

Contents

1. Introduction
2. What is a Call Center?
3. This Call Center
4. Key Performance Indicator
5. Raw Data
6. KPIs of this business
7. ETL
8. DW Architecture
9. Schema
10. Dimensions and Facts
11. Dashboard
12. References

Introduction

- This is the documentation of Business Intelligence Call Center Dashboard based on Kimall's Data warehouse standard as a final Project .
- This project used both Oracle and Microsoft tools.

What is a Call Center?

- People call to Call Centers to give services.
- An inbound Call Center, have a group of operators to answer to the clients and make it document.
- The document contains details of the call.
- But what are that details?
- It directly depends on the title of the business of the call Center.



This Call Center

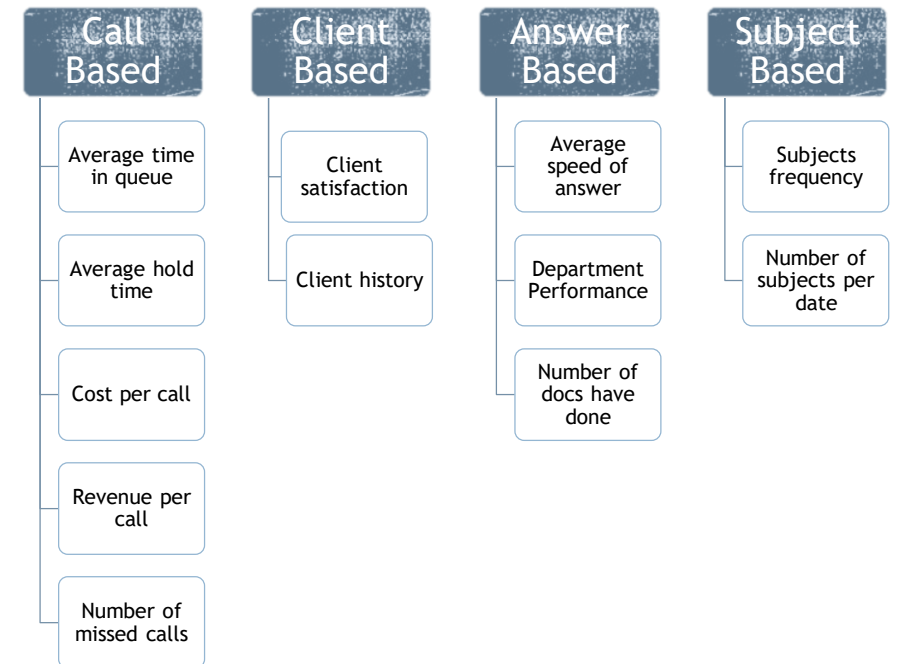
It is a Municipality Call Center, which gives information about city services and then, report it to relevant departments to handle it.



Key Performance Indicator

KPI is a measurable value that demonstrates how effectively a company is achieving key business objectives. Organizations use KPIs to evaluate success at reaching targets.

There are a lot of insight to look at a business.
But **Data** is determinative.



Raw Data

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Def. On Null	Histogram	Num Distinct	Num Nulls
CASE_ID	1	1	1	N	INTEGER		<input type="checkbox"/>	None		
SOURCE	2			Y	VARCHAR2 (100 Char)		<input type="checkbox"/>	None		
DEPARTMENT	3			Y	VARCHAR2 (100 Char)		<input type="checkbox"/>	None		
WORK_GROUP	4			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
REQUEST_TYPE	5			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
CATEGORY	6			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
TYPE	7			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
DETAIL	8			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
CREATION_DATE	9			Y	DATE		<input type="checkbox"/>	None		
CREATION_MONTH	10			Y	INTEGER		<input type="checkbox"/>	None		
CREATION_YEAR	11			Y	INTEGER		<input type="checkbox"/>	None		
STATUS	12			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
EXCEEDED_EST_TIMEFRAME	13			Y	VARCHAR2 (50 Char)		<input type="checkbox"/>	None		
CLOSED_DATE	14			Y	DATE		<input type="checkbox"/>	None		
CLOSED_MONTH	15			Y	INTEGER		<input type="checkbox"/>	None		
CLOSED_YEAR	16			Y	INTEGER		<input type="checkbox"/>	None		
DAYS_TO_CLOSE	17			Y	INTEGER		<input type="checkbox"/>	None		
STREET_ADDRESS	18			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
ADDRESS_WITH_GEOCODE	19			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
ZIP_CODE	20			Y	INTEGER		<input type="checkbox"/>	None		
NEIGHBORHOOD	21			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		
COUNCIL_DISTRICT	22			Y	INTEGER		<input type="checkbox"/>	None		
PARCEL_ID_NO	23			Y	INTEGER		<input type="checkbox"/>	None		
XCOORDINATE	24			Y	NUMBER		<input type="checkbox"/>	None		
YCOORDINATE	25			Y	NUMBER		<input type="checkbox"/>	None		
CASE_URL	26			Y	VARCHAR2 (500 Char)		<input type="checkbox"/>	None		

ETL

As the KPIs come from raw data, we use one or more than one of columns to calculate the KPIs.

ETL cleans raw data and push it in Data warehouse.



Cleaning Data

There is some cells is in this data set without any value. At first we need to find them, then we can fill them with a specific value. Such as unknown and 0.

Because null values will have bad effects on schema structure or KPIs calculation.

```
• select * from cc_raw_data  
  where days_to_close is null;
```

shows if there is any null value in days_to_close column.

```
• update cc_raw_data set(days_to_close) = 00  
  where days_to_close is null;
```

will fill null values in days_to_close column with 00.

After using it for all columns, we will clarify data.

DW Architecture

We need to organize raw data in a Data Warehouse.

Check the requirements to find out how we can categorize data, which column should be used and etc.

After all we can manage this business by this categorization:

What do people need (Request), type of requests, which department is responsible for, the location of reports and the time.



KPIs of this business

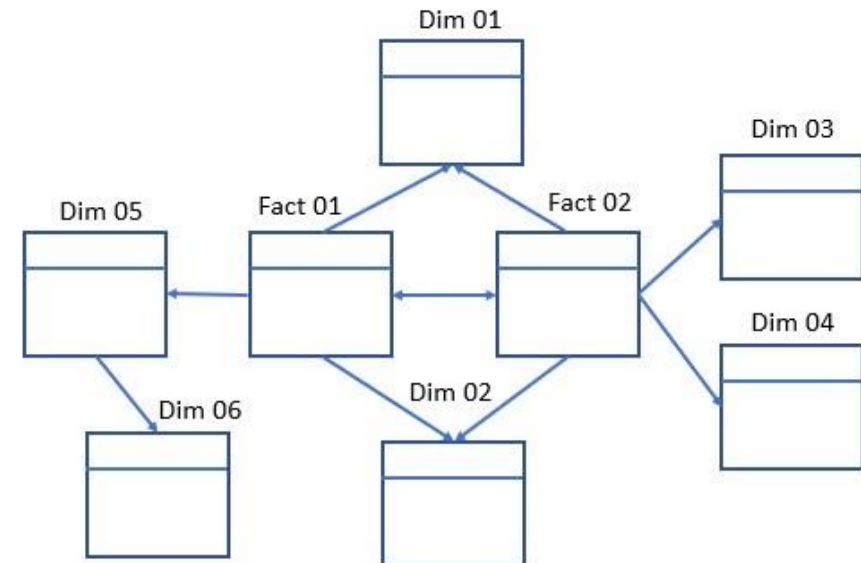
Model	Dimensions	Formula	Measure	
Fact 1	Dim location - customer	Count	Number of calls per location	1
Fact 1	Dim department - customer	Count	Number of calls per department	2
Fact 1	Dim customer - source	Count	Number of calls per source	3
Fact 1	Dim customer - request	Count	Number of calls per request	4
Fact 1	Dim customer - type	Count	Number of calls per type	5
Fact 1	Dim customer - status	Count	Number of calls per status	6
Fact 1	Dim customer - time	Count	Number of calls per date	7
Fact 2	Dim case - location	Sum duration / count duration	Avg of response time per location	8
Fact 2	Dim case - department	Sum duration / count duration	Avg of response time per department	9
Fact 2	Dim case - type	Sum duration / count duration	Avg of response time per type	10
Fact 2	Dim case - request	Sum duration / count duration	Avg of response time per request	11

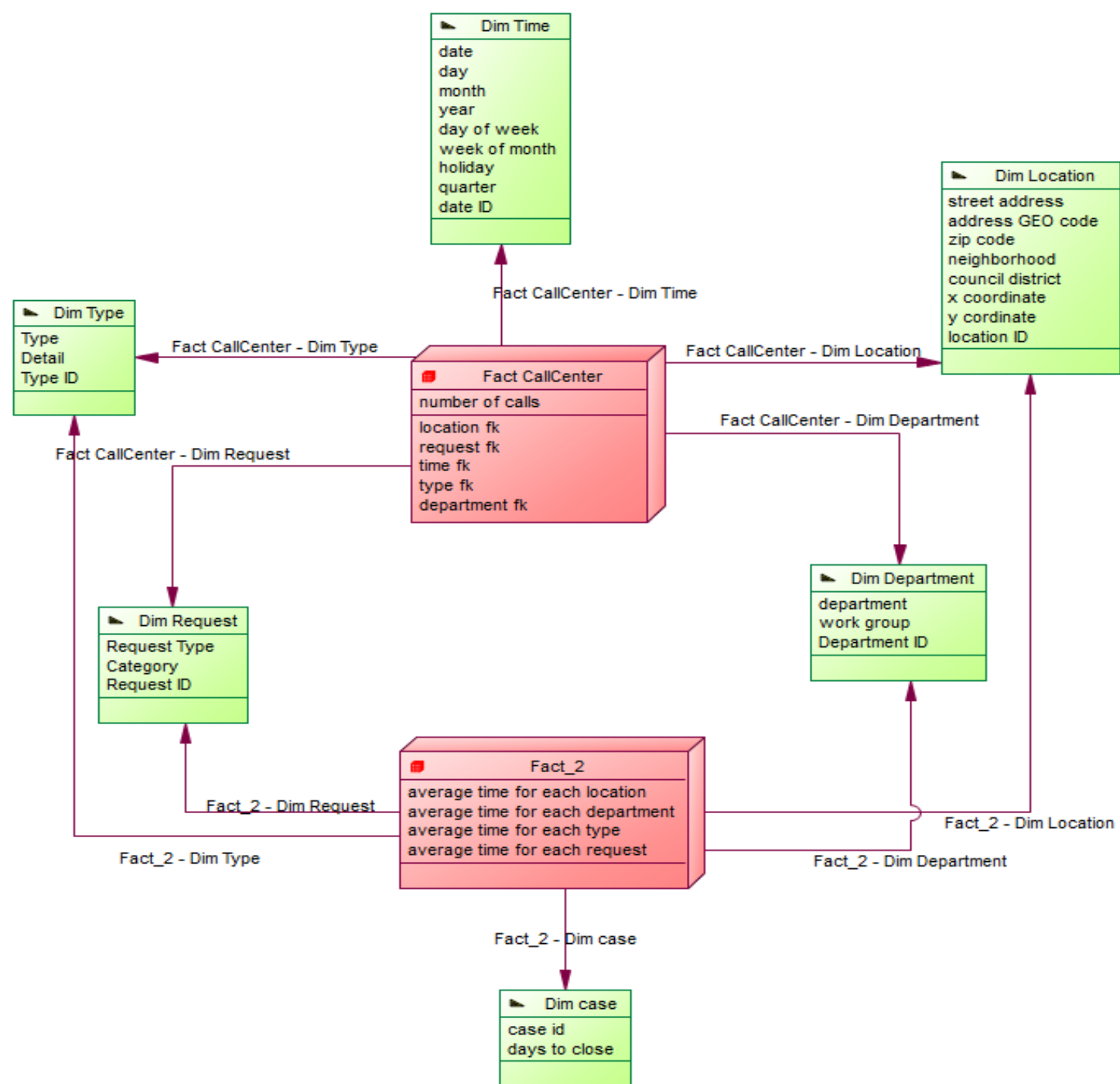
Schema

According to table page 11, we have 2 feature type (Fact tables), it means a multidimensional model works.

As measurements share dimensions, the **Star Schema** is the best Data warehouse architecture.

Schema Contains 2 Fact tables and 6 Dimension tables.





Dimension Table and Fact Table

Dimension Tables include features. Actual data transforms here. It may have more details or a hierarchical structure.

All dimensions have foreign keys which connect dimension to fact table.

Fact Tables include measurements and dimension's foreign key as primary key.



Creation of dimensions

Here data transforms directly to data warehouse.

Other dimensions creation is the same.

```
15
16 • create table CC_dim_request as(
17     select
18         request_type as request_type,
19         category as category,
20         rownum id
21     from
22         (select REQUEST_TYPE,
23             CATEGORY
24         from CC_raw_data
25         group by REQUEST_TYPE,CATEGORY)
26 );
27 • alter table CC_dim_request
28     ADD CONSTRAINT PK_request PRIMARY KEY (id);
29
```

Date dimension

Time dimension structure is different. We don't make it from creation date or closed date columns. We make a base, including day of week, day type, name of month, number of that and etc.

DATE_D	DATE_KEY	YEAR_NUM	QUARTER_NUM	MONTH_NUM	MONTH_DESC	DAY_OF_MONTH_NUM	DAY_OF_WEEK_DESC	DAY_TYPE_DESC
--------	----------	----------	-------------	-----------	------------	------------------	------------------	---------------

SQL query:

```
82 • CREATE TABLE cc_dim_date
83   as(
84     select
85       DATE_D,
86       to_number(TO_CHAR(DATE_D,'YYYYMMDD', 'nls_calendar=persian')) as DATE_key,
87       TO_NUMBER(TO_CHAR(DATE_D,'YYYY')) as YEAR_NUM,
88       TO_NUMBER(TO_CHAR(DATE_D,'Q')) as QUARTER_NUM,
89       TO_NUMBER(TO_CHAR(DATE_D,'MM')) as MONTH_NUM,
90       TRIM(TO_CHAR(DATE_D,'Month','nls_date_language=english')) as MONTH_DESC,
91       TO_NUMBER(TO_CHAR(DATE_D,'DD')) as DAY_OF_MONTH_NUM,
92       TRIM(TO_CHAR(DATE_D,'Day','nls_date_language=english')) as DAY_OF_WEEK_DESC,
93       (CASE WHEN TRIM(TO_CHAR(DATE_D,'Day','nls_date_language=english')) IN ('thursday','friday') THEN 'Weekend' ELSE 'Weekday' END) as DAY_TYPE_DESC
94   )
95   from(
96     select
97       to_date('13110123','YYYYMMDD', 'nls_calendar=persian')+ROWNUM as DATE_D
98     from
99       dual
100    connect by level <= sysdate-to_date('19320413','YYYYMMDD')
101  );
102 • ALTER TABLE cc_dim_date
103   ADD CONSTRAINT date_pk PRIMARY KEY (date_key);
104
```

Fact Tables creation

```
105 • create table fact_callcenter(  
106     call_num number not NULL,  
107     location_fk int not NULL,  
108     request_fk int not NULL,  
109     date_fk number not NULL,  
110     type_fk int not NULL,  
111     department_fk int not NULL,  
112  
113     CONSTRAINT FK_loc FOREIGN KEY (location_fk)  
114         REFERENCES dim_calllocation(id),  
115     CONSTRAINT FK_req FOREIGN KEY (request_fk)  
116         REFERENCES dim_request(id),  
117     CONSTRAINT FK_date FOREIGN KEY (date_fk)  
118         REFERENCES dim_date(date_key),  
119     CONSTRAINT FK_type FOREIGN KEY (type_fk)  
120         REFERENCES dim_Type(id),  
121     CONSTRAINT FK_dep FOREIGN KEY (department_fk)  
122         REFERENCES dim_department(id)  
123 );
```

```
11 ► insert into cc_fact_callcenter (  
12     call_num,  
13     location_fk,  
14     request_fk,  
15     date_fk,  
16     type_fk,  
17     department_fk)  
18  
19     with  
20     base as (  
21         select  
22             count(case_id) num,  
23             TYPE, DETAIL,  
24             DEPARTMENT, WORK_GROUP,  
25             to_number(to_char(CREATION_DATE, 'yyyymmdd', 'nls_calendar=persian')) as date_id,  
26             STREET_ADDRESS, ZIP_CODE,  
27             REQUEST_TYPE, CATEGORY  
28         from  
29             cc_raw_data  
30         group by  
31             TYPE, DETAIL,  
32             DEPARTMENT, WORK_GROUP,  
33             to_number(to_char(CREATION_DATE, 'yyyymmdd', 'nls_calendar=persian')),  
34             STREET_ADDRESS, ZIP_CODE,  
35             REQUEST_TYPE, CATEGORY  
36     ),  
37     request as(  
38         select  
39             t1.num,  
40             t1.TYPE, t1.DETAIL,  
41             t1.DEPARTMENT, t1.WORK_GROUP,  
42             t1.date_id,
```



```

45     from
46         base t1 join cc_dim_request t2
47     on (t1.REQUEST_TYPE = t2.REQUEST_TYPE
48         and
49         t1.CATEGORY = t2.CATEGORY)
50     ),
51     type as(
52     select
53         t1.num,
54         t1.DEPARTMENT, t1.WORK_GROUP,
55         t1.date_id,
56         t1.STREET_ADDRESS, t1.ZIP_CODE,
57         t1.req_id,
58         t2.ID as typ_id
59     from
60         request t1 join cc_dim_type t2
61     on (t1.TYPE = t2.type
62         and
63         t1.DETAIL = t2.DETAIL)
64     ),
65     department as(
66     select
67         t1.num,
68         t1.date_id,
69         t1.STREET_ADDRESS, t1.ZIP_CODE,
70         t1.req_id,
71         t1.typ_id,
72         t2.ID as dep_id
73     from
74         type t1 join cc_dim_department t2
75     on (t1.DEPARTMENT = t2.DEPARTMENT
76         and
77         t1.WORK_GROUP = t2.WORK_GROUP)
78     ),

```

```

79     location as(
80     select
81         t1.num num,
82         t1.date_id date_id,
83         t1.req_id req_id,
84         t1.typ_id typ_id,
85         t1.dep_id dep_id,
86         t2.ID as loc_id
87     from
88         department t1 join cc_dim_calllocation t2
89     on (t2.STREET_ADDRESS = t1.STREET_ADDRESS
90         and
91         t2.ZIP_CODE = t1.ZIP_CODE)
92     )
93     select
94         num,
95         loc_id,
96         req_id,
97         date_id,
98         typ_id,
99         dep_id
100 from
101     location
102 ;
103

```

Visualization

All data will appear in charts. Data warehouse make connection with data Marts and data mart gives them to visualization layer.

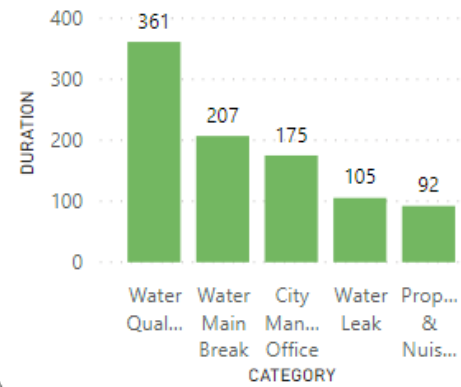
The data which is require for business management, comes in Dashboard.

The Dashboard helps management team to easily understand data and history of that.

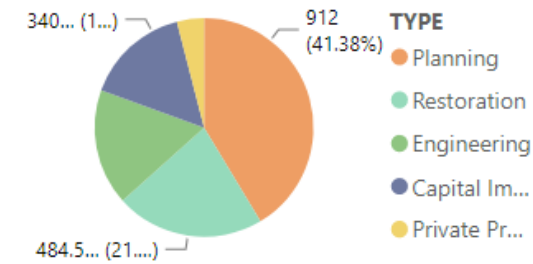


Duration of Calls

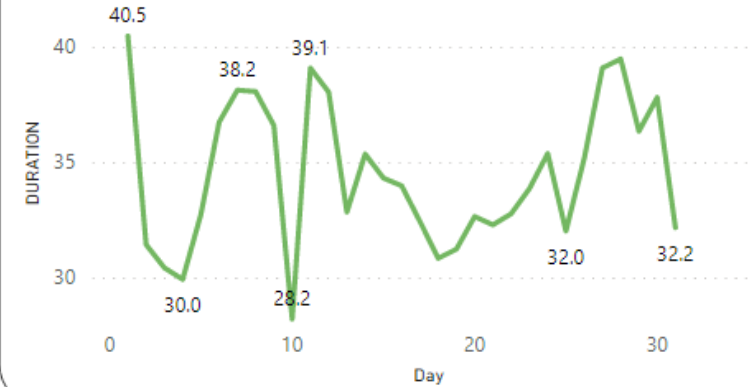
Duration by Request



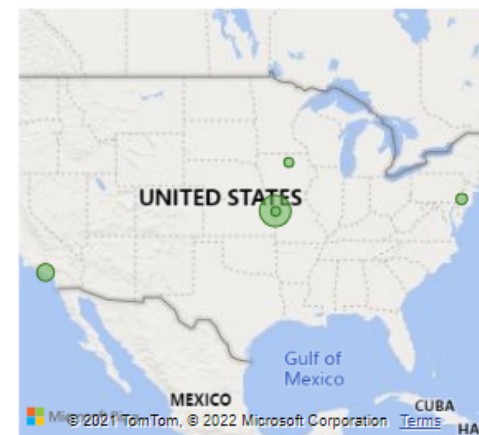
Duration by TYPE



Duration by Day



Duration by Address



Duration by DEPARTMENT





Call Center Performance

Total Call Quantity

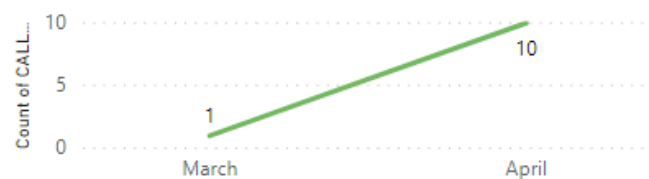
22.00

CALL_NUM

Y...

- ☐ 2,000.00
- ☐ 2,001.00
- ☐ 2,002.00
- ☐ 2,003.00
- ☐ 2,004.00
- ☐ 2,005.00

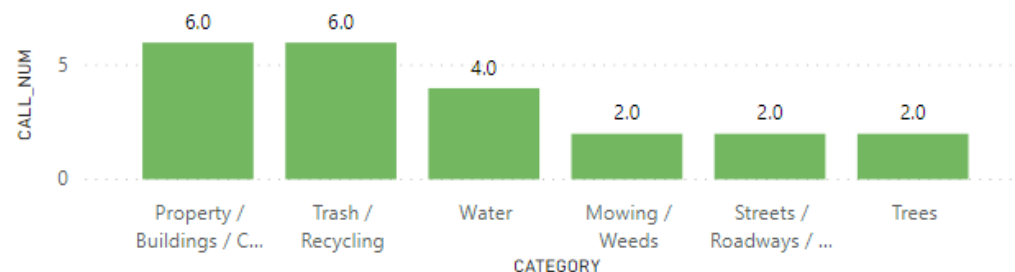
Quantity by Time



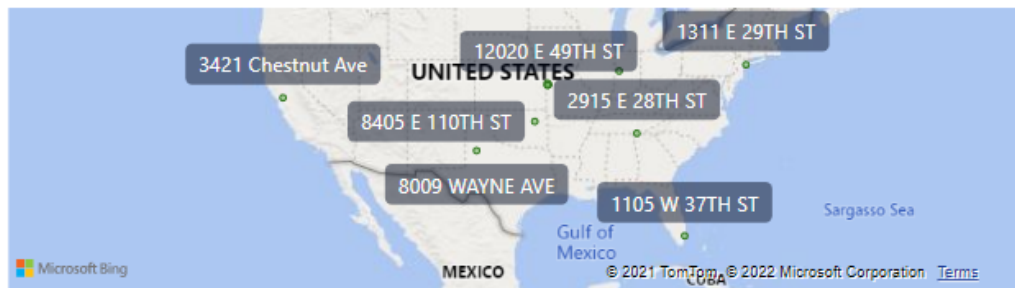
Quantity by Department



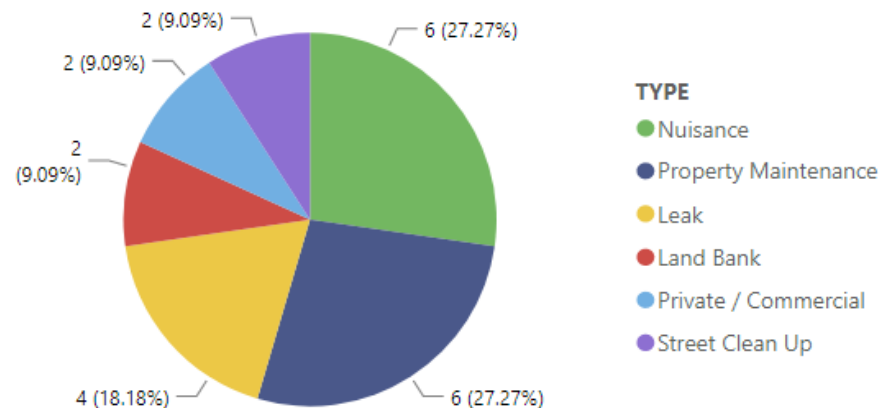
Quantity by Request



Quantity by Location



Quantity by Type



References

1- Oracle 19C database service

Toad for Oracle, as Database developer tool.

PL SQL as querying language in DMS.

2- Designed by Power Designer, Architecture visualization tool.

3- Designed by Power BI, Microsoft Data visualization tool.



Thanks for your time and attention

Connect me with:
Sahar.hrti@gmail.com