workers who earn the highest salary.

**Syntax =** SELECT FIRST_NAME, SALARY FROM WORKER WHERE SALARY=(SELECT MAX(SALARY) FROM WORKER);

| FIRST_NAME | SALARY |
|---|---|
| Amitabh | 500000 |
| Vivek | 500000 |

Q-45. Write an SQL query to fetch the names of the workers who work as executive.

**Syntax =** SELECT WORKER.FIRST_NAME,WORKER.LAST_NAME,TITLE.WORKER_TITLE FROM WORKER INNER JOIN TITLE ON WORKER.WORKER_ID=TITLE.WORKER_REF_ID WHERE TITLE.WORKER_TITLE = "EXECUTIVE";

| FIRST_NAME | LAST_NAME | WORKER_TITLE |
|---|---|---|
| Niharika | Verma | Executive |
| Geetika | Chauhan | Executive |
| Satish | Kumar | Executive |

b

----------xxx----------xxx----------xxx----------xxx----------xxx----------