**CREATE TABLE CODE:-**

1)

Create table guest (

guest\_id number,

guest\_FN varchar(20),

guest\_LN varchar(20),

guest\_add varchar(20),

guest\_email varchar(20),

guest\_phno number,

password varchar(10),

CONSTRAINT Gid primary key(guest\_id));

Create sequence guest\_id\_seq;

2)

create table room(

room\_no number,

room\_type varchar(4),

room\_rate number,

CONSTRAINT Rno primary key (room\_no));

3)

create table booking (

booking\_id number,

room\_no number,

guest\_id number,

adult number,

children number,

check\_in\_date date,

check\_out\_date date,

tariff number,

CONSTRAINT Rnofk foreign key(room\_no) references room(room\_no),

CONSTRAINT Gidfk foreign key(guest\_id) references guest(guest\_id));

Create sequence bookid start with 1111;

4)

create table bill(

bill\_no number,

guest\_id number,

amount number,

b\_date date,

payment\_method varchar(20),

CONSTRAINT Bno primary key(bill\_no),

foreign key (guest\_id) references guest(guest\_id));

create sequence bill\_id start with 101;

5)

create table food(

food\_item\_id number,

type varchar(10),

rate number,

CONSTRAINT Fid primary key(food\_item\_id));

6)

create table ordering\_food(

food\_item\_id number,

guest\_id number,

quantity number,

foreign key (food\_item\_id) references food(food\_item\_id) on delete cascade,

foreign key (guest\_id) references guest(guest\_id));

7)

create table facilities(

facility\_id number,

type varchar(10),

rate number,

CONSTRAINT Faid primary key(facility\_id));

8)

create table use\_facility(

facility\_id number,

guest\_id number,

foreign key (facility\_id) references facilities(facility\_id) on delete cascade,

foreign key (guest\_id) references guest(guest\_id));

9)

create table feedback(

guest\_id number,

feedback varchar(100),

foreign key (guest\_id) references guest(guest\_id));

10)

create table user\_report1(

sq11 number,

username1 varchar(100),

modify1 date,

primary key(sq11));

create sequence sq11 start with 1;

**PROCEDURES TRIGGERS AND FUNCTIONS-**

1. CREATE ACCOUNT

create or replace procedure create\_account(fname IN varchar,lname IN varchar,add IN varchar,email IN varchar,phoneno IN number,pswd IN varchar) IS

gid number;

begin

insert into guest values(guest\_id\_seq.nextval,fname,lname,add,email,phoneno,pswd);

select guest\_id into gid from guest where guest\_FN=fname and guest\_LN=lname;

dbms\_output.put\_line('YOUR UNIQUE GUEST ID IS:'||gid);

end;

1. USER LOGIN

create or replace procedure verify\_user(username varchar, pswd varchar) is

gid number;

begin

select guest\_id into gid from guest where guest\_email=username and password=pswd;

if gid>0 then

dbms\_output.put\_line('Welcome');

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('Invalid username or password');

end;

1. BILL GENERATION

create or replace procedure bill\_gen(guestid IN number) IS

total1 number;

total2 number;

total3 number;

total4 number;

begin

select sum(tariff) into total1 from booking where guest\_id=guestid;

if total1 is null then total1:=0;

end if;

select sum(f.rate) into total2 from facilities f,use\_facility u where u.facility\_id=f.facility\_id and guest\_id=guestid;

if total2 is null then total2:=0;

end if;

select sum(o.quantity\*f.rate) into total4 from ordering\_food o,food f where guest\_id=guestid and o.food\_item\_id=f.food\_item\_id;

if total4 is null then total4:=0;

total3:=total1+total2+total4;

end if;

insert into bill values (bil\_id.nextval,guestid,total3,sysdate,'cash');

dbms\_output.put\_line('your total bill amount is:'||total3);

end;

1. BOOK ROOM

create or replace procedure book\_room(guestid IN int, checkin IN date,checkout IN date, roomtype IN varchar,adult1 IN varchar,children1 IN varchar) IS

num1 number;

num2 number;

roomno number;

tariff1 number;

t1 booking%rowtype;

begin

select b.room\_no into roomno from room r, booking b where

r.room\_no=b.room\_no and r.room\_type=roomtype and checkin not between check\_in\_date and check\_out\_date and checkout not between check\_in\_date and check\_out\_date and rownum=1;

t1.room\_no:=roomno;

t1.guest\_id:=guestid;

t1.check\_in\_date:=checkin;

t1.check\_out\_date:=checkout;

t1.adult:=adult1;

t1.children:=children1;

select room\_rate into tariff1 from room where room\_no=roomno;

num2:=check\_freq(guestid);

if num2=1 then

dbms\_output.put\_line('CONGRATS !! YOU RECEIVE 20% DISCOUNT');

t1.tariff:=tariff1\*(checkout-checkin)\*0.8;

else

t1.tariff:=tariff1\*(checkout-checkin);

end if;

insert into booking values(bookid.nextval,t1.room\_no,t1.guest\_id,t1.adult,t1.children,t1.check\_in\_date,t1.check\_out\_date,t1.tariff);

dbms\_output.put\_line('YOUR ROOM NO IS: '||t1.room\_no||'TOTAL PRICE IS: '||t1.tariff);

exception

when no\_data\_found then

dbms\_output.put\_line('no data found');

end;

5) CANCEL BOOKING

create or replace procedure cancel\_booking(guestid IN number,bookingid IN number)

IS

Begin

delete from booking where booking\_id=bookingid and guest\_id=guestid;

dbms\_output.put\_line('guestid ' ||guestid||' booking is cancelled');

exception

when no\_data\_found then

dbms\_output.put\_line('no such booking');

end;

1. VALIDATE ADMIN

create or replace function validate\_admin( username in varchar, password in varchar) return number is

begin

if username='admin' and password='admin123' then

return 1;

else

return 0;

end if;

end;

1. CHECK AVAILAIBILITY

create or replace procedure check\_avail(checkin in date, checkout in date,roomtype in varchar)

is

cursor c1 is select b.room\_no from room r, booking b where

r.room\_no=b.room\_no and r.room\_type=roomtype and checkin not between check\_in\_date and check\_out\_date and checkout not between check\_in\_date and check\_out\_date;

r number;

count1 number:=0;

begin

open c1;

loop

fetch c1 into r;

exit when c1%notfound;

count1:=count1+1;

end loop;

if count1=0 then

dbms\_output.put\_line('no rooms available');

else

dbms\_output.put\_line(count1||' rooms are available on entered date');

end if;

end;

1. TO MAKE CHANGES IN ROOM TABLE

create or replace procedure room\_changes(username in varchar,password in varchar,roomno in number,roomtype in varchar,rate in number,val in number) is

num number;

begin

num:=validate\_admin(username,password);

if num=1 then

if val=1 then

insert into room values(roomno,roomtype,rate);

dbms\_output.put\_line('INSERTION SUCCESSFUL');

elsif val=2 then

delete from room where room\_no=roomno;

dbms\_output.put\_line('RECORD DELETED SUCCESSFULLY');

elsif val=3 then

update room set room\_rate=rate where room\_no=roomno;

dbms\_output.put\_line('UPDATED ROOM DETAILS');

end if;

else

dbms\_output.put\_line('INVALID LOGIN');

end if;

end;

1. DELETE ROWS AFTER BILL GENERATION

create or replace trigger delete\_rows

after insert on bill

declare

guest number;

begin

select guest\_id into guest from bill where bill\_no =(select max(bill\_no) from bill);

delete from booking where guest\_id=guest;

delete from ordering\_food where guest\_id=guest;

delete from use\_facility where guest\_id=guest;

end;

1. TO CHECK IF FREQUENT USER

create or replace function check\_freq(guestid number)

return number

is

num number;

freq number;

begin

select count(\*) into freq from bill where guest\_id=guestid;

if freq>2 then

return 1;

else

return 0;

end if;

end;

1. AUDITING BILL TABLE

create or replace trigger trig1

after insert or update or delete on bill

begin

insert into user\_report1 values(sq11.nextval,user,sysdate);

end;

1. TO MAKE CHANGES IN FOOD TABLE

create or replace procedure food\_changes(username in varchar,password in varchar,fid in number,type1 in varchar,rate1 in number,val in number) is

num number;

begin

num:=validate\_admin(username,password);

if num=1 then

if val=1 then

insert into food values(fid,type1,rate1);

dbms\_output.put\_line('INSERTION SUCCESSFUL');

elsif val=2 then

delete from food where food\_item\_id=fid;

dbms\_output.put\_line('RECORD DELETED SUCCESSFULLY');

elsif val=3 then

update food set rate=rate1 where food\_item\_id=fid;

dbms\_output.put\_line('UPDATED FOOD DETAILS');

end if;

else

dbms\_output.put\_line('INVALID LOGIN');

end if;

end;

1. TO MAKE CHANGES IN FACILITIES TABLE

create or replace procedure facility\_changes(username in varchar,password in varchar,fid in number,type1 in varchar,rate1 in number,val in number) is

num number;

begin

num:=validate\_admin(username,password);

if num=1 then

if val=1 then

insert into facilities values(fid,type1,rate1);

dbms\_output.put\_line('INSERTION SUCCESSFUL');

elsif val=2 then

delete from facilities where facility\_id=fid;

dbms\_output.put\_line('RECORD DELETED SUCCESSFULLY');

elsif val=3 then

update facilities set rate=rate1 where facility\_id=fid;

dbms\_output.put\_line('UPDATED FOOD DETAILS');

end if;

else

dbms\_output.put\_line('INVALID LOGIN');

end if;

end;

create or replace procedure check\_avail(checkin in date, checkout in date,roomtype in varchar)

is

cursor c1 is select room\_no from room where room\_type=roomtype and room\_no not in(select b.room\_no from booking b,room r where r.room\_no=b.room\_no and checkin between check\_in\_date and check\_out\_date and checkout between check\_in\_date and check\_out\_date and room\_type=roomtype);

r number;

count1 number:=0;

begin

open c1;

loop

fetch c1 into r;

exit when c1%notfound;

count1:=count1+1;

dbms\_output.put\_line(r);

end loop;

if count1=0 then

dbms\_output.put\_line('no rooms available');

else

dbms\_output.put\_line(count1||' rooms are available on entered date');

end if;

end;