

PORCEDURE

I. Create customer

```
create or replace procedure creaet_customer(
    name IN customer.cus_name%type,
    ph_no in customer.cus_no%type,
    address in customer.address%type,
    uname in customer.user_name%type,
    pass customer.password%type) as card_id number;

id customer.cus_id%type;

begin
    card_id := card_seq.nextval;
    select cus_id into id from customer where user_name = uname;
    if id>0 then
        dbms_output.put_line('User is alread exists....');
    end if;
    exception
        when no_data_found then
            id :=cus_seq.nextval;
            insert into card values(card_id,0,'on');
            insert into customer values(id,name,ph_no,card_id,pass,uname,sysdate,address);
            dbms_output.put_line('Wlcome ' || name || ' your account is been created....');
            dbms_output.put_line(' your account id is....' || id);
end;
```

II. Create Employee

```
create or replace procedure create_employee(
    name IN employee.emp_name%type,
    ph_no in employee.emp_no%type,
    address in employee.emp_address%type,
    uname in employee.user_name%type,
    pass employee.password%type,
    b_name employee.branch_name%type
) as id employee.emp_id%type;

branch_rebranch%rowtype;
e_invalid_branch exception;

begin
    select emp_id into id from employee where user_name = uname;
    if id>0 then
        dbms_output.put_line('User is alread exists....');
    end if;
    select * into branch_re from branch where branch_name= b_name;
    if sql%notfound then
        raise e_invalid_branch ;
    end if;
```

exception

```
when no_data_found then
insert into card values(card_id,0,'on');
insert into employee values(
            emp_seq.nextval,
            name,
            address,
            ph_no,
            pass,
            uname,
            5660,
            sysdate,
            b_name );
```

```
dbms_output.put_line('Welcome ' || name || ' your account is been created....');
when e_invalid_branch then
            insert into branch(branch_name) values(b_name);
end;
```

III. Create_account

```
create or replace procedure create_accoutn (type in varchar) as fname customer.cus_name%type;
p_nocustomer.cus_no%type;
address customer.address%type;
unamecustomer.user_name%type;
pass customer.password%type;
add customer.address%type;
begin
    if type = 'customer' then
        Fname := '&Fname' ;
        uname := '&user_name';
        pass:= '&password';
        p_no := &phoneno;
        add := '&address';
        cre_cus(fname,p_no,add,uname,pass);
    end if;
end;
```

IV. Delete_customer

```
create or replace procedure delete_customer (id in customer.cus_id%type) as
card number;
uname varchar2(20);
r_id number;
begin
    uname := user_validity(id);
    if uname=null then
        dbms_output.put_line('account id dpoesntoexsists');
    else
```

```

        select card_id into card from customer where cus_id=id;
        select rent_id into r_id from rent where cus_id = id;
        delete from rent where cus_id=id;
        delete from customer where cus_id =card;
        delete from card where c_id=card;
        dbms_output.put_line(uname || ' your Account id : ' || id || ' is been deleted...' );
    end if;
exception
    when no_data_found then
        delete from customer where cus_id=id;
        delete from card where c_id=card;
        dbms_output.put_line(uname || ' your Account id : ' || id || ' is been deleted...' );
end;

```

V. Delete_employee

```

create or replace procedure delete_employee (id employee.emp_id%type) as
uname employee.user_name%type;
begin
    select user_name into uname from employee where emp_id=id;
    if sql%notfound then
        dbms_output.put_line(id || ' doesnotexist');
    else
        delete from employee where emp_id =id;
        dbms_output.put_line(uname || ', your id ' || id || ' is been deleted');
    end if;
end;

```

VI. Rent_pro

```

create or replace procedure rent_pro (
    name in varchar2,
    customer_id in customer.cus_id%type) as id number;
card_id number;
card_fine_rup number;
valid varchar2(10);
begin
    valid := user_validity(customer_id);
    if valid is null then
        dbms_output.put_line(customer_id || ' does not exists....Please check again');
        return;
    else
        card_id :=find_card_id(customer_id);
        if card_id is null then
            dbms_output.put_line(customer_id || ' does not exists....Please check again');
            return;
        else
            card_fine_rup := card_fine(card_id);
            if card_fine_rup> 0 then

```

```

        return;
    else
        dbms_output.put_line(customer_id || ' everything is perfect');
    end if;
end if;
end if;
end ;

```

VII. Issue_Book_Video

```

create or replace procedure issue_book_video (
    name in varchar2,
    customer_id in customer.cus_id%type) as flag varchar2(6);

card_id number;
book number :=0;
video number :=0;
avail number ;
item_id number;
begin
    flag :=fine_from_cus_id(customer_id);
    if flag = 'true' then
        item_id :=book_find(name,avail);
        book :=1;
        if item_id< 0 then
            item_id :=video_find(name,avail);
            video :=1;
        end if;
        if item_id> 0 then
            if avail >0 then
                card_id :=find_card_id(customer_id);
                update card set issue_date=sysdate where c_id=card_id;
                update card set return_date=sysdate+15 where c_id=card_id;
                insert into rent values (
                    sysdate,
                    sysdate+15,
                    customer_id,
                    rent_seq.nextval,
                    item_id);
                if book>0 then
                    update book set availability = availability-1 where book_id=item_id;
                    dbms_output.put_line('video is been issued by ' || item_id || ' is ' || name);
                elsif video >0 then
                    update video set availability =availability-1 where video_id=item_id;
                    dbms_output.put_line('video is been issued by ' || item_id || ' is ' || name);
                else
                    dbms_output.put_line('Something is wrong');
                    return;
                end if;
            else

```

```

        dbms_output.put_line(name || ' right now is not available');
        return;
    end if;
    else
        dbms_output.put_line(name || ' there is not such item available');
        return;
    end if;
else
    return;
end if;
end ;

```

VIII. Book_Add

```

create or replace procedure book_add(
    name in book.b_name %type,
    total in book.total_book%type) as avail number;
book_id number;
begin
    book_id :=book_find(name,avail);
    if book_id>0 then
        dbms_output.put_line('Book is already in database with id ' || book_id);
        dbms_output.put_line('Total availability is ' || total);
    else
        insert into book values (book_seq.nextval,total,total,upper(name || ' :The Book'),sysdate);
        dbms_output.put_line('Book is successfully added....');
    end if;
end;

```

IX. Video_Add

```

create or replace procedure video_add(name in video.v_name %type,total in
video.total_video%type) as
avail number;
video_id number;
begin
    video_id :=book_find(name,avail);
    if video_id>0 then
        dbms_output.put_line('video is already in database with id ' || video_id);
        dbms_output.put_line('Total availability is ' || total);
    else
        insert into video values (video_seq.nextval,total,total,upper(name || ' :The video'),sysdate);
        dbms_output.put_line('Video is successfully added....');
    end if;
end;

```

X. Book_Del

```

create or replace procedure book_del(name in book.b_name %type) as
avail number;
b_id number;
begin

```

```

b_id :=book_find(name,avail);
if b_id>0 then
    if avail > 0 then
        delete from book where book_id =b_id;
        dbms_output.put_line(name || ' is been deleted');
        dbms_output.put_line('Total book deleted is/are : ' || avail);
    elsif avail < 0 then
        dbms_output.put_line('Availability of book' || name || ' is/are:' || avail);
    end if;
else
    dbms_output.put_line('There is no book with name ' || name || ' is in database');
end if;
end;

```

XI. Video_Del

```

create or replace procedure video_del(name in video.v_name %type) as
avail number;
v_id number;
begin
v_id :=video_find(name,avail);
if v_id>0 then
    if avail > 0 then
        delete from video where video_id =v_id;
        dbms_output.put_line(name || ' is been deleted');
        dbms_output.put_line('Total video deleted is/are : ' || avail);
    elsif avail < 0 then
        dbms_output.put_line('Availability of video' || name || ' is/are:' || avail);
    end if;
else
    dbms_output.put_line('There is no video with name ' || name || ' is in database');
end if;
end;

```