ساخت جدول:

```
CREATE TABLE People(
  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:

```
SELECT * FROM dbo.People ORDER BY LastName
```

100 %

Results | Messages

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

اضافه کردن تلفن:

```
begin transaction

begin
ALTER TABLE dbo.People
ADD Phone nvarchar(18) CHECK (Phone LIKE '^001');
end
go
update dbo.People
set Phone = case P_id
                  when 1 then '0011'
                  when 2 then '0012'
                  when 2 then '0013'
                  when 2 then '0014'
                end
commit
```

نام - نام خانوادگی - آدرس:

```
select LastName, FirstName, Address from dbo.People;
-- CASE was not required --
```

100 %

Results | Messages

	LastName	FirstName	Address
1	Hansen	Ola	Timoteivn 10
2	Nilsen	Tom	Vingvn 23
3	Pettersen	Karl	Storgt 20
4	Svendson	Tove	Borgvn 23

فورس ID:

```
begin transaction

SET IDENTITY_INSERT dbo.People ON;

go

INSERT INTO dbo.People (P_Id, LastName, FirstName, Address, City, Phone)
VALUES
(7, 'Tjessem', 'Jakob', 'Nissestien 67', 'Sandes', '0018');

SELECT TOP 3 * FROM dbo.People ORDER BY FirstName;

commit
```

0 %

Results | Messages

	P_Id	LastName	FirstName	Address	City	Phone
1	7	Tjessem	Jakob	Nissestien 67	Sandes	0018
2	3	Pettersen	Karl	Storgt 20	Stavanger	0013
3	1	Hansen	Ola	Timoteivn 10	Sandes	0011

بعد از 10 ثانیه:

```
WAITFOR DELAY '00:00:10';

SELECT * FROM dbo.People WHERE City Like 'S%';
```

100 %

Results | Messages

	P_Id	LastName	FirstName	Address	City	Phone
1	1	Hansen	Ola	Timoteivn 10	Sandes	0011
2	4	Nilsen	Tom	Vingvn 23	Stavanger	0014
3	3	Pettersen	Karl	Storgt 20	Stavanger	0013
4	2	Svendson	Tove	Borgvn 23	Sandes	0012
5	7	Tjessem	Jakob	Nissestien 67	Sandes	0018

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 %

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
                    end
else
    update dbo.People
    set P_id = case LastName
                    when 'Jackson' then 8
                    end
```

پاسخ سوال ها:

1. **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**