SQL Interview Questions

PART I

Q1) What is SQL?

Structured Query Language is a database tool which is used to create and access database to support software application.

Q2) What are tables in SQL?

The table is a collection of record and its information at a single view.

Q3) What are different types of statements supported by SQL?

There are 3 types of SQL statements

1) DDL (Data Definition Language): It is used to define the database structure such as tables. It includes three statements such as Create, Alter, and Drop.

2) DML (Data Manipulation Language): These statements are used to manipulate the data in records. Commonly used DML statements are Insert, Update, and Delete.

The Select statement is used as partial DML statement that is used to select all or relevant records in the table.

3) DCL (Data Control Language): These statements are used to set privileges such as Grant and Revoke database access permission to the specific user

----

Q4) How do we use DISTINCT statement? What is its use?

The DISTINCT statement is used with the SELECT statement. If the records contain duplicate values then DISTINCT is used to select different values among duplicate records.

Q5) What are different Clauses used in SQL?

WHERE Clause: This clause is used to define the condition, extract and display only those records which fulfill the given condition

- GROUP BY Clause: It is used with SELECT statement to group the result of the executed query using the value specified in it. It matches the value with the column name in tables and groups the end result accordingly.

- HAVING clause: This clause is used in association with the GROUP BY clause. It is applied to each group of result or the entire result as a single group and much similar as WHERE clause, the only difference is you cannot use it without GROUP BY clause

- ORDER BY clause: This clause is to define the order of the query output either in ascending (ASC) or in descending (DESC) order. Ascending (ASC) is the default one but descending (DESC) is set explicitly.

- USING clause: USING clause comes in use while working with SQL Joins. It is used to check equality based on columns when tables are joined. It can be used instead ON clause in Joins.

Q6) Why do we use SQL constraints? Which constraints we can use while creating database in SQL?

Constraints are used to set the rules for all records in the table. If any constraints get violated then it can abort the action that caused it.

Constraints are defined while creating the database itself with CREATE TABLE statement or even after the table is created once with ALTER TABLE statement.

There are 5 major constraints are used in SQL, such as

- NOT NULL: That indicates that the column must have some value and cannot be left null
- UNIQUE: This constraint is used to ensure that each row and column has unique value and no value is being repeated in any other row or column
- PRIMARY KEY: This constraint is used in association with NOT NULL and UNIQUE constraints such as on one or the combination of more than one columns to identify the particular record with a unique identity.
- FOREIGN KEY: It is used to ensure the referential integrity of data in the table and also matches the value in one table with another using Primary Key
- CHECK: It is used to ensure whether the value in columns fulfills the specified condition

----

Q7) What are different JOINS used in SQL?

There are 4 major types of joins made to use while working on multiple tables in SQL databases

- INNER JOIN: It is also known as SIMPLE JOIN which returns all rows from BOTH tables when it has at least one column matched

- LEFT JOIN (LEFT OUTER JOIN): This join returns all rows from a LEFT table and its matched rows from a RIGHT table.
- RIGHT JOIN (RIGHT OUTER JOIN): This joins returns all rows from the RIGHT table and its matched rows from a LEFT table

- FULL JOIN (FULL OUTER JOIN): This joins returns all when there is a match either in the RIGHT table or in the LEFT table.

Q8) What are properties of the transaction?

Properties of transaction are known as ACID properties, such as

- Atomicity: Ensures the completeness of all transactions performed. Checks whether every transaction is completed successfully if not then transaction is aborted at the failure point and the previous transaction is rolled back to its initial state as changes undone
- Consistency: Ensures that all changes made through successful transaction are reflected properly on database
- Isolation: Ensures that all transactions are performed independently and changes made by one transaction are not reflected on other
- Durability: Ensures that the changes made in database with committed transactions persist as it is even after system failure

  ----

Q9) How many Aggregate Functions are available there in SQL?

SQL Aggregate Functions calculates values from multiple columns in a table and returns a single value.

There are 7 aggregate functions we use in SQL

- AVG(): Returns the average value from specified columns
- COUNT(): Returns number of table rows

- **MAX():** Returns largest value among the records
- **MIN():** Returns smallest value among the records
- **SUM():** Returns the sum of specified column values
- **FIRST():** Returns the first value
- **LAST():** Returns Last value

Q10) What are Scalar Functions in SQL?

Scalar Functions are used to return a single value based on the input values. Scalar Functions are as follows

- **UCASE():** Converts the specified field in upper case
- **LCASE():** Converts the specified field in lower case
- **MID():** Extracts and returns character from text field
- **FORMAT():** Specifies the display format
- **LEN():** Specifies the length of text field
- **ROUND():** Rounds up the decimal field value to a number

Q11) What are triggers?

Triggers in SQL is kind of stored procedures used to create a response to a specific action performed on the table such as Insert, Update or Delete. You can invoke triggers explicitly on the table in the database.

Action and Event are two main components of SQL triggers when certain actions are performed the event occurs in response to that action.

Q12) What is View in SQL?

A View can be defined as a virtual table that contains rows and columns with fields from one or more table.

Q13) How we can update the view?

SQL CREATE and REPLACE can be used for updating the view.

Q14) What is SQL Injection?

SQL Injection is a type of database attack technique where malicious SQL statements are inserted into an entry field of database such that once it is executed the database is opened for an attacker. This technique is usually used for attacking Data-Driven Applications to have an access to sensitive data and perform administrative tasks on databases.

Q15) What is the use of NVL function?

NVL function is used to convert the null value to its actual value.

Q16) What do you mean by Subquery?

Query within another query is called as Subquery. A subquery is called inner query which returns output that is to be used by another query.

Q17) What is the difference between DELETE and TRUNCATE?

- The basic difference in both is DELETE is DML command and TRUNCATE is DDL
- DELETE is used to delete a specific row from the table whereas TRUNCATE is used to remove all rows from the table

- We can use DELETE with WHERE clause but cannot use TRUNCATE with it

----

Q18) What is the difference between DROP and TRUNCATE?

TRUNCATE removes all rows from the table which cannot be retrieved back, DROP removes the entire table from the database and it cannot be retrieved back.

Q19) What do you mean by Stored Procedures? How do we use it?

A stored procedure is a collection of SQL statements which can be used as a function to access the database. We can create these stored procedures previously before using it and can execute these them wherever we require and also apply some conditional logic to it. Stored procedures are also used to reduce network traffic and improve performance.

Q20) What is a Cursor?

A cursor is a database object which is used to manipulate data in a row-to-row manner.

Cursor follows steps as given below

- Declare Cursor
- Open Cursor
- Retrieve row from the Cursor
- Process the row
- Close Cursor
- Deallocate Cursor

Q21. What is the Primary key?

Ans. A Primary key is a column whose values uniquely identify every row in a table. Primary key values can never be reused.

Q22. What are Foreign keys?

Ans. When a one table’s primary key field is added to related tables in order to create the common field which relates the two tables, it called a foreign key in other tables.

Foreign Key constraints enforce referential integrity.

Q23. What is CHECK Constraint?

Ans. A CHECK constraint is used to limit the values or type of data that can be stored in a column. They are used to enforce domain integrity.

Q24. Is it possible for a table to have more than one foreign key?
Ans. Yes, a table can have many foreign keys and only one primary key.

Q25. What is the difference between UNIQUE and PRIMARY KEY constraints?

Ans. A table can have only one PRIMARY KEY whereas there can be any number of UNIQUE keys.
The primary key cannot contain Null values whereas Unique key can contain Null values.

Q26. What is a composite primary key?

Ans. Primary key created on more than one column is called composite primary key.

Q27. What is an Index?

Ans. An Index is a special structure associated with a table speed up the performance of queries. The index can be created on one or more columns of a table.

Q28. What is a View?

Ans. A view is a virtual table which contains data from one or more tables. Views restrict data access of the table by selecting only required values and make complex queries easy.

Q29. What are the advantages of Views?

Ans. Advantages of Views:

1. Views restrict access to the data because the view can display selective columns from the table.
2. Views can be used to make simple queries to retrieve the results of complicated queries. For example, views can be used to query information from multiple tables without the user knowing.

Q30. Does View contain Data?

Ans. No, Views are virtual structure.

Q31. Can a View based on another View?

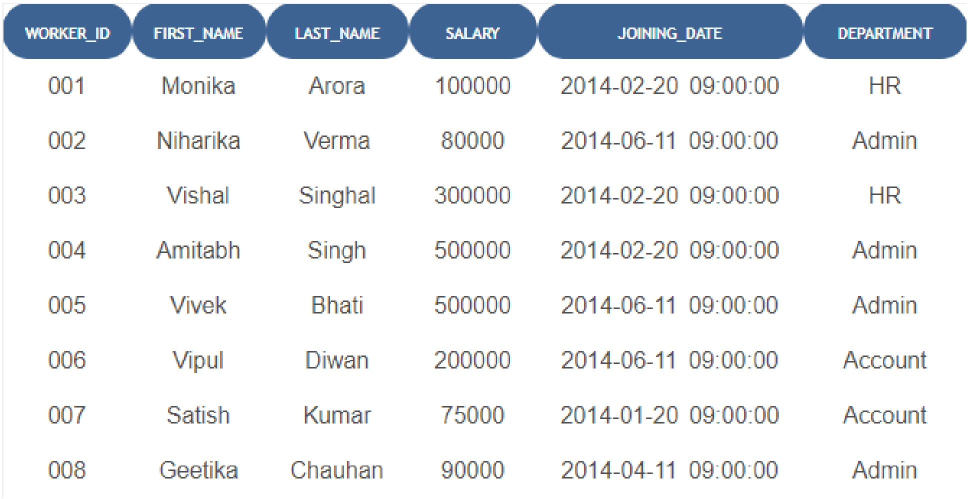
Ans. Yes, A View is based on another View.

Q32. What is the difference between Having clause and Where clause?

Ans. Both specify a search condition but Having clause is used only with the SELECT statement and typically used with GROUP BY clause. If GROUP BY clause is not used then Having behaved like WHERE clause only.

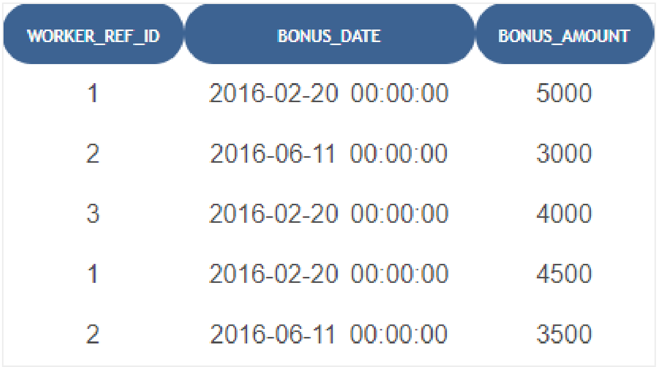
PART II

Table 1 : Worker



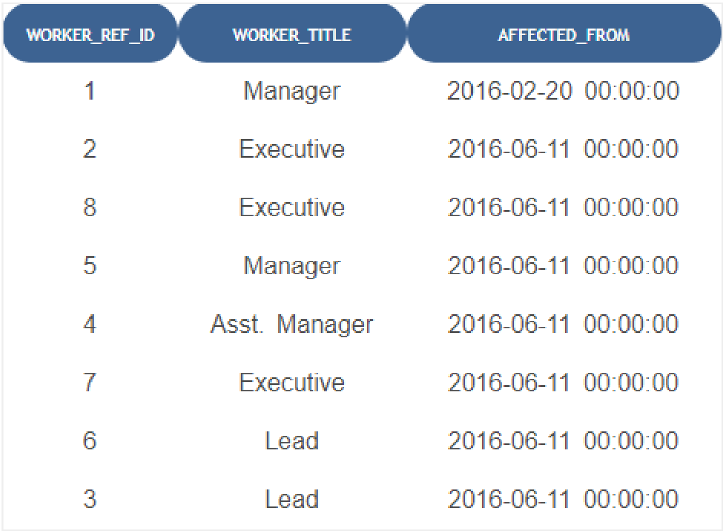
WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
001	Monika	Arora	100000	2014-02-20 09:00:00	HR
002	Niharika	Verma	80000	2014-06-11 09:00:00	Admin
003	Vishal	Singhal	300000	2014-02-20 09:00:00	HR
004	Amitabh	Singh	500000	2014-02-20 09:00:00	Admin
005	Vivek	Bhati	500000	2014-06-11 09:00:00	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

Table 2: 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

Table 3: 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

Q-1. Write a SQL Query to Fetch “FIRST_NAME” From Worker Table Using The Alias Name As <WORKER_NAME>.

Ans.
The required query is:

```
Select FIRST_NAME AS WORKER_NAME from Worker;
```

Q-2. Write A SQL Query To Fetch “FIRST_NAME” From Worker Table In Upper Case.

Ans.
The required query is:

```
Select upper(FIRST_NAME) from Worker;
```

Q-3. Write a SQL Query To Fetch Unique Values Of DEPARTMENT From Worker Table.

Ans.
The required query is:

```
Select distinct DEPARTMENT from Worker;
```

Q-4. Write a SQL Query To Print The First Three Characters Of FIRST_NAME From Worker Table.

Ans.
The required query is:

```
Select substring(FIRST_NAME,1,3) from Worker;
```

Q-5. Write a SQL Query to Find The Position Of The Alphabet (‘A’) In The First Name Column ‘Amitabh’ From Worker Table.

Ans.
The required query is:

```
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 a SQL Query To Print The FIRST_NAME From Worker Table After Removing White Spaces From The Right Side.

Ans.
The required query is:

```
Select RTRIM(FIRST_NAME) from Worker;
```

Q-7. Write a 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 a 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 a 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 a 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 a 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 a 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 a 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 a 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 a 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 a 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 a 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 a 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 a 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 a SQL Query To Fetch The Count Of Employees Working In The Department ‘Admin’.

Ans.
The required query is:

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

Q-22. Write a 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 a 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) AS No_Of_Workers
FROM worker
GROUP BY DEPARTMENT
ORDER BY No_Of_Workers DESC;
```

Q-24. Write a SQL Query To Print Details Of The Workers Who Are Also Managers.

Ans.
The required query is:

```
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 a SQL Query To Fetch Duplicate Records Having Matching Data In Some Fields Of A Table.

Ans.
The required query is:

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

Q-26. Write a SQL Query To Show Only Odd Rows From A Table.

Ans.
The required query is:

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

Q-27. Write a SQL Query To Show Only Even Rows From A Table.

Ans.
The required query is:

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

Q-28. Write a 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 a SQL Query To Fetch Intersecting Records Of Two Tables.

Ans.

The required query is:

```
(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);
```

Q-30. Write a 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 a 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 a 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 a 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 a 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 a SQL Query To Fetch The List Of Employees With The Same Salary.

Ans.
The required query is:

```
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 a SQL Query To Show The Second Highest Salary From A Table.

Ans.
The required query is:

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

Q-37. Write a SQL Query To Show One Row Twice In Results From A Table.

Ans.

The required query is:

```
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 a SQL Query To Fetch Intersecting Records Of Two Tables.

Ans.
The required query is:

```
(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM WorkerClone);
```

Q-39. Write a SQL Query To Fetch The First 50% Records From A Table.

Ans.
The required query is:

```
SELECT *
FROM WORKER
WHERE WORKER_ID <= (SELECT count(WORKER_ID)/2 from Worker);
```

Q-40. Write a SQL Query To Fetch The Departments That Have Less Than Five People In It.

Ans.
The required query is:

```
SELECT DEPARTMENT, COUNT(WORKER_ID) as 'Number of Workers' FROM Worker
GROUP BY DEPARTMENT HAVING COUNT(WORKER_ID) < 5;
```

Q-41. Write a SQL Query To Show All Departments Along With The Number Of People In There.

Ans.
The following query returns the expected result:

```
SELECT DEPARTMENT, COUNT(DEPARTMENT) as 'Number of Workers' FROM Worker
GROUP BY DEPARTMENT;
```

Q-42. Write a SQL Query To Show The Last Record From A Table.

Ans.
The following query will return the last record from the Worker table:

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

Q-43. Write a SQL Query To Fetch The First Row Of A Table.

Ans.
The required query is:

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

Q-44. Write a SQL Query To Fetch The Last Five Records From A Table.

Ans.
The required query is:

```
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 a SQL Query To Print The Name Of Employees Having The Highest Salary In Each Department.

Ans.
The required query is:

```
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 a SQL Query To Fetch Three Max Salaries From A Table.

Ans.
The required query is:

```
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 a SQL Query To Fetch Three Min Salaries From A Table.

Ans.
The required query is:

```
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 a SQL Query To Fetch Nth Max Salaries From A Table.

Ans.
The required query is:

```
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 a SQL Query to Fetch Departments Along with The Total Salaries Paid For Each Of Them.

Ans.
The required query is:

```
SELECT DEPARTMENT, sum(Salary) from worker group by DEPARTMENT;
```

Q-50. Write a SQL Query to Fetch The Names Of Workers Who Earn The Highest Salary.

Ans.
The required query is:

```
SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY)
from Worker);
```