ساخت جدول:

```
P_Id INT NOT NULL IDENTITY(1,1),
    LastName VARCHAR(50) NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    Address VARCHAR(100) NOT NULL,
    City VARCHAR(50) NOT NULL,
    PRIMARY KEY(LastName, FirstName)
);
```

وارد کردن داده ها:

```
□INSERT INTO dbo.People (LastName, FirstName, Address, City)

VALUES

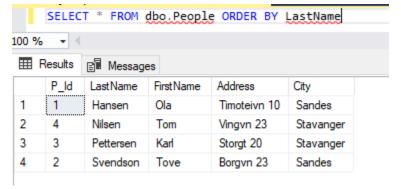
('Hansen', 'Ola', 'Timoteivn 10', 'Sandes'),

('Svendson', 'Tove', 'Borgvn 23', 'Sandes'),

('Pettersen', 'Karl', 'Storgt 20', 'Stavanger'),

('Nilsen', 'Tom', 'Vingvn 23', 'Stavanger');
```

مرتب بر اساس نام خانوادگی:



اضافه كردن تلفن:

نام - نام خانوادگی - آدرس: ☐ select LastName, FirstName, Address from dbo.People; -- CASE was not required --100 % -Results 📳 Messages Address LastName First Name Hansen Ola Timoteivn 10 1 2 Nilsen Vingvn 23 Tom 3 Pettersen Karl Storgt 20 4 Borgvn 23 Svendson Tove فورس ID: □ begin transaction SET IDENTITY_INSERT dbo.People ON; □INSERT INTO dbo.People (P Id, LastName, FirstName, Address, City, Phone) (7, 'Tjessem', 'Jakob', 'Nissestien 67', 'Sandes', '0018'); SELECT TOP 3 * FROM dbo.People ORDER BY FirstName; Messages First Name Address City Phone Jakob Nissestien 67 Sandes 0018 Karl Storgt 20 Stavanger 0013 Ola Timoteivn 10 0011 Sandes بعد از 10 ثانیه: ■WAITFOR DELAY '00:00:10'; SELECT * FROM dbo.People WHERE City Like 'S%'; Messages First Name LastName Address Phone City Hansen Ola Timoteivn 10 0011 Sandes Vingvn 23 0014 Tom Stavanger Pettersen Karl Storgt 20 Stavanger 0013 Svendson Tove Borgvn 23 Sandes 0012

0018

Sandes

go

commit

0 % 🕶 🔻 ■ Results

P_ld

7

3

1

100 % ▼ < ⊞ Results

P_ld

Nilsen

Tjessem

Jakob

Nissestien 67

1

4

3

2

7

2

3

4

5

}

LastName

Tjessem

Pettersen

Hansen

```
چاپ کردن OK:
```

```
⊟declare @temp int;
     declare @cnt int = 0;
     set @temp = (select TOP 1 p_id from dbo.PEOPLE ORDER BY -P_id);
   ⊟while @cnt < @temp
   ⊟BEGIN
     PRINT 'OK';
     SET @cnt = @cnt + 1;
    END;
100 % - 4
Messages
   OK
   OK
   OK
   OK
   OK
   OK
   OK
```

مقایسه:

```
☐ INSERT INTO dbo.People (FirstName, LastName, Address, City, Phone)
 VALUES
 ('Taylor', 'Jackson', 'Nissestien 87', 'Sandes', '0011234567');
 declare @tj varchar(18);
 set @tj = (select TOP 1 Phone from dbo.PEOPLE WHERE LastName = 'Tjessem');
 declare @jk varchar(18);
 set @tj = (select TOP 1 Phone from dbo.PEOPLE WHERE LastName = 'Jackson');
if @jk < @tj
     update dbo.People
     set P id = case LastName
                       when 'Jackson' then 6
 else
   update dbo.People
     set P id = case LastName
                       when 'Jackson' then 8
                       end
```

پاسخ سوال ها:

TRUNCATE is faster than DELETE, as it doesn't scan every record before removing it.
 TRUNCATE TABLE locks the whole table to remove data from a table; thus, this command also uses less transaction space than DELETE

2. The DROP command is used to remove table definition and its contents. Whereas the TRUNCATE command is used to delete all the rows from the table. ... DROP is a DDL(Data Definition Language) command. Whereas the TRUNCATE is also a DDL(Data Definition Language) command