Project Team 12 - Electricity Billing System

Team Members:

1) Anchal Atlani
2) Anushka Kamalapurkar
3) Sagar Kulkarni
4) Gaurav Mahadik

Context:

The Electricity Billing System will give a single platform to customers through which they can pay and view their bills of any service provider listed in the system

Objective:

The main objective of the project is to develop a database system that will help a customer to register under a service provider, view and pay electricity bill. Apart from this a customer can also change service provider, upgrade to use solar net metering, view historic payments, peak and off peak hours and billing cycle.

Scope:

1. **Customer Registration**: Customer register under the system under a specific service provider for smart meter or solar net meter.
2. **View and Pay bills**:
    a. Customer can pay and view historic bills.
    b. Customer can check the unit consumption of the selected period.
    c. Customer can view the off peak and peak hours for the area.
3. **Stop or Change the connection**: Customer can switch to other service provider and customer can deregister from the system.
4. **Admin can add Service Provider/ Customer:** Admin can add any new service provider or new customer.
5. **Admin modify the tariff rates:**
    a. Admin can modify the tariff rates of any service provider.

b. Admin can modify the peak and off peak hours corresponding to any service provider in any area.

ENVIRONMENT SETUP:

We are using MySQL Database management System. For making connection to MySQL server we are using mysql-workbench-community-8.0.12-winx64 client. For user interface

and database connectivity we are using javax.swing api and jdbc.odbc 7.0 version. We will use Java 8 for our project.

HIGH LEVEL REQUIREMENTS:

Initial User Roles:

Role	**Description**
Customer	New customer can register under a service provider for the electricity billing system. Existing customer can view and pay electricity bills, change service provider, can see unit consumption, can see peak and off peak hours for any specific area.
Admin	System admin who can modify the tariff rate, Register/Deregister a customer, Register/Deregister a Service Provider.

Initial user story descriptions:

Story ID	**Story Description**
US1	As a customer I want to register under a service provider.
US2	As a registered customer I want to view bill of any selected month.
US3	As a registered customer I want to pay the bills.
US4	As a customer I want to view the tariff rates of any area offered by various service providers already listed.
US5	As a customer I want to change a service provider.
US6	As a customer I want to deregister.
US7	As an admin I want to change the tariff rates of service provider.
US8	As an admin I want to add a customer in the electricity billing system.
US9	As an admin I want to remove a customer from the electricity billing system.
US10	As an admin I want to add a service provider.

High level conceptual design:

Entities:

Customer
Admin
Service_Provider

Electricity_Bill
Tariff_Details

Relationships:

Customer has connection under a single Service_Provider.
Customer pays Electricity_Bill.
Customer changes a Service_Provider.
Admin updates Tariff_Details.
Service_Provider maintains area specific Tariff_Details.

Sprint 1

REQUIREMENTS:

Story ID	Story Description
US1	As a customer I want to register under a service provider.
US2	As a customer I want to login into the system.
US3	As a registered customer I want to view all electricity bills.
US4	As an admin I want to view all registered customers.
US5	As a customer I want to view number of units consumed.
US6	As a registered customer I want to view bill of any selected month.
US7	As a registered customer I want to pay the bills.
US8	As a customer I want to view the tariff rates of any area offered by various service providers already listed.
US9	As a customer I want to change a service provider.
US10	As a customer I want to deregister.
US11	As an admin I want to change the tariff rates of service provider.
US12	As an admin I want to add a customer in the electricity billing system.
US13	As an admin I want to remove a customer from the electricity billing system.
US14	As an admin I want to add a service provider.

CONCEPTUAL DESIGN

Entity:**Customer**

Attributes:

customer_id

login_email_id

login_password

ssn

name[composite]

first_name

last_name

address[composite]

address_line_1

address_line_2

city

state

zipcode

phone_number[multivalued]

service_provider_name

register_date

meter_type

customer_id- this customer id is generated whenever a user registers under a service provider.

Entity: **Electricity_Billing_Details**

Attributes:

electricity_bill_id

bill_generated_date

bill_due_date

billing_cycle_start_date

number_of_billing_days

payment_amount

amount_due

last_meter_reading

present_meter_reading

Note:

last_meter_reading and present_meter_reading are in kWH

Entity: **Admin**

Attributes:

- login_id[Email id of Admin]
- login_password
- name[composite]
  - first_name
  - last_name
- address[composite]
  - address_line_1
  - address_line_2
  - city
  - state
  - zipcode
- phone_no[multivalued]

Relationship: **Customer** has **Electricity_Billing_Details**

Cardinality: One to Many

Participation:

Customer has partial participation

Electricity_Billing_Details have total participation

LOGICAL DESIGN

Table: **Customer**

Columns:

- customer_id
- login_email_id
- login_password
- ssn
- first_name
- last_name
- address_line_1
- address_line_2
- city
- state
- country
- zipcode
- phone_number_primary

phone_number_secondary
service_provider_name
register_date
meter_type

Table: **Electricity_Billing_Details**

Columns:

electricity_bill_id

customer_id [foreign key;references customer_id of Customer table]
bill_generated_date
bill_due_date
billing_cycle_start_date
number_of_billing_days
payment_amount
amount_due
last_meter_reading
present_meter_reading

Note:

last_meter_reading and present_meter_reading are in kWH

Table: **Admin**

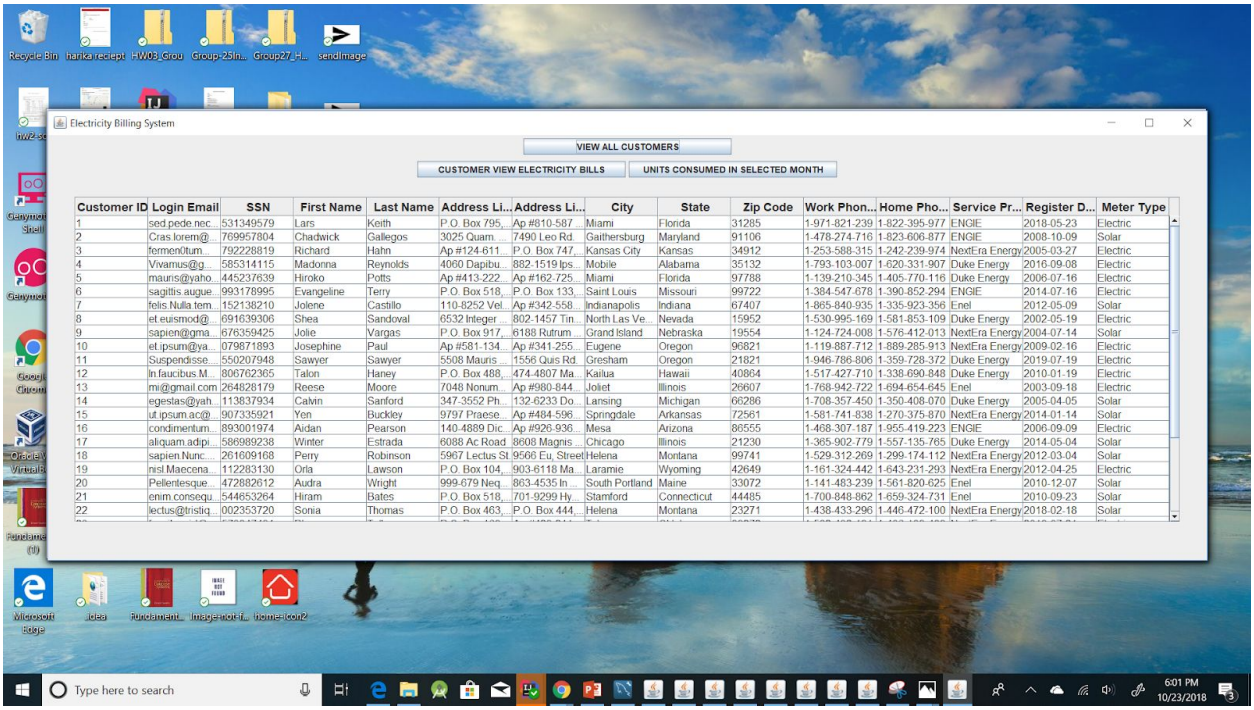
Columns:

login_id[Email id of Admin]

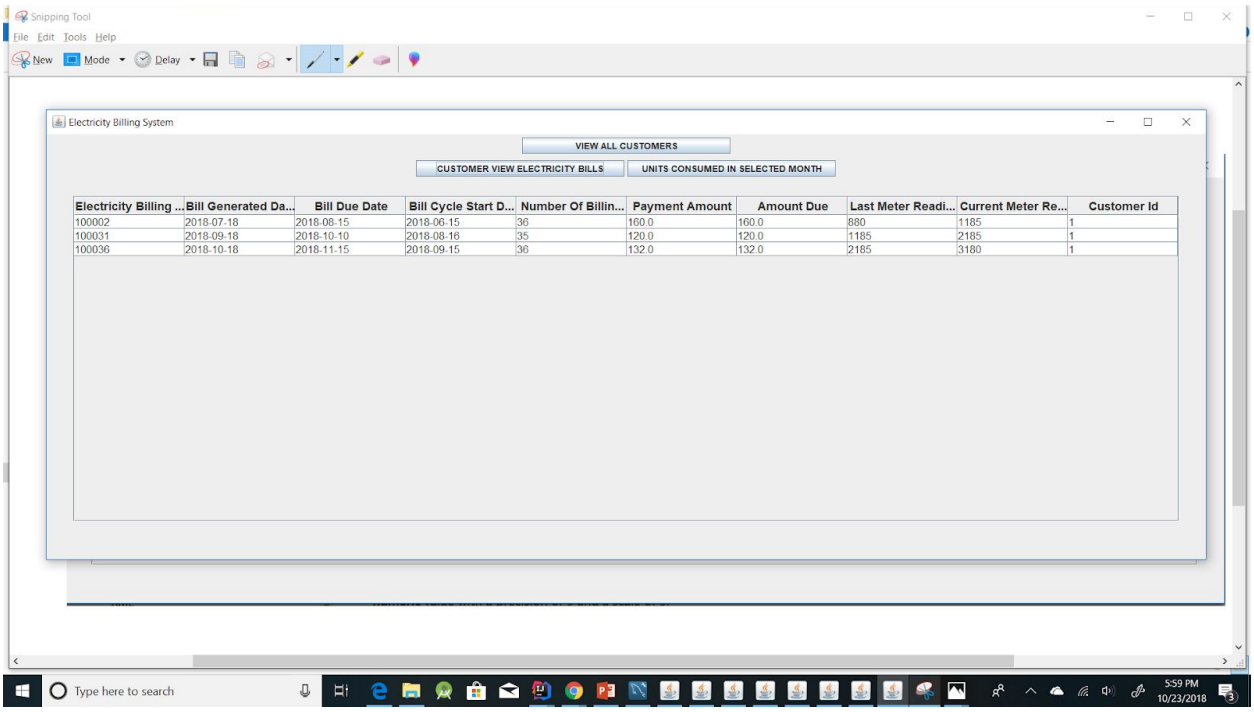
login_password
first_name
last_name
address_line_1
address_line_2
city
state
zipcode
phone_no_primary
phone_no_secondary

Queries:

1. SELECT * FROM Customer;

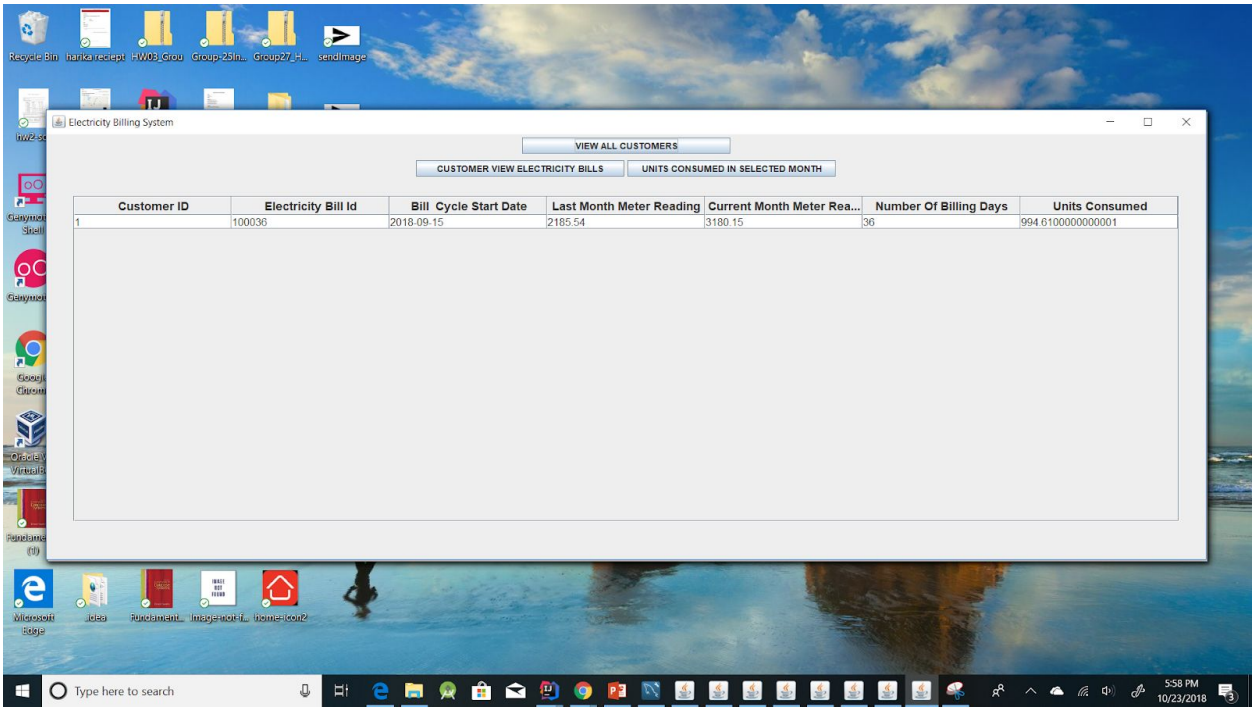


2. SELECT * FROM Electricity_Billing_Details where customer_id=1;



3.

```
SELECT Electricity_Billing_Details.customer_id,
Electricity_Billing_Details.electricity_bill_id,
Electricity_Billing_Details.billing_cycle_start_date, last_meter_reading,
Electricity_Billing_Details.present_meter_reading,
Electricity_Billing_Details.number_of_billing_days,
(Electricity_Billing_Details.present_meter_reading-Electricity_Billing_Details.last_meter_reading) as Units_Consumed FROM Electricity_Billing_Details  where customer_id=1 AND MONTH(billing_cycle_start_date)=9 AND YEAR(billing_cycle_start_date)=2018;
```


Electricity Billing System
VIEW ALL CUSTOMERS
CUSTOMER VIEW ELECTRICITY BILLS
UNITS CONSUMED IN SELECTED MONTH
Customer ID
Electricity Bill Id
Bill Cycle Start Date
Last Month Meter Reading
Current Month Meter Rea...
Number Of Billing Days
Units Consumed
1
100036
2018-09-15
2185.54
3180.15
36
994.6100000000001
Type here to search
5:58 PM
10/23/2018

Sprint 2

Part 1: Refine requirements:

Story ID	Story Description
US1	As a customer I want to register under a service provider.
US2	As a customer I want to login into the system.
US3	As a registered customer I want to view all electricity bills.
US4	As an admin I want to view all registered customers.
US5	As a customer I want to view number of units consumed.
US6	As a registered customer I want to view bill of any selected month.
US7	As a customer I want to view the tariff rates of any area offered by various service providers already listed.
US8	As a customer I want to change a service provider.
US9	As a customer I want to deregister.
US10	As an admin I want to change the tariff rates of service provider.
US11	As an admin I want to add a customer in the electricity billing system.
US12	As an admin I want to remove a customer from the electricity billing system.
US13	As an admin I want to add a service provider.
US14	As a registered customer I want to pay the bills.

Part 2: Conceptual design

Entity: **Customer**
Attributes:

- customer_id
- login_email_id
- login_password
- ssn
- name[composite]
  - first_name
  - last_name
- address[composite]
  - address_line_1
  - address_line_2
  - zipcode
  - state
  - city
- phone_number[multivalued]
- service_provider_name
- register_date
- meter_type

Entity: **Electricity_Billing_Details**
Attributes:

- electricity_bill_id
- bill_generated_date
- bill_due_date
- billing_cycle_start_date
- number_of_billing_days
- payment_amount
- amount_due

Entity: **Admin**
Attributes:
- login_id[Email id of Admin]
- login_password
- employee_number
- name[composite]
  - first_name
  - last_name
- address[composite]
  - address_line_1
  - address_line_2
  - zipcode
    - state
    - city
- phone_no[multivalued]

Entity: **Service_Provider**
Attributes:
- service_provider_name
- license_number
- license_expiry
- address[composite]
  - address_line_1
  - address_line_2
  - zipcode
  - state
  - city
- head_office_number
- helpline_number
- email_contact

Entity: **Tariff_Details**
Attributes:
- zip_code
- provider_name
- state
- city
- peak_time_charge
- off_peak_charge
- peak_time
- off_peak_time

Entity: **Meter_Reading**
Attribute:

- meter_number
- peak_hour_reading
- off_peak_hour_reading
- meter_reading_date
- total_current_reading
- total_previous_reading

Relationship: **Customer** has **Meter_Reading**
Cardinality: One to One
Participation:
Customer has partial participation
Meter_Reading have total participation

Relationship: **Meter_Reading** generates **Electricity_Billing_Details**
Cardinality: One to One
Participation:
Customer has partial participation
Meter_Reading have total participation

Relationship: **Service_Provider** has **Tariff_Details**
Cardinality: One to many
Participation:
Service_Provider has partial Participation
Tariff_Details have partial Participation

Relationship: **Customer** views **Tariff_Details**
Cardinality: Many to Many
Participation:
Customer has partial participation
Tariff_Details have partial_participation

Relationship: **Admin** adds **Customer**
Cardinality: One to Many
Participation:
Admin has partial Participation
Customer has total participation

Part 3: Logical design

Table: **Customer**
Columns:

customer_id
login_email_id
login_password
ssn
first_name
last_name
address_line_1
address_line_2
zip_code [foreign key; references zip_code of ZipCode_Details Table]
phone_number_primary
phone_number_secondary
service_provider_name
register_date
meter_type

Primary Key Justification: customer_id of each customer will be unique, and it will be assign during registration of customer in system.

Normal form:
Highest normalization level:4NF

Table: **Admin**
Columns:

employee_id
login_id[Email id of Admin]
login_password
first_name
last_name
address_line_1
address_line_2
zip_code [foreign key; references zip_code of ZipCode_Details Table]
phone_no_primary
phone_no_secondary

Primary Key Justification: employee_id of the system admin will be unique.

Normal form:
Highest normalization level:4NF

Table: **Service_Provider**
Column:

provider_name
license_number
office_Address_line_1
office_address_line_2
zip_code[foreign key; references zip_code of ZipCode_Details Table]
head_Office_Number
helpline_Number
email_contact

Primary Key Justification: Name of Service provider is unique

Normal form:
Highest normalization level:4NF

Table: **Tariff_Details**
Columns:
service_provider_name[foreign key; references provider_name in Service_Provider]
zip_code
peak_time_charge
off_peak_charge
peak_time
off_peak_time

Foreign key mapping:
We have decided to map the primary key of Service_Provider as the foreign key in Tariff_Details as there is a one to many relationship in the conceptual design.

Primary Key Justification:
The zip_code and service_provider_name uniquely determines the tariff details of each for any area of specific service provider.

Normal Form:
Highest normalization level:4NF

Table: **Meter_Reading**
Columns:
meter_number
peak_hour_reading
off_peak_hour_reading
meter_reading_date
total_current_reading
total_previous_reading
customer_id[foreign key: references customer_id of Customer Table]

Composite Primary Key:
We have a composite primary key here as only those two columns can uniquely identify the table.

Normal Form:
Highest normalization level: 2NF

Justification: We need composite key here as this is used to record meter readings of each customer for each billing-cycle.

Table: **Electricity_Billing_Details**
Columns:
electricity_bill_id
meter_number [foreign key;references meter_number of Meter_Reading table]
bill_generated_date
bill_due_date
billing_cycle_start_date
number_of_billing_days
payment_amount
amount_due

Primary Key Justification: electricity_bill_id will be unique, as this will be assign whenever new bill is being added into the system.

Normal Form:
Highest normalization level:4NF

Table: **Zipcode_Details**
Columns:
zip_code
city
state

Primary Key Justification: zip_code will be unique for every region and that uniquely determines the state and city and here as or system is for United States so we have not included country column.

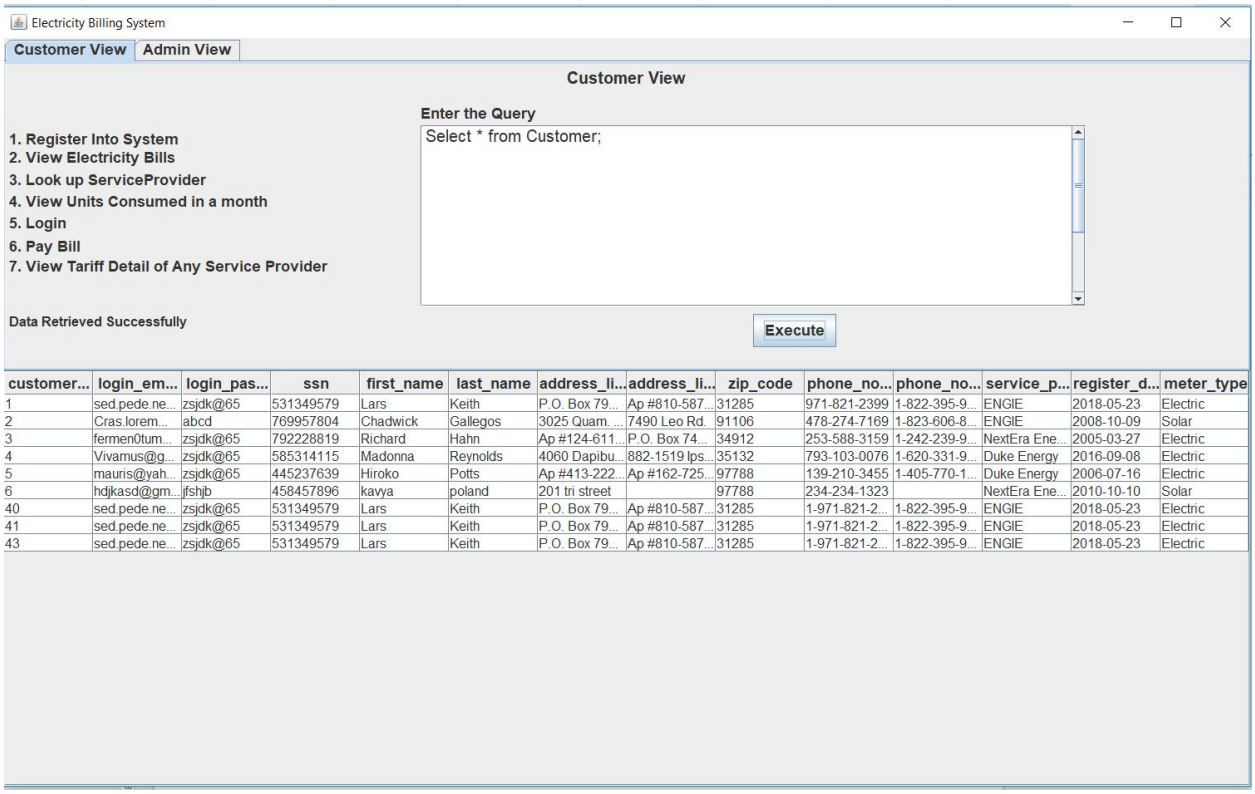
Normal Form:
Highest normalization level:4NF

Queries:

1. Register a Customer

Before

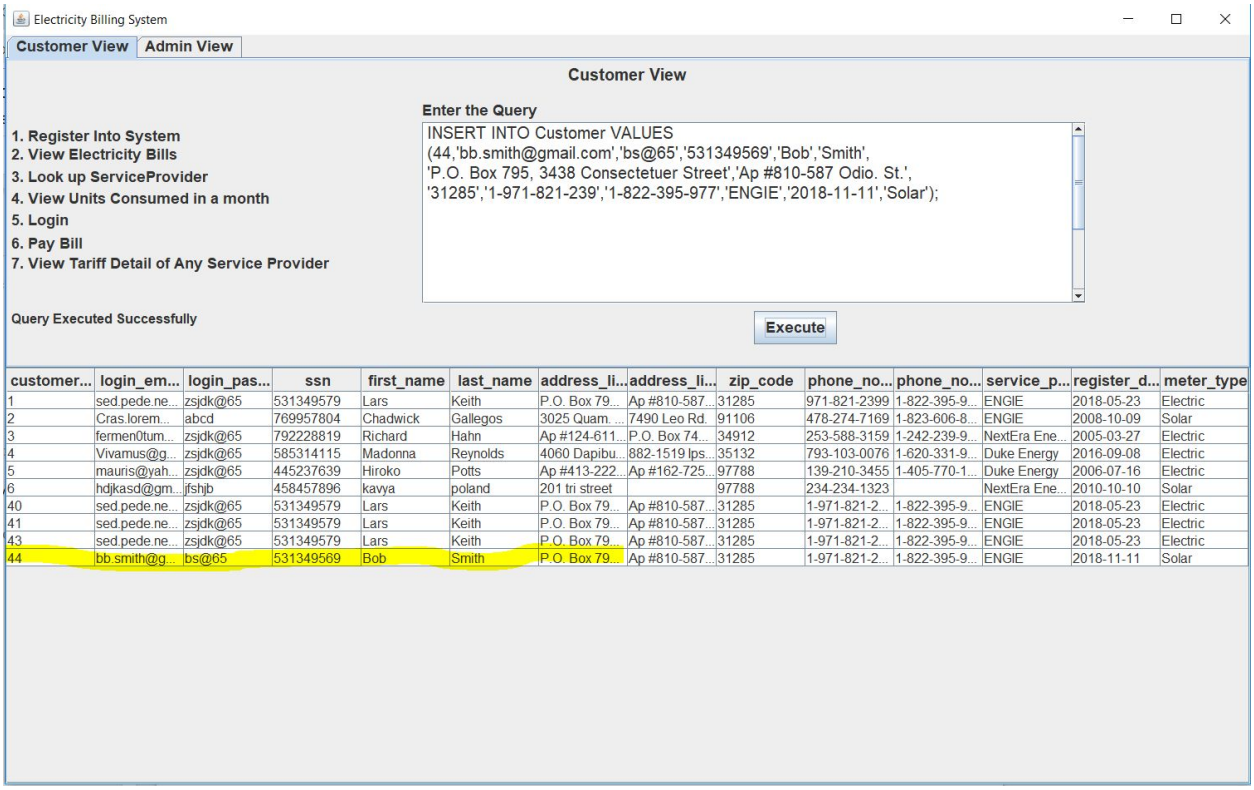
Select * from Customer;



customer...	login_em...	login_pas...	ssn	first_name	last_name	address_li...	address_li...	zip_code	phone_no...	phone_no...	service_p...	register_d...	meter_type
1	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	971-821-2399	1-822-395-9...	ENGIE	2018-05-23	Electric
2	Cras.lorem...	abcd	769957804	Chadwick	Gallegos	3025 Quam. ...	7490 Leo Rd.	91106	478-274-7169	1-823-606-8...	ENGIE	2008-10-09	Solar
3	fermen0tum...	zsjdk@65	792228819	Richard	Hahn	Ap #124-611...	P.O. Box 74...	34912	253-588-3159	1-242-239-9...	NextEra Ene...	2005-03-27	Electric
4	Vivamus@g...	zsjdk@65	585314115	Madonna	Reynolds	4060 Dapibu...	882-1519 Ips...	35132	793-103-0076	1-620-331-9...	Duke Energy	2016-09-08	Electric
5	mauris@yah...	zsjdk@65	445237639	Hiroko	Potts	Ap #413-222...	Ap #162-725...	97788	139-210-3455	1-405-770-1...	Duke Energy	2006-07-16	Electric
6	hdjkasd@gm...	jfshjb	458457896	kavya	poland	201 tri street		97788	234-234-1323		NextEra Ene...	2010-10-10	Solar
40	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric
41	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric
43	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric

After insertion:

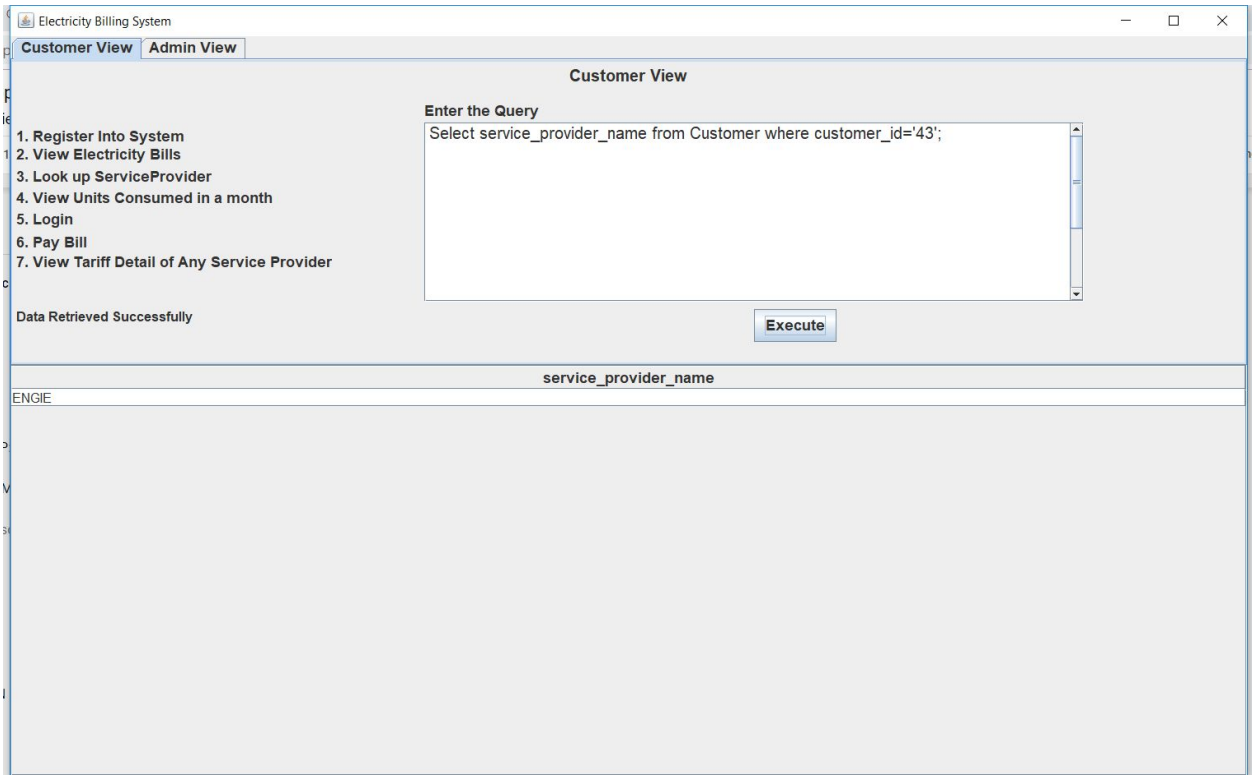
INSERT INTO Customer VALUES
(44,'bb.smith@gmail.com','bs@65','531349569','Bob','Smith',
'P.O. Box 795, 3438 Consectetuer Street','Ap #810-587 Odio. St.',
'31285','1-971-821-239','1-822-395-977','ENGIE','2018-11-11','Solar');



customer...	login_em...	login_pas...	ssn	first_name	last_name	address_li...	address_li...	zip_code	phone_no...	phone_no...	service_p...	register_d...	meter_type
1	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	971-821-2399	1-822-395-9...	ENGIE	2018-05-23	Electric
2	Cras.lorem...	abcd	769957804	Chadwick	Gallegos	3025 Quam. ...	7490 Leo Rd.	91106	478-274-7169	1-823-606-8...	ENGIE	2008-10-09	Solar
3	fermen0tum...	zsjdk@65	792228819	Richard	Hahn	Ap #124-611...	P.O. Box 74...	34912	253-588-3159	1-242-239-9...	NextEra Ene...	2005-03-27	Electric
4	Vivamus@g...	zsjdk@65	585314115	Madonna	Reynolds	4060 Dapibu...	882-1519 Ips...	35132	793-103-0076	1-620-331-9...	Duke Energy	2016-09-08	Electric
5	mauris@yah...	zsjdk@65	445237639	Hiroko	Potts	Ap #413-222...	Ap #162-725...	97788	139-210-3455	1-405-770-1...	Duke Energy	2006-07-16	Electric
6	hdjkasd@gm...	jfshjb	458457896	kavya	poland	201 tri street		97788	234-234-1323		NextEra Ene...	2010-10-10	Solar
40	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric
41	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric
43	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric
44	bb.smith@g...	bs@65	531349569	Bob	Smith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-11-11	Solar

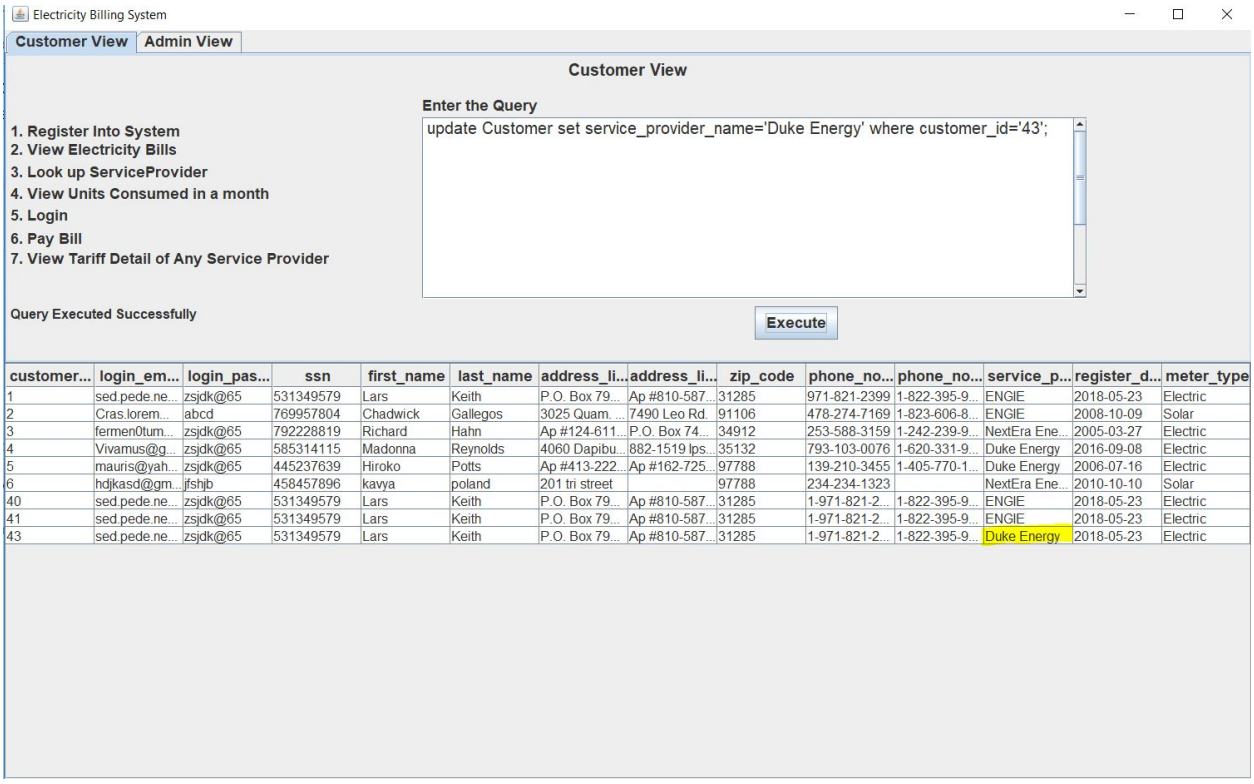
2. Customer changes Service Provider.

Before: Here customer_id we are taking is 43**.**



After updating:

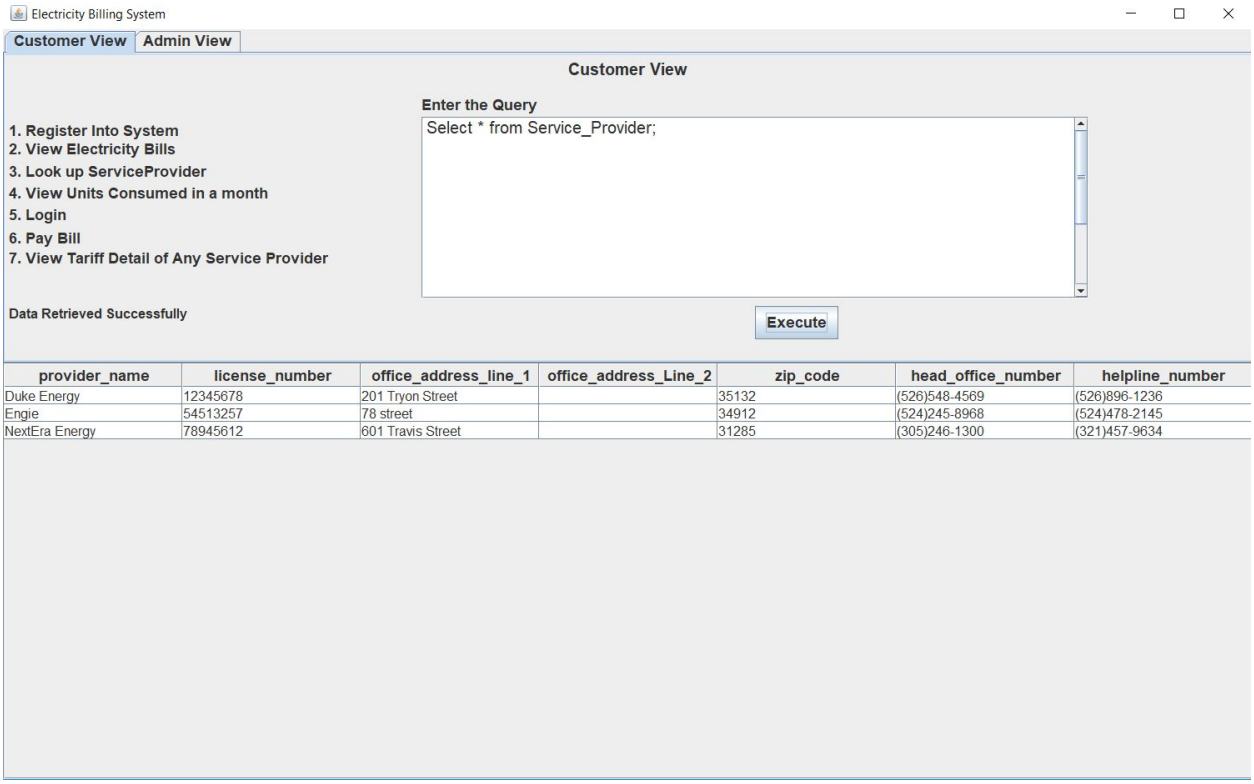
update Customer set service_provider_name='Duke Energy' where customer_id='43';



customer...	login_em...	login_pas...	ssn	first_name	last_name	address_li...	address_li...	zip_code	phone_no...	phone_no...	service_p...	register_d...	meter_type
1	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	971-821-2399	1-822-395-9...	ENGIE	2018-05-23	Electric
2	Cras.lorem...	abcd	769957804	Chadwick	Gallegos	3025 Quam. ...	7490 Leo Rd.	91106	478-274-7169	1-823-606-8...	ENGIE	2008-10-09	Solar
3	fermen0tum...	zsjdk@65	792228819	Richard	Hahn	Ap #124-611...	P.O. Box 74...	34912	253-588-3159	1-242-239-9...	NextEra Ene...	2005-03-27	Electric
4	Vivamus@g...	zsjdk@65	585314115	Madonna	Reynolds	4060 Dapibu...	882-1519 Ips...	35132	793-103-0076	1-620-331-9...	Duke Energy	2016-09-08	Electric
5	mauris@yah...	zsjdk@65	445237639	Hiroko	Potts	Ap #413-222...	Ap #162-725...	97788	139-210-3455	1-405-770-1...	Duke Energy	2006-07-16	Electric
6	hdjkasd@gm...	jfshjb	458457896	kavya	poland	201 tri street		97788	234-234-1323		NextEra Ene...	2010-10-10	Solar
40	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric
41	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	ENGIE	2018-05-23	Electric
43	sed.pede.ne...	zsjdk@65	531349579	Lars	Keith	P.O. Box 79...	Ap #810-587...	31285	1-971-821-2...	1-822-395-9...	Duke Energy	2018-05-23	Electric

3. Look up Service Providers available;

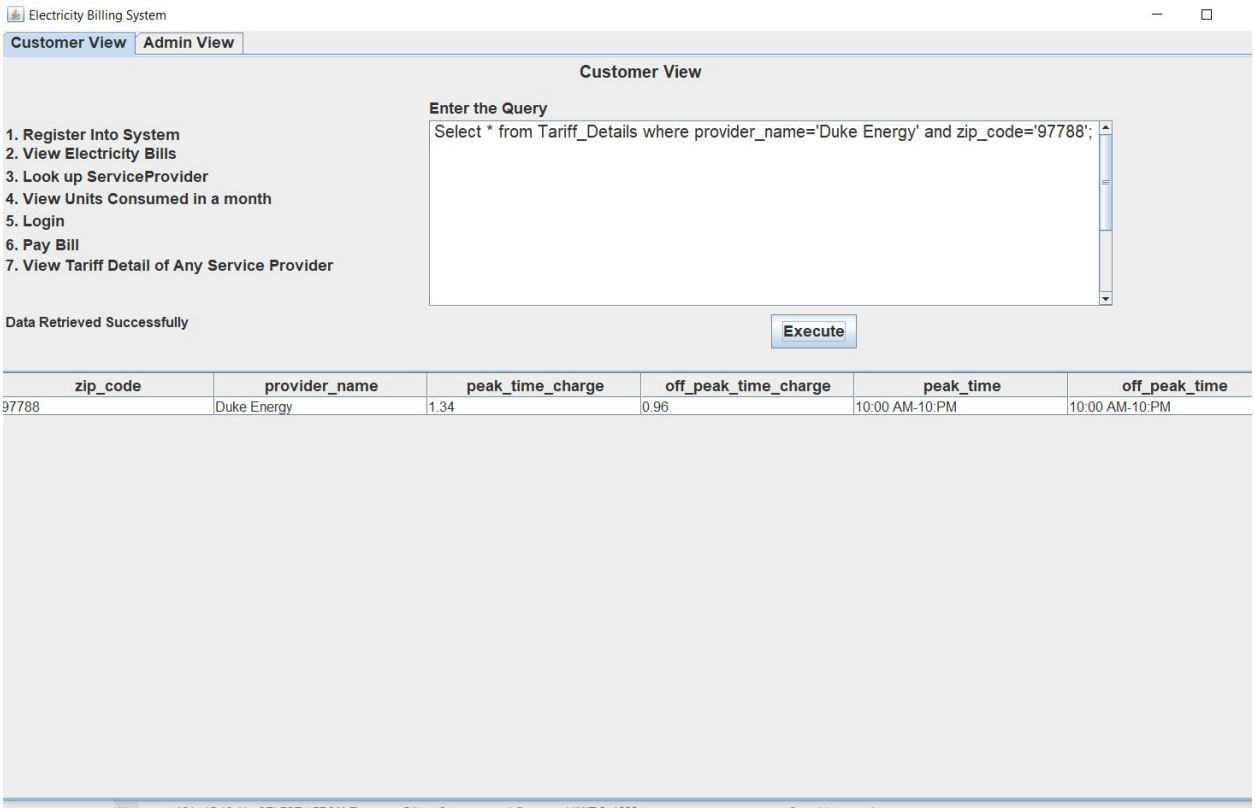
Select * from Service_Provider;



provider_name	license_number	office_address_line_1	office_address_Line_2	zip_code	head_office_number	helpline_number
Duke Energy	12345678	201 Tryon Street		35132	(526)548-4569	(526)896-1236
Engie	54513257	78 street		34912	(524)245-8968	(524)478-2145
NextEra Energy	78945612	601 Travis Street		31285	(305)246-1300	(321)457-9634

4. **Customer can view Tariff_Details of any Service Provider of any specific area.**

Select * from Tariff_Details where provider_name='Duke Energy' and zip_code='97788';



Sprint -3

Part 1: Refine requirements:

Story ID	Story Description
US1	As a customer I want to register under a service provider.
US2	As a customer I want to login into the system.
US3	As a registered customer I want to view all electricity bills.
US4	As an admin I want to view all registered customers.
US5	As a customer I want to view number of units consumed.
US6	As a registered customer I want to view bill of any selected month.
US7	As a customer I want to view the tariff rates of any area offered by various service providers already listed.
US8	As a customer I want to change a service provider.
US9	As a customer I want to deregister.
US10	As an admin I want to change the tariff rates of service provider.
US11	As an admin I want to add a customer in the electricity billing system.
US12	As an admin I want to remove a customer from the electricity billing system.
US13	As an admin I want to add a service provider.
US14	As a registered customer I want to pay the bills.

Part 2: Conceptual design

Entity: **Customer**
Attributes:

- login_email_id
- login_password
- ssn
- name[composite]
  - first_name
  - last_name
- address[composite]
  - address_line_1
  - address_line_2
  - zipcode
    - state
    - city
- phone_number[multivalued]
- service_provider_name
- register_date
- meter_type

Entity: **Electricity_Billing_Details**
Attributes:

- bill_generated_date
- bill_due_date
- billing_cycle_start_date
- number_of_billing_days
- payment_amount
- amount_due

Entity: **Admin**
Attributes:
- login_id[Email id of Admin]
- login_password
- employee_number
- name[composite]
  - first_name
  - last_name
- address[composite]
  - address_line_1
  - address_line_2
  - zipcode
    - state
    - city
- phone_no[multivalued]

Entity: **Service_Provider**
Attributes:
- service_provider_name
- license_number
- license_expiry
- address[composite]
  - address_line_1
  - address_line_2
  - zipcode
    - state
    - city
- head_office_number
- helpline_number
- email_contact

Entity: **Tariff_Details**
Attributes:
- zip_code
  - state
  - city
- provider_name
- peak_time_charge
- off_peak_charge
- peak_time
- off_peak_time

Entity: **Meter_Reading**
Attribute:

- meter_number
- peak_hour_reading
- off_peak_hour_reading
- meter_reading_date
- total_current_reading
- total_previous_reading

Entity: **Payment**
Attributes:

- payment_date
- amount_due
- card_number
- cvv
- card_expiry
- payment_amount

Relationship: **Customer** has **Meter_Reading**
Cardinality: One to One
Participation:
Customer has partial participation
Meter_Reading have total participation

Relationship: **Meter_Reading** generates **Electricity_Billing_Details**
Cardinality: One to One
Participation:
Customer has partial participation
Meter_Reading have total participation

Relationship: **Service_Provider** has **Tariff_Details**
Cardinality: One to many
Participation:
Service_Provider has partial Participation
Tariff_Details have partial Participation

Relationship: **Customer** views **Tariff_Details**
Cardinality: Many to Many
Participation:
Customer has partial participation
Tariff_Details have partial_participation

Relationship: **Admin** adds **Customer**
Cardinality: One to Many
Participation:
Admin has partial Participation
Customer has total participation

Relationship: **Customer** makes **Payment**
Cardinality: One to Many
Participation:
Customer has partial Participation
Payment has total participation

Relationship: **Electricity_Billing_Details** has **Payment**
Cardinality: Many to One
Participation:
Payment has partial Participation
Electricity_Billing_Details has total participation

Justification : Here the cardinality is many to one because the customer is provided with the facility to pay the present bill along with any past bills and the outstanding amount is added to the present amount. So the cardinality of this relationship is Many to One.

Part 3: Logical design

Table: **Customer**
Columns:

- customer_id
- login_email_id
- login_password
- ssn
- first_name
- last_name
- address_line_1
- address_line_2
- zip_code [foreign key; references zip_code of ZipCode_Details Table]
- phone_number_primary
- phone_number_secondary
- service_provider_name[foreign key; references provider_name of Service_Provider Table]
- register_date
- meter_type

Primary Key Justification: customer_id of each customer will be unique, and it will be generated during registration of customer in system.

Foreign key mapping:
We have decided to map the primary key of Service_Provider as the foreign key in Customer as there is a one to one relationship in the conceptual design.

Normal form:
Highest normalization level:4NF

Table: **Admin**
Columns:

- employee_id
- login_id[Email id of Admin]
- login_password
- first_name
- last_name
- address_line_1
- address_line_2
- zip_code [foreign key; references zip_code of ZipCode_Details Table]
- phone_no_primary

phone_no_secondary

Primary Key Justification: employee_id of the system admin will be unique.

Foreign key mapping:
We have decided to map the primary key of Zipcode_Details as the foreign key in Admin.

Normal form:
Highest normalization level:4NF

Table: **Service_Provider**
Column:

provider_name
license_number
office_Address_line_1
office_address_line_2
zip_code[foreign key; references zip_code of ZipCode_Details Table]
head_Office_Number
helpline_Number
email_contact

Primary Key Justification: Name of Service provider is unique

Foreign key mapping:
We have decided to map the primary key of Zipcode_Details table as the foreign key in Service_Provider as there is a one to many relationship in the conceptual design.

Normal form:
Highest normalization level:4NF

Table: **Tariff_Details**
Columns:

service_provider_name[foreign key; references provider_name in Service_Provider]

zip_code
peak_time_charge
off_peak_charge
peak_time
off_peak_time

Primary Key Justification:

The zip_code and service_provider_name uniquely determines the tariff details of each for any area of specific service provider.

Foreign key mapping:
We have decided to map the primary key of Service_Provider as the foreign key in Tariff_Details as there is a one to many relationship in the conceptual design.

Normal Form:
Highest normalization level:4NF

Table: **Meter_Reading**
Columns:
meter_reading_id
meter_number
peak_hour_reading
off_peak_hour_reading
meter_reading_date
total_current_reading
total_previous_reading
customer_id[foreign key: references customer_id of Customer Table]

Primary Key:
We have meter_reading_id as primary key here as every time when admin enetres the meter reading of any particular month then this will uniquely determine the entity.

Foreign key mapping:
We have decided to map the primary key of Customer table as the foreign key in Meter_Reading as there is a one to many relationship in the conceptual design.

Normal Form:
Highest normalization level: 4NF

Table: **Electricity_Billing_Details**
Columns:
electricity_bill_id
meter_reading_id [foreign key;references meter_reading_id of Meter_Reading table]
bill_generated_date
bill_due_date
billing_cycle_start_date
number_of_billing_days

payment_amount
amount_due
payment_id[foreign key;references payment_id of Payment table]

Primary Key Justification: electricity_bill_id will be unique, as this will be assign whenever new bill is being added into the system.

Foreign key mapping:

a. We have decided to map the primary key of Payment table as the foreign key in Electricity_Billing_Details as there is a one to many relationship in the conceptual design.
b. The other foreign key is meter_reading_id, which is the Primary Key of Meter_Reading table, this will uniquely determine the bill specific to each meter reading entered.

Normal Form:
Highest normalization level:4NF

Table: **Zipcode_Details**
Columns:
zip_code
city
state

Primary Key Justification: zip_code will be unique for every region and that uniquely determines the state and city and here as or system is for United States so we have not included country column.

Normal Form:
Highest normalization level:4NF

Table: **Payment**
Columns:
payment_id
payment_date
amount_due
card_number
cvv
customer_id[foreign key;references customer_id of Customer table]
card_expiry

Payment_amount

Primary Key Justification: payment_id will be the primary key as it will uniquely identify a row in the table.

Foreign key mapping:
We have decided to map the primary key of Customer table as the foreign key in Payment as there is a one to many relationship in the conceptual design.

Normal Form:
Highest normalization level:4NF

Part 4: Indexes

Indexes

Table :Admin

Indexes:

The following are the indexes of the movie table
1.PRIMARY Index on Column login_id which will be a clustered index.
2.employee_number on Column employee_number which will be a non-clustered index.
3.zip_code on Column zip_code which will be a non-clustered index.
4.admin_name Index on Columns first_name and last_name which is a non-clustered index.

Justification :
The default index has already been created as PRIMARY index made on login_id as it is the primary key in the table Admin.

The default index has already been created as employee_number index made on employee_number as it is the unique key in the table Admin.

The default foreign key index has already been created as zip_code index made on zip_code as it is the foreign key in the table Admin.

The admin_name index has been created on columns first_name and last_name as these columns has been used in many queries and this index will speed up the performance of the queries and increase the efficiency by providing the result quicker than it would have been without the admin_name index.

Table :Customer

Indexes:

The following are the indexes of the movie table.

1. PRIMARY Index on Column customer_id which will be a clustered index.
2. ssn_UNIQUE on Column ssn which will be a non-clustered index.
3. login_email_id_UNIQUE on Column login_email_id which will be a non-clustered index.
4. zip_code on Column zip_code which will be a non-clustered index.
5. service_provider_name_idx Index on Columns service_provider_name which is a non-clustered index.
6. customer_name Index on Columns first_name and last_name which is a nonclustered index.

Justification :
The default index has already been created as PRIMARY index made on customer_id as it is the primary key in the table Customer.

The default index has already been created as ssn_UNIQUE index made on ssn as it is the unique key in the table Customer.

The default index has already been created as login_email_id_UNIQUE index made on login_email_id as it is the unique key in the table Customer.

The default foreign key index has already been created as zip_code index made on zip_code as it is the foreign key in the table Customer.

The default foreign key index has already been created as service_provider_name_idx index made on service_provider_name as it is the foreign key in the table Customer.

The customer_name index has been created on columns first_name and last_name as these columns has been used in many queries and this index will speed up the performance of the queries and increase the efficiency by providing the result quicker than it would have been without the customer_name index.

Table :Electricity_Billing_Details

Indexes:

The following are the indexes of the movie table
1. PRIMARY Index on Column electricity_billing_id which will be a clustered index.
2. Meter_reading_id index on Column meter_reading_id which will be a non-clustered index.
3. payment_id index on Column payment_id which will be a non-clustered index.

Justification :

The default index has already been created as PRIMARY index made on electricity_billing_id as it is the primary key in the table Electricity_Billing_Details.

The default foreign key index has already been created as meter_reading_id index made on meter_reading_id as it is the foreign key in the table Electricity_Billing_Details.

The default foreign key index has already been created as payment_id index made on payment_id as it is the foreign key in the table Electricity_Billing_Details.

Table :Meter_Reading

Indexes:

The following are the indexes of the movie table
1. PRIMARY Index on Column meter_reading_id which will be a clustered index.
2. customer_id on Column customer_id which will be a non-clustered index.

Justification :

The default index has already been created as meter_reading_id index made on meter_reading_id as it is the primary key in the table Meter_Reading.

The default foreign key index has already been created as customer_id index made on customer_id as it is the foreign key in the table Meter_Reading.

Table :Service_Provider

Indexes:

The following are the indexes of the movie table
1.PRIMARY Index on Column provider_name which will be a clustered index.
2. license_number_UNIQUE Index on Column license_number which will be a non-clustered index.
3.zip_code_idx on Column zip_code which will be a non-clustered index.

Justification :

The default index has already been created as provider_name index made on provider_name as it is the primary key in the table Service_Provider.

The default index has already been created as license_number_UNIQUE index made on license_number as it is the unique key in the table Service_Provider.

The default foreign key index has already been created as zip_code_idx index made on zip_code as it is the foreign key in the table Service_Provider.

Table :Tariff_Details

Indexes:

The following are the indexes of the movie table
1.PRIMARY Index on Column zip_code and provider_name which will be a clustered index.
2. provider_name Index on Column provider_name which will be a clustered index.

Justification :

The default index has already been created as PRIMARY Index made on zip_code as it is the primary key in the table Tariff_Details.

The foreign key index has already been created as provider_name Index made on provider_name as it is the foreign key in the table Tariff_Details.

Table :Zipcode_Details

Indexes:

The following are the indexes of the movie table
1. PRIMARY Index on Column zip_code which will be a clustered index.

Justification :

The default index has already been created as PRIMARY Index made on zip_code as it is the primary key in the table Zipcode_Details.

Table :Payment
Indexes:

The following are the indexes of the movie table
1.PRIMARY Index on Column payment_id which will be a clustered index.
2. customer_id index on Column customer_id which will be a non-clustered index.

Justification :

The default index has already been created as PRIMARY Index made on payment_id as it is the primary key in the table Payment.
The default foreign key index has already been created as customer_id Index made on customer_id as it is the foreign key in the table Payment.

Part 5: Stored programs and views

STORED PROCEDURE:

a.

Procedure Name: generate_electricity_bill

Input Parameter:

1. meter_num int(10)- This is the meter_number of a particular customer.
2. peak_hr_reading double- This is the reading during peak hours.
3. off_peak_hr_reading double -This is the reading during off peak hours.
4. customer_id - This is the customer_id of the customer for which the admin will enter the details.

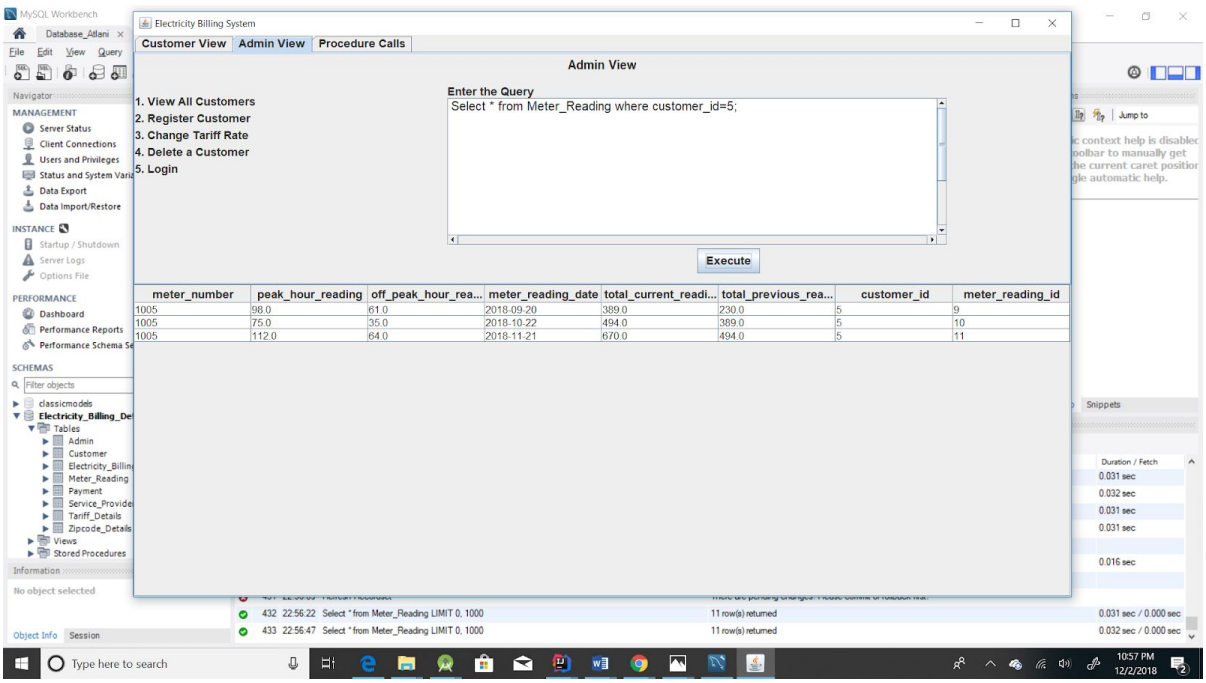
Description:

Whenever admin will enter the meter_reading of current month specific to any customer, electricity bill should get generated and payment amount should also get calculated using tariff rates of the service provider which we will get from Tariff _Details table.

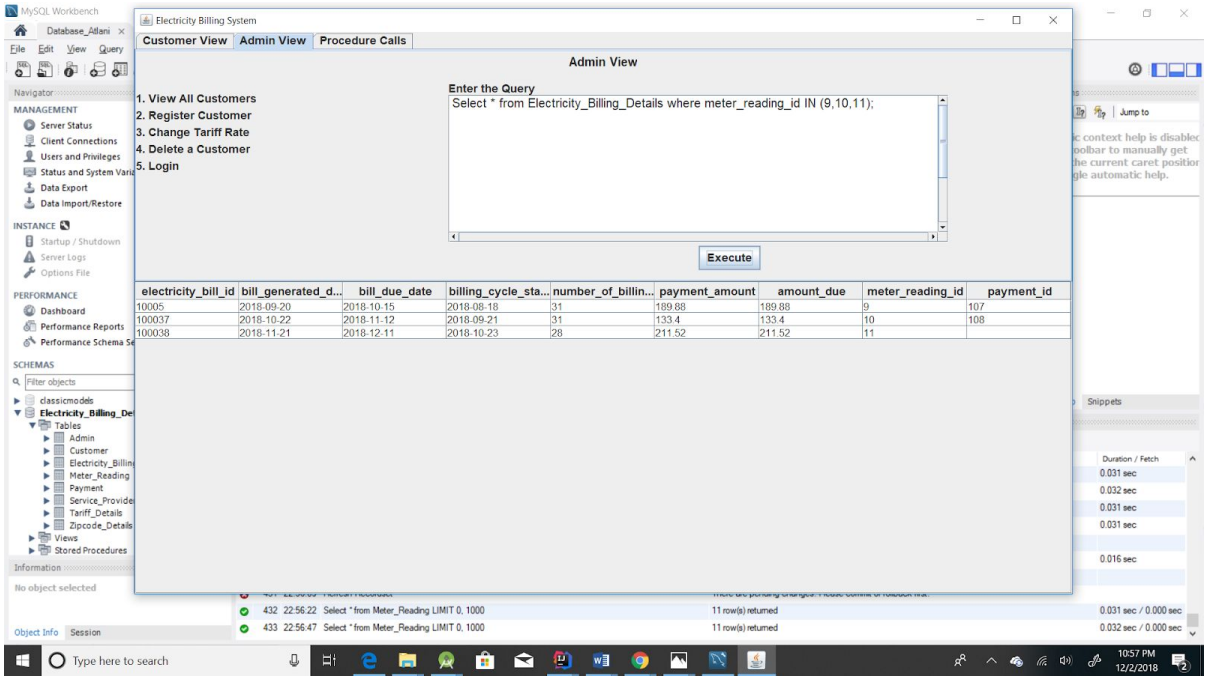
Assumption: Here whenever we are adding entry in Meter Reading we are assuming that meter reading of the current month i.e month in current date is being added.

Before Procedure Run:

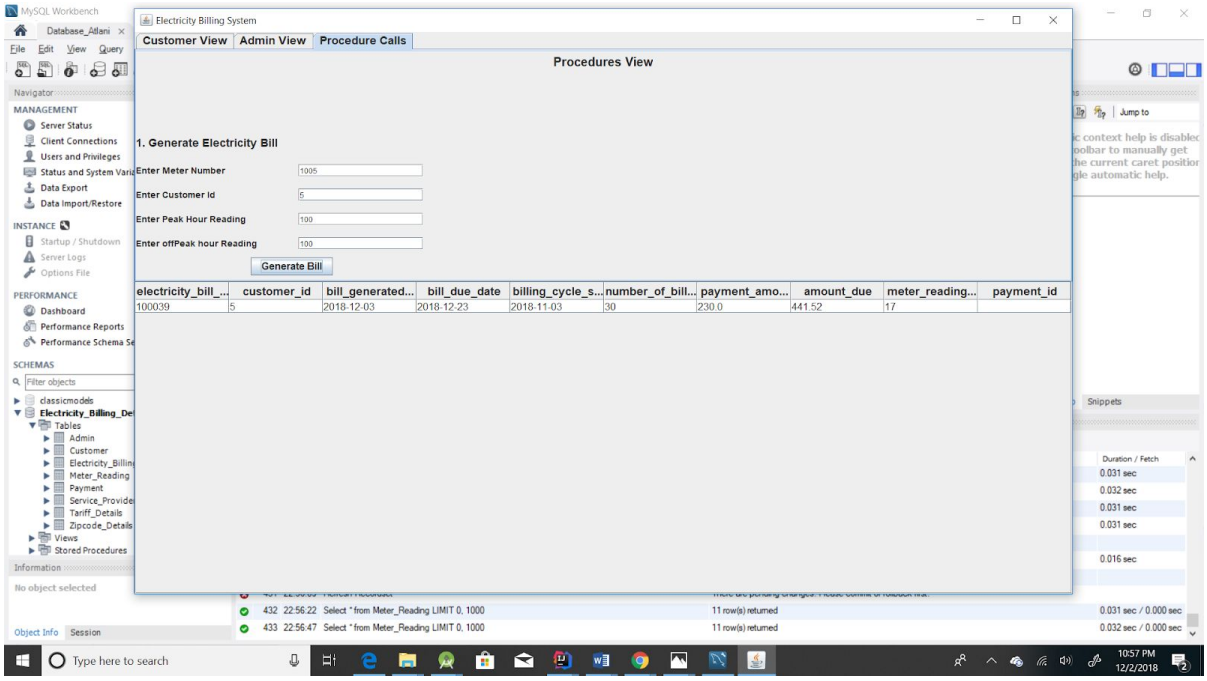
Meter_Reading table view for customer_id=5:



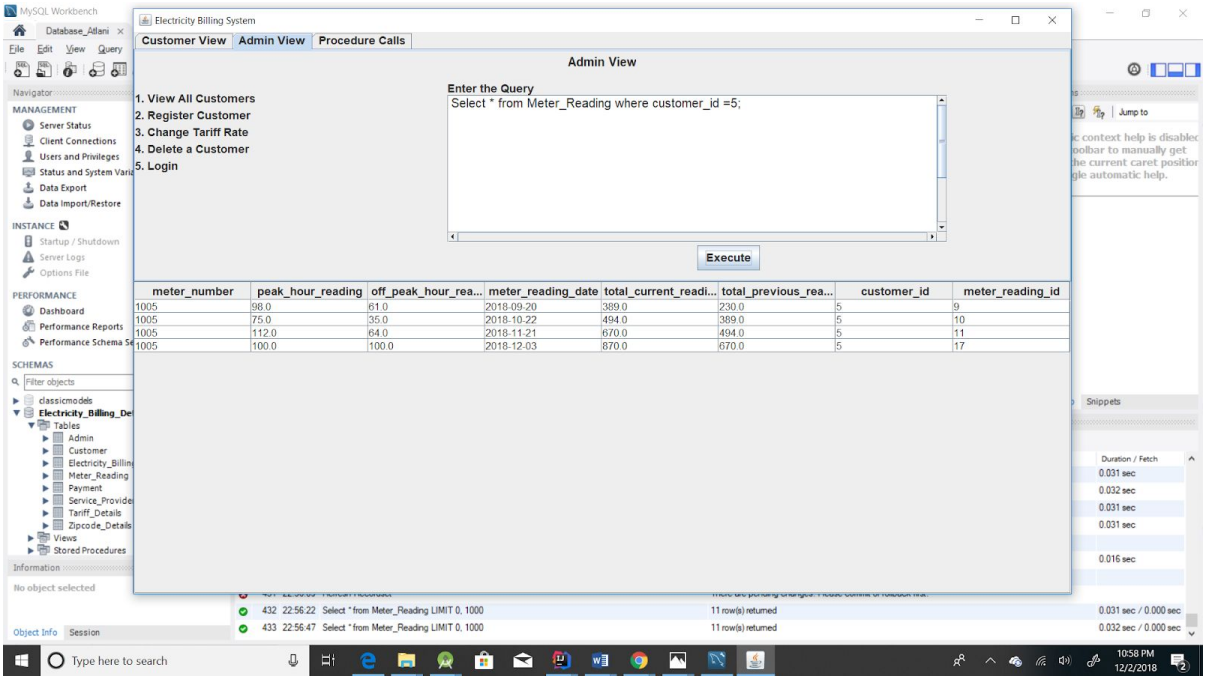
View of Electricity_Billing_Details:



After Procedure run, new bill generated when we click generate bill



View of Meter_Reading table for customer id 5 after procedure run:



b.

Procedure Name: get_amount_due

Input Parameter:

cust_id bigint(20)-- This is the customer id of customer for which we need to find the amount due.

Output Parameter:

amountDue double - This is return parameter as this will be used in above procedure for calculation of cumulative amount due.

Description:

Procedure for getting amount not paid in previous bill. This procedure is used to get the amount due of any customer id who has not paid the bill of the last month. This is being used inside the above procedure.

Above procedure called inside the above procedure.

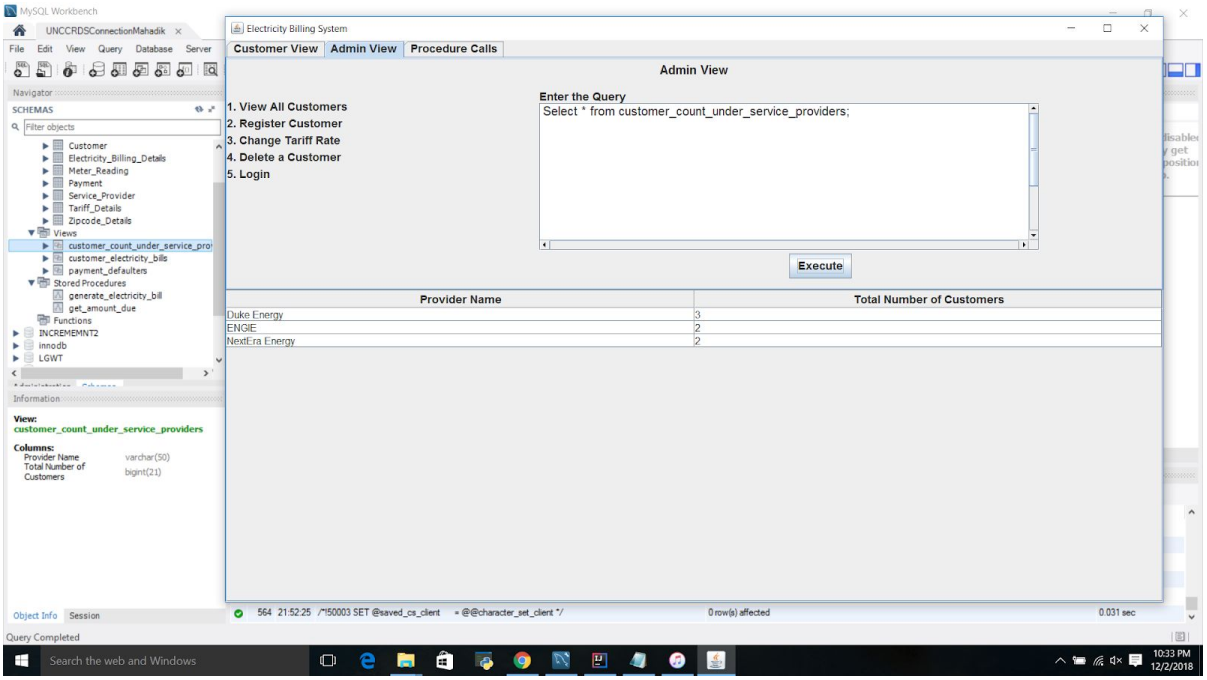
TRIGGER:

a.
Trigger Name:after_payment_insert
Description: This trigger is there because as soon as Customer Pays the Bill the payment id should get updated in Electricity_Billing_Details table.

VIEWS:

a.
View Name: customer_electricity_bills
Description: This view is used by admin whenever he wants to view all bills of any particular customer.

Create View customer_electricity_bills As
Select e.electricity_bill_id,m.customer_id, e.bill_generated_date,
e.bill_due_date,e.billing_cycle_start_date,
e.number_of_billing_days,e.payment_amount,e.amount_due,e.meter_reading_id,e.payment_id
from Electricity_Billing_Details e INNER JOIN Meter_Reading m using(meter_reading_id)
INNER JOIN
Customer c using(customer_id);

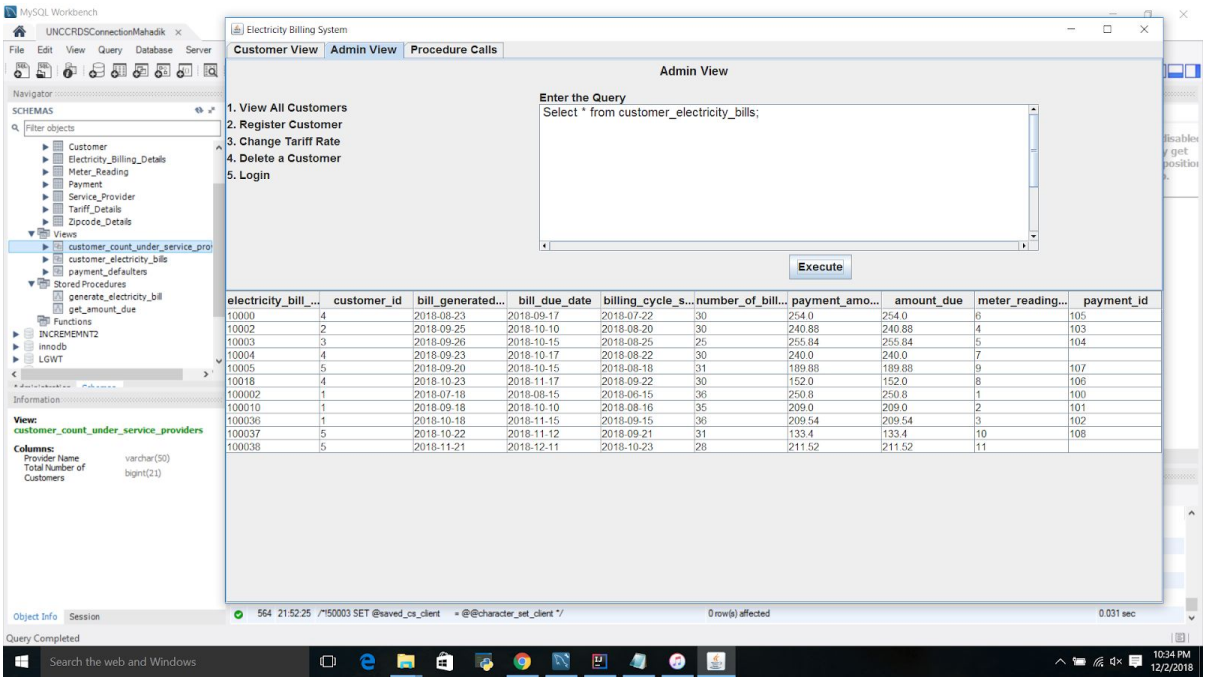


b.

View Name: customer_count_under_service_providers

Description: Admin View where the admin can view the number of customer under each service provider.

```
create view customer_count_under_service_providers as
select c.service_provider_name as 'Provider Name',count(customer_id) 'Total Number of Customers' from Customer c
inner join Service_Provider sp on
sp.provider_name=c.service_provider_name
group by sp.provider_name;
```



c.

View Name: payment_defaulters

Description: To find all customers who are defaulters i.e they did not paid the bill and due date is also crossed.

```
create view payment_defaulters as
select c.first_name,c.last_name, e.amount_due,e.bill_due_date
from Customer c
inner join Meter_Reading mr on
mr.customer_id=c.customer_id
inner join Electricity_Billing_Details e
on e.meter_reading_id=mr.meter_reading_id
where e.bill_due_date<curdate() and e.payment_id is null;
```