

Nama : Horas Marolop Amsal Siregar

NIM : 11321067

Kelas : 32TI2

```
CREATE TABLE product (
    prod_nr INT NOT NULL,
    CONSTRAINT pk_product PRIMARY KEY (prod_nr),
    Name VARCHAR(30) NOT NULL,
    Price MONEY NOT NULL,
    Type VARCHAR(30) NOT NULL
)
GO

INSERT INTO product (prod_nr, Name, Price, Type) VALUES (1, 'tv', 500, 'electronics')
INSERT INTO product (prod_nr, Name, Price, Type) VALUES (2, 'radio', 100, 'electronics')
INSERT INTO product (prod_nr, Name, Price, Type) VALUES (3, 'ball', 100, 'sports')
INSERT INTO product (prod_nr, Name, Price, Type) VALUES (4, 'racket', 200, 'sports')
GO

SELECT * FROM product
GO
```

Result:

	prod...	Name	Price	Type
1	1	tv	500,00	electronics
2	2	radio	100,00	electronics
3	3	ball	100,00	sports
4	4	racket	200,00	sports

Exercise 1

Create a function with an input parameter the name of the product. Based on this input, the function should return or print a message like this: 'There are (the name of the product) in stock' or 'There are no (the name of the product) in stock'.

```
-- Exercise 1
-- Create a function with an input parameter the name of the product. Based on this input,
-- the function should return or print a message like this: 'There are (the name of the
-- product) in stock' or 'There are no (the name of the product) in stock'.
CREATE FUNCTION CheckProduct (@Name VARCHAR(30))
RETURNS VARCHAR(50)
AS
BEGIN
    DECLARE @Message VARCHAR(50)
    IF EXISTS (SELECT * FROM product WHERE Name = @Name)
        SET @Message = 'There are ' + @Name + ' in stock'
    ELSE
        SET @Message = 'There are no ' + @Name + ' in stock'
    RETURN @Message
END
GO

SELECT dbo.CheckProduct('book')
SELECT dbo.CheckProduct('tv')
GO
```

Result:

	(No column name)
1	There are no book in stock

	(No column name)
1	There are Tv in stock

Pada fungsi CheckProduct, terdapat parameter @Name yang bertipe data varchar(30). Fungsi tersebut akan mengembalikan nilai berupa string yang berisi pesan bahwa produk tersebut ada atau tidak ada di dalam stock. Pada fungsi tersebut, terdapat variabel @Message yang bertipe data varchar(50). Variabel tersebut akan diisi dengan pesan yang sesuai dengan kondisi yang terjadi. Jika produk yang dicari ada di dalam stock, maka variabel @Message akan diisi dengan pesan bahwa produk tersebut ada di dalam stock. Jika produk yang dicari tidak ada di dalam stock, maka variabel @Message akan diisi dengan pesan bahwa produk tersebut tidak ada di dalam stock.

Exercise 2

```
-- Exercise 2
-- Create a function with a numeric input parameter. Based on this input, the function should
-- return or print a message like this: 'the average price of sport products is greater or equal
-- or less than (the value of the input)' when that is the case in the database.
CREATE FUNCTION CheckPrice (@Price MONEY)
RETURNS VARCHAR(50)
AS
BEGIN
    DECLARE @Message VARCHAR(50)
    IF EXISTS (SELECT * FROM product WHERE Type = 'sports' AND Price >= @Price)
        SET @Message = 'the average price of sport products is greater or equal than ' + CAST(@Price AS VARCHAR(50))
    ELSE
        SET @Message = 'the average price of sport products is less than ' + CAST(@Price AS VARCHAR(50))
    RETURN @Message
END
GO

SELECT AVG(Price) AS AVG_PRICE FROM product WHERE Type = 'sports'
SELECT dbo.CheckPrice(100)
SELECT dbo.CheckPrice(150)
SELECT dbo.CheckPrice(400)
GO
```

Result:

	AVG_PRICE
1	150,00

	(No column name)
1	the average price of sport products is greater or

	(No column name)
1	the average price of sport products is greater or

	(No column name)
1	the average price of sport products is less than 4

Pada fungsi CheckPrice, terdapat parameter @Price yang bertipe data money. Fungsi tersebut akan mengembalikan nilai berupa string yang berisi pesan bahwa rata-rata harga produk olahraga lebih besar atau sama dengan atau lebih kecil dari nilai input. Pada fungsi tersebut, terdapat variabel @Message yang bertipe data varchar(50). Variabel tersebut akan diisi dengan pesan yang sesuai dengan kondisi yang terjadi. Jika rata-rata harga produk olahraga lebih besar atau sama dengan nilai input, maka variabel @Message akan diisi dengan pesan bahwa rata-rata harga produk olahraga lebih besar atau sama dengan nilai input. Jika rata-rata harga produk olahraga lebih kecil dari nilai input, maka variabel @Message akan diisi dengan pesan bahwa rata-rata harga produk olahraga lebih kecil dari nilai input.

Exercise 3

```
-- Exercise 3
-- Create a function to update the price of all records in table product by 10% until the average
-- price is greater than 500 (500 is defined by user through an input parameter).
-- Hint: You have to return a table.
CREATE FUNCTION UpdatePrice (@Price MONEY)
RETURNS @Table TABLE (prod_nr INT, Name VARCHAR(30), Price INT, Type VARCHAR(30))
AS
BEGIN
    DECLARE @AVG_PRICE MONEY
    SET @AVG_PRICE = (SELECT AVG(Price) FROM product)
    INSERT INTO @Table (prod_nr, Name, Price, Type)
    SELECT prod_nr, Name, Price, Type FROM product
    WHILE @AVG_PRICE < @Price
    BEGIN
        UPDATE @Table SET Price = Price * 1.1
        SET @AVG_PRICE = (SELECT AVG(Price) FROM @Table)
    END
    RETURN
END
GO

SELECT * FROM product
SELECT AVG(Price) AS AVG_PRICE FROM product
SELECT * FROM dbo.UpdatePrice(500)
SELECT AVG(Price) AS AVG_PRICE FROM dbo.UpdatePrice(500)
GO
```

Results				
	prod...	Name	Price	Type
1	1	tv	500,00	electronics
2	2	radio	100,00	electronics
3	3	ball	100,00	sports
4	4	racket	200,00	sports

AVG_PRICE	
1	225,00

	prod...	Name	Price	Type
1	1	tv	1175	electronics
2	2	radio	233	electronics
3	3	ball	233	sports
4	4	rac...	468	sports

AVG_PRICE	
1	527

Pada fungsi UpdatePrice, terdapat parameter @Price yang bertipe data money. Fungsi tersebut akan mengembalikan nilai berupa tabel yang berisi kolom prod_nr, Name, Price, dan Type. Pada fungsi

tersebut, terdapat variabel @AVG_PRICE yang bertipe data money. Variabel tersebut akan diisi dengan rata-rata harga produk yang ada di dalam tabel product. Pada fungsi tersebut, terdapat perulangan WHILE yang akan berjalan selama rata-rata harga produk yang ada di dalam tabel product kurang dari nilai input. Pada perulangan tersebut, tabel product akan diupdate dengan harga yang ditambah 10%. Setelah perulangan selesai, fungsi tersebut akan mengembalikan nilai berupa tabel yang berisi kolom prod_nr, Name, Price, dan Type.

Exercise 4

```
-- Exercise 4
-- Create a function with heading:
-- Which returns the dates of all sundaysbetween @dateFrom and @dateTo in a table with columns number and date
CREATE FUNCTION fnTableSundays (@dateFrom DATETIME, @dateTo DATETIME)
RETURNS @Table TABLE (number SMALLINT, date DATETIME)
AS
BEGIN
    DECLARE @i SMALLINT
    SET @i = 1
    WHILE @dateFrom <= @dateTo
    BEGIN
        IF DATEPART(dw, @dateFrom) = 1
        BEGIN
            INSERT INTO @Table (number, date) VALUES (@i, @dateFrom)
            SET @i = @i + 1
        END
        SET @dateFrom = DATEADD(dd, 1, @dateFrom)
    END
    RETURN
END
GO

SELECT * FROM dbo.fnTableSundays('2008-03-08', '2008-05-09')
GO
```

Result:

	numb...	date
1	1	2008-03-09 00:00:00.000
2	2	2008-03-16 00:00:00.000
3	3	2008-03-23 00:00:00.000
4	4	2008-03-30 00:00:00.000
5	5	2008-04-06 00:00:00.000
6	6	2008-04-13 00:00:00.000
7	7	2008-04-20 00:00:00.000
8	8	2008-04-27 00:00:00.000
9	9	2008-05-04 00:00:00.000

Pada fungsi fnTableSundays, terdapat parameter @dateFrom dan @dateTo yang bertipe data datetime. Fungsi tersebut akan mengembalikan nilai berupa tabel yang berisi kolom number dan date. Pada fungsi tersebut, terdapat variabel @i yang bertipe data smallint. Variabel tersebut akan diisi dengan angka 1. Pada fungsi tersebut, terdapat perulangan WHILE yang akan berjalan selama tanggal awal kurang dari tanggal akhir. Pada perulangan tersebut, jika tanggal yang ditunjuk oleh variabel @dateFrom adalah hari minggu, maka tanggal tersebut akan dimasukkan ke dalam tabel yang akan dikembalikan. Setelah perulangan selesai, fungsi tersebut akan mengembalikan nilai berupa tabel yang berisi kolom number dan date.