

Linnaeus University
2DV513 – Database Theory
Assignment 3 Final project

Name: Rashed Qazizada

Student id: rq222ah



Contents

Idea.....	3
Logical Model.....	3
Design in SQL	4
Queries	5
Views.....	6

Tasks

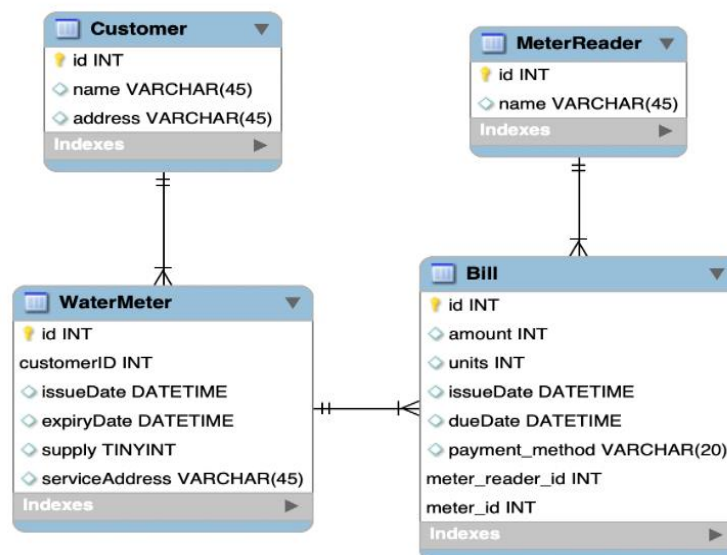
Idea

Idea for this project is to manage the system of WASA (A water supplying company) which helps to provide water at their customer doorstep and monitors its customer water usage with the Water Meter attached to their doors. For the company to monitor these meters, they must hire Meter Readers which will note the readings from the meters every month and inform the company and generate a bill.

They require to have a database which will maintain the information of their customers. The water meters for which the customer has requested and are up and running. Alongside this, they want to manage the information of all meter readers to maintain their link with every bill whose reading has been taken by respective meter reader. For this scenario, I created a logical model, constraints and a high-level view is discussed below.

Customer can have multiple meters. Customers address and meter address is not required to be same. If It is not same, then I can change the service address in the water meter table. Then I have bills which have many to one relation with the meters. Meters will have multiple bills based on monthly basis and that bill is connected to Meter Reader.

Logical Model



As my main objective was to manage customers, their meters, their bills, and meter readers information. Therefore, I created four tables: customer, bill, water meter and meter reader. I have as less as needed information on the client as I do not require much to store in a WASA database.

In the water meter table, I needed expiry and issuance date, supply flag as well, which will define the connectivity and if the supply is working for this service address or not. As I have a service address here as well and an address in the customer table. This is for if a customer has multiple connections on their name.

In the bill table, I have everything usual and I have a payment method column as well. As the World is getting digital the payment of the bill can be made through multiple channels.

In the meter reader table, I only have unique IDs and names, which I will be using to crossmatch with the meter reader id in the bill table to create a link between bills and every meter reader.

Design in SQL

As one-meter reader or multiple meter readers could be attached to the same bill which is why I added a meter_reader_id column in the bill table and create a many to one relation with the meter reader. After that, every meter will have multiple bills, which is why I again create a meter_id column in the bill table. Which will point to the unique id in the meters table. In the meters table, as a single customer could have multiple meters connections, so again I have many to one relation. For which I created a customer_id in the meter table to backtrack the customer details for any meter.

```
CREATE SCHEMA IF NOT EXISTS `assignment3`;
USE `assignment3`;
```

```
CREATE TABLE IF NOT EXISTS `Customer` (
  `id` INT NOT NULL,
  `name` VARCHAR (45) NULL,
  `address` VARCHAR (45) NULL,
  PRIMARY KEY (`id`));
```

```
CREATE TABLE IF NOT EXISTS `WaterMeter` (
  `id` INT NOT NULL,
  `customerID` INT NOT NULL,
  `issueDate` DATETIME NULL,
  `expiryDate` DATETIME NULL,
  `supply` TINYINT NULL,
  `serviceAddress` VARCHAR (45) NULL,
  PRIMARY KEY (`id`, `customerID`),
  INDEX `cus_fk_id_idx` (`customerID` ASC) VISIBLE,
  CONSTRAINT `cus_fk_id`
  FOREIGN KEY (`customerID`)
  REFERENCES `Customer` (`id`));
```

```
CREATE TABLE IF NOT EXISTS `MeterReader` (
  `id` INT NOT NULL,
  `name` VARCHAR (45) NULL,
  PRIMARY KEY (`id`));
```

```
CREATE TABLE IF NOT EXISTS `Bill` (
  `id` INT NOT NULL,
  `amount` INT NULL,
  `units` INT NULL,
  `issueDate` DATETIME NULL,
  `dueDate` DATETIME NULL,
  `payment_method` VARCHAR (20) NULL,
  `meter_reader_id` INT NOT NULL,
  `meter_id` INT NOT NULL,
  PRIMARY KEY (`id`, `meter_reader_id`, `meter_id`),
  INDEX `meter_id_idx` (`meter_id` ASC) VISIBLE,
  INDEX `meter_reader_idx` (`meter_reader_id` ASC) VISIBLE,
  CONSTRAINT `meter_id`
    FOREIGN KEY (`meter_id`)
      REFERENCES `WaterMeter` (`id`),
  CONSTRAINT `meter_reader`
    FOREIGN KEY (`meter_reader_id`)
      REFERENCES `MeterReader` (`id`));
```

Queries

1. **This query is to fetch the bill amount and customer name to determine the bills sent to a specific person.**

```
Select mr.name, b.id, b.amount from bill b join meter_reader mr on mr.id=b.meter_reader_id;
```

2. **This query is to check for the history of water meter to check the issuance of water meter and get information details of all the bills and their amounts.**

```
Select wm.id,wm.issueDate, b.id, b.amount from bill b join water_meter wm on wm.id=b.meter_id;
```

3. **This query is to fetch the information on how many readings a meter reader has done.**

```
Select mr.id,mr.name, count(*) from meter_reader mr join bill b on mr.id=b.meter_reader_id group by mr.id;
```

4. **This is to fetch the complete information of the meter, it is issuance date and the total bills created on any meter.**

```
Select wm.id, wm.issueDate, count(*) from bill b join water_meter wm on wm.id=b.meter_id group by wm.id,wm.issueDate;
```

5. This is the list of customers who has a meter connection with us.

Select * from customer where id in (Select distinct(customerID) from water_meter);

6. This is the list of all the meters whose bill has been generated in the system.

Select * from customer water_meter where id in (Select distinct(meter_id) from bill);

Views

1. This is to get the count of total meters issued on monthly basis.

Create view meterCount_view as select left(issuedate,7), count (*) from bill group by 1;

2. This is to get the information of units consumed monthly after the month of Jan 2020.

Create view readingHistory_view as select left(issuedate,7), units from bill where
left(issuedate,7)>='2020-01';

Individual work

Name: Rashed Qazizada

Student id: rq222ah

Bibliography

[1] Abhraham S, Peter B. G, Greg G, Operating System Concepts, 10th ed, Wiley, 2018, pp. 105-156.