**SQL Job Preparation Assignment 4**

**1.EMP TABLE'S 3 DYNAMIC COLUMNS SHOULD BE DISPLAYED. Instead of "Select EMP, SAL from EMP," your application should prompt you to choose the column you want to select at runtime in SQL.**

DECLARE @column1 NVARCHAR(50) = 'EMPNO'

DECLARE @column2 NVARCHAR(50) = 'ENAME'

DECLARE @column3 NVARCHAR(50) = 'SAL'

DECLARE @query NVARCHAR(MAX) = 'SELECT ' + @column1 + ', ' + @column2 + ', ' + @column3 + ' FROM EMP'

EXECUTE sp\_executesql @query

This code prompts the user to input the column names and stores them in variables @column1, @column2 and @column3. And then use these variables to construct the query.

We should keep in mind that you should validate the user input and check if the user inputs are actual column names.Also, you should also handle the cases where the column does not exist in the table or the user inputs an incorrect column name.

-----------------------------------------------------------------------------------------------------------------------

**2. Write a query to retrieve the data, and column names should be given at Runtime. Sort the query based on the first column. Note: As we are hardcoding the name of the column, we cannot use "Order By EMP" as the Column name is not yet selected**

DECLARE @column1 NVARCHAR(50) = 'EMPNO'

DECLARE @column2 NVARCHAR(50) = 'ENAME'

DECLARE @column3 NVARCHAR(50) = 'SAL'

DECLARE @sort\_column NVARCHAR(50) = @column1

DECLARE @query NVARCHAR(MAX) = 'SELECT ' + @column1 + ', ' + @column2 + ', ' + @column3 + ' FROM EMP ORDER BY '+ @sort\_column

EXECUTE sp\_executesql @query

This code prompts the user to input the column names and stores them in variables @column1, @column2 and @column3. Also, it prompts to input the sort column and stores in the variable @sort\_column. And then use these variables to construct the query and sort the result based on the selected column.Keep in mind that you should validate the user input and check if the user inputs are actual column names.Also, you should also handle the cases where the column does not exist in the table or the user inputs an incorrect column name.

**3. WHAT IS THE DIFF BETWEEN "&" AND "&&" IN SQL**

In SQL, "&" and "&&" have different meanings and are used in different contexts.

"&" is a bitwise operator, which is used to perform bitwise operations on integers. This operator compares each bit of the first operand to the corresponding bit of the second operand, and returns a new integer with the bits set to 1 where both operands had bits set to 1. For example:

5 & 3 = 1

This operation compares the binary representation of 5 (101) and 3 (011) and returns the result 001.

"&&" is a logical operator, which is used to perform logical AND operations on boolean expressions. This operator returns true if and only if both operands are true. For example:

(x > 0) && (x < 10)

This expression returns true if x is greater than 0 and less than 10.

So, "&" is used for bitwise operations, and "&&" is used for logical operations. It's important to use the correct operator for the intended purpose, as using the wrong operator can lead to unexpected results.

**4. Write a query to demonstrate '&&' and '&' in a single statement.**

Here's an example of a query that demonstrates the use of both the "&&" and "&" operators in a single statement:

SELECT ename, sal

FROM emp

WHERE (sal > 1000 && sal < 2000) AND (job = 'MANAGER' & deptno = 20)

In this query:

The first part of the WHERE clause, (sal > 1000 && sal < 2000), uses the "&&" operator to check that the sal is greater than 1000 AND less than 2000.

The second part of the WHERE clause, (job = 'MANAGER' & deptno = 20), uses the "&" operator to check if the job is 'MANAGER' and the deptno is 20

It will select the employee name and salary from the table where the salary is greater than 1000 and less than 2000 and the job is 'MANAGER' and the department number is 20. It's important to note that the logical operator '&&' has more precedence than the bitwise operator '&'. Therefore, the logical operator '&&' will be executed first before the bitwise operator '&'.

**5. Write a Query for the below statements**

**Table DEPT:**

**------------------------------**

**DEPTNO DNAME LOC**

**10 AC HDC**

**20 RES CDC**

**------------------------------**

**Table DEPT1:**

**------------------------------**

**DEPTNO DNAME LOC**

**10 AC HDC**

------------------------------

**Use the SET operator to merge the above two tables; also, no repetition data should be there.**

You can use the UNION operator to combine the results of two or more SELECT statements and eliminate duplicate rows. The UNION operator combines the result set of two or more SELECT statements and eliminates duplicate rows.

SELECT DEPTNO, DNAME, LOC FROM DEPT

UNION

SELECT DEPTNO, DNAME, LOC FROM DEPT1

This query will select the DEPTNO, DNAME, LOC from both the tables DEPT and DEPT1, and combine the results. Since the UNION operator eliminates duplicate rows, it will not return any duplicated data.

You can also use UNION ALL operator to combine the results of two or more SELECT statements, it does not eliminate duplicate rows

SELECT DEPTNO, DNAME, LOC FROM DEPT

UNION ALL

SELECT DEPTNO, DNAME, LOC FROM DEPT1

This query will select the DEPTNO, DNAME, LOC from both the tables DEPT and DEPT1, and combine the results with duplicate rows if any.