FINAL REPORT
ON

SMS P2A AND A2P APPLICATION SERVICE

By
(Rishabh Rawat)
MTS Sistema Shyam Teleservices Ltd.

Company Profile

Mobile operator company · MTS is the largest mobile operator in Russia and CIS with over 102.4 million subscribers as of 31 December 2009.	Commercial organization
Stock price	: MTSS (MCX) RUB 211.30 -1.70 (-0.80%)
Headquarters	: Moscow, Russia
CEO:	Andrei A. Dubovskov
Founded:	October 1993
Parent organization:	Sistema
Industry	Telecommunications
Products	Mobile telephony Wireless broadband
Parent	Sistema (56.68%) Shyam Group (23.98%) Russian government (17.14%)
Slogan	You know that you can!
Website	www.mtsindia.in

Company Overview

Sistema Shyam TeleServices Limited (SSTL), also known as Mobile TeleSystems and commonly referred to by the abbreviation MTS, is the Indian subdivision of Russian Mobile TeleSystems telecommunication company headquartered in New Delhi, India. It provides wireless voice, broadband Internet, messaging and data services in India. MTS India is a subsidiary of Russian conglomerate Sistema and is the ninth largest mobile operator in India with 8.9 million subscribers as of June 2015.

History

Sistema, the largest public diversified corporation in Russia, acquired a 10% stake in Shyam Telelink for a total cash consideration of US$ 11.4 million at the end of September 2007. Shantanu Telecom along with their partner Sistema had applied for UASL licence in 22 telecom circles of India. In August 2008, they got a pan-India start-up spectrum to start their mobile service operations in the country. They provide mobile services based on CDMA technology under the brand name MTS India and gave contracts to ZTE and Huawei for network expansion. MTS launched operations in Uttar Pradesh East and West in October 2010

Services

MTS launched EVDO Rev A based high-speed mobile broadband service, MBlaze, in November 2009 and has seen tremendous market acceptance with over 5 lakh (As per February 2011 Data) customers in a short span of time. In April 2010, MTS launched MTS TV for MTS MBlaze customers. MTS MBlaze have its coverage in 100+ cities as of February 2011. MTS has also announced pan-India roaming for its users in April–May 2010. MTS also provides MBrowse which is CDMA-1x technology based internet service. According to Vsevolod Rozanov, President & CEO, MTS India, With a view to enhance customer experience, MTS has prepared a blue print to take the mobile broadband usage in the country to the next level. In a phased manner, MTS will be launching seamless HSD services across some of the busiest highways in the country. On 5 September, MTS India announced the commercial availability of its EVDO Rev B Phase 2 network. On 23 October 2013 SSTL announced its roll out plan of its 3GPLUS network in nine circles namely Delhi, Rajasthan, Gujarat, Kerala, Karnataka, Tamil Nadu, Kolkata, Uttar Pradesh (West) and West Bengal.

Acknowledgement

This Interim Report highlights the project details pursued by Rishabh Rawat at the MTS Sistema Shyam Teleservices Limited, Jaipur in accordance with the Technical Internship Program (TIP) in curriculum of Semester VII of B – Tech.

The project is a keyword based SMS application service that helps users to retrieve information about recharge plans that are available or suitable for their current tariff plans. It is a pull based application, a value added service that is free of cost that recommends to the user the available plans he can choose to activate.

The project assigned to the intern is to study the concept of SMS as a Value added service, and then to apply that knowledge to create a keyword based application that provides a service to the customer when they ask for it.

Therefore, this report has demarcated the content accordingly to facilitate easier reading. It gives the detailed summary of the work done by the intern as per the project schedule, given by the company.

Project Introduction

These days, Telecom Companies come up with a lot of plans and schemes for data and network usage. Often, the operators convey this information to their users in personal messages using Value added services like Short Message Service (SMS) or by recorded calls. These calls are often very tedious to the user and generally ignored. MTS proposes an application which uses a Pull mechanism, where in a user sends a message (SMS) to an operator provided Short Code. Instead of receiving all irrelevant messages, the user can request for those which would best fit his requirements.

The user sends an SMS to the service operator, which is routed to the application server, which uses a keyword. After receiving a request with the specific keyword, the application maps relevant plans from a database. The database is an integrated pool of information based on user's current plans and data. The application extracts all plans which could help a user. For example, if a user tends to stay on Roaming a lot, the application would offer him plans that reduce Roaming rates. This is also called a person to application (p2a) service, where the person sends a message to the application, which responds with an auto generated reply unique to each user number.

The first task is to understand and learn the concepts of Value Added Services, SMS in particular. We look into its various uses and applications, as well as the technical flow of a message originating from a mobile device till its termination. This includes important features and working of SMSC and MSC, explained in detail later.

Next, we build a database schema using existing data, which consists of Customer Details, Region details, Tariff plans, Recharge Descriptions etc. The database we used here is Oracle, by connecting to the company server. Since the data is on a very large scale, it is difficult to insert and manipulate into the database manually. For this purpose we use software (TOAD), which makes it easy to import data and manipulate it.

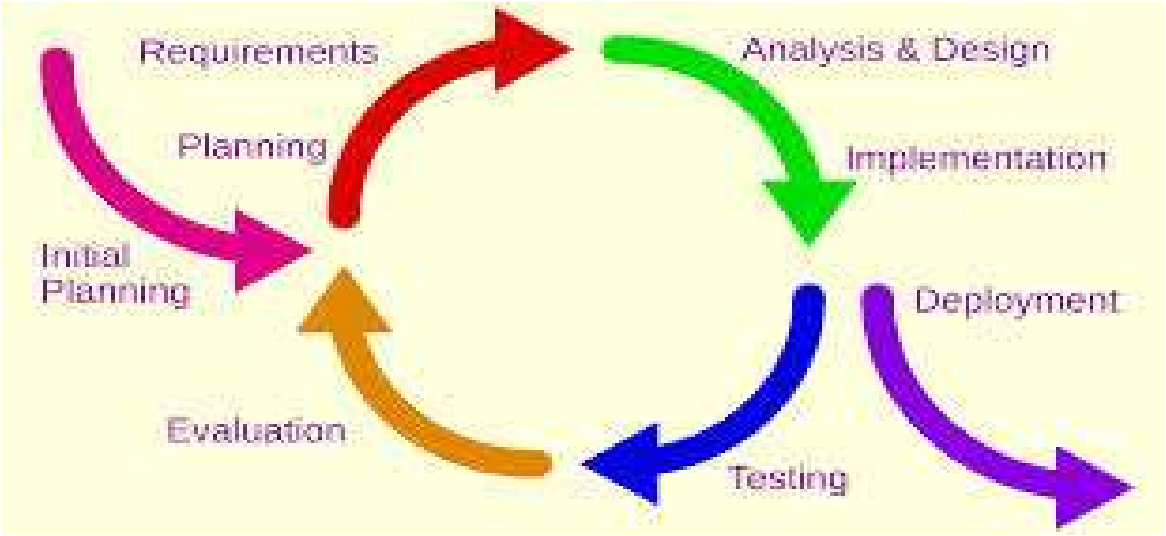
After we've built the schema, we move towards the making of application. The most used and relevant platform for building automated applications in Telecom Industry is ASP (Active Server Pages). We first study the language basics, and then move towards coding, which is loosely divided into parts.

The last part would include testing and deployment, starting in the latter half, once we've successfully accomplished the above parts.

Iterative Model

In Iterative model, the process begins with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions till the complete system is implemented and ready to be deployed.

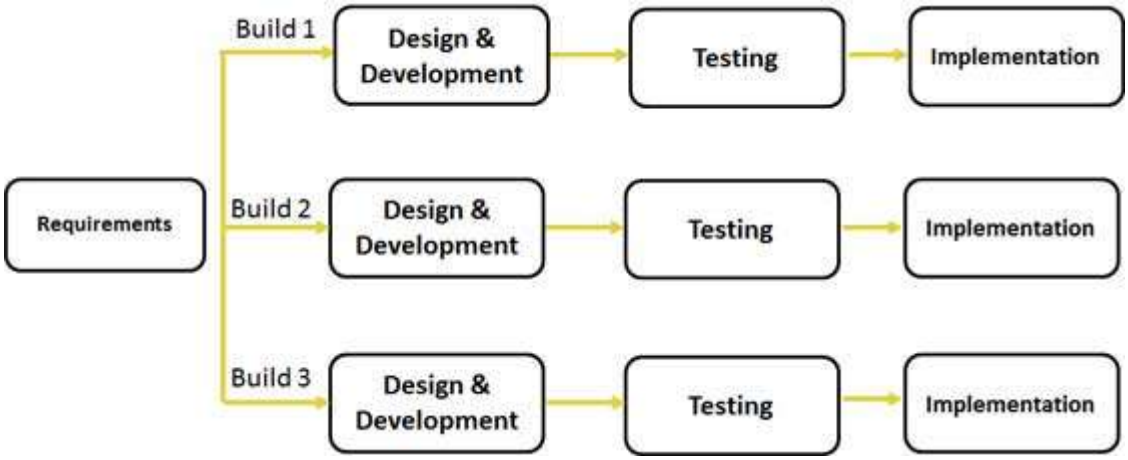
An iterative life cycle model does not attempt to start with full specification of requirements. Instead, development begins by specifying and implementing just parts of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each version of the model.



Iterative Model design

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. With each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this is to develop a system through repeated cycles i.e. iterative and in smaller portions at a time (incremental).

Following is the pictorial representation of Iterative and Incremental model:-



Iterative and Incremental development is a combination of both iterative design and incremental build model for development. During software development, more than one iteration of the software development cycle may be in progress at the same time and this process may be described as an "evolutionary acquisition" or "incremental build" approach.

In incremental model the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

Working of SMS

The Short Message Service (SMS) is a method of personal data communication using SS7 of mobile networks to carry short packets. An SMSC is an entity that delivers short messages between mobile subscribers and between mobile subscribers and various External Short Message Entities (ESMEs), such as voice mailboxes, integrated information centers, and manual agents. In addition, an SMSC provides a series of value-added services on this base. The "short" here means that the length of each message is limited.

In recent years, mobile communications have been developing rapidly worldwide. To help network operators win more subscribers and profits, the SMSC provides a stable information carrier platform in addition to the point-to-point (P2P) short message service (SMS).

The delivery and transfer of point-to-point short messages are implemented by the SMSC. The SMSC has similar functions as a postal office. It receives a short message from the source and then forwards it to the destination. The major functions of the SMSC are to receive, store, and forward short messages. Through the SMSC, short messages can be delivered to the destination in a more reliable way. If a delivery failed, the SMSC will keep the short message until it is delivered successfully or failed permanently. Following terms explain types of short messages.

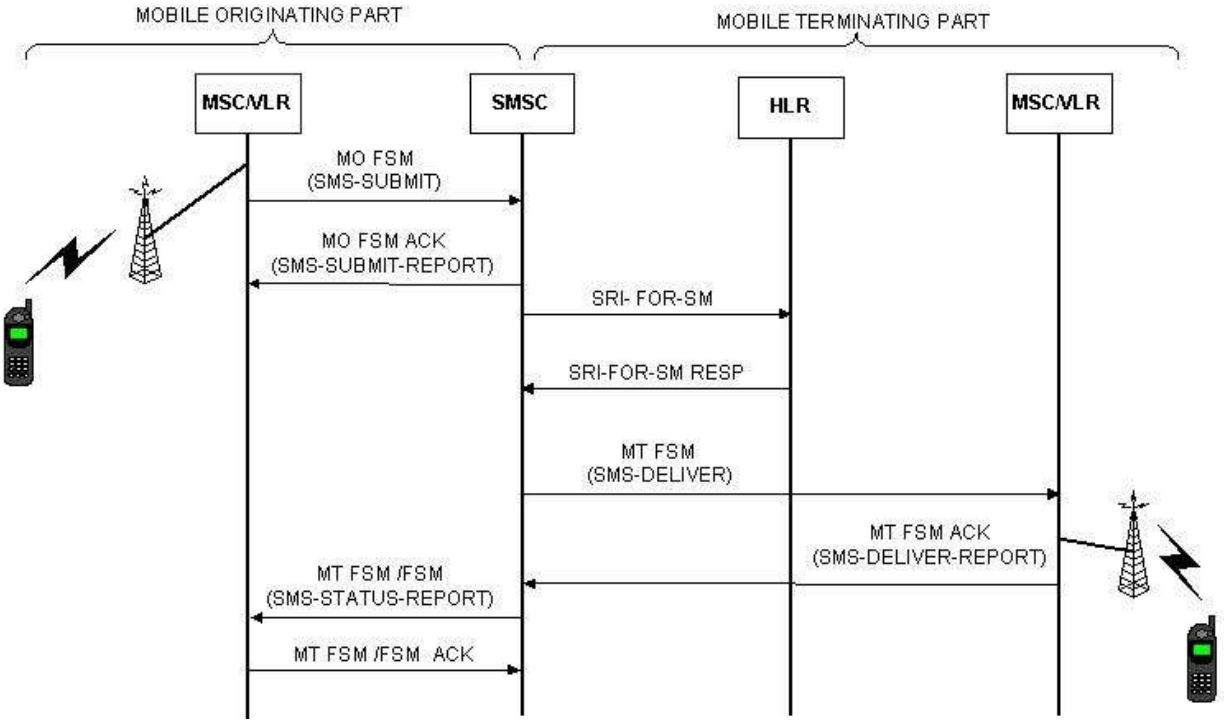
- MO: Short messages originated by mobile stations. It is a process that a mobile station submits a short message to the SMSC till it receives a response from the SMSC.

- MT: Short messages terminated by mobile stations. It is a process that the SMSC sends a short message to the destination mobile station following certain rules till it receives a response from the mobile station.

- AO: Short messages originated by ESMEs. It is a process that an ESME submits a short message to the SMSC till it receives a response from the SMSC.

- AT: Short messages terminated by ESMEs. It is a process that the SMSC sends a short message to the destination ESME following certain rules till it receives a response from the ESME.

A short message originated from a mobile subscriber is sent to the service processing module via the SMS signaling gateway.

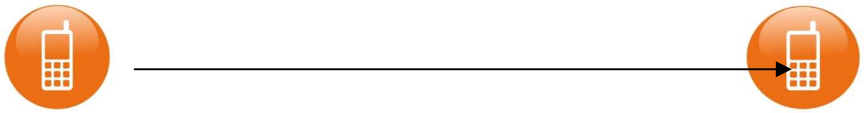

MOBILE ORIGINATING PART
MOBILE TERMINATING PART
MSC/VLR
SMSC
HLR
MSC/VLR
MO FSM
(SMS-SUBMIT)
MO FSM ACK
(SMS-SUBMIT-REPORT)
SRI- FOR-SM
SRI-FOR-SM RESP
MT FSM
(SMS-DELIVER)
MT FSM ACK
(SMS-DELIVER-REPORT)
MT FSM /FSM
(SMS-STATUS-REPORT)
MT FSM /FSM ACK

Types

Person-to-Person (P2P), Application-to-Person (A2P) and Person-To-Application (P2A) are the three categories of SMS messages. While P2P messages connect one individual to another or you can say Mobile to Mobile, A2P messages are used to interact with consumers via an SMS application and P2A messages are used by consumers to interact with SMS application.

Person-to-Person(P2P)

Mobile to Mobile SMS are used majorly for personal interaction with family, friends & co-workers. Today application platform like WhatsApp has changed the way we looked at Short Messaging Service world over.

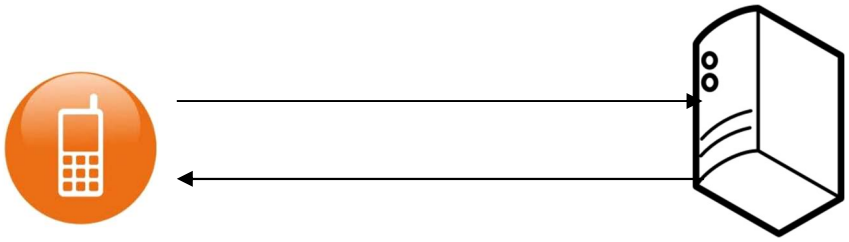


Application-to-Person(A2P)

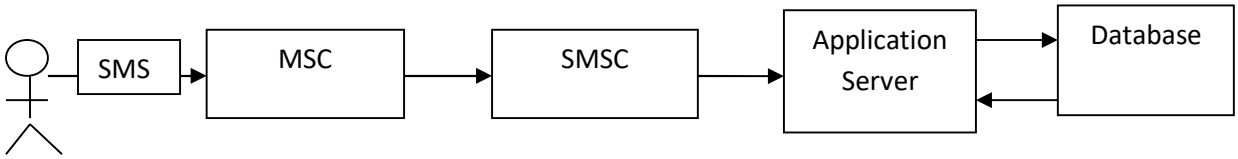
A2P SMSes are generated by application to interact with customer. We get Auto generated messages from Banks, Airlines, Fee payment, Stock Market, etc. Even we get Alert SMS messages from various sources as and when something happens. These types of SMS are mostly informative & we love to receive them. A2P messages are also used for promotional activities & we hate most of such SMSes coming.

Person-to-Application(P2A)

When consumer want some information or wants to participate in activity, he sends SMS to number which in turn replies back automatically. If you want to know your bank balance using SMS, you have to send one SMS on particular number with prescribed format & in turn you get answer from SMS application hosted in bank with you bank balance details. You can also participate in Voting, Lucky Draw, Opt-In, Competition, Text-To-Win, etc. by sending SMS.
All above categories are widely used for gaming platform, e-governance, etc.



Our application pertains to user or person or peer to application and vice versa interaction. The basic flow diagram of the project is like this-



SMS Call Flow(P2A)

This diagram shows that the user sends an SMS with a specific Keyword, which is sent to the nearest MSC (Mobile Switching Center), which then routs the path to the SMSC of the particular operator eg MTS, Vodafone, etc. The message is then sent to the application server which is capable of processing it and providing a suitable response. The application server runs the code, maps results from the database and returns them to the SMSC in a response SMS, which then travels the same route, back.

Since we need a platform to store data to access and map, we use ORACLE.

Oracle

Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system produced and marketed by Oracle Corporation. An Oracle database system-identified by an alphanumeric system identifier or SID, comprises at least one instance of the application, along with data storage. An instance-identified persistently by an instantiation number (or activation id: SYS.V_$DATABASE.ACTIVATION#)-comprises a set of operating-system processes and memory-structures that interact with the storage. Users of Oracle databases refer to the server-side memory-structure as the SGA (System Global Area). The SGA holds cache information such as data-buffer, SQL Commands, and user information.

The Oracle DBMS can store and execute stored procedures and functions within it. PL/SQL (Oracle Corporation's proprietary procedural extension to SQL), or the object-oriented language Java can invoke such code objects and/or provide the programming structures for writing them.

SQL

We use SQL to manipulate and update information in tables.

SQL (Structured Query Language) is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

Originally based upon relational algebra and tuple (row) relational calculus, SQL consists of a data definition language and a data manipulation language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Some basic commands-

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table

SQL Commands are divided into categories. The ones we use are:

- Data Definition Language (DDL) Statements
- Data Manipulation Language (DML) Statements
- Transaction Control Statements

Data definition language (DDL) statements mainly perform these tasks:

- Create, alter, and drop schema objects
- Grant and revoke privileges and roles
- Analyze information in a table or index
- Establish auditing options
- Add comments to the data dictionary

Data Manipulation Language (DML) Statements

Data manipulation language (DML) statements access and manipulate data in existing schema objects. These statements do not implicitly commit the current transaction. The data manipulation language statements are:

CALL
DELETE
INSERT
MERGE
SELECT
UPDATE

The SELECT statement is a limited form of DML statement in that it can only access data in the database. It cannot manipulate data in the database, although it can operate on the accessed data before returning the results of the query.

Transaction Control Statements

Transaction control statements manage changes made by DML statements. The transaction control statements are:

COMMIT
ROLLBACK

Commit statement ends your current transaction and makes permanent all changes performed in the transaction. A transaction is a sequence of SQL statements that the Oracle Database treats as a single unit.

Rollback Statement tolls back an explicit or implicit transaction to the beginning of the transaction, or to a save-point inside the transaction.

Each column value and constant in a SQL statement has a data type, which is associated with a specific storage format, constraints, and a valid range of values. When you create a table, you must specify a data type for each of its columns.

Although SQL is a standard for most of the database platforms, small changes are present for each of them. For example, data type integer is specified as INTEGER in MySQL while the one we use-Oracle specifies it as NUMBER.

Here is a list of the most commonly used data types:

The CHAR data type stores fixed-length character strings.

The VARCHAR2 data type stores variable-length character strings.

The LOB data types for character data are CLOB and NCLOB. They can store up to 8 terabytes of character data (CLOB) or national character set data (NCLOB).

The NUMBER data type stores fixed and floating-point numbers. Numbers of virtually any magnitude can be stored and are guaranteed portable among different systems operating Oracle Database, up to 38 digits of precision.

Optionally, you can also specify a precision (total number of digits) and scale (number of digits to the right of the decimal point):

Syntax: Column_name NUMBER (precision, scale)

The DATE data type stores point-in-time values (dates and times) in a table. The DATE data type stores the year (including the century), the month, the day, the hours, the minutes, and the seconds (after midnight).

TOAD

Toad is a software application from Dell Software that database developers, database administrators, and data analysts use to manage both relational and non-relational databases using SQL.

Toad for Oracle provides an intuitive and efficient way for database professionals of all skill and experience levels to perform their jobs with an overall improvement in workflow effectiveness and productivity.

With Toad for Oracle, one can:

- Understand database environment through visual representations
- Meet deadlines easily through automation and smooth workflows
- Perform essential development and administration tasks from a single tool
- Deploy high-quality applications that meet user requirements; perform predictably and reliably in production
- Validate database code to ensure the best-possible performance and adherence to best practice standards
- Manage and share projects, templates, scripts, and more with ease

Toad has separate buttons to Commit or Rollback any statements, one can choose to auto commit after every statement, although it is advised not to, and by default it does not, since mistakes can be made and database can be falsely modified.

Functions we used in TOAD-

Database browser- To specify and connect to a specific database.

Schema Browser- Presents a list of all tables and views in the database.

New- To open new tab either in SQL or PL/SQL type or others.

Commit- To make permanent changes made by statements.

Rollback- To go back to initial state.

Save- Save the SQL script so that it can be opened later.

Active Server Pages(ASP)

- Active Server Pages (ASP), also known as Classic ASP or ASP Classic, was Microsoft's first server-side script engine for dynamically generated web pages. Initially released as an add-on to Internet Information Services (IIS) via the Windows NT 4.0 Option Pack (ca. 1996), it was subsequently included as a free component of Windows Server (since the initial release of Windows 2000 Server). ASP.NET, first released in January 2002, has superseded ASP. ASP is a program that runs inside IIS. An ASP file is just the same as an HTML file. An ASP file can contain text, HTML, XML, and scripts. Scripts in an ASP file are executed on the server. An ASP file has the file extension ".asp".

ASP can-

- Dynamically edit, change, or add content of a Web page
- Respond to user queries, and/or data submitted from HTML forms
- Access any data, databases and return the results to a browser
- Customize a Web page to make it more useful for individual users
- The advantages of using ASP instead of CGI and Perl, are those of simplicity and speed
- Provide security - since ASP code cannot be viewed from the browser
- Clever ASP programming can minimize the network traffic.

We use ASP to develop and deploy our code.

With the help of the #include directive, we can insert the contents of one ASP file into another, before the server executes it. Since we need to connect to the same database schema for all programs and purposes, it is prudent to create a common file to connect to the particular database and subsequently include the same into multiple other programs which serve various properties. This is what we do while developing our code.

This is how we use the include directive-

<!--#include file=”example.asp”-->

Where, example.asp is the other asp file the contents of which we wish to include in the current one.

Connection to Database

ADO is used to access database from Web Servers.

ADO is a Microsoft technology. It stands for ActiveX Data Objects.

ADO is a Microsoft Active-X component.

ADO is automatically installed with Microsoft IIS.

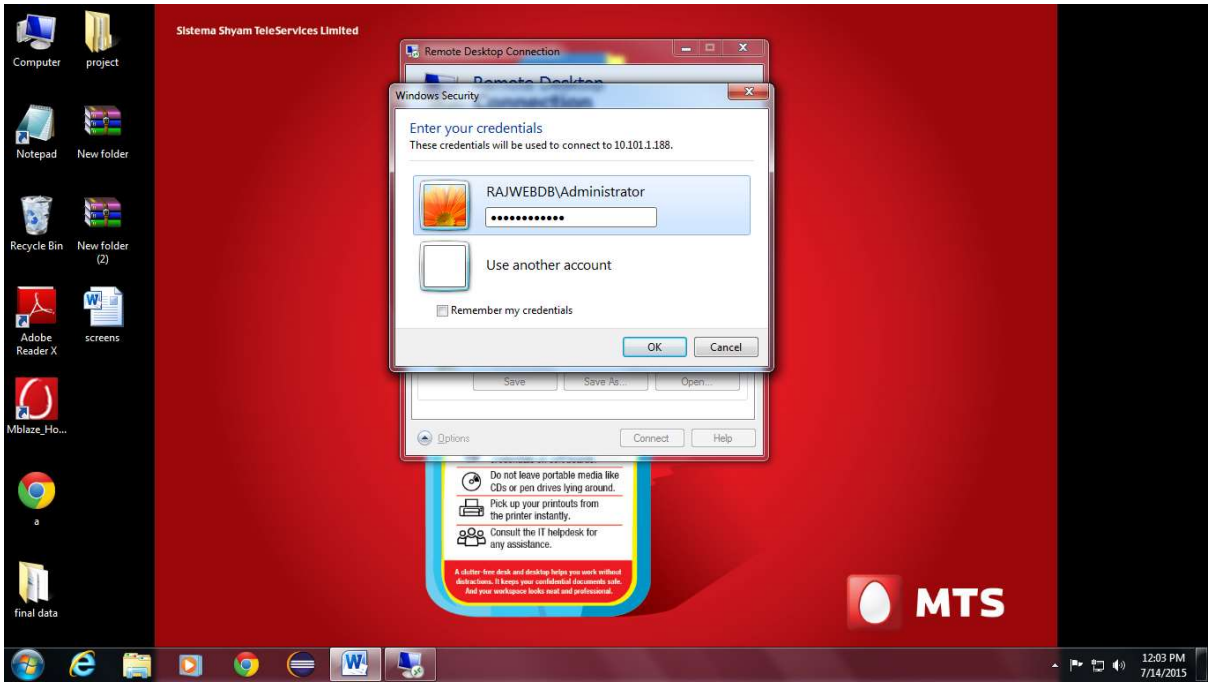
ADO is a programming interface to access data in a database.

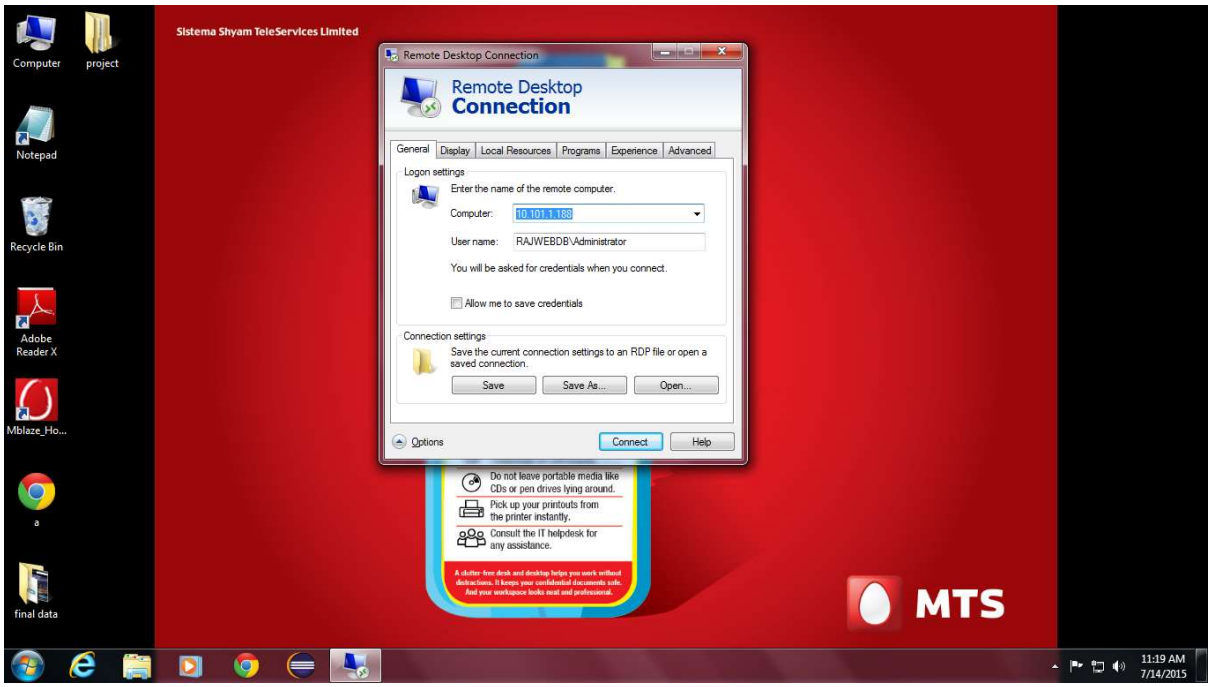
The common way to access a database from inside an ASP page is to:

1. Create an ADO connection to a database
2. Open the database connection
3. Create an ADO recordset
4. Open the recordset
5. Extract the data you need from the recordset
6. Close the recordset
7. Close the connection

Screenshots

Connecting to MTS Server

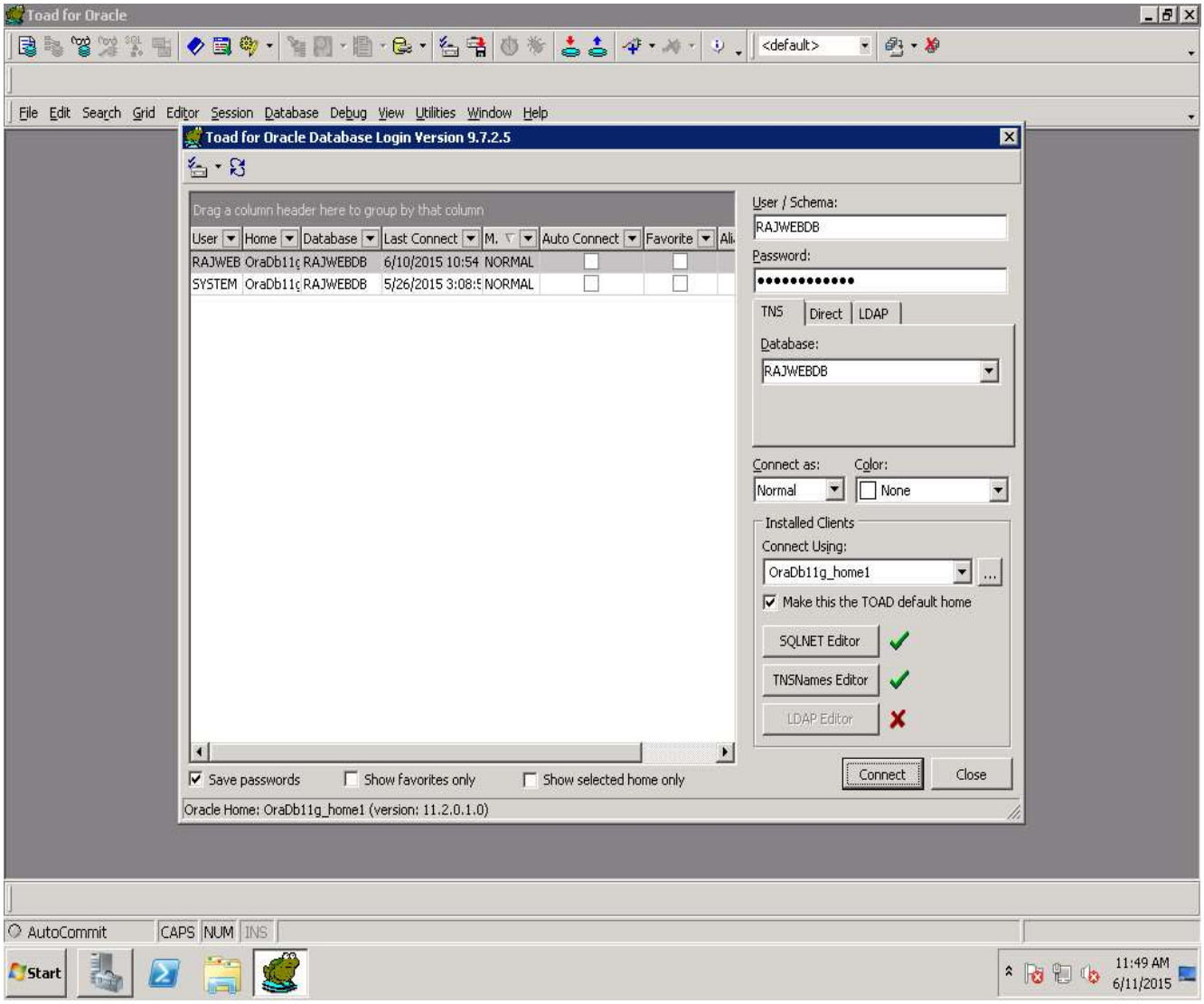




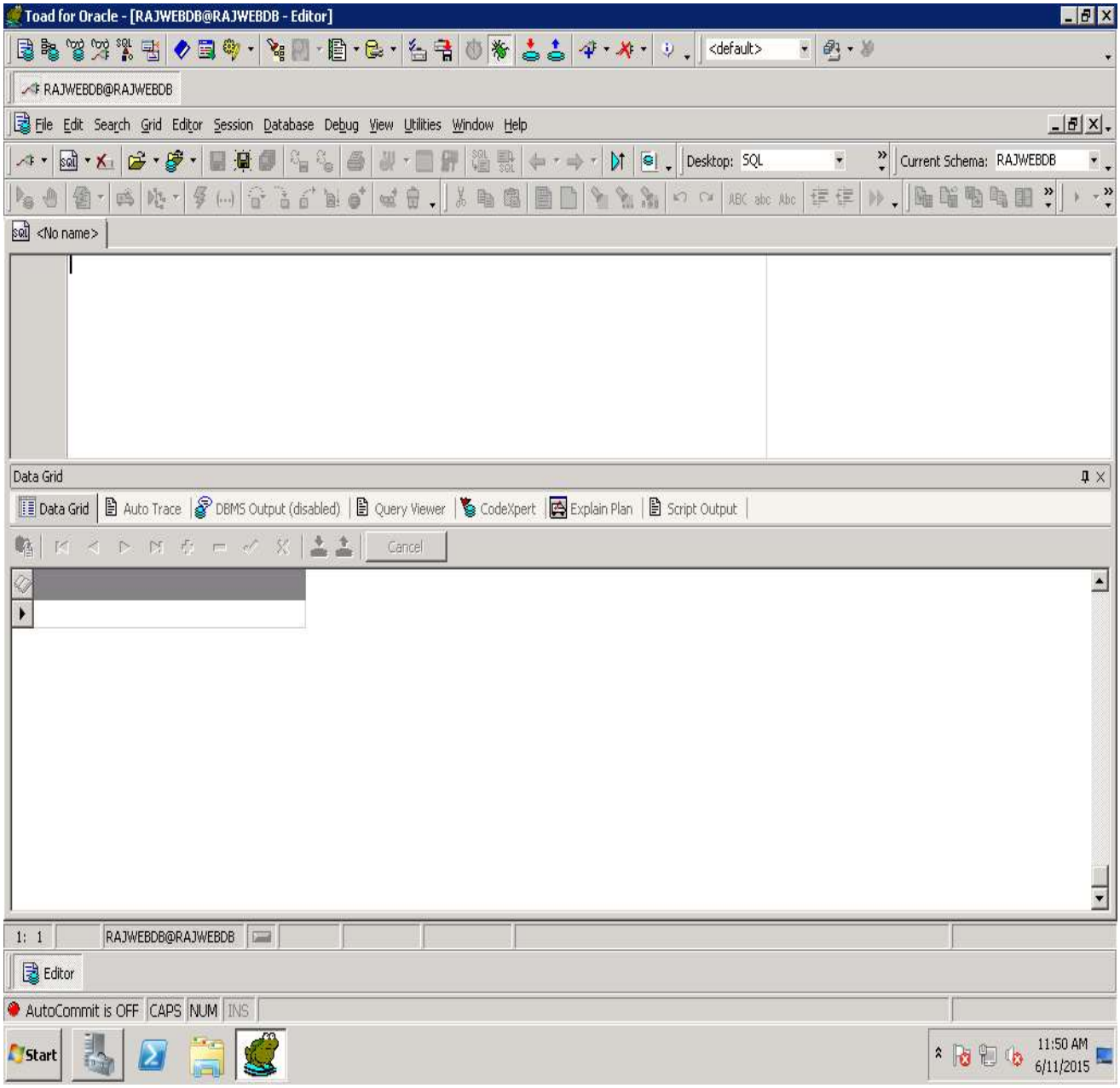

Computer
PREPAID5
New folder (3)
Recycle Bin
raj_activati...
TABLES
raj_opkg_d...
raj_postpai...
eclipse-java...
sylink
raj_activati...
New folder
Toad for Oracle 9.7.2
raj_opkg_d...
PREPAID5
New folder (2)
raj_activati...
raj_postpai...
Start
12:14 PM
7/15/2015

TOAD

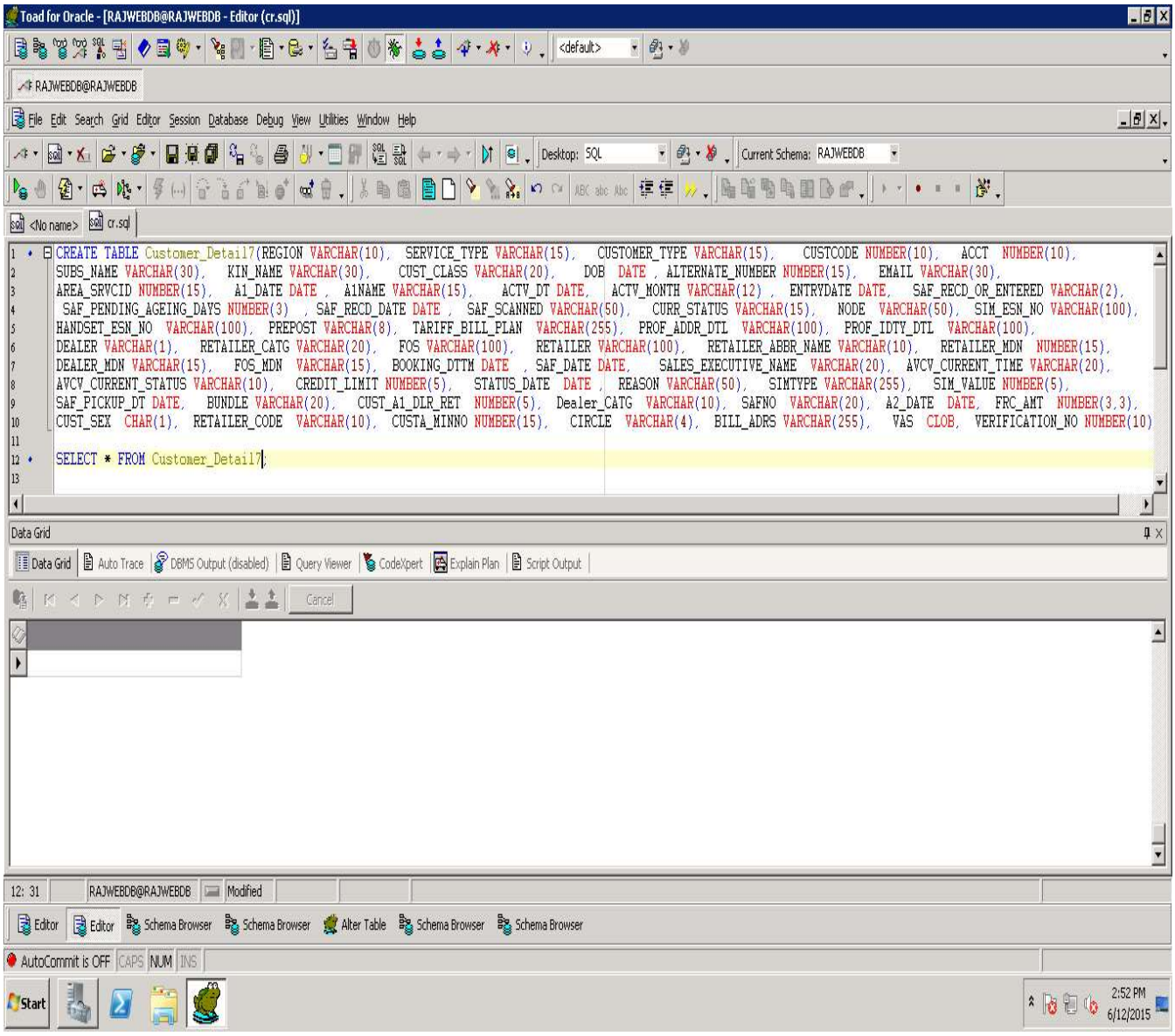
Here is the screenshot of the opening page of TOAD, when we connect to the database. The version we are using is 9.7.



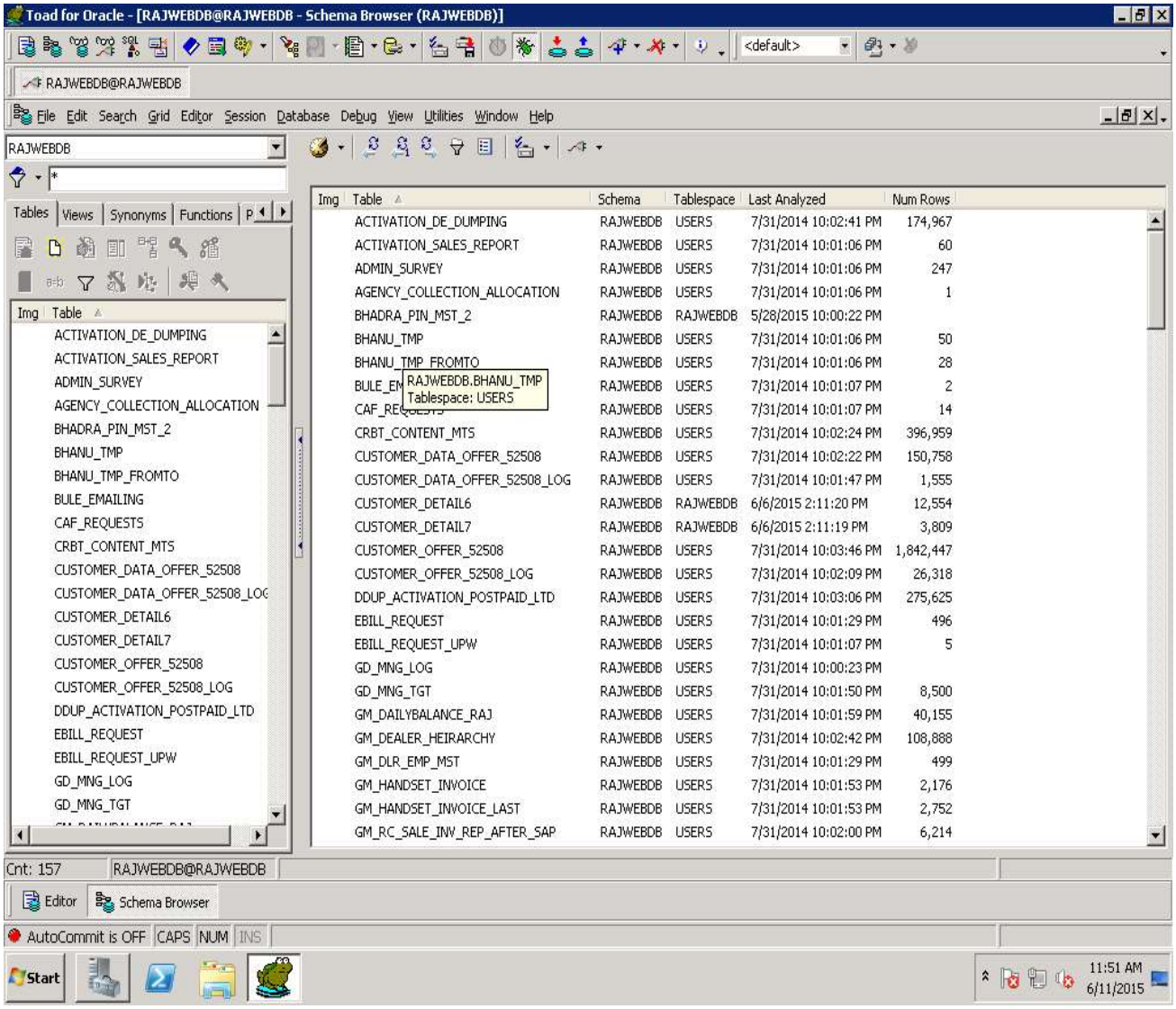
The next screenshot displays the editor window, where we write and run the SQL statements, after establishing the connection.



We first create a table using SQL commands. Example- Table Customer_Detail7-

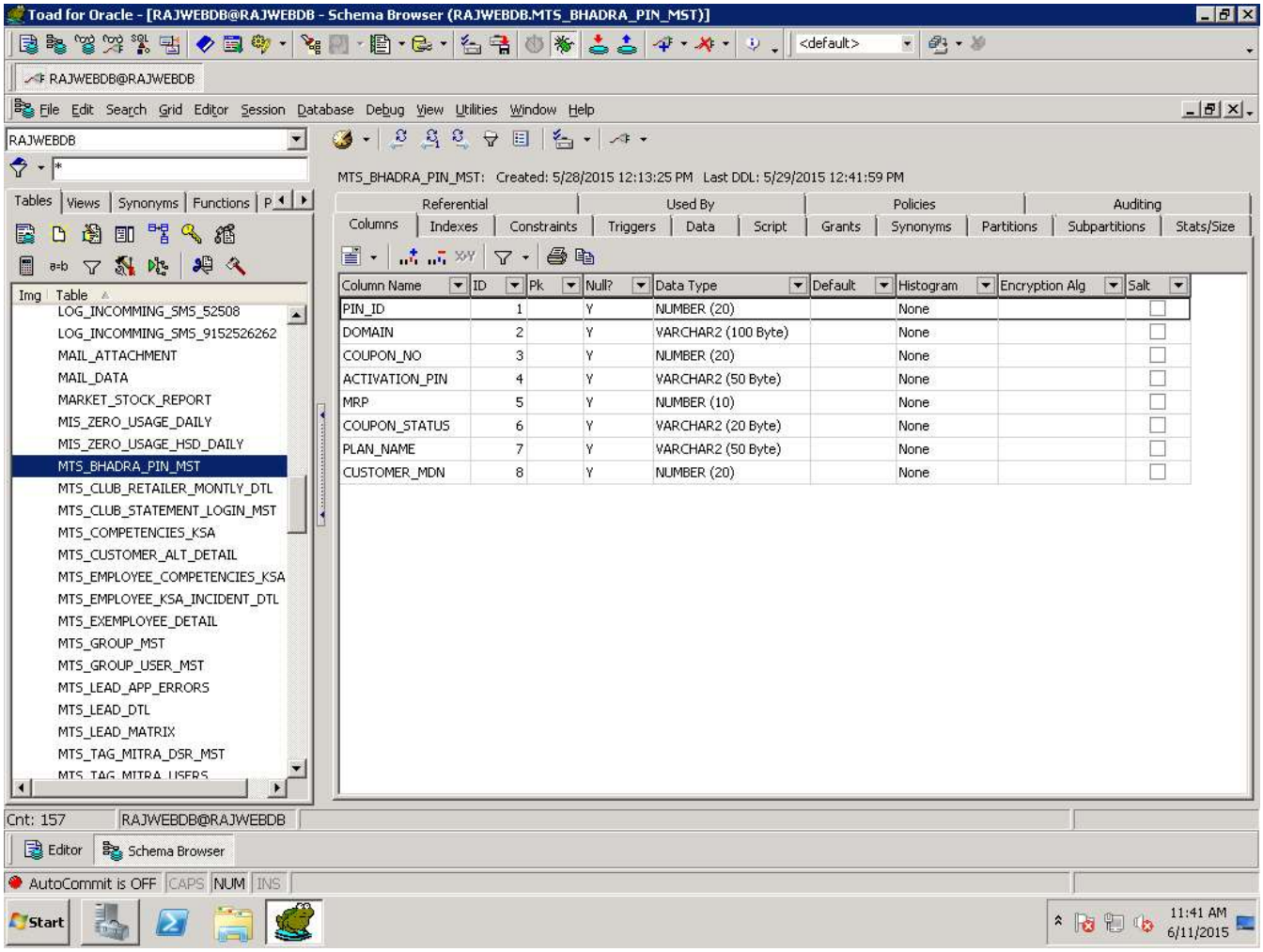


We require many tables, related to each other and which need to be integrated or collectively used, together, they make a schema. This is shown as follows with all tables and views in a list.



The schema shows all tables and their framework, i.e., column names and respective data types.

Example of table MTS_BHADRA_PIN_MST:



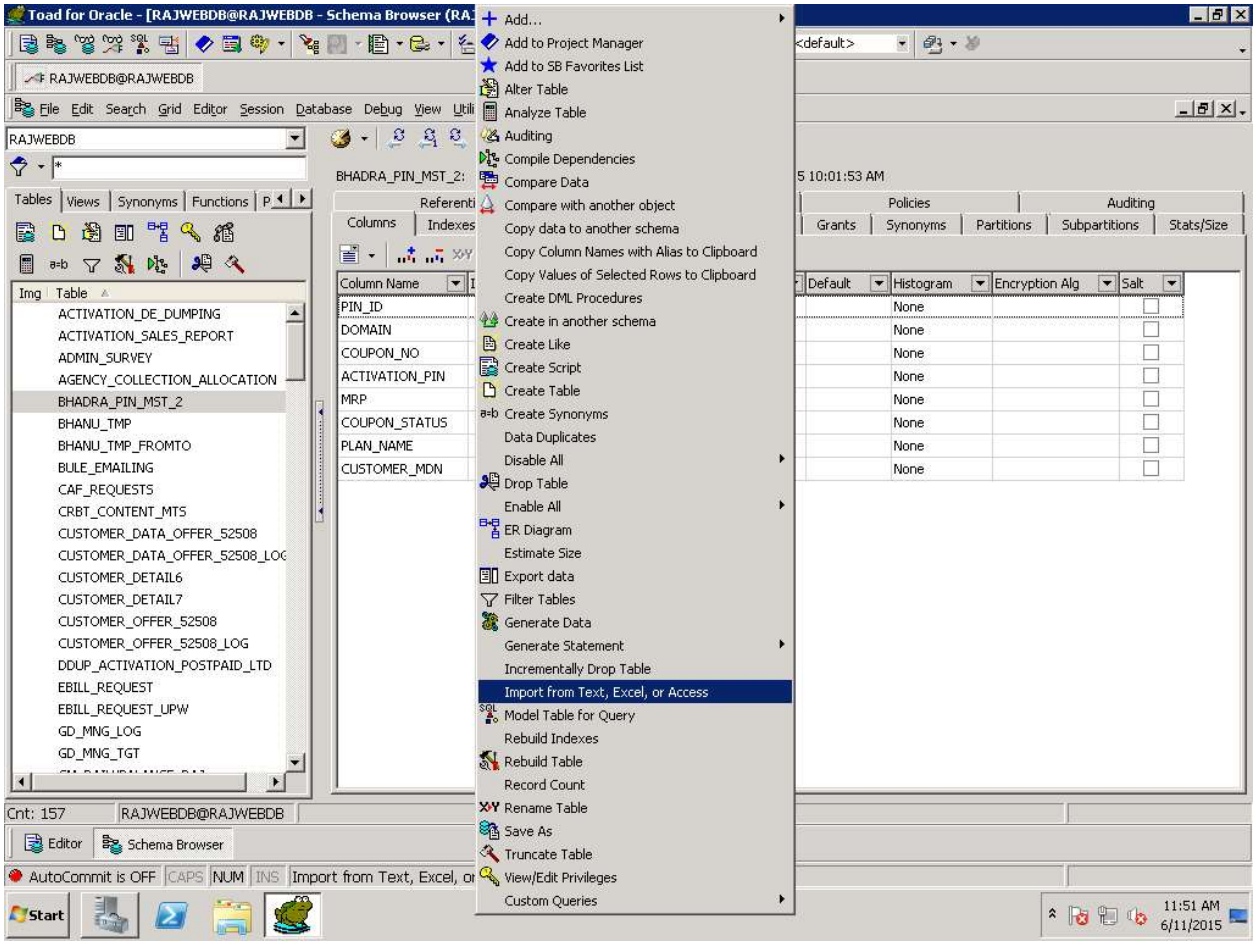
Since the data in tables is very large, manually entering data into a table by writing all values in commands is a tedious and time consuming task. For this, TOAD provides a feature from which we can automatically import data from a given text or excel or any other kind of file, by making suitable adjustments. Here's a look at how that's done.

Following is the procedure to import data in a table.

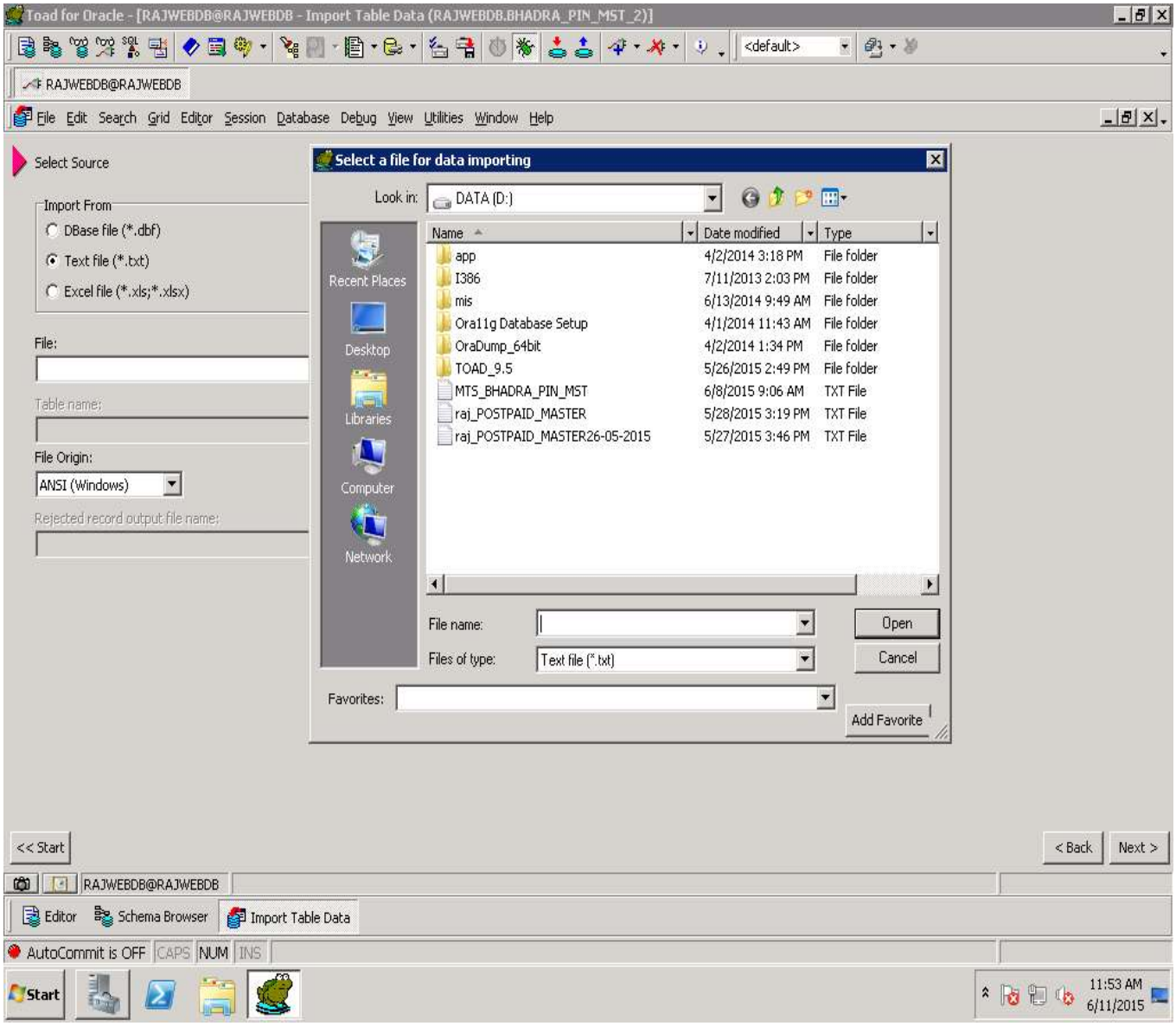
First we create the table, writing column names with their data types.

After we've created the table, we insert values in it.

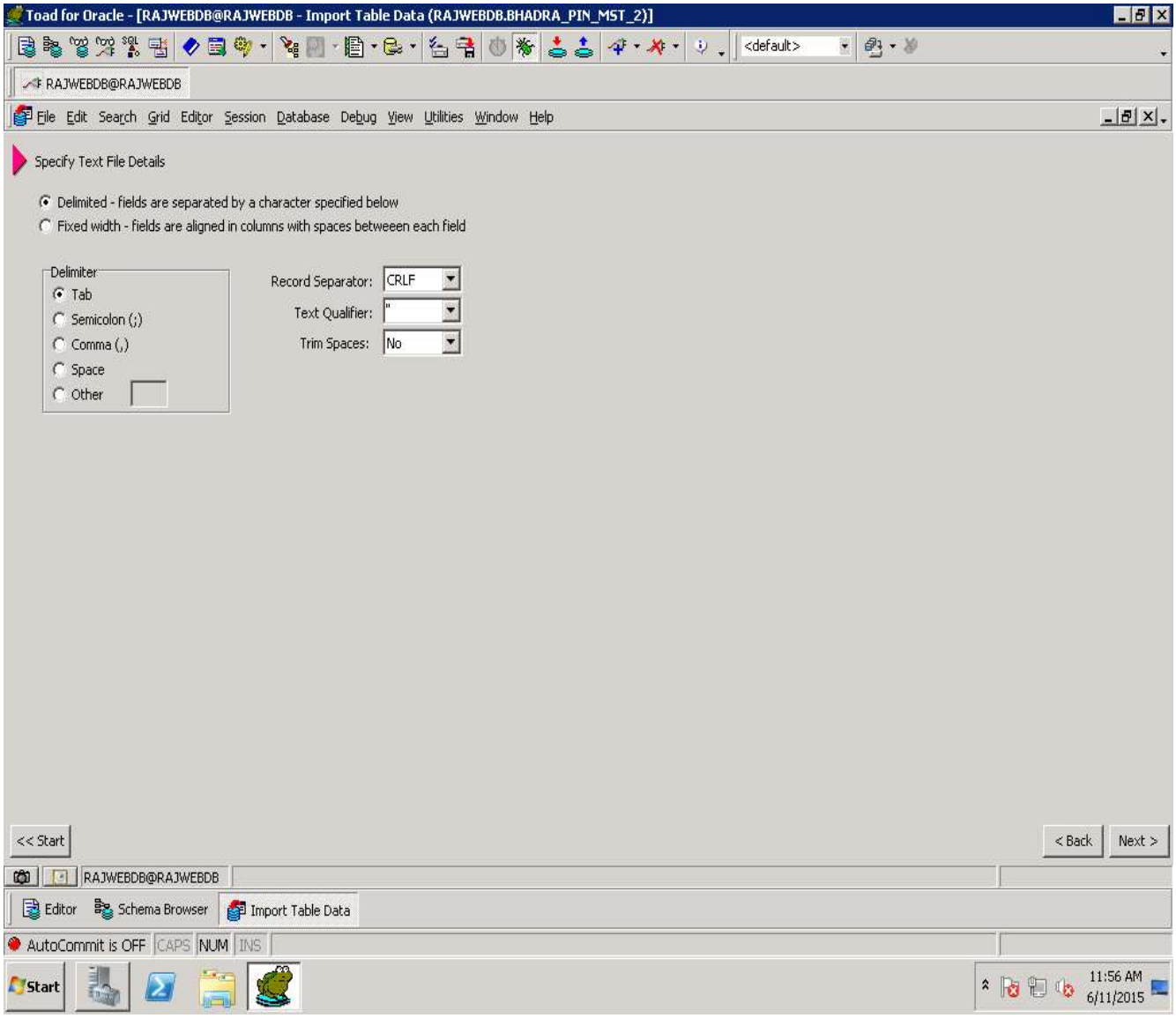
Right click and select import data.



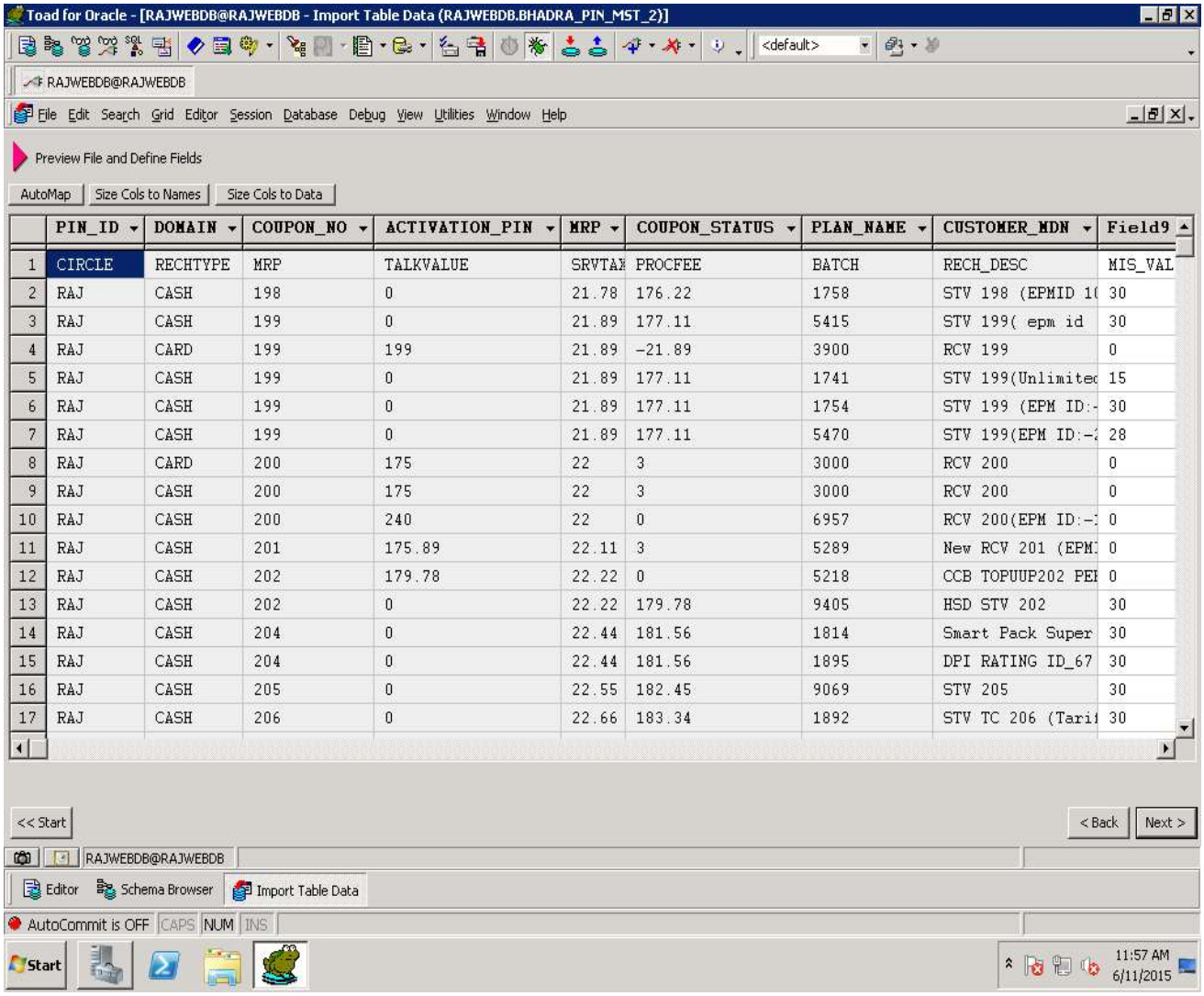
Select the file type from the radio buttons and browse for file name.



Select the delimiter on which to separate columns.

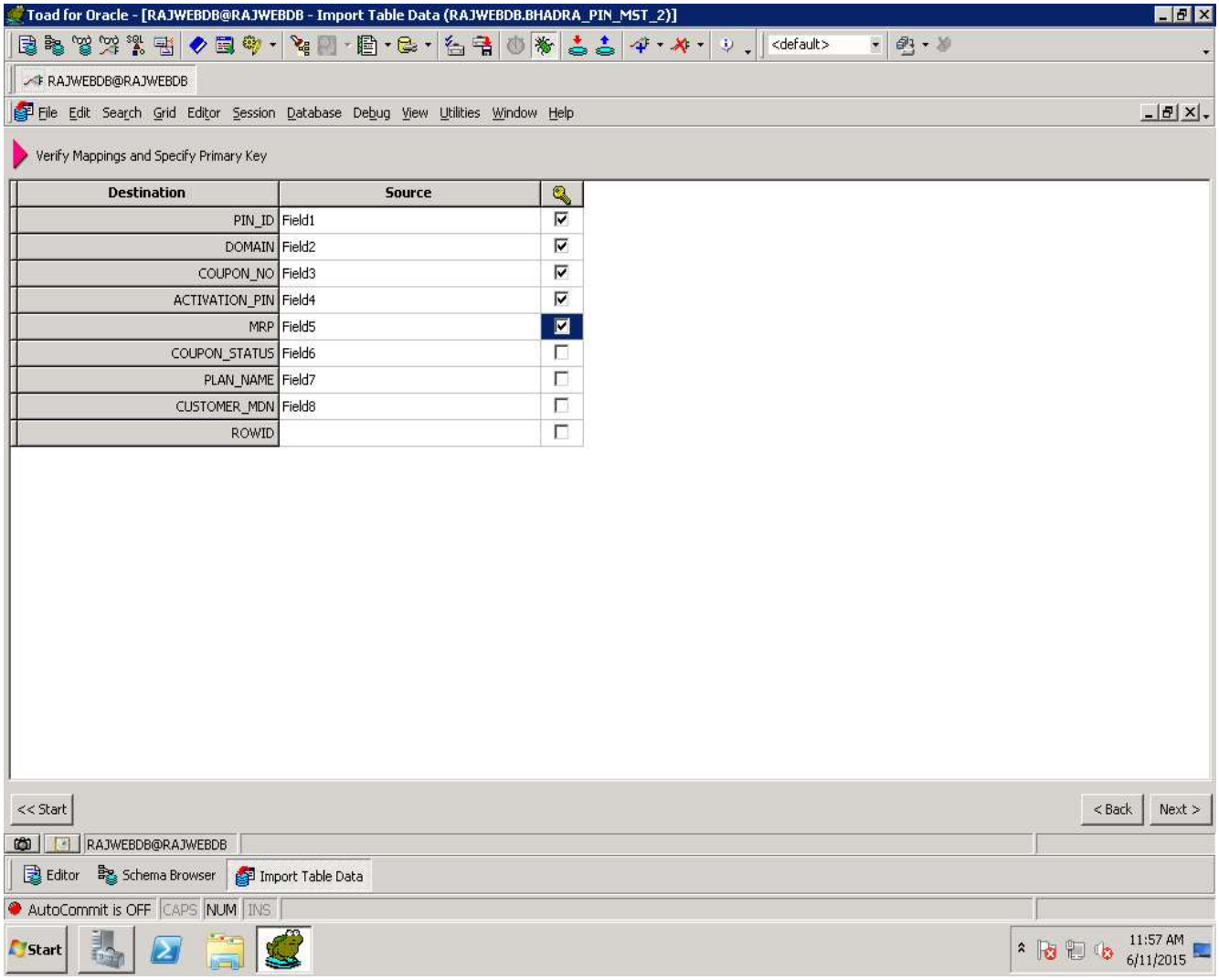


A preview is shown as follows:

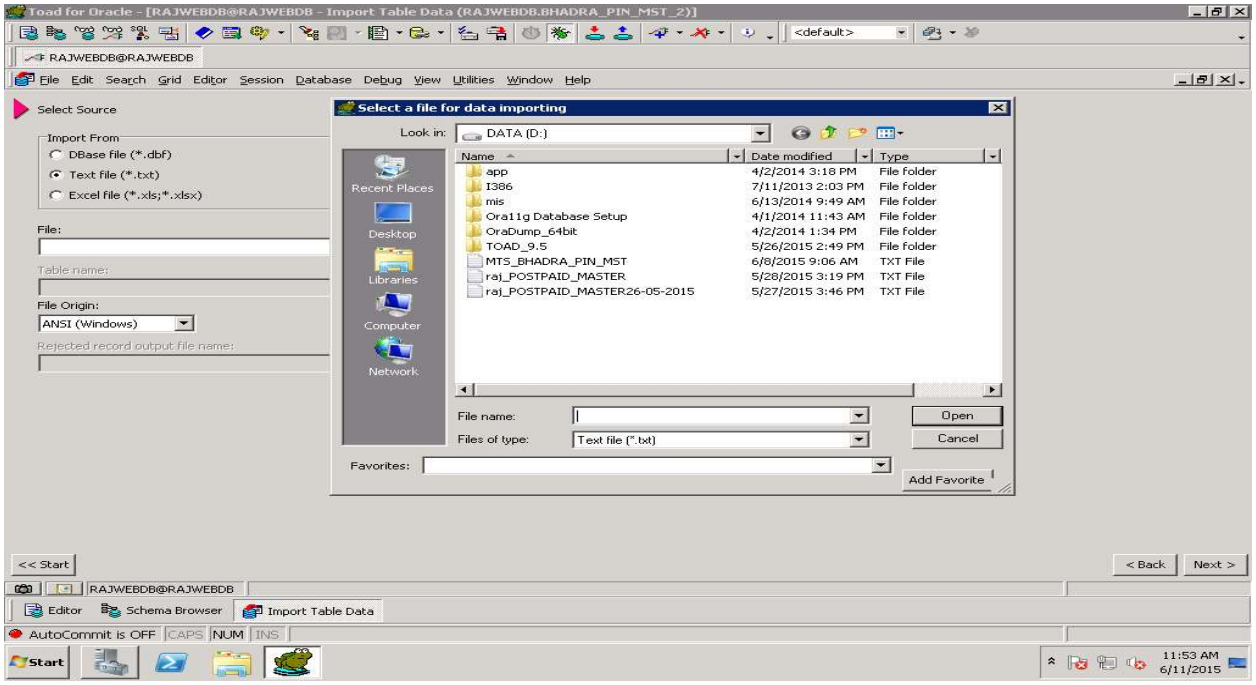


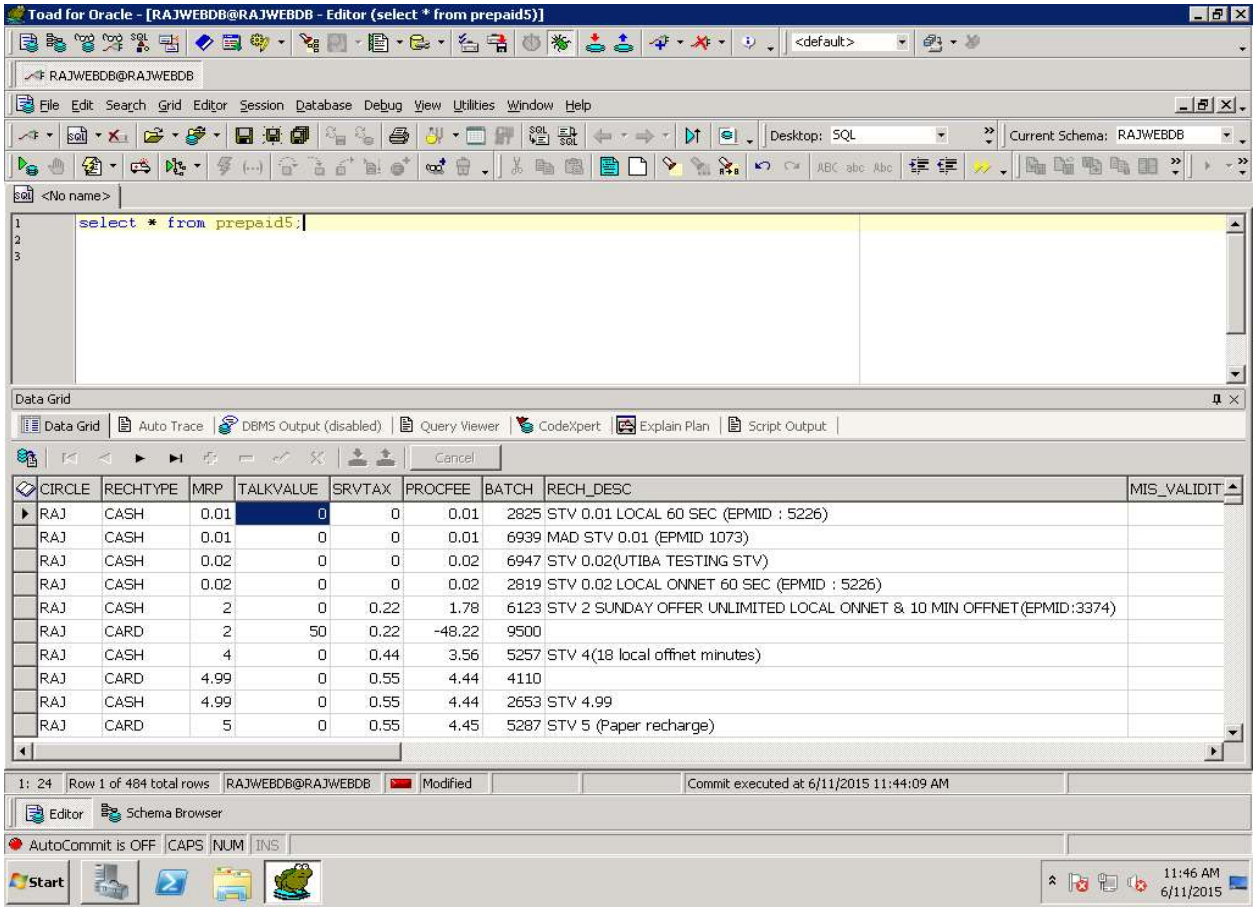
	PIN_ID	DOMAIN	COUPON_NO	ACTIVATION_PIN	MRP	COUPON_STATUS	PLAN_NAME	CUSTOMER_MDN	Field9
1	CIRCLE	RECHTYPE	MRP	TALKVALUE	SRVTAX	PROCFEE	BATCH	RECH_DESC	MIS_VAL
2	RAJ	CASH	198	0	21.78	176.22	1758	STV 198 (EPMID 1(	30
3	RAJ	CASH	199	0	21.89	177.11	5415	STV 199(epm id	30
4	RAJ	CARD	199	199	21.89	-21.89	3900	RCV 199	0
5	RAJ	CASH	199	0	21.89	177.11	1741	STV 199(Unlimitec	15
6	RAJ	CASH	199	0	21.89	177.11	1754	STV 199 (EPM ID:-	30
7	RAJ	CASH	199	0	21.89	177.11	5470	STV 199(EPM ID:-2	28
8	RAJ	CARD	200	175	22	3	3000	RCV 200	0
9	RAJ	CASH	200	175	22	3	3000	RCV 200	0
10	RAJ	CASH	200	240	22	0	6957	RCV 200(EPM ID:-1	0
11	RAJ	CASH	201	175.89	22.11	3	5289	New RCV 201 (EPM:	0
12	RAJ	CASH	202	179.78	22.22	0	5218	CCB TOPUUP202 PEI	0
13	RAJ	CASH	202	0	22.22	179.78	9405	HSD STV 202	30
14	RAJ	CASH	204	0	22.44	181.56	1814	Smart Pack Super	30
15	RAJ	CASH	204	0	22.44	181.56	1895	DPI RATING ID_67	30
16	RAJ	CASH	205	0	22.55	182.45	9069	STV 205	30
17	RAJ	CASH	206	0	22.66	183.34	1892	STV TC 206 (Tari	30

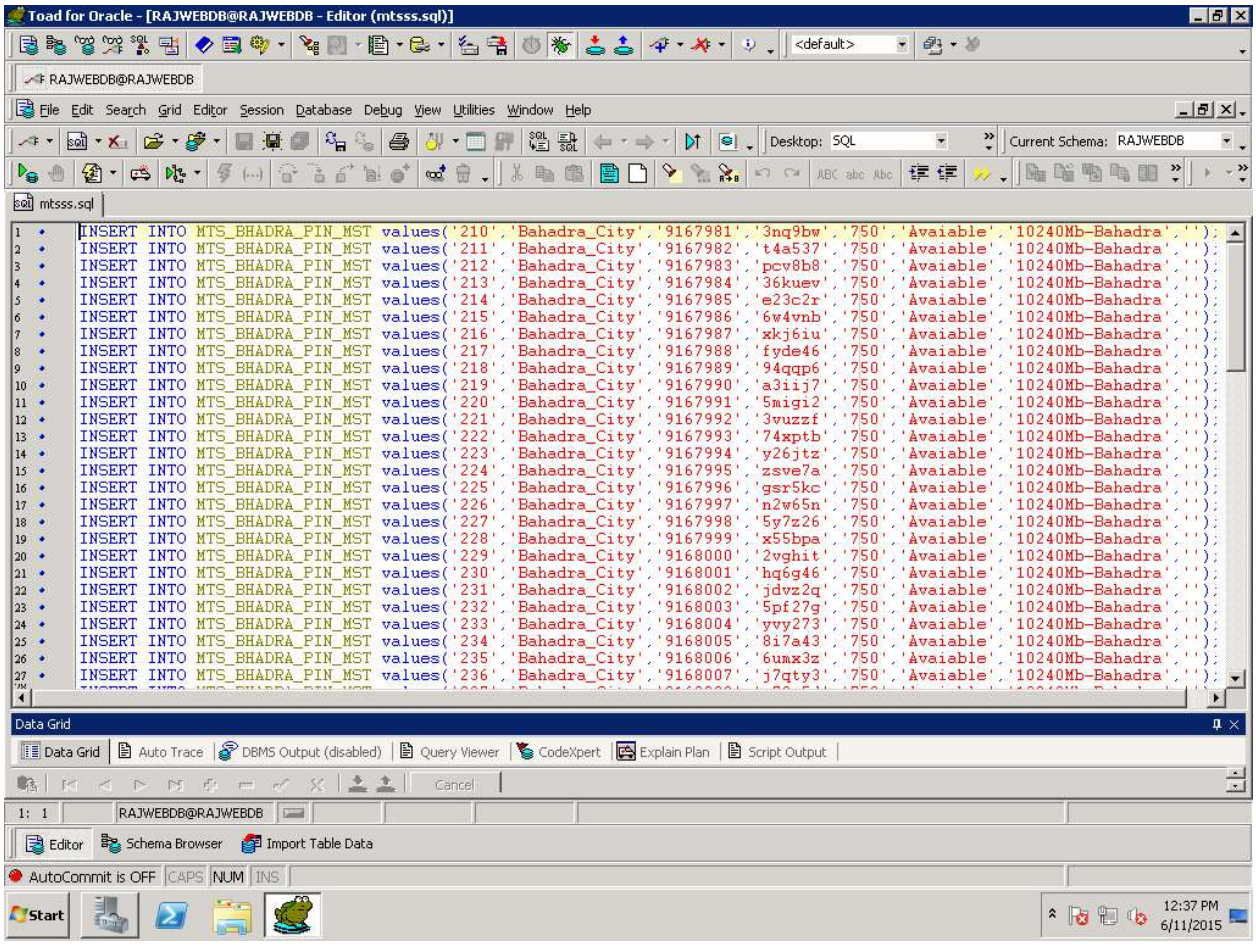
Tick the fields you want as per the columns in table.

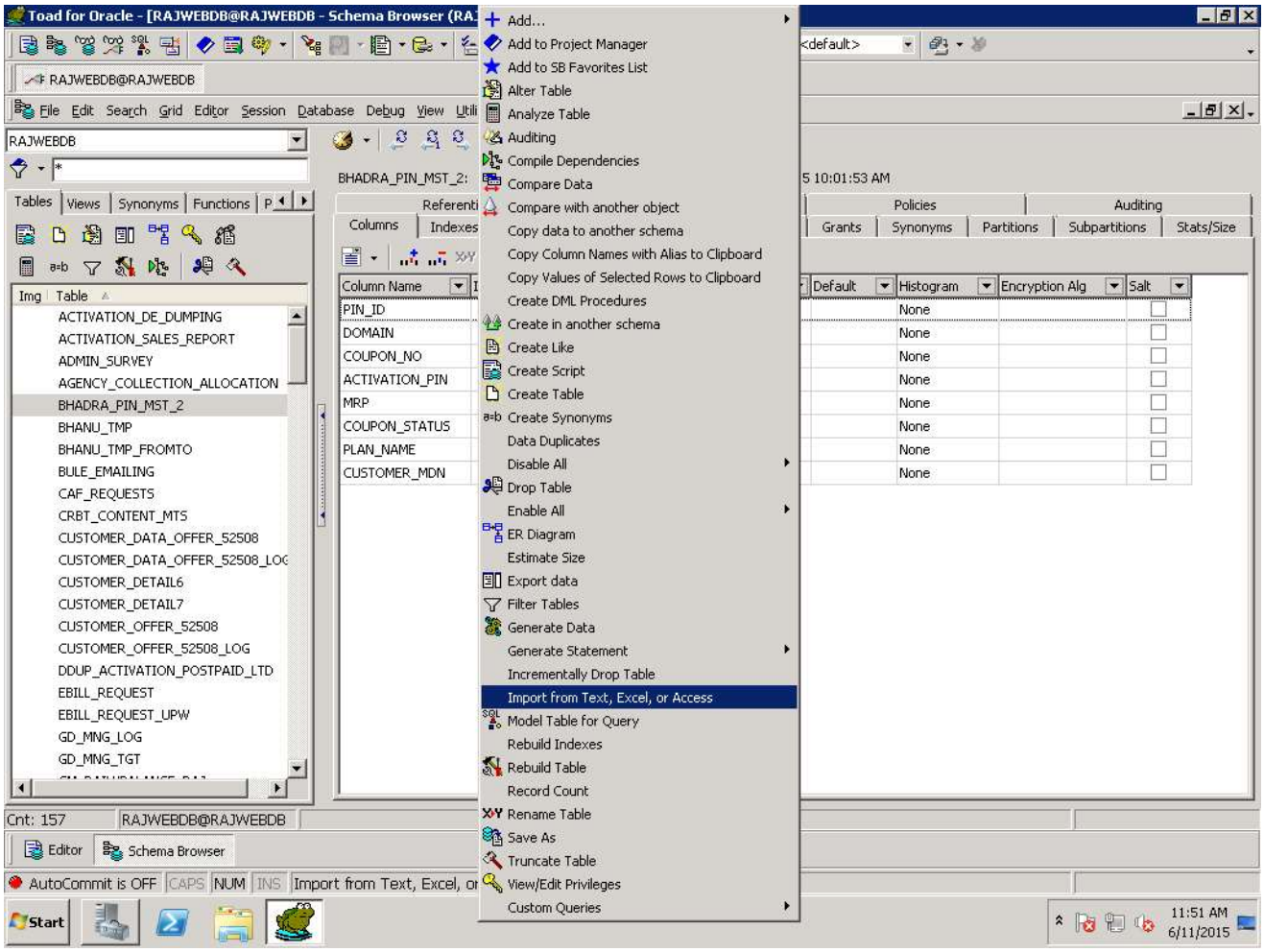


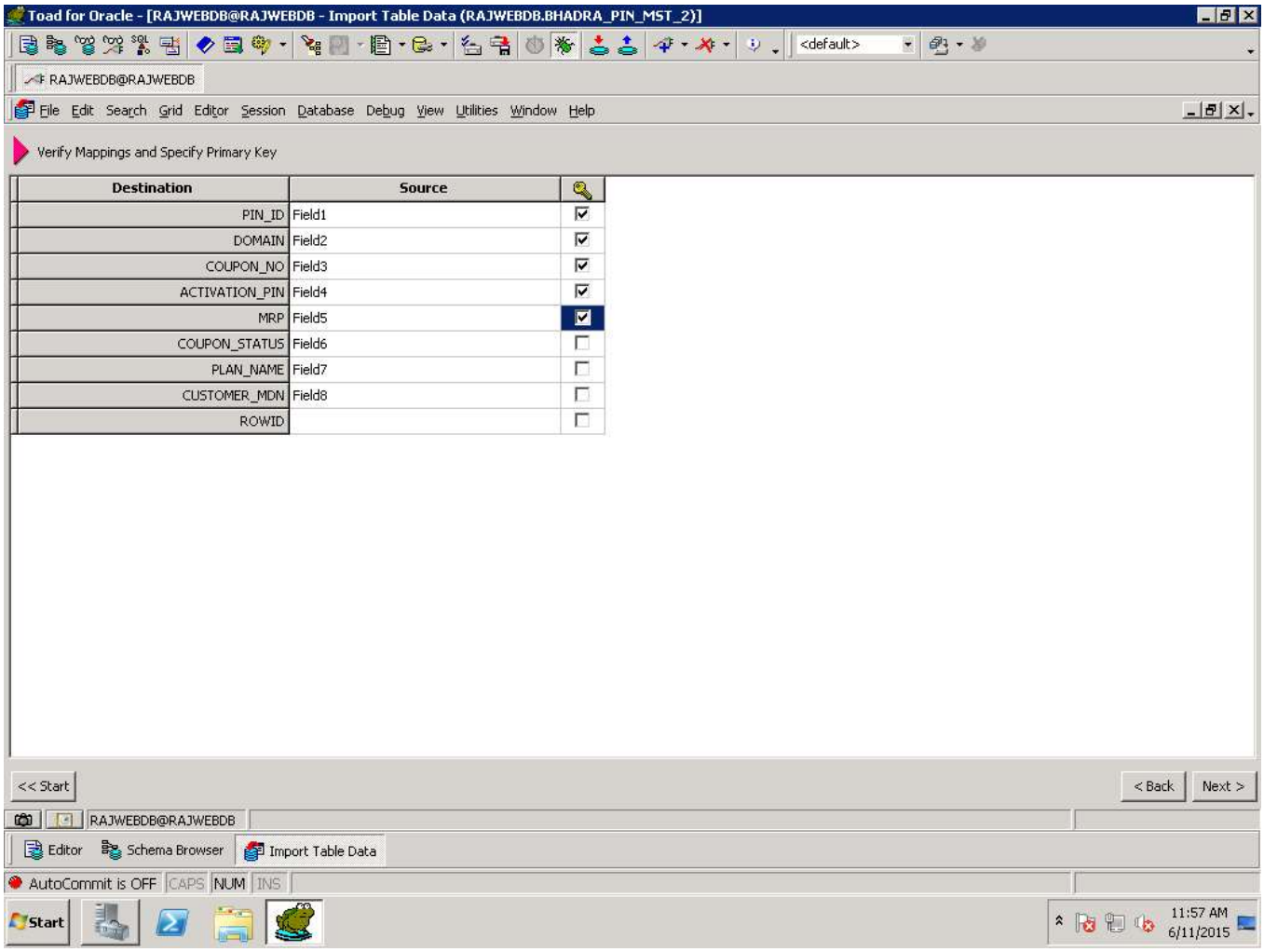
An example-**displaying data after inserting values.**

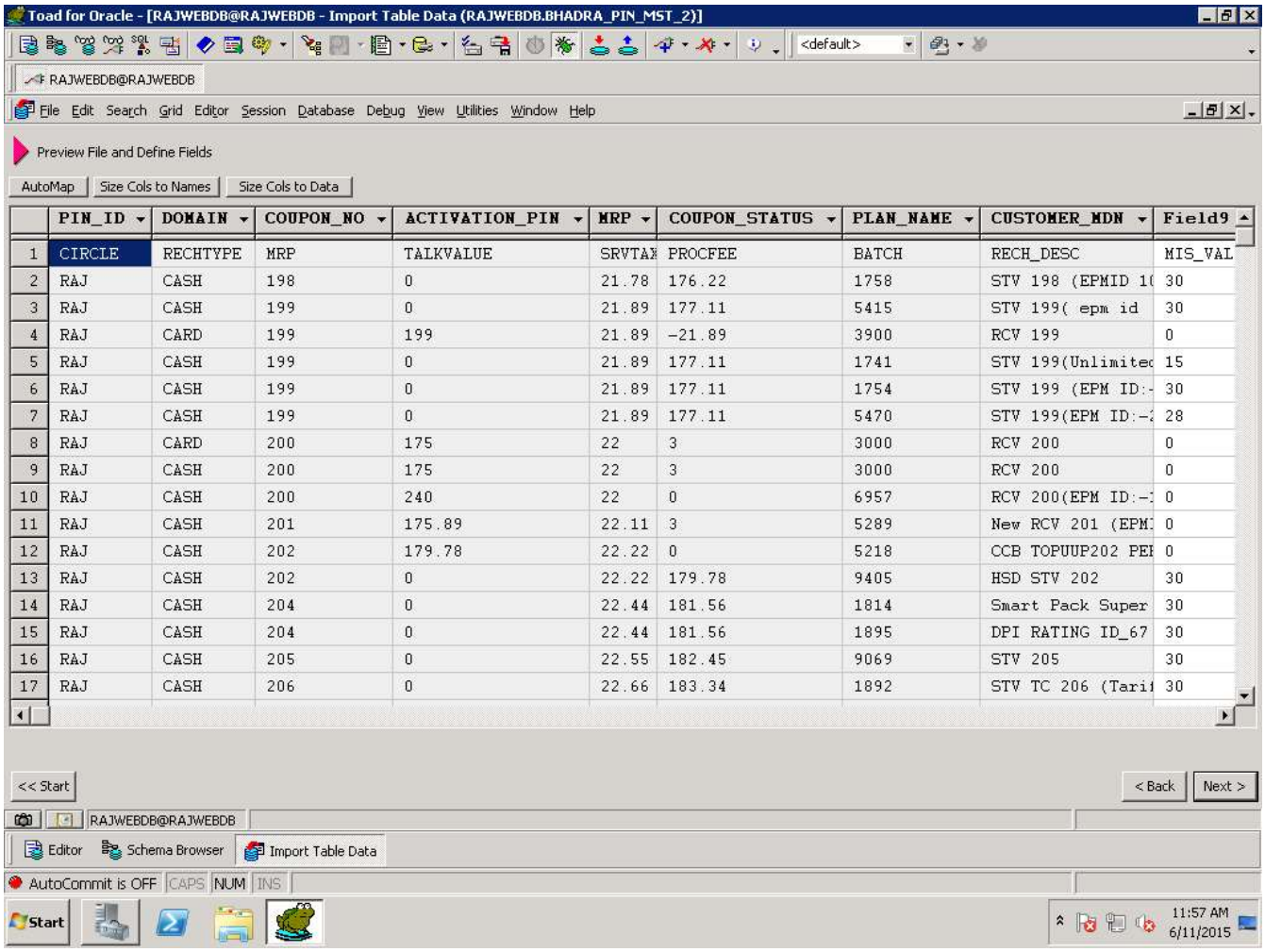





Toad for Oracle - [RAJWEBDB@RAJWEBDB - Editor (mtsss.sql)]
RAJWEBDB@RAJWEBDB
File Edit Search Grid Editor Session Database Debug View Utilities Window Help
Desktop: SQL
Current Schema: RAJWEBDB
mtsss.sql
INSERT INTO MTS_BHADRA_PIN_MST values('210','Bahadra_City','9167981','3nq9bw','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('211','Bahadra_City','9167982','t4a537','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('212','Bahadra_City','9167983','pcv8b8','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('213','Bahadra_City','9167984','36kuev','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('214','Bahadra_City','9167985','e23c2r','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('215','Bahadra_City','9167986','6w4vnb','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('216','Bahadra_City','9167987','xkj6iu','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('217','Bahadra_City','9167988','fyde46','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('218','Bahadra_City','9167989','94qqp6','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('219','Bahadra_City','9167990','a3iij7','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('220','Bahadra_City','9167991','5migi2','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('221','Bahadra_City','9167992','3vuzzf','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('222','Bahadra_City','9167993','74xptb','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('223','Bahadra_City','9167994','y26jtz','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('224','Bahadra_City','9167995','zsve7a','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('225','Bahadra_City','9167996','gsr5kc','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('226','Bahadra_City','9167997','n2w65n','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('227','Bahadra_City','9167998','5y7z26','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('228','Bahadra_City','9167999','x55bpa','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('229','Bahadra_City','9168000','2vghit','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('230','Bahadra_City','9168001','hq6g46','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('231','Bahadra_City','9168002','jdvz2q','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('232','Bahadra_City','9168003','5pf27g','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('233','Bahadra_City','9168004','yvy273','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('234','Bahadra_City','9168005','8i7a43','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('235','Bahadra_City','9168006','6umx3z','750','Avaiable','10240Mb-Bahadra','');
INSERT INTO MTS_BHADRA_PIN_MST values('236','Bahadra_City','9168007','j7qty3','750','Avaiable','10240Mb-Bahadra','');
Data Grid
Data Grid | Auto Trace | DBMS Output (disabled) | Query Viewer | CodeXpert | Explain Plan | Script Output
Cancel
1: 1
RAJWEBDB@RAJWEBDB
Editor | Schema Browser | Import Table Data
AutoCommit is OFF
CAPS
NUM
INS
Start
12:37 PM
6/11/2015

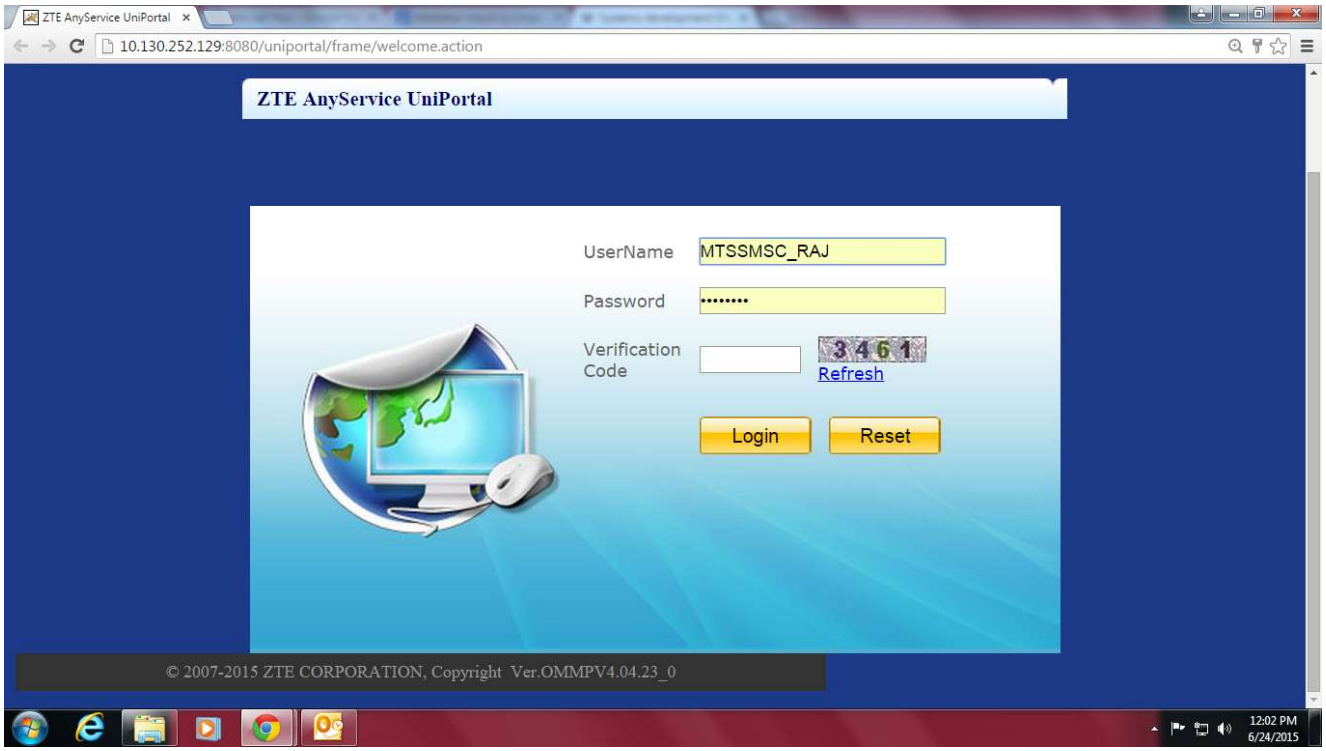

Toad for Oracle - [RAJWEBDB@RAJWEBDB - Schema Browser (RA
RAJWEBDB@RAJWEBDB
File Edit Search Grid Editor Session Database Debug View Utili
RAJWEBDB
Tables Views Synonyms Functions P
Img Table
ACTIVATION_DE_DUMPING
ACTIVATION_SALES_REPORT
ADMIN_SURVEY
AGENCY_COLLECTION_ALLOCATION
BHADRA_PIN_MST_2
BHANU_TMP
BHANU_TMP_FROMTO
BULE_EMAILING
CAF_REQUESTS
CRBT_CONTENT_MTS
CUSTOMER_DATA_OFFER_52508
CUSTOMER_DATA_OFFER_52508_LOG
CUSTOMER_DETAIL6
CUSTOMER_DETAIL7
CUSTOMER_OFFER_52508
CUSTOMER_OFFER_52508_LOG
DDUP_ACTIVATION_POSTPAID_LTD
EBILL_REQUEST
EBILL_REQUEST_UPW
GD_MNG_LOG
GD_MNG_TGT
BHADRA_PIN_MST_2:
Columns Indexes
Column Name
PIN_ID
DOMAIN
COUPON_NO
ACTIVATION_PIN
MRP
COUPON_STATUS
PLAN_NAME
CUSTOMER_MDN
Add...
Add to Project Manager
Add to SB Favorites List
Alter Table
Analyze Table
Auditing
Compile Dependencies
Compare Data
Compare with another object
Copy data to another schema
Copy Column Names with Alias to Clipboard
Copy Values of Selected Rows to Clipboard
Create DML Procedures
Create in another schema
Create Like
Create Script
Create Table
Create Synonyms
Data Duplicates
Disable All
Drop Table
Enable All
ER Diagram
Estimate Size
Export data
Filter Tables
Generate Data
Generate Statement
Incrementally Drop Table
Import from Text, Excel, or Access
Model Table for Query
Rebuild Indexes
Rebuild Table
Record Count
Rename Table
Save As
Truncate Table
View/Edit Privileges
Custom Queries
Policies
Auditing
Grants
Synonyms
Partitions
Subpartitions
Stats/Size
Default
Histogram
Encryption Alg
Salt
None
Cnt: 157
RAJWEBDB@RAJWEBDB
Editor
Schema Browser
AutoCommit is OFF
CAPS
NUM
INS
Start
11:51 AM
6/11/2015

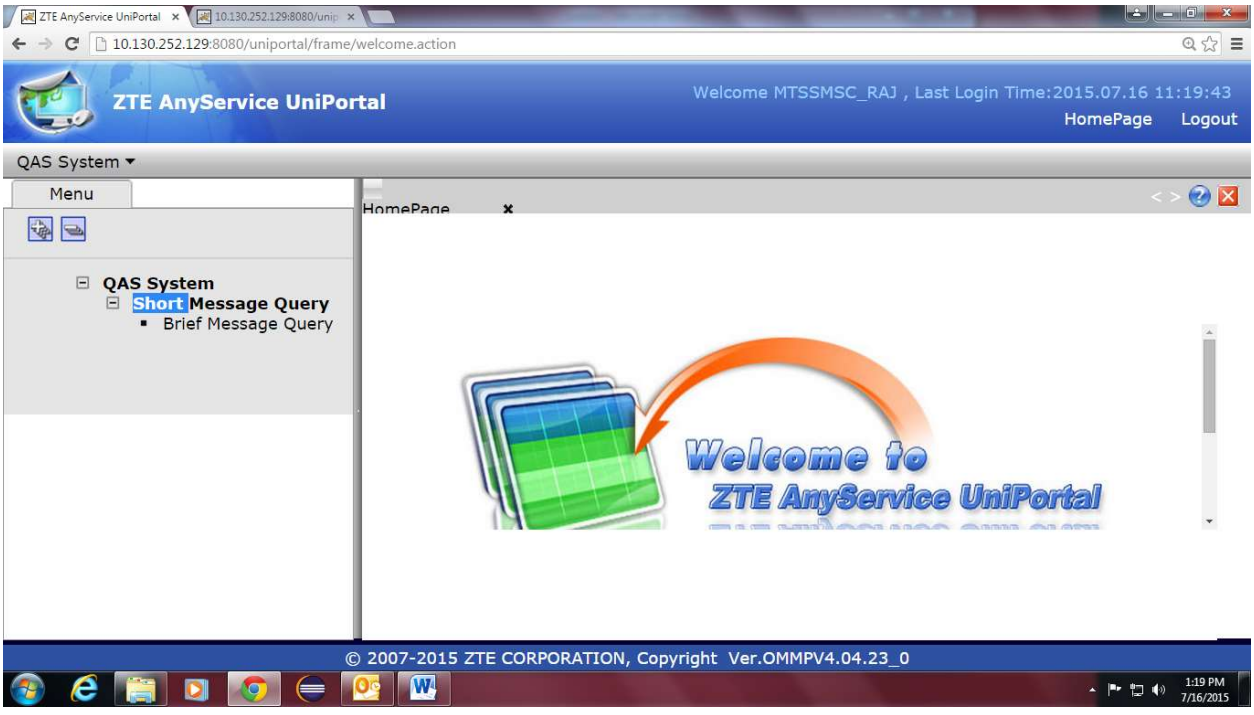

Toad for Oracle - [RAJWEBDB@RAJWEBDB - Import Table Data (RAJWEBDB.BHADRA_PIN_MST_2)]
<default>
RAJWEBDB@RAJWEBDB
File Edit Search Grid Editor Session Database Debug View Utilities Window Help
Verify Mappings and Specify Primary Key
Destination
Source
PIN_ID Field1
DOMAIN Field2
COUPON_NO Field3
ACTIVATION_PIN Field4
MRP Field5
COUPON_STATUS Field6
PLAN_NAME Field7
CUSTOMER_MDN Field8
ROWID
<< Start
< Back
Next >
RAJWEBDB@RAJWEBDB
Editor
Schema Browser
Import Table Data
AutoCommit is OFF
CAPS
NUM
INS
Start
11:57 AM
6/11/2015

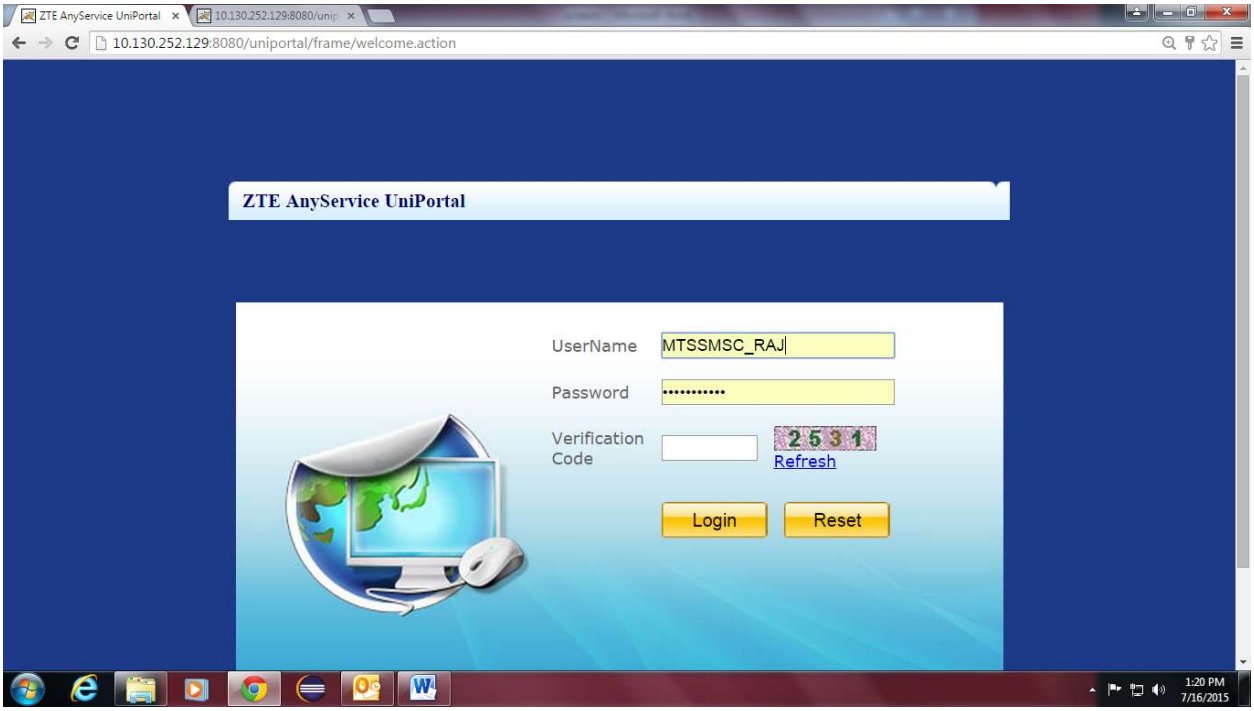


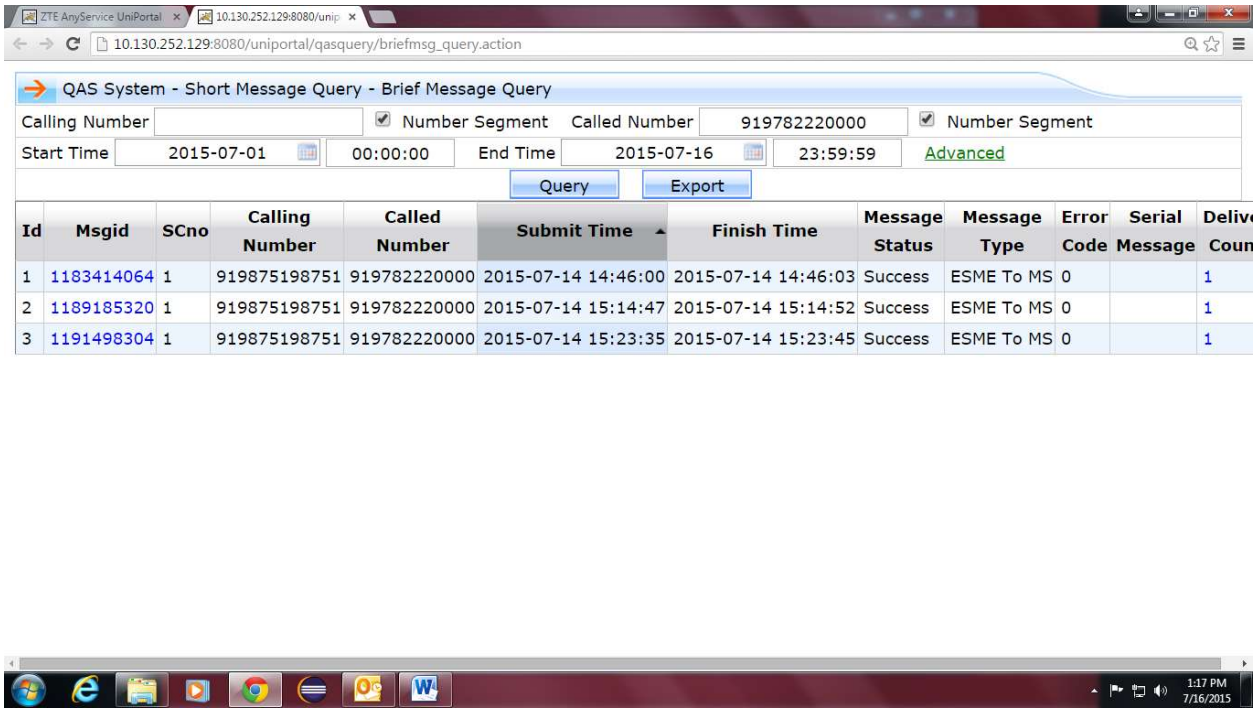
	PIN_ID	DOMAIN	COUPON_NO	ACTIVATION_PIN	MRP	COUPON_STATUS	PLAN_NAME	CUSTOMER_MDN	Field9
1	CIRCLE	RECHTYPE	MRP	TALKVALUE	SRVTAX	PROCFEE	BATCH	RECH_DESC	MIS_VAL
2	RAJ	CASH	198	0	21.78	176.22	1758	STV 198 (EPMID 1(	30
3	RAJ	CASH	199	0	21.89	177.11	5415	STV 199(epm id	30
4	RAJ	CARD	199	199	21.89	-21.89	3900	RCV 199	0
5	RAJ	CASH	199	0	21.89	177.11	1741	STV 199(Unlimitec	15
6	RAJ	CASH	199	0	21.89	177.11	1754	STV 199 (EPM ID:-	30
7	RAJ	CASH	199	0	21.89	177.11	5470	STV 199(EPM ID:-2	28
8	RAJ	CARD	200	175	22	3	3000	RCV 200	0
9	RAJ	CASH	200	175	22	3	3000	RCV 200	0
10	RAJ	CASH	200	240	22	0	6957	RCV 200(EPM ID:-1	0
11	RAJ	CASH	201	175.89	22.11	3	5289	New RCV 201 (EPM1	0
12	RAJ	CASH	202	179.78	22.22	0	5218	CCB TOPUUP202 PEI	0
13	RAJ	CASH	202	0	22.22	179.78	9405	HSD STV 202	30
14	RAJ	CASH	204	0	22.44	181.56	1814	Smart Pack Super	30
15	RAJ	CASH	204	0	22.44	181.56	1895	DPI RATING ID_67	30
16	RAJ	CASH	205	0	22.55	182.45	9069	STV 205	30
17	RAJ	CASH	206	0	22.66	183.34	1892	STV TC 206 (Tari	30

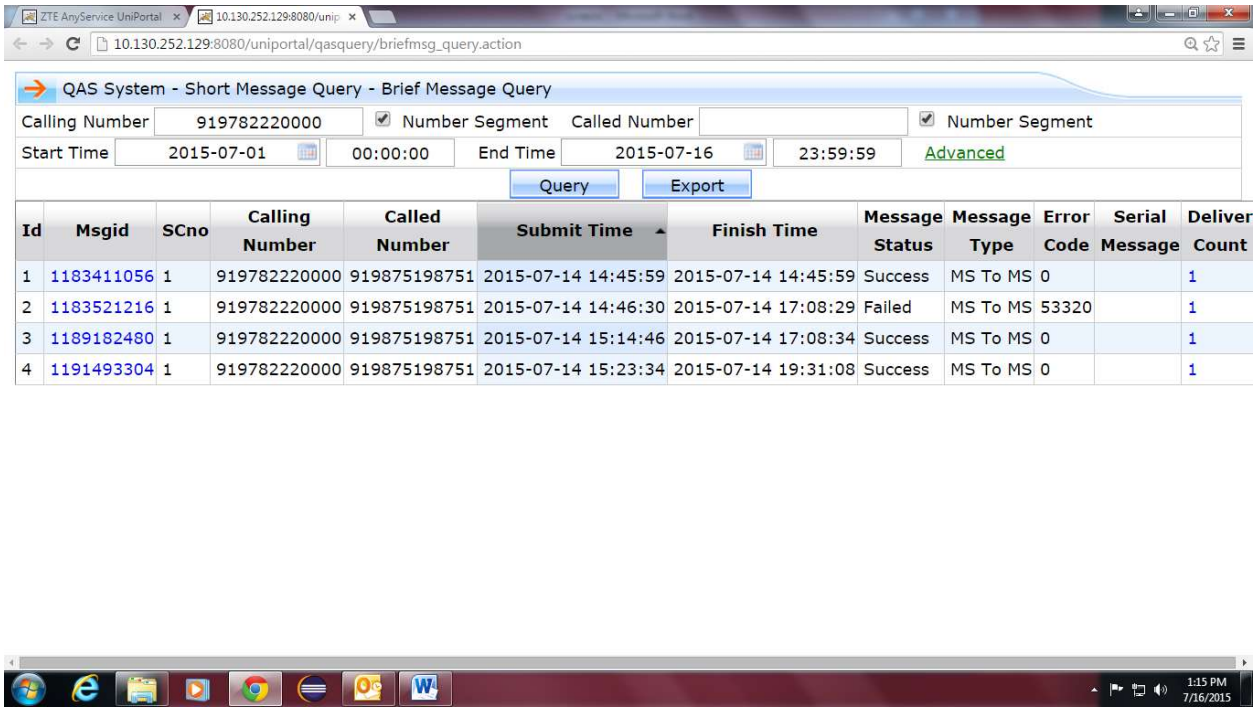
MTS AnyService Portal






ZTE AnyService UniPortal
10.130.252.129:8080/uniportal/frame/welcome.action
ZTE AnyService UniPortal
UserName
MTSSMSC_RAJ
Password
Verification Code
2531
Refresh
Login
Reset
1:20 PM
7/16/2015


10.130.252.129:8080/uniportal/qasquery/briefmsg_query.action
QAS System - Short Message Query - Brief Message Query
Calling Number
Number Segment
Called Number
919782220000
Number Segment
Start Time
2015-07-01
00:00:00
End Time
2015-07-16
23:59:59
Advanced
Query
Export
Id
Msgid
SCno
Calling Number
Called Number
Submit Time
Finish Time
Message Status
Message Type
Error Code
Serial Message
Deliv Coun
1 1183414064 1 919875198751 919782220000 2015-07-14 14:46:00 2015-07-14 14:46:03 Success ESME To MS 0 1
2 1189185320 1 919875198751 919782220000 2015-07-14 15:14:47 2015-07-14 15:14:52 Success ESME To MS 0 1
3 1191498304 1 919875198751 919782220000 2015-07-14 15:23:35 2015-07-14 15:23:45 Success ESME To MS 0 1
1:17 PM
7/16/2015



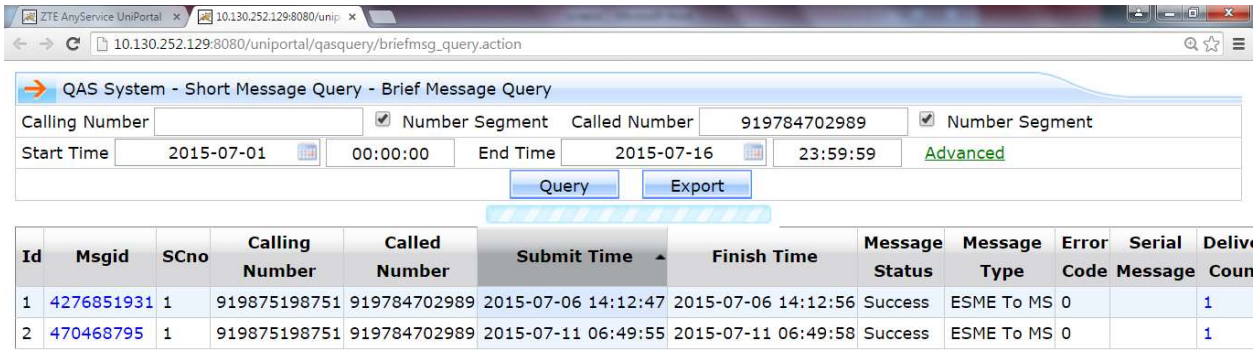
QAS System - Short Message Query - Brief Message Query

Calling Number 919782220000 Number Segment Called Number Number Segment

Start Time 2015-07-01 00:00:00 End Time 2015-07-16 23:59:59 Advanced

Query Export

Id	Msgid	SCno	Calling Number	Called Number	Submit Time	Finish Time	Message Status	Message Type	Error Code	Serial Message	Deliver Count
1	1183411056	1	919782220000	919875198751	2015-07-14 14:45:59	2015-07-14 14:45:59	Success	MS To MS	0		1
2	1183521216	1	919782220000	919875198751	2015-07-14 14:46:30	2015-07-14 17:08:29	Failed	MS To MS	53320		1
3	1189182480	1	919782220000	919875198751	2015-07-14 15:14:46	2015-07-14 17:08:34	Success	MS To MS	0		1
4	1191493304	1	919782220000	919875198751	2015-07-14 15:23:34	2015-07-14 19:31:08	Success	MS To MS	0		1



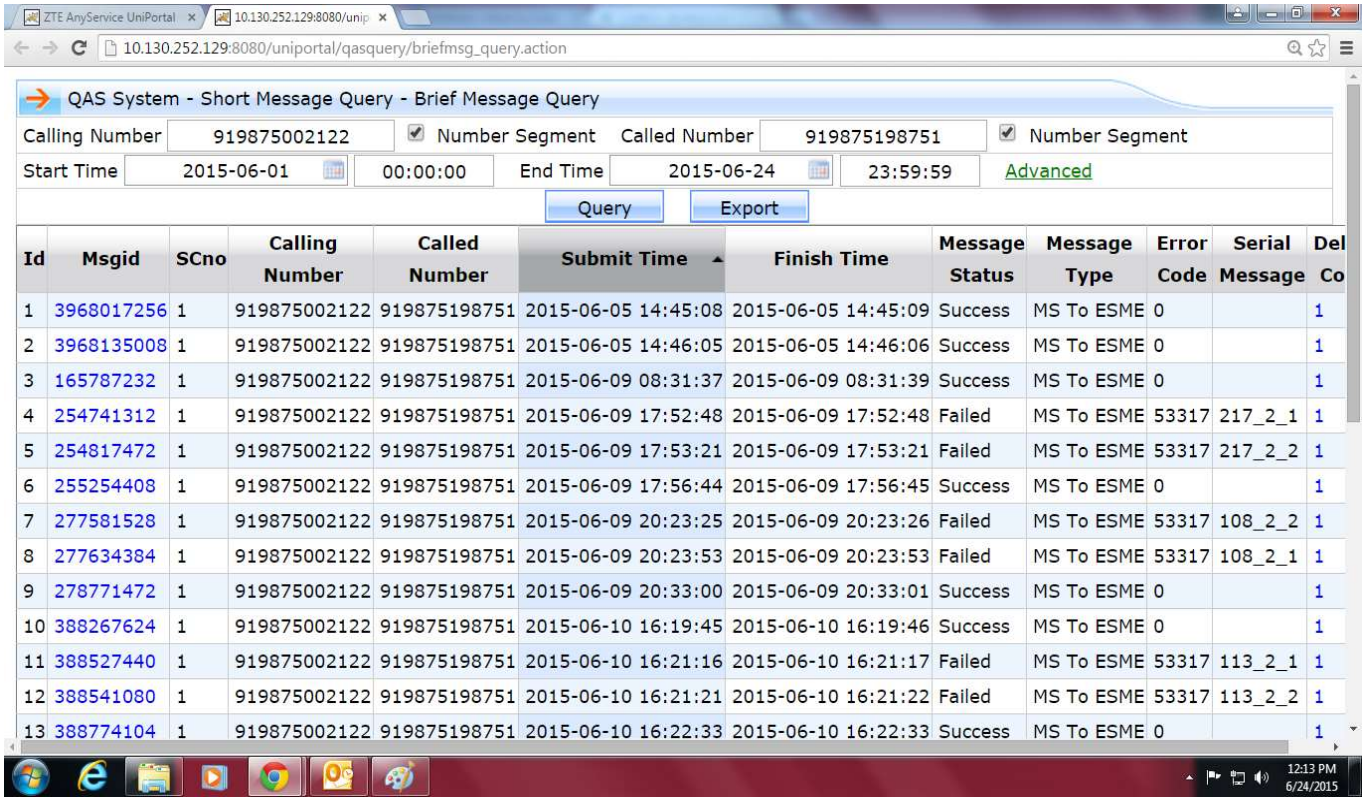
QAS System - Short Message Query - Brief Message Query

Calling Number Number Segment Called Number 919784702989 Number Segment

Start Time 2015-07-01 00:00:00 End Time 2015-07-16 23:59:59 Advanced

Query Export

Id	Msgid	SCno	Calling Number	Called Number	Submit Time	Finish Time	Message Status	Message Type	Error Code	Serial Message	Deliv Coun
1	4276851931	1	919875198751	919784702989	2015-07-06 14:12:47	2015-07-06 14:12:56	Success	ESME To MS	0		1
2	470468795	1	919875198751	919784702989	2015-07-11 06:49:55	2015-07-11 06:49:58	Success	ESME To MS	0		1

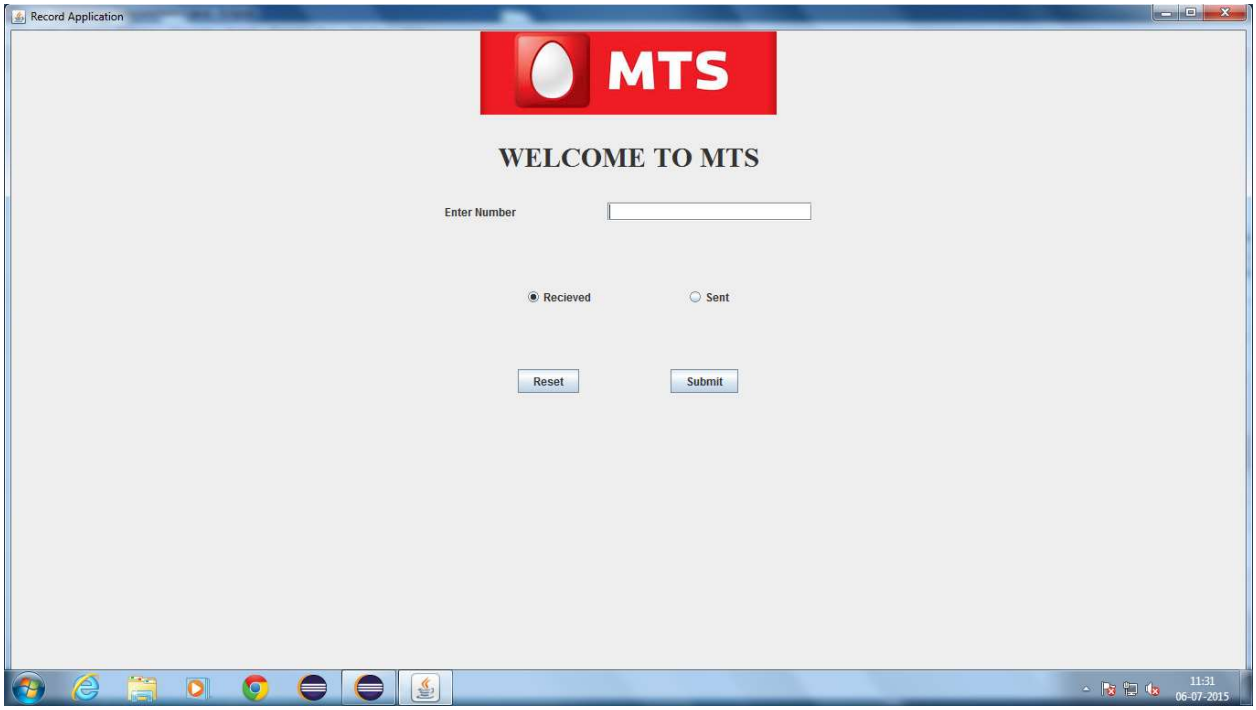

ZTE AnyService UniPortal
10.130.252.129:8080/uniportal/qasquery/briefmsg_query.action
QAS System - Short Message Query - Brief Message Query
Calling Number 919875002122 Number Segment Called Number 919875198751 Number Segment
Start Time 2015-06-01 00:00:00 End Time 2015-06-24 23:59:59 Advanced
Query Export
Id | Msgid | SCno | Calling Number | Called Number | Submit Time | Finish Time | Message Status | Message Type | Error Code | Serial Message | Del Co
1 | 3968017256 | 1 | 919875002122 | 919875198751 | 2015-06-05 14:45:08 | 2015-06-05 14:45:09 | Success | MS To ESME | 0 | | 1
2 | 3968135008 | 1 | 919875002122 | 919875198751 | 2015-06-05 14:46:05 | 2015-06-05 14:46:06 | Success | MS To ESME | 0 | | 1
3 | 165787232 | 1 | 919875002122 | 919875198751 | 2015-06-09 08:31:37 | 2015-06-09 08:31:39 | Success | MS To ESME | 0 | | 1
4 | 254741312 | 1 | 919875002122 | 919875198751 | 2015-06-09 17:52:48 | 2015-06-09 17:52:48 | Failed | MS To ESME | 53317 | 217_2_1 | 1
5 | 254817472 | 1 | 919875002122 | 919875198751 | 2015-06-09 17:53:21 | 2015-06-09 17:53:21 | Failed | MS To ESME | 53317 | 217_2_2 | 1
6 | 255254408 | 1 | 919875002122 | 919875198751 | 2015-06-09 17:56:44 | 2015-06-09 17:56:45 | Success | MS To ESME | 0 | | 1
7 | 277581528 | 1 | 919875002122 | 919875198751 | 2015-06-09 20:23:25 | 2015-06-09 20:23:26 | Failed | MS To ESME | 53317 | 108_2_2 | 1
8 | 277634384 | 1 | 919875002122 | 919875198751 | 2015-06-09 20:23:53 | 2015-06-09 20:23:53 | Failed | MS To ESME | 53317 | 108_2_1 | 1
9 | 278771472 | 1 | 919875002122 | 919875198751 | 2015-06-09 20:33:00 | 2015-06-09 20:33:01 | Success | MS To ESME | 0 | | 1
10 | 388267624 | 1 | 919875002122 | 919875198751 | 2015-06-10 16:19:45 | 2015-06-10 16:19:46 | Success | MS To ESME | 0 | | 1
11 | 388527440 | 1 | 919875002122 | 919875198751 | 2015-06-10 16:21:16 | 2015-06-10 16:21:17 | Failed | MS To ESME | 53317 | 113_2_1 | 1
12 | 388541080 | 1 | 919875002122 | 919875198751 | 2015-06-10 16:21:21 | 2015-06-10 16:21:22 | Failed | MS To ESME | 53317 | 113_2_2 | 1
13 | 388774104 | 1 | 919875002122 | 919875198751 | 2015-06-10 16:22:33 | 2015-06-10 16:22:33 | Success | MS To ESME | 0 | | 1
12:13 PM
6/24/2015

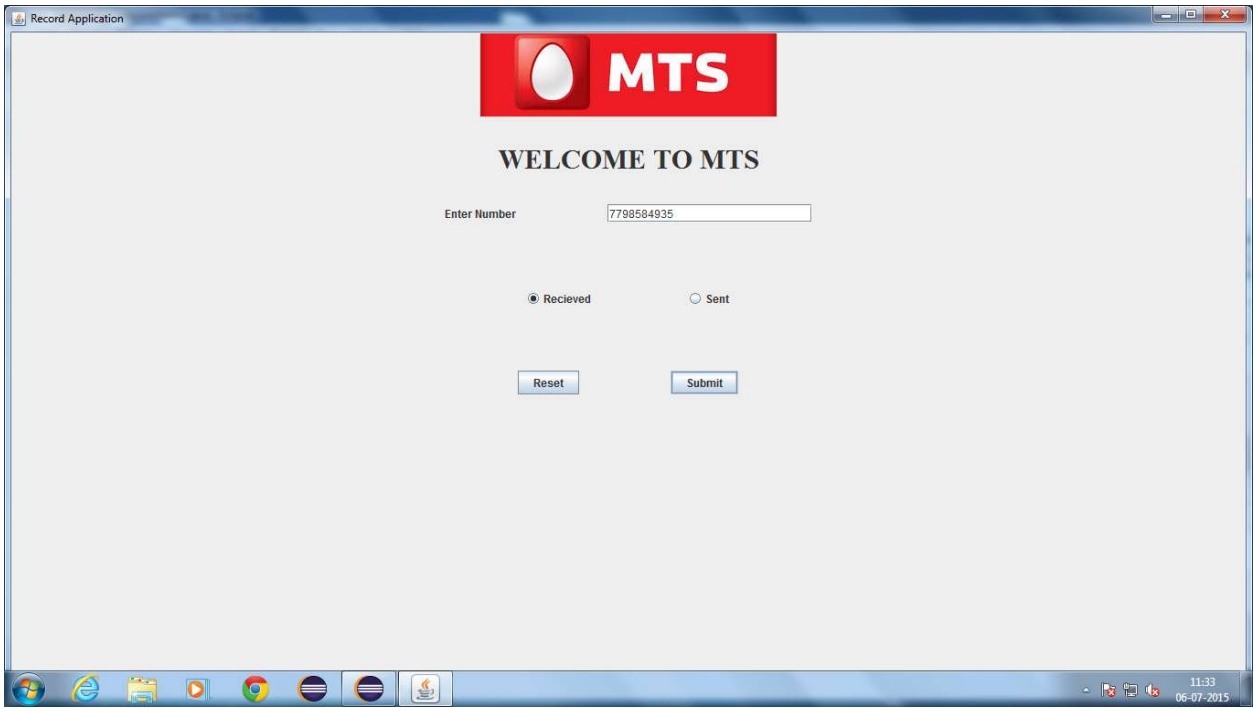
FRONT END

For the front end, we create an application that can for the benefit of the administrator, retrieve SMS information from all transactions in a given period of time. **Java** is a set of several computer software and specifications developed by Sun Microsystems, later acquired by Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. While less common, Java applets run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

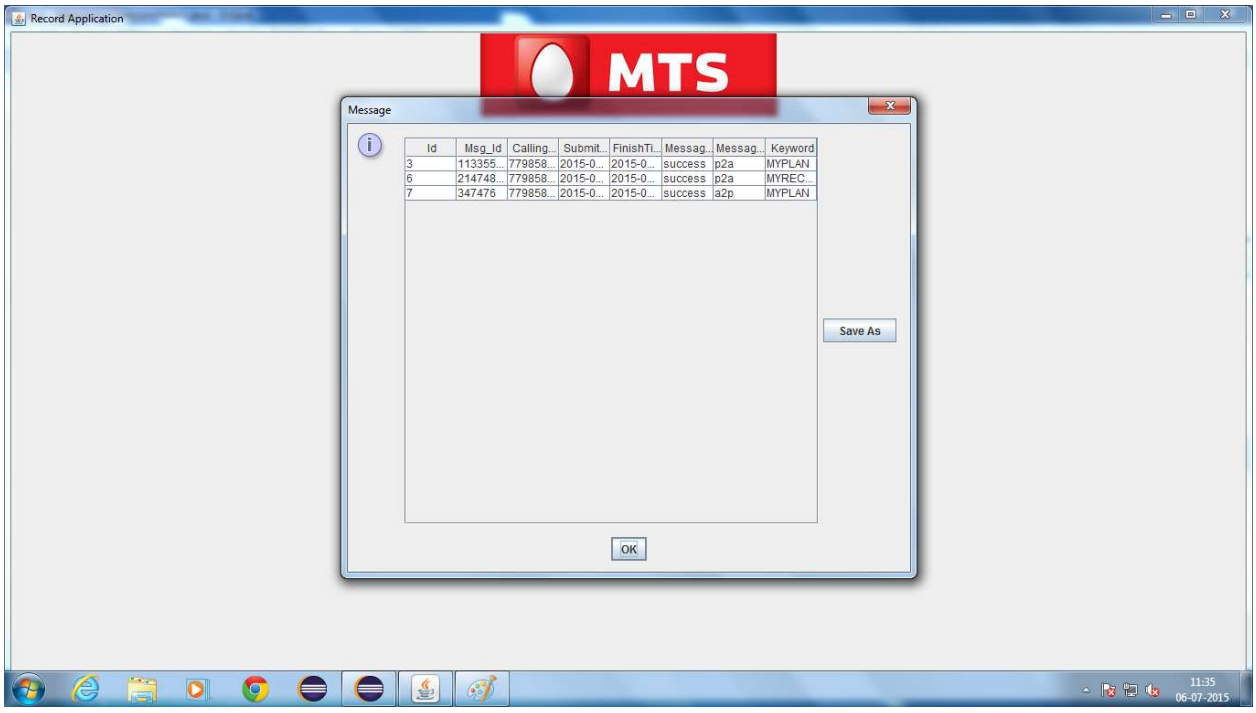
We use Eclipse Luna for creating the same. In computer programming, **Eclipse** is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications.

Here is the screenshot the application. First we enter the number and select which records we need to view; messages sent or received.

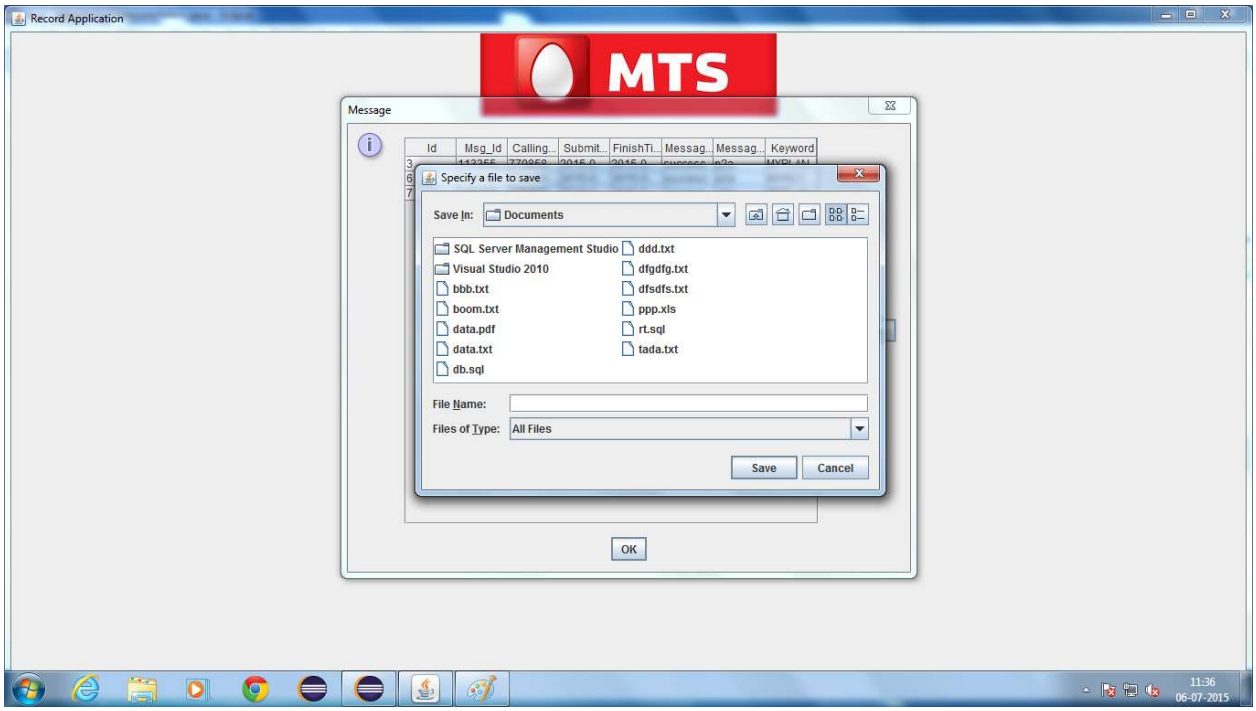




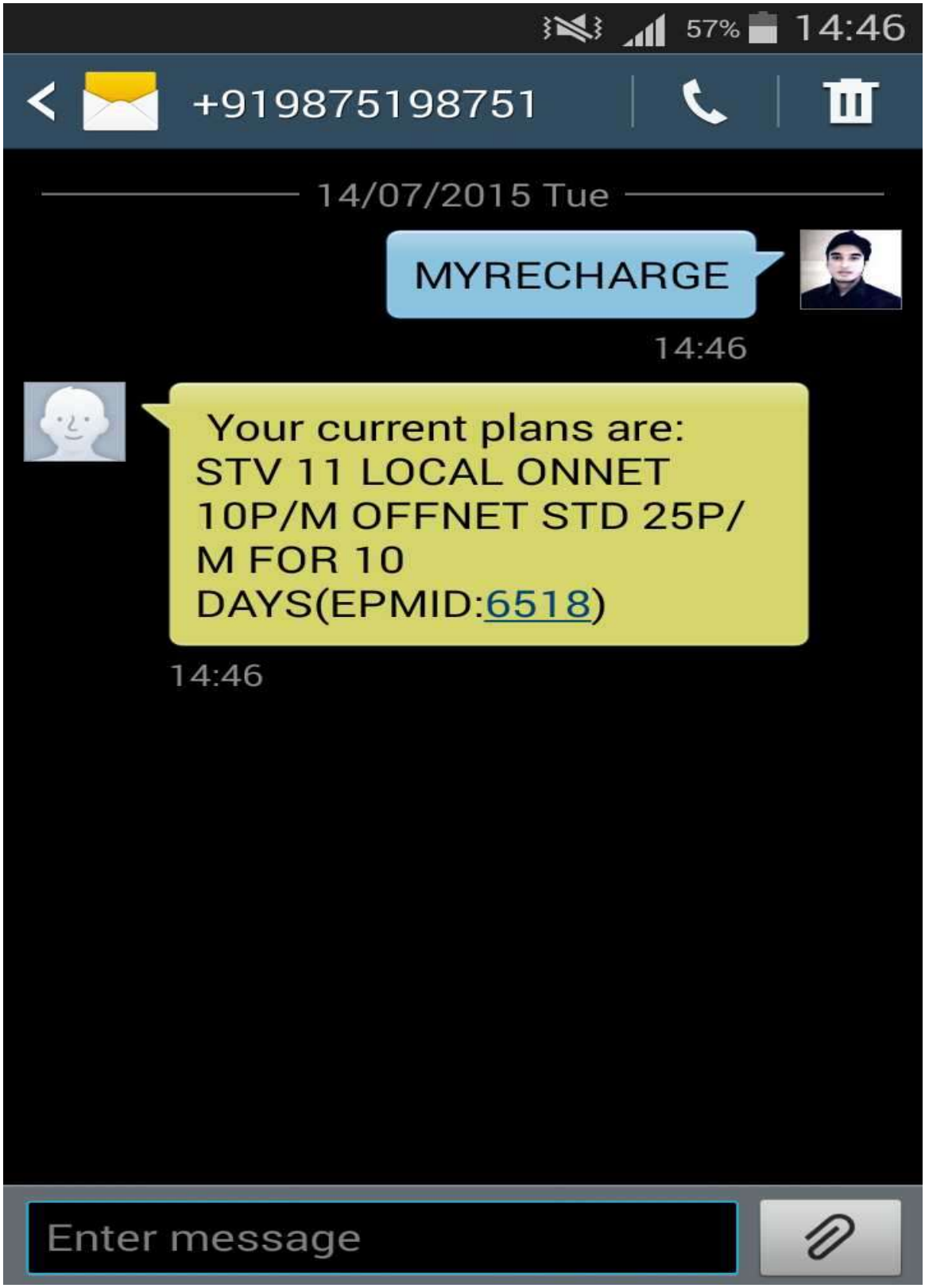
Once we press the submit button, the application retrieves the table from the database.

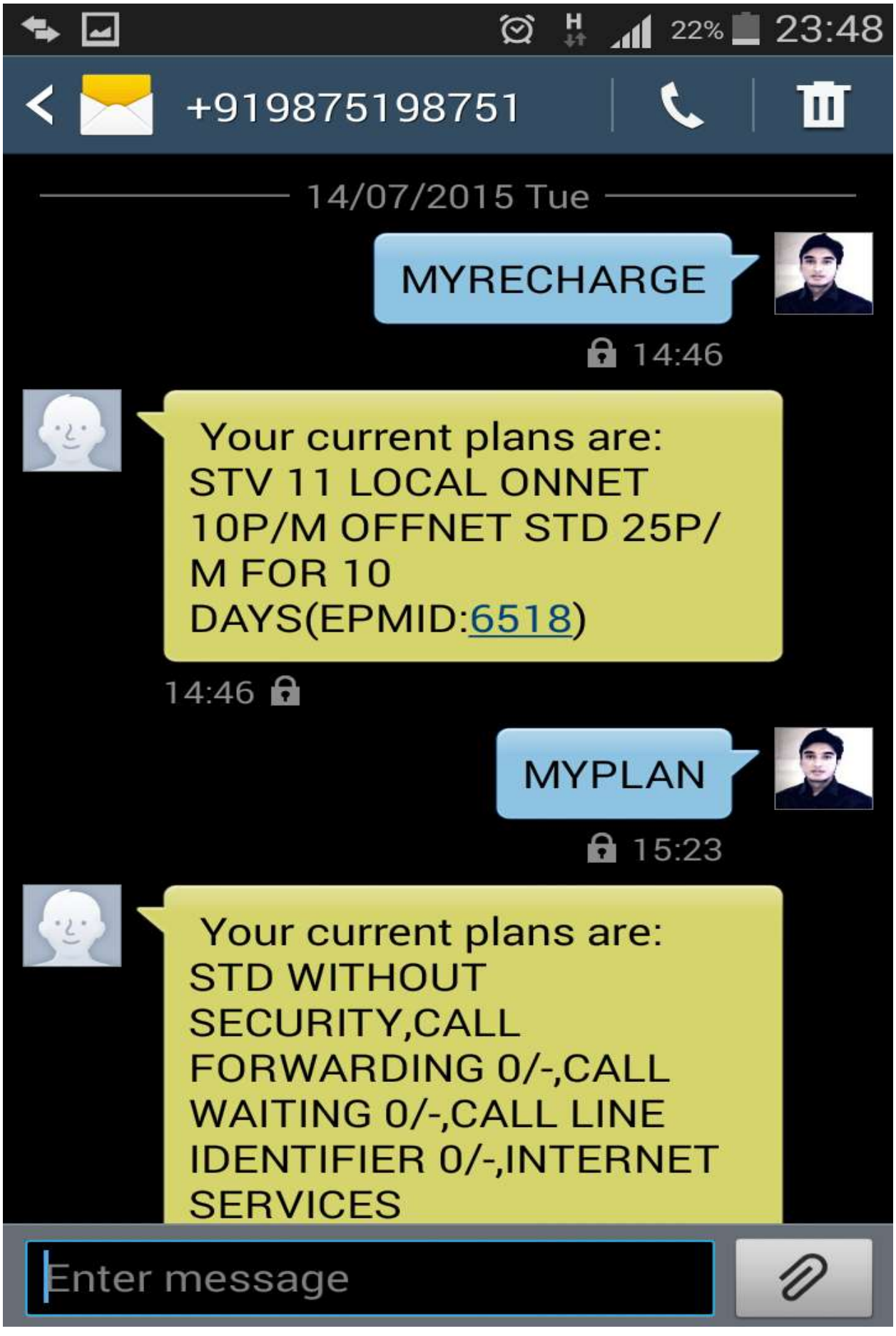


We can even save the data from the table as a text file.


Record Application
MTS
Message
Id
Msg_Id
Calling...
Submit...
FinishTi...
Messag...
Messag...
Keyword
Specify a file to save
Save In:
Documents
SQL Server Management Studio
Visual Studio 2010
bbb.txt
boom.txt
data.pdf
data.txt
db.sql
ddd.txt
dfgdfg.txt
dfsdfs.txt
ppp.xls
rt.sql
tada.txt
File Name:
Files of Type:
All Files
Save
Cancel
OK
11:36
06-07-2015

User Front End




22% 23:48
+919875198751
14/07/2015 Tue
MYRECHARGE
14:46
Your current plans are:
STV 11 LOCAL ONNET
10P/M OFFNET STD 25P/
M FOR 10
DAYS(EPMID:6518)
14:46
MYPLAN
15:23
Your current plans are:
STD WITHOUT
SECURITY,CALL
FORWARDING 0/-,CALL
WAITING 0/-,CALL LINE
IDENTIFIER 0/-,INTERNET
SERVICES
Enter message

Conclusion

In conclusion, we successfully created an SMS p2a application for customers, as well as a front end for the admin to successfully store and retrieve all SMS communication.

References

Books:

- Professional ASP.NET by John Galloway & Brad Wilson
- Oracle Database 11g SQL – Tata McGraw Hill
- ZTE Corporation: ZXSC and SMSC Working (Study material)

Websites:

- W3Schools-http://www.w3schools.com
- TutorialsPoint-http://www.tutorialspoint.com
- TOAD-http://www.toadworld.com.