Group 8: Extra credit assignment

SQL Code:

1. Create a table with extra column to display our results:

```
drop table challengerORing

CREATE TABLE challengerORing (ORingFailure varchar(10),LaunchTemperature
int,LeakCheckPressure varchar(10),cluste varchar(10))
go
INSERT INTO challengerORing VALUES ('N',66,'Low','X')
INSERT INTO challengerORing VALUES ('N',69,'Low','X')
INSERT INTO challengerORing VALUES ('N',68,'Low','X')
INSERT INTO challengerORing VALUES ('N',67,'Low','X')
INSERT INTO challengerORing VALUES ('N',72,'Low','X')
INSERT INTO challengerORing VALUES ('N',73,'Low','X')
INSERT INTO challengerORing VALUES ('N',70,'Low','X')
INSERT INTO challengerORing VALUES ('N',78,'High','X')
INSERT INTO challengerORing VALUES ('N',67,'High','X')
INSERT INTO challengerORing VALUES ('N',67,'High','X')
INSERT INTO challengerORing VALUES ('N',75,'High','X')
INSERT INTO challengerORing VALUES ('N',70,'High','X')
INSERT INTO challengerORing VALUES ('N',81,'High','X')
INSERT INTO challengerORing VALUES ('N',76,'High','X')
INSERT INTO challengerORing VALUES ('N',79,'High','X')
INSERT INTO challengerORing VALUES ('N',75,'High','X')
INSERT INTO challengerORing VALUES ('N',76,'High','X')
INSERT INTO challengerORing VALUES ('Y',70,'Low','X')
INSERT INTO challengerORing VALUES ('Y',57,'High','X')
INSERT INTO challengerORing VALUES ('Y',63,'High','X')
INSERT INTO challengerORing VALUES ('Y',70,'High','X')
INSERT INTO challengerORing VALUES ('Y',53,'High','X')
INSERT INTO challengerORing VALUES ('Y',58,'High','X')
```

2. Write the procedure

```
drop procedure usp_clustering1
drop procedure usp_clustering2

use O_ring_failure
go


create procedure usp_clustering1 @cen1_temp float output, @cen2_temp float
as
declare @predY_N nvarchar(max)
declare @temp int
declare @cluste_temp varchar(10)
declare @C1_result_temp float
declare @C2_result_temp     float
declare @total1 int
set @total1 = 0
declare @count1 int
set @count1 =0
declare @total2 int
set @total2 = 0;
declare @count2 int
set @count2 =0
declare cluster_cursor1 cursor for select
dbo.challengerORing.ORingFailure,dbo.challengerORing.LaunchTemperature,dbo.challengerORin
g.cluste from challengerORing
for update of dbo.challengerORing.cluste
open cluster_cursor1
fetch next from cluster_cursor1 into @predY_N,@temp,@cluste_temp
while (@@FETCH_STATUS=0)
begin
       set @C1_result_temp = abs(@temp - @cen1_temp)
       set @C2_result_temp = abs(@temp - @cen2_temp)
       if(@C1_result_temp<@C2_result_temp)
              begin
              set @count1 = @count1 +1
              set @total1 = @total1 + @temp
              update challengerORing
              set cluste='Y'
              where current of cluster_cursor1
              end
       else
              begin
              set @count2 = @count2 +1
              set @total2 = @total2 + @temp
              update challengerORing
              set cluste='N'
              where current of cluster_cursor1
              end
       fetch next from cluster_cursor1 into @predY_N,@temp,@cluste_temp
end
close cluster_cursor1
deallocate cluster_cursor1
set @cen1_temp = @total1/@count1
```

```
go

--XX----------------------------------------------------------------------------------XX

use O_ring_failure
go

create procedure usp_clustering2 @cen1_temp float, @cen2_temp float output
as
declare @predY_N nvarchar(max)
declare @temp int
declare @cluste_temp varchar(10)
declare @C1_result_temp float
declare @C2_result_temp    float
declare @total1 int
set @total1 = 0
declare @count1 int
set @count1 =0
declare @total2 int
set @total2 = 0;
declare @count2 int
set @count2 =0
declare cluster_cursor2 cursor for select
dbo.challengerORing.ORingFailure,dbo.challengerORing.LaunchTemperature,dbo.challengerORin
g.cluste from challengerORing
for update of dbo.challengerORing.cluste
open cluster_cursor2
fetch next from cluster_cursor2 into @predY_N,@temp,@cluste_temp
while (@@FETCH_STATUS=0)
begin
       set @C1_result_temp = abs(@temp - @cen1_temp)
       set @C2_result_temp = abs(@temp - @cen2_temp)
       if(@C1_result_temp<@C2_result_temp)
              begin
              set @count1 = @count1 +1
              set @total1 = @total1 + @temp
              update challengerORing
              set cluste = 'Y'
              where current of cluster_cursor2
              end
       else
              begin
              set @count2 = @count2 +1
              set @total2 = @total2 + @temp
              update challengerORing
              set cluste='N'
              where current of cluster_cursor2
              end
       fetch next from cluster_cursor2 into @predY_N,@temp,@cluste_temp
end
close cluster_cursor2
deallocate cluster_cursor2
set @cen2_temp = @total2/@count2

go
```

```
--XX----------------------------------------------------------------------------------XX

declare  @cen1 float
set @cen1 = 65
declare @cen2 float
set @cen2 =70
declare @check float
set @check = 5.0
while(@check > 0.1)
begin
	declare @cen_temp1 float
	declare @cen_temp2 float
	set @cen_temp1=@cen1
	set @cen_temp2=@cen2

	exec usp_clustering1 @cen_temp1 output,@cen_temp2
	set @cen1 = @cen_temp1
	exec usp_clustering2 @cen_temp1, @cen_temp2 output
	set @cen2 = @cen_temp2
	set @check = abs(@cen_temp1-@cen1)
end
```

3. Results

Results | Messages

	ORingFailure	LaunchTemperature	LeakCheckPressure	cluste
1	N	66	Low	N
2	N	69	Low	N
3	N	68	Low	N
4	N	67	Low	N
5	N	72	Low	N
6	N	73	Low	N
7	N	70	Low	N
8	N	78	High	N
9	N	67	High	N
10	N	67	High	N
11	N	75	High	N
12	N	70	High	N
13	N	81	High	N
14	N	76	High	N
15	N	79	High	N
16	N	75	High	N
17	N	76	High	N
18	Y	70	Low	N
19	Y	57	High	Y
20	Y	63	High	Y
21	Y	70	High	N
22	Y	53	High	Y
23	Y	58	High	Y

Query executed successfully. | DESKTOP-AES5US6 (14.0 RTM) | DESKTOP-AES5US6\Kartik... | O_ring_failure | 00:00:00 | 23 rows