**SQL & Python Practice**

**Date:-10-12-2024 -Task**

**1.** **SQL Queries:-**

**🡪 New Column Creation:-**

**🡪 Syntax:-**

-- Alter table name

--Add column name

**🡪 Value Updates:-**

**🡪 Syntax:-**

--update table name

--set column name (new column name)

--where column name (it’s already column name)

**2.Python Syntax:-**

**🡪 New Column Creation:-**

**🡪 Syntax:-**

Df=df.copy()

Df[‘New\_Column’]=’Value’

Print(Df)

**🡪 Value Updates:-**

**🡪 Syntax:-**

Df[‘New\_Column’]=[‘Value1’, ‘Value2’, ‘Value3’, ‘Value4’, ‘Value5’]

**Date:-11-12-2024 \_Task**

**1.** **SQL Queries:-**

**🡪 Case When queries:- (single value update)**

**🡪 Syntax:-**

SELECT column\_name1, column\_name2, (it’s already column names in table)

CASE  
     WHEN *condition1* THEN *result1*  
    WHEN *condition2* THEN *result2*  
     ELSE default\_result

END AS Column Nema (New Creation Column Name)    
 FROM Table Name

**2.Python Syntax:-**

**🡪 Np Where condition:- (single value update)**

**🡪 Syntax:-**

Df[‘New\_Column’]=np.where(df[‘condition’], value\_if\_true, value\_if\_false)

**Date:-12-12-2024 \_Task**

**1.** **SQL Queries:-**

**🡪 Case When queries:- (multiple values updates)**

**🡪 Syntax:-**

SELECT column\_name1, column\_name2, column\_name3, (it’s already column names in table)

CASE

WHEN condition1 THEN result1

WHEN condition2 THEN result2

WHEN condition2 THEN result2

ELSE default\_result

END AS Column Name (New Creation Column Name)

FROM Table Name

**2.Python Syntax:-**

**🡪Np Where condition:- (multiple value updates)**

**🡪 Syntax:-**

Df[‘New\_Column’] = np.where (df[‘Score’] >= 50, ‘Pass’, ‘Fail’)

**Date:-13-12-2024 \_Task**

**1.SQL Join Queries:-**

**🡪 Inner Joins:-**

**🡪 Syntax:-**

Select\*

From table name (Base)

Inner join table name (Child)

On table name (Base). Column name (Matched column)=table name (Child). Column name (Matched column)

**🡪 Left Joins:-**

**🡪 Syntax:-**

Select\*

From table name (Base)

Left join table name (Child)

On table name (Base). Column name (Matched column)=table name (Child). Column name (Matched column)

**2.Python Syntax:-**

**🡪 Inner Joins:-**

**🡪 Syntax:-**

Df\_table name (Base).Merge(df\_ table name (Child),how=’inner’,on=’key’)

**🡪 Left Joins:-**

**🡪 Syntax:-**

Df\_table name (Base).Merge(df\_table name (Child),how=’left’,on=’key’)

**🡪 Right Joins:-**

**🡪 Syntax:-**

Df\_table name (Base).Merge(df\_table name (Child),how=’right’,on=’key’)

**🡪 Outer Joins:-**

**🡪 Syntax:-**

Df\_table name (Base).Merge(df\_table name (Child),how=’outer’,on=’key’)