Detailed Design

For Red Turf

Version 1.3

Tanvi IT Solutions Inc

4211 Pleasant Valley Rd,

Suite 230

Chantilly

VA-20151

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version (x.y) | Date of Revision | Description of Change | Reason for Change | Affected Sections | Approved By |
| 1.1 | 1/10/18 | Baseline Version |  |  | Nagoor Inaganti |
| 1.2 | 8/10/18 | Addition of modules | Requirement Development | Module Architecture | Nagoor Inaganti |
| 1.3 | 15/10/18 | Change in the design approach | Integration of modules | Design Approach | Nagoor Inaganti |

**Approval History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version (x.y) | Prepared By | Reviewed By/Date | Approved By/Date |
| 1.1 | Vijaya Somarepetta | Monalisha Mishra  1/10/18 | Nagoor Inaganti  1/10/18 |
| 1.2 | Vijaya Somarepetta | Monalisha Mishra  8/10/18 | Nagoor Inaganti  8/10/18 |
| 1.3 | Vijaya Somarepetta | Monalisha Mishra  15/10/18 | Nagoor Inaganti  15/10/18 |

Table of Contents

[1 Overview 4](#_Toc382397411)

[1.1 Scope of this Document 4](#_Toc382397412)

[1.3 Platform and Tools 4](#_Toc382397414)

[1.4 References 4](#_Toc382397415)

[1.5 Assumptions 4](#_Toc382397416)

[1.6 Design Constraints 4](#_Toc382397417)

[2 Design Approach/ System Overview 5](#_Toc382397418)

[3 Alternative Design Approaches Considered 5](#_Toc382397419)

[4 Module Design 5](#_Toc382397420)

[4.2 Class Diagram 5](#_Toc382397422)

[4.2.1 Class 1 6](#_Toc382397423)

[4.2.2 Class 2 6](#_Toc382397424)

[4.2.3 Class 3 7](#_Toc382397425)

[4.3 Database Design 7](#_Toc382397426)

[4.3.1 Table 1 7](#_Toc382397427)

[4.3.2 Table 2 7](#_Toc382397428)

[4.4 Interaction Diagram 7](#_Toc382397429)

[4.5 State Transition Diagram (Optional) 7](#_Toc382397430)

[4.5.1 State Transition Table 8](#_Toc382397431)

[4.6 Error & Exception Handling 8](#_Toc382397432)

[5 Unit Test Recommendations 8](#_Toc382397433)

[6 Traceability Matrix 8](#_Toc382397434)

# Overview

Red Turf project is an Enterprise Business Intelligence product. The project is targeted to be used as a part of Campaign and Constituent Management services. Red Turf is a cloud based integrated online system which can be accessed from any location from devices like Laptops, tablets and/ or on smart phones. The project is focused on alleviating issues of data integrity and technology to streamline the process. The Red Turf system performs elastic search along with various social media integration without any security breach.

## Scope of this Document

The scope of the document is to perform the high-level design by identifying the different modules and their interfaces. The modules in the Red Turf project include Search, Dashboard, Resident, Task, List, Mapping and Electronic Check-in.

## Platform and Tools

The following are the tools that are being used and the platform on which the system is being designed for

|  |  |
| --- | --- |
| Tool Name | Purpose for which the Tool is proposed to be used |
| MVC framework .NET | Tool is used for coding and development of the project. |
| SQL | Tool used for communicating with the traditional database. |
| SOLR- LUCENE | Tool used for Big Data storage |
| Azure | Tool used for hosting the project in the cloud |
| ArcGIS | Tool used for Geospatial information |

## References

The documents referred to in the preparation of this document or those on which this document is based, are as follows:

Software Requirement Specification

Project Management Plan

Build Plan

## Assumptions

The following assumptions are made during this design:

The coding is performed using .NET programming language and in the MVC framework.

The code is compatible with the hosting environment.

## Design Constraints

The following are the constraints considered while preparing this document:

External integration requirement (for open architecture)

Customer specific architecture

# Design Approach/ System Overview

The design approach includes overview of the system that is being architected. This section provides the information needed for a system development team to build and integrate the code and integrate the software modules into a functional product. The design approach chosen for this project is object-oriented approach. Hence, the features such as Search, Dashboard, Resident, Task, List, Mapping and Electronic Check-in are integrated along with the Geographic Information System.

# Alternative Design Approaches Considered

The different alternatives considered in the design approach includes functional-oriented approach. The following are the advantages of choosing object-oriented programming. It provides a great benefit in designing large programs, which can be easily divided into smaller parts and helps in distinguishing the components or phases that need to be executed or planned in a certain way. Object-oriented programming based on the main features that are: 1. *Abstraction*: It helps in letting the useful information or relevant data to a user, which increases the efficiency of the program and make the things simple. 2. *Inheritance*. It helps in inheriting the methods, functions, properties, and fields of a base class in derived class. 3. *Polymorphism*: It helps in doing one task in many ways with help of overloading and overriding which is also known as compile time and run time polymorphism respectively. 4. *Encapsulation:*It helps in hiding the irrelevant data from a user and prevents the user from unauthorized access. Since data integrity is the important feature of the application, encapsulation feature of the approach assists in maintaining high level security of the data.

# Module Design

The modules are accompanied by a UML class diagram. The class diagram contains the important classes and the classes used in interacting with the other modules.

 Login

Userid : string

Name: string

Address: string

Contact: string

Location: string

Email: string

viewAnswers()

manageData()

viewReports()

 Login

User name: string

Password: string

authenticate()

 Administrator

User name: string

Password: string

Create ()

Update ()

Delete ()

View ()

Data

DataId : Number

View ()

Campaign

Manager

User name: string

Password: string

Name: string

Address: string

Contact: string

Subject : string

Email: string

Attendance()

View ()

Volunteer

VolunteerId : String VolunteerName: string

VolunteerLocation: string

warehouseAddress : String

warehouseManagerId : String

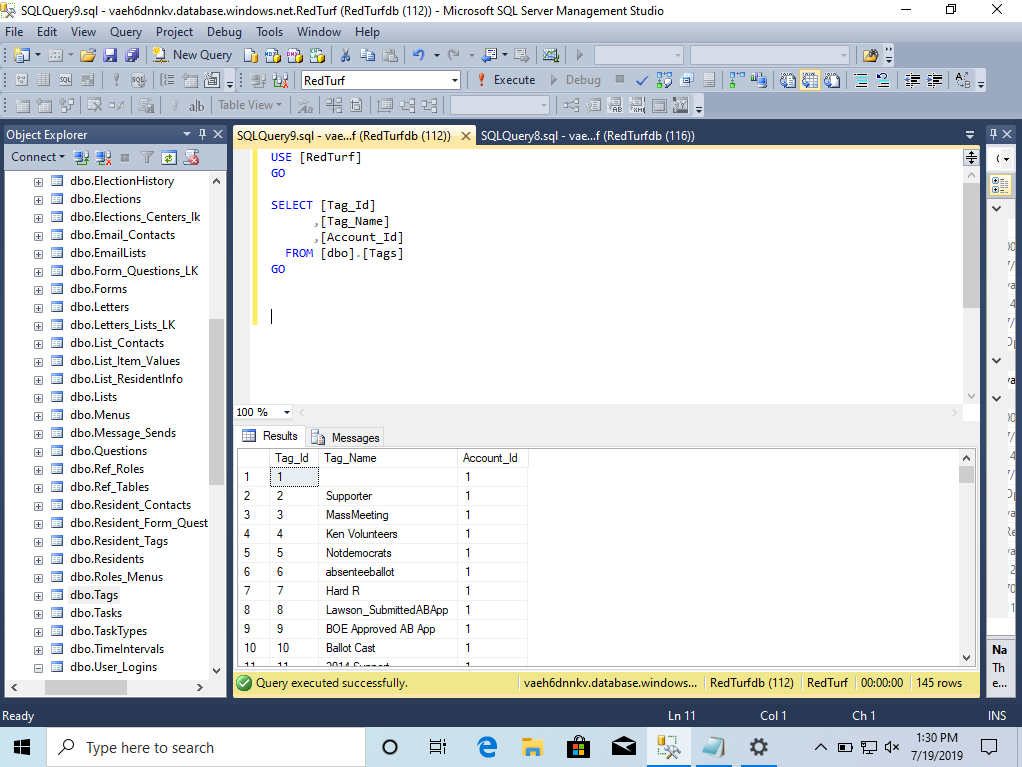
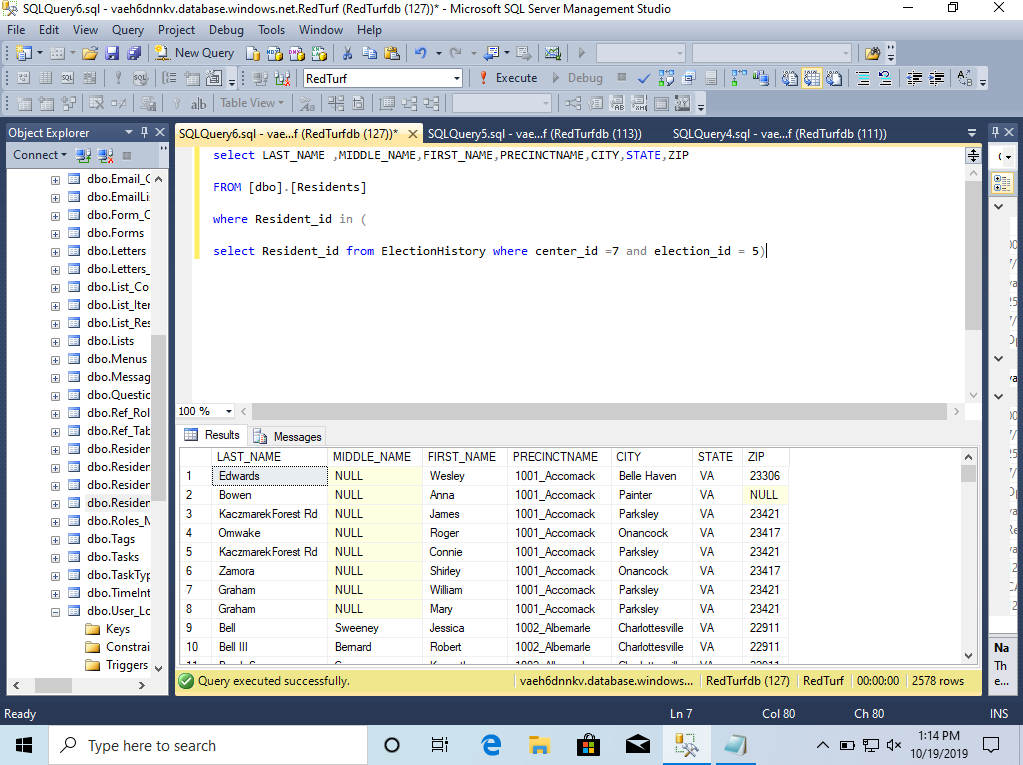
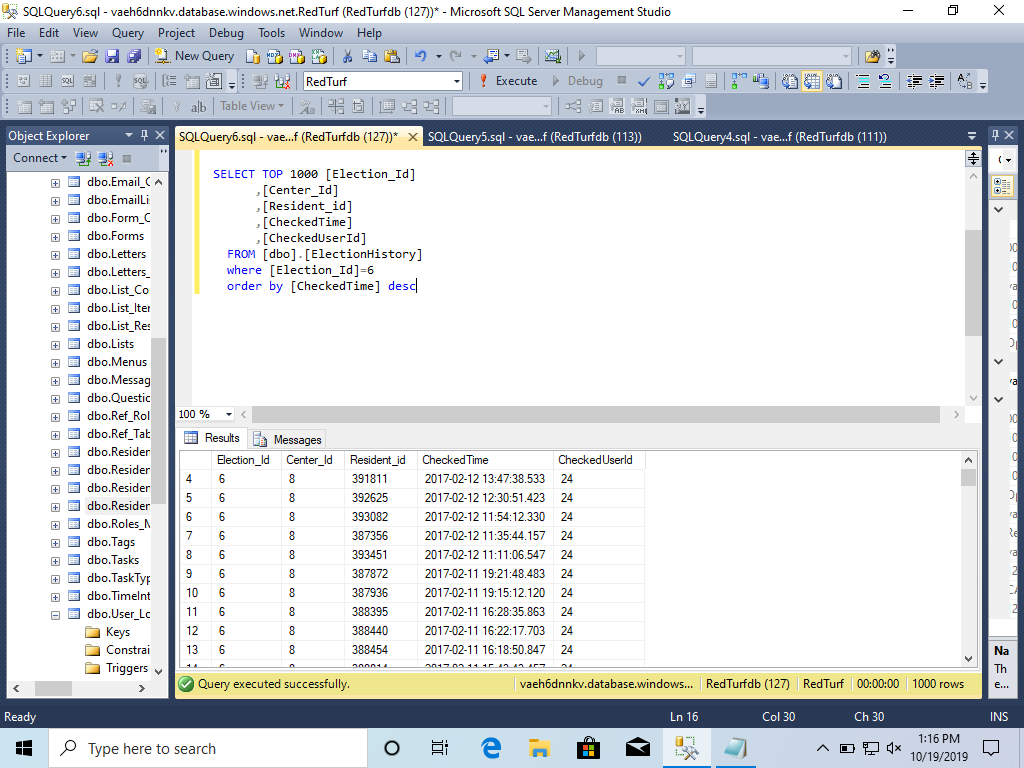
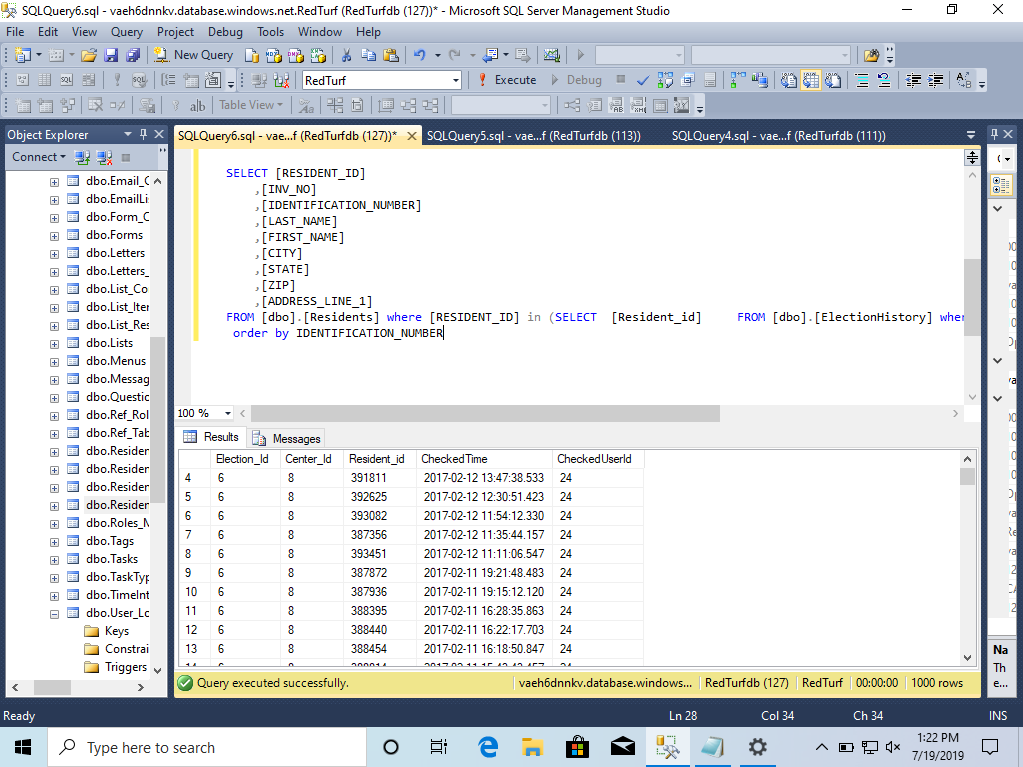
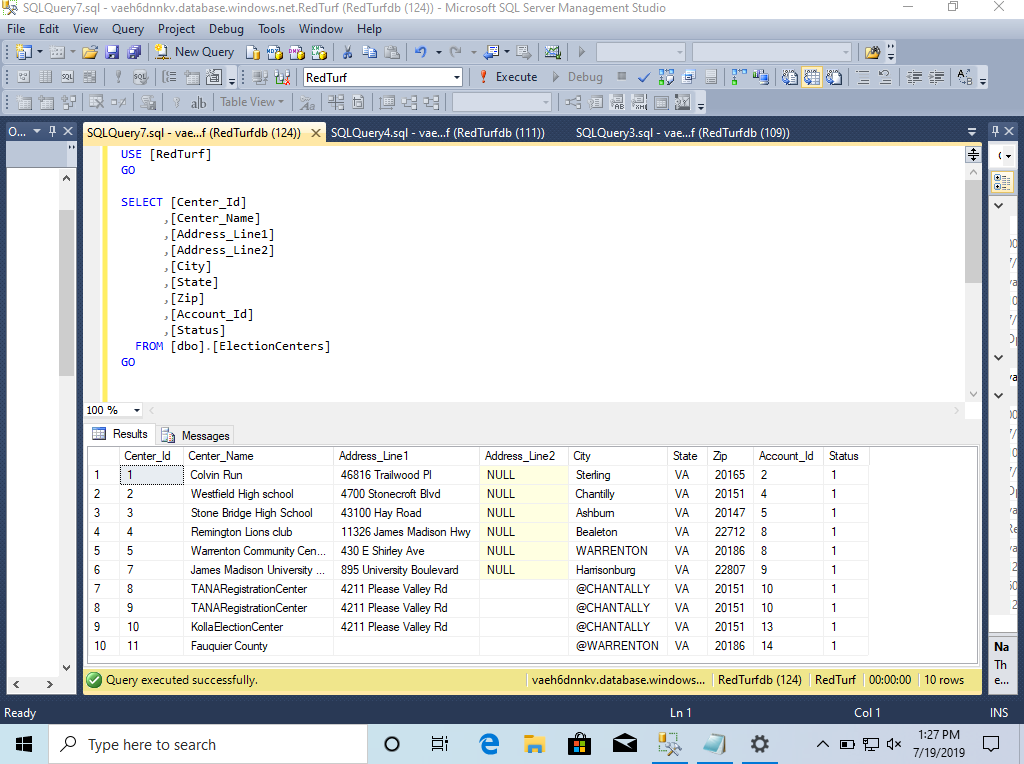
 Reports

View ()

Charts ()

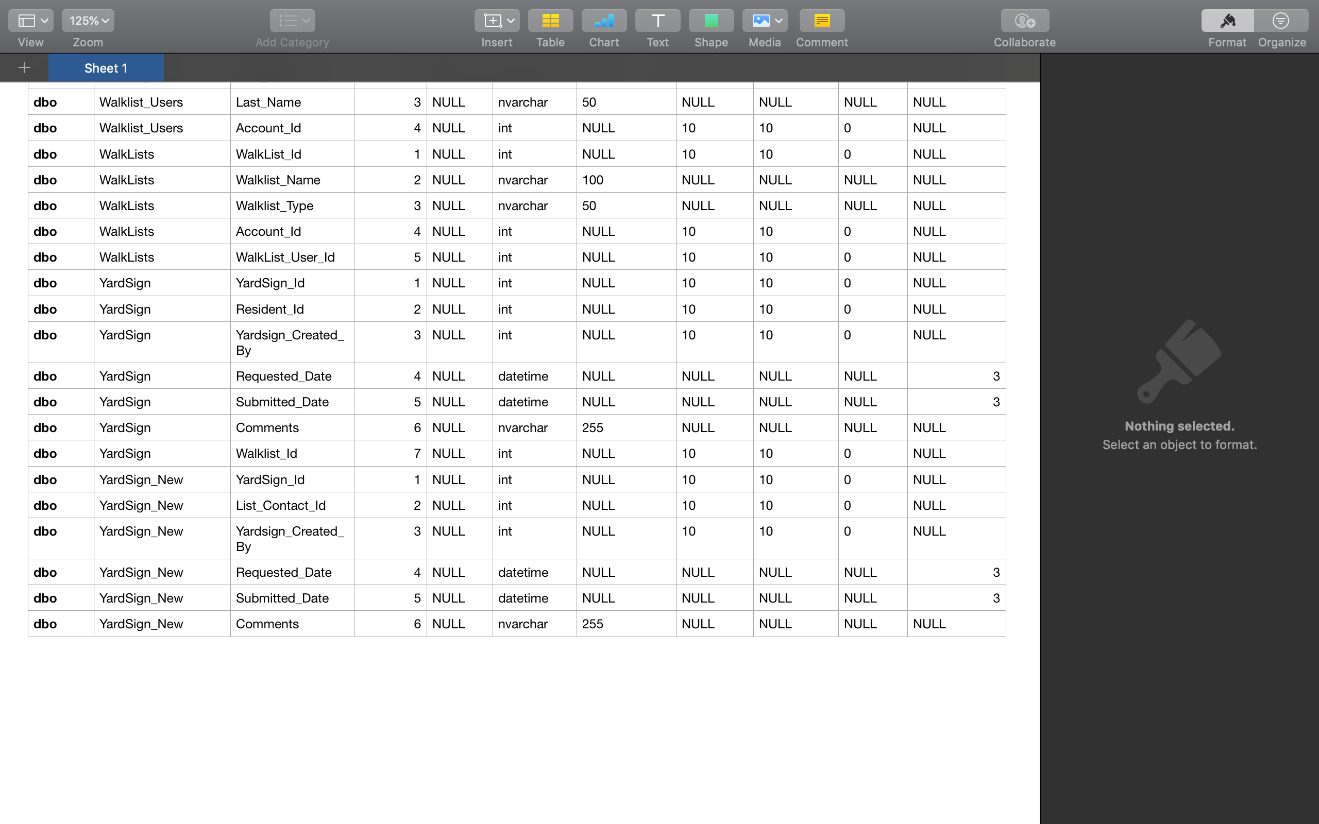
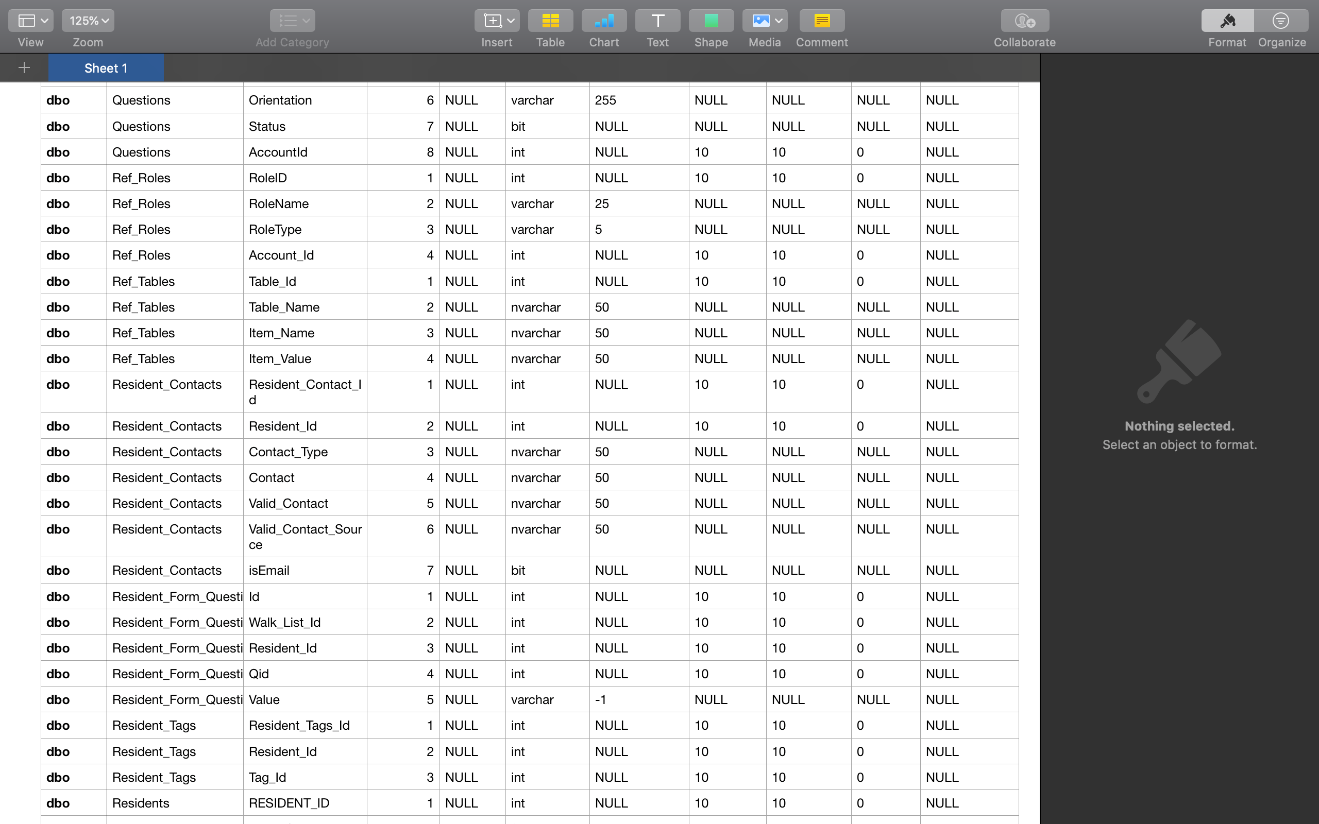
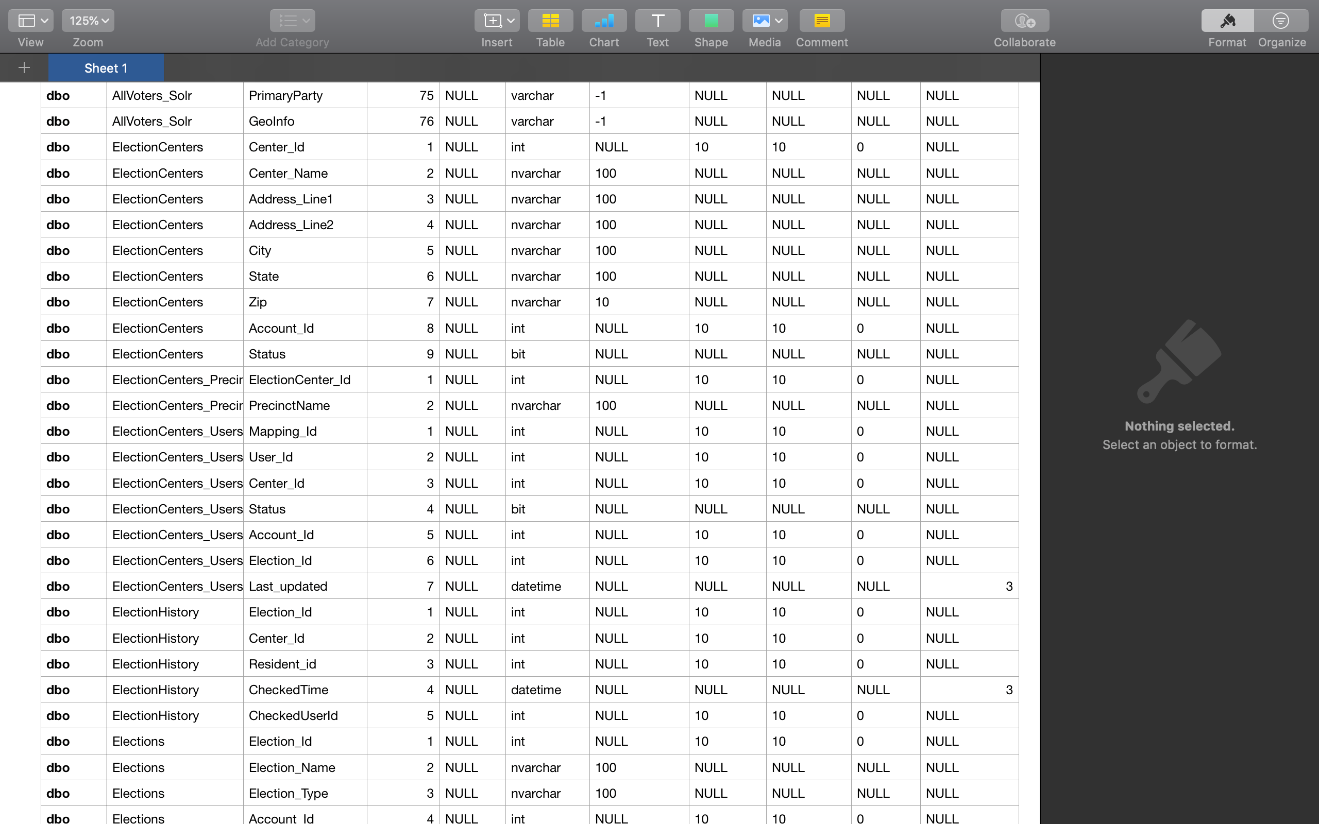
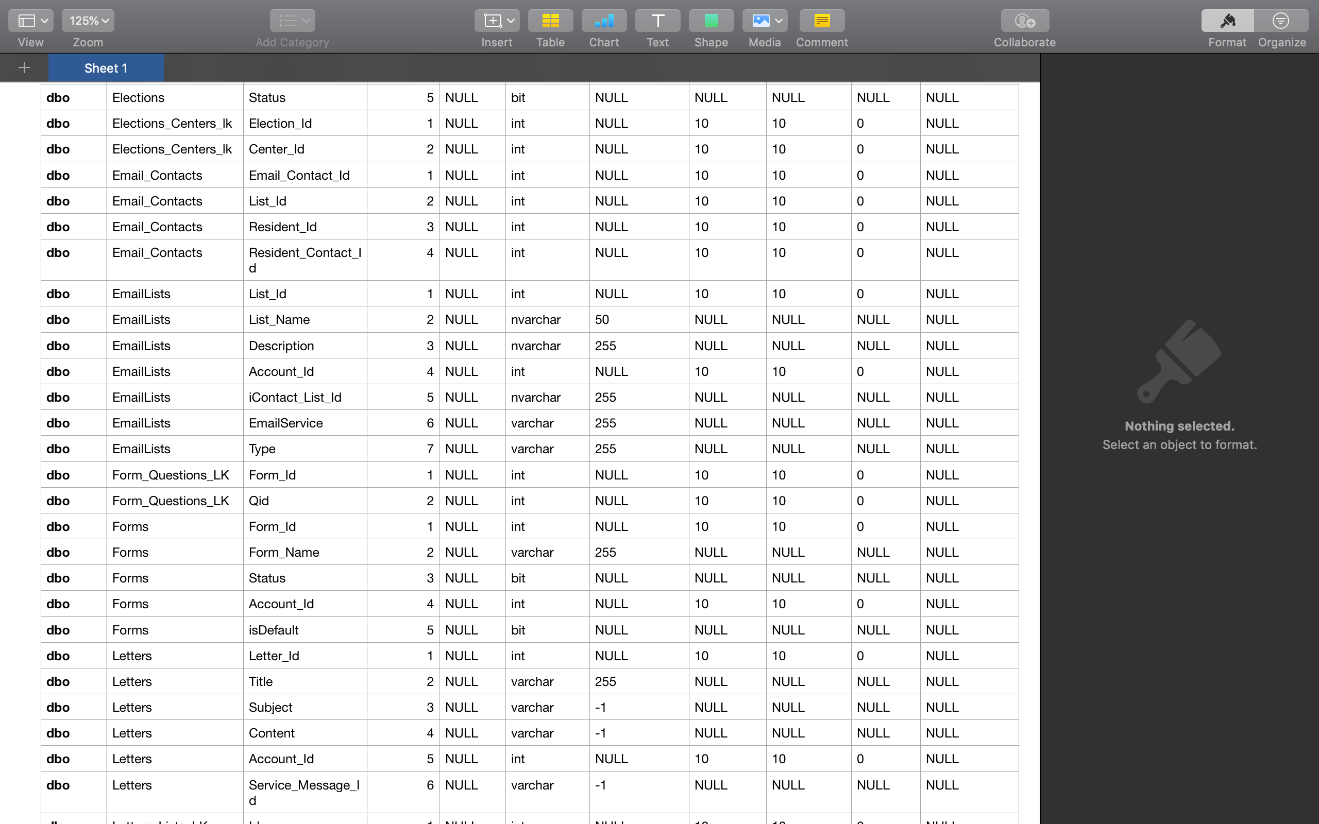
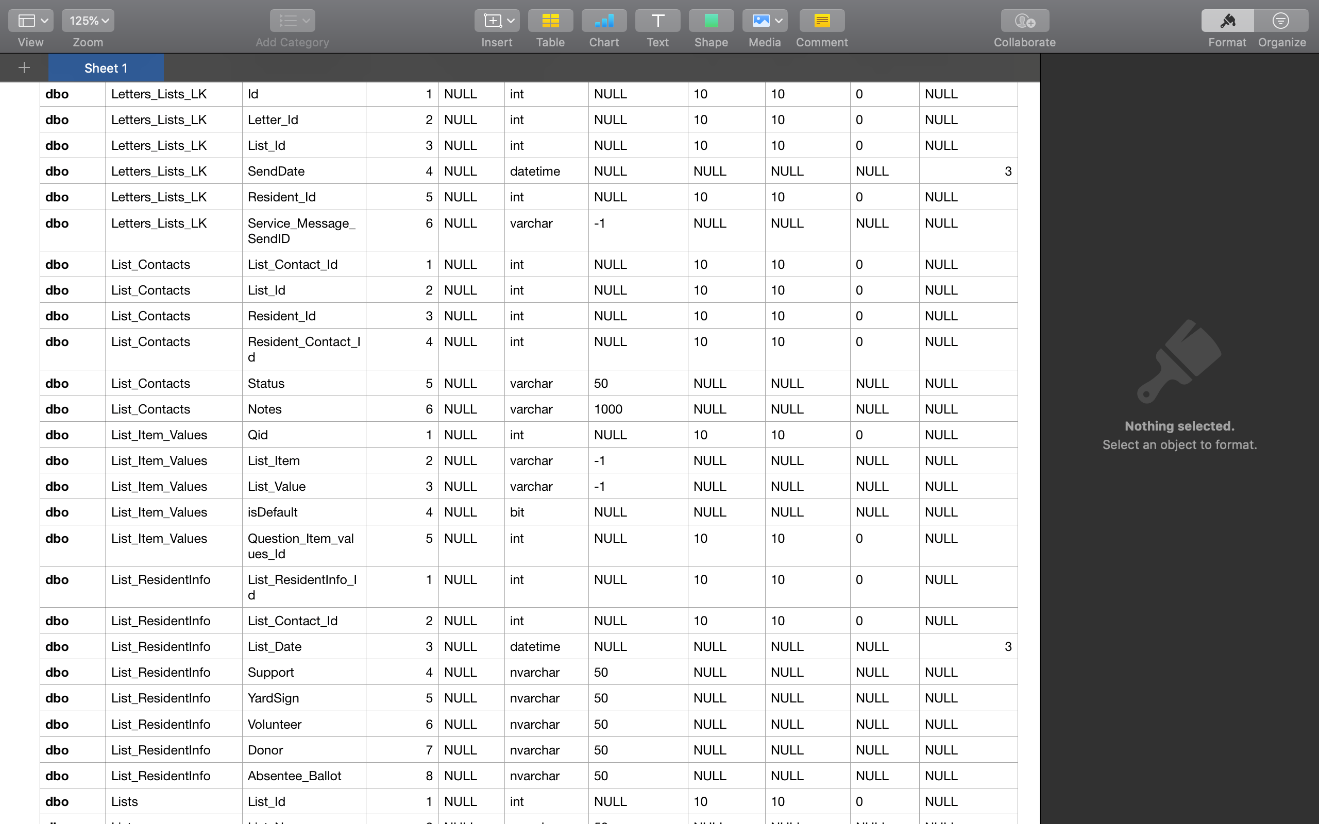
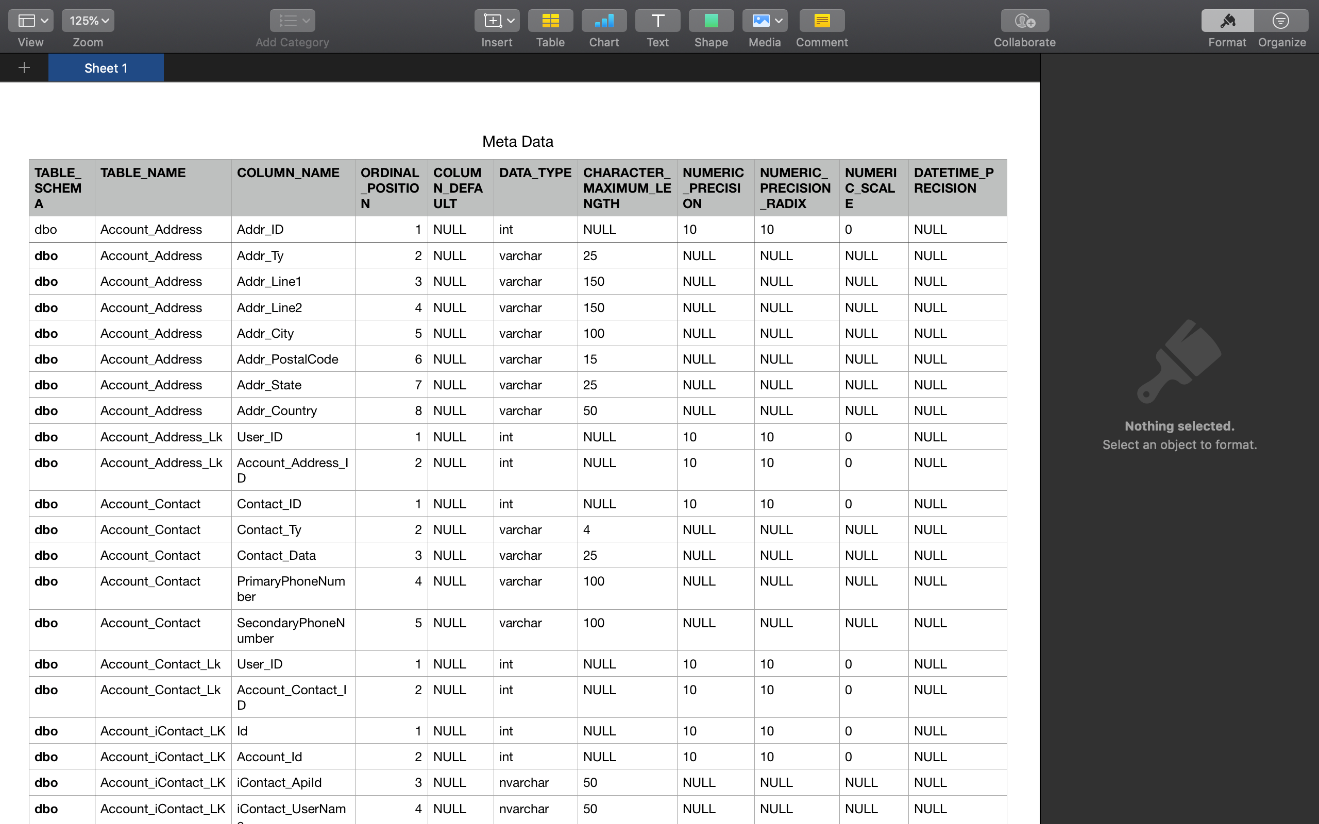
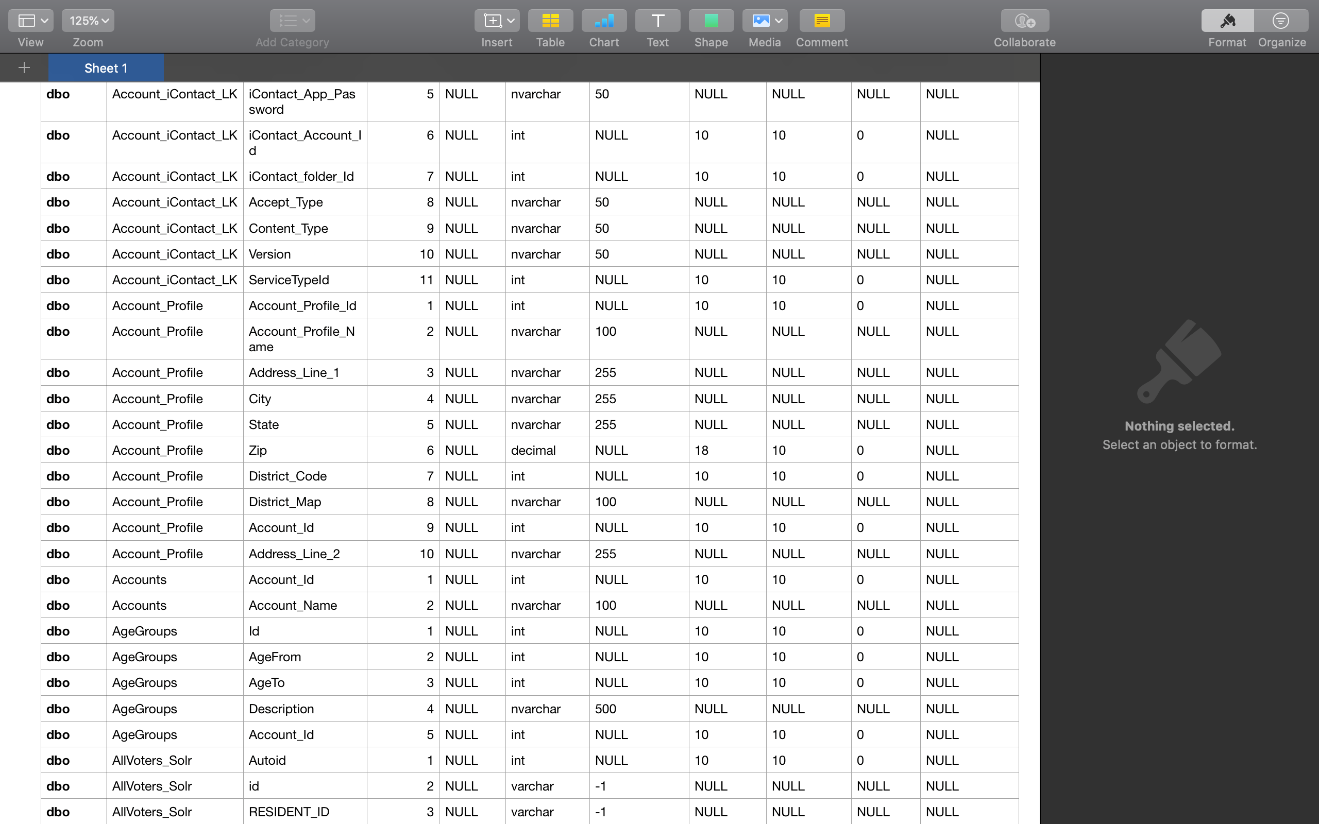
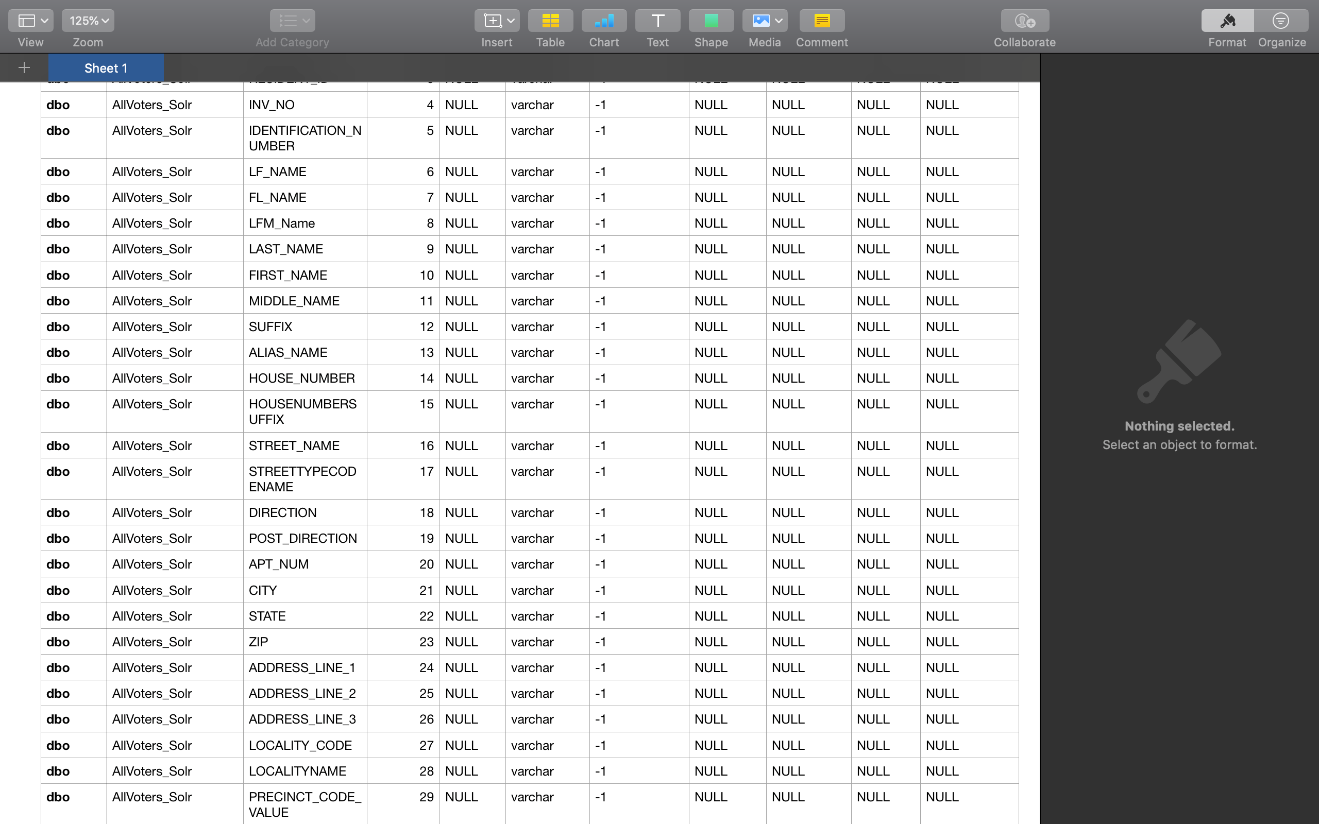
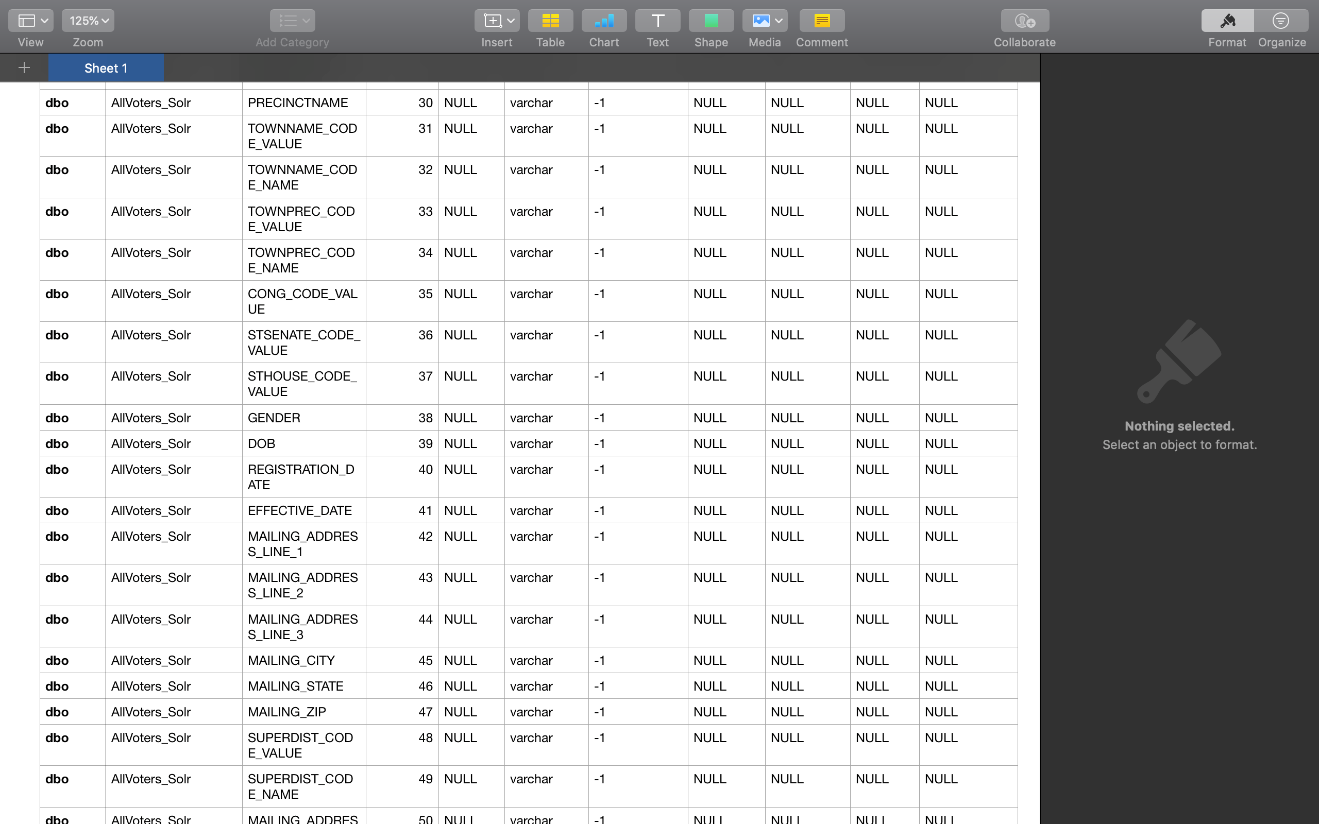
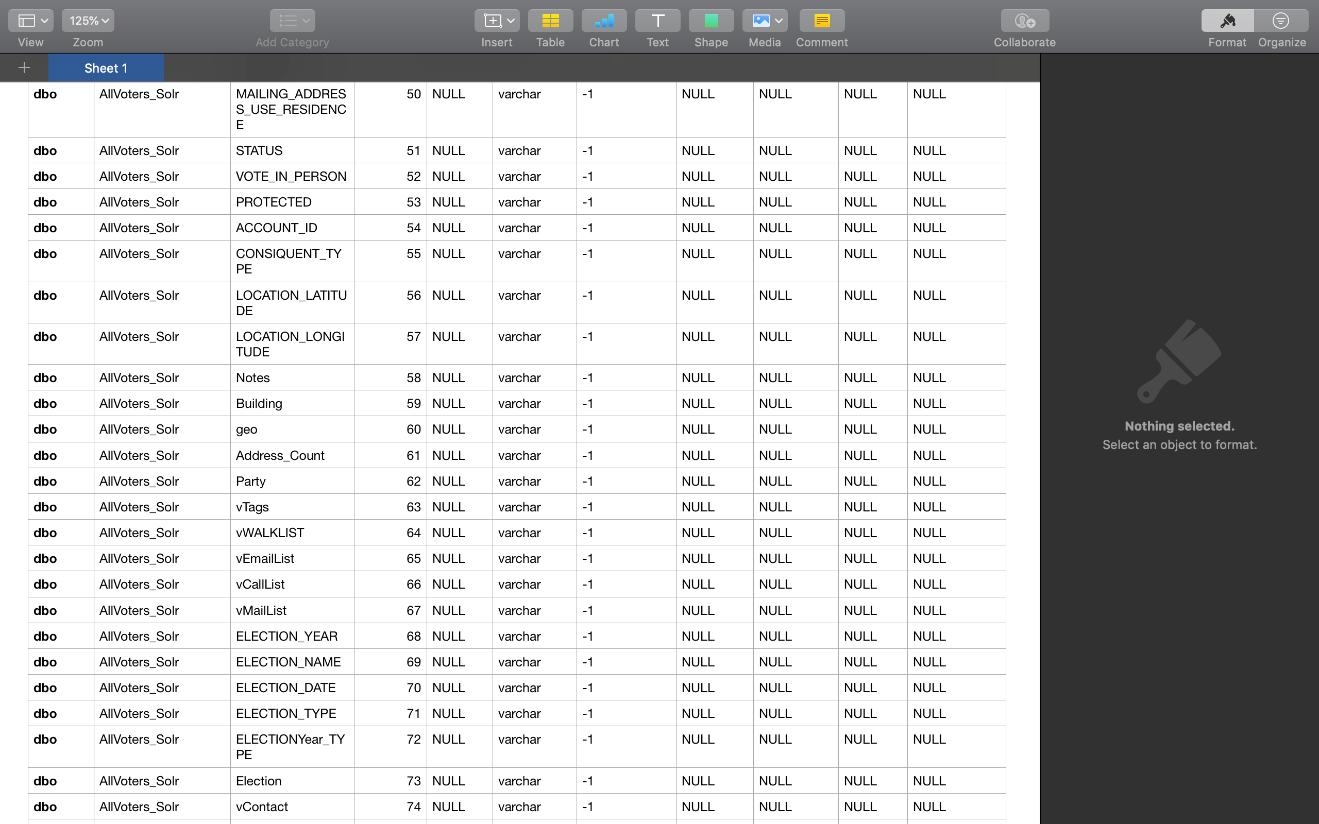
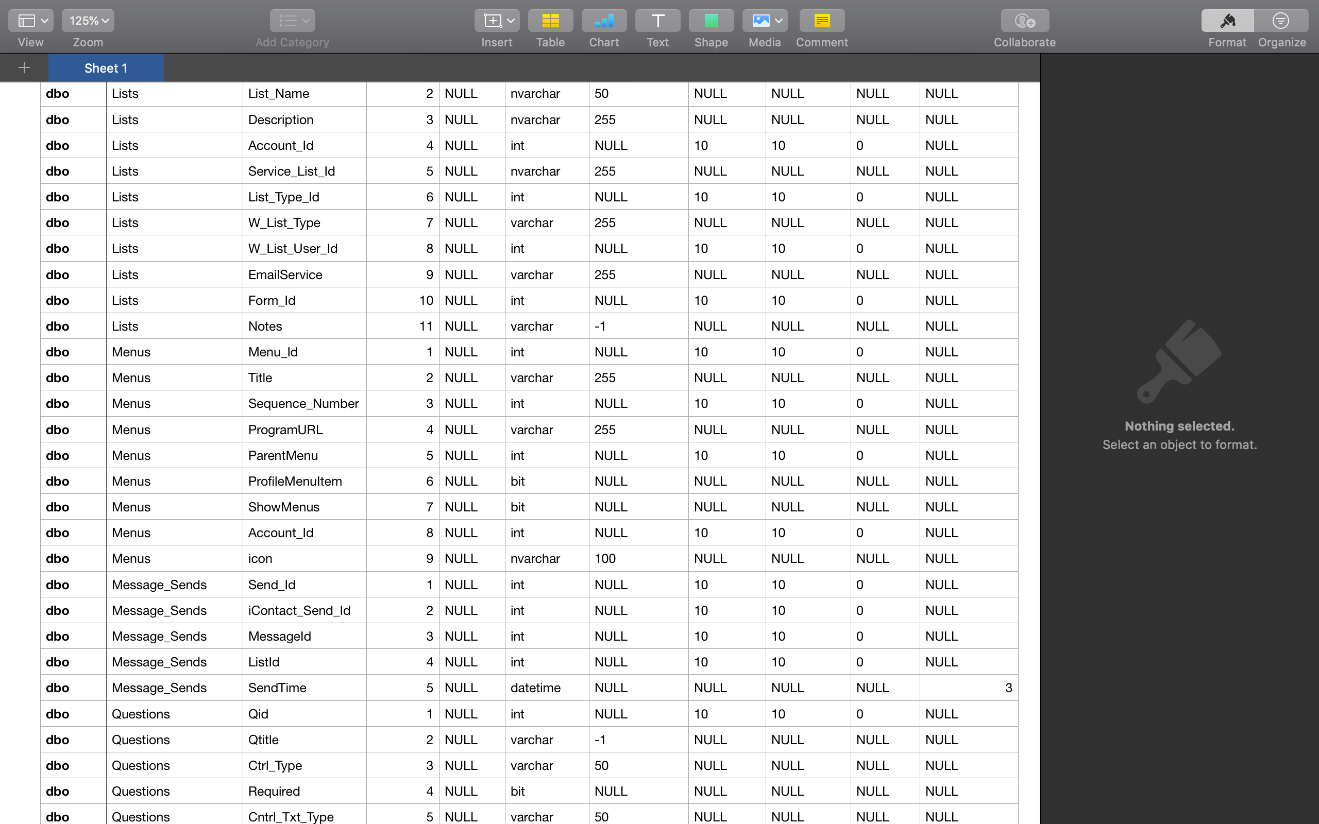
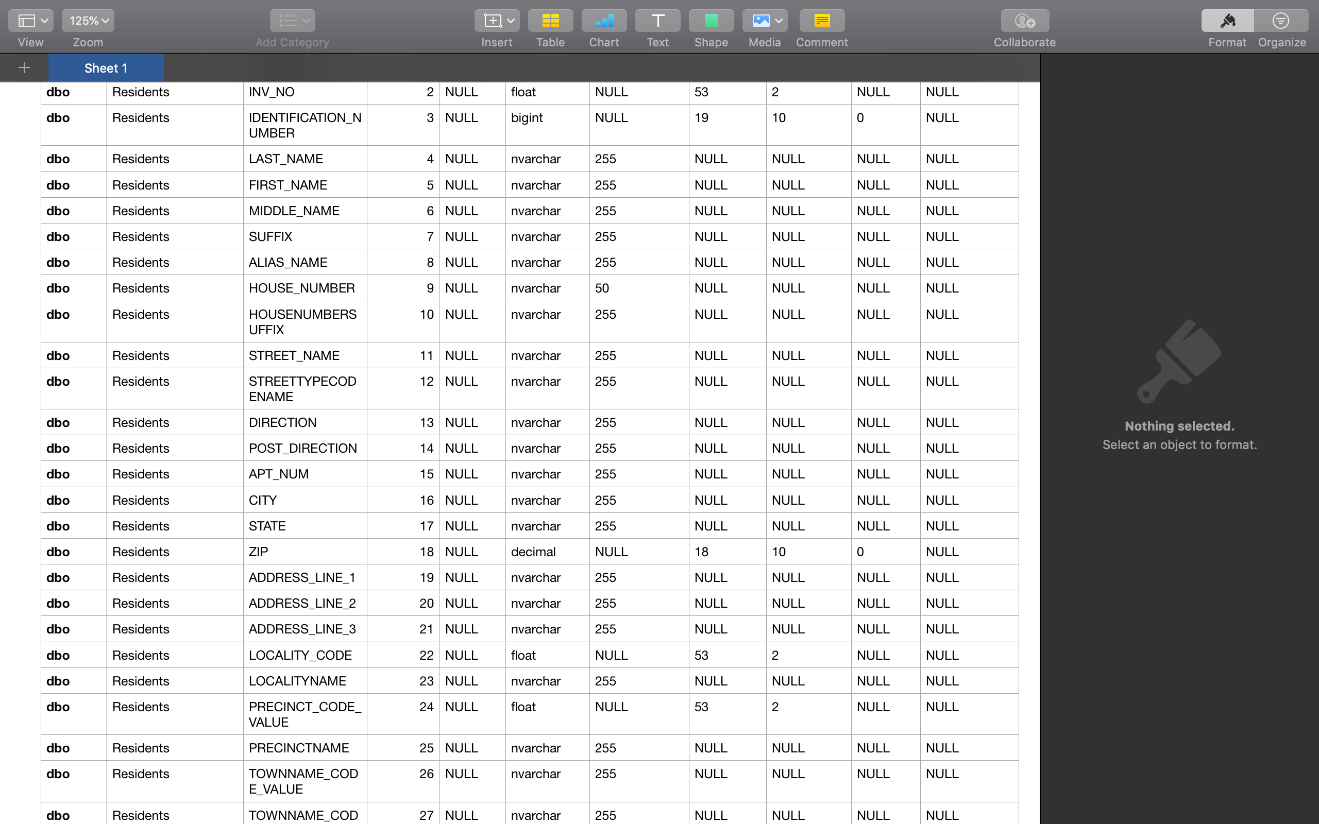
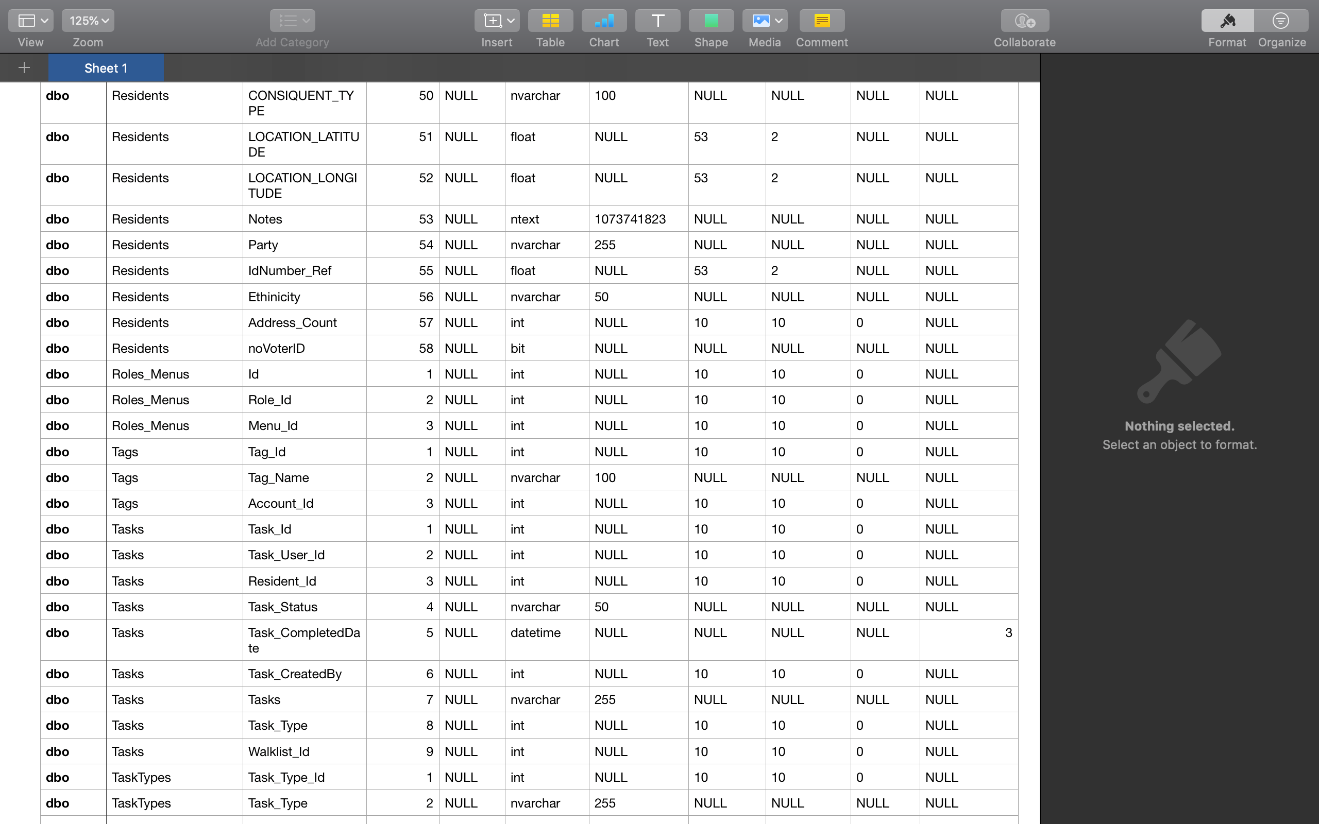
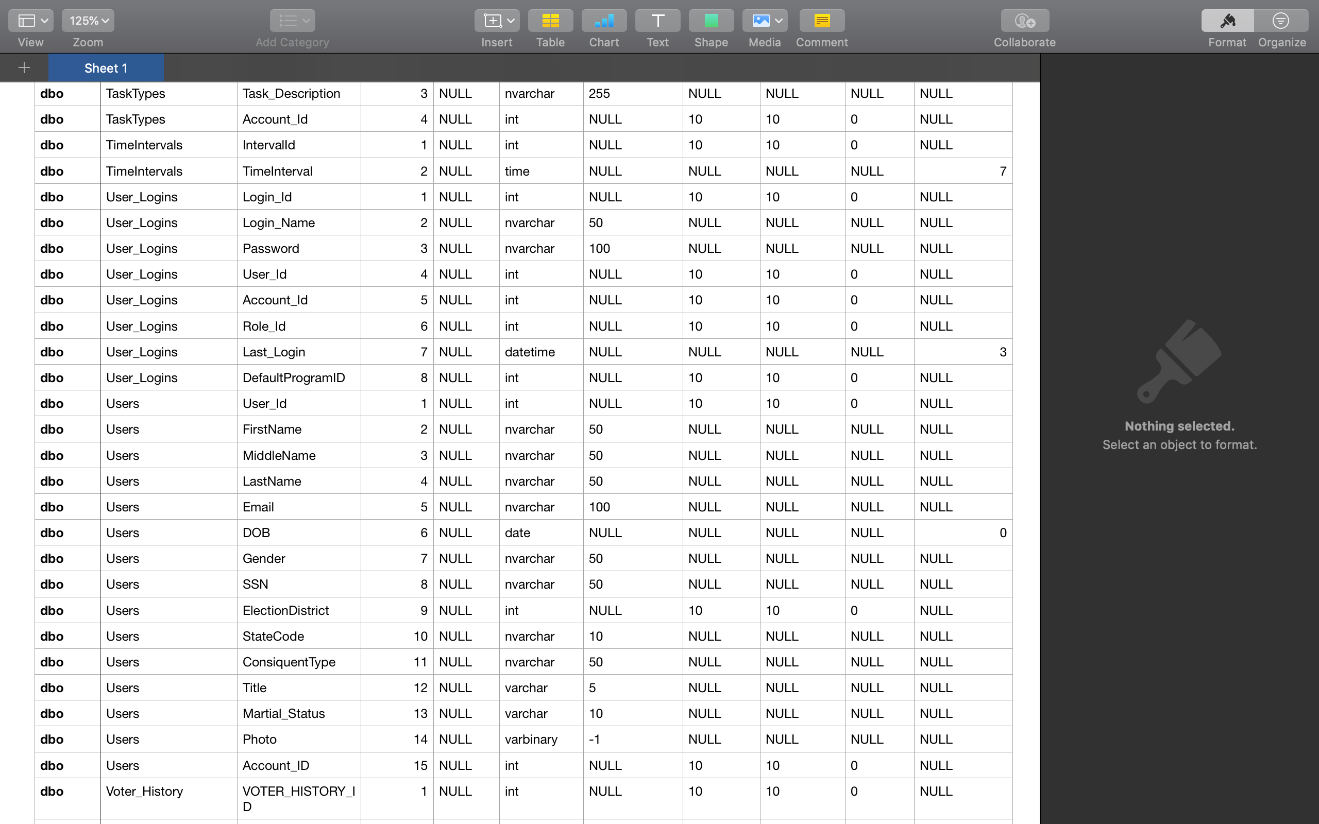
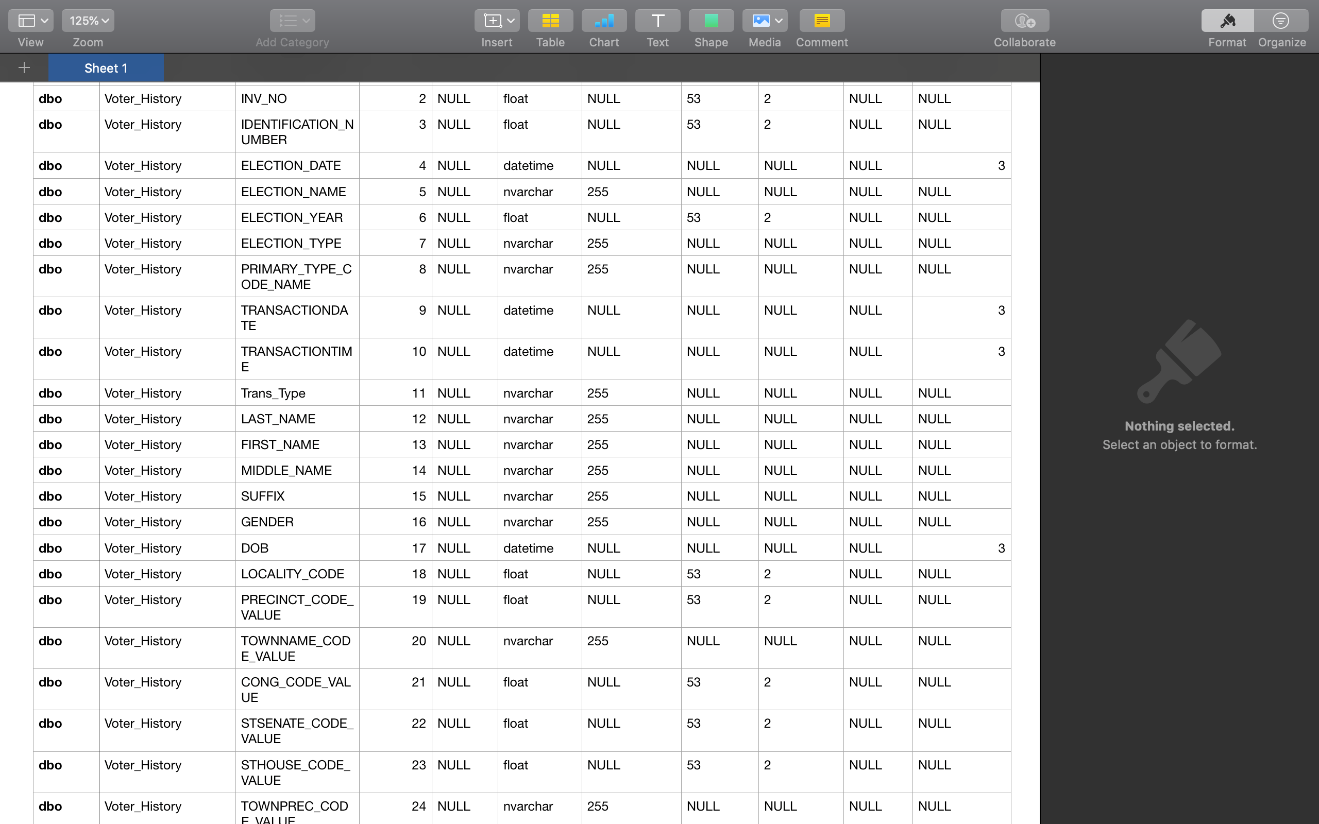
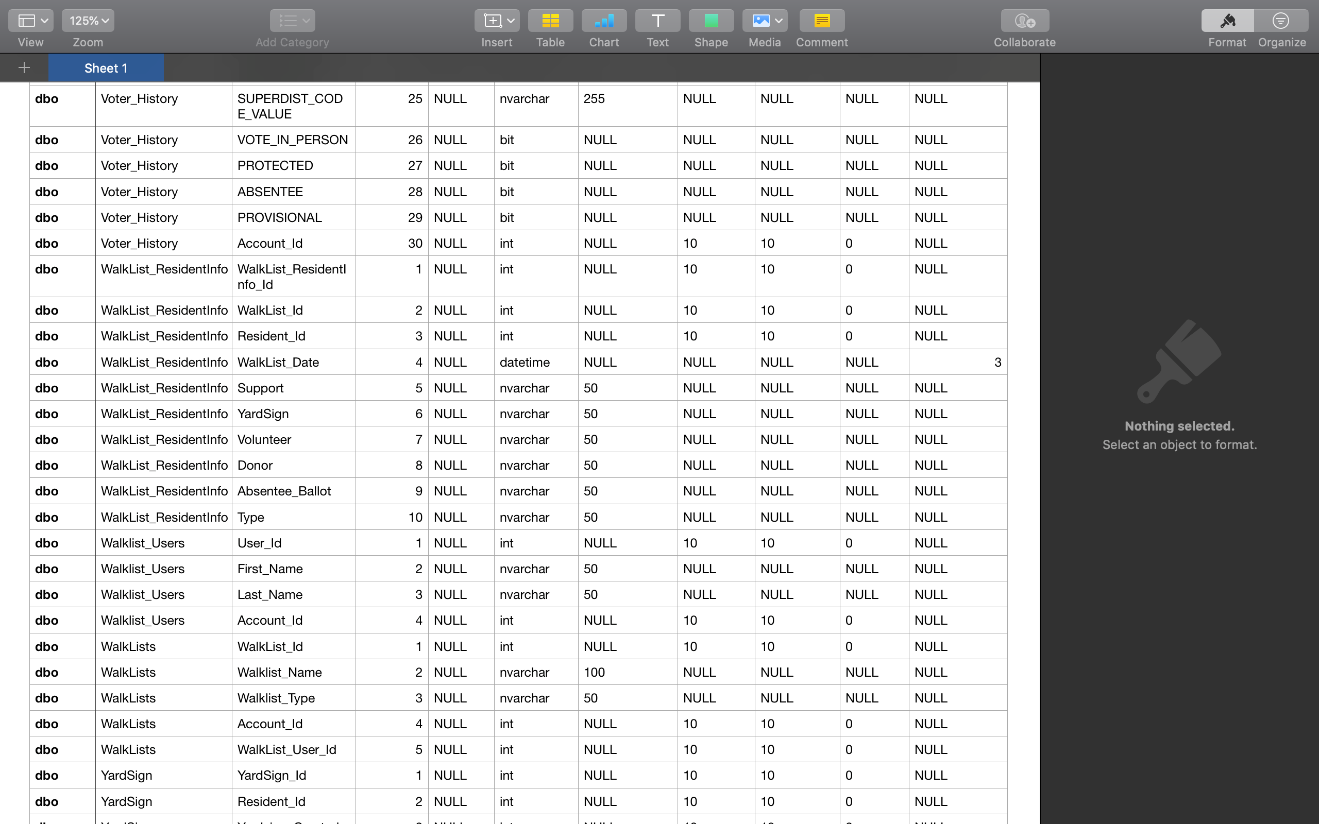
Close ()

### Class 1



## Database Design

The back-end database is a relational SQL database, this section describes all the tables that are maintained during the process lifetime.



## Error & Exception Handling

|  |  |
| --- | --- |
| Error Code | Error Description |
| HTTP 200 | Successful |
| HTTP 300 | Resource Temporarily not available |
| HTTP 400 | Page cannot be displayed |
| HTTP 500 | Internal Server Error |
| NUL Point Exception | No reference |
| SERVLET Exception | SERVLET failure |
| SQL Exception | SQL Error |
| I/O Exception | I/O Error |
| Array\_Index\_Outbound | Array Size Issue |

# Unit Test Recommendations

Refer Requirement Traceability Matrix

# Traceability Matrix

Refer the updated Traceability matrix. Traceability matrix is updated with High Level Design references.