**Experiment-6**

**Aim:**

Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULL IF and COALESCE functions.

**Description**

**Case**

* The CASE statement in SQL is a conditional expression that evaluates conditions and returns values based on those conditions.
* It acts like a simple IF-ELSE construct.
* A **CASE expression** is an inline conditional expression used to modify query results or calculations dynamically.

**NULLIF Function**

The NULLIF function compares two expressions and returns NULL if they are equal. If the expressions are not equal, it returns the first expression.

**COALESCE Function**

* The COALESCE function returns the first non-NULL value from a list of expressions.
* It’s often used to handle NULL values by providing default values.

**Procedure**

Step1: Initially Create the folder PL/SQL in the D or E drive.

Step2:Open the PL/SQL environment through as administrator role.

Step3: Enter the Credentials.

Step 4: Define\_editor=”notepad”;

Step5:Type below command for creating sql file

SQL>ed d:\PL\_SQL\filename.sql

Step 6:Enter the below program in the sql file

Step 7: Type below command to display the output

SQL> set serveroutput on;

Step 7: Type below command for executing sql file

SQL> @ E:\PLSQL\filename.sql;

**Program**

CREATE TABLE stugrade(sid INT PRIMARY KEY ,Name VARCHAR(50) NOT NULL, Age INT,Gender VARCHAR(10),

TotalScore INT,Grade CHAR(1) NULL );

INSERT INTO stugrade(sid,Name,Age,Gender,TotalScore,Grade) VALUES (1,'Alice',20,'Female',85,NULL);

INSERT INTO stugrade(sid,Name,Age,Gender,TotalScore,Grade) VALUES(2,'Bob',22,'Male',75,NULL);

INSERT INTO stugrade(sid,Name,Age,Gender,TotalScore,Grade) VALUES(3,'Charlie',21,'Male',92,NULL);

INSERT INTO stugrade(sid,Name,Age,Gender,TotalScore,Grade) VALUES(4,'Diana',19,'Female',45,NULL);

INSERT INTO stugrade(sid,Name,Age,Gender,TotalScore,Grade) VALUES(5,'Eve',NULL,'Female',NULL,NULL);

UPDATE stugrade SET Grade =

CASE

WHEN TotalScore IS NULL THEN 'N'

WHEN TotalScore >= 90 THEN 'A'

WHEN TotalScore >= 75 THEN 'B'

WHEN TotalScore >= 50 THEN 'C'

ELSE 'F'

END;

SELECT sid,Name,Age,Gender,TotalScore,Grade,

CASE

WHEN Grade = 'A' THEN 'Excellent'

WHEN Grade = 'B' THEN 'Good'

WHEN Grade = 'C' THEN 'Average'

WHEN Grade = 'F' THEN 'Fail'

ELSE 'Not Graded'

END AS PerformanceCategory,

-- Use COALESCE to handle NULL values for Age and TotalScore

COALESCE(Age, 18) AS AssumedAge, -- Default age is 18 if NULL

COALESCE(TotalScore, 0) AS AssumedScore -- Default score is 0 if NULL

FROM stugrade;

SELECT Name,TotalScore,Age,

COALESCE(TotalScore / NULLIF(Age, 0), 0) AS ScorePerYear -- Avoid division by zero

FROM stugrade;

-- Delete students with low scores (optional)

DELETE FROM stugrade WHERE TotalScore < 50;

SELECT \* FROM stugrade;

/