

OPIM 5272

**PHASE 3 – REPORT**

**PROCESS STUDIED**

**POINT Of SALES MAINTENANCE CALL MANAGEMENT PROCESS**

**COMPANY**

**ALHAMRANI UNIVERSAL COMPANY LIMITED**

**CUSTOMER**

**BANKS**

**BY,**

**TEAM 7:**

**MINGPEI YANG**

**MOHAMMED SHAKEB**

**NEHA TELAGUBANAJIGARA JAGANNATH**

**SHAIK JAVID**

**YUE JIAO**

**Table of Contents**

1. [Executive Summary 2](#_Toc468057104)
2. [As Is process 3](#_Toc468057105)
3. [To Be Process 4](#_Toc468057106)
4. [Conclusion 5](#_Toc468057107)
5. [ER Diagram 6](#_Toc468057108)
6. [Logical Database Design 7](#_Toc468057109)
7. [Tables 8](#_Toc468057111)
8. [Table Creation 10](#_Toc468057112)-11
9. [Reports 12](#_Toc468057113)-33
10. [Insights 34](#_Toc468057114)-35

# Executive Summary

**Business Need**: This project is initiated to improve the call management processes for Point of Sales (POS) calls reported by the major Banks. The project is expected to benefit the customers by reducing the call response time and improving the percentage of first time fix. The project will benefit the company by: reducing the cost of operations by reducing the number of field engineers needed and reducing the penalties for SLA breaches. It is expected to improve customer satisfaction which will result in increased business and market share for the company.

**Problem Statement:** During the past one year, an average of 5000 maintenance calls per month were reported, out of which 50-60% of them could be fixed by phone without the visit by the Service Engineer. Also 30% of these calls which needed terminal replacement, were fixed by multiple visits. There were few cases of wrong installations were reported which is very critical for the business. This resulted in delays, penalties and additional staff costing about 2-3 million dollars per year. This project identifies causes for these problems, provides solutions to fix them and improve the POS maintenance call management processes.

**Goal Statement:** To improve the customer experience by reducing the calls transferred to field services by 50% and increase the percentage of Service Level Agreement (SLA) compliance from 70% to 95%.

# As Is process

The following swim lane diagram depicts the As Is process that was studied. In the following process, the company was not meeting the SLAs with customers and the following problems were identified which needs to be rectified to meet the SLA requirements and reduce cost.

1. The process that is followed currently is the engineer will go to the site once the customer places the call and try to fix the problem. If he is required to replace the terminal he will go back to the warehouse to pick up the terminal, prepare it and go back to the site to fix it. This process was creating a delay in fixing the machine.
2. The call was directly transferred to the engineer without any filtering process. After analyzing the data from engineer’s updates on the tickets, we found that about 50% of calls does not require the site visit by the engineer and could be fixed over the phone.
3. Engineers were not aware of the time remaining for each ticket as resolution end times were not mentioned on tickets and some tickets were delayed beyond the resolution time.



# To Be Process

To fix the problems identified in the As Is process we have come up with a process change and plan to use the database to define the SLAs and update them on each ticket. The following actions were brainstormed to improve the business process.

1. We will create two new departments to minimize the call resolution time a) A First Line Maintenance (FLM) department will be created consisting of technical people who can identify the problem over phone and try to fix and filter out the calls b) A staging department will be created that will be within the warehouse to get the terminal from stock and prepare it as soon the FLM department is unable to fix the call over phone and requires as replacement terminal. This will ensure that the engineer will take a replacement terminal with him and fix the issue within the first visit without going back to merchant second time thereby targeting to fix it within the ticket resolution time.
2. We will define the SLAs in the database for each customer and while creating the tickets the resolution end time will be displayed for the engineer to prioritize the calls and ensure that the problem is fixed on time.

# Conclusion

As stated, the As Is process provided us with the information about the current process for POS call management at Alhamrani Universal Company. After collecting the data for the service orders handled by the engineers for past few months, we discovered that the company was not able to meet the customer SLA obligations on the resolution times. While analyzing the data on the engineer’s remarks on what was done to fix the problem, we discovered that about 50% of the calls did not require the visit of the engineer and could be fixed over the phone. Another reason for the SLA breach that was discovered was, the engineers were not aware of the ticket resolution time as it was not mentioned in the ticket. Additionally, there was excursion time involved due to the engineers going back to the warehouse to get the terminal in case a terminal replacement was required. We tried to find the solution considering these issues to improve the process.

After brainstorming and conducting a financial analysis we decided to create two new departments focused on filtering the calls and preparing the terminals. By this, we estimate to reduce the number of calls going to the service engineers by 50%. The remaining 50% calls that do not get fixed over the phone is estimated to get fixed in the first attempt by providing the engineer with a prepared terminal. Furthermore, we decided to enhance the database by adding the SLA resolution durations for each customer and imprinting the call’s resolution start time and end time on the ticket. This will allow the engineer to know the resolution end time for each of the ticket assigned to him and prioritize the tickets accordingly.

We expect SLA compliance to increase from the current 70% to about 95% by making the proposed changes to the process. Also, because of creating the FLM department the number of calls transferred to the engineers is estimated to get reduced by 50% which will reduce the number of engineers required for the field services. We will need to move a few engineers from the field to the FLM and Staging departments to manage these new services. Overall we expect to reduce the engineers by 30% of the current level. This will help the company to reduce the cost of operations as well.

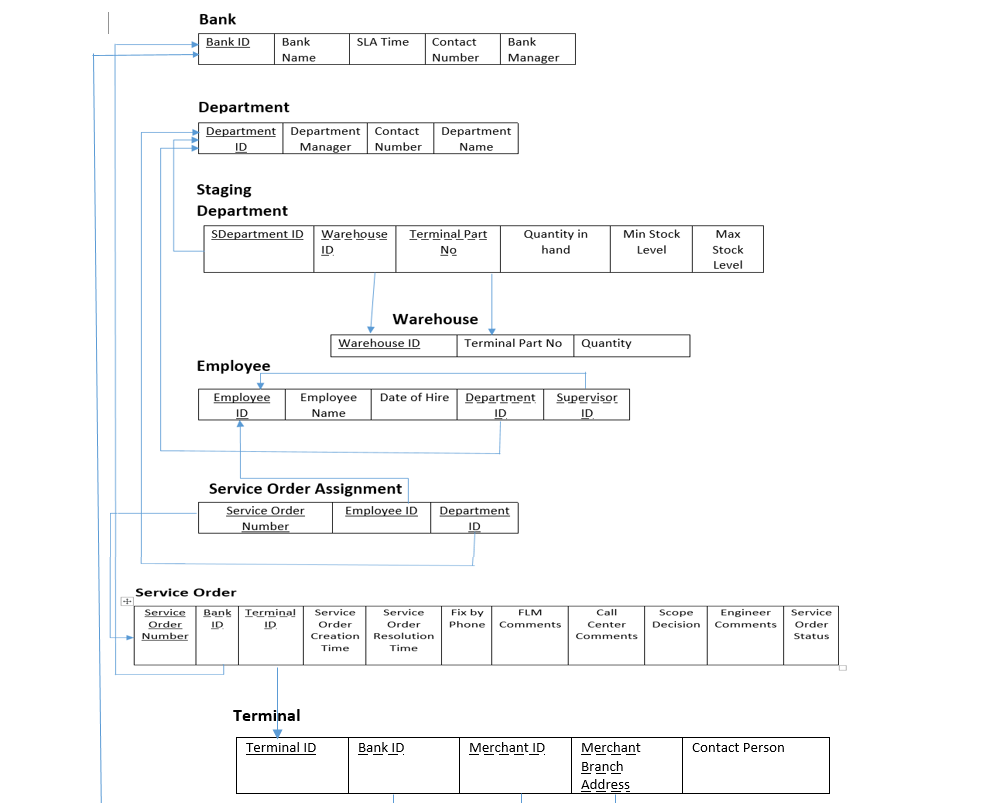
The customers, which in our case are the Banks will be the main beneficiaries as their merchants using the POS terminals will be served quickly, thereby increasing their merchant base and improving customer satisfaction. It will also help the company get new business by providing better services against its competitors. We expect the penalties to be reduced significantly thereby benefiting the company in multiple ways.

To ensure that we meet the expected results and process compliance, we must produce reports measuring the SLA performance for each customer, engineer and department; develop reports related to terminal stocks level in staging department for each terminal model; develop dash board reports for the management to visualize the level of customer satisfaction and the overall performance of the company, departments, and employees. The reports will ensure that the process is continuously monitored and refined based on the results.

# ER Diagram

After developing the Swimlane Diagram in Phase 1 of the project, the following EER diagram was developed identifying the entities and the relationships between them. 

# Logical Database Design

The below logical database design was developed referring to the ER diagram.

# 

# Tables

Note:

1. Attribute type: Single-valued attributes shown in **black** / multivalued attributes shown in **blue**)
2. Attribute data type: Shown in **red**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sl No** | **Table Name** | **Attributes** | **Attribute Type** | **Data Type** | **Attributes and Data types** |
| 1 | Customer Bank | Bank ID | Single-valued | NUMBER | Customer bank (bank\_ID [CHAR], bank\_Name [VARCHAR], sla\_time [NUMBER], bank\_contact [NUMBER], bank\_Manager\_Name [VARCHAR2]) |
|  |  | Bank Name | Single-valued | VARCHAR |
|  |  | SLA Time | Single-valued | DECIMAL |
|  |  | Contact Number | Single-valued | NUMBER |
|  |  | Bank Manager | Single-valued | VARCHAR |
| 2 | Department | Department ID | Single-valued | NUMBER | Department (dept\_ID [CHAR], dept\_Name [VARCHAR2], dept\_Manager\_Name [VARCHAR2], dept\_Contact [NUMBER], dept\_Cat [CHAR]) |
|  |  | Department Manager | Single-valued | VARCHAR |
|  |  | Contact Number | Single-valued | NUMBER |
|  |  | Department Name | Single-valued | VARCHAR |
|  |  | Department Category | Single-valued | CHAR |
| 3 | Employee | Employee ID | Single-valued | NUMBER | Employee (emp\_ID [NUMBER], dept\_ID [CHAR], emp\_FirstName [VARCHAR2] , emp\_LastName [VARCHAR2], date\_hire [DATE], , supervisor\_emp\_ID [NUMBER]) |
|  |  | Department ID | Single-valued | NUMBER |
|  |  | Employee Name | Multi-valued | VARCHAR |
|  |  | Date of Hire | Single-valued | DATE |
|  |  | Supervisor Employee ID | Single-valued | NUMBER |
| 4 | Staging Department | Department ID | Single-valued | NUMBER | Staging (dept\_ID [CHAR], whouse\_ID [CHAR] , terminal\_part\_num [CHAR], quantity\_InHand [NUMBER], min\_Stocklevel [NUMBER], max\_Stocklevel [NUMBER]) |
|  |  | Warehouse ID | Single-valued | NUMBER |
|  |  | Terminal Part No | Single-valued | VARCHAR |
|  |  | Quantity in Hand | Single-valued | NUMBER |
|  |  | Minimum stock level | Single-valued | NUMBER |
|  |  | Maximum stock level | Single-valued | NUMBER |
| 5 | Warehouse | Warehouse ID | Single-valued | NUMBER | Warehouse (whouse\_ID [CHAR], terminal\_part\_num [CHAR], quantity [NUMBER]) |
|  |  | Terminal Part No | Single-valued | VARCHAR |
|  |  | Quantity | Single-valued | NUMBER |
| 6 | Service Order Assignment | Service Order ID | Single-valued | NUMBER | Service Order Assignment (so\_num [CHAR], emp\_ID [NUMBER], dept\_ID [CHAR]) |
|  |  | Employee ID | Single-valued | NUMBER |
|  |  | Department ID | Single-valued | NUMBER |
| 7 | Service Order | Service Order ID | Single-valued | NUMBER | Service Order (so\_ID [CHAR], bank\_ID [CHAR], term\_ID [CHAR], so\_CreationTime [TIMESTAMP], so\_ResolutionTIme [TIMESTAMP], flm\_comments [VARCHAR2], cc\_comments [VARCHAR2], eng\_comments [VARCHAR2], so\_Status [VARCHAR2]) |
|  |  | Bank ID | Single-valued | NUMBER |
|  |  | Terminal ID | Single-valued | NUMBER |
|  |  | Service Order Creation Time | Single-valued | DATE TIME |
|  |  | Service Order Resolution Time | Single-valued | DATE TIME |
|  |  | FLM Comments | Single-valued | VARCHAR |
|  |  | Call Center Comments | Single-valued | VARCHAR |
|  |  | Engineer Comments | Single-valued | VARCHAR |
|  |  | Service Order Status | Single-valued | CHAR |
| 8 | Terminal | Terminal ID | Single-valued | NUMBER | Terminal (terminal\_ID [CHAR], bank\_ID [CHAR], mer\_ID [CHAR], mer\_br\_Add [VARCHAR2], contact\_per\_Name [VARCHAR2]) |
|  |  | Bank ID | Single-valued | NUMBER |
|  |  | Merchant ID | Single-valued | NUMBER |
|  |  | Merchant Branch Address | Single-valued | VARCHAR |
|  |  | Contact Person | Single-valued | VARCHAR |
| 9 | Merchant | Merchant ID | Single-valued | NUMBER | Merchant (mer\_ID [CHAR], mer\_br\_Add [VARCHAR2], mer\_name [VARCHAR], mer\_Contact [NUMBER]) |
|  |  | Merchant Branch Address | Single-valued | VARCHAR |
|  |  | Merchant Name | Single-valued | VARCHAR |
|  |  | Contact Number | Single-valued | NUMBER |

# Table Creation

The below tables were created in Oracle

Note: All the table names and constraint names are prefixed with T7 – to refer to our team number

|  |  |  |
| --- | --- | --- |
| Table Name | Columns and Data Types | Constraints |
| T7\_BANK |  |  |
| T7\_DEPARTMENT |  |  |
| T7\_EMPLOYEE |  |  |
| T7\_MERCHANT |  |  |
| T7\_TERMINAL |  |  |
| T7\_WAREHOUSE |  |  |
| T7\_STAGING\_DEP |  |  |
| T7\_SERVICE\_ORDER\_ASSIGNMENT |  |  |
| T7\_SERVICE\_ORDER |  |  |

# Reports

The below queries were written to get insights about the business before and after implementing the solution. Here, we assume that our solution was implemented on 16th November 2016 at 12 am.

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No | Query | Output | Comments |
| 1 | CREATE VIEW T7\_EmployeeView  AS SELECT employee\_id AS "Employee ID", ee\_last\_name AS "Last Name", date\_of\_hire AS "Hire Date"  FROM t7\_employee  WHERE department\_id = 'D2'  WITH CHECK OPTION CONSTRAINT depidD2\_ck; |  | A view gets created for the employee table with the selected variables. Any attempt to insert a duplicate with department\_id = ‘D2’ will throw an error due to the check constraint |
| 2 | DESCRIBE T7\_EmployeeView; |  | Provides the description of the view created |
| 3 | select bank\_id AS "Bank Id", terminal\_id AS "Terminal ID", count(service\_order\_id) AS "No. of Tickets"  from t7\_service\_order  group by bank\_id, terminal\_id  order by count(service\_order\_id) desc; |  | Provides the details about the number of tickets created for each terminal in a duration of 1 month |
| 4 | select bank\_id AS "Bank Id", count(service\_order\_id) AS "No. of Tickets"  from t7\_service\_order  group by bank\_id  order by count(service\_order\_id) desc; |  | Provides the details about the number of tickets that were opened for each bank |
| 5 | select distinct terminal\_id as "Terminal ID",  count(service\_order\_id) As "No.of Tickets",  sum(case when service\_order\_status = 'Open' then 1 else 0 end) as "No. of open Tickets",  sum(case when service\_order\_status = 'Closed' then 1 else 0 end) as "No. of closed Tickets"  from t7\_service\_order  group by terminal\_id  order by sum(case when service\_order\_status = 'Open' then 1 else 0 end) desc; |  | Provides information about the total number of tickets which are in the status “open” or “close” for each terminal |
| 6 | select b.BANK\_NAME,  count(so.service\_order\_id) As "No.of Tickets",  sum(case when so.service\_order\_status = 'Open' then 1 else 0 end) as "No. of open Tickets",  sum(case when so.service\_order\_status = 'Closed' then 1 else 0 end) as "No. of closed Tickets"  from t7\_service\_order so  join t7\_bank b  using (bank\_id)  group by b.bank\_name; |  | Provides information about the total number of tickets which are in the status “open” or “close” for each bank |
| 7 | select bank\_name as "Bank Name", sla\_time AS "SLA Time"  from t7\_bank  group by bank\_name, sla\_time  order by sla\_time; |  | Provides information on the SLA time for each bank |
| 8 | select bank\_name as "Bank with least SLA", sla\_time AS "SLA time in hours"  from t7\_bank  group by bank\_name, sla\_time  HAVING min(sla\_time) = (select distinct min(sla\_time) from t7\_bank)  order by sla\_time; |  | Here we get the information about the bank with the least SLA time. ANB has the least SLA which means its tickets will be prioritized compared to tickets created for other banks at the same time |
| 9 | select service\_order\_id as "Service Order ID", NVL(engineer\_comments,'Comment not updated by Engineer') AS "Engineer Comments", SERVICE\_ORDER\_STATUS AS "Service Order Status"  from t7\_service\_order; |  | Here the NVL command is used to replace null values with the comment “Comment not updated”. From the result it can be inferred that the tickets in “Open” status usually have a null value for comments |
| 10 | select bank\_id as "Bank ID", 'Service Order '||service\_order\_id||'was created for this bank on '||service\_order\_creation\_time||' and was resolved on '||service\_order\_resolution\_time||' in a duration of '||round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2)||' hours' AS "Order Summary Before 16th"  from t7\_service\_order  where service\_order\_resolution\_time < to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS'); |  | Provides a brief history of the service order before the solution was implemented |
| 11 | select bank\_id as "Bank ID", 'Service Order '||service\_order\_id||'was created for this bank on '||service\_order\_creation\_time||' and was resolved on '||service\_order\_resolution\_time||' in a duration of '||round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2)||' hours' AS "Order Summary On/After 16th"  from t7\_service\_order  where service\_order\_resolution\_time >= to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS'); |  | Provides a brief history of the service order after the solution was implemented |
| 12 | CREATE synonym t7\_soa for t7\_service\_order\_assignment; |  | Creates a synonym for the table object which can be used later interchangeably with the existing table name |
| 13 | select t7\_soa.service\_order\_id, rtrim(emp.ee\_first\_name)||' '||rtrim(emp.ee\_last\_name) as "Engineer Assigned", emp.department\_id, dep.department\_name  from t7\_soa  JOIN t7\_employee emp  USING (employee\_id)  JOIN t7\_department dep  on emp.department\_id = dep.department\_id; |  | Shows the service order assignment of each employee |
| 14 | alter table t7\_staging\_dep  RENAME COLUMN mini\_stock to min\_stock; |  | Alters the name of the column mini\_stock to min\_stock |
| 15 | **NOTE: The below query must be executed after the query above as the column name (min\_stock) was changed**  select sd.warehouse\_id as "Warehouse ID", sd.terminal\_part\_no as "Terminal Part No.", sd.quantity\_in\_hand as "Total In Stock", sd.min\_stock as "Minimum Required", sd.max\_stock as "Maximum Required",  (case when sd.quantity\_in\_hand < sd.min\_stock then (sd.max\_stock - sd.quantity\_in\_hand) else 0 end) AS "Reorder Quantity", w.quantity as "Quantity in Main Warehouse"  from t7\_staging\_dep sd  join t7\_warehouse w  on sd.terminal\_part\_no = w.terminal\_part\_no; |  | Provides details about the terminal part numbers which require a re-order and the re-order quantity suggested by the system |
| 16 | select so.service\_order\_id as "Service Order", sa.employee\_id as "employee\_id",round(((extract(day from so.service\_order\_resolution\_time)\*24+extract(hour FROM so.service\_order\_resolution\_time)+  extract(minute FROM so.service\_order\_resolution\_time)/60+extract(second FROM so.service\_order\_resolution\_time)/3600)  -(extract(day from so.service\_order\_creation\_time)\*24+extract(hour FROM so.service\_order\_creation\_time)+  extract(minute FROM so.service\_order\_creation\_time)/60+extract(second FROM so.service\_order\_creation\_time)/3600)),2) AS "Resolution time in hours",b.sla\_time as "Agreed SLA",  (case when round(((extract(day from so.service\_order\_resolution\_time)\*24+extract(hour FROM so.service\_order\_resolution\_time)+  extract(minute FROM so.service\_order\_resolution\_time)/60+extract(second FROM so.service\_order\_resolution\_time)/3600)  -(extract(day from so.service\_order\_creation\_time)\*24+extract(hour FROM so.service\_order\_creation\_time)+  extract(minute FROM so.service\_order\_creation\_time)/60+extract(second FROM so.service\_order\_creation\_time)/3600)),2) <= b.sla\_time then 'Yes' else 'No' end) as "Is\_within\_SLA",  (case when so.service\_order\_resolution\_time < to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS') then 'Before Solution' else 'After Solution' end) as "Before/After Implementation"  from t7\_service\_order\_assignment sa  JOIN t7\_service\_order so  on sa.SERVICE\_ORDER\_ID = so.SERVICE\_ORDER\_ID  join t7\_bank b  on b.bank\_id = so.BANK\_ID  where so.service\_order\_status = 'Closed'  order by 6 desc; |  | Provides details about the order resolution time of engineers |
| 17 | select so.bank\_id, so.service\_order\_id, round(((extract(day from so.service\_order\_resolution\_time)\*24+extract(hour FROM so.service\_order\_resolution\_time)+  extract(minute FROM so.service\_order\_resolution\_time)/60+extract(second FROM so.service\_order\_resolution\_time)/3600)  -(extract(day from so.service\_order\_creation\_time)\*24+extract(hour FROM so.service\_order\_creation\_time)+  extract(minute FROM so.service\_order\_creation\_time)/60+extract(second FROM so.service\_order\_creation\_time)/3600)),2) AS "Resolution Time in Hours", b.sla\_time AS "Bank's SLA", so.engineer\_comments AS "Engineer Comments"  FROM t7\_service\_order so  JOIN t7\_bank b  on b.bank\_id = so.bank\_id  where so.service\_order\_resolution\_time < to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS')  AND round(((extract(day from so.service\_order\_resolution\_time)\*24+extract(hour FROM so.service\_order\_resolution\_time)+extract(minute FROM so.service\_order\_resolution\_time)/60+extract(second FROM so.service\_order\_resolution\_time)/3600)-(extract(day from so.service\_order\_resolution\_time)\*24+extract(hour FROM so.service\_order\_creation\_time)+extract(minute FROM so.service\_order\_creation\_time)/60+extract(second FROM so.service\_order\_creation\_time)/3600)),2) > b.sla\_time  AND so.service\_order\_status = 'Closed'; |  | Provides information about the orders which did not meet the SLA time before 16th and the Engineer’s commence for the same. It can be observed that most of the tickets required a reset of the terminal only and did not require the engineer to visit the site |
| 18 | select count(engineer\_comments) AS "No. of calls ", engineer\_comments AS "Engineer Comments"  from t7\_service\_order  where service\_order\_resolution\_time < to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS')  and service\_order\_status = 'Closed'  and ENGINEER\_COMMENTS IS NOT NULL  group by engineer\_comments  order by 1 desc; |  | Provides information about the number of calls that could be fixed by phone. Based on the result most of them just needed a terminal reset and could be fixed by phone |
| 19 | select count(flm\_comments) AS "No. of calls fixed by phone", flm\_comments AS "FLM Comments"  from t7\_service\_order  where flm\_comments = UPPER('Fixed through Phone')  group by flm\_comments; |  | Provides information about the number of calls fixed by first line maintenance on/after 16th |
| 20 | select so.service\_order\_id, so.BANK\_ID,  round(((extract(day from so.service\_order\_resolution\_time)\*24+extract(hour FROM so.service\_order\_resolution\_time)+  extract(minute FROM so.service\_order\_resolution\_time)/60+extract(second FROM so.service\_order\_resolution\_time)/3600)  -(extract(day from so.service\_order\_creation\_time)\*24+extract(hour FROM so.service\_order\_creation\_time)+  extract(minute FROM so.service\_order\_creation\_time)/60+extract(second FROM so.service\_order\_creation\_time)/3600)),2) AS "Issue resolution time",  b.SLA\_TIME,  (case when round(((extract(day from so.service\_order\_resolution\_time)\*24+extract(hour FROM so.service\_order\_resolution\_time)+  extract(minute FROM so.service\_order\_resolution\_time)/60+extract(second FROM so.service\_order\_resolution\_time)/3600)  -(extract(day from so.service\_order\_creation\_time)\*24+extract(hour FROM so.service\_order\_creation\_time)+  extract(minute FROM so.service\_order\_creation\_time)/60+extract(second FROM so.service\_order\_creation\_time)/3600)),2) <= b.sla\_time then 'Yes' else 'No' end) as "Is\_within\_SLA",  (case when so.service\_order\_resolution\_time < to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS') then 'Before Solution' else 'After Solution' end) as "Before/After Implementation"  FROM t7\_service\_order so join t7\_bank b on so.BANK\_ID = b.BANK\_ID  where so.SERVICE\_ORDER\_STATUS = 'Closed'  order by so.bank\_id; |  | Provides a history of the resolution time and sla compliance for each of the tickets created and closed |
| 21 | select t7\_bank.BANK\_NAME,  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 1 end) as "Total no. of calls",  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 0 end) as "Calls fixed within SLA",  round(((  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 0 end)  )/(  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 1 end)  ))\*100,2) as "%SLA Compliance before sol"  FROM t7\_service\_order join t7\_bank on t7\_service\_order.BANK\_ID = t7\_bank.BANK\_ID  where t7\_service\_order.SERVICE\_ORDER\_STATUS = 'Closed' and service\_order\_resolution\_time < to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS')  group by t7\_bank.BANK\_name  order by t7\_bank.BANK\_name; |  | Provides details about the % of sla compliance of each bank before process reengineering i.e. before 16th NOV |
| 22 | select t7\_bank.BANK\_NAME,  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 1 end) as "Total no. of calls",  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 0 end) as "Calls fixed within SLA",  round(((  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 0 end)  )/(  sum(case when round(((extract(day from service\_order\_resolution\_time)\*24+extract(hour FROM service\_order\_resolution\_time)+  extract(minute FROM service\_order\_resolution\_time)/60+extract(second FROM service\_order\_resolution\_time)/3600)  -(extract(day from service\_order\_creation\_time)\*24+extract(hour FROM service\_order\_creation\_time)+  extract(minute FROM service\_order\_creation\_time)/60+extract(second FROM service\_order\_creation\_time)/3600)),2) <= t7\_bank.sla\_time then 1 else 1 end)  ))\*100,2) as "%SLA Compliance after sol"  FROM t7\_service\_order join t7\_bank on t7\_service\_order.BANK\_ID = t7\_bank.BANK\_ID  where t7\_service\_order.SERVICE\_ORDER\_STATUS = 'Closed' and service\_order\_resolution\_time >= to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS')  group by t7\_bank.BANK\_name  order by t7\_bank.BANK\_name; |  | Provides details about % sla compliance of each bank after process reengineering i.e. on and after 16th NOV |
| 23 | select count(service\_order\_id) as "Total calls fixed after sol",  sum(case when flm\_comments = Upper('fixed through phone') then 1 else 0 end) as "Calls fixed by phone",  round(((sum(case when flm\_comments = Upper('fixed through phone') then 1 else 0 end)) / count(service\_order\_id))\*100,2) as "%Calls fixed by FLM"  from t7\_service\_order  where service\_order\_status = 'Closed' and service\_order\_resolution\_time >= to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS'); |  | Provides information about the % of calls fixed by first line maintenance after process reengineering |
| 24 | select count(service\_order\_id) as "Total calls fixed before sol",  sum(case when engineer\_comments = Upper('reset terminal and tested ok') then 1 else 0 end) as "Calls fixed by terminal RESET",  round(((sum(case when engineer\_comments = Upper('reset terminal and tested ok') then 1 else 0 end)) / count(service\_order\_id))\*100,2) as "%Calls fixed by RESET"  from t7\_service\_order  where service\_order\_status = 'Closed' and service\_order\_resolution\_time < to\_timestamp('16-NOV-16 12.00.00', 'DD-MON-YY HH12.MI.SS'); |  | Provides information about % calls fixed by terminal reset (before process reengineering) which could be fixed on call |

# Insights

**SLA Compliance for each bank before and after process reengineering**

From the two visualizations above it can be inferred that there has been an improvement in the SLA compliance of all the banks by a substantial amount.

From the visualization above, it can be inferred that, before reengineering, 57% of issues, were fixed by an engineer visiting the site to do a terminal reset. After reengineering the same issue, could be fixed by engineers in the First Level Maintenance over call, thereby eliminating the excursion time (time taken to travel to the site) involved.

Since, more than 50% of the calls could be fixed by phone, the number of engineers required can be reduced thereby reducing operational costs.The process reengineering will help improve customer(bank) satisfaction in terms of improvement in the SLA compliance thereby increasing the company’s profit.