ADAMSON UNIVERSITY

COLLEGE OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

DATABASE DESIGN AND DEVELOPMENT LAB

58018 / SAT 14:00-17:00 / OZ212

ELECTRICITY BILLING MANAGEMENT SYSTEM

Project Documentation

Group A:

De Guzman, Jefferson Allen B.
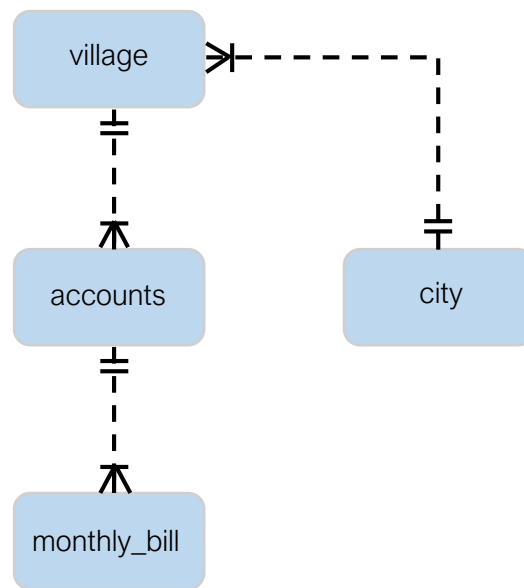
Mbonyechris, Christian

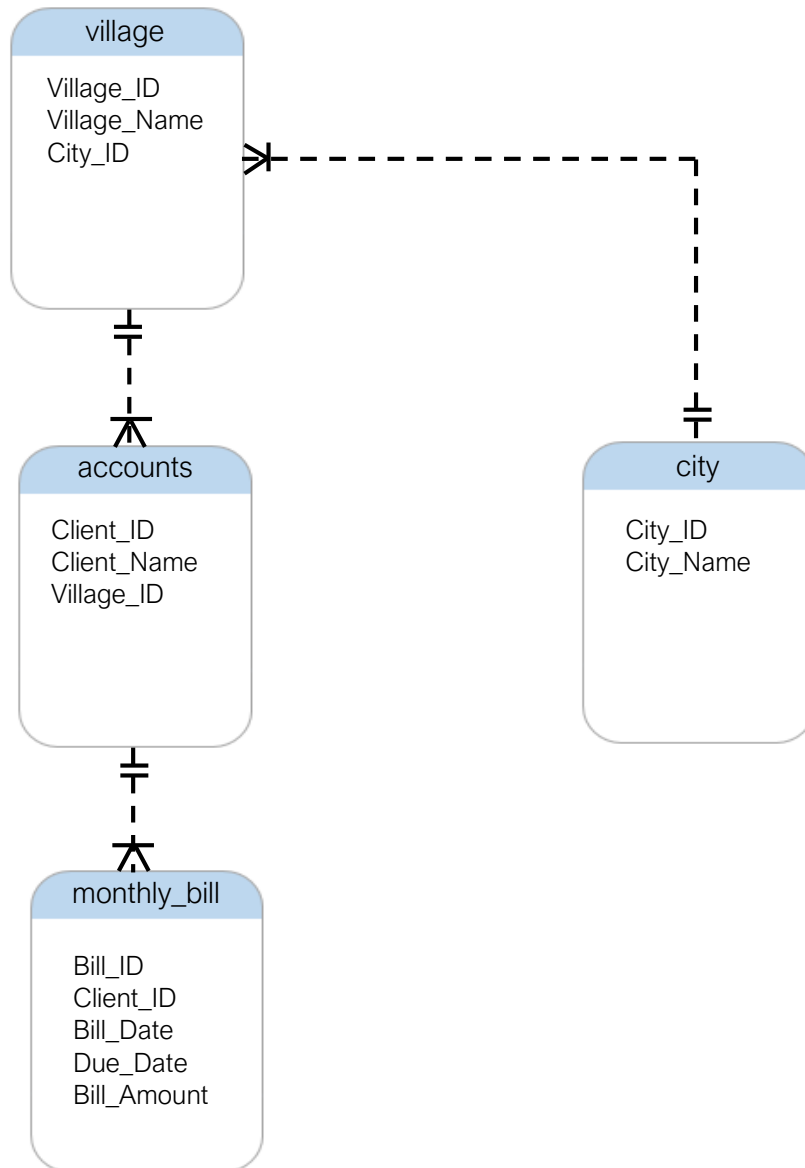Reomales, Cedric Joel

Villegas, Nicole Anne R.

Submitted to:
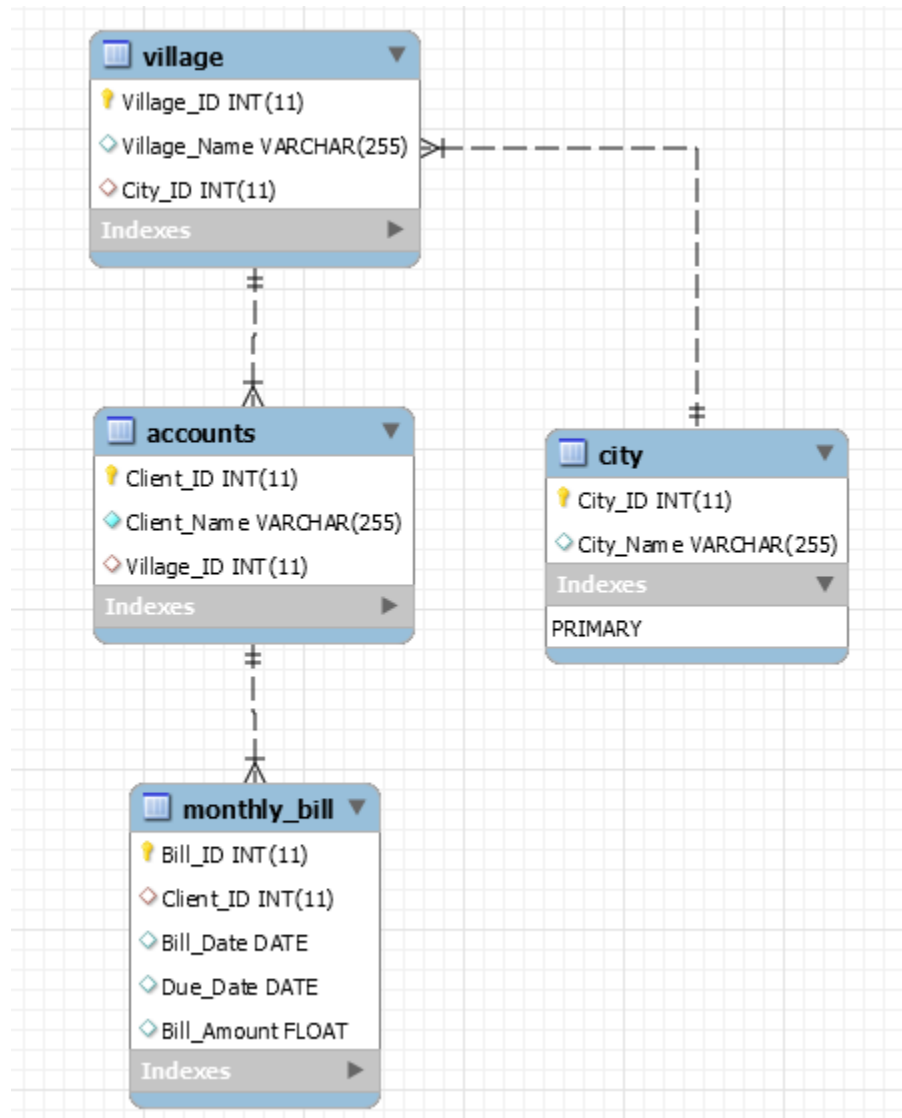
Engr. Rafaello M. Baluyot

December 10, 2018

I.   CONCEPTUAL ENTITY RELATIONSHIP DIAGRAM

II.   LOGICAL ENTITY RELATIONSHIP DIAGRAM

village

Village_ID
Village_Name
City_ID

accounts

Client_ID
Client_Name
Village_ID

city

City_ID
City_Name

monthly_bill

Bill_ID
Client_ID
Bill_Date
Due_Date
Bill_Amount

III.   PHYSICAL ENTITY RELATIONSHIP DIAGRAM

village
- 🔑 Village_ID INT(11)
- ◇ Village_Name VARCHAR(255)
- ◇ City_ID INT(11)
- Indexes ▶

accounts
- 🔑 Client_ID INT(11)
- ◆ Client_Name VARCHAR(255)
- ◇ Village_ID INT(11)
- Indexes ▶

city
- 🔑 City_ID INT(11)
- ◇ City_Name VARCHAR(255)
- Indexes ▼
- PRIMARY

monthly_bill
- 🔑 Bill_ID INT(11)
- ◇ Client_ID INT(11)
- ◇ Bill_Date DATE
- ◆ Due_Date DATE
- ◇ Bill_Amount FLOAT
- Indexes ▶

IV.   STORED PROCEDURES

- Create, Update, Delete Queries

  1. AddCity(IN add_city varchar(255))
     - stored procedure for adding new city to city table

```sql
-- Add New City --
DELIMITER $$
CREATE PROCEDURE AddCity(IN add_city varchar(255))
BEGIN
    INSERT INTO City(City_Name)
    VALUES(add_city);
END$$
DELIMITER ;
CALL AddCity('Manila');
```

	#	Time	Action	Message
✓	239	18:25:37	CALL AddCity('Manila')	1 row(s) affected
✓	240	18:25:41	SELECT * FROM City LIMIT 0, 1000	4 row(s) returned

	City_ID	City_Name
▶	1	Muntinlupa
	2	Las Piñas
	3	Parañaque
	4	Manila

  2. AddVillage(IN add_village varchar(255), IN id int)
     - stored procedure for adding new village to village table

```sql
-- Add New Village --
DELIMITER $$
CREATE PROCEDURE AddVillage(IN add_village varchar(255), IN id int)
BEGIN
    INSERT INTO Village(Village_Name, City_ID)
    VALUES(add_village, id);
END$$
DELIMITER ;
CALL AddVillage('Sta. Mesa', 4);
```

	#	Time	Action	Message
✓	242	18:35:51	CALL AddVillage('Sta. Mesa', 4)	1 row(s) affected
✓	243	18:35:56	SELECT * FROM Village LIMIT 0, 1000	46 row(s) returned

Village_ID	Village_Name	City_ID
40	San Isidro	3
41	San Martin de...	3
42	Sto. Niño	3
43	Sun Valley	3
44	Tambo	3
45	Vitalez	3
46	Sta. Mesa	4

3. AddAccount(IN add_account varchar(255), IN id int)
   - stored procedure for adding new account to accounts table

```
-- Add New Account --
DELIMITER $$
CREATE PROCEDURE AddAccount(IN add_account varchar(255), IN id int)
BEGIN
    INSERT INTO Accounts(Client_Name, Village_ID)
    VALUES(add_account, id);
END$$
DELIMITER ;
CALL AddAccount('Villegas', 8);
```

#	Time	Action	Message
✓	245 18:43:40	CALL AddAccount('Villegas', 8)	1 row(s) affected
✓	246 18:43:40	select * from accounts LIMIT 0, 1000	201 row(s) returned

Client_ID	Client_Name	Village_ID
196	Javier	39
197	Miranda	12
198	Yap	1
199	Sarmiento	3
200	Pineda	6
201	Villegas	8

4. UpdateCity(IN id int, IN new_city varchar(255))
   - stored procedure for updating city table

```
-- Update City Table --
DELIMITER $$
CREATE PROCEDURE UpdateCity(IN id int, IN new_city varchar(255))
BEGIN
    UPDATE City
    SET City_Name = new_city
    WHERE City_ID = id;
END$$
DELIMITER ;
CALL UpdateCity(4, 'Canlaon');
```

#	Time	Action	Message
✓	248 19:00:24	CALL UpdateCity(4, 'Canlaon')	1 row(s) affected
✓	249 19:00:31	SELECT * FROM City LIMIT 0, 1000	4 row(s) returned

	City_ID	City_Name
▶	1	Muntinlupa
	2	Las Piñas
	3	Parañaque
	4	Canlaon

5. **UpdateVillage(IN v_id int, IN new_village varchar(255))**
   - stored procedure for updating village name in village table

```sql
-- Update Village Table --
DELIMITER $$
CREATE PROCEDURE UpdateVillage(IN v_id int, IN new_village varchar(255))
BEGIN
    UPDATE Village
    SET Village_Name = new_village
    WHERE Village_ID = v_id;
END$$
DELIMITER ;
CALL UpdateVillage(4, 'Masulog');
```

#	Time	Action	Message
✓	251 19:03:42	CALL UpdateVillage(4, 'Masulog')	1 row(s) affected
✓	252 19:04:45	SELECT * FROM Village LIMIT 0, 1000	46 row(s) returned

	Village_ID	Village_Name	City_ID
▶	1	Alabang	1
	2	Ayala Alabang	1
	3	Bayanan	1
	4	Masulog	1
	5	Cupang	1

6. **UpdateAccount(IN id int, IN new_name varchar(255), new_village int)**
   - stored procedure for updating client name and/or village ID

```sql
-- Update Accounts Table--
DELIMITER $$
CREATE PROCEDURE UpdateAccount(IN id int, IN new_name varchar(255), IN new_village int)
BEGIN
    UPDATE Accounts
    SET Client_Name = new_name, Village_ID = new_village
    WHERE Client_ID = id;
END$$
DELIMITER ;
CALL UpdateAccount(8, 'Raymundo', 18);
```

#	Time	Action	Message
✅ 254	19:08:41	CALL UpdateAccount(8, 'Raymundo', 18)	1 row(s) affected
✅ 255	19:08:46	SELECT * FROM Accounts LIMIT 0, 1000	201 row(s) returned

Client_ID	Client_Name	Village_ID
3	Cruz	6
4	Bautista	21
5	Ocampo	36
6	Garcia	17
7	Mendoza	44
8	Raymundo	18
9	Andrada	41
10	Castillo	12

7. UpdateMonthlyBill(IN id int, IN b_date date, IN d_date date, IN kwh int)
   - stored procedure for updating monthly bill in monthly bill table

```
-- Updates Monthly Bill Table --
DELIMITER $$
CREATE PROCEDURE UpdateMonthlyBill(IN id int, IN b_date date, IN d_date date, IN kwh int)
BEGIN
    DECLARE b_amount float;
    SET b_amount = (820 + 80 + (kwh * 8.17));
    INSERT INTO Monthly_Bill(Client_ID, Bill_Date, Due_date, Bill_amount)
    SELECT id, b_date , d_date, b_amount;
END$$
DELIMITER ;
CALL UpdateMonthlyBill(4, '2019-6-13', '2019-6-22', 78);
```

#	Time	Action	Message
✅ 258	19:11:59	CALL UpdateMonthlyBill(4, '2019-6-13', '2...	1 row(s) affected
✅ 259	19:12:08	SELECT * FROM Monthly_Bill LIMIT 0, 10...	1 row(s) returned

Bill_ID	Client_ID	Bill_Date	Due_Date	Bill_Amount
1	4	2019-06-13	2019-06-22	1537.26

8. DeleteFromCity(IN c_id int)
   - stored procedure for deleting city name from city table

```
-- Delete from City table --
DELIMITER $$
CREATE PROCEDURE DeleteFromCity(IN c_id int)
BEGIN
    DELETE FROM City
    WHERE City_ID = c_id;
END$$
DELIMITER ;
CALL DeleteFromCity(1);
```

✓	261 19:24:36 CALL DeleteFromCity(1)	1 row(s) affected
✓	262 19:24:36 SELECT * FROM City LIMIT 0, 1000	3 row(s) returned

	City_ID	City_Name
▶	2	Las Piñas
	3	Parañaque
	4	Canlaon

9. DeleteFromVillage(IN v_id int)

- stored procedure for deleting village from village table

```
-- Delete from Village table --
DELIMITER $$
CREATE PROCEDURE DeleteFromVillage(IN v_id int)
BEGIN
    DELETE FROM Village
    WHERE Village_ID = v_id
    LIMIT 1;
END$$
DELIMITER ;
CALL DeleteFromVillage(2);
```

✓	279 19:45:17 CALL DeleteFromVillage(2)	1 row(s) affected
✓	280 19:45:17 SELECT * FROM Village LIMIT 0, 1000	44 row(s) returned

	Village_ID	Village_Name	City_ID
▶	1	Alabang	1
	3	Bayanan	1
	4	Buli	1

10. DeleteFromAccounts(IN a_id int)

- stored procedure for deleting account from accounts table

```
-- Delete from Accounts table --
DELIMITER $$
CREATE PROCEDURE DeleteFromAccounts(IN a_id int)
BEGIN
    DELETE FROM Accounts
    WHERE Client_ID = a_id;
END$$
DELIMITER ;
CALL DeleteFromAccounts(1);
```

✓	282 19:46:51 CALL DeleteFromAccounts(1)	1 row(s) affected
✓	283 19:46:51 SELECT * FROM Accounts LIMIT 0, 1000	194 row(s) returned

	Client_ID	Client_Name	Village_ID
▶	2	Reyes	30
	3	Cruz	6
	4	Bautista	21

11. DeleteFromMonthlyBill(IN m_id int)

- Stored procedure for deleting monthly bill from monthly bill table

```
-- Delete from Monthly Bill table
DELIMITER $$
CREATE PROCEDURE DeleteFromMonthlyBill(IN m_id int)
BEGIN
    DELETE FROM Monthly_Bill
    WHERE Bill_ID = m_id;
END$$
DELIMITER ;
CALL DeleteFromMonthlyBill(1);
```

| ✓ | 295 | 19:50:00 | CALL DeleteFromMonthlyBill(6) | 1 row(s) affected |
| ✓ | 296 | 19:50:05 | SELECT * FROM Monthly_Bill LIMIT 0, 10... | 6 row(s) returned |

Bill_ID	Client_ID	Bill_Date	Due_Date	Bill_Amount
4	4	2018-06-13	2018-06-22	1872.23
5	5	2018-06-13	2018-06-22	1618.96
7	7	2018-06-13	2018-06-22	1333.01
8	8	2018-06-13	2018-06-22	1757.85

- **Complex Queries**

12. GetCityVillageAndAverage()

- stored procedure for getting the city name, village name, and average of the average monthly bill per acocunt

```
-- City Name, Village Name, and average of the average monthly bill per account --
DELIMITER $$
CREATE PROCEDURE GetCityVillageAndAverage()
BEGIN
    SELECT City_Name, Village_Name, AVG(avgamt) as 'Average Monthly Bill'
    FROM City as c
    INNER JOIN Village as v on v.City_ID = c.City_ID
    INNER JOIN (
    SELECT Village_ID, (SELECT avg(Bill_Amount)
    FROM GroupA.Monthly_Bill b WHERE b.Client_ID = a.Client_ID) as avgamt FROM Accounts as a
    ) as x on x.Village_ID = v.Village_ID
    GROUP BY City_Name, Village_Name;
END$$
DELIMITER ;
CALL GetCityVillageAndAverage();
```

	City_Name	Village_Name	Average Monthly Bill
▶	Muntinlupa	Alabang	2607.530029296875
	Muntinlupa	Ayala Alabang	1222.7150268554688
	Muntinlupa	Bayanan	1271.7349853515625
	Muntinlupa	Buli	1475.9849853515625
	Muntinlupa	Cupang	1553.5999755859375
	Muntinlupa	Poblacion	1440.8540283203124
	Muntinlupa	Putatan	1537.260009765625
	Muntinlupa	Sucat	1463.72998046875
	Muntinlupa	Tunasan	1696.5750122070312
	Las Piñas	Almanza Uno	2615.699951171875
	Las Piñas	Almanza Dos	2288.89990234375
	Las Piñas	BF International	1724.0806884765625
	Las Piñas	Daniel Fajardo	1667.97998046875
	Las Piñas	Elias Aldana	2025.4174194335938

13. GetCityAndVillageGreaterThanAverage()

- Stored procedure for getting cities and villages with number of accounts greater than the average accounts per village

```
-- Cities and Villages with number of accounts greater than the average accounts per village --
DELIMITER $$
CREATE PROCEDURE GetCityAndVillageGreaterThanAverage()
BEGIN
    SELECT c.City_Name, v.village_name, count(a.Client_ID) as 'Number of Accounts'
    FROM GroupA.Accounts as a
    INNER JOIN GroupA.Village as v on v.Village_ID = a.Village_ID
    INNER JOIN GroupA.City as c on c.City_ID = v.City_ID
    GROUP BY a.Village_ID
    HAVING COUNT(a.Client_ID) > (
        SELECT AVG(ave) FROM (
            SELECT a.Village_ID, COUNT(a.Client_ID) ave
            FROM GroupA.Accounts as a
            GROUP BY a.Village_ID
        ) as x
);
END$$
DELIMITER ;
CALL GetCityAndVillageGreaterThanAverage();
```

City_Name	village_name	Number of Accounts
Muntinlupa	Alabang	5
Muntinlupa	Ayala Alabang	5
Muntinlupa	Bayanan	11
Muntinlupa	Buli	6
Muntinlupa	Cupang	8
Muntinlupa	Sucat	8
Las Piñas	Almanza Uno	5
Las Piñas	Almanza Dos	5
Las Piñas	BF International	6
Las Piñas	Daniel Fajardo	6
Las Piñas	Elias Aldana	6
Las Piñas	Manuyo Dos	5
Las Piñas	Zapote	6
Parañaque	Baclaran	5
Parañaque	Marcelo Green	5
Parañaque	Moonwalk	5
Parañaque	San Antonio	5
Parañaque	San Dionisio	5
Parañaque	San Martin de ...	9
Parañaque	Sto. Niño	8

14. GetAccountsGreaterThanParameters(IN amount int)
- stored procedure for getting the accounts with average monthly bill greater than the parameters inputed

```
-- Accounts with average monthly bill greater than parameters --
DELIMITER $$
CREATE PROCEDURE GetAccountsGreaterThanParameters(IN amount int)
BEGIN
    SELECT Client_Name
    FROM Accounts as a
    WHERE (SELECT AVG(Bill_Amount) FROM GroupA.Monthly_Bill as b WHERE b.Client_ID = a.Client_ID) > amount;
END$$
DELIMITER ;
CALL GetAccountsGreaterThanParameters(1500);
```

Client_Name
Reyes
Ocampo
Garcia
Torres
Andrada
Castillo
Flores
Villanueva
Aquino
Castro
Mercado
de la Reyes
del Rosario
de los Santos
de Guzman
de Castro
de la Rosa
de Asis
de Rosales
Divinagracia
Ortiz
Marcos

15. GetMinAndMaxBill(IN start_date date, IN end_date date)
- stored procedure for getting the minimum and maximum bill for each account between given parameters *start_date* and *end_date*

```sql
-- Minimum and maximum bill for each account between param_start and param_end --
DELIMITER $$
CREATE PROCEDURE GetMinAndMaxBill(IN start_date date, IN end_date date)
BEGIN
    SELECT a.Client_ID, a.Client_Name, MIN(Bill_Amount) as 'Minimum', MAX(Bill_Amount) as 'Maximum'
    FROM monthly_bill as mb
    INNER JOIN groupa.accounts as a on a.Client_ID=mb.Client_ID
    WHERE Bill_Date BETWEEN start_date AND end_date
    GROUP BY a.Client_ID
    ORDER BY a.Client_ID;
END$$
DELIMITER ;
CALL GetMinAndMaxBill('2018-01-13', '2019-05-22');
```

Client_ID	Client_Name	Minimum	Maximum
1	Santos	1177.78	1757.85
2	Reyes	1471.9	1790.53
3	Cruz	1087.91	1684.32
4	Bautista	1341.18	1635.3
5	Ocampo	1447.39	2256.22
6	Garcia	1161.44	1798.7
7	Mendoza	1194.12	1667.98
8	Torres	1406.54	2313.41
9	Andrada	1316.67	2354.26
10	Castillo	1177.78	2476.81
11	Flores	1431.05	2476.81
12	Villanueva	1651.64	1733.34
13	Rivera	1365.69	1390.2
14	Aquino	1537.26	1561.77
15	Navarro	1079.74	1463.73
16	Ramos	1324.84	1627.13
17	Salazar	1177.78	1267.65
18	Ramos	1357.52	1512.75

16. GetNumberOfAccounts()
- stored procedure for getting the number of accounts per city and village

```sql
-- Number of Accounts per City and Village --
DELIMITER $$
CREATE PROCEDURE GetNumberOfAccounts()
BEGIN
    SELECT c.City_Name, (SELECT COUNT(*) FROM GroupA.Accounts as a2
    INNER JOIN GroupA.Village as v2 on v2.Village_ID = a2.Village_ID
    WHERE v2.City_ID = c.City_ID) as City_Account, v.village_name,
    COUNT(a.Client_ID) Village_Account
    FROM GroupA.Accounts as a
    INNER JOIN Village as v on v.Village_ID = a.Village_ID
    INNER JOIN City as c on c.City_ID = v.City_ID
    GROUP BY c.City_Name, v.village_name
    ORDER BY c.City_Name, v.village_name;
END$$
DELIMITER ;
CALL GetNumberOfAccounts();
```

City_Name	City_Account	village_name	Village_Account
Las Piñas	78	Talon Kuatro	2
Las Piñas	78	Talon Singko	2
Las Piñas	78	Talon Tres	4
Las Piñas	78	Talon Uno	4
Las Piñas	78	Zapote	6
Muntinlupa	50	Alabang	5
Muntinlupa	50	Ayala Alabang	5
Muntinlupa	50	Bayanan	11
Muntinlupa	50	Buli	6
Muntinlupa	50	Cupang	8
Muntinlupa	50	Poblacion	2
Muntinlupa	50	Putatan	3
Muntinlupa	50	Sucat	8
Muntinlupa	50	Tunasan	2
Parañaque	72	Baclaran	5
Parañaque	72	Don Bosco	3
Parañaque	72	Don Galo	4
Parañaque	72	La Huerta	2
Parañaque	72	Marcelo Green	5
Parañaque	72	Merville	2

V.   Other Queries

17. SELECT * FROM City;
- views the city table

```
--  View City Table --
SELECT * FROM City;
```

	City_ID	City_Name
▶	1	Muntinlupa
	2	Las Piñas
	3	Parañaque

18. SELECT * FROM Village;
- views the village table

```
-- View Village Table --
SELECT * FROM Village;
```

Village_ID	Village_Name	City_ID
1	Alabang	1
2	Ayala Alabang	1
3	Bayanan	1
4	Buli	1
5	Cupang	1
6	Poblacion	1
7	Putatan	1
8	Sucat	1
9	Tunasan	1
10	Almanza Uno	2
11	Almanza Dos	2
12	BF International	2
13	Daniel Fajardo	2
14	Elias Aldana	2
15	Ilaya	2
16	Manuyo Uno	2
17	Manuyo Dos	2
18	Pamplona Uno	2

19. SELECT * FROM Accounts;

- views the accounts table

```
-- View Accounts Table --
SELECT * FROM Accounts;
```

Client_ID	Client_Name	Village_ID
1	Santos	41
2	Reyes	30
3	Cruz	6
4	Bautista	21
5	Ocampo	36
6	Garcia	17
7	Mendoza	44
8	Torres	38
9	Andrada	41
10	Castillo	12
11	Flores	12
12	Villanueva	32
13	Rivera	14
14	Aquino	22
15	Navarro	3
16	Ramos	4
17	Salazar	2
18	Ramos	25
19	Castro	35
20	Mercado	42

20. SELECT * FROM Monthly_Bill;

- views the monthly bill table

```
-- View Monthly Bill Table --
SELECT * FROM Monthly_Bill;
```

Bill_ID	Client_ID	Bill_Date	Due_Date	Bill_Amount
1	1	2018-01-13	2018-01-22	1757.85
2	2	2018-01-13	2018-01-22	1610.79
3	3	2018-01-13	2018-01-22	1341.18
4	4	2018-01-13	2018-01-22	1471.9
5	5	2018-01-13	2018-01-22	1651.64
6	6	2018-01-13	2018-01-22	1798.7
7	7	2018-01-13	2018-01-22	1194.12
8	8	2018-01-13	2018-01-22	1455.56
9	9	2018-01-13	2018-01-22	1316.67
10	10	2018-01-13	2018-01-22	2411.45