**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Content** | **Page no.** | **Remark** |
| **1.** | **Abstract** |  |  |
| **2.** | **Procedures used and Command reference** |  |  |
| **3.** | **Schema and table description** |  |  |
| **4.** | **Creating Tables** |  |  |
| **5.** | **Procedures** |  |  |
| **6.** | **Triggers** |  |  |
| **7.** | **Conclusion** |  |  |

**ABSTRACT**

My project entitled “**Electricity Board System**” aims is to generate electricity bill with all the charges and penalty. Manual system that is employed is extremely laborious and quite inadequate. It only makes the process more difficult and hard. The aim of my project is to develop a system that is meant to partially computerize the work performed in the Electricity Board like generating monthly electricity bill, record of consuming unit of energy, store record of the customer and unpaid record. It will simplifies the task and reduce the paper work.

The problem with current system is that the client uses MS Excel, and maintains their records, however it is not possible them to share the data from multiple system in multi user environment, there is lot of duplicate work, and chance of mistake. When the records are changed they need to update each and every excel file. There is no option to find and print previous saved records. This Electricity Bill Payment System is used to overcome the entire problem which they are facing currently, and making complete atomization of manual system to computerized system.

**Procedures Used :**

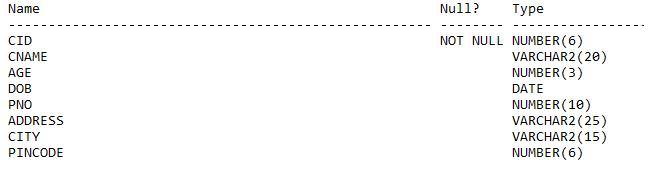
* NEW\_CONSUMER
* SHOW\_CONSUMER
* GENERATE\_BILL
* SHOW\_BILL
* PAY\_BILL
* DELETE\_CONSUMER

**Command Reference :**

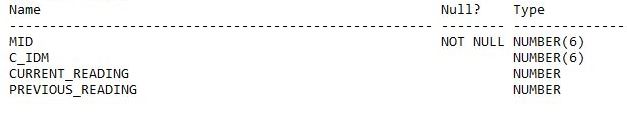
1. For Inserting New consumer
2. To Show consumer detail - EXEC SHOW\_CONSUMER;
3. To Generate Bill - EXEC GENERATE \_BILL;
4. To Show Bill - EXEC SHOW\_BILL;
5. For Payment of bill - EXEC PAY\_BILL;
6. To Delete consumer from record - EXEC DELETE\_CONSUMER;

**Schema and Tables Description :**

**Table : CONSUMER**



**Table : METER**



**Table : AMOUNT**



**Creating Tables**

**1. TABLE CONSUMER**  
create table consumer  
(  
cid number(6) primary key,  
cname varchar2(20),  
age number(3),  
dob date,  
pno number(10),  
address varchar(25),  
city varchar(15),  
pincode number(6)  
);

**2. TABLE METER**  
create table meter  
(  
mid number(6) primary key,  
c\_idm number(6),  
current\_reading number,  
previous\_reading number,  
foreign key(c\_idm) references consumer(cid)  
);

**3. TABLE AMOUNT**  
create table amount  
(  
bid number(7) primary key,  
c\_ida number(6),  
m\_id number(6),  
units number,  
amount number,  
due number,  
last\_date date,  
status char(6),  
foreign key(m\_id) references meter(mid),  
foreign key(c\_ida) references consumer(cid)  
);

**PROCEDURES**

**A. To add new consumer :**

declare

billno number(7);

cno consumer.cid%type;

name consumer.cname%type;

ag consumer.age%type;

birthdate consumer.dob%type;

ph consumer.pno%type;

add consumer.address%type;

ct consumer.city%type;

pin number;

meterno meter.mid%type;

begin

cno:=&cno;

name := '&name';

birthdate:='&birthdate';

ph := &ph;

add := '&add';

ct := '&ct';

pin := &pin;

meterno:= &meterno;

ag:=round((sysdate-birthdate)/365);

insert into consumer values (cno, name,ag, birthdate, ph, add, ct, pin);

insert into meter values (meterno,cno,0,0);

insert into amount values (billno,cno,meterno,0,0,0,null,null);

end;

/

**B. To show consumer detail :**

create or replace procedure show\_consumer  
as  
 cno consumer.cid%type;   
 name consumer.cname%type;   
 ag consumer.age%type;  
 birth consumer.dob%type;  
 mobile consumer.pno%type;   
 add consumer.address%type;  
 ct consumer.city%type;  
 pin number;

begin  
 cno:=&cno;

select cname,dob, age, pno, address,city,pincode into name,birth,ag,mobile,add,ct,pi  
from consumer  
where consumer.cid = cno;

dbms\_output.put\_line('================================'); dbms\_output.put\_line('consumer details');  
 dbms\_output.put\_line('================================');  
 dbms\_output.put\_line('consumer no : '||cno);   
 dbms\_output.put\_line('consumer name : '||name);  
 dbms\_output.put\_line('consumer age : '||ag);   
 dbms\_output.put\_line('date of birth : '||birth);  
 dbms\_output.put\_line('contact no. : '||mobile);   
 dbms\_output.put\_line('address : '||add);   
 dbms\_output.put\_line('city : '||ct);   
 dbms\_output.put\_line('pincode : '||pin);   
 dbms\_output.put\_line('================================');   
 end;   
 /  
 --After giving input

SQL> EXEC SHOW\_CONSUMER;

**C. To generate bill :**

create or replace procedure generate\_bill  
as  
id consumer.cid%type;  
cr meter.current\_reading%type;  
pr meter.previous\_reading%type;  
ld amount.last\_date%type;  
c number;  
m number;  
amt number;  
unt number;  
begin  
id:=&id;  
pr:= &pr;  
cr:= &cr;  
ld:= '&ld';  
 unt:=cr-pr;  
 if unt<15 then  
 amt:=150;  
 else  
 amt:=unt\*7;  
 end if;

select count(\*) into c from consumer where cid = id;  
if c>0 then  
 begin  
 select count(\*) into m from meter where c\_idm = id;  
 if m>0 then  
 begin  
 update meter set current\_reading=cr,previous\_reading=pr  
 where c\_idm=id;  
update amount set units=unt,amount=amt,due=amt,status='n',last\_date=ld  
where c\_ida=id;   
 end;  
 else  
 dbms\_output.put\_line('meter id not found');  
 end if;  
 end;  
else   
 dbms\_output.put\_line('consumer id not found');  
end if;  
end;  
/  
 --After giving input

SQL> EXEC GENERATE\_BILL;

**D. To show bill :**

create or replace procedure show\_bill  
as  
id consumer.cid%type;  
bno number(7);  
cno consumer.cid%type;  
name consumer.cname%type;  
add consumer.address%type;  
ct consumer.city%type;  
pin consumer.city%type;  
meterno meter.mid%type;  
cr meter.current\_reading%type;  
pr meter.previous\_reading%type;  
ld amount.last\_date%type;  
unt amount.units%type;  
amt number;  
c number;  
m number;  
begin  
id:=&id;  
 select count(\*) into c from consumer where cid = id;

if c>0 then  
 begin  
 select count(\*) into m from amount where c\_ida = id and last\_date is not null;  
 if m>0 then  
 begin  
 select cid,cname,address,city,pincode into cno,name,add,ct,pin from consumer  
 where cid=id;  
 select mid,current\_reading,previous\_reading into meterno,cr,pr from meter  
 where c\_idm=id;  
 select bid,units,amount,last\_date into bno,unt,amt,ld from amount where c\_ida=id;  
 dbms\_output.put\_line('=================================');  
 dbms\_output.put\_line(' electricity bill');  
 dbms\_output.put\_line('=================================');  
 dbms\_output.put\_line('bill no. : '||bno);  
 dbms\_output.put\_line('consumer id : '||cno);  
 dbms\_output.put\_line('name : '||name);  
 dbms\_output.put\_line('address : '||add);  
 dbms\_output.put\_line('city : '||ct);  
 dbms\_output.put\_line('zipcode : '||pin);  
 dbms\_output.put\_line('meter no. : '||meterno);  
 dbms\_output.put\_line('previous month reading : '||pr);  
 dbms\_output.put\_line('current month reading : '||cr);  
 dbms\_output.put\_line('units consumed : '||unt);  
 dbms\_output.put\_line('amount : '||amt);  
 dbms\_output.put\_line('last date of payment : '||ld);  
 dbms\_output.put\_line('=================================');  
 end;  
 else  
 dbms\_output.put\_line('bill not generated yet.');  
 end if;  
end;  
else  
 dbms\_output.put\_line('consumer id not found');  
end if;  
end;  
/  
--After giving input  
SQL> EXEC SHOW\_BILL;

**E. To pay bill :**

create or replace procedure pay\_bill   
as   
cno consumer.cid%type;  
name consumer.cname%type;  
amnt number;  
amt number;  
du number;   
ld date;   
sts char(6);   
days number;   
begin  
cno:=&cno;  
amt:=&amt;  
select cname into name from consumer where cid=cno;  
select due, status, last\_date,amount into du, sts, ld, amnt from amount where c\_ida = cno;   
if sts = 'n' or sts = 'n' then   
 days := round(sysdate-ld);  
 if days < 1 then  
 if amt-du=0 then  
 update amount set due=0, status='y' where c\_ida=cno;  
 dbms\_output.put\_line('=============================');   
 dbms\_output.put\_line(' bill receipt');   
 dbms\_output.put\_line('=============================');   
 dbms\_output.put\_line('id : '||cno);   
 dbms\_output.put\_line('name : '||name);   
 dbms\_output.put\_line('pay date : '||sysdate);   
 dbms\_output.put\_line('amount paid : '||amt);  
 dbms\_output.put\_line('==================================');   
 else  
 dbms\_output.put\_line('not sufficient amount entered');  
 end if;

elsif days > 0 and days < 31 then   
 update amount set due=amnt+100 where c\_ida=cno;  
 select due into du from amount where c\_ida=cno;  
 if amt-du=0 then  
 update amount set due=0, status='y' where c\_ida=cno;  
 dbms\_output.put\_line('=============================');   
 dbms\_output.put\_line(' bill receipt');   
 dbms\_output.put\_line('=============================');   
 dbms\_output.put\_line('id : '||cno);   
 dbms\_output.put\_line('name : '||name);   
 dbms\_output.put\_line('pay date : '||sysdate);  
 dbms\_output.put\_line('amount paid with penalty : '||amt);  
 dbms\_output.put\_line('==================================');   
 else  
 dbms\_output.put\_line('not sufficient amount entered, pay with penalty of  
 rs.100, due date exceeded!!!');  
 end if;   
 else   
 dbms\_output.put\_line('=======================');   
 dbms\_output.put\_line('your connection has been suspended!!!');   
 dbms\_output.put\_line('-------------------------------------');  
 dbms\_output.put\_line('contact office.');   
 dbms\_output.put\_line('=======================');   
 end if;   
end if;

if sts = 'y' or sts = 'y' then   
 dbms\_output.put\_line('=================================');   
 dbms\_output.put\_line('you have already paid the bill...');   
 dbms\_output.put\_line('=================================');   
end if;   
end;   
/  
--After giving input  
SQL> EXEC PAY\_BILL;

**F. To delete record of consumer :**

create or replace procedure delete\_consumer   
as  
cno consumer.cid%type;  
begin  
cno:=&cno;  
delete from amount where c\_ida=cno;  
delete from meter where c\_idm=cno;  
delete from consumer where cid=cno;  
end;  
/  
--After giving input  
SQL> EXEC DELETE\_CONSUMER;

**TRIGGERS**

**A. To auto generate bill\_id and store in amount table :**

create or replace trigger vautogbid  
before insert on amount   
for each row  
declare   
cnt number(7);  
cursor bill is select \* from amount;  
data amount%rowtype;  
begin  
cnt:=12101;  
 for data in bill loop  
 cnt:=cnt+1;  
 end loop;  
 :new.bid:=cnt;  
end;  
/

**B. To store mobile number of 10 digit only :**

create or replace trigger vmob  
before insert or update  
on consumer  
for each row  
begin  
 if :new.pno<7000000000 or :new.pno>9999999999 then  
 raise\_application\_error(-20120,'mobile no. entry is invalid');  
 end if;  
end;  
/

**C. Date of Birth should not exceed current date :**

create or replace trigger vdob  
before insert or update  
on consumer  
for each row  
begin  
 if :new.dob>sysdate then  
 raise\_application\_error(-20121,'date of birth must be less then current date');  
 end if;  
end;  
/

**D. Age should be greater than 18 for new consumer :**

create or replace trigger vage  
before insert or update  
on consumer  
for each row  
begin  
 if :new.age<18 then  
 raise\_application\_error(-20121,'age must be greater than 18');  
 end if;  
end;  
/

**CONCLUSION**

This project can be used by electricity board billing counters. This can be easily  
combined with any frontend application in order to get GUI interface. This project  
can be developed in future with other added features.