--SQL CASE STATEMENTS

/\*

CASE is used to provide if-then-else type of logic

There are 2 types of case statements: SIMPLE AND SEARCHED

SIMPLE: where we compare an expression to static values.

SEARCHED: where we compare an expression to one or more logical conditions.

\*/

Use master

Go

-- Create a database

CREATE DATABASE MOVIES

USE MOVIES

GO

--Create table

CREATE TABLE MOVIE

(MOVIEID INT IDENTITY (1, 1) PRIMARY KEY NOT NULL,

TITLE VARCHAR (50) NULL,

MOVIEDESC VARCHAR (100) NULL,

MOVIETIME INT NULL,

RATING CHAR (3) NULL,

MOVIEDATE SMALLDATETIME NULL

)

SELECT \* FROM MOVIE

--Insert data into movies table (made up descriptions)

INSERT INTO MOVIE

VALUES

('SCARFACE','TONY MONTANA TAKES ON MOB','180','R','10-01-2005'),

('BRAVEHEART','WALLACE CHALLANGES THE KING OF ENGLAND','200','R','10-01-2009'),

('IDENTITY','8 PEOPLE STRANDED IN A HOTEL','120','PG','08-01-2010'),

('SHREK','OGAR LOVES BEAUTIFUL LADY','80','PG','10-05-2010'),

('BRAVE ONE','WOMAN TAKES REVENGE','120', NULL,'11-01-2010'),

('JUMBO','FLYING ELEPHANT','65','G','10-01-1965'),

('ALIEN','ENCOUNTER ALIEN','210', NULL,'10-01-2000'),

('PLANET EARTH','DOCUMENTARTY ABOUT PLANET','400','PG','10-01-2010'),

('MULAN','ANIMATION ABOUT A GIRL TAKING ON THE HUN','160','PG','10-01-2005'),

('ALCATRAZ','CLINT ESCAPES THE ROCK','180', NULL,'11-01-2010'),

('ROCK','THE ROCK IS PENETRATED BY TERRORISTS','130','R','10-05-2000')

SELECT \* FROM MOVIE

--SIMPLE CASE EXAMPLE (USING THE RATING EXPRESSION)

SELECT TITLE, RATING,

CASE RATING --<< EXPRESSION

WHEN 'R' THEN 'ADULT SUPERVISION NEEDED' --<< States, if you find 'R' in the column Rating, then substitute the 'R' for 'ADULT SUPERVISION NEEDED'

WHEN 'G' THEN 'NO SUPERVISION NEEDED' --<< then go through this value

WHEN 'PG' THEN 'MUST BE 14 OR OLDER' --<< then go through this value

ELSE 'N/A' --<< and if you don't find any of the above, call it N/A

END AS 'GUIDENCE' --<< End the case statement with an alias for the new column

FROM MOVIE

--SEARCHED CASE EXAMPLE (NOT USING THE RATING EXPRESSION, BUT RATHER BOOLEAN EXPRESSION WITH OPERATORS)

SELECT \* FROM MOVIE

SELECT MOVIETIME, TITLE, RATING,

CASE

WHEN MOVIETIME <= 90 THEN 'IT’S A SHORT MOVIE'

WHEN MOVIETIME BETWEEN 90 AND 200 THEN 'IT’S A MEDUIM MOVIE'

WHEN MOVIETIME BETWEEN 201 AND 300 THEN 'IT’S A LONG MOVIE'

ELSE 'TOO LONG'

END AS 'LENGHT OF MOVIE'

FROM MOVIE

ORDER BY MOVIETIME

USE MOVIES

GO

--Create table

CREATE TABLE GRADES

(GRADEID INT IDENTITY (1, 1) PRIMARY KEY NOT NULL,

FNAME VARCHAR (20) NULL,

GRADE CHAR (3) NULL

)

--Insert data into table

INSERT INTO GRADES

VALUES

('BOB','75'),

('TOM','77'),

('SAM','65'),

('SUE','79'),

('LEWIS','95'),

('VAN','45'),

('ANDY','85'),

('MIKE','100'),

('BRAD','55'),

('DREW','51'),

('MANDY','88'),

('HENRY','85')

SELECT \* FROM GRADES

--WHAT ARE THE LETTER GRADES?

--90 - 100 = A

--80 - 89 = B

--70 - 79 = C

--60 - 69 = D

--50 - 59 = F

--SEARCHED CASE EXAMPLE

SELECT FNAME, GRADE,

CASE

WHEN GRADE = 100 THEN 'A+' --<< Go to the grade column and see it contains 100; if it does, then substitute it for 'A+'

WHEN GRADE > 90 THEN 'A'

WHEN (GRADE < 89) AND (GRADE > 80) THEN 'B'

WHEN (GRADE <= 79) AND (GRADE > 70) THEN 'C'

WHEN (GRADE < 69) AND (GRADE > 60) THEN 'D'

ELSE 'FAILED' --<< If all of the above conditions are not met, then enter 'Failed'

END AS 'LETTER GRADES' --<< End the case statement with an alias for the new column

FROM GRADES

ORDER BY GRADE DESC