

# **ASSIGNMENT**

Vinesh Kumar M V

281878

-- Create Worker table

--CREATE TABLE Worker (

-- WORKER\_ID INT IDENTITY(1,1) PRIMARY KEY,

-- FIRST\_NAME CHAR(25),

-- LAST\_NAME CHAR(25),

-- SALARY INT,

-- JOINING\_DATE DATETIME,

-- DEPARTMENT CHAR(25)

--);

---- Insert data into Worker table

--INSERT INTO Worker (FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE,  
DEPARTMENT)

--VALUES

-- ('Monika', 'Arora', 100000, '2020-02-14 09:00:00', 'HR'),

-- ('Niharika', 'Verma', 80000, '2011-06-14 09:00:00', 'Admin'),

-- ('Vishal', 'Singhal', 300000, '2020-02-14 09:00:00', 'HR'),

-- ('Amitabh', 'Singh', 500000, '2020-02-14 09:00:00', 'Admin'),

-- ('Vivek', 'Bhati', 500000, '2011-06-14 09:00:00', 'Admin'),

-- ('Vipul', 'Diwan', 200000, '2011-06-14 09:00:00', 'Account'),

-- ('Satish', 'Kumar', 75000, '2020-01-14 09:00:00', 'Account'),

-- ('Geetika', 'Chauhan', 90000, '2011-04-14 09:00:00', 'Admin');

---- Create Bonus table

--CREATE TABLE Bonus (

```
-- BONUS_ID INT IDENTITY(1,1) PRIMARY KEY,
-- WORKER_REF_ID INT,
-- BONUS_AMOUNT INT,
-- BONUS_DATE DATETIME,
-- FOREIGN KEY (WORKER_REF_ID) REFERENCES Worker(WORKER_ID) ON DELETE
CASCADE
--);
```

---- Insert data into Bonus table

```
--INSERT INTO Bonus (WORKER_REF_ID, BONUS_AMOUNT, BONUS_DATE)
--VALUES
-- (1, 5000, '2020-02-16'),
-- (2, 3000, '2011-06-16'),
-- (3, 4000, '2020-02-16'),
-- (1, 4500, '2020-02-16'),
-- (2, 3500, '2011-06-16');
```

---- Create Title table

```
--CREATE TABLE Title (
-- TITLE_ID INT IDENTITY(1,1) PRIMARY KEY,
-- WORKER_REF_ID INT,
-- WORKER_TITLE CHAR(25),
-- AFFECTED_FROM DATETIME,
-- FOREIGN KEY (WORKER_REF_ID) REFERENCES Worker(WORKER_ID) ON DELETE
CASCADE
--);
```

---- Insert data into Title table

--INSERT INTO Title (WORKER\_REF\_ID, WORKER\_TITLE, AFFECTED\_FROM)

--VALUES

-- (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');

select \* from WORKER;

/\*----- TASK -----\*/

--1

select upper(FIRST\_NAME) FIRST\_NAME from worker;

--2

select distinct DEPARTMENT from worker;

--3

SELECT LEFT(FIRST\_NAME,3) First\_three\_characters FROM WORKER;

--4

SELECT CHARINDEX('a',FIRST\_NAME) Index\_of\_A FROM WORKER WHERE FIRST\_NAME  
= 'Amitabh';

--5

```
SELECT DISTINCT DEPARTMENT, LEN(DEPARTMENT) DEP_LENGTH FROM WORKER;
```

--6

```
SELECT * FROM WORKER ORDER BY FIRST_NAME, DEPARTMENT DESC;
```

--7

```
SELECT * FROM worker WHERE FIRST_NAME in('Vipul','Satish');
```

--8

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

--9

```
SELECT * FROM WORKER WHERE FIRST_NAME LIKE '____H';
```

--10

```
SELECT * FROM WORKER WHERE SALARY BETWEEN 100000 and 500000;
```

--11

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

--12

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

--13

```
SELECT DEPARTMENT, COUNT(*) NUMBER FROM WORKER  
GROUP BY DEPARTMENT ORDER BY DEPARTMENT DESC;
```

--14

```
SELECT W.* FROM WORKER W JOIN TITLE T
ON W.WORKER_ID = T.WORKER_REF_ID
WHERE T.WORKER_TITLE = 'MANAGER';
```

--15

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

--16

```
SELECT
W.WORKER_ID,W.FIRST_NAME,W.LAST_NAME,W.DEPARTMENT,B.BONUS_AMOUNT
FROM WORKER W JOIN BONUS B
ON W.WORKER_ID = B.WORKER_REF_ID
```

--17

```
SELECT W.* FROM WORKER W
LEFT JOIN BONUS B ON W.WORKER_ID = B.WORKER_REF_ID
WHERE WORKER_REF_ID IS NULL
```

--18

```
SELECT DISTINCT TOP(2) SALARY FROM WORKER ORDER BY SALARY DESC
```

--19

```
SELECT * FROM (SELECT *, ROW_NUMBER() OVER (ORDER BY SALARY DESC) RANKED
FROM WORKER) WORKERS
WHERE RANKED = 2
```

--20

```
SELECT A.FIRST_NAME WORKER_1,B.FIRST_NAME WORKER_2,A.SALARY
FROM WORKER A, WORKER B
WHERE A.SALARY = B.SALARY AND A.WORKER_ID < B.WORKER_ID
```

--21

```
WITH Ordered AS (
    SELECT *,
        ROW_NUMBER() OVER (ORDER BY WORKER_ID) AS RowNum,
        COUNT(*) OVER () AS TotalCount
    FROM Worker
)
SELECT *
FROM Ordered
WHERE RowNum <= CEILING(TotalCount / 2.0);
```

--22

```
SELECT
    DEPARTMENT,
    COUNT(*) AS NumEmployees
FROM Worker
GROUP BY DEPARTMENT
HAVING COUNT(*) > 3;
```

--23

```
SELECT *
FROM (
    SELECT *, ROW_NUMBER() OVER (ORDER BY WORKER_ID) AS RowAsc,
           ROW_NUMBER() OVER (ORDER BY WORKER_ID DESC) AS RowDesc
    FROM Worker
) AS T
WHERE RowAsc = 1 OR RowDesc = 1;
```

--24

```
SELECT TOP 5 * FROM Worker ORDER BY WORKER_ID DESC
```

--25

```
SELECT *
FROM (
    SELECT *,
           ROW_NUMBER() OVER (PARTITION BY DEPARTMENT ORDER BY SALARY DESC) AS
RowNum
    FROM Worker
) AS T
WHERE RowNum = 1;
```

--26

```
SELECT DEPARTMENT,SUM(SALARY)from Worker GROUP BY DEPARTMENT
```

--27

```
SELECT * FROM Worker WHERE SALARY = (SELECT MAX(SALARY) FROM Worker)
```