NAME: Anurag

RegNo: 21BCI0188

Course: MongoDB Certification

Weekly Assessment

1. Fetch "FIRST NAME" from Worker table using the alias name as `<WORKER NAME>`:

```
SELECT FIRST_NAME AS "WORKER NAME" FROM Worker;
```

2. Fetch "FIRST NAME" from Worker table in upper case:

```
SELECT UPPER(FIRST_NAME) AS "FIRST NAME" FROM Worker;
```

3. Fetch unique values of DEPARTMENT from Worker table:

```
SELECT DISTINCT DEPARTMENT FROM Worker;
```

4. Print the first three characters of FIRST NAME from Worker table:

```
SELECT LEFT(FIRST_NAME, 3) FROM Worker;
```

5. Find the position of the alphabet ('a') in the first name "Amitabh" from Worker table:

```
SELECT POSITION('a' IN FIRST_NAME) FROM Worker WHERE FIRST_NAME =
'Amitabh';
```

6. Print the FIRST NAME from Worker table after removing white spaces from the right side:

```
SELECT RTRIM(FIRST_NAME) FROM Worker;
```

7. Print the DEPARTMENT from Worker table after removing white spaces from the left side:

```
SELECT LTRIM(DEPARTMENT) FROM Worker;
```

8. Fetch the unique values of DEPARTMENT from Worker table and print its length:

```
SELECT DISTINCT DEPARTMENT, LENGTH(DEPARTMENT) AS DEPT_LENGTH FROM Worker;
```

9. Print the FIRST NAME from Worker table after replacing 'a' with 'A':

```
SELECT REPLACE(FIRST_NAME, 'a', 'A') FROM Worker;
```

10. Print the FIRST NAME and LAST NAME from Worker table in a single column `COMPLETE NAME` with a space separating them:

```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS "COMPLETE NAME" FROM Worker;
```

11. Print all Worker details from the Worker table ordered by `FIRST_NAME` Ascending:

```
SELECT * FROM Worker ORDER BY FIRST_NAME ASC;
```

12. Print all Worker details from the Worker table ordered by `FIRST_NAME` Ascending and `DEPARTMENT` Descending:

```
SELECT * FROM Worker ORDER BY FIRST_NAME ASC, DEPARTMENT DESC;
```

13. Print details for Workers with the first name as "Vipul" and "Satish" from Worker table:

```
SELECT * FROM Worker WHERE FIRST_NAME IN ('Vipul', 'Satish');
```

14. Print details of workers excluding first names "Vipul" and "Satish" from Worker table:

```
SELECT * FROM Worker WHERE FIRST_NAME NOT IN ('Vipul', 'Satish');
```

15. Print details of Workers with DEPARTMENT name as "Admin":

```
SELECT * FROM Worker WHERE DEPARTMENT = 'Admin';
```

16. Print details of the Workers whose `FIRST_NAME` contains 'a':

```
SELECT * FROM Worker WHERE FIRST_NAME LIKE '%a%';
```

17. Print details of the Workers whose `FIRST_NAME` ends with 'a':

```
SELECT * FROM Worker WHERE FIRST_NAME LIKE '%a';
```

18. Print details of the Workers whose `FIRST_NAME` ends with 'h' and contains six alphabets:

```
SELECT * FROM Worker WHERE FIRST_NAME LIKE '_____h';  -- 5 underscores +
'h'
```

19. Print details of the Workers whose `SALARY` lies between 100000 and 500000:

```
SELECT * FROM Worker WHERE SALARY BETWEEN 100000 AND 500000;
```

20. Print details of the Workers who have joined in Feb 2014:

```
SELECT * FROM Worker WHERE JOINING_DATE BETWEEN '2014-02-01' AND '2014-02-
28 23:59:59';
```

21. Fetch the count of employees working in the department 'Admin':

```
SELECT COUNT(*) AS Admin_Count FROM Worker WHERE DEPARTMENT = 'Admin';
```

22. Fetch worker names with salaries between 50000 and 100000:

```
SELECT FIRST_NAME, LAST_NAME FROM Worker WHERE SALARY BETWEEN 50000 AND
100000;
```

23. Fetch the number of workers for each department in descending order:

```
SELECT DEPARTMENT, COUNT(*) AS Worker_Count FROM Worker GROUP BY DEPARTMENT
ORDER BY Worker_Count DESC;
```

24. Print details of the Workers who are also Managers:

```
SELECT w.*
FROM Worker w
JOIN Title t ON w.WORKER_ID = t.WORKER_REF_ID
WHERE t.WORKER_TITLE = 'Manager';
```

25. Fetch duplicate records having matching data in some fields of a table:

```
SELECT FIRST_NAME, LAST_NAME, COUNT(*)
FROM Worker
GROUP BY FIRST_NAME, LAST_NAME
HAVING COUNT(*) > 1;
```

26. Show only odd rows from a table: In MySQL, you can use the `MOD()` function along with `ROW_NUMBER()` or `@rownum` variable to get odd rows.

```
SELECT *
FROM (
    SELECT *, ROW_NUMBER() OVER (ORDER BY WORKER_ID) AS row_num
    FROM Worker
) AS temp
WHERE MOD(row_num, 2) = 1;
```

27. Show only even rows from a table:

```
SET @rownum := 0;

SELECT *
FROM (
    SELECT *, @rownum := @rownum + 1 AS row_num
    FROM Worker
) AS temp
WHERE MOD(row_num, 2) = 0;
```

28. Clone a new table from another table:

```
CREATE TABLE NewWorker AS SELECT * FROM Worker;
```

29. Fetch intersecting records of two tables:

```
SELECT * FROM Worker
INTERSECT
SELECT * FROM AnotherTable;
```

Note: `INTERSECT` is supported by some SQL databases like PostgreSQL, SQL Server, but not directly in MySQL. For MySQL, you can use:

```
SELECT * FROM Worker w
JOIN AnotherTable a ON w.WORKER_ID = a.WORKER_ID;
```

30. Show records from one table that another table does not have:

```
SELECT * FROM Worker w
WHERE NOT EXISTS (SELECT 1 FROM AnotherTable a WHERE w.WORKER_ID =
a.WORKER_ID);
```

31. Show the current date and time:

```
SELECT NOW();
```

32. Show the top n (say 10) records of a table:

```
SELECT * FROM Worker
ORDER BY WORKER_ID
LIMIT 10;
```

33. Determine the nth (say 5th) highest salary from a table:

```
SELECT DISTINCT SALARY
FROM Worker
ORDER BY SALARY DESC
LIMIT 1 OFFSET 4;
```

34. Determine the 5th highest salary without using `TOP` or `LIMIT` method:

```
SELECT SALARY FROM Worker w1
WHERE 4 = (SELECT COUNT(DISTINCT SALARY) FROM Worker w2 WHERE w2.SALARY >
w1.SALARY);
```

35. Fetch the list of employees with the same salary:

```
SELECT SALARY, GROUP_CONCAT(FIRST_NAME, ' ', LAST_NAME) AS EMPLOYEES
FROM Worker
GROUP BY SALARY
HAVING COUNT(*) > 1;
```

36. Show the second highest salary from a table:

```
SELECT MAX(SALARY) FROM Worker
WHERE SALARY < (SELECT MAX(SALARY) FROM Worker);
```

37. Show one row twice in results from a table:

```
SELECT * FROM Worker
UNION ALL
SELECT * FROM Worker
WHERE WORKER_ID = 1
LIMIT 1;
```

38. Fetch intersecting records of two tables:

```
SELECT * FROM Worker w
JOIN AnotherTable a ON w.WORKER_ID = a.WORKER_ID;
```

39. Fetch the first 50% records from a table:

```
SELECT * FROM Worker
ORDER BY WORKER_ID
LIMIT (SELECT COUNT(*)/2 FROM Worker);
```

40. Fetch the departments that have less than five people in it:

```
SELECT DEPARTMENT
FROM Worker
GROUP BY DEPARTMENT
HAVING COUNT(*) < 5;
```

41. Show all departments along with the number of people in them:

```
SELECT DEPARTMENT, COUNT(*) AS NumberOfPeople
FROM Worker
GROUP BY DEPARTMENT;
```

42. Show the last record from a table:

```
SELECT * FROM Worker
ORDER BY WORKER_ID DESC
LIMIT 1;
```

43. Fetch the first row of a table:

```
SELECT * FROM Worker
ORDER BY WORKER_ID
LIMIT 1;
```

44. Fetch the last five records from a table:

```
SELECT * FROM Worker
ORDER BY WORKER_ID DESC
LIMIT 5;
```

45. Print the name of employees having the highest salary in each department:

```
SELECT DEPARTMENT, FIRST_NAME, LAST_NAME, SALARY
FROM Worker w
WHERE SALARY = (SELECT MAX(SALARY) FROM Worker WHERE DEPARTMENT =
w.DEPARTMENT);
```

46. Fetch three max salaries from a table:

```
SELECT DISTINCT SALARY
FROM Worker
ORDER BY SALARY DESC
LIMIT 3;
```

47. Fetch three min salaries from a table:

```
SELECT DISTINCT SALARY
FROM Worker
ORDER BY SALARY
LIMIT 3;
```

48. Fetch nth max salaries from a table:

```
SELECT DISTINCT SALARY
FROM Worker
ORDER BY SALARY DESC
LIMIT 1 OFFSET (n-1);  -- Replace n with the desired rank, for example,
OFFSET 2 for the 3rd highest salary
```

49. Fetch departments along with the total salaries paid for each of them:

```
SELECT DEPARTMENT, SUM(SALARY) AS TotalSalaries
FROM Worker
GROUP BY DEPARTMENT;
```

50. Fetch the names of workers who earn the highest salary:

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