Project Report CS 6360.004 Fall 2015

(Professor Nurcan Yuruk)

Telecom and Wireless Provider (VConnect)

By

Suraj Poojary (ssp151830)

Soroush Bateni (sxb155530)

Mahesh Sankaranarayanan (mxs154930)

Project Description

VConnect Telecom is a telecommunication company with primary focus on Wireless Mobile network. It also provides other services like Internet, VoIP, Cloud, and Cable TV to customers. The company database stored details about customers and employees. It holds information about various mobile and service plans that it offers. It also stores customer account details and keeps track of the various services associated to the account. Customer credit information may also be stored if the customer opts for automated payment. The company database also tracks the mobile and service transactions from user devices and uses them for billing purposes. The company also keeps a pool of mobile numbers in their database that is used to allocate mobile numbers to new customers.

Whenever a new customer requests for a service, the staff will create an account for the customer and ask for the services that he/she is going to avail. If the person is interested in availing the mobile service. Staff first asks for the number of mobile numbers that the customer wants in his/her account. Staff will pick the mobile numbers from the available pool of mobile numbers and allocate it to the customer's account. The staff also has the responsibility to assign the desired mobile plan to the account. Apart from the generic family or individual plan associated with the account, the customer can also avail different mobile services for each mobile number namely international text/calling, data stash, radio service, etc. If the customer does not want to avail mobile services but other services like Home-Wi-Fi, VOIP, Cable TV, then the staff has to set up the account accordingly. Customer service usage is tracked in the database and this is used for calculating monthly bills. If the customer fails to pay the bill before the specified deadline, fine amount will be automatically added to the bill.

Data Requirements

1. The company stores demographic information about employees as well as customers like name, address, contact number, gender, date of birth, email id, etc. Employees need to have a Social Security Number (SSN) but customer need not have one in order to avail the telecom services.

2. The company needs to store CUSTOMER information like identification and its type in its database. For e.g.: Customer may be registered with the company using his/her SSN, driving license or passport. The company may also store CREDIT card information about the customer if the customer opts for automatic payment.

3. Each EMPLOYEE has SSN, work email, salary and manager details associated. An employee can be a TECHNICIAN or a customer service STAFF. Employees are allowed to avail telecom services similar to customers. Technician has details like area of supervision and availability

status associated. Each customer service staff has a primary operation language that he/she operates in.

4. The company also stores a list of MOBILE NOS. (SIM) that it owns. Each SIM record contains details like mobile no., PUK, IMSI, Activate by date, status (available/unavailable) and whether it is prepaid or postpaid.

5. The company provides a host of MOBILE PLANS with details like plan id (unique), plan name, plan type (family, individual), payment type (prepaid/postpaid), bill duration (monthly, weekly, daily), talk time, data, charge, etc.

6. In addition to the normal mobile plans associated with an account, an individual mobile number can also be associated with multiple additional MOBILE SERVICES like international calls/messages, international roaming, radio service, data stash, caller tune, etc. Each of these service records will have a service id (unique), service name and a charge.

7. Apart from mobile services, the company also provides a host of other services like Wi-Fi, VOIP, Cable TV, Cloud services, Wi-Fi Dongle, etc. Each SERVICE has a service id (unique), service name and monthly charge.

8. Each Customer will have only one ACCOUNT. In case of a family plan, the account will be registered in the name of the primary account holder. Each account will hold details like account id which is basically the mobile number of the primary account holder, account password, security question and answer, activated date, billing address, bill type (paper/paperless), payment type (automated/normal), status (active/inactive) and balance.

9. Each customer who opts for mobile service is registered into the system by a staff member who allocates mobile numbers to his/her account from the available list of mobile numbers. The staff also has the responsibility to setup the account with appropriate mobile plans, mobile services or other services based on the customer’s request.

10. Each account is linked to zero, one (individual) or more MOBILE numbers (family plan). It is possible for an account to have zero mobile services but multiple other services like Cable TV and Wi-Fi.

11. The company stores BILLING information for accounts containing details like bill id (unique), cycle start and end date, payment due date, total charge and payment status.

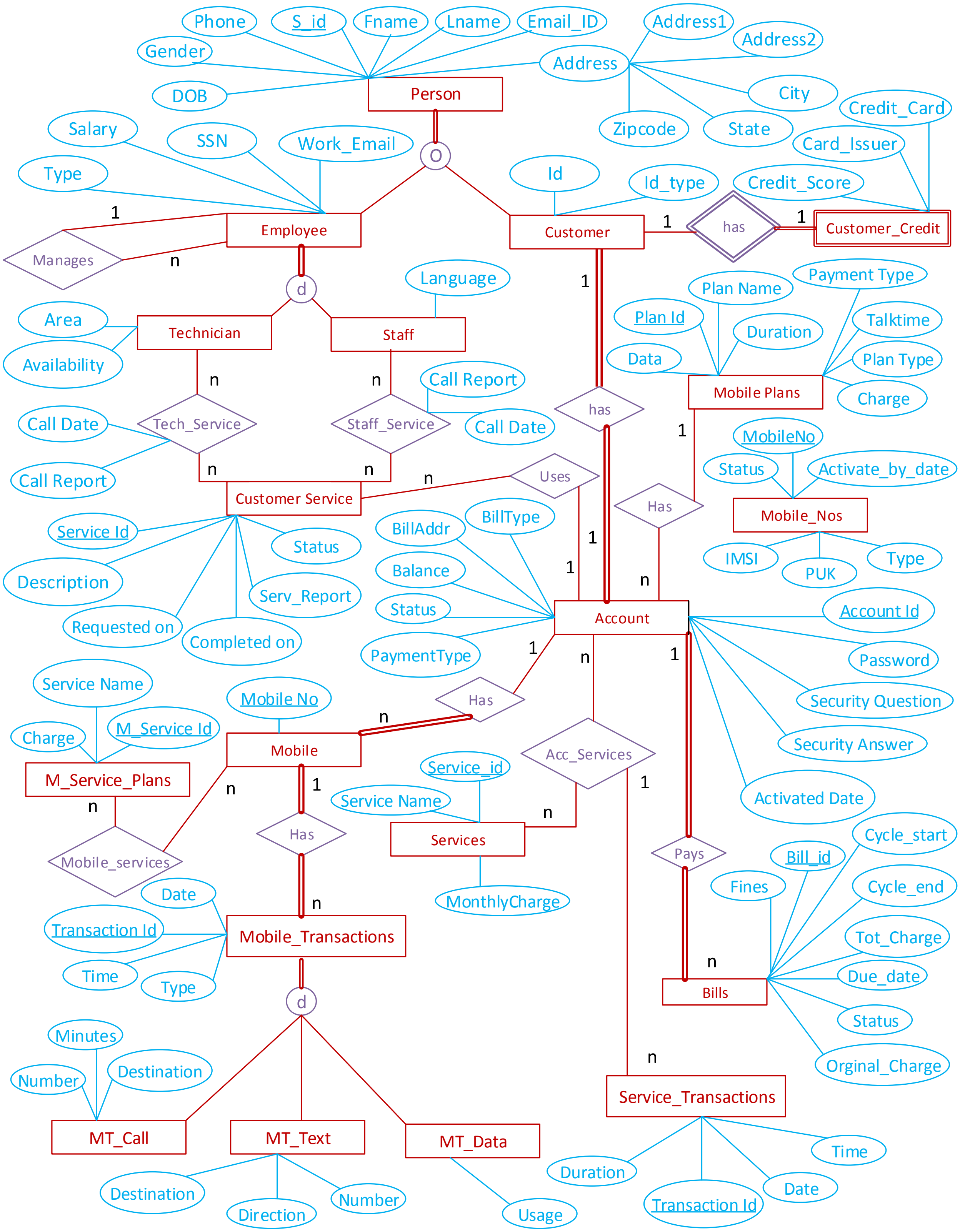
12. The company also keeps details of the MOBILE TRANSACTIONS for each mobile number for customers to track their usages. Each transaction can be a CALL/TEXT/DATA transaction. Each transaction will have generic details like transaction id (unique), date and time. A call

transaction will have additional information like number (recipient number in case of outgoing and caller number in case of incoming calls), minutes (call duration) and destination (place from where the call originated). A text transaction will have additional information like number, destination and direction (incoming/outgoing). A data transaction will also hold usage information.

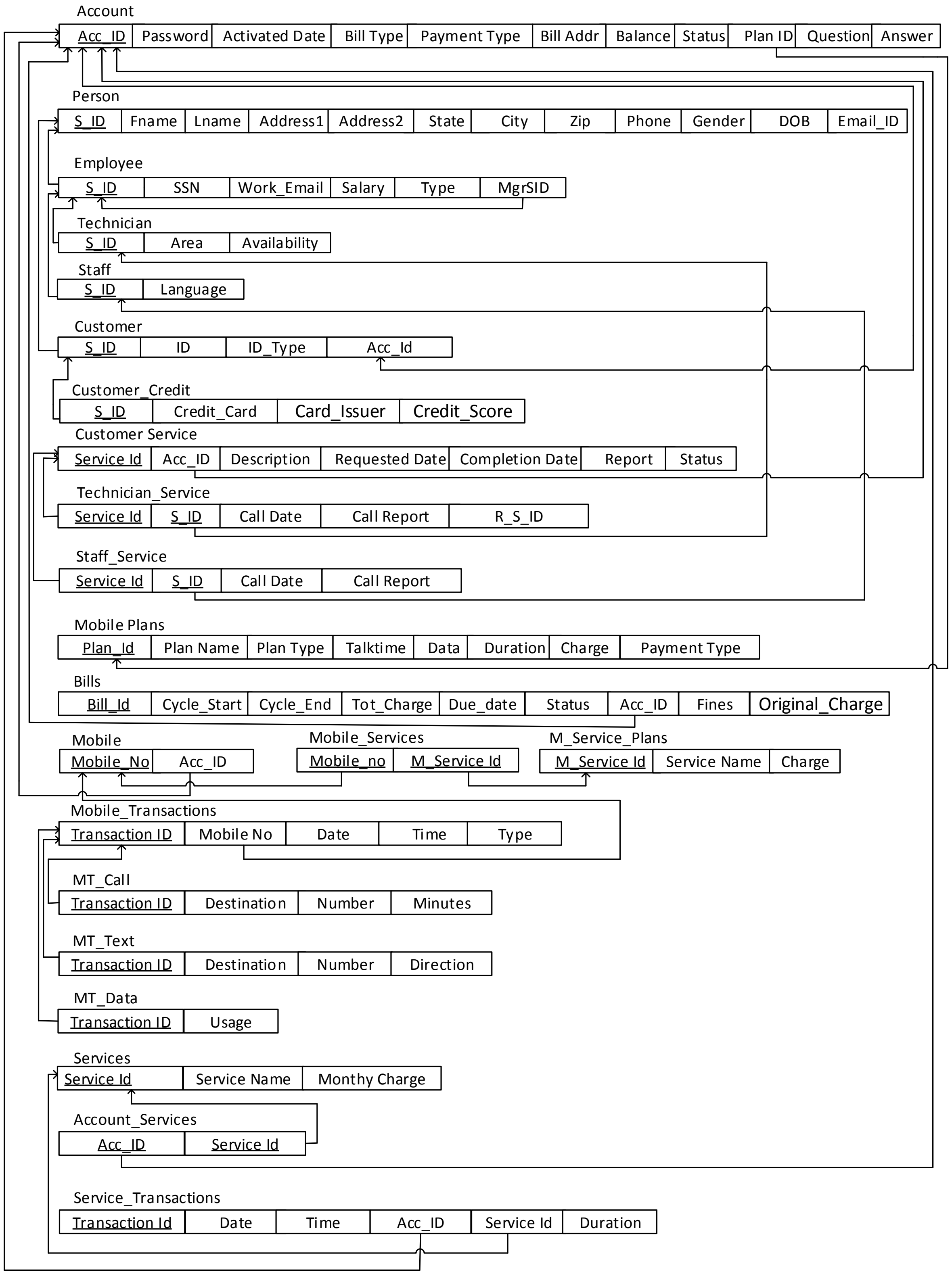
13. Similarly SERVICE TRANSACTIONS (Wi-Fi, Cloud and Cable TV) are also maintained in the database. Each transaction holds details like transaction id (unique), date, time and duration of usage.

14. The company also focuses on the CUSTOMER SERVICE aspect and hence has a separate entity called customer service that holds the service/complaint id (unique), service description, service request date, service complete date, service status and service report. Each staff handles complaints related to accounts and assigns a technician to the complaint if necessary. The database keeps track of the call date and call report for each call handled by the staff and technician.

ER Diagram


Phone
S_id
Fname
Lname
Email_ID
Address1
Address2
Gender
Address
DOB
Person
City
Credit_Card
Zipcode
State
Salary
SSN
Work_Email
O
Card_Issuer
Type
Id
Id_type
Credit_Score
1
1
1
Employee
Customer
has
Customer_Credit
Manages
n
d
Language
1
Plan Name
Payment Type
Area
Technician
Staff
Plan Id
Duration
Talktime
Availability
Data
Plan Type
n
n
Call Report
Mobile Plans
Call Date
Tech_Service
Staff_Service
Call Date
has
Charge
1
MobileNo
Call Report
n
n
n
Uses
Status
Activate_by_date
Customer Service
Has
Mobile_Nos
BillType
Service Id
BillAddr
1
Status
IMSI
Type
Description
Balance
1
n
PUK
Serv_Report
Status
Account
Account Id
Requested on
Completed on
PaymentType
1
n
1
Password
Service Name
Has
Security Question
Mobile No
n
Charge
M_Service Id
Mobile
Acc_Services
Security Answer
M_Service_Plans
Service_id
n
1
1
n
Service Name
Activated Date
n
Has
Services
Pays
Cycle_start
Mobile_services
Bill_id
Date
MonthlyCharge
Fines
Cycle_end
Transaction Id
Mobile_Transactions
n
Tot_Charge
n
Time
Type
Bills
Due_date
d
Status
Minutes
Destination
n
Orginal_Charge
Number
Service_Transactions
MT_Call
MT_Text
MT_Data
Time
Duration
Date
Destination
Number
Transaction Id
Direction
Usage

Relational Schema



Account

Acc_ID	Password	Activated Date	Bill Type	Payment Type	Bill Addr	Balance	Status	Plan ID	Question	Answer

Person

S_ID	Fname	Lname	Address1	Address2	State	City	Zip	Phone	Gender	DOB	Email_ID

Employee

S_ID	SSN	Work_Email	Salary	Type	MgrSID

Technician

S_ID	Area	Availability

Staff

S_ID	Language

Customer

S_ID	ID	ID_Type	Acc_Id

Customer_Credit

S_ID	Credit_Card	Card_Issuer	Credit_Score

Customer Service

Service Id	Acc_ID	Description	Requested Date	Completion Date	Report	Status

Technician_Service

Service Id	S_ID	Call Date	Call Report	R_S_ID

Staff_Service

Service Id	S_ID	Call Date	Call Report

Mobile Plans

Plan_Id	Plan Name	Plan Type	Talktime	Data	Duration	Charge	Payment Type

Bills

Bill_Id	Cycle_Start	Cycle_End	Tot_Charge	Due_date	Status	Acc_ID	Fines	Original_Charge

Mobile

Mobile_No	Acc_ID

Mobile_Services

Mobile_no	M_Service Id

M_Service_Plans

M_Service Id	Service Name	Charge

Mobile_Transactions

Transaction ID	Mobile No	Date	Time	Type

MT_Call

Transaction ID	Destination	Number	Minutes

MT_Text

Transaction ID	Destination	Number	Direction

MT_Data

Transaction ID	Usage

Services

Service Id	Service Name	Monthy Charge

Account_Services

Acc_ID	Service Id

Service_Transactions

Transaction Id	Date	Time	Acc_ID	Service Id	Duration

Functional Dependencies

S_ID -> {Fname,Lname,Address1,Address2,City,State_Name,Zipcode,Phone,Gender,DOB,Email_ID,C_ID,ID_Type,Account_ID,Credit_Card,Card_Issuer,Credit_Score,SSN,Work_Email,Salary,Type,MgrSID,Supervision_Area_Code,Primary_Language}

Plan_ID -> {Plan_Name,Plan_Type,Talktime,Data,Plan_Period,Charge,Payment_Type}

Account_ID -> {Acc_Password,Activated_Date,Security_Question,Security_Answer,Billing_Type,Payment_Type,Billing_Address,Outstanding_Balance,Plan_ID,Account_Status}

Bill_ID -> {Cycle_Start_Date,Cycle_End_DataCycle_End_Date,Total_Charge,Payment_Due_Date,Payment_Status,Fines,Original_Charge,Account_ID}

Service_ID -> {Account_ID,Service_Complaint_Description,Service_Requested_Date,Service_Completion_Date,Service_Report,Service_Status,Service_Name,Monthly_Charge}

{Service_ID,S_ID} -> {Call_Date,Call_Report,R_S_ID}

Mobile_No -> {IMSI,PUK,Type,Activate_By_Date,Availablity,Account_ID,M_Service_ID}

M_Service_ID -> {Service_Name,Charge}

Transaction_ID -> {Mobile_No,MT_Date,MT_Time,MT_Type,Destination,Call_Number,Minutes,Text_Number,Direction,Data_Usage,ST_Date,ST_Time,Account_ID,Service_ID,ST_Duration}

Initial Relational Schema (Normalized)

The intial relational schema mapping created from the ER diagram was already in 3NF since all attributes within each entity were fully functional dependent on the primary key and there were no transitive dependencies within each entity.

Person (S_ID,Fname,Lname,Address1,Address2,City,State_Name,Zipcode,Phone,Gender,DOB,Email_ID)

Customer (S_ID,C_ID,ID_Type,Account_ID)

Customer_Credit (S_ID,Credit_Card,Card_Issuer,Credit_Score)

Employee (S_ID,SSN,Work_Email,Salary,Type,MgrSID)

Technician (S_ID,Supervision_Area_Code)

Staff (S_ID,Primary_Language)

Mobile_Plans (Plan_ID ,Plan_Name,Plan_Type,Talktime,Data,Plan_Period,Charge,Payment_Type)

Account (Account_ID,Acc_Password,Activated_Date,Security_Question,Security_Answer,Billing_Type,Payment_Type,Billing_Address,Outstanding_Balance,Plan_ID,Account_Status)

Bills (Bill_ID,Cycle_Start_Date,Cycle_End_DataCycle_End_Date,Total_Charge,Payment_Due_Date,Payment_Status,Fines,Original_Charge,Account_ID)

Customer_Service (Service_ID,Account_ID,Service_Complaint_Description,Service_Requested_Date,Service_Completion_Date,Service_Report,Service_Status)

Technician_Service (Service_ID,S_ID,Call_Date,Call_Report,R_S_ID)

Staff_Service (Service_ID,S_ID,Call_Date,Call_Report)

Mobile_Nos (Mobile_No,IMSI,PUK,Type,Activate_By_Date)

Mobile (Mobile_No,Account_ID)

Mobile_Services (Mobile_No,M_Service_ID)

M_Service_Plans (M_Service_ID,Service_Name,Charge)

Mobile_Transactions (Transaction_ID,Mobile_No,MT_Date,MT_Time,MT_Type)

MT_Call (Transaction_ID,Destination,Call_Number,Minutes)

MT_Text (Transaction_ID,Destination,Text_Number,Direction)

MT_Data (Transaction_ID,Data_Usage)

Services (Service_ID,Service_Name,Monthly_Charge)

Account_Services (Service_ID,Account_ID)

Service_Transactions (Transaction_ID,ST_Date,ST_Time,Account_ID,Service_ID,ST_Duration)

SQL: Create Table Queries

Creating a Table for Person with Constraints

```
CREATE TABLE Person (
S_ID  NUMBER(10) NOT NULL,
Fname VARCHAR(255) NOT NULL,
Lname VARCHAR(255),
Address1 VARCHAR(255) NOT NULL,
Address2 VARCHAR(255),
City VARCHAR(255) NOT NULL,
State_Name VARCHAR(255) NOT NULL,
Zipcode NUMBER(10) NOT NULL,
Phone NUMBER(10) DEFAULT 0000000000 NOT NULL UNIQUE ,
Gender CHAR,
DOB DATE,
Email_ID VARCHAR(255) NOT NULL,
CONSTRAINT PERSONPK PRIMARY KEY (S_ID),
CONSTRAINT ckGender check (Gender in ('M', 'F'))
);
```

Creating a Table for Mobile_Plans with Constraints

```
CREATE TABLE Mobile_Plans (
Plan_Id NUMBER(10) NOT NULL,
Plan_Name VARCHAR(255) NOT NULL,
Plan_Type NUMBER(1) NOT NULL,
Talktime INTEGER NOT NULL,
Data INTEGER NOT NULL,
Plan_Period NUMBER(1) NOT NULL,
Charge INTEGER NOT NULL,
Payment_Type VARCHAR(8) NOT NULL,
CONSTRAINT MOBILE_PLANSPK PRIMARY KEY (Plan_Id),
constraint ckType check (Plan_Type in (0, 1, 2)),
constraint ckPlanPeriod check (Plan_Period in (0, 1, 2)),
constraint ckPlanType check (Payment_Type in ('Prepaid', 'Postpaid'))
);
```

Creating a Table for Account with Constraints

```
CREATE TABLE Account (
Account_Id NUMBER(10) NOT NULL,
Acc_Password VARCHAR(32) NOT NULL,
Activated_Date DATE NOT NULL,
Security_Question NVARCHAR2(255) NOT NULL,
Security_Answer NVARCHAR2(255) NOT NULL,
Billing_Type VARCHAR(9) NOT NULL,
Payment_Type VARCHAR(6) NOT NULL,
Billing_Address NVARCHAR2(255) NOT NULL,
Outstanding_Balance INTEGER,
Plan_Id NUMBER(10) NOT NULL,
Account_Status INTEGER,
CONSTRAINT AccountPK PRIMARY KEY (Account_Id),
CONSTRAINT Account_Mobile_PlansFK FOREIGN KEY (Plan_Id) REFERENCES Mobile_Plans,
constraint ckBillingType check (Billing_Type in ('Paper', 'Paperless')),
constraint ckPaymentType check (Payment_Type in ('EZpay', 'Normal')),
constraint ckAccountStatus check (Account_Status in (0,1))
);
```

Creating a Table for Customer with Constraints

```
CREATE TABLE Customer (
S_ID NUMBER(10) NOT NULL,
C_ID  NUMBER(10) NOT NULL,
ID_Type NUMBER(1) NOT NULL,
Account_Id NUMBER(10) NOT NULL,
CONSTRAINT CUSTOMERPK PRIMARY KEY (S_ID),
CONSTRAINT CUSTOMER_PERSONFK FOREIGN KEY (S_ID) REFERENCES Person,
CONSTRAINT CUSTOMER_ACCOUNTFK FOREIGN KEY (Account_Id) REFERENCES Account,
constraint ckIDTYPE check (ID_Type in (0, 1, 2))
);
```

Creating a Table for Customer_Credit with Constraints

```
CREATE TABLE Customer_Credit (
S_ID NUMBER(10) NOT NULL,
Credit_Card NUMBER(16) NOT NULL,
Card_Issuer VARCHAR(30) NOT NULL,
Credit_Score INTEGER,
CONSTRAINT CUSTOMER_CREDITPK PRIMARY KEY (S_ID),
CONSTRAINT CUSTOMER_CREDIT_CUSTOMERFK FOREIGN KEY (S_ID) REFERENCES Customer
ON DELETE CASCADE
);
```

Creating a Table for Employee with Constraints

```
CREATE TABLE Employee (
S_ID NUMBER(10) NOT NULL,
SSN NUMBER(9) NOT NULL UNIQUE,
Work_Email VARCHAR(255),
Salary INTEGER,
Type CHAR,
MgrSID NUMBER(10) NOT NULL,
CONSTRAINT EMPLOYEEPK PRIMARY KEY (S_ID),
CONSTRAINT EMPLOYEE_PERSONFK FOREIGN KEY (S_ID) REFERENCES Person,
CONSTRAINT ckTypeEmp check (Type in ('T','S'))
);

ALTER TABLE EMPLOYEE
ADD CONSTRAINT EMPLOYEE_EMPLOYEEFK FOREIGN KEY (MgrSID) REFERENCES
Employee(S_ID);
```

Creating a Table for Technician with Constraints

```
CREATE TABLE Technician (
Supervision_Area_Code NUMBER(5),
S_ID NUMBER(10) NOT NULL,
Availability VARCHAR(3) DEFAULT 'yes' NOT NULL ,
CONSTRAINT TECHNICIANPK PRIMARY KEY (S_ID),
CONSTRAINT TECHNICIAN_EMPLOYEEFK FOREIGN KEY (S_ID) REFERENCES Employee,
CONSTRAINT ckAvailablity check (Availability in ('yes','no'))
);
```

Creating a Table for Staff with Constraints

```
CREATE TABLE Staff (
Primary_Language NVARCHAR2(25),
S_ID NUMBER(10) NOT NULL,
CONSTRAINT STAFFPK PRIMARY KEY (S_ID),
CONSTRAINT STAFF_EMPLOYEEFK FOREIGN KEY (S_ID) REFERENCES Employee
);
```

Creating a Table for Bills with Constraints

```
CREATE TABLE Bills (
Bill_Id NUMBER(38) NOT NULL,
Cycle_Start_Date DATE NOT NULL,
Cycle_End_DateCycle_End_Date DATE NOT NULL,
Total_Charge INTEGER NOT NULL,
Payment_Due_Date DATE NOT NULL,
Payment_Status INTEGER NOT NULL,
Fines INTEGER DEFAULT 0 NOT NULL,
Original_Charge INTEGER NOT NULL,
Account_Id NUMBER(10) NOT NULL,
CONSTRAINT BILLSPK PRIMARY KEY (Bill_Id),
CONSTRAINT BILLS_ACCOUNTFK FOREIGN KEY (Account_Id) REFERENCES Account ON DELETE
CASCADE,
CONSTRAINT ckPayment_Status check (Payment_Status in ('Y','N'))
);
```

Creating a Table for Customer_Service with Constraints

```
CREATE TABLE Customer_Service (
Service_Id NUMBER(38) NOT NULL,
Account_Id NUMBER(10),
Service_Complaint_Description NVARCHAR2(2000) NOT NULL,
Service_Requested_Date DATE,
Service_Completion_Date DATE,
Service_Report NVARCHAR2(2000),
Service_Status INTEGER,
CONSTRAINT CUSTOMER_SRVCPK PRIMARY KEY (Service_Id),
CONSTRAINT CUSTOMER_SRVC_ACCOUNTFK FOREIGN KEY (Account_Id) REFERENCES Account
ON DELETE CASCADE
);
```

Creating a Table for Staff_Service with Constraints

```
CREATE TABLE Staff_Service (
Service_Id NUMBER(38) NOT NULL,
S_ID NUMBER(10) NOT NULL,
Call_Date DATE,
Call_Report NVARCHAR2(2000),
CONSTRAINT STAFF_SRVCPK PRIMARY KEY (Service_Id,S_ID),
CONSTRAINT STAFF_SRVC_CUSTOMER_SRVCFK FOREIGN KEY (Service_Id) REFERENCES
Customer_Service ON DELETE CASCADE,
CONSTRAINT STAFF_SRVC_STAFFFK FOREIGN KEY (S_ID) REFERENCES Staff
);
```

Creating a Table for Technician_Service with Constraints

```
CREATE TABLE Technician_Service (
Service_Id NUMBER(38) NOT NULL,
S_ID NUMBER(10) NOT NULL,
Call_Date DATE,
Call_Report NVARCHAR2(2000),
R_S_ID NUMBER(10) NOT NULL,
CONSTRAINT TECHN_SRVCPK PRIMARY KEY (Service_Id,S_ID),
CONSTRAINT TECHN_SRVC_CUSTOMER_SRVCFK FOREIGN KEY (Service_Id) REFERENCES
Customer_Service ON DELETE CASCADE,
CONSTRAINT TECHN_SRVC_TECHNICIANFK FOREIGN KEY (S_ID) REFERENCES Technician
);
```

Creating a Table for Mobile_Nos with Constraints

```
CREATE TABLE Mobile_Nos (
Mobile_No NUMBER(10) NOT NULL,
IMSI NUMBER(15) NOT NULL,
PUK NUMBER(8) NOT NULL,
Type VARCHAR(8) NOT NULL,
Activate_By_Date DATE NOT NULL,
Availability NUMBER(1) NOT NULL,
CONSTRAINT MOBILE_NOSPK PRIMARY KEY (Mobile_No),
constraint ckTypeMobNos check (Type in ('Prepaid', 'Postpaid')),
constraint ckAvailability check (Availability in (0,1))
);
```

Creating a Table for Mobile with Constraints

```
CREATE TABLE Mobile (
Mobile_No NUMBER(10) NOT NULL,
Account_Id NUMBER(10) NOT NULL,
CONSTRAINT MOBILEPK PRIMARY KEY (Mobile_No),
CONSTRAINT MOBILE_ACCOUNTFK FOREIGN KEY (Account_Id) REFERENCES Account ON
DELETE CASCADE
);
```

Creating a Table for M_Service_Plans with Constraints

```
CREATE TABLE M_Service_Plans (
M_Service_Id NUMBER(10) NOT NULL,
Service_Name VARCHAR(255) NOT NULL,
Charge INTEGER NOT NULL,
CONSTRAINT M_SRVC_PLANSPK PRIMARY KEY (M_Service_Id)
);
```

Creating a Table for Mobile_Services with Constraints

```
CREATE TABLE Mobile_Services (
Mobile_No NUMBER(10) NOT NULL,
M_Service_Id NUMBER(10) NOT NULL,
CONSTRAINT MOBILE_SRVCSPK PRIMARY KEY (Mobile_No,M_Service_Id),
CONSTRAINT MOBILE_SRVCS_MOBILEFK FOREIGN KEY (Mobile_No) REFERENCES Mobile ON
DELETE CASCADE,
CONSTRAINT MOBILE_SRVCS_M_SRVCS_PLANSFK FOREIGN KEY (M_Service_Id) REFERENCES
M_Service_Plans
);
```

Creating a Table for Mobile_Transactions with Constraints

```
CREATE TABLE Mobile_Transactions (
Transaction_ID NUMBER(38) NOT NULL,
Mobile_No NUMBER(10) NOT NULL,
MT_Date DATE NOT NULL,
MT_Time TIMESTAMP(0) NOT NULL,
MT_Type NUMBER(1) NOT NULL,
CONSTRAINT MOBILE_XSPK PRIMARY KEY (Transaction_ID),
CONSTRAINT MOBILE_XS_MOBILEFK FOREIGN KEY (Mobile_No) REFERENCES Mobile ON DELETE CASCADE,
CONSTRAINT ckTypeMobTrans check (MT_Type in (0, 1, 2))
);
```

Creating a Table for MT_Call with Constraints

```
CREATE TABLE MT_Call          (
Transaction_ID NUMBER(38) NOT NULL,
Destination VARCHAR(255),
Call_Number NUMBER(10),
Minutes INTEGER,
CONSTRAINT MT_CALLPK PRIMARY KEY (Transaction_ID),
CONSTRAINT MT_CALL_MOBILE_XSFK FOREIGN KEY (Transaction_ID) REFERENCES Mobile_Transactions ON DELETE CASCADE
);
```

Creating a Table for MT_Text with Constraints

```
CREATE TABLE MT_Text          (
Transaction_ID NUMBER(38) NOT NULL,
Destination VARCHAR(255),
Text_Number NUMBER(10),
Direction VARCHAR(8),
CONSTRAINT MT_TEXTPK PRIMARY KEY (Transaction_ID),
CONSTRAINT MT_TEXT_MOBILE_XSFK FOREIGN KEY (Transaction_ID) REFERENCES Mobile_Transactions ON DELETE CASCADE,
CONSTRAINT ckDirection check (Direction in ('Outgoing','Incoming'))
);
```

Creating a Table for MT_Data with Constraints

```
CREATE TABLE MT_Data          (
Transaction_ID NUMBER(38) NOT NULL,
Data_Usage DECIMAL(7,3),
CONSTRAINT MT_DATAPK PRIMARY KEY (Transaction_ID),
CONSTRAINT MT_DATA_MOBILE_XSFK FOREIGN KEY (Transaction_ID) REFERENCES Mobile_Transactions ON DELETE CASCADE
);
```

Creating a Table for Services with Constraints

```
CREATE TABLE Services          (
Service_Id NUMBER(10) NOT NULL,
Service_Name VARCHAR(255) NOT NULL,
Monthy_Charge INTEGER NOT NULL,
CONSTRAINT SERVICESPK PRIMARY KEY (Service_Id)
);
```

Creating a Table for Account_Services with Constraints

```
CREATE TABLE Account_Services        (
Account_Id NUMBER(10),
Service_Id NUMBER(10),
CONSTRAINT ACCOUNT_SRVCSPK PRIMARY KEY (Account_Id,Service_Id),
CONSTRAINT ACCOUNT_SRVCS_ACCOUNTFK FOREIGN KEY (Account_Id) REFERENCES Account
ON DELETE CASCADE,
CONSTRAINT ACCOUNT_SRVCS_SRVCSFK FOREIGN KEY (Service_Id) REFERENCES Services
);
```

Creating a Table for Service_Transactions with Constraints

```
CREATE TABLE Service_Transactions (
Transaction_Id NUMBER(38) NOT NULL,
ST_Date DATE NOT NULL,
ST_Time TIMESTAMP(0) NOT NULL,
Account_Id NUMBER(10),
Service_Id NUMBER(10),
ST_Duration INTEGER NOT NULL,
CONSTRAINT SRVC_XSPK PRIMARY KEY (Transaction_Id),
CONSTRAINT SRVC_XS_ACCOUNT_SRVCSFK FOREIGN KEY (Account_Id,Service_Id) REFERENCES
Account_Services ON DELETE CASCADE
);
```

SQL: Insert Queries

```
insert into Person values(0000012345,'John','Liner','145 Mccallum Glenn','Coit
Road','Dallas','Texas',75252,4699639182,'M','18-JUL-90','johnliner@gmail.com');
insert into Person values(0000012346,'James','Bond','135 Mccallum Glenn','Coit
Road','Dallas','Texas',75252,7896373883,'M','27-AUG-89','jamesbond@gmail.com');
insert into Person values(0000012347,'Jake','Layer','4 Mccallum Glenn','Coit
Road','Dallas','Texas',75252,1234567890,'M','12-FEB-97','jakelayer@hotmail.com');
insert into Person values(0000012348,'Chris','Martin','45 Mccallum Glenn','Coit
Road','Dallas','Texas',75252,5679876152,'M','11-JAN-99','chrismartin@gmail.com');
insert into Person values(0000012349,'Sebastian','Pear','145 Mccallum Crossings','Coit
Road','Dallas','Texas',75252,1837484848,'M','11-AUG-90','sebastianpear@yahoo.com');
insert into Person values(0000012350,'Emilie','Namterme','145 Mccallum Crossings','Coit
Road','Dallas','Texas',75252,1837484849,'F','11-AUG-91','emilie@hotmail.com');
insert into Person values(0000012351,'Clara','Ponting','80 Mccallum Crossings','Coit
Road','Dallas','Texas',75252,1837484850,'F','11-AUG-92','clara@yahoo.com');
insert into Person values(0000012352,'Savita','Tendulkar','100 Mccallum Glen','Coit
Road','Dallas','Texas',75252,1837484851,'F','11-AUG-93','savita@gmail.com');
```

```
insert into Person values(0000012353,'Kleen','Dsouza','145 Mccallum Crossings','Coit
Road','Dallas','Texas',75252,1837484852,'F','11-AUG-94','kleen.dsouza@hotmail.com');
insert into Person values(0000012354,'Virat','Kohli','100 Chatham','Coit
Road','Dallas','Texas',75252,1837484853,'M','11-AUG-95','virat@rashroad.com');
insert into Person values(0000012355,'Rohit','Sharma','145 Mccallum Crossings','Coit
Road','Dallas','Texas',75252,1837484854,'M','11-AUG-96','rohit@lazybug.com');
insert into Person values(0000012356,'Hello','World','145 Chatham','Coit
Road','Dallas','Texas',75252,1837484855,'M','11-AUG-97','hello.world@gmail.com');
insert into Person values(0000012357,'George','Wale','900 Mccallum Crossings','Coit
Road','Dallas','Texas',75252,1837484856,'M','11-AUG-98','geoge.wale@gmail.com');
insert into Person values(0000012358,'Renny','James','145 Chatham','Coit
Road','Dallas','Texas',75252,1837484857,'M','11-AUG-99','renny@james@gmail.com');
insert into Person values(0000012359,'Pier','Namterme','800 Mccallum Crossings','Coit
Road','Dallas','Texas',75252,1837484858,'M','11-AUG-100','pier@accenture.com');

insert into Mobile_Plans values(1,'Family Plan',0,-1,-1,0,100,'Postpaid');
insert into Mobile_Plans values(2,'Family Plan',0,-1,-1,0,100,'Postpaid');
insert into Mobile_Plans values(3,'Small Family Plan',0,-1,-1,0,80,'Postpaid');
insert into Mobile_Plans values(4,'Simple Choice',1,100,4000,0,40,'Postpaid');
insert into Mobile_Plans values(5,'Pay on the Go',2,200,4000,2,10,'Prepaid');

insert into Account values(0000012386,'Account@123','01-Dec-15','What is your favorite
pet?','Cat','Paperless','EZpay','145 Mccallum Glenn',0,0000000001,0);
insert into Account values(0000012387,'Account@001','5-Dec-15','What is your favorite
pet?','Dog','Paperless','EZpay','135 Mccallum Glenn',0,0000000001,0);
insert into Account values(0000012388,'Acc1234','01-Nov-15','What is your favorite
pet?','Lion','Paperless','Normal','4 Mccallum Glenn',0,0000000001,0);
insert into Account values(0000012389,'Mobile123','01-Dec-15','What is your favorite
pet?','Cat','Paperless','Normal','45 Mccallum Glenn',0,0000000001,0);
insert into Account values(0000012390,'Hello09','01-Dec-15','What is your favorite
pet?','Dog','Paperless','Normal','145 Mccallum Crossings',0,0000000001,0);

insert into Customer values(0000012345,1210000000,0,0000012386);
insert into Customer values(0000012346,1210000001,0,0000012387);
insert into Customer values(0000012347,1210000002,0,0000012388);
insert into Customer values(0000012348,1210000003,1,0000012389);
insert into Customer values(0000012349,1210000004,2,0000012390);

insert into Customer_Credit values(0000012345,9383747488668878,'Wells Fargo',0);
insert into Customer_Credit values(0000012346,9383747488668878,'Wells Fargo',100);
insert into Customer_Credit values(0000012347,9383747488668878,'Wells Fargo',10);
insert into Customer_Credit values(0000012348,9383747488668878,'Wells Fargo',20);
insert into Customer_Credit values(0000012349,9383747488668878,'Wells Fargo',200);

insert into Employee
values(0000012350,123678367,'asas1516@company.com',30,'S',0000012351);
insert into Employee
values(0000012351,123678368,'adyads172@company.com',50,'S',0000012351);
insert into Employee
values(0000012352,123678369,'sdhsd6126@company.com',35,'S',0000012351);
insert into Employee
values(0000012353,123678370,'acasd12@company.com',45,'S',0000012351);
insert into Employee
values(0000012354,123678371,'adasd67@company.com',45,'S',0000012351);
insert into Employee
values(0000012355,123678372,'asdf12@company.com',20,'T',0000012351);
insert into Employee
values(0000012356,123678373,'sdafs23e2@company.com',15,'T',0000012351);
insert into Employee
values(0000012357,123678374,'sadfsdf122@company.com',15,'T',0000012351);
insert into Employee
values(0000012358,123678375,'sadf23e@company.com',15,'T',0000012351);
insert into Employee
values(0000012359,123678376,'sdsdf12@company.com',15,'T',0000012351);

insert into Technician values(75252,0000012355,'yes');
insert into Technician values(75252,0000012356,'yes');
insert into Technician values(75252,0000012357,'yes');
insert into Technician values(75252,0000012358,'yes');
insert into Technician values(75252,0000012359,'yes');

insert into Staff values('English',0000012350);
insert into Staff values('French',0000012351);
insert into Staff values('English',0000012352);
insert into Staff values('English',0000012353);
insert into Staff values('French',0000012354);

insert into Bills values(000001000000,'01-Nov-15','30-Nov-15',200,'10-Dec-
15',0,0,200,0000012386);
insert into Bills values(000001000001,'01-Oct-15','31-Oct-15',100,'10-Nov-
16',0,0,100,0000012387);
```

```
insert into Bills values(000001000002,'01-Nov-15','30-Nov-15',150,'10-Dec-17',0,0,150,0000012388);
insert into Bills values(000001000003,'01-Nov-15','30-Nov-15',50,'10-Dec-18',0,0,50,0000012389);
insert into Bills values(000001000004,'01-Nov-15','30-Nov-15',120,'10-Dec-19',0,0,120,0000012390);

insert into Customer_Service values(111112237876,0000012386,'No Internet','6-DEC-15','6-DEC-15','Issue with mobile. Fixed',1);
insert into Customer_Service values(111112234545,0000012387,'Huge Bills','6-NOV-15','6-NOV-15','Bill had issue. Fixed',1);
insert into Customer_Service values(111112238989,0000012388,'Payment not reflected','6-DEC-15','6-DEC-15','Fixed',1);
insert into Customer_Service values(111112237880,0000012389,'No connection','1-DEC-15','1-DEC-15','Issue with mobile. Fixed',1);
insert into Customer_Service values(111112237899,0000012390,'Payment not reflected','1-DEC-15','1-DEC-15','Fixed',1);

insert into Staff_Service values(111112237876,0000012350,'6-DEC-15','Issue with mobile. Fixed');
insert into Staff_Service values(111112234545,0000012351,'6-NOV-15','Bill had issue. Fixed');
insert into Staff_Service values(111112238989,0000012352,'6-DEC-15','Fixed');
insert into Staff_Service values(111112237880,0000012353,'1-DEC-15','Issue with mobile. Fixed');
insert into Staff_Service values(111112237899,0000012354,'1-DEC-15','Fixed');

insert into Technician_Service values(111112237876,0000012355,'6-DEC-15','Issue with mobile. Fixed',0000012350);
insert into Technician_Service values(111112234545,0000012356,'6-NOV-15','Bill had issue. Fixed',0000012351);
insert into Technician_Service values(111112238989,0000012357,'6-DEC-15','Fixed',0000012352);
insert into Technician_Service values(111112237880,0000012358,'1-DEC-15','Issue with mobile. Fixed',0000012353);
insert into Technician_Service values(111112237899,0000012359,'1-DEC-15','Fixed',0000012354);

insert into Mobile_Nos values(4699639182,3223423427,83735373,'Postpaid','12-DEC-16',1);
insert into Mobile_Nos values(7896373883,3223423428,83735374,'Postpaid','12-DEC-17',1);
insert into Mobile_Nos values(1234567890,3223423429,83735375,'Postpaid','12-DEC-18',1);
insert into Mobile_Nos values(5679876152,3223423430,83735376,'Postpaid','12-DEC-19',1);
insert into Mobile_Nos values(1837484848,3223423431,83735377,'Postpaid','12-DEC-20',1);
insert into Mobile_Nos values(1234567891,3223423432,83735378,'Postpaid','12-DEC-21',0);
insert into Mobile_Nos values(1234567892,3223423433,83735379,'Postpaid','12-DEC-22',0);
insert into Mobile_Nos values(1234567893,3223423434,83735380,'Postpaid','12-DEC-23',0);
insert into Mobile_Nos values(1234567894,3223423435,83735381,'Postpaid','12-DEC-24',0);
insert into Mobile_Nos values(1234567895,3223423436,83735382,'Postpaid','12-DEC-25',0);

insert into Mobile values(4699639182,0000012386);
insert into Mobile values(7896373883,0000012387);
insert into Mobile values(1234567890,0000012388);
insert into Mobile values(5679876152,0000012389);
insert into Mobile values(1837484848,0000012390);

insert into M_Service_Plans values(000000001,'International Roaming',15);
insert into M_Service_Plans values(000000002,'International Text',15);
insert into M_Service_Plans values(000000003,'Radio Services',10);
insert into M_Service_Plans values(000000004,'Data Stash',10);
insert into M_Service_Plans values(000000005,'Caller Tunes',10);

insert into Mobile_Services values(4699639182,000000001);
insert into Mobile_Services values(7896373883,000000002);
insert into Mobile_Services values(1234567890,000000003);
insert into Mobile_Services values(5679876152,000000004);
insert into Mobile_Services values(1837484848,000000005);

insert into Mobile_Transactions values(0000110001,4699639182,'10-NOV-15','10-NOV-2015
08.00.05 AM',0);
insert into Mobile_Transactions values(0000110002,7896373883,'10-NOV-15','10-NOV-2015
08.00.05 PM',0);
insert into Mobile_Transactions values(0000110003,1234567890,'10-NOV-15','10-NOV-2015
12.00.05 AM',0);
insert into Mobile_Transactions values(0000110004,5679876152,'10-NOV-15','10-NOV-2015
11.00.05 AM',0);
insert into Mobile_Transactions values(0000110005,1837484848,'10-NOV-15','10-NOV-2015
11.20.05 AM',0);
insert into Mobile_Transactions values(0000110006,4699639182,'10-NOV-15','10-NOV-2015
08.00.05 AM',1);
insert into Mobile_Transactions values(0000110007,7896373883,'10-NOV-15','10-NOV-2015
08.00.05 AM',1);
insert into Mobile_Transactions values(0000110008,1234567890,'10-NOV-15','10-NOV-2015
08.00.05 AM',1);
```

```
insert into Mobile_Transactions values(0000110009,5679876152,'10-NOV-15','10-NOV-2015
08.00.05 AM',1);
insert into Mobile_Transactions values(0000110010,1837484848,'10-NOV-15','10-NOV-2015
08.00.05 AM',1);
insert into Mobile_Transactions values(0000110011,4699639182,'10-NOV-15','10-NOV-2015
08.00.05 AM',2);
insert into Mobile_Transactions values(0000110012,7896373883,'10-NOV-15','10-NOV-2015
08.00.05 AM',2);
insert into Mobile_Transactions values(0000110013,1234567890,'10-NOV-15','10-NOV-2015
08.00.05 AM',2);
insert into Mobile_Transactions values(0000110014,5679876152,'10-NOV-15','10-NOV-2015
08.00.05 AM',2);
insert into Mobile_Transactions values(0000110015,1837484848,'10-NOV-15','10-NOV-2015
08.00.05 AM',2);

insert into MT_Call values(0000110001,'Texas',1231231231,10);
insert into MT_Call values(0000110002,'Austin',1231231232,20);
insert into MT_Call values(0000110003,'Austin',1231231233,21);
insert into MT_Call values(0000110004,'Texas',1231231234,22);
insert into MT_Call values(0000110005,'Texas',1231231235,23);

insert into MT_Text values(0000110006,'Texas',1231231231,'Outgoing');
insert into MT_Text values(0000110007,'Austin',1231231232,'Outgoing');
insert into MT_Text values(0000110008,'Austin',1231231233,'Outgoing');
insert into MT_Text values(0000110009,'Texas',1231231234,'Outgoing');
insert into MT_Text values(0000110010,'Texas',1231231235,'Outgoing');

insert into MT_Data values(0000110011,12.46);
insert into MT_Data values(0000110012,12.47);
insert into MT_Data values(0000110013,12.48);
insert into MT_Data values(0000110014,12.49);
insert into MT_Data values(0000110015,12.50);

insert into Services values(0001,'Wifi',60);
insert into Services values(0002,'Cable TV',100);
insert into Services values(0003,'Cloud',20);
insert into Services values(0004,'VoIP',10);
insert into Services values(0005,'Wifi-Dongle',25);

insert into Account_Services values(0000012386,0001);
insert into Account_Services values(0000012387,0002);
insert into Account_Services values(0000012388,0003);
insert into Account_Services values(0000012389,0004);
insert into Account_Services values(0000012390,0005);

insert into Service_Transactions values(0000001,'10-NOV-15','10-NOV-2015 08.00.05
AM',0000012386,0001,23);
insert into Service_Transactions values(0000002,'10-NOV-15','10-NOV-2015 08.00.05
PM',0000012387,0002,24);
insert into Service_Transactions values(0000003,'10-NOV-15','10-NOV-2015 12.00.05
AM',0000012388,0003,25);
insert into Service_Transactions values(0000004,'10-NOV-15','10-NOV-2015 11.00.05
AM',0000012389,0004,26);
insert into Service_Transactions values(0000005,'10-NOV-15','10-NOV-2015 11.20.05
AM',0000012390,0005,27);
```

PL/SQL

Stored Procedure 1: To pick up available technicians and assign them to current service id

```
CREATE OR REPLACE PROCEDURE assign_tech (Service_Id IN
Customer_Service.Service_Id%Type, R_SID IN Staff.S_ID%Type, tmpDate IN DATE)
AS
CURSOR C_TECH
        IS
                SELECT S_ID from Technician where Availability='yes' and rownum=1;
        R_TECH C_TECH%ROWTYPE;
BEGIN
        OPEN C_TECH;
        FETCH C_TECH INTO R_TECH;
        IF C_TECH%NOTFOUND THEN
                DBMS_OUTPUT.put_line ('No technician is available right now. Your Request had
been queued.');
        ELSE
                insert into Technician_Service values (Service_Id,R_TECH.S_ID,tmpDate,'',R_SID);
                UPDATE Technician SET Availability='no' WHERE S_ID=R_TECH.S_ID;
        END IF;
        CLOSE C_TECH;
END;
```

Stored Procedure 2: To update the availability of the technician

```
CREATE OR REPLACE PROCEDURE CALL_REPORT(
P_Service_Id IN Customer_Service.Service_Id%Type,
P_S_ID IN Technician.S_ID%Type,
P_Call_Report IN VARCHAR) AS
BEGIN
        UPDATE Technician_Service
        SET Call_Report=P_Call_Report WHERE Service_Id=P_Service_Id AND S_ID=P_S_ID;
END;
```

Stored Procedure 3: Bill Calculation

```
CREATE OR REPLACE PROCEDURE Calc_Bill(
ACC_ID IN Mobile.Account_Id%Type,
START_DATE IN BILLS.Cycle_Start_Date%TYPE,
END_DATE IN BILLS.Cycle_End_Date%TYPE) AS
CURSOR c_bill
  IS
   SELECT SUM(CHARGE) AS COST
        FROM MOBILE M JOIN MOBILE_SERVICES MS
        ON M.MOBILE_NO=MS.MOBILE_NO
        JOIN M_SERVICE_PLANS MP
        ON MS.M_SERVICE_ID=MP.M_SERVICE_ID
        WHERE M.Account_ID=ACC_ID
        GROUP BY M.MOBILE_NO;
        r_bill   c_bill%ROWTYPE;
        BASIC_CHARGE NUMBER;
        TOT_CHARGE NUMBER;
        NEXTBILLID NUMBER;
BEGIN
        TOT_CHARGE:=0;
        OPEN c_bill;
        LOOP
                FETCH c_bill INTO r_bill;
                        TOT_CHARGE:=TOT_CHARGE+r_bill.COST;
                EXIT WHEN c_bill%NOTFOUND;
        END LOOP;
        SELECT MAX(BILL_ID)+1 INTO NEXTBILLID FROM BILLS;

        SELECT CHARGE INTO BASIC_CHARGE
        FROM MOBILE_PLANS M JOIN ACCOUNT A
        ON A.PLAN_ID=M.PLAN_ID
        WHERE ACCOUNT_ID=ACC_ID;
```

```
    INSERT INTO BILLS VALUES
    (NEXTBILLID,START_DATE,END_DATE,BASIC_CHARGE+TOT_CHARGE,END_DATE+10,'N',A
    CC_ID,0,BASIC_CHARGE+TOT_CHARGE);
    CLOSE c_bill;
END;
```

Stored Procedure 4: Fines for delay in bill payment

```
CREATE OR REPLACE PROCEDURE Calc_Fines AS
CURSOR c_due_check
IS
 SELECT BILL_ID FROM BILLS WHERE Payment_Due_Date < TO_DATE(SYSDATE,'DD-MON-YY')
AND Payment_Status='N';
    due_check   c_due_check%ROWTYPE;
BEGIN
    OPEN c_due_check;
    LOOP
        FETCH c_due_check INTO due_check;
            UPDATE BILLS
            SET FINES=(0.005*TOTAL_CHARGE)*ROUND(SYSDATE-
Payment_Due_Date),

    TOTAL_CHARGE=ORIGINAL_CHARGE+(0.005*TOTAL_CHARGE)*ROUND(SYSDATE-
Payment_Due_Date)
            WHERE BILL_ID=due_check.BILL_ID;
        EXIT WHEN c_due_check%NOTFOUND;
    END LOOP;
    CLOSE c_due_check;
END;
```

Stored Procedure 5: Allocates Mobile Number to Account

```
CREATE OR REPLACE PROCEDURE Fetch_mobile (
M_Type IN Mobile_Nos.Type%Type,
ACC_ID IN Mobile.Account_Id%Type,
SID IN PERSON.S_ID%Type,
Nos IN Number)
AS
CURSOR C_mob
    IS
        SELECT Mobile_No FROM
            (SELECT Mobile_No FROM Mobile_Nos
            where Type=M_Type and Availability=0
```

```
                                        ORDER BY Activate_By_Date ASC)
                        WHERE  rownum <= Nos;
            R_mob C_mob%ROWTYPE;
            COUNTER NUMBER;
BEGIN
            COUNTER:=0;
            OPEN C_mob;
            FETCH C_mob INTO R_mob;
            IF C_mob%NOTFOUND THEN
                        IF COUNTER > 0 THEN
                                    UPDATE PERSON SET Phone=R_mob.Mobile_No WHERE S_ID=SID;
                        END IF;
                        INSERT INTO MOBILE VALUES (R_mob.Mobile_No,ACC_ID);
                        COUNTER:=1;
            END IF;
            CLOSE C_mob;
END;
```