**Start service for a house:**

startService(houseID  in number, start\_date in date, customerId in number)

*Algorithm:* Check if the service has been started on the house.

**StartService** will create an account for the customer on the house if the customer does not have an account on that house earlier or otherwise the old account will be used. A flat connection fee will be also charged on successful creation of account and then a message will be displayed to message table stating service is scheduled to start on the start date and new account number.

*Input:* 1) House Id

                  2)Start date

                  3)Customer Id

*Output:* Message will be displayed stating the date the service is to be started and the new account number.

Create Or Replace Procedure Startservice (Houseid In Number,Startdate In Date,Customerid In Number)

Is

count\_acc Number;

v\_connection\_fee rate.connection\_fee%Type;

v\_cname customer.cname%type;

Begin

SELECT connection\_fee INTO v\_connection\_fee FROM rate WHERE  EXTRACT(MONTH FROM rate\_date) = EXTRACT(MONTH FROM Startdate) AND

EXTRACT(YEAR FROM rate\_date) = EXTRACT(YEAR FROM Startdate)  ;

Select Count(1) Into count\_acc From Account a,Customer c

WHERE a.cid = c.cid AND a.cid = Customerid AND a.hid = Houseid;

select cname into v\_cname from customer where cid = Customerid;

/\*If account existes\*/

IF count\_acc = 1 THEN

Update account Set Status = 'Active',account\_balance = account\_balance + v\_connection\_fee ,start\_date =Startdate Where cid = Customerid AND hid = Houseid ;

dbms\_output.put\_line('Account for '||v\_cname||'has been activated on house id '||Houseid );

/\*If account doesnot existes\*/

ELSIF count\_acc = 0 THEN

INSERT INTO account VALUES (seq\_account.nextVal,Customerid,Houseid,'Active',v\_connection\_fee,sysdate,NULL);

dbms\_output.put\_line('New account has been created for '||v\_cname||'on house '||Houseid||'. Account number is '|| seq\_account.nextVal);

commit;

dbms\_output.put\_line( 'Account number is '|| (seq\_account.nextVal - 2));

END IF;

EXCEPTION

when no\_data\_found then  dbms\_output.put\_line('Data for the specified account does not exists');

when too\_many\_rows then dbms\_output.put\_line('Too many rows');

when others then dbms\_output.put\_line('Error in data.');

End;