**Sybase Assignment**

1. Write a query in SQL to list the employees who joined in the month having second char is 'A' (Month Name Second Char i.e. January).

SYBASE QUERY:

SELECT \*   
FROM   employees   
WHERE  *CONVERT*(*CHAR*(12), hire\_date, 0) LIKE '\_a%'

OR

SELECT \*   
FROM   employees   
WHERE  *CONVERT*(*CHAR*(12), hire\_date) LIKE '\_a%'

ORACLE QUERY:

SELECT \*   
FROM   employees   
WHERE  **to\_char**(hire\_date, 'mon') LIKE '\_a%'

1. Write a query in SQL to list the employees whose name is ending with 'S' and six characters long.

SYBASE QUERY:

SELECT emp\_name   
FROM   employees   
WHERE  emp\_name LIKE '%S'   
       AND *len*(emp\_name) = 6

OR

SELECT first\_name   
FROM   employees   
WHERE  first\_name LIKE '%S'   
       AND *len*(first\_name) = 6

ORACLE QUERY:

SELECT emp\_name   
FROM   employees   
WHERE  emp\_name LIKE '%S'   
       AND **length**(emp\_name) = 6

OR

SELECT first\_name   
FROM   employees   
WHERE  first\_name LIKE '%S'   
       AND **length**(first\_name) = 6

1. Write a query in SQL to list the employees first name + last name separated by a Blank Char.

SYBASE QUERY:

SELECT fullname = ( first\_name + ' ' + last\_name )   
FROM   employees

OR

SELECT ( first\_name + ' ' + last\_name ) AS fullname   
FROM   employees

ORACLE QUERY:

SELECT ( first\_name   
         || ' '   
         || last\_name ) AS fullname   
FROM   employees

1. Write a query in SQL to display the unique designations for the employees

SYBASE QUERY:

SELECT DISTINCT designation   
FROM   employees

ORACLE QUERY:

SELECT DISTINCT designation   
FROM   employees

1. Write a query in SQL to list the employees with Hire date in the format like February 22, 1991.

SYBASE QUERY:

SELECT employee\_id,   
       first\_name,   
       last\_name,   
       email,   
       phone\_number,   
       salary,   
       manager\_id,   
       department\_id,   
       *CONVERT*(*VARCHAR*(12), *getdate*(), 107)   
FROM   employees

ORACLE QUERY:

SELECT employee\_id,   
       first\_name,   
       last\_name,   
       email,   
       phone\_number,   
       salary,   
       manager\_id,   
       department\_id,   
       **To\_char**(hire\_date, 'MONTH DD,YYYY') AS hiredate   
FROM   employees

1. Write a query in SQL to count the no. of characters without considering the spaces for each name

SYBASE QUERY:

SELECT first\_name, last\_name, *Len*(*Ltrim*(*Rtrim*(first\_name))   
           + *Ltrim*(*Rtrim*(last\_name))) AS 'Number of Characters in Name'   
FROM   employees

OR

SELECT first\_name, last\_name, *Len*(*Ltrim*(*Rtrim*(emp\_name))) AS 'Number of Characters in Name'   
FROM   employees

MYSQL QUERY:

SELECT **Length**(**Trim**(emp\_name))   
FROM   employees

1. Write a query in SQL to list the employees who does not belong to department “Administration”

SYBASE QUERY:

(If data is coming from a single table ‘employees’)

SELECT \*   
FROM   employees   
WHERE  department\_name NOT IN ( 'Administration' )

OR

(If data is coming from both employees and departments tables we need to join them)

SELECT \*   
FROM   employees e,   
       departments d   
WHERE  e.department\_id = d.department\_id   
HAVING NOT d.department\_name = 'Administration'   
ORDER  BY e.department\_id ASC

1. Write a query in SQL to list the employees who joined before 1991

SYBASE QUERY:

SELECT \*   
FROM   employees   
WHERE  hire\_date < '1/1/1991'

1. Write a query in SQL to display the average salaries of all the employees who works as Marketing Manager

SYBASE QUERY:

SELECT *avg*(salary)   
FROM   employees   
WHERE  job\_name = 'Marketing Manager'

1. Write a query in SQL to list the employees whose salary is more than 3000 after giving 25% increment

SYBASE QUERY:

SELECT \*   
FROM   employees   
WHERE  ( 1.25 \* salary ) > 3000

1. Write a query in SQL to list the name of employees and their manager separated by the string 'works for'

SYBASE QUERY:

(If employee name is coming from just one column ‘emp\_name’)

SELECT e.emp\_name + ' works for ' + m.emp\_name   
FROM   employees e,   
       employees m   
WHERE  e.manager\_id = m.employee\_id

OR

(If employee name is coming from two columns ‘first\_name’ and ‘last\_name’)

SELECT ( e.first\_name + ' ' + e.last\_name ) + ' works for ' + (   
              m.first\_name + ' ' + m.last\_name )   
FROM   employees e,   
       employees m   
WHERE  e.manager\_id = m.employee\_id

1. List of employees who are getting highest and second highest salary in each department.

ORACLE QUERY:

SELECT DISTINCT \*   
FROM   (SELECT department\_id,   
               salary,   
               **Dense\_rank**()   
                 OVER (   
                   partition BY department\_id   
                   ORDER BY salary DESC) rank   
        FROM   employees)   
WHERE  rank <= 2   
ORDER  BY department\_id,   
          rank

1. Write a query using cursor and increase all employee salary with 20% who are belonging to Administrator department.

SYBASE QUERY:

DROP EXISTING PROCEDURE AND DECLARATION OF PROCEDURE:

IF EXISTS (SELECT 1   
           FROM   sysobjects   
           WHERE  NAME = 'sp\_INCREASE\_SALARY'   
                  AND type = 'P')   
  BEGIN   
      DROP PROCEDURE sp\_increase\_salary   
  END   
  
go

CREATE PROCEDURE **Sp\_increase\_salary**   
AS   
    DECLARE @eemployee\_id     *INT*,   
            @efirst\_name      *VARCHAR*(20),   
            @elast\_name       *VARCHAR*(20),   
            @edepartment\_id   *INT*,   
            @ddepartment\_name *VARCHAR*(20)   
  
  BEGIN   
      DECLARE cursor1 CURSOR FOR   
        SELECT e.employee\_id,   
               e.first\_name,   
               e.last\_name,   
               e.department\_id,   
               d.department\_name   
        FROM   employees e   
               INNER JOIN departments d   
                       ON e.department\_id = d.department\_id   
        WHERE  d.department\_name = 'Administration'   
  
      OPEN cursor1   
  
      FETCH cursor1 INTO @eemployee\_id, @efirst\_name, @elast\_name,   
      @edepartment\_id   
      , @ddepartment\_name   
  
      WHILE ( @@sqlstatus = 0 )   
        BEGIN   
            UPDATE employees   
            SET    salary = salary \* 1.2   
            WHERE  department\_id = @edepartment\_id   
  
            FETCH cursor1 INTO @eemployee\_id, @efirst\_name, @elast\_name,   
            @edepartment\_id   
            , @ddepartment\_name   
        END   
  
      CLOSE cursor1   
  
      DEALLOCATE cursor1   
  END   
  
go

EXECUTING PROCEDURE:

EXEC **Sp\_increase\_salary**

VALIDATION:

SELECT \*   
FROM   employees e   
       INNER JOIN departments d   
               ON e.department\_id = d.department\_id   
WHERE  d.department\_name = 'Administration'

1. List all employee whose manager department is not same department as he/she belong to.

SYBASE QUERY:

SELECT e1.employee\_id                         AS manager\_emp\_id,   
       ( e1.first\_name + ' ' + e1.last\_name ) AS manager\_full\_name,   
       e2.employee\_id                         AS employee\_emp\_id,   
       ( e2.first\_name + ' ' + e2.last\_name ) AS employee\_full\_name   
FROM   employees e1   
       INNER JOIN employees e2   
               ON e1.employee\_id = e2.manager\_id   
WHERE  e1.department\_id != e2.department\_id   
ORDER  BY manager\_emp\_id,   
          employee\_emp\_id

ORACLE QUERY:

SELECT e1.employee\_id      AS manager\_emp\_id,   
       ( e1.first\_name   
         || ' '   
         || e1.last\_name ) AS manager\_full\_name,   
       e2.employee\_id      AS employee\_emp\_id,   
       ( e2.first\_name   
         || ' '   
         || e2.last\_name ) AS employee\_full\_name   
FROM   employees e1   
       INNER JOIN employees e2   
               ON e1.employee\_id = e2.manager\_id   
WHERE  e1.department\_id != e2.department\_id   
ORDER  BY manager\_emp\_id,   
          employee\_emp\_id

1. List all employees who are having same first name and display count.

SYBASE QUERY:

SELECT first\_name,   
       *count*(\*) AS 'duplicate\_name\_count'   
FROM   employees   
GROUP  BY first\_name   
HAVING *count*(\*) > 1   
ORDER  BY first\_name

16. List of the employee with Salary scale as below.

1000- 5000 – Scale -1

5001- 7500 – Scale -2

7501- 10000 – Scale -3

10001 - 15000 – Scale -4

15001 - 20000 – Scale -5

All above 20001– Scale -6

SYBASE QUERY:

1. USING OPERATORS >= AND <= :

SELECT employee\_id,   
       first\_name,   
       last\_name,   
       salary,   
       ( CASE   
           WHEN salary >= 1000   
                AND salary <= 5000 THEN 'Scale-1'   
           WHEN salary >= 5001   
                AND salary <= 7500 THEN 'Scale-2'   
           WHEN salary >= 7501   
                AND salary <= 10000 THEN 'Scale-3'   
           WHEN salary >= 10001   
                AND salary <= 15000 THEN 'Scale-4'   
           WHEN salary >= 15001   
                AND salary <= 20000 THEN 'Scale-5'   
           WHEN salary >= 20001 THEN 'Scale-6'   
           ELSE 'Out Of Scale'   
         END ) AS ScaleType   
FROM   employees

OR

1. USING BETWEEN

SELECT employee\_id,   
       first\_name,   
       last\_name,   
       salary,   
       ( CASE   
           WHEN salary BETWEEN 1000 AND 5000 THEN 'Scale-1'   
           WHEN salary BETWEEN 5001 AND 7500 THEN 'Scale-2'   
           WHEN salary BETWEEN 7501 AND 10000 THEN 'Scale-3'   
           WHEN salary BETWEEN 10001 AND 15000 THEN 'Scale-4'   
           WHEN salary BETWEEN 15001 AND 20000 THEN 'Scale-5'   
           WHEN salary >= 20001 THEN 'Scale-6'   
           ELSE 'Out Of Scale'   
         END ) AS ScaleType   
FROM   employees

1. List all employee names along with number of characters used in their names using function.

SYBASE QUERY:

DEFINING FUNCTION IN SYBASE:

CREATE FUNCTION **Fullnamelength**(@first\_name *CHAR*(40),   
                               @last\_name  *CHAR*(40))   
returns *INT*   
AS   
  BEGIN   
      DECLARE @full\_name\_length *INT*   
  
      SET @full\_name\_length = *Len*(*Ltrim*(*Rtrim*(@first\_name))   
                                  + *Ltrim*(*Rtrim*(@last\_name)))   
  
      RETURN @full\_name\_length   
  END

CALLING FUNCTION IN SYBASE:

SELECT ( first\_name + ' ' + last\_name )          AS 'Employee Full Name',   
       dbo.**Fullnamelength**(first\_name, last\_name) AS 'Employee Full Name Length'   
FROM   employees

1. Display all month’s names from today to previous 12 months back.

SYBASE QUERY:

SELECT *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, 0, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -1, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -2, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -3, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -4, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -5, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -6, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -7, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -8, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -9, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -10, *Getdate*()), 103)),   
       *Datename*(month, *CONVERT*(*DATE*, *Dateadd*(month, -11, *Getdate*()), 103))

1. From the above question, also display first and last day (Name of the day i.e. Monday, Tuesday..etc) of the all 12 months.

SYBASE QUERY:

DECLARE @today *DATETIME*   
  
SELECT @today = *Getdate*()   
  
SELECT *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, 0, *Dateadd*(dd, -( *Day*(   
                                                        @today) - 1   
                                                         ),   
                                                        @today))   
                           ,   
                    101))                                                 AS   
       first\_day\_june\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -1, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_may\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -2, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_april\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -3, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_march\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -4, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_feb\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -5, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_jan\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -6, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_dec\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -7, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_nov\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -8, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_oct\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -9, *Dateadd*(dd, -( *Day*(   
                                                         @today) - 1   
                                                                      ), @today)   
                                         ),   
                    101))                                                 AS   
       first\_day\_sep\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -10, *Dateadd*(dd, -(   
                                                          *Day*(@today) - 1 ),   
                                                          @today))   
                    , 101))                                               AS   
       first\_day\_aug\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -11, *Dateadd*(dd, -(   
                                                          *Day*(@today) - 1 ),   
                                                          @today))   
                    , 101))                                               AS   
       first\_day\_jul\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, 1, *Dateadd*(dd, -*Day*(@today)   
                                                        ,   
                                                        @today)), 101))   AS   
       last\_day\_june\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, 0, *Dateadd*(dd, -*Day*(@today)   
                                                        ,   
                                                        @today)), 101))   AS   
       last\_day\_may\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -1, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_april\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -2, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_march\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -3, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_feb\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -4, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_jan\_2020,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -5, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_dec\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -6, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_nov\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -7, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_oct\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -8, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_sept\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -9, *Dateadd*(dd, -   
                                                         *Day*(@today),   
                                                         @today)), 101))  AS   
       last\_day\_aug\_2019,   
       *Datename*(dw, *CONVERT*(*VARCHAR*(10), *Dateadd*(mm, -10, *Dateadd*(dd, -*Day*(   
                                                          @today),   
                                                          @today)), 101)) AS   
       last\_day\_jul\_2019