Muhammad Shas K T
EMP ID: 281775

SQL SERVER : ASSIGNMENT

- Create three tables (worker, bonus and title) and workout below 27 tasks

Solutions

- Table creation and data insertion

```
CREATE TABLE worker (
    WORKER_ID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FIRST_NAME CHAR(25),
    LAST_NAME CHAR(25),
    SALARY INT,
    JOINING_DATE DATETIME,
    DEPARTMENT CHAR(25)
);

CREATE TABLE bonus (
    BONUS_ID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    WORKER_REF_ID INT,
    BONUS_AMOUNT INT,
    BONUS_DATE DATETIME,
    FOREIGN KEY (WORKER_REF_ID) REFERENCES Worker(WORKER_ID) ON DELETE
CASCADE
);

CREATE TABLE title (
    TITLE_ID INT IDENTITY(1,1) NOT NULL 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
);
```

Data insertion

```
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');
```

```
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');

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');
```

Tasks

```
-- 1.Write a query to display all the first_name  in upper case
SELECT UPPER(FIRST_NAME) as 'First Name (upper)' FROM Worker

-- 2.Write a querty to display unique department from workers table
SELECT DISTINCT DEPARTMENT as 'Unique Departments' FROM Worker

-- 3.Write an SQL query to print the first three characters of FIRST_NAME from Worker table
SELECT SUBSTRING(FIRST_NAME,1,3) 'FIRST NAME [3]' FROM worker

-- 4.Write an SQL query to find the position of the alphabet (‘a’) in the first name column ‘Amitabh’
from Worker table.
SELECT CHARINDEX('a',FIRST_NAME COLLATE SQL_Latin1_General_CP1_CS_AS) as
'Position_of_a' FROM worker WHERE FIRST_NAME='Amitabh'

-- 5.Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and
prints its length
SELECT DISTINCT DEPARTMENT as 'Unique Departments', LEN(DEPARTMENT) as 'Length of
Dept' FROM worker

-- 6.Write an SQL query to print all Worker details from the Worker table order by FIRST_NAME
Ascending and DEPARTMENT Descending
SELECT * FROM worker ORDER BY FIRST_NAME ASC, DEPARTMENT DESC

-- 7.Write a query to get workers whose name are Vipul and Satish
SELECT * FROM worker WHERE FIRST_NAME='Vipul' or FIRST_NAME='Satish'

-- 8.Write an SQL query to print details of the Workers whose FIRST_NAME contains 'a'
SELECT * FROM worker WHERE FIRST_NAME LIKE '%a%'

-- 9.Write an SQL query to print details of the Workers whose FIRST_NAME ends with ‘h’ and
contains six alphabets
SELECT * FROM worker WHERE FIRST_NAME LIKE '_____h'
```

-- 10.Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000
SELECT * FROM worker WHERE SALARY BETWEEN 100000 and 500000

-- 11.Write an SQL query to print details of the Workers who have joined in Feb’2014
SELECT * FROM worker WHERE JOINING_DATE >= '2014-02-01' AND JOINING_DATE < '2014-03-01'

-- 12.Write an SQL query to fetch the count of employees working in the department ‘Admin’
SELECT COUNT(*) as 'EMPLOYEES COUNT IN ADMIN' FROM worker WHERE DEPARTMENT='Admin'

-- 13.Write an SQL query to fetch the no. of workers for each department in the descending order
SELECT DEPARTMENT, COUNT(*) as 'NUMBER OF WORKERS' FROM worker GROUP BY DEPARTMENT ORDER BY COUNT(*) DESC

-- 14.Write a query to display workers who are managers
SELECT w.FIRST_NAME as 'Workers who are managers' FROM worker w INNER JOIN title t ON w.WORKER_ID=t.WORKER_REF_ID WHERE t.WORKER_TITLE='Manager'

-- 15.Write query to find duplicate rows title table
SELECT WORKER_REF_ID, WORKER_TITLE, AFFECTED_FROM, COUNT(*) as 'Number of duplicates' FROM title GROUP BY WORKER_REF_ID, WORKER_TITLE, AFFECTED_FROM HAVING COUNT(*) >= 2

-- 16.Write an SQL query to show all workers who got the bonus along with bonus amount
SELECT w.FIRST_NAME as 'WORKERS WHO GOT BONUS_AMOUND', SUM(b.BONUS_AMOUNT) as 'BONUS_RECIEVED' FROM worker w INNER JOIN bonus b ON w.WORKER_ID=b.WORKER_REF_ID GROUP BY w.FIRST_NAME

-- 17.Write a query to find employees in worker table that do not exist in bonus table (ie did not get bonus)
SELECT FIRST_NAME as 'WORKERS DID NOT GET BONUS' from worker WHERE WORKER_ID NOT IN (SELECT WORKER_REF_ID FROM bonus)

-- 18.Write a query to find the highest 2 salaries
SELECT DISTINCT TOP(2) SALARY as 'HIGHEST TWO SALARIES' FROM worker ORDER BY SALARY DESC

-- 19.Find 2nd highest without using TOP or LIMIT
SELECT MAX(SALARY) as 'SECOND HIGHEST SALARY' FROM worker WHERE SALARY NOT IN (SELECT MAX(SALARY) FROM worker)

-- 20.Find people who have the same salary
SELECT FIRST_NAME as 'PEOPLE WHO HAVE SAME SALARY', SALARY FROM worker WHERE SALARY IN (SELECT SALARY FROM worker GROUP BY SALARY HAVING COUNT(*)>1) ORDER BY SALARY

-- 21.Write a query to fetch 1st 50% records without using Top
SELECT * FROM (SELECT *, ROW_NUMBER() OVER (ORDER BY WORKER_ID) AS RowNum FROM worker) AS RankedWorkers WHERE RowNum <= (SELECT COUNT(*) / 2 FROM worker)

```
-- 22.Write a query to select a department with more than 3 people in worker table
SELECT DEPARTMENT, COUNT(*) as 'Number of peoples' FROM worker GROUP BY
DEPARTMENT HAVING COUNT(*)>3

-- 23.Write a query to select 1st and last row of a worker table
SELECT * FROM (SELECT *, ROW_NUMBER() OVER (ORDER BY WORKER_ID ASC) AS RowAsc
FROM worker) AS FirstWorker WHERE RowAsc = 1 UNION
SELECT * FROM (SELECT *, ROW_NUMBER() OVER (ORDER BY WORKER_ID DESC) AS
RowDesc FROM worker) AS LastWorker WHERE RowDesc = 1

-- 24.Write a query to select last 5 entries from worker table
SELECT TOP(5) * FROM worker ORDER BY WORKER_ID DESC

-- 25.Write a query to select people with highest salary in each group
SELECT w.FIRST_NAME as 'NAME', w.DEPARTMENT, w.SALARY FROM worker w INNER
JOIN(SELECT MAX(SALARY) AS SAL, DEPARTMENT FROM worker GROUP BY DEPARTMENT)
AS max_table ON w.DEPARTMENT = max_table.DEPARTMENT AND w.SALARY = max_table.SAL

-- 26.Write a query to fetch departments along with the total salaries paid for each of them
SELECT DEPARTMENT, SUM(SALARY) as 'TOTAL SALARY' FROM worker GROUP BY
DEPARTMENT

-- 27.Write a query to fetch the names of workers who earn the highest salary
SELECT FIRST_NAME as 'WORKERS WHO EARN HIGHEST SALARY', SALARY FROM worker
WHERE SALARY IN (SELECT MAX(SALARY) FROM worker)
```