**NAME:**Kartikay Agrawal

**ROLL NUMBER:**2148064

**Exercise:**Demonstration of PL/SQL and ECA Rules

***1. Write a PL/SQL block to print the following pattern***

***1***

***1 2***

***1 2 3***

***1 2 3 4***

***1 2 3 4 5***

create or replace procedure s4(n in number) as

k varchar(40);

BEGIN

for i in 1..n

LOOP

k:=' ';

for j in 1..i

LOOP

k:= s|| ' ';

k := k || to\_char(j);

END LOOP;

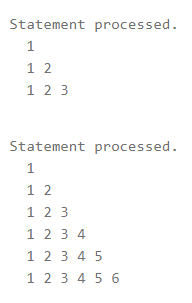
dbms\_output.put\_line(k);

END LOOP;

END;

execute s4(3);

execute s4(6);



***2. Writ a PL/SQL procedure using cursor to update the customer rating in the following***

***schema based on their feed back as detailed below.***

***Customer (Customer\_id, Customer\_name, Customer\_feedback, Customer\_Rating)***

***Feedback Rating***

***Excellent 5***

***Very Good 4***

***Good 3***

***Satisfactory 2***

***Bad 1***

***No Feedback 0***

create table Customer

(Customer\_id number(2) primary key,

Customer\_name varchar(20) not null,

Customer\_feedback varchar(15) not null check (Customer\_feedback in('Excellent','Very Good','Good','Satisfactory','Bad','No Feedback')),

Customer\_Rating number(1));

insert into Customer values(11,'Riya,'Good',NULL);

insert into Customer values(12,'Manan','Very Good',NULL);

insert into Customer values(13,'Piyush','Bad',NULL);

insert into Customer values(14,'Prachi','Excellent',NULL);

insert into Customer values(15,'Hitesh','No Feedback',NULL);

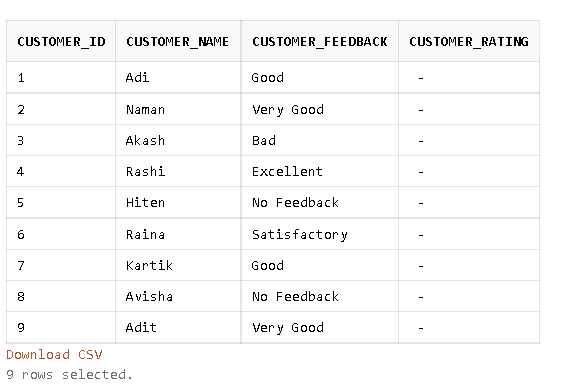
insert into Customer values(16,'Naina','Satisfactory',NULL);

insert into Customer values(17,'Adi','Good',NULL);

insert into Customer values(18,'Avi','No Feedback',NULL);

insert into Customer values(19,'Aditi','Very Good',NULL);

select \* from Customer;



create or replace procedure cursor\_demo

is

Feedback Customer.Customer\_feedback%type;

Rating Customer.Customer\_Rating%type:=0;

cnt number :=0;

cursor cur1 is select Customer\_Feedback from Customer for update of Customer\_Rating;

BEGIN

open cur1;

LOOP

fetch cur1 into Feedback;

exit when cur1%notfound;

if

(Feedback='Excellent') then

Rating:=5;

elsif

(Feedback='Very Good') then

Rating:=4;

elsif

(Feedback='Good') then

Rating:=3;

elsif

(Feedback='Satisfactory') then

Rating:=2;

elsif

(Feedback='Bad') then

Rating:=1;

else

Rating:=0;

end if;

cnt:=cnt+1;

update Customer set Customer\_Rating=Rating where Customer\_feedback=Feedback;

dbms\_output.put\_line(cnt||' '||Feedback);

END LOOP;

close cur1;

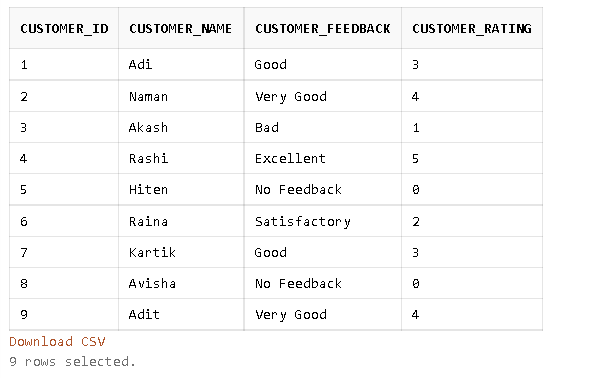
END;

Procedure created.

execute cursor\_demo;



select \* from Customer;



**The values for Rating are automatically inserted into Customer when we execute the cursor.**

***3. Use the following student schema; write a PL/SQL Trigger to add the records***

***automatically in to the respective category tables based on the percentage calculated***

***as total\_marks/100. Based on the percentage, the students were categorized as***

***Achievers, learners and Challengers as detailed below.***

***Achievers: 90% to 100%***

***Learners: 79% to 89%***

***Challengers: 0%-69%***

***Student (Roll\_No, Course, Course\_Code, Semester, Total\_Marks)***

***Achievers (Roll\_No, Percentage)***

***Learners (Roll\_No, Percentage)***

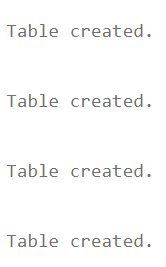
***Challengers (Roll\_No, Percentage)***

CREATE TABLE Student (Roll\_No NUMBER(1) PRIMARY KEY, Course VARCHAR(10) NOT NULL , Course\_Code NUMBER(3) NOT NULL, Semester CHAR(2) DEFAULT 'II', Total\_Marks number(3));

CREATE TABLE Achievers (Roll\_No NUMBER(1) PRIMARY KEY, Percentage REAL);

CREATE TABLE Learners (Roll\_No NUMBER(1) PRIMARY KEY, Percentage REAL);

CREATE TABLE Challengers (Roll\_No NUMBER(1) PRIMARY KEY, Percentage REAL);



create or replace trigger trig3

after insert on Student for each row

begin

if ((:new.Total\_Marks\*100)/120) between 90 and 100 then

insert into Achievers(Roll\_no,Percentage) values (:new.Roll\_no,((:new.Total\_Marks\*100)/120));

elsif((:new.Total\_Marks\*100)/120) between 79 and 89 then

insert into Learners (Roll\_no,Percentage) values (:new.Roll\_no,((:new.Total\_Marks\*100)/120));

elsif((:new.Total\_Marks\*100)/120) between 0 and 79 then

insert into Challengers (Roll\_no,Percentage) values (:new.Roll\_no,((:new.Total\_Marks\*100)/120));

else

raise\_application\_error(-200000,'Marks must be between 0-120');

end if;

end;



INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(1,'STATS',101,111);

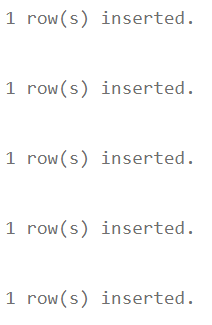
INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(2,'MATHS',102,63);

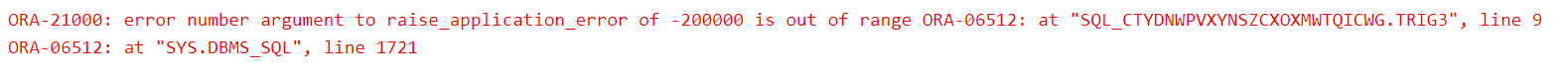
INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(3,'CHEMISTRY',104,22);

INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(4,'CS',103,89);

INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(5,'STATS',101,95);

INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(6,'CS',103,134);

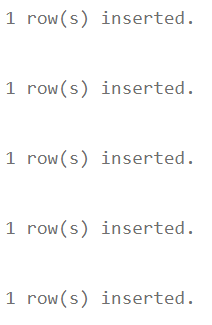




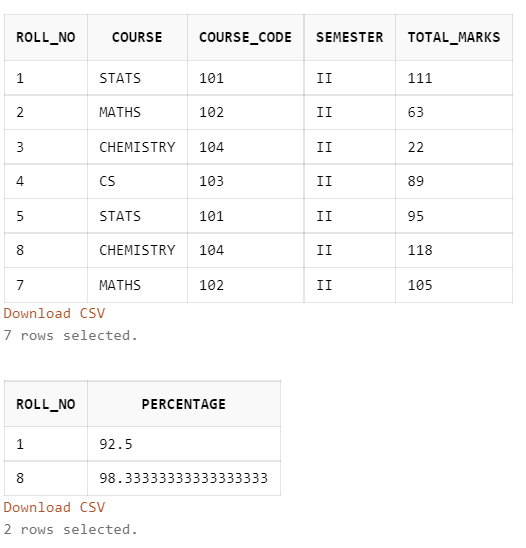
**The entry with total marks >120 raises an error and does not get inserted because of the trigger that percentage should be less than 100.**

INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(7,'MATHS',102,105);

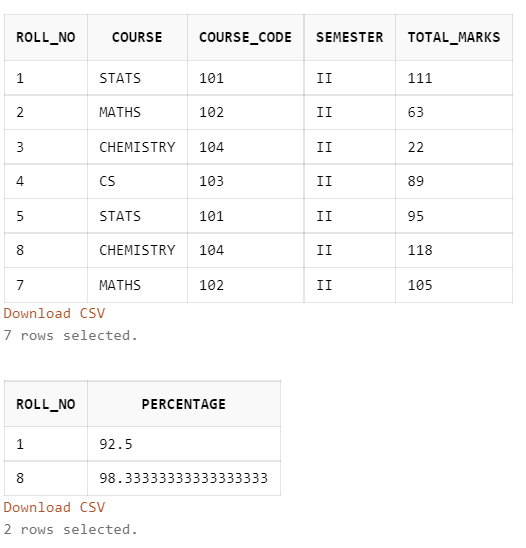
INSERT INTO STUDENT(ROLL\_NO,COURSE,COURSE\_CODE,TOTAL\_MARKS) VALUES(8,'CHEMISTRY',104,118);



SELECT \* FROM STUDENT;

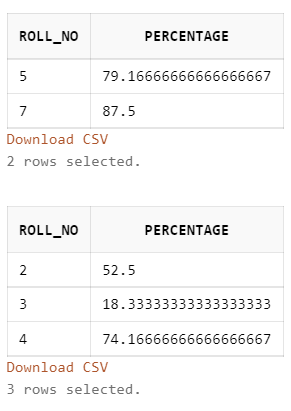


SELECT \* FROM ACHIEVERS;



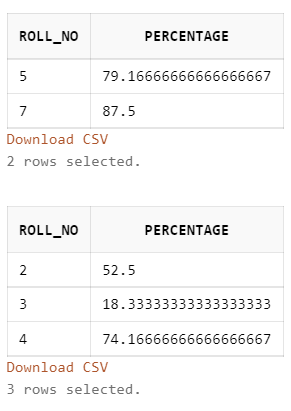
**The values with percentage between 90-100 are directly inserted into table Achievers because of the trigger.**

SELECT \* FROM LEARNERS;



**The values with percentage between 79-89 are directly inserted into table Learners because of the trigger.**

SELECT \* FROM CHALLENGERS;



**The values with percentage between 0-79 are directly inserted into table Challengers because of the trigger.**