**REPORT**

**(Mobile Mart System)**

**1.INTRODUCTION**

**1.1.Background**

In the fast-changing world, information technology and information management are going to play an important role. We are living in the computer age during past some year. The computer has gaining popularity. Computer revolution found its way into almost every aspect of human life and living. A computer is admirably suited to handle any information and hence is an information processor that is, it can receive data, perform some basic operations on that data and produce results according to a predetermined program.

This software is used mainly for Mobile stores to maintain the details of mobile store such as stock and account.

**1.2.Objective**

The mobile mart system software is so designed as to ease the workload of mobile shop professionals. The main feature includes Inventory, and stock control and client management. Today’s world is computer world because most of work is doing with the help of computer. Dependency on computer is behind the few reasons. We cannot easily manage to store large number of data or information single handle. If we will be need some information or data in urgency then we cannot manage in manually these works are very difficult if we cannot use computer.

As the generic software it can be used by a wide variety of outlets (Retailer and Wholesalers) to automate the process of manually maintaining records related to the subject of maintain the Stock and Cash flow. This software is basically updating the manual mobile mart system to automated Inventory system. So that organization can manage their record in efficient way and organize them.

* The main objective is to automate non-computer environment.
* To save manpower.
* It will speed the processing of data and transaction.
* It will provide best security features such as provision of passwords.

**2.Purpose and Future Scope**

**1.3.1.Purpose**

The purpose of this document is to specify requirements and to give guidelines for the development of above said project. In particular it gives guidelines on how to prepare the above said project.

This document is intended to be a practical guide for people who developing this software.

**1.3.2. Scope**

The proposed system helps them in many ways. It helps them do billing very easily. Account maintenance also becomes easier. They can keep track of their contacts, feeding details, messages and records. The software is provided with all the master entries to enter any new contact, or data, to add or modify and delete.

As this is generic software it can be used by a wide variety of outlets (Retailers and Wholesalers) to automate the process of manually maintaining the records related to the subject of maintaining the stock and cash flow.

In future it can be modify, so that it can be done online. In order to meet these facility this is the major change which can be done in future regarding this project.

**3.SURVEY OF TECHNOLOGIES**

**2.1.JAVA**

Java is a popular general-purpose programming language and computing platform. It is fast, reliable, and secure. According to Oracle, the company that owns Java, Java runs on 3 billion devices worldwide.

Considering the number of Java developers, devices running Java, and companies adapting it, it's safe to say that Java will be around for many years to come.

This guide will provide everything you need to know about Java programming language before you learn it. More specifically, you will learn about features of Java programming, its applications, reasons to learn it, and how you can learn it the right way.

**2.2.MySQL**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

**4.REQUIREMENTS AND ANALYSIS**

**3.1.Problem Definition**

Problem introduction or problem starting is the starting point of the software development activity. The objective of this statement is to answer: Exactly *what must the system do*? The software project is initiated by the client’s need. In the beginning, these needs are on the minds of various people in the client’s organization. The analyst has to identify the requirements by talking to the people and understanding to their needs. It goes without saying that an accurate and through understanding of software requirement are essentials to the success of software development effort. All further development like system analysis. System design and coding will depends on how accurate and well understood the requirements are poorly analyzed and specified software will disappoint the user and will bring brief to the developer. No matter how well designed and well appearances are often deceiving. Chances of misinterpretation are very high, ambiguity is probable and communication gap between customer and developer is bound to bring confusions. Requirements understanding begin with a clear and concise heading stating in sentence the task to be performed. Then requirements are describe in a technical manner precise statement.

**Feasibility Study**

All projects are feasible given unlimited resources and infinite time! Unfortunately, the development of computer based system is more likely to be plagued by a scarcity of resources and difficult delivery dates. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Months or years of effort, Money loss and untold professional embarrassment can be averted I better understand the project at its study time. This type of study determines if an application can and should be developed. Once it has been determining that, application is feasible. After that analyst can go ahead and prepares the project specification, which finalizes project requirements. Feasibility studies are undertaken within tight time constraints.

1. **Technical Feasibility**
2. **Operational Feasibility**
3. **Economic Feasibility**
4. **Legal Feasibility**

**1.Technical Feasibility**

As we know the technical feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

* The facility to produce outputs of advertisements, shopping and mailing in a given time for ease of use.
* Response time under certain condition is minimal.
* Ability to process a certain volume of transaction at a particular speed.
* Facility to communicate data to distinct location.

In examining the technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system’s requirements- how many workstations are required, how these units are interconnected so that they could operate and communicate smoothly.

**2.Operational Feasibility**

Proposed projects are beneficial only if they can be turned into information system that will meet the financial management requirements of the business/organization. This test of feasibility asks if the system will work when it developed and installed. Are there major barriers to implementation?

Some of the important questions that are useful to test the operational feasibility of a project are given below:

* + Is there sufficient support for the project from the implementation? From user? If the present system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
  + Are current business methods acceptable to the user? If they are not, user may welcome a change that will bring about a more operational and useful system.
  + Have the user been involved in the planning and development of the Project? If they are involved at the earliest stage of project development, the chances of resistance can be possibly reduced.
  + Will the proposed system cause harm? Will it produce poorer result in any case or area?
  + Will the performance of staff member fall down after implementation? Issue that
  + Appears to be quite minor at the early stage can grow into major problem after Implementation.

**3.Economical Feasibility**

Economic analysis is the most frequently used technique for evaluating the effectiveness of the proposed system. More commonly known as cost/benefits analysis, the procedure is to determine the benefits and savings that are expected from the purposed system and compared with costs.

If benefits outweigh cost, a decision is taken to design and implement the system. Otherwise, further justification or alternative of the proposed system will have to be made if it has a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle. The analysis part also clears the doubt of economic problems which could be possible in developing the system. As already mentioned that the company has to just pay the developed software cost and not other investment is needed at the time of implementation of the new system as the preliminary requirements already exist in the company.

**4.Legal Feasibility**

In the legal feasibility is necessary to check that the software we are going to develop is legally correct which means that the ideas which we have taken for the proposed system will be legally implemented or not so, it is also an important step in feasibility study.

**3.2.Planning and Scheduling**

A project plan needs to be created to ensure the timely completion of the project. As part of project analysis, we break the project down to a number of stages and use a Gantt chart and PERT chart to describe specific tasks and status.

The Work Breakdown Structure of our proposed system “E-Commerce” is shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Task Name** | **Duration** | **Start** | **Finish** | **Predecessors** |
| 1 | Project Initiation | 2 days |  |  |  |
| 2 | Draft Project Plan | 5 days |  |  |  |
| 3 | Analysis Phase | 8 days |  |  |  |
| 4 | Plan User Interviews | 2 days |  |  |  |
| 5 | Schedule users Interviews | 4 days |  |  |  |
| 6 | Conducting users Interviews | 3 days |  |  |  |
| 7 | System Design | 15 days |  |  | 6 |
| 8 | Modules Design | 10 days |  |  |  |
| 9 | Data Structure Design | 5 days |  |  | 8 |
| 10 | User Interface Design | 4 days |  |  |  |
| 11 | Coding Phase | 36 days |  |  |  |
| 12 | Testing Phase | 10 days |  |  |  |
| 13 | Integration Testing | 5 days |  |  |  |
| 14 | System Level Testing | 6 days |  |  |  |
| 15 | Implementation | 3 days |  |  |  |
| 16 | Post-Implementation Review | 4 days |  |  |  |

**3.3.Requirements Specification**

**3.3.1.Softwar**

* Java/JDK
* NetBeans
* MySQL
* SQL YOG

**3.3.2.Hardware**

* Pentium IV Processor
* 512 MB RAM
* 40 GB HDD
* Color Monitor
* Keyboard, Mouse

**3.4.Preliminary Product Description**

* **Login:** This module has a drop-down list box from where we have to select

**ADMIN or USER**. The **ADMIN** has all the rights in the software including updating the status of his site. The other fields in login **are** username and password. If the username and password are correct then it is directed to next page.

* **New user:** This module is for the users who do not have their account. Here user is allowed to create an account to login. The account creation is done by filling the registration form with user details such as name, phone, email etc.
* **Product:** This module has information regarding the mobiles such as its name, model, color, price information, its features etc. The **ADMIN** has the authority to Add, Delete, Update etc. The **USER** can only view the Mobile, add to cart only those in the stock etc.
* **Accessories:** This module consists of various available accessories of the Mobile with its name and picture, price information etc.
* **Search:** This module helps the customer to ease his search based on his budget or interest. The search can be done on different categories like mobile model name, model number, color, price etc.
* **Cart:** User can select any number of Mobile and add to the cart. He can also remove from the cart if he dislikes it later.
* **Payment:** This module describes the payment done by the customer. The payment information can include information like the model purchased, quantity, mode of payment (cash, loan) etc.
* **Stocks:** This gives the details regarding the products available for sale.

**5.Conceptual Models**

**SYSTEM ARCHITECTURE DESIGN**

**1.Defining a System**

Collection of component, which are interconnected, and work together to realize some objective, from a system. There are three components in every system, namely input, processing and output

Input Output

Processing

**SYSTEM DEVELOPMENT LIFE CYCLE**

The **System development life cycle (SDLC)**, or **Software development processing** [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering), [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), is a process of creating or altering information systems, and the models and [methodologies](http://en.wikipedia.org/wiki/Methodologies) that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies). These methodologies form the framework for planning and controlling the creation of an information system the [software development process](http://en.wikipedia.org/wiki/Software_development_process).

Broadly, following are the different activities to be considered while defining the system development life cycle for the said project:

* Problem Definition
* System Analysis
* Study of existing system
* Drawback of the existing system
* Proposed system
* System Requirement study
* Data flow analysis
* Feasibility study
* System design
* Input Design (Database & Forms)
* Updating
* Query /Report design
* Administration
* Testing
* Implementation
* Maintenance

**1.2.System Analysis**

**Systems analysis** is the study of sets of [interacting](http://en.wikipedia.org/wiki/Interaction) [entities](http://en.wikipedia.org/wiki/Entity), including computer systems analysis. This field is closely related to [requirement analysis](http://en.wikipedia.org/wiki/Requirement_analysis) or [operations research](http://en.wikipedia.org/wiki/Operations_research). It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made. System development can generally be thought of having two major components: systems analysis and systems design. In System Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvement to the system.

**1.3.System Design**

Systems design is the process of defining the architecture, components, modules, interfaces, and [data](http://en.wikipedia.org/wiki/Data) for a [system](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement). One could see it as the application of [systems theory](http://en.wikipedia.org/wiki/Systems_theory) to [product development](http://en.wikipedia.org/wiki/Product_development). There is some overlap with the disciplines of [systems analysis](http://en.wikipedia.org/wiki/Systems_analysis), [systems architecture](http://en.wikipedia.org/wiki/Systems_architecture) and [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering). If the broader topic of [product development](http://en.wikipedia.org/wiki/Product_development) "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing [systems](http://en.wikipedia.org/wiki/System) to satisfy specified [requirements](http://en.wikipedia.org/wiki/Requirement) of the user. Until the 1990s systems design had a crucial and respected role in the [data processing](http://en.wikipedia.org/wiki/Data_processing) industry. In the 1990s [standardization](http://en.wikipedia.org/wiki/Standardization) of hardware and software resulted in the ability to build [modular](http://en.wikipedia.org/wiki/Modularity_(programming)) systems. The increasing importance of software running on generic platforms has enhanced the discipline of [software engineering](http://en.wikipedia.org/wiki/Software_engineering). [Object-oriented analysis and design](http://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) methods are becoming the most widely used methods for computer systems design. The [UML](http://en.wikipedia.org/wiki/Unified_Modeling_Language) has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

**ENTITY RELATION DIAGRAMS**

The Entity Relation Model or Entity Relation Diagram (ERD) is a data model or diagram for high-level description of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity relationship diagrams. Such models are typically used in the first stage of Management information system design, they are used for example, to describe information needs and/ or the type of information that is to be stored in the Database during the requirement analysis.

In the case of design a Management Information System that is based on a database, the conceptual data model is, a later stage (usually called logical design), mapped to a logical data model such as, relational data model; this is turn in mapped to a physical model during physical design. Note that sometimes, both of the phases are referred a “physical design”. There are number of convention for entity-relation diagrams (ERDs). The classical notation is describe in the remainder of this article, and mainly related to the conceptual modeling. There is a range of notation more typically employed in physical and logical database design.

Request

Mobile Detail

Employee

Place Order

Check Login ID

Order Detail

Login

Request new member

Registration

**DATA FLOW DIAGRAM**

The data flow diagram shows the flow of data within any system. It is an important tool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It’s also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD.

In the DFD, four symbols are used and they are as follows.

1. A square defines a source (originator) or destination of system data.
2. An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.
3. A circle or a “bubble “(Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.
4. An open rectangle is a data store-data at rest, or a temporary

Repository of data.

**Context Level Data Flow Diagram**

This level shows the overall context of the system and its operating environment and shows the whole system as just one process. Online book store is shown as one process in the context diagram; which is also known as zero level DFD, shown below. The context diagram plays important role in understanding the system and determining the boundaries. The main process can be broken into sub-processes and system can be studied with more detail; this is where 1st level DFD comes into play.

**Zero Level DFD**

PRODUCT

COUSTOMER

Request

ADMIN

Request fulfil

Zero Level Data Flow Diagram

**First Level DFD**

This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major high-level processes of the system and their interrelation. A process model will have one, and only one, level-1 diagram. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level 1.

Enter the software

****

Display errors

Admin

Employee

Buy product

View product

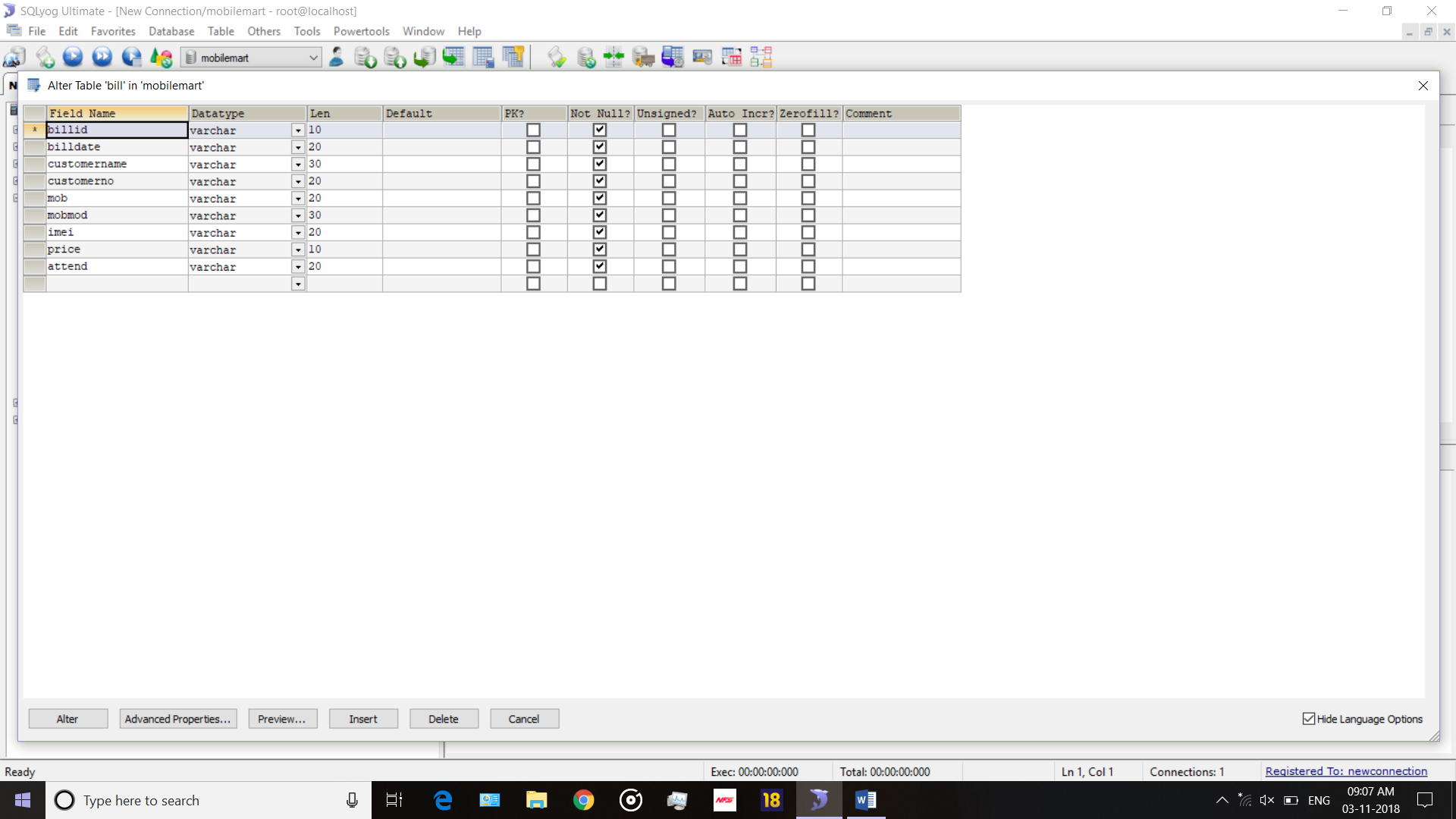
Customer Details

Details

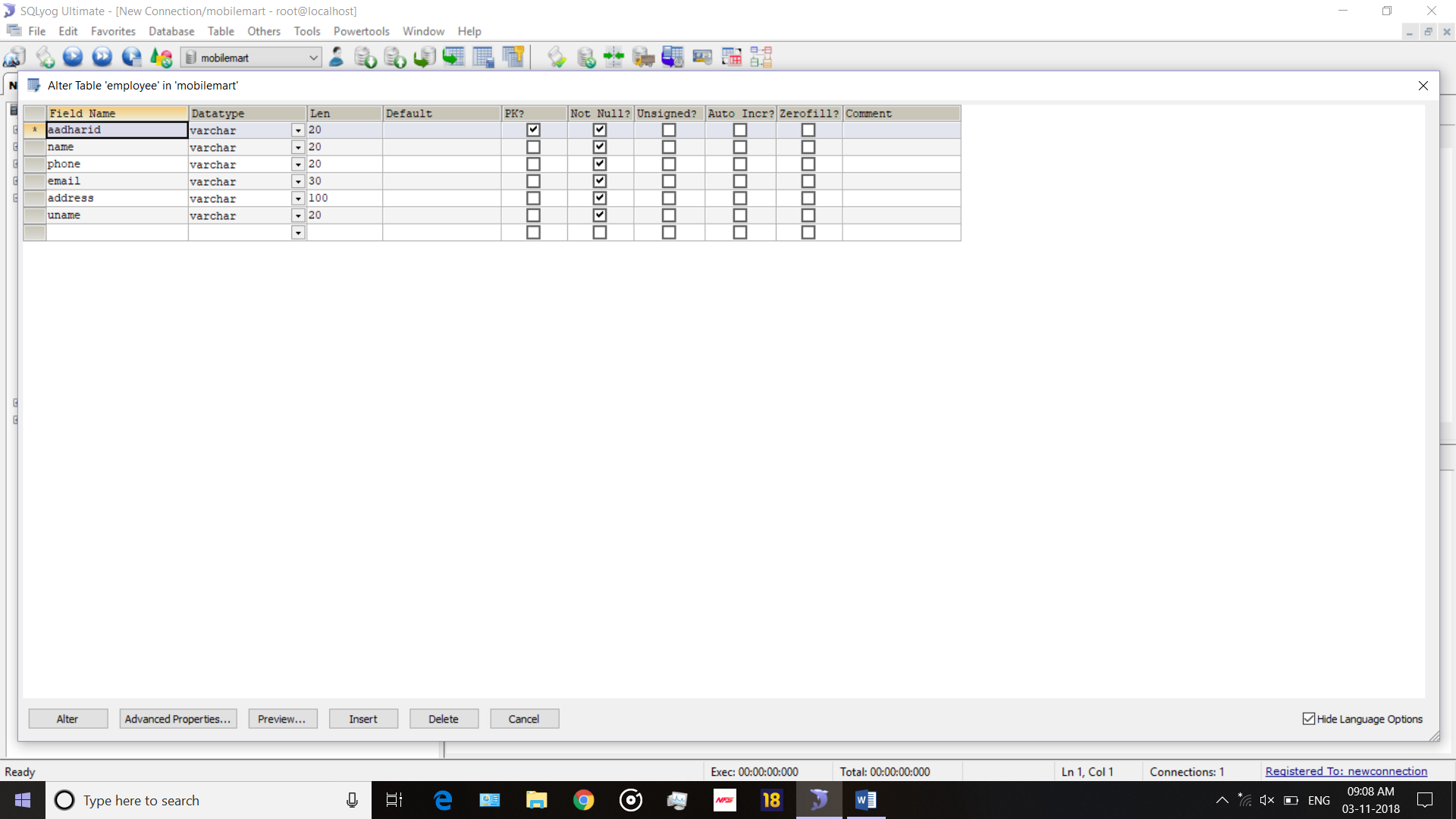
Products

**Data Tables:-**

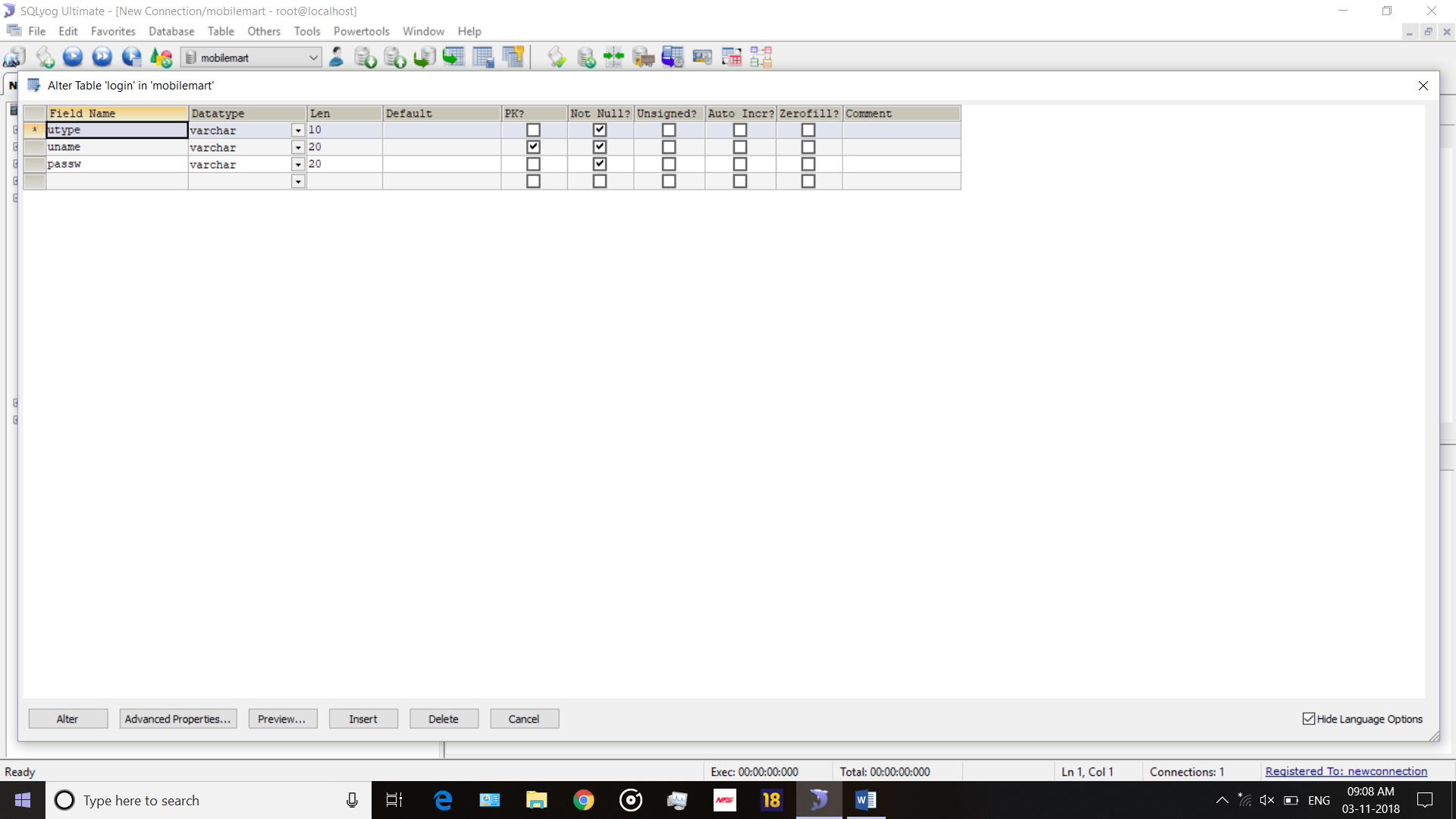
**Bill**



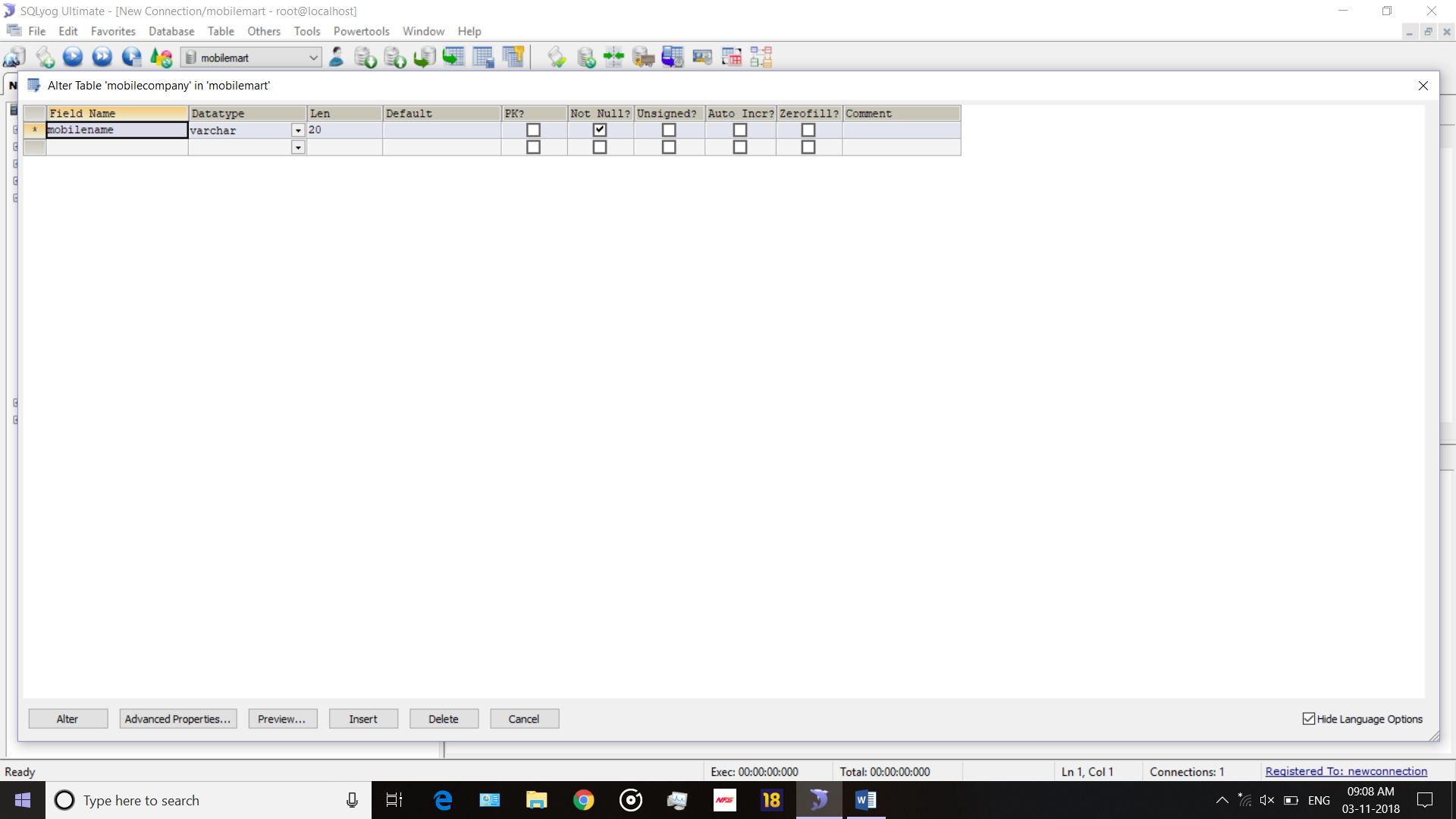
**Employee**



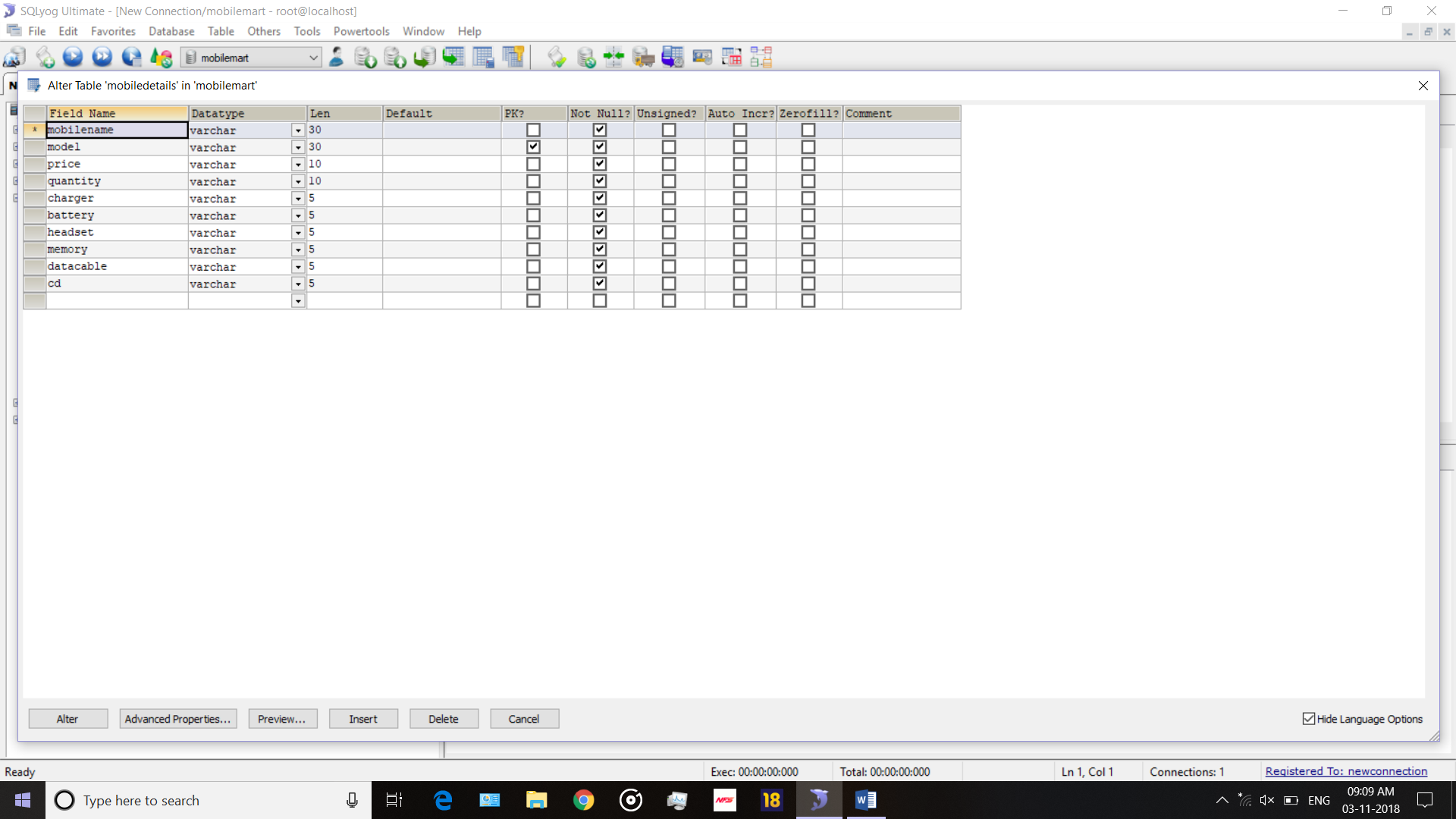
**Login**



**Mobile company**



**Mobile Details**



**LOGIN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Column** | **Data Type** | **Constraints** | **Size** |
| User Name | Varchar | Primary Key | 20 |
| User type | Varchar | Not Null | 10 |
| Password | Varchar | Not Null | 20 |

**Mobile Company**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Column** | **Data Type** | **Constraints** | **Size** |
| Model name | Varchar | Not Null | 20 |

**Mobile Details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Column** | **Data Type** | **Constraints** | **Size** |
| Mobile Name | Varchar | Not Null | 30 |
| Model | Varchar | Primary Key | 20 |
| Price | Numeric | Not Null | 7,2 |
| Quantity | Varchar | Not Null | 10 |
| Charger | Varchar | Not Null | 5 |
| Battery | Varchar | Not Null | 5 |
| Headset | Varchar | Not Null | 5 |
| Memory | Varchar | Not Null | 5 |
| Data cable | Varchar | Not Null | 5 |
| Cd | Varchar | Not Null | 5 |

**BILL**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Column** | **Data Type** | **Constraints** | **Size** |
| Bill id | Number | Not Null | 10 |
| Customer name | Varchar | Not Null | 20 |
| Mobile | Varchar | Not Null | 20 |
| Price | Numeric | Not Null | 7,2 |
| Customer number | Varchar | Not Null | 20 |
| Bill date | Date | Not Null | 20 |
| Mobile Module | Varchar | Not Null | 20 |
| imei | Varchar | Not Null | 10 |

**Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name Column** | **Data Type** | **Constraints** | **Size** |
| Aadhar id | Varchar | Primary Key | 30 |
| Name | Varchar | Not Null | 20 |
| Phone | Numeric | Not Null | 7,2 |
| E-mail | Varchar | Not Null | 20 |
| Address | Varchar | Not Null | 100 |
| User Name | Varchar | Not Null | 20 |