Amirkabir University of Technology (AUT)
Tehran, Iran,
Lab Database, Computer Engineering Department
Tina Gholami (9531307) Report 3

Amirkabir University of Technology
(Tehran Polytechnic)

Part 1)

First, the following ER diagram's table was made in Microsoft SQL server, with the according primary keys, and inserted data:

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

```
create table Persons(
    P_Id int not null,
    LastName varchar(50) not null,
    FirstName varchar(50) not null,
    Address varchar(50),
    City varchar(50),
    primary key(LastName, FirstName)
);
```

```
insert into Persons(P_Id, LastName, FirstName, Address, City) values(1, 'Hansen', 'Ola', 'Timoteivn 10', 'Sandnes');
insert into Persons(P_Id, LastName, FirstName, Address, City) values(2, 'Svendson', 'Tove', 'Bargvn 23', 'Sandnes');
insert into Persons(P_Id, LastName, FirstName, Address, City) values(3, 'Pettersen', 'Kari', 'Storgt 20', 'Stavanger');
insert into Persons(P_Id, LastName, FirstName, Address, City) values(4, 'Nilsen', 'Tom', 'Vingvn 23', 'Stavanger');
```

Here is how the table looks like:

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

a)

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

```
select * from Persons
order by LastName ASC;
```

And the output is:

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

b)

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 تلفن‌های اضافه-شده برابر باشد).

```
begin transaction t1;

alter table Persons
add Phone nvarchar(10) constraint phone_beggining check (Phone like '001%');

commit transaction t1;
```

Then, we have to add data to the new field “Phone”. The next query can either be written inside a transaction, or not. I wrote it in a transaction, too.

```
begin transaction t2;

update Persons
set Phone = '0011111111'
where P_Id = 1;

update Persons
set Phone = '0012222222'
where P_Id = 2;

update Persons
set Phone = '0013333333'
where P_Id = 3;

update Persons
set Phone = '0014444444'
where P_Id = 4;

commit transaction t2;
```

And the output is:

	P_Id	LastName	FirstName	Address	City	Phone
1	1	Hansen	Ola	Timoteivn 10	Sandnes	0011111111
2	2	Svendson	Tove	Bargvn 23	Sandnes	0012222222
3	3	Pettersen	Kari	Storgt 20	Stavanger	0013333333
4	4	Nilsen	Tom	Vingvn 23	Stavanger	0014444444

c)

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	Nissestien 67	Sandnes

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

```
begin transaction t3;

insert into Persons(P_Id, LastName, FirstName, Address, City, Phone) values(7, 'Tjessem', 'Jakob', 'Nissestien 67', 'Sandnes', '0017777777');

select P_Id, LastName, FirstName
from Persons
order by FirstName ASC;

commit transaction t3;
```

And the output is:

	P_Id	LastName	FirstName
1	7	Tjessem	Jakob
2	3	Pettersen	Kari
3	1	Hansen	Ola
4	4	Nilsen	Tom
5	2	Svendson	Tove

d)

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

```
waitfor delay '000:00:10';

select LastName, FirstName
from Persons
where city like 'S%';
```

And the output is:

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

e)

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.

```
declare @temp int;

select @temp = P_Id from Persons;

while @temp > 0
    begin
        print 'okay';
        set @temp = @temp - 1;
    end
```

And the output is: (As can be seen below, 7 “okay”s are printed (the number of last P_Id = 7).)

```
okay
okay
okay
okay
okay
okay
okay
```

f)

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

```
declare @number nvarchar(10);
set @number = '0011234567';

if @number < (select phone
from Persons
where LastName = 'Tjessem')
insert into Persons(P_Id, LastName, FirstName, Address, City, Phone) values(6, 'Taylor', 'Jackson', 'Nisseisten 87', 'Sandnes', @number);
else
insert into Persons(P_Id, LastName, FirstName, Address, City, Phone) values(8, 'Taylor', 'Jackson', 'Nisseisten 87', 'Sandnes', @number);
```

And the output is: (As can be seen below, P_Id for the newly inserted data “Taylor Jackson” is 6, since his phone number is less than that of “Tjessem Jakob”.)

	P_Id	LastName	FirstName	Address	City	Phone
1	1	Hansen	Ola	Timoteivn 10	Sandnes	0011111111
2	4	Nilsen	Tom	Vingvn 23	Stavanger	0014444444
3	3	Pettersen	Kari	Storgt 20	Stavanger	0013333333
4	2	Svendson	Tove	Bargvn 23	Sandnes	0012222222
5	6	Taylor	Jackson	Nisseisten 87	Sandnes	0011234567
6	7	Tjessem	Jakob	Nissestien 67	Sandnes	0017777777

Part 2)

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

The table is created as the following:

```
create table Students(
    name varchar(10),
    student_id int not null primary key,
    grade int
);
```

And the data is inserted as below:

```
insert into Students values ('R1', 8831047, 12);
insert into Students values ('R2', 8831043, 10);
insert into Students values ('R3', 8831031, 15);
insert into Students values ('R4', 8831051, 16);
insert into Students values ('R5', 8831012, 11);
```

This is how the table looks like: (ordered by columns “name”)

	name	student_id	grade
1	R1	8831047	12
2	R2	8831043	10
3	R3	8831031	15
4	R4	8831051	16
5	R5	8831012	11

Next, it is asked to do the following query:

می خواهیم به کسانی که نمره‌ی 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)

```
declare @temp table(
    name varchar(10),
    student_id int,
    grade_old int,
    grade_new int
);


update Students
set grade = grade + 2
output inserted.name, inserted.student_id, deleted.grade, inserted.grade
into @temp
where grade<15;


select * from @temp;
```

The output is:

	name	student_id	grade_old	grade_new
1	R1	8831047	14	16
2	R2	8831043	12	14
3	R5	8831012	13	15

Extra Questions)

1.

1- چرا استفاده از دستور truncate سریعتر از استفاده از دستور delete برای حذف تمام سطرهای یک جدول است؟

“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”. Unlike “Delete”, “Truncate” does not return the number of rows deleted from the table. It also resets the table auto-increment value to the starting value (usually 1). That is why “Truncate” is generally faster than “Delete”.

2.

2- تفاوت دستور truncate و drop چیست؟

“Drop” statement is a Data Definition Language (DDL) Command which is used to delete existing database objects. It can be used to delete databases, tables, views, triggers, etc. from the relational database system (RDBS). Objects deleted using “Drop” are permanently lost and it cannot be rolled back. Unlike “Truncate” which only deletes the data of the tables, the “Drop” command deletes the data of the table as well as removes the entire schema/structure of the table from the database.