-----CREATION OF TABLES QUERIES------------------

CREATE TABLE PROFESSION(

PID VARCHAR(10),

PNAME VARCHAR(20),

POPULARITY INT,

PRIMARY KEY(PID),

CHECK (POPULARITY>=0 AND POPULARITY<=10)

);

CREATE TABLE PERSON1(

ID INT,

PID VARCHAR(10),

AADHAR INT UNIQUE,

PRIMARY KEY(ID),

FOREIGN KEY(PID) REFERENCES PROFESSION(PID),

CHECK (ID>0)

);

CREATE TABLE PERSON2(

ID INT,

NAME VARCHAR(20),

GENDER VARCHAR(10),

DOB DATE,

STATE VARCHAR(20),

CITY VARCHAR(20),

AREA VARCHAR(20),

PINCODE INT,

PHONE\_NUMBER INT,

RATING INT,

ABOUT VARCHAR(100),

PRIMARY KEY(ID),

FOREIGN KEY(ID) REFERENCES PERSON1(ID),

CHECK (RATING>=0 AND RATING<=10)

);

CREATE TABLE COMPLAINT(

ID INT,

CID INT,

COMPLAINT VARCHAR(100),

PRIMARY KEY (CID),

FOREIGN KEY (ID) REFERENCES PERSON1(ID)

);

CREATE TABLE DETAILS1(

ID INT,

PID VARCHAR(10),

PRIMARY KEY(ID,PID),

FOREIGN KEY (ID) REFERENCES PERSON1(ID),

FOREIGN KEY (PID) REFERENCES PROFESSION(PID)

);

CREATE TABLE DETAILS2(

ID INT,

DATE\_AVAILABLE DATE,

INITIAL\_TIME TIMESTAMP,

FINAL\_TIME TIMESTAMP,

FEES INT,

PRIMARY KEY(ID),

FOREIGN KEY (ID) REFERENCES PERSON1(ID)

);

------------------------------------------------

ALTER SESSION SET NLS\_TIMESTAMP\_FORMAT='HH24:MI:SSXFF';

CREATE TABLE UPDATE\_LOGS\_PERSON2

( ID INT,

OPERATION VARCHAR(50),

DATE\_t DATE,

foreign key(ID) references PERSON2);

CREATE TABLE UPDATE\_LOGS\_DETAILS2

( ID INT ,

OPERATION VARCHAR(50),

DATE\_t DATE,

foreign key(ID) references PERSON2);

CREATE TABLE UPDATE\_LOGS\_COMPLAINT

( ID INT,

OPERATION VARCHAR(50),

DATE\_t DATE,

foreign key(ID) references PERSON2);

-----------------------------------------------------------------------------

--TRIGGERS HERE---

CREATE OR REPLACE TRIGGER CHANGE\_OF\_STATE

AFTER UPDATE OF STATE

ON PERSON2

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('OLD VALUE OF STATE=>'||:OLD.STATE);

dbms\_output.put\_line('NEW VALUE OF STATE=>'||:NEW.STATE);

end;

---TRIGGER2-------

CREATE OR REPLACE TRIGGER CHANGE\_OF\_CITY

AFTER UPDATE OF CITY

ON PERSON2

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('OLD VALUE OF STATE=>'||:OLD.CITY);

dbms\_output.put\_line('NEW VALUE OF STATE=>'||:NEW.CITY);

end;

-----------TRIGGER3---------------

CREATE OR REPLACE TRIGGER EVERYTHING\_CAPS

BEFORE INSERT OR UPDATE

ON PERSON2

FOR EACH ROW

BEGIN

:NEW.NAME :=UPPER(:NEW.NAME);

:NEW.STATE :=UPPER(:NEW.STATE);

:NEW.CITY :=UPPER(:NEW.CITY);

:NEW.AREA :=UPPER(:NEW.AREA);

END;

------------------------------------

------TRIGGER4-----------

CREATE OR REPLACE TRIGGER UPDATE\_LOG\_TRIG\_PERSON2

BEFORE INSERT OR UPDATE OR DELETE

ON PERSON2

FOR EACH ROW

BEGIN

IF INSERTING THEN

insert into UPDATE\_LOGS\_PERSON2 VALUES(:NEW.ID,'INSERTING',SYSDATE);

ELSIF UPDATING THEN

INSERT INTO UPDATE\_LOGS\_PERSON2 VALUES(:NEW.ID,'UPDATING',SYSDATE);

ELSIF DELETING THEN

INSERT INTO UPDATE\_LOGS\_PERSON2 VALUES(:OLD.ID,'DELETING',SYSDATE);

END IF;

END;

---------------------------------------------------------------------

----------------TRIGGER 5------------------------------

CREATE OR REPLACE TRIGGER UPDATE\_LOG\_TRIG\_DETAILS2

BEFORE INSERT OR UPDATE OR DELETE

ON DETAILS2

FOR EACH ROW

BEGIN

IF INSERTING THEN

insert into UPDATE\_LOGS\_DETAILS2 VALUES(:NEW.ID,'INSERTING',SYSDATE);

ELSIF UPDATING THEN

INSERT INTO UPDATE\_LOGS\_DETAILS2 VALUES(:NEW.ID,'UPDATING',SYSDATE);

ELSIF DELETING THEN

INSERT INTO UPDATE\_LOGS\_DETAILS2 VALUES(:OLD.ID,'DELETING',SYSDATE);

END IF;

END;

---------------------------------------------------------

--------------TRIGGER 6-----------------

CREATE OR REPLACE TRIGGER UPDATE\_LOG\_TRIG\_COMPLAINT

BEFORE INSERT OR UPDATE OR DELETE

ON COMPLAINT

FOR EACH ROW

BEGIN

IF INSERTING THEN

insert into UPDATE\_LOGS\_COMPLAINT VALUES(:NEW.ID,'INSERTING',SYSDATE);

ELSIF UPDATING THEN

INSERT INTO UPDATE\_LOGS\_COMPLAINT VALUES(:NEW.ID,'UPDATING',SYSDATE);

ELSIF DELETING THEN

INSERT INTO UPDATE\_LOGS\_COMPLAINT VALUES(:OLD.ID,'DELETING',SYSDATE);

END IF;

END;

-----------------------------------------------

---------------TRIGGER 7-----------------------

------------DATE IN DETAILS2 SHOULD NOT BE A SUNDAY------------

CREATE OR REPLACE TRIGGER DAY\_NOT\_SUNDAY

BEFORE INSERT OR UPDATE OF DATE\_AVAILABLE

ON DETAILS2

FOR EACH ROW

DECLARE

DAYY VARCHAR(50);

BEGIN

DAYY :=TRIM(TO\_CHAR(:NEW.DATE\_AVAILABLE,'DAY'));

IF DAYY='SUNDAY' THEN

RAISE\_APPLICATION\_ERROR(-20022,'WORKER NOT AVAILABLE ON SUNDAY');

END IF;

END;

-----------------------------------------------------------

-----------TRIGGER 8--------------------

CREATE OR REPLACE TRIGGER FIX\_SAL

AFTER INSERT

--BEFORE INSERT

ON DETAILS2

FOR EACH ROW

BEGIN

IF :NEW.FEES>1000 THEN

RAISE\_APPLICATION\_ERROR(-20023,'FEE INPUTTED IS ABOVE 1000');

END IF;

END;

-----------------------------------------

/\*

TESTING OF THE TRIGGERS

--FOR TRIGGER1

update PERSON2 set STATE='JHARKHAND' where ID=2;

--FOR TRIGGER2

update PERSON2 set CITY='MEERUT' where ID=2;

select \* from PERSON2

--EVERYTHING IS IN CAPS DUE TO WORKING OF TRIGGER3

SELECT \* FROM UPDATE\_LOGS\_PERSON2

--DUE TO TRIGGER 4 EVERY UPDATE INSERTION OR DELETE OPERATION IS STORED IN A TABLE WITH THE TIMESTAMP OF CHANGE

--WORKING OF TRIGGER 5

--DELETING FROM DETAILS2 TABLE

delete from DETAILS2 where ID=15

SELECT \* FROM UPDATE\_LOGS\_DETAILS2

SELECT \* FROM COMPLAINT

--WORKING OF TRIGGER 6 DELETEING A RECORD FROM COMPLAINT TABLE

DELETE FROM COMPLAINT WHERE ID=2

SELECT \* FROM UPDATE\_LOGS\_COMPLAINT

--WORKING OF TRIGGER8

select \* from DETAILS2

--TRIED TO ADD FEE ABOVE 1000 IT WILL BE NORMALIZED TO 1000

insert into DETAILS2 values(15,TO\_DATE('16-05-2022','DD-MM-YYYY'),'12:00:00','20:00:00',1200);

SELECT \* FROM USER\_TRIGGERS WHERE TABLE\_NAME='DETAILS2'

\*/

--ALL THE DATA INSERTION COMMANDS HAS TO BE ENTERED IN THIS AREA------------

INSERT INTO PROFESSION

VALUES ('LAB','Labor',7);

INSERT INTO PROFESSION

VALUES ('ELE','Electrician',10);

INSERT INTO PROFESSION

VALUES ('MEC','Mechanic',4);

INSERT INTO PROFESSION

VALUES ('NET','Network Expert',8);

INSERT INTO PERSON1

VALUES (1,'LAB','2758493726');

INSERT INTO PERSON1

VALUES (2,'ELE','3958673641');

INSERT INTO PERSON1

VALUES (3,'MEC','5938471645');

INSERT INTO PERSON1

VALUES (4,'NET','4356197546');

INSERT INTO PERSON1

VALUES (5,'MEC','4659124578');

INSERT INTO PERSON1

VALUES (6,'LAB','7946185347');

INSERT INTO PERSON1

VALUES (7,'NET','1379468251');

INSERT INTO PERSON1

VALUES (8,'ELE','4916573184');

INSERT INTO PERSON1

VALUES (9,'MEC','7946851465');

INSERT INTO PERSON1

VALUES (10,'LAB','3184679514');

INSERT INTO PERSON1

VALUES (11,'ELE','5564855613');

INSERT INTO PERSON1

VALUES (12,'MEC','6795415975');

INSERT INTO PERSON1

VALUES (13,'NET','9856741658');

INSERT INTO PERSON1

VALUES (14,'MEC','9453217568');

INSERT INTO PERSON1

VALUES (15,'NET','95154753684');

INSERT INTO PERSON2

VALUES (1,'Harshvardhan Rahane','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Rajasthan','Jaipur','Laxminagar',5462110,9595959595,5,'Average Worker');

INSERT INTO PERSON2

VALUES (2,'Ajay Dixit','Male',TO\_DATE('13/03/1989','DD/MM/YYYY'),'Punjab','Gurdaspur','Jaamnagar',684552,9191919191,8,'Very goood worker');

INSERT INTO PERSON2

VALUES (3,'Shikha Aggarwal','Female',TO\_DATE('13/12/1995','DD/MM/YYYY'),'Gujarat','Ahmedabad','Kamalnagar',892665,92929292,7,'Good Worker');

INSERT INTO PERSON2

VALUES (4,'Rajesh Mishra','Male',TO\_DATE('02/05/1995','DD/MM/YYYY'),'Maharashtra','Pune','Shivajinagar',9513655,9393939393,3,'Below Average Worker');

INSERT INTO PERSON2

VALUES (5,'Vishwas Kumar','Male',TO\_DATE('16/05/1990','DD/MM/YYYY'),'Uttar Pradesh','Lucknow','Laxminagar',856633,9494949494,9,'Amazing Worker');

INSERT INTO PERSON2

VALUES (6,'Tarun Singh','Male',TO\_DATE('23/05/1994','DD/MM/YYYY'),'Punjab','Patiala','Model Town',523002,9696969696,3,'Below Average Worker');

INSERT INTO PERSON2

VALUES (7,'Sankalp Saxena','Male',TO\_DATE('23/05/1999','DD/MM/YYYY'),'Haryana','Sirsa','Ambarsariya Road',252112,8181818181,5,'Average Worker');

INSERT INTO PERSON2

VALUES (8,'Anukrit Chugh','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Rajasthan','Jaipur','Laxmi Marg',5462110,8282828282,9,'Amazing Worker');

INSERT INTO PERSON2

VALUES (9,'Hreyhaan Malpani','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Gujarat','Ahmedabad','Dandiya Nagar',848955,8383838383,5,'Average Worker');

INSERT INTO PERSON2

VALUES (10,'Harneet Singhvi','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Uttar Pradesh','Prayagraj','Kamal Road',534545,84848484,3,'Below Average Worker');

INSERT INTO PERSON2

VALUES (11,'Shreyash Soni','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Gujarat','Surat','Saraswatinagar',3845575,8585858585,10,'Outstanding Worker');

INSERT INTO PERSON2

VALUES (12,'Madhav Kumar','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Maharashtra','Kolhapur','Sector 13',934772,8686868686,5,'Average Worker');

INSERT INTO PERSON2

VALUES (13,'Harshit Sodhi','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Gujarat','Vadodra','Sector 55',824112,8787878787,7,'Good Worker');

INSERT INTO PERSON2

VALUES (14,'Suresh Wadekar','Male',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Uttar Pradesh','Meerut','Kali Mata Road',942114,8888888888,3,' Below Average Worker');

INSERT INTO PERSON2

VALUES (15,'Pinky Chaddha','Female',TO\_DATE('23/05/1993','DD/MM/YYYY'),'Maharashtra','Mumbai','Laxminagar',482112,9999999999,6,'Above Average Worker');

INSERT INTO COMPLAINT

VALUES (2,6587,'My ceiling fan is not working. Need help.');

INSERT INTO COMPLAINT

VALUES (15,5146,'Internet connection is down since three days. Need it fixed ASAP.');

INSERT INTO COMPLAINT

VALUES (9,4584,'My Ford Ikon is not starting. I have to leave for a trip in two days!');

INSERT INTO COMPLAINT

VALUES (10,5483,'Need a gardener to water my plants and tend to my garden.');

INSERT INTO COMPLAINT

VALUES (2,2354,'My rooms switchboard is not working.');

INSERT INTO DETAILS1

VALUES (1,'LAB');

INSERT INTO DETAILS1

VALUES (2,'ELE');

INSERT INTO DETAILS1

VALUES (3,'MEC');

INSERT INTO DETAILS1

VALUES (4,'NET');

INSERT INTO DETAILS1

VALUES (5,'MEC');

INSERT INTO DETAILS1

VALUES (6,'LAB');

INSERT INTO DETAILS1

VALUES (7,'NET');

INSERT INTO DETAILS1

VALUES (8,'ELE');

INSERT INTO DETAILS1

VALUES (9,'MEC');

INSERT INTO DETAILS1

VALUES (10,'LAB');

INSERT INTO DETAILS1

VALUES (11,'ELE');

INSERT INTO DETAILS1

VALUES (12,'MEC');

INSERT INTO DETAILS1

VALUES (13,'NET');

INSERT INTO DETAILS1

VALUES (14,'MEC');

INSERT INTO DETAILS1

VALUES (15,'NET');

SELECT \* FROM DETAILS1

INSERT INTO DETAILS2

VALUES (1,TO\_DATE('07/05/2022','DD/MM/YYYY'),'12:00:00','20:00:00',500);

INSERT INTO DETAILS2

VALUES (2,TO\_DATE('09/05/2022','DD/MM/YYYY'),'08:00:00','12:00:00',600);

INSERT INTO DETAILS2

VALUES (3,TO\_DATE('14/05/2022','DD/MM/YYYY'),'9:00:00','14:00:00',900);

INSERT INTO DETAILS2

VALUES (4,TO\_DATE('21/05/2022','DD/MM/YYYY'),'15:00:00','20:00:00',400);

INSERT INTO DETAILS2

VALUES (5,TO\_DATE('09/06/2022','DD/MM/YYYY'),'08:00:00','20:00:00',100);

INSERT INTO DETAILS2

VALUES (6,TO\_DATE('11/06/2022','DD/MM/YYYY'),'06:00:00','15:00:00',700);

INSERT INTO DETAILS2

VALUES (7,TO\_DATE('12/05/2022','DD/MM/YYYY'),'18:00:00','20:00:00',300);

INSERT INTO DETAILS2

VALUES (8,TO\_DATE('16/05/2022','DD/MM/YYYY'),'12:00:00','16:00:00',200);

INSERT INTO DETAILS2

VALUES (9,TO\_DATE('09/06/2022','DD/MM/YYYY'),'15:00:00','18:00:00',700);

INSERT INTO DETAILS2

VALUES (10,TO\_DATE('21/05/2022','DD/MM/YYYY'),'12:00:00','20:00:00',600);

INSERT INTO DETAILS2

VALUES (11,TO\_DATE('02/06/2022','DD/MM/YYYY'),'18:00:00','20:00:00',400);

INSERT INTO DETAILS2

VALUES (12,TO\_DATE('10/05/2022','DD/MM/YYYY'),'09:00:00','14:00:00',300);

INSERT INTO DETAILS2

VALUES (13,TO\_DATE('12/05/2022','DD/MM/YYYY'),'16:00:00','20:00:00',800);

INSERT INTO DETAILS2

VALUES (14,TO\_DATE('13/05/2022','DD/MM/YYYY'),'11:00:00','17:00:00',900);

INSERT INTO DETAILS2

VALUES (15,TO\_DATE('16/05/2022','DD/MM/YYYY'),'14:00:00','18:00:00',100);

declare

NOT\_ENOUGH\_VALUES EXCEPTION;

PRAGMA EXCEPTION\_INIT(NOT\_ENOUGH\_VALUES,-06550);

CHECK\_VIOLATE EXCEPTION;

PRAGMA EXCEPTION\_INIT(CHECK\_VIOLATE,-02290);

-----------------------BUSINESS EXCEPTIONS ARE HERE-------------------------------

NOT\_ENOUGH\_WORKERS EXCEPTION;

MANY\_COM EXCEPTION;--too many complaints exception

NO\_FREE\_WORKER EXCEPTION;--NO WORKER IS FREE AT THE MOMENT

TOO\_LOW\_RATING EXCEPTION;--RATING OF A WORKER IS VERY LOW SO AN EXCEPTION IS RAISED

nume int;

num1 int;

num2 int;

num3 int;

num4 int;

num5 int;

del\_id int;

BEGIN

--------------------------------------------------------

select count(\*) INTO nume from PERSON1;

if nume<15 then

raise NOT\_ENOUGH\_WORKERS;

END IF;

-------------------------------------------------------

---------------------------------------------------------

select count(\*) into num1 from COMPLAINT;

if num1 > 20 then

raise MANY\_COM;

end if;

---------------------------------------------------------

select count(\*) into num2 from DETAILS2;

IF num2 =0 THEN

RAISE NO\_FREE\_WORKER;

END IF;

--------------------------------------------------------

--THIS CODE NORMALIZES THE FEES SET BY THE WORKERS,IF FOUND ABOVE 500 IT WILL BE REDUCED TO 500

select count(\*) into num3 from DETAILS2;

for i in 1..num3 loop

select FEES into num4 from DETAILS2 where ID=i;

if num4 > 1000 then

update DETAILS2 set FEES=1000 where ID=i;

end if;

end loop;

--------------------------------------------------------

--this code finds the worker with low rating and raises an exception

select count(\*) into num3 from PERSON2;

for i in 1..num3 loop

select RATING into num4 from PERSON2 where ID=i;

if num4<=1 then

raise TOO\_LOW\_RATING;

end if;

end loop;

------------------------------------------------------

exception

--predefined exceptions starts

when DUP\_VAL\_ON\_INDEX then

dbms\_output.put\_line('EXCEPTION:ATTEMPT TO ADD DUPLICATE RECORD IN A UNIQUE CONSTRAINT COLUMN');

when INVALID\_NUMBER then

dbms\_output.put\_line('EXCEPTION:ATTEMPT TO CONVERSION OF STRING TO A NUMBER');

when VALUE\_ERROR then

dbms\_output.put\_line('EXCEPTION:ATTEMPT TO CONVERSION OF STRING TO A NUMBER');

when NO\_DATA\_FOUND then

dbms\_output.put\_line('EXCEPTION: NO DATA FOUND IN THE TABLE COORESPONDING TO THE GIVEN VALUE');

when TOO\_MANY\_ROWS then

dbms\_output.put\_line('EXCEPTION: MULTIPLE ROWS RETURNED BY THE QUERY: EXCPETED A SINGLE ROW');

WHEN ZERO\_DIVIDE then

dbms\_output.put\_line('EXCEPTION: AN ATTEMPT IS MADE TO DIVIDE A NUMBER OF ZERO');

--PREDEFINED EXCEPTION ENDS

WHEN NOT\_ENOUGH\_VALUES then

dbms\_output.put\_line('EXCEPTION: NOT ENOUGH VALUES IN THE INSERT QUERY');

WHEN CHECK\_VIOLATE THEN

DBMS\_OUTPUT.PUT\_LINE('EXCEPTION: CHECK CONSTRAINT IS VIOLATED');

WHEN NOT\_ENOUGH\_WORKERS THEN

DBMS\_OUTPUT.PUT\_LINE('EXCEPTION: WE ARE IN A DIRE NEED OF WORKERS');

WHEN MANY\_COM THEN

DBMS\_OUTPUT.PUT\_LINE('EXCEPTION: WE HAVE TOO MANY COMPLAINTS DO SOMETHING');

WHEN NO\_FREE\_WORKER THEN

DBMS\_OUTPUT.PUT\_LINE('EXCEPTION: NO WORKER IS FREE AT THE MOMENT');

WHEN TOO\_LOW\_RATING THEN

DBMS\_OUTPUT.PUT\_LINE('EXCEPTION: RATING IS TOO LOW');

END;