

## گزارش آزمایش سوم – آزمایشگاه پایگاه داده

۱- جدول زیر را با استفاده از دستورات SQL ایجاد کرده و داده های داده شده را در آن وارد کنید. (P-ID) از نوع auto increment بوده و زوج (نام ، نام خانوادگی) برابر کلید اصلی هستند.

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Tom	Vingvn 23	Stavanger

جدول persons

۱- ابتدا جدول بالا را تعریف کرده و سپس داده های خواسته شده را به آن می افزاییم:

## |-- Question 1

```
CREATE TABLE Persons_table(
  P_Id int IDENTITY(1,1),
  LastName varchar(255) NOT NULL,
  FirstName varchar(255) NOT NULL,
  Address varchar(255),
  City varchar(255),
  CONSTRAINT PK_Persons_table PRIMARY KEY (LastName, FirstName)
);
INSERT INTO Persons_table (LastName, FirstName, Address, City)
VALUES ('Hansen', 'Ola', 'Timoteivn 10', 'Sandnes');
INSERT INTO Persons_table (LastName, FirstName, Address, City)
VALUES ('Svendson', 'Tove', 'Borgvn 23', 'Sandnes');
INSERT INTO Persons_table (LastName, FirstName, Address, City)
VALUES ('Pettersen', 'Kari', 'Storgt 20', 'Stavanger');
INSERT INTO Persons_table (LastName, FirstName, Address, City)
VALUES ('Nilsen', 'Tom', 'Vingvn 23', 'Stavanger');
```

	P_Id	LastName	FirstName	Address	City
▶	1	Hansen	Ola	Timoteivn 10	Sandnes
	4	Nilsen	Tom	Vingvn 23	Stavanger
	3	Pettersen	Kari	Storgt 20	Stavanger
	2	Svendson	Tove	Borgvn 23	Sandnes

الف) افراد را برحسب نام خانوادگی به ترتیب صعودی مرتب کنید.

```
-- a
SELECT *
FROM Persons_table
ORDER BY LastName ASC;
```

ب) در یک transaction فیلد جدید شماره تلفن را از نوع nvarchar اضافه کنید. شماره تلفن ها حتما باید با پیش شماره ی ۰۰۱ شروع شوند(تعریف قید) و شماره تلفن های دلخواهی را به ۴ داده ی درون جدول نسبت دهید(طول شماره تلفن های اضافه شده برابر باشد).

```
-- b
BEGIN TRANSACTION;

-- add PhoneNumber column that allows NULL values
ALTER TABLE Persons_table
ADD PhoneNumber nvarchar(6) NULL
    CONSTRAINT CHK_Persons_table_PhoneNumber CHECK (PhoneNumber LIKE '001%');
COMMIT;

-- update the PhoneNumber for the first four rows
UPDATE Persons_table
SET PhoneNumber = '001123'
WHERE P_Id = 1;

UPDATE Persons_table
SET PhoneNumber = '001124'
WHERE P_Id = 2;

UPDATE Persons_table
SET PhoneNumber = '001125'
WHERE P_Id = 3;

UPDATE Persons_table
SET PhoneNumber = '001126'
WHERE P_Id = 4;
```

	P_Id	LastName	FirstName	Address	City	PhoneNum...
▶	1	Hansen	Ola	Timoteivn 10	Sandnes	001123
	4	Nilsen	Tom	Vingvn 23	Stavanger	001126
	3	Pettersen	Kari	Storgt 20	Stavanger	001125
	2	Svendson	Tove	Borgvn 23	Sandnes	001124

پ) با استفاده از دستور case نام، نام خانوادگی و آدرس افراد را نشان دهید به طوری که هر آدرس فعلی (نام خیابان)، نام میدان و شماره ی پلاک را شامل شود (این موارد را به دلخواه وارد کنید).

در گروه گفته شد میتوان بر حسب ایدی ها سناریو نوشت و سپس آدرس هارا تغییر داد:

```
--c
UPDATE Persons_table
SET Address = CASE
    WHEN P_Id = 1 THEN 'Tim Square Plaque 10'
    WHEN P_Id = 2 THEN 'Bong Square Plaque 23'
    WHEN P_Id = 3 THEN 'Stor Square Plaque 20'
    WHEN P_Id = 4 THEN 'Ving Square Plaque 23'
    ELSE Address -- if P_Id is not mentioned, keep the original Address
END;
```

ت) داده ی زیر را با شماره تلفن دلخواه وارد جدول کرده و ۳ فیلد اول جدول را به گونه ای نمایش دهید که برحسب نامشان به صورت صعودی مرتب شده اند. (در یک transaction)

7	Tjessem	Jakob	Nissetien 67	Sandnes
---	---------	-------	--------------	---------

توجه داشته باشید که P-ID داده ی جدید وارد شده برابر ۷ است.

```
--d
BEGIN TRANSACTION;

-- Add new data to table with P_Id = 7
-- Set identity insert to ON
SET IDENTITY_INSERT Persons_table ON

INSERT INTO Persons_table (P_Id, LastName, FirstName, Address, City, PhoneNumber)
VALUES (7, 'Tjessem', 'Jakob', 'Nissetien 67', 'Sandnes', '001127');

-- Show first three fields of table sorted by FirstName
SELECT FirstName, LastName, Address
FROM Persons_table
ORDER BY FirstName ASC;

COMMIT;

-- Set identity insert back to OFF
SET IDENTITY_INSERT Persons_table OFF
```

	P_Id	LastName	FirstName	Address	City	PhoneNum...
▶	1	Hansen	Ola	Tim Square ...	Sandnes	001123
	4	Nilsen	Tom	Ving Square...	Stavanger	001126
	3	Pettersen	Kari	Stor Square ...	Stavanger	001125
	2	Svendson	Tove	Borg Squar...	Sandnes	001124
	7	Tjessem	Jakob	Nissetien 67	Sandnes	001127

ث) اشخاصی را که شهری زندگی آن ها با حرف "S" شروع می شود را پس از ده ثانیه مشخص کنید.

```
--e
BEGIN TRANSACTION;
-- Wait for 10 seconds
WAITFOR DELAY '00:00:10';

-- Select people who live in a city starting with 'S'
SELECT FirstName, LastName, City
FROM Persons_table
WHERE City LIKE 'S%';

COMMIT;
```

100 %

Results Messages

	FirstName	LastName	City
1	Ola	Hansen	Sandnes
2	Tom	Nilsen	Stavanger
3	Kari	Pettersen	Stavanger
4	Tove	Svendson	Sandnes
5	Jakob	Tjessem	Sandnes

ج) روالی بنویسید که در آن متغیر temp از نوع int آخرین مقدار P\_ID در جدول را بگیرد، و به آن تعداد عبارت "okay" چاپ شود.

```
--f
-- Retrieve the last P_Id value and store it in the 'temp' variable
DECLARE @temp int;
SELECT TOP 1 @temp = P_Id FROM Persons_table ORDER BY P_Id DESC;

-- Print 'okay' as last P_Id value times
DECLARE @counter int = 1;
WHILE @counter <= @temp
BEGIN
    PRINT 'okay';
    SET @counter = @counter + 1;
END;
```

100 %

## Messages

okay  
okay  
okay  
okay  
okay  
okay  
okay

Completion time: 2023-03-17T20:17:21.9231379+03:30

چ داده ی زیر با شماره تلفن ۰۰۱۱۲۳۴۵۶۷ را در نظر بگیرید، در صورتی که شماره تلفن آن از Tjessem کوچک تر است، P-ID آن را ۶ قرار داده و در غیر این صورت P-ID را ۸ قرار دهید.

taylor	Jackson	Nisseisten87	Sandnes
--------	---------	--------------	---------

```
--g
-- Set identity insert to ON
SET IDENTITY_INSERT Persons_table ON

DECLARE @newPhoneNumber nvarchar(6) = '001567';

-- Retrieve Tjessem's phone number
DECLARE @tjessemPhoneNumber nvarchar(6);
SELECT @tjessemPhoneNumber = PhoneNumber
FROM Persons_table
WHERE LastName = 'Tjessem' AND FirstName = 'Jakob';

-- Determine the P_Id value for the new data based on phone number comparison
DECLARE @newP_Id int;
IF @newPhoneNumber < @tjessemPhoneNumber
BEGIN
    SET @newP_Id = 6;
END
ELSE
BEGIN
    SET @newP_Id = 8;
END

-- Insert the new data into the table with the appropriate P_Id value
INSERT INTO Persons_table (LastName, FirstName, Address, City, PhoneNumber, P_Id)
VALUES ('Taylor', 'Jackson', 'Nisseisten87', 'Sandnes', @newPhoneNumber, @newP_Id);

-- Set identity insert back to OFF
SET IDENTITY_INSERT Persons_table OFF
```

100 %

Messages

(1 row affected)

100 %

✓ Query executed successfully.



قابل ذکر است که با این شیوه نگرش در نهایت اگر مقدار تلفن کوچکتر باشد ما دوتا مقدار ۶ برای P\_Id ها خواهیم داشت که مطلوب نیست پس میتوان کد را بصورت دیگری هم نوشت که اگر مقدار تلفن کوچکتر بود مقادیر جدید را با مقدار قبلی جایگزین کند.

```
--g
-- Set identity insert to ON
SET IDENTITY_INSERT Persons_table ON

-- Check if the new phone number is lesser than Tjessem's
DECLARE @newPhoneNumber nvarchar(6) = '001567';
DECLARE @tjessemPhoneNumber nvarchar(6) = (
    SELECT PhoneNumber
    FROM Persons_table
    WHERE LastName = 'Tjessem'
);

IF @newPhoneNumber < @tjessemPhoneNumber
BEGIN
    -- Check if a data with P_Id=6 exists
    IF EXISTS (SELECT * FROM Persons_table WHERE P_Id = 6)
    BEGIN
        -- Replace the data with lesser phone number than Tjessem's
        UPDATE Persons_table
        SET LastName = 'Taylor',
            FirstName = 'Jackson',
            Address = 'Nisseisten87',
            City = 'Sandnes',
            PhoneNumber = @newPhoneNumber
        WHERE P_Id = 6 AND PhoneNumber < @tjessemPhoneNumber;
    END
    ELSE
    BEGIN
        -- Add the new data with P_Id=6
        INSERT INTO Persons_table (P_Id, LastName, FirstName, Address, City, PhoneNumber)
        VALUES (6, 'Taylor', 'Jackson', 'Nisseisten87', 'Sandnes', @newPhoneNumber);
    END
    END
    ELSE
    BEGIN
        -- Add the new data with P_Id=8
        INSERT INTO Persons_table (P_Id, LastName, FirstName, Address, City, PhoneNumber)
        VALUES (8, 'Taylor', 'Jackson', 'Nisseisten87', 'Sandnes', @newPhoneNumber);
    END
    -- Set identity insert back to OFF
    SET IDENTITY_INSERT Persons_table OFF
```

۲- جدول زیر را ایجاد کرده و داده های زیر را وارد کنید(کلید اصلی=student\_id)

name	student_id	grade
R1	8831047	12
R2	8831043	10
R3	8831031	15
R4	8831051	16
R5	8831012	11

می خواهیم به کسانی که نمره ی کمتر از ۱۵ دارند، ۲ نمره اضافه کنیم، به طوری که نمره ی جدید و نمره ی قدیمی آن ها نیز نمایش داده شود.(راهنمایی: استفاده از output)

ابتدا جدول ذکر شده را ایجاد و اطلاعات را وارد میکنیم:

```
--Question 2
CREATE TABLE Students(
    name varchar(255) NOT NULL,
    student_id int NOT NULL,
    grade int NOT NULL,
    CONSTRAINT PK_student_grades PRIMARY KEY (student_id)
);

INSERT INTO Students (name, student_id, grade)
VALUES ('R1', '8831047', 12);

INSERT INTO Students (name, student_id, grade)
VALUES ('R2', '8831043', 10);

INSERT INTO Students (name, student_id, grade)
VALUES ('R3', '8831031', 15);

INSERT INTO Students (name, student_id, grade)
VALUES ('R4', '8831051', 16);

INSERT INTO Students (name, student_id, grade)
VALUES ('R5', '8831012', 11);
```

در ادامه خواسته سوال را انجام میدهیم:

```
-- Update the grades and return old and new grades using OUTPUT command
UPDATE Students
SET grade = grade + 2
OUTPUT inserted.name, deleted.grade AS old_grade, inserted.grade AS new_grade
WHERE grade < 15;
```

100 %

Results Messages

	name	old_grade	new_grade
1	R1	13	15
2	R2	12	14
3	R1	14	16